Open Source Software Information

General Information

This media, software or hardware (“Product”) obtained from Extreme Networks, Inc. (“Extreme Networks”) may include Extreme Networks Software, Third Party Software (defined below), and/or Open Source Software (defined below).

The object code or source code (collectively, the “Software”) included with the Product is the exclusive property of Extreme Networks or its licensors, and any use is subject to the terms and conditions of one or more agreements in force between the purchaser of the Extreme Networks Product or licensee of the Extreme Networks Software and Extreme Networks. SOFTWARE IS PROTECTED BY U.S. COPYRIGHT LAWS AND INTERNATIONAL LAWS AND TREATIES. UNAUTHORIZED COPYING, DISTRIBUTION OR OTHER USE OF THIS PRODUCT IS STRICTLY PROHIBITED. ANY DISTRIBUTION OR USE NOT SPECIFICALLY APPROVED BY EXTREME NETWORKS IS STRICTLY PROHIBITED.

Extreme Networks Software is subject to the commercial terms and conditions entered into with Extreme Networks, any other use is strictly prohibited. Commercial Software licensed for redistribution by Extreme Networks (“Third Party Software”) is subject to the terms and conditions in force between Extreme Networks and the licensor of the Third party Software. The terms and conditions governing the usage of Third Party Software may be part of the agreement entered into by the purchaser of the Product with Extreme Networks or separate Third Party Software license agreement(s) included with the Product.

Software provided by Extreme Networks that is covered by a publicly available license governed solely under Copyright law, in which the complete terms and obligations of such license attach to a licensee solely through the act of copying, using and/or distribution of the licensed Software, and in which such obligations often include one or more of attribution obligations, distribution obligations, copyleft obligations, and intellectual property encumbrances is referred to herein as “Open Source Software”. The use of any Open Source Software is subject to the licenses, terms and conditions of the commercial agreement in force between the purchaser of the Product and Extreme Networks as well as the terms and conditions of the corresponding license of each Open Source Software package. If there is a conflict between the terms and conditions of any commercial agreement and the terms and conditions of the Open Source Software license, the applicable Open Source Software license will take precedence. Copies of the licenses for the included Open Source Software as well as their attributions, acknowledgements, and software information details, are listed below. Extreme Networks is required to reproduce the software licenses, acknowledgements and copyright notices as provided by the authors and owners, thus, all such information is provided in its native language form, without modification or translation.

For instructions on how to obtain a copy of any source code made publicly available by Extreme Networks related to Open Source Software distributed by Extreme Networks, you may send your request in writing to:

```
Extreme Networks, Inc.

Legal Department

6480 Via Del Oro

San Jose, CA, 95119 USA
```

In your request, please include the Extreme Networks Product name and version, along with the Open Source Software specifics, such as the Open Source Software name, license, and version.

Note: The source code for the Open Source Software included by Extreme Networks may be resident on the Product’s installation media provided with the Product or on supplemental Product media or subsequent download. Please reference and review the entire Extreme Networks Open Source Software “Legal Notices” and any corresponding “End User License Agreement” provided with the Product or the commercial agreement under which the Product was purchased for details on the availability, location and method for obtaining source code. Further, depending on the license terms of the specific Open Source Software, source code may not be provided. Please reference and review the entire Extreme Networks Open Source Software "Legal Notices" and "End User License Agreement" to identify which Open Source Software packages have source code provided or available.

Notices for packages distributed in this release:

Notice for package(s)

acl
attr
bc
glibc-external
gnutls
kmod
libcheck
libmnl
libtool
liburcu
libusb1
lttng-modules
lttng-tools

lttng-ust
quota
sysfsutils
udev
util-linux
xz

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Soft

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave yo

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permissi

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Publi

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combin

We call this license the "Lesser" General Public License because it does Less to protect the user`s freedom than the ordinary General Public I

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free sof

Although the Lesser General Public License is Less protective of the users` freedom, it does ensure that the user of a program that is linked

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work base

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other aut

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library"

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all th

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a

1. You may copy and distribute verbatim copies of the Library`s complete source code as you receive it, in any medium, provided that you consp

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute s

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facili
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Ther

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be r

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to ex

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do thi

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subseque

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the ter

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked w

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contain

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a deriva

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any execu

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work conta

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the l
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, ab
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the ope

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library faciliti

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must b
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find th
8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt oth

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distrib

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the origina

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), cond

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to a

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the origi

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new vers

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that ever

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effect

one line to give the library`s name and an idea of what it does.
Copyright (C) year name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if n

Yoyodyne, Inc., hereby disclaims all copyright interest in
the library `Frob` (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
That`s all there is to it!

Notice for package(s)

zlib

```
/* zlib.h -- interface of the 'zlib' general purpose compression library
  version 1.2.8, April 28th, 2013

  Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler

  This software is provided 'as-is', without any express or implied
  warranty.  In no event will the authors be held liable for any damages
  arising from the use of this software.

  Permission is granted to anyone to use this software for any purpose,
  including commercial applications, and to alter it and redistribute it
  freely, subject to the following restrictions:

  1. The origin of this software must not be misrepresented; you must not
     claim that you wrote the original software. If you use this software
     in a product, an acknowledgment in the product documentation would be
     appreciated but is not required.
  2. Altered source versions must be plainly marked as such, and must not be
     misrepresented as being the original software.
  3. This notice may not be removed or altered from any source distribution.

  Jean-loup Gailly        Mark Adler
  jloup@gzip.org          madler@alumni.caltech.edu


  The data format used by the zlib library is described by RFCs (Request for
  Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950
  (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).
*/

#ifndef ZLIB_H
#define ZLIB_H

#include "zconf.h"

#ifdef __cplusplus
extern "C" {
#endif

#define ZLIB_VERSION "1.2.8"
#define ZLIB_VERNUM 0x1280
#define ZLIB_VER_MAJOR 1
#define ZLIB_VER_MINOR 2
#define ZLIB_VER_REVISION 8
#define ZLIB_VER_SUBREVISION 0

/*
    The 'zlib' compression library provides in-memory compression and
  decompression functions, including integrity checks of the uncompressed data.
  This version of the library supports only one compression method (deflation)
  but other algorithms will be added later and will have the same stream
  interface.

    Compression can be done in a single step if the buffers are large enough,
  or can be done by repeated calls of the compression function.  In the latter
  case, the application must provide more input and/or consume the output
  (providing more output space) before each call.

    The compressed data format used by default by the in-memory functions is
  the zlib format, which is a zlib wrapper documented in RFC 1950, wrapped
  around a deflate stream, which is itself documented in RFC 1951.

    The library also supports reading and writing files in gzip (.gz) format
  with an interface similar to that of stdio using the functions that start
  with "gz".  The gzip format is different from the zlib format.  gzip is a
  gzip wrapper, documented in RFC 1952, wrapped around a deflate stream.

    This library can optionally read and write gzip streams in memory as well.

    The zlib format was designed to be compact and fast for use in memory
  and on communications channels.  The gzip format was designed for single-
  file compression on file systems, has a larger header than zlib to maintain
  directory information, and uses a different, slower check method than zlib.

    The library does not install any signal handler.  The decoder checks
  the consistency of the compressed data, so the library should never crash
  even in case of corrupted input.
*/

typedef voidpf (*alloc_func) OF((voidpf opaque, uInt items, uInt size));
typedef void   (*free_func)  OF((voidpf opaque, voidpf address));

struct internal_state;

typedef struct z_stream_s {
    z_const Bytef *next_in;     /* next input byte */
    uInt     avail_in;  /* number of bytes available at next_in */
    uLong    total_in;  /* total number of input bytes read so far */

    Bytef    *next_out; /* next output byte should be put there */
    uInt     avail_out; /* remaining free space at next_out */
    uLong    total_out; /* total number of bytes output so far */

    z_const char *msg;  /* last error message, NULL if no error */
    struct internal_state FAR *state; /* not visible by applications */

    alloc_func zalloc;  /* used to allocate the internal state */
```

```
    free_func  zfree;   /* used to free the internal state */
    voidpf     opaque;  /* private data object passed to zalloc and zfree */

    int     data_type;  /* best guess about the data type: binary or text */
    uLong   adler;      /* adler32 value of the uncompressed data */
    uLong   reserved;   /* reserved for future use */
} z_stream;

typedef z_stream FAR *z_streamp;

/*
     gzip header information passed to and from zlib routines.  See RFC 1952
  for more details on the meanings of these fields.
*/
typedef struct gz_header_s {
    int     text;       /* true if compressed data believed to be text */
    uLong   time;       /* modification time */
    int     xflags;     /* extra flags (not used when writing a gzip file) */
    int     os;         /* operating system */
    Bytef   *extra;     /* pointer to extra field or Z_NULL if none */
    uInt    extra_len;  /* extra field length (valid if extra != Z_NULL) */
    uInt    extra_max;  /* space at extra (only when reading header) */
    Bytef   *name;      /* pointer to zero-terminated file name or Z_NULL */
    uInt    name_max;   /* space at name (only when reading header) */
    Bytef   *comment;   /* pointer to zero-terminated comment or Z_NULL */
    uInt    comm_max;   /* space at comment (only when reading header) */
    int     hcrc;       /* true if there was or will be a header crc */
    int     done;       /* true when done reading gzip header (not used
                           when writing a gzip file) */
} gz_header;

typedef gz_header FAR *gz_headerp;

/*
     The application must update next_in and avail_in when avail_in has dropped
   to zero.  It must update next_out and avail_out when avail_out has dropped
   to zero.  The application must initialize zalloc, zfree and opaque before
   calling the init function.  All other fields are set by the compression
   library and must not be updated by the application.

     The opaque value provided by the application will be passed as the first
   parameter for calls of zalloc and zfree.  This can be useful for custom
   memory management.  The compression library attaches no meaning to the
   opaque value.

     zalloc must return Z_NULL if there is not enough memory for the object.
   If zlib is used in a multi-threaded application, zalloc and zfree must be
   thread safe.

     On 16-bit systems, the functions zalloc and zfree must be able to allocate
   exactly 65536 bytes, but will not be required to allocate more than this if
   the symbol MAXSEG_64K is defined (see zconf.h).  WARNING: On MSDOS, pointers
   returned by zalloc for objects of exactly 65536 bytes *must* have their
   offset normalized to zero.  The default allocation function provided by this
   library ensures this (see zutil.c).  To reduce memory requirements and avoid
   any allocation of 64K objects, at the expense of compression ratio, compile
   the library with -DMAX_WBITS=14 (see zconf.h).

     The fields total_in and total_out can be used for statistics or progress
   reports.  After compression, total_in holds the total size of the
   uncompressed data and may be saved for use in the decompressor (particularly
   if the decompressor wants to decompress everything in a single step).
*/

                        /* constants */

#define Z_NO_FLUSH      0
#define Z_PARTIAL_FLUSH 1
#define Z_SYNC_FLUSH    2
#define Z_FULL_FLUSH    3
#define Z_FINISH        4
#define Z_BLOCK         5
#define Z_TREES         6
/* Allowed flush values; see deflate() and inflate() below for details */

#define Z_OK            0
#define Z_STREAM_END    1
#define Z_NEED_DICT     2
#define Z_ERRNO        (-1)
#define Z_STREAM_ERROR (-2)
#define Z_DATA_ERROR   (-3)
#define Z_MEM_ERROR    (-4)
#define Z_BUF_ERROR    (-5)
#define Z_VERSION_ERROR (-6)
/* Return codes for the compression/decompression functions. Negative values
 * are errors, positive values are used for special but normal events.
 */

#define Z_NO_COMPRESSION         0
#define Z_BEST_SPEED             1
#define Z_BEST_COMPRESSION       9
#define Z_DEFAULT_COMPRESSION  (-1)
/* compression levels */

#define Z_FILTERED            1
#define Z_HUFFMAN_ONLY        2
#define Z_RLE                 3
#define Z_FIXED               4
```

```
#define Z_DEFAULT_STRATEGY    0
/* compression strategy; see deflateInit2() below for details */

#define Z_BINARY   0
#define Z_TEXT     1
#define Z_ASCII    Z_TEXT   /* for compatibility with 1.2.2 and earlier */
#define Z_UNKNOWN  2
/* Possible values of the data_type field (though see inflate()) */

#define Z_DEFLATED   8
/* The deflate compression method (the only one supported in this version) */

#define Z_NULL  0  /* for initializing zalloc, zfree, opaque */

#define zlib_version zlibVersion()
/* for compatibility with versions < 1.0.2 */


                        /* basic functions */

ZEXTERN const char * ZEXPORT zlibVersion OF((void));
/* The application can compare zlibVersion and ZLIB_VERSION for consistency.
   If the first character differs, the library code actually used is not
   compatible with the zlib.h header file used by the application.  This check
   is automatically made by deflateInit and inflateInit.
 */

/*
ZEXTERN int ZEXPORT deflateInit OF((z_streamp strm, int level));

     Initializes the internal stream state for compression.  The fields
   zalloc, zfree and opaque must be initialized before by the caller.  If
   zalloc and zfree are set to Z_NULL, deflateInit updates them to use default
   allocation functions.

     The compression level must be Z_DEFAULT_COMPRESSION, or between 0 and 9:
   1 gives best speed, 9 gives best compression, 0 gives no compression at all
   (the input data is simply copied a block at a time).  Z_DEFAULT_COMPRESSION
   requests a default compromise between speed and compression (currently
   equivalent to level 6).

     deflateInit returns Z_OK if success, Z_MEM_ERROR if there was not enough
   memory, Z_STREAM_ERROR if level is not a valid compression level, or
   Z_VERSION_ERROR if the zlib library version (zlib_version) is incompatible
   with the version assumed by the caller (ZLIB_VERSION).  msg is set to null
   if there is no error message.  deflateInit does not perform any compression:
   this will be done by deflate().
*/


ZEXTERN int ZEXPORT deflate OF((z_streamp strm, int flush));
/*
    deflate compresses as much data as possible, and stops when the input
  buffer becomes empty or the output buffer becomes full.  It may introduce
  some output latency (reading input without producing any output) except when
  forced to flush.

    The detailed semantics are as follows.  deflate performs one or both of the
  following actions:

  - Compress more input starting at next_in and update next_in and avail_in
    accordingly.  If not all input can be processed (because there is not
    enough room in the output buffer), next_in and avail_in are updated and
    processing will resume at this point for the next call of deflate().

  - Provide more output starting at next_out and update next_out and avail_out
    accordingly.  This action is forced if the parameter flush is non zero.
    Forcing flush frequently degrades the compression ratio, so this parameter
    should be set only when necessary (in interactive applications).  Some
    output may be provided even if flush is not set.

    Before the call of deflate(), the application should ensure that at least
  one of the actions is possible, by providing more input and/or consuming more
  output, and updating avail_in or avail_out accordingly; avail_out should
  never be zero before the call.  The application can consume the compressed
  output when it wants, for example when the output buffer is full (avail_out
  == 0), or after each call of deflate().  If deflate returns Z_OK and with
  zero avail_out, it must be called again after making room in the output
  buffer because there might be more output pending.

    Normally the parameter flush is set to Z_NO_FLUSH, which allows deflate to
  decide how much data to accumulate before producing output, in order to
  maximize compression.

    If the parameter flush is set to Z_SYNC_FLUSH, all pending output is
  flushed to the output buffer and the output is aligned on a byte boundary, so
  that the decompressor can get all input data available so far.  (In
  particular avail_in is zero after the call if enough output space has been
  provided before the call.) Flushing may degrade compression for some
  compression algorithms and so it should be used only when necessary.  This
  completes the current deflate block and follows it with an empty stored block
  that is three bits plus filler bits to the next byte, followed by four bytes
  (00 00 ff ff).

    If flush is set to Z_PARTIAL_FLUSH, all pending output is flushed to the
  output buffer, but the output is not aligned to a byte boundary.  All of the
  input data so far will be available to the decompressor, as for Z_SYNC_FLUSH.
  This completes the current deflate block and follows it with an empty fixed
```

```
  codes block that is 10 bits long.  This assures that enough bytes are output
  in order for the decompressor to finish the block before the empty fixed code
  block.

    If flush is set to Z_BLOCK, a deflate block is completed and emitted, as
  for Z_SYNC_FLUSH, but the output is not aligned on a byte boundary, and up to
  seven bits of the current block are held to be written as the next byte after
  the next deflate block is completed.  In this case, the decompressor may not
  be provided enough bits at this point in order to complete decompression of
  the data provided so far to the compressor.  It may need to wait for the next
  block to be emitted.  This is for advanced applications that need to control
  the emission of deflate blocks.

    If flush is set to Z_FULL_FLUSH, all output is flushed as with
  Z_SYNC_FLUSH, and the compression state is reset so that decompression can
  restart from this point if previous compressed data has been damaged or if
  random access is desired.  Using Z_FULL_FLUSH too often can seriously degrade
  compression.

    If deflate returns with avail_out == 0, this function must be called again
  with the same value of the flush parameter and more output space (updated
  avail_out), until the flush is complete (deflate returns with non-zero
  avail_out).  In the case of a Z_FULL_FLUSH or Z_SYNC_FLUSH, make sure that
  avail_out is greater than six to avoid repeated flush markers due to
  avail_out == 0 on return.

    If the parameter flush is set to Z_FINISH, pending input is processed,
  pending output is flushed and deflate returns with Z_STREAM_END if there was
  enough output space; if deflate returns with Z_OK, this function must be
  called again with Z_FINISH and more output space (updated avail_out) but no
  more input data, until it returns with Z_STREAM_END or an error.  After
  deflate has returned Z_STREAM_END, the only possible operations on the stream
  are deflateReset or deflateEnd.

    Z_FINISH can be used immediately after deflateInit if all the compression
  is to be done in a single step.  In this case, avail_out must be at least the
  value returned by deflateBound (see below).  Then deflate is guaranteed to
  return Z_STREAM_END.  If not enough output space is provided, deflate will
  not return Z_STREAM_END, and it must be called again as described above.

    deflate() sets strm->adler to the adler32 checksum of all input read
  so far (that is, total_in bytes).

    deflate() may update strm->data_type if it can make a good guess about
  the input data type (Z_BINARY or Z_TEXT).  In doubt, the data is considered
  binary.  This field is only for information purposes and does not affect the
  compression algorithm in any manner.

    deflate() returns Z_OK if some progress has been made (more input
  processed or more output produced), Z_STREAM_END if all input has been
  consumed and all output has been produced (only when flush is set to
  Z_FINISH), Z_STREAM_ERROR if the stream state was inconsistent (for example
  if next_in or next_out was Z_NULL), Z_BUF_ERROR if no progress is possible
  (for example avail_in or avail_out was zero).  Note that Z_BUF_ERROR is not
  fatal, and deflate() can be called again with more input and more output
  space to continue compressing.
*/


ZEXTERN int ZEXPORT deflateEnd OF((z_streamp strm));
/*
     All dynamically allocated data structures for this stream are freed.
   This function discards any unprocessed input and does not flush any pending
   output.

     deflateEnd returns Z_OK if success, Z_STREAM_ERROR if the
   stream state was inconsistent, Z_DATA_ERROR if the stream was freed
   prematurely (some input or output was discarded).  In the error case, msg
   may be set but then points to a static string (which must not be
   deallocated).
*/


/*
ZEXTERN int ZEXPORT inflateInit OF((z_streamp strm));

     Initializes the internal stream state for decompression.  The fields
   next_in, avail_in, zalloc, zfree and opaque must be initialized before by
   the caller.  If next_in is not Z_NULL and avail_in is large enough (the
   exact value depends on the compression method), inflateInit determines the
   compression method from the zlib header and allocates all data structures
   accordingly; otherwise the allocation will be deferred to the first call of
   inflate.  If zalloc and zfree are set to Z_NULL, inflateInit updates them to
   use default allocation functions.

     inflateInit returns Z_OK if success, Z_MEM_ERROR if there was not enough
   memory, Z_VERSION_ERROR if the zlib library version is incompatible with the
   version assumed by the caller, or Z_STREAM_ERROR if the parameters are
   invalid, such as a null pointer to the structure.  msg is set to null if
   there is no error message.  inflateInit does not perform any decompression
   apart from possibly reading the zlib header if present: actual decompression
   will be done by inflate().  (So next_in and avail_in may be modified, but
   next_out and avail_out are unused and unchanged.) The current implementation
   of inflateInit() does not process any header information -- that is deferred
   until inflate() is called.
*/
```

```
ZEXTERN int ZEXPORT inflate OF((z_streamp strm, int flush));
/*
    inflate decompresses as much data as possible, and stops when the input
  buffer becomes empty or the output buffer becomes full.  It may introduce
  some output latency (reading input without producing any output) except when
  forced to flush.

  The detailed semantics are as follows.  inflate performs one or both of the
  following actions:

  - Decompress more input starting at next_in and update next_in and avail_in
    accordingly.  If not all input can be processed (because there is not
    enough room in the output buffer), next_in is updated and processing will
    resume at this point for the next call of inflate().

  - Provide more output starting at next_out and update next_out and avail_out
    accordingly.  inflate() provides as much output as possible, until there is
    no more input data or no more space in the output buffer (see below about
    the flush parameter).

    Before the call of inflate(), the application should ensure that at least
  one of the actions is possible, by providing more input and/or consuming more
  output, and updating the next_* and avail_* values accordingly.  The
  application can consume the uncompressed output when it wants, for example
  when the output buffer is full (avail_out == 0), or after each call of
  inflate().  If inflate returns Z_OK and with zero avail_out, it must be
  called again after making room in the output buffer because there might be
  more output pending.

    The flush parameter of inflate() can be Z_NO_FLUSH, Z_SYNC_FLUSH, Z_FINISH,
  Z_BLOCK, or Z_TREES.  Z_SYNC_FLUSH requests that inflate() flush as much
  output as possible to the output buffer.  Z_BLOCK requests that inflate()
  stop if and when it gets to the next deflate block boundary.  When decoding
  the zlib or gzip format, this will cause inflate() to return immediately
  after the header and before the first block.  When doing a raw inflate,
  inflate() will go ahead and process the first block, and will return when it
  gets to the end of that block, or when it runs out of data.

    The Z_BLOCK option assists in appending to or combining deflate streams.
  Also to assist in this, on return inflate() will set strm->data_type to the
  number of unused bits in the last byte taken from strm->next_in, plus 64 if
  inflate() is currently decoding the last block in the deflate stream, plus
  128 if inflate() returned immediately after decoding an end-of-block code or
  decoding the complete header up to just before the first byte of the deflate
  stream.  The end-of-block will not be indicated until all of the uncompressed
  data from that block has been written to strm->next_out.  The number of
  unused bits may in general be greater than seven, except when bit 7 of
  data_type is set, in which case the number of unused bits will be less than
  eight.  data_type is set as noted here every time inflate() returns for all
  flush options, and so can be used to determine the amount of currently
  consumed input in bits.

    The Z_TREES option behaves as Z_BLOCK does, but it also returns when the
  end of each deflate block header is reached, before any actual data in that
  block is decoded.  This allows the caller to determine the length of the
  deflate block header for later use in random access within a deflate block.
  256 is added to the value of strm->data_type when inflate() returns
  immediately after reaching the end of the deflate block header.

    inflate() should normally be called until it returns Z_STREAM_END or an
  error.  However if all decompression is to be performed in a single step (a
  single call of inflate), the parameter flush should be set to Z_FINISH.  In
  this case all pending input is processed and all pending output is flushed;
  avail_out must be large enough to hold all of the uncompressed data for the
  operation to complete.  (The size of the uncompressed data may have been
  saved by the compressor for this purpose.) The use of Z_FINISH is not
  required to perform an inflation in one step.  However it may be used to
  inform inflate that a faster approach can be used for the single inflate()
  call.  Z_FINISH also informs inflate to not maintain a sliding window if the
  stream completes, which reduces inflate's memory footprint.  If the stream
  does not complete, either because not all of the stream is provided or not
  enough output space is provided, then a sliding window will be allocated and
  inflate() can be called again to continue the operation as if Z_NO_FLUSH had
  been used.

     In this implementation, inflate() always flushes as much output as
  possible to the output buffer, and always uses the faster approach on the
  first call.  So the effects of the flush parameter in this implementation are
  on the return value of inflate() as noted below, when inflate() returns early
  when Z_BLOCK or Z_TREES is used, and when inflate() avoids the allocation of
  memory for a sliding window when Z_FINISH is used.

     If a preset dictionary is needed after this call (see inflateSetDictionary
  below), inflate sets strm->adler to the Adler-32 checksum of the dictionary
  chosen by the compressor and returns Z_NEED_DICT; otherwise it sets
  strm->adler to the Adler-32 checksum of all output produced so far (that is,
  total_out bytes) and returns Z_OK, Z_STREAM_END or an error code as described
  below.  At the end of the stream, inflate() checks that its computed adler32
  checksum is equal to that saved by the compressor and returns Z_STREAM_END
  only if the checksum is correct.

    inflate() can decompress and check either zlib-wrapped or gzip-wrapped
  deflate data.  The header type is detected automatically, if requested when
  initializing with inflateInit2().  Any information contained in the gzip
  header is not retained, so applications that need that information should
  instead use raw inflate, see inflateInit2() below, or inflateBack() and
  perform their own processing of the gzip header and trailer.  When processing
  gzip-wrapped deflate data, strm->adler32 is set to the CRC-32 of the output
```

```
  producted so far.  The CRC-32 is checked against the gzip trailer.

    inflate() returns Z_OK if some progress has been made (more input processed
  or more output produced), Z_STREAM_END if the end of the compressed data has
  been reached and all uncompressed output has been produced, Z_NEED_DICT if a
  preset dictionary is needed at this point, Z_DATA_ERROR if the input data was
  corrupted (input stream not conforming to the zlib format or incorrect check
  value), Z_STREAM_ERROR if the stream structure was inconsistent (for example
  next_in or next_out was Z_NULL), Z_MEM_ERROR if there was not enough memory,
  Z_BUF_ERROR if no progress is possible or if there was not enough room in the
  output buffer when Z_FINISH is used.  Note that Z_BUF_ERROR is not fatal, and
  inflate() can be called again with more input and more output space to
  continue decompressing.  If Z_DATA_ERROR is returned, the application may
  then call inflateSync() to look for a good compression block if a partial
  recovery of the data is desired.
*/


ZEXTERN int ZEXPORT inflateEnd OF((z_streamp strm));
/*
     All dynamically allocated data structures for this stream are freed.
   This function discards any unprocessed input and does not flush any pending
   output.

     inflateEnd returns Z_OK if success, Z_STREAM_ERROR if the stream state
   was inconsistent.  In the error case, msg may be set but then points to a
   static string (which must not be deallocated).
*/


                        /* Advanced functions */

/*
    The following functions are needed only in some special applications.
*/

/*
ZEXTERN int ZEXPORT deflateInit2 OF((z_streamp strm,
                                     int  level,
                                     int  method,
                                     int  windowBits,
                                     int  memLevel,
                                     int  strategy));

     This is another version of deflateInit with more compression options.  The
   fields next_in, zalloc, zfree and opaque must be initialized before by the
   caller.

     The method parameter is the compression method.  It must be Z_DEFLATED in
   this version of the library.

     The windowBits parameter is the base two logarithm of the window size
   (the size of the history buffer).  It should be in the range 8..15 for this
   version of the library.  Larger values of this parameter result in better
   compression at the expense of memory usage.  The default value is 15 if
   deflateInit is used instead.

     windowBits can also be -8..-15 for raw deflate.  In this case, -windowBits
   determines the window size.  deflate() will then generate raw deflate data
   with no zlib header or trailer, and will not compute an adler32 check value.

     windowBits can also be greater than 15 for optional gzip encoding.  Add
   16 to windowBits to write a simple gzip header and trailer around the
   compressed data instead of a zlib wrapper.  The gzip header will have no
   file name, no extra data, no comment, no modification time (set to zero), no
   header crc, and the operating system will be set to 255 (unknown).  If a
   gzip stream is being written, strm->adler is a crc32 instead of an adler32.

     The memLevel parameter specifies how much memory should be allocated
   for the internal compression state.  memLevel=1 uses minimum memory but is
   slow and reduces compression ratio; memLevel=9 uses maximum memory for
   optimal speed.  The default value is 8.  See zconf.h for total memory usage
   as a function of windowBits and memLevel.

     The strategy parameter is used to tune the compression algorithm.  Use the
   value Z_DEFAULT_STRATEGY for normal data, Z_FILTERED for data produced by a
   filter (or predictor), Z_HUFFMAN_ONLY to force Huffman encoding only (no
   string match), or Z_RLE to limit match distances to one (run-length
   encoding).  Filtered data consists mostly of small values with a somewhat
   random distribution.  In this case, the compression algorithm is tuned to
   compress them better.  The effect of Z_FILTERED is to force more Huffman
   coding and less string matching; it is somewhat intermediate between
   Z_DEFAULT_STRATEGY and Z_HUFFMAN_ONLY.  Z_RLE is designed to be almost as
   fast as Z_HUFFMAN_ONLY, but give better compression for PNG image data.  The
   strategy parameter only affects the compression ratio but not the
   correctness of the compressed output even if it is not set appropriately.
   Z_FIXED prevents the use of dynamic Huffman codes, allowing for a simpler
   decoder for special applications.

     deflateInit2 returns Z_OK if success, Z_MEM_ERROR if there was not enough
   memory, Z_STREAM_ERROR if any parameter is invalid (such as an invalid
   method), or Z_VERSION_ERROR if the zlib library version (zlib_version) is
   incompatible with the version assumed by the caller (ZLIB_VERSION).  msg is
   set to null if there is no error message.  deflateInit2 does not perform any
   compression: this will be done by deflate().
*/

ZEXTERN int ZEXPORT deflateSetDictionary OF((z_streamp strm,
```

```
                                        const Bytef *dictionary,
                                        uInt  dictLength));
/*
     Initializes the compression dictionary from the given byte sequence
   without producing any compressed output.  When using the zlib format, this
   function must be called immediately after deflateInit, deflateInit2 or
   deflateReset, and before any call of deflate.  When doing raw deflate, this
   function must be called either before any call of deflate, or immediately
   after the completion of a deflate block, i.e. after all input has been
   consumed and all output has been delivered when using any of the flush
   options Z_BLOCK, Z_PARTIAL_FLUSH, Z_SYNC_FLUSH, or Z_FULL_FLUSH.  The
   compressor and decompressor must use exactly the same dictionary (see
   inflateSetDictionary).

     The dictionary should consist of strings (byte sequences) that are likely
   to be encountered later in the data to be compressed, with the most commonly
   used strings preferably put towards the end of the dictionary.  Using a
   dictionary is most useful when the data to be compressed is short and can be
   predicted with good accuracy; the data can then be compressed better than
   with the default empty dictionary.

     Depending on the size of the compression data structures selected by
   deflateInit or deflateInit2, a part of the dictionary may in effect be
   discarded, for example if the dictionary is larger than the window size
   provided in deflateInit or deflateInit2.  Thus the strings most likely to be
   useful should be put at the end of the dictionary, not at the front.  In
   addition, the current implementation of deflate will use at most the window
   size minus 262 bytes of the provided dictionary.

     Upon return of this function, strm->adler is set to the adler32 value
   of the dictionary; the decompressor may later use this value to determine
   which dictionary has been used by the compressor.  (The adler32 value
   applies to the whole dictionary even if only a subset of the dictionary is
   actually used by the compressor.) If a raw deflate was requested, then the
   adler32 value is not computed and strm->adler is not set.

     deflateSetDictionary returns Z_OK if success, or Z_STREAM_ERROR if a
   parameter is invalid (e.g.  dictionary being Z_NULL) or the stream state is
   inconsistent (for example if deflate has already been called for this stream
   or if not at a block boundary for raw deflate).  deflateSetDictionary does
   not perform any compression: this will be done by deflate().
*/

ZEXTERN int ZEXPORT deflateCopy OF((z_streamp dest,
                                    z_streamp source));
/*
     Sets the destination stream as a complete copy of the source stream.

     This function can be useful when several compression strategies will be
   tried, for example when there are several ways of pre-processing the input
   data with a filter.  The streams that will be discarded should then be freed
   by calling deflateEnd.  Note that deflateCopy duplicates the internal
   compression state which can be quite large, so this strategy is slow and can
   consume lots of memory.

     deflateCopy returns Z_OK if success, Z_MEM_ERROR if there was not
   enough memory, Z_STREAM_ERROR if the source stream state was inconsistent
   (such as zalloc being Z_NULL).  msg is left unchanged in both source and
   destination.
*/

ZEXTERN int ZEXPORT deflateReset OF((z_streamp strm));
/*
     This function is equivalent to deflateEnd followed by deflateInit,
   but does not free and reallocate all the internal compression state.  The
   stream will keep the same compression level and any other attributes that
   may have been set by deflateInit2.

     deflateReset returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent (such as zalloc or state being Z_NULL).
*/

ZEXTERN int ZEXPORT deflateParams OF((z_streamp strm,
                                      int level,
                                      int strategy));
/*
     Dynamically update the compression level and compression strategy.  The
   interpretation of level and strategy is as in deflateInit2.  This can be
   used to switch between compression and straight copy of the input data, or
   to switch to a different kind of input data requiring a different strategy.
   If the compression level is changed, the input available so far is
   compressed with the old level (and may be flushed); the new level will take
   effect only at the next call of deflate().

     Before the call of deflateParams, the stream state must be set as for
   a call of deflate(), since the currently available input may have to be
   compressed and flushed.  In particular, strm->avail_out must be non-zero.

     deflateParams returns Z_OK if success, Z_STREAM_ERROR if the source
   stream state was inconsistent or if a parameter was invalid, Z_BUF_ERROR if
   strm->avail_out was zero.
*/

ZEXTERN int ZEXPORT deflateTune OF((z_streamp strm,
                                    int good_length,
                                    int max_lazy,
                                    int nice_length,
                                    int max_chain));
```

```
/*
     Fine tune deflate's internal compression parameters.  This should only be
   used by someone who understands the algorithm used by zlib's deflate for
   searching for the best matching string, and even then only by the most
   fanatic optimizer trying to squeeze out the last compressed bit for their
   specific input data.  Read the deflate.c source code for the meaning of the
   max_lazy, good_length, nice_length, and max_chain parameters.

     deflateTune() can be called after deflateInit() or deflateInit2(), and
   returns Z_OK on success, or Z_STREAM_ERROR for an invalid deflate stream.
 */

ZEXTERN uLong ZEXPORT deflateBound OF((z_streamp strm,
                                       uLong sourceLen));
/*
     deflateBound() returns an upper bound on the compressed size after
   deflation of sourceLen bytes.  It must be called after deflateInit() or
   deflateInit2(), and after deflateSetHeader(), if used.  This would be used
   to allocate an output buffer for deflation in a single pass, and so would be
   called before deflate().  If that first deflate() call is provided the
   sourceLen input bytes, an output buffer allocated to the size returned by
   deflateBound(), and the flush value Z_FINISH, then deflate() is guaranteed
   to return Z_STREAM_END.  Note that it is possible for the compressed size to
   be larger than the value returned by deflateBound() if flush options other
   than Z_FINISH or Z_NO_FLUSH are used.
*/

ZEXTERN int ZEXPORT deflatePending OF((z_streamp strm,
                                       unsigned *pending,
                                       int *bits));
/*
     deflatePending() returns the number of bytes and bits of output that have
   been generated, but not yet provided in the available output.  The bytes not
   provided would be due to the available output space having being consumed.
   The number of bits of output not provided are between 0 and 7, where they
   await more bits to join them in order to fill out a full byte.  If pending
   or bits are Z_NULL, then those values are not set.

     deflatePending returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent.
 */

ZEXTERN int ZEXPORT deflatePrime OF((z_streamp strm,
                                     int bits,
                                     int value));
/*
     deflatePrime() inserts bits in the deflate output stream.  The intent
   is that this function is used to start off the deflate output with the bits
   leftover from a previous deflate stream when appending to it.  As such, this
   function can only be used for raw deflate, and must be used before the first
   deflate() call after a deflateInit2() or deflateReset().  bits must be less
   than or equal to 16, and that many of the least significant bits of value
   will be inserted in the output.

     deflatePrime returns Z_OK if success, Z_BUF_ERROR if there was not enough
   room in the internal buffer to insert the bits, or Z_STREAM_ERROR if the
   source stream state was inconsistent.
*/

ZEXTERN int ZEXPORT deflateSetHeader OF((z_streamp strm,
                                         gz_headerp head));
/*
     deflateSetHeader() provides gzip header information for when a gzip
   stream is requested by deflateInit2().  deflateSetHeader() may be called
   after deflateInit2() or deflateReset() and before the first call of
   deflate().  The text, time, os, extra field, name, and comment information
   in the provided gz_header structure are written to the gzip header (xflag is
   ignored -- the extra flags are set according to the compression level).  The
   caller must assure that, if not Z_NULL, name and comment are terminated with
   a zero byte, and that if extra is not Z_NULL, that extra_len bytes are
   available there.  If hcrc is true, a gzip header crc is included.  Note that
   the current versions of the command-line version of gzip (up through version
   1.3.x) do not support header crc's, and will report that it is a "multi-part
   gzip file" and give up.

     If deflateSetHeader is not used, the default gzip header has text false,
   the time set to zero, and os set to 255, with no extra, name, or comment
   fields.  The gzip header is returned to the default state by deflateReset().

     deflateSetHeader returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent.
*/

/*
ZEXTERN int ZEXPORT inflateInit2 OF((z_streamp strm,
                                     int  windowBits));

     This is another version of inflateInit with an extra parameter.  The
   fields next_in, avail_in, zalloc, zfree and opaque must be initialized
   before by the caller.

     The windowBits parameter is the base two logarithm of the maximum window
   size (the size of the history buffer).  It should be in the range 8..15 for
   this version of the library.  The default value is 15 if inflateInit is used
   instead.  windowBits must be greater than or equal to the windowBits value
   provided to deflateInit2() while compressing, or it must be equal to 15 if
   deflateInit2() was not used.  If a compressed stream with a larger window
   size is given as input, inflate() will return with the error code
```

```
   Z_DATA_ERROR instead of trying to allocate a larger window.

     windowBits can also be zero to request that inflate use the window size in
   the zlib header of the compressed stream.

     windowBits can also be -8..-15 for raw inflate.  In this case, -windowBits
   determines the window size.  inflate() will then process raw deflate data,
   not looking for a zlib or gzip header, not generating a check value, and not
   looking for any check values for comparison at the end of the stream.  This
   is for use with other formats that use the deflate compressed data format
   such as zip.  Those formats provide their own check values.  If a custom
   format is developed using the raw deflate format for compressed data, it is
   recommended that a check value such as an adler32 or a crc32 be applied to
   the uncompressed data as is done in the zlib, gzip, and zip formats.  For
   most applications, the zlib format should be used as is.  Note that comments
   above on the use in deflateInit2() applies to the magnitude of windowBits.

     windowBits can also be greater than 15 for optional gzip decoding.  Add
   32 to windowBits to enable zlib and gzip decoding with automatic header
   detection, or add 16 to decode only the gzip format (the zlib format will
   return a Z_DATA_ERROR).  If a gzip stream is being decoded, strm->adler is a
   crc32 instead of an adler32.

     inflateInit2 returns Z_OK if success, Z_MEM_ERROR if there was not enough
   memory, Z_VERSION_ERROR if the zlib library version is incompatible with the
   version assumed by the caller, or Z_STREAM_ERROR if the parameters are
   invalid, such as a null pointer to the structure.  msg is set to null if
   there is no error message.  inflateInit2 does not perform any decompression
   apart from possibly reading the zlib header if present: actual decompression
   will be done by inflate().  (So next_in and avail_in may be modified, but
   next_out and avail_out are unused and unchanged.) The current implementation
   of inflateInit2() does not process any header information -- that is
   deferred until inflate() is called.
*/

ZEXTERN int ZEXPORT inflateSetDictionary OF((z_streamp strm,
                                             const Bytef *dictionary,
                                             uInt  dictLength));
/*
     Initializes the decompression dictionary from the given uncompressed byte
   sequence.  This function must be called immediately after a call of inflate,
   if that call returned Z_NEED_DICT.  The dictionary chosen by the compressor
   can be determined from the adler32 value returned by that call of inflate.
   The compressor and decompressor must use exactly the same dictionary (see
   deflateSetDictionary).  For raw inflate, this function can be called at any
   time to set the dictionary.  If the provided dictionary is smaller than the
   window and there is already data in the window, then the provided dictionary
   will amend what's there.  The application must insure that the dictionary
   that was used for compression is provided.

     inflateSetDictionary returns Z_OK if success, Z_STREAM_ERROR if a
   parameter is invalid (e.g.  dictionary being Z_NULL) or the stream state is
   inconsistent, Z_DATA_ERROR if the given dictionary doesn't match the
   expected one (incorrect adler32 value).  inflateSetDictionary does not
   perform any decompression: this will be done by subsequent calls of
   inflate().
*/

ZEXTERN int ZEXPORT inflateGetDictionary OF((z_streamp strm,
                                             Bytef *dictionary,
                                             uInt  *dictLength));
/*
     Returns the sliding dictionary being maintained by inflate.  dictLength is
   set to the number of bytes in the dictionary, and that many bytes are copied
   to dictionary.  dictionary must have enough space, where 32768 bytes is
   always enough.  If inflateGetDictionary() is called with dictionary equal to
   Z_NULL, then only the dictionary length is returned, and nothing is copied.
   Similary, if dictLength is Z_NULL, then it is not set.

     inflateGetDictionary returns Z_OK on success, or Z_STREAM_ERROR if the
   stream state is inconsistent.
*/

ZEXTERN int ZEXPORT inflateSync OF((z_streamp strm));
/*
     Skips invalid compressed data until a possible full flush point (see above
   for the description of deflate with Z_FULL_FLUSH) can be found, or until all
   available input is skipped.  No output is provided.

     inflateSync searches for a 00 00 FF FF pattern in the compressed data.
   All full flush points have this pattern, but not all occurrences of this
   pattern are full flush points.

     inflateSync returns Z_OK if a possible full flush point has been found,
   Z_BUF_ERROR if no more input was provided, Z_DATA_ERROR if no flush point
   has been found, or Z_STREAM_ERROR if the stream structure was inconsistent.
   In the success case, the application may save the current current value of
   total_in which indicates where valid compressed data was found.  In the
   error case, the application may repeatedly call inflateSync, providing more
   input each time, until success or end of the input data.
*/

ZEXTERN int ZEXPORT inflateCopy OF((z_streamp dest,
                                    z_streamp source));
/*
     Sets the destination stream as a complete copy of the source stream.

     This function can be useful when randomly accessing a large stream.  The
```

```
   first pass through the stream can periodically record the inflate state,
   allowing restarting inflate at those points when randomly accessing the
   stream.

     inflateCopy returns Z_OK if success, Z_MEM_ERROR if there was not
   enough memory, Z_STREAM_ERROR if the source stream state was inconsistent
   (such as zalloc being Z_NULL).  msg is left unchanged in both source and
   destination.
*/

ZEXTERN int ZEXPORT inflateReset OF((z_streamp strm));
/*
     This function is equivalent to inflateEnd followed by inflateInit,
   but does not free and reallocate all the internal decompression state.  The
   stream will keep attributes that may have been set by inflateInit2.

     inflateReset returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent (such as zalloc or state being Z_NULL).
*/

ZEXTERN int ZEXPORT inflateReset2 OF((z_streamp strm,
                                      int windowBits));
/*
     This function is the same as inflateReset, but it also permits changing
   the wrap and window size requests.  The windowBits parameter is interpreted
   the same as it is for inflateInit2.

     inflateReset2 returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent (such as zalloc or state being Z_NULL), or if
   the windowBits parameter is invalid.
*/

ZEXTERN int ZEXPORT inflatePrime OF((z_streamp strm,
                                     int bits,
                                     int value));
/*
     This function inserts bits in the inflate input stream.  The intent is
   that this function is used to start inflating at a bit position in the
   middle of a byte.  The provided bits will be used before any bytes are used
   from next_in.  This function should only be used with raw inflate, and
   should be used before the first inflate() call after inflateInit2() or
   inflateReset().  bits must be less than or equal to 16, and that many of the
   least significant bits of value will be inserted in the input.

     If bits is negative, then the input stream bit buffer is emptied.  Then
   inflatePrime() can be called again to put bits in the buffer.  This is used
   to clear out bits leftover after feeding inflate a block description prior
   to feeding inflate codes.

     inflatePrime returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent.
*/

ZEXTERN long ZEXPORT inflateMark OF((z_streamp strm));
/*
     This function returns two values, one in the lower 16 bits of the return
   value, and the other in the remaining upper bits, obtained by shifting the
   return value down 16 bits.  If the upper value is -1 and the lower value is
   zero, then inflate() is currently decoding information outside of a block.
   If the upper value is -1 and the lower value is non-zero, then inflate is in
   the middle of a stored block, with the lower value equaling the number of
   bytes from the input remaining to copy.  If the upper value is not -1, then
   it is the number of bits back from the current bit position in the input of
   the code (literal or length/distance pair) currently being processed.  In
   that case the lower value is the number of bytes already emitted for that
   code.

     A code is being processed if inflate is waiting for more input to complete
   decoding of the code, or if it has completed decoding but is waiting for
   more output space to write the literal or match data.

     inflateMark() is used to mark locations in the input data for random
   access, which may be at bit positions, and to note those cases where the
   output of a code may span boundaries of random access blocks.  The current
   location in the input stream can be determined from avail_in and data_type
   as noted in the description for the Z_BLOCK flush parameter for inflate.

     inflateMark returns the value noted above or -1 << 16 if the provided
   source stream state was inconsistent.
*/

ZEXTERN int ZEXPORT inflateGetHeader OF((z_streamp strm,
                                         gz_headerp head));
/*
     inflateGetHeader() requests that gzip header information be stored in the
   provided gz_header structure.  inflateGetHeader() may be called after
   inflateInit2() or inflateReset(), and before the first call of inflate().
   As inflate() processes the gzip stream, head->done is zero until the header
   is completed, at which time head->done is set to one.  If a zlib stream is
   being decoded, then head->done is set to -1 to indicate that there will be
   no gzip header information forthcoming.  Note that Z_BLOCK or Z_TREES can be
   used to force inflate() to return immediately after header processing is
   complete and before any actual data is decompressed.

     The text, time, xflags, and os fields are filled in with the gzip header
   contents.  hcrc is set to true if there is a header CRC.  (The header CRC
   was valid if done is set to one.) If extra is not Z_NULL, then extra_max
   contains the maximum number of bytes to write to extra.  Once done is true,
```

```
   extra_len contains the actual extra field length, and extra contains the
   extra field, or that field truncated if extra_max is less than extra_len.
   If name is not Z_NULL, then up to name_max characters are written there,
   terminated with a zero unless the length is greater than name_max.  If
   comment is not Z_NULL, then up to comm_max characters are written there,
   terminated with a zero unless the length is greater than comm_max.  When any
   of extra, name, or comment are not Z_NULL and the respective field is not
   present in the header, then that field is set to Z_NULL to signal its
   absence.  This allows the use of deflateSetHeader() with the returned
   structure to duplicate the header.  However if those fields are set to
   allocated memory, then the application will need to save those pointers
   elsewhere so that they can be eventually freed.

     If inflateGetHeader is not used, then the header information is simply
   discarded.  The header is always checked for validity, including the header
   CRC if present.  inflateReset() will reset the process to discard the header
   information.  The application would need to call inflateGetHeader() again to
   retrieve the header from the next gzip stream.

     inflateGetHeader returns Z_OK if success, or Z_STREAM_ERROR if the source
   stream state was inconsistent.
*/

/*
ZEXTERN int ZEXPORT inflateBackInit OF((z_streamp strm, int windowBits,
                                        unsigned char FAR *window));

     Initialize the internal stream state for decompression using inflateBack()
   calls.  The fields zalloc, zfree and opaque in strm must be initialized
   before the call.  If zalloc and zfree are Z_NULL, then the default library-
   derived memory allocation routines are used.  windowBits is the base two
   logarithm of the window size, in the range 8..15.  window is a caller
   supplied buffer of that size.  Except for special applications where it is
   assured that deflate was used with small window sizes, windowBits must be 15
   and a 32K byte window must be supplied to be able to decompress general
   deflate streams.

     See inflateBack() for the usage of these routines.

     inflateBackInit will return Z_OK on success, Z_STREAM_ERROR if any of
   the parameters are invalid, Z_MEM_ERROR if the internal state could not be
   allocated, or Z_VERSION_ERROR if the version of the library does not match
   the version of the header file.
*/

typedef unsigned (*in_func) OF((void FAR *,
                                z_const unsigned char FAR * FAR *));
typedef int (*out_func) OF((void FAR *, unsigned char FAR *, unsigned));

ZEXTERN int ZEXPORT inflateBack OF((z_streamp strm,
                                    in_func in, void FAR *in_desc,
                                    out_func out, void FAR *out_desc));
/*
     inflateBack() does a raw inflate with a single call using a call-back
   interface for input and output.  This is potentially more efficient than
   inflate() for file i/o applications, in that it avoids copying between the
   output and the sliding window by simply making the window itself the output
   buffer.  inflate() can be faster on modern CPUs when used with large
   buffers.  inflateBack() trusts the application to not change the output
   buffer passed by the output function, at least until inflateBack() returns.

     inflateBackInit() must be called first to allocate the internal state
   and to initialize the state with the user-provided window buffer.
   inflateBack() may then be used multiple times to inflate a complete, raw
   deflate stream with each call.  inflateBackEnd() is then called to free the
   allocated state.

     A raw deflate stream is one with no zlib or gzip header or trailer.
   This routine would normally be used in a utility that reads zip or gzip
   files and writes out uncompressed files.  The utility would decode the
   header and process the trailer on its own, hence this routine expects only
   the raw deflate stream to decompress.  This is different from the normal
   behavior of inflate(), which expects either a zlib or gzip header and
   trailer around the deflate stream.

     inflateBack() uses two subroutines supplied by the caller that are then
   called by inflateBack() for input and output.  inflateBack() calls those
   routines until it reads a complete deflate stream and writes out all of the
   uncompressed data, or until it encounters an error.  The function's
   parameters and return types are defined above in the in_func and out_func
   typedefs.  inflateBack() will call in(in_desc, &buf) which should return the
   number of bytes of provided input, and a pointer to that input in buf.  If
   there is no input available, in() must return zero--buf is ignored in that
   case--and inflateBack() will return a buffer error.  inflateBack() will call
   out(out_desc, buf, len) to write the uncompressed data buf[0..len-1].  out()
   should return zero on success, or non-zero on failure.  If out() returns
   non-zero, inflateBack() will return with an error.  Neither in() nor out()
   are permitted to change the contents of the window provided to
   inflateBackInit(), which is also the buffer that out() uses to write from.
   The length written by out() will be at most the window size.  Any non-zero
   amount of input may be provided by in().

     For convenience, inflateBack() can be provided input on the first call by
   setting strm->next_in and strm->avail_in.  If that input is exhausted, then
   in() will be called.  Therefore strm->next_in must be initialized before
   calling inflateBack().  If strm->next_in is Z_NULL, then in() will be called
   immediately for input.  If strm->next_in is not Z_NULL, then strm->avail_in
   must also be initialized, and then if strm->avail_in is not zero, input will
```

```
   initially be taken from strm->next_in[0 ..  strm->avail_in - 1].

     The in_desc and out_desc parameters of inflateBack() is passed as the
   first parameter of in() and out() respectively when they are called.  These
   descriptors can be optionally used to pass any information that the caller-
   supplied in() and out() functions need to do their job.

     On return, inflateBack() will set strm->next_in and strm->avail_in to
   pass back any unused input that was provided by the last in() call.  The
   return values of inflateBack() can be Z_STREAM_END on success, Z_BUF_ERROR
   if in() or out() returned an error, Z_DATA_ERROR if there was a format error
   in the deflate stream (in which case strm->msg is set to indicate the nature
   of the error), or Z_STREAM_ERROR if the stream was not properly initialized.
   In the case of Z_BUF_ERROR, an input or output error can be distinguished
   using strm->next_in which will be Z_NULL only if in() returned an error.  If
   strm->next_in is not Z_NULL, then the Z_BUF_ERROR was due to out() returning
   non-zero.  (in() will always be called before out(), so strm->next_in is
   assured to be defined if out() returns non-zero.) Note that inflateBack()
   cannot return Z_OK.
*/

ZEXTERN int ZEXPORT inflateBackEnd OF((z_streamp strm));
/*
     All memory allocated by inflateBackInit() is freed.

     inflateBackEnd() returns Z_OK on success, or Z_STREAM_ERROR if the stream
   state was inconsistent.
*/

ZEXTERN uLong ZEXPORT zlibCompileFlags OF((void));
/* Return flags indicating compile-time options.

    Type sizes, two bits each, 00 = 16 bits, 01 = 32, 10 = 64, 11 = other:
     1.0: size of uInt
     3.2: size of uLong
     5.4: size of voidpf (pointer)
     7.6: size of z_off_t

    Compiler, assembler, and debug options:
     8: DEBUG
     9: ASMV or ASMINF -- use ASM code
     10: ZLIB_WINAPI -- exported functions use the WINAPI calling convention
     11: 0 (reserved)

    One-time table building (smaller code, but not thread-safe if true):
     12: BUILDFIXED -- build static block decoding tables when needed
     13: DYNAMIC_CRC_TABLE -- build CRC calculation tables when needed
     14,15: 0 (reserved)

    Library content (indicates missing functionality):
     16: NO_GZCOMPRESS -- gz* functions cannot compress (to avoid linking
                          deflate code when not needed)
     17: NO_GZIP -- deflate can't write gzip streams, and inflate can't detect
                    and decode gzip streams (to avoid linking crc code)
     18-19: 0 (reserved)

    Operation variations (changes in library functionality):
     20: PKZIP_BUG_WORKAROUND -- slightly more permissive inflate
     21: FASTEST -- deflate algorithm with only one, lowest compression level
     22,23: 0 (reserved)

    The sprintf variant used by gzprintf (zero is best):
     24: 0 = vs*, 1 = s* -- 1 means limited to 20 arguments after the format
     25: 0 = *nprintf, 1 = *printf -- 1 means gzprintf() not secure!
     26: 0 = returns value, 1 = void -- 1 means inferred string length returned

    Remainder:
     27-31: 0 (reserved)
 */

#ifndef Z_SOLO

                        /* utility functions */

/*
     The following utility functions are implemented on top of the basic
   stream-oriented functions.  To simplify the interface, some default options
   are assumed (compression level and memory usage, standard memory allocation
   functions).  The source code of these utility functions can be modified if
   you need special options.
*/

ZEXTERN int ZEXPORT compress OF((Bytef *dest,   uLongf *destLen,
                                 const Bytef *source, uLong sourceLen));
/*
     Compresses the source buffer into the destination buffer.  sourceLen is
   the byte length of the source buffer.  Upon entry, destLen is the total size
   of the destination buffer, which must be at least the value returned by
   compressBound(sourceLen).  Upon exit, destLen is the actual size of the
   compressed buffer.

     compress returns Z_OK if success, Z_MEM_ERROR if there was not
   enough memory, Z_BUF_ERROR if there was not enough room in the output
   buffer.
*/

ZEXTERN int ZEXPORT compress2 OF((Bytef *dest,   uLongf *destLen,
                                  const Bytef *source, uLong sourceLen,
```

```
                                  int level));
/*
     Compresses the source buffer into the destination buffer.  The level
   parameter has the same meaning as in deflateInit.  sourceLen is the byte
   length of the source buffer.  Upon entry, destLen is the total size of the
   destination buffer, which must be at least the value returned by
   compressBound(sourceLen).  Upon exit, destLen is the actual size of the
   compressed buffer.

     compress2 returns Z_OK if success, Z_MEM_ERROR if there was not enough
   memory, Z_BUF_ERROR if there was not enough room in the output buffer,
   Z_STREAM_ERROR if the level parameter is invalid.
*/

ZEXTERN uLong ZEXPORT compressBound OF((uLong sourceLen));
/*
     compressBound() returns an upper bound on the compressed size after
   compress() or compress2() on sourceLen bytes.  It would be used before a
   compress() or compress2() call to allocate the destination buffer.
*/

ZEXTERN int ZEXPORT uncompress OF((Bytef *dest,   uLongf *destLen,
                                   const Bytef *source, uLong sourceLen));
/*
     Decompresses the source buffer into the destination buffer.  sourceLen is
   the byte length of the source buffer.  Upon entry, destLen is the total size
   of the destination buffer, which must be large enough to hold the entire
   uncompressed data.  (The size of the uncompressed data must have been saved
   previously by the compressor and transmitted to the decompressor by some
   mechanism outside the scope of this compression library.) Upon exit, destLen
   is the actual size of the uncompressed buffer.

     uncompress returns Z_OK if success, Z_MEM_ERROR if there was not
   enough memory, Z_BUF_ERROR if there was not enough room in the output
   buffer, or Z_DATA_ERROR if the input data was corrupted or incomplete.  In
   the case where there is not enough room, uncompress() will fill the output
   buffer with the uncompressed data up to that point.
*/

                        /* gzip file access functions */

/*
     This library supports reading and writing files in gzip (.gz) format with
   an interface similar to that of stdio, using the functions that start with
   "gz".  The gzip format is different from the zlib format.  gzip is a gzip
   wrapper, documented in RFC 1952, wrapped around a deflate stream.
*/

typedef struct gzFile_s *gzFile;    /* semi-opaque gzip file descriptor */

/*
ZEXTERN gzFile ZEXPORT gzopen OF((const char *path, const char *mode));

     Opens a gzip (.gz) file for reading or writing.  The mode parameter is as
   in fopen ("rb" or "wb") but can also include a compression level ("wb9") or
   a strategy: 'f' for filtered data as in "wb6f", 'h' for Huffman-only
   compression as in "wb1h", 'R' for run-length encoding as in "wb1R", or 'F'
   for fixed code compression as in "wb9F".  (See the description of
   deflateInit2 for more information about the strategy parameter.)  'T' will
   request transparent writing or appending with no compression and not using
   the gzip format.

     "a" can be used instead of "w" to request that the gzip stream that will
   be written be appended to the file.  "+" will result in an error, since
   reading and writing to the same gzip file is not supported.  The addition of
   "x" when writing will create the file exclusively, which fails if the file
   already exists.  On systems that support it, the addition of "e" when
   reading or writing will set the flag to close the file on an execve() call.

     These functions, as well as gzip, will read and decode a sequence of gzip
   streams in a file.  The append function of gzopen() can be used to create
   such a file.  (Also see gzflush() for another way to do this.)  When
   appending, gzopen does not test whether the file begins with a gzip stream,
   nor does it look for the end of the gzip streams to begin appending.  gzopen
   will simply append a gzip stream to the existing file.

     gzopen can be used to read a file which is not in gzip format; in this
   case gzread will directly read from the file without decompression.  When
   reading, this will be detected automatically by looking for the magic two-
   byte gzip header.

     gzopen returns NULL if the file could not be opened, if there was
   insufficient memory to allocate the gzFile state, or if an invalid mode was
   specified (an 'r', 'w', or 'a' was not provided, or '+' was provided).
   errno can be checked to determine if the reason gzopen failed was that the
   file could not be opened.
*/

ZEXTERN gzFile ZEXPORT gzdopen OF((int fd, const char *mode));
/*
     gzdopen associates a gzFile with the file descriptor fd.  File descriptors
   are obtained from calls like open, dup, creat, pipe or fileno (if the file
   has been previously opened with fopen).  The mode parameter is as in gzopen.

     The next call of gzclose on the returned gzFile will also close the file
   descriptor fd, just like fclose(fdopen(fd, mode)) closes the file descriptor
   fd.  If you want to keep fd open, use fd = dup(fd_keep); gz = gzdopen(fd,
   mode);.  The duplicated descriptor should be saved to avoid a leak, since
```

```
   gzdopen does not close fd if it fails.  If you are using fileno() to get the
   file descriptor from a FILE *, then you will have to use dup() to avoid
   double-close()ing the file descriptor.  Both gzclose() and fclose() will
   close the associated file descriptor, so they need to have different file
   descriptors.

     gzdopen returns NULL if there was insufficient memory to allocate the
   gzFile state, if an invalid mode was specified (an 'r', 'w', or 'a' was not
   provided, or '+' was provided), or if fd is -1.  The file descriptor is not
   used until the next gz* read, write, seek, or close operation, so gzdopen
   will not detect if fd is invalid (unless fd is -1).
*/

ZEXTERN int ZEXPORT gzbuffer OF((gzFile file, unsigned size));
/*
     Set the internal buffer size used by this library's functions.  The
   default buffer size is 8192 bytes.  This function must be called after
   gzopen() or gzdopen(), and before any other calls that read or write the
   file.  The buffer memory allocation is always deferred to the first read or
   write.  Two buffers are allocated, either both of the specified size when
   writing, or one of the specified size and the other twice that size when
   reading.  A larger buffer size of, for example, 64K or 128K bytes will
   noticeably increase the speed of decompression (reading).

     The new buffer size also affects the maximum length for gzprintf().

     gzbuffer() returns 0 on success, or -1 on failure, such as being called
   too late.
*/

ZEXTERN int ZEXPORT gzsetparams OF((gzFile file, int level, int strategy));
/*
     Dynamically update the compression level or strategy.  See the description
   of deflateInit2 for the meaning of these parameters.

     gzsetparams returns Z_OK if success, or Z_STREAM_ERROR if the file was not
   opened for writing.
*/

ZEXTERN int ZEXPORT gzread OF((gzFile file, voidp buf, unsigned len));
/*
     Reads the given number of uncompressed bytes from the compressed file.  If
   the input file is not in gzip format, gzread copies the given number of
   bytes into the buffer directly from the file.

     After reaching the end of a gzip stream in the input, gzread will continue
   to read, looking for another gzip stream.  Any number of gzip streams may be
   concatenated in the input file, and will all be decompressed by gzread().
   If something other than a gzip stream is encountered after a gzip stream,
   that remaining trailing garbage is ignored (and no error is returned).

     gzread can be used to read a gzip file that is being concurrently written.
   Upon reaching the end of the input, gzread will return with the available
   data.  If the error code returned by gzerror is Z_OK or Z_BUF_ERROR, then
   gzclearerr can be used to clear the end of file indicator in order to permit
   gzread to be tried again.  Z_OK indicates that a gzip stream was completed
   on the last gzread.  Z_BUF_ERROR indicates that the input file ended in the
   middle of a gzip stream.  Note that gzread does not return -1 in the event
   of an incomplete gzip stream.  This error is deferred until gzclose(), which
   will return Z_BUF_ERROR if the last gzread ended in the middle of a gzip
   stream.  Alternatively, gzerror can be used before gzclose to detect this
   case.

     gzread returns the number of uncompressed bytes actually read, less than
   len for end of file, or -1 for error.
*/

ZEXTERN int ZEXPORT gzwrite OF((gzFile file,
                                voidpc buf, unsigned len));
/*
     Writes the given number of uncompressed bytes into the compressed file.
   gzwrite returns the number of uncompressed bytes written or 0 in case of
   error.
*/

ZEXTERN int ZEXPORTVA gzprintf Z_ARG((gzFile file, const char *format, ...));
/*
     Converts, formats, and writes the arguments to the compressed file under
   control of the format string, as in fprintf.  gzprintf returns the number of
   uncompressed bytes actually written, or 0 in case of error.  The number of
   uncompressed bytes written is limited to 8191, or one less than the buffer
   size given to gzbuffer().  The caller should assure that this limit is not
   exceeded.  If it is exceeded, then gzprintf() will return an error (0) with
   nothing written.  In this case, there may also be a buffer overflow with
   unpredictable consequences, which is possible only if zlib was compiled with
   the insecure functions sprintf() or vsprintf() because the secure snprintf()
   or vsnprintf() functions were not available.  This can be determined using
   zlibCompileFlags().
*/

ZEXTERN int ZEXPORT gzputs OF((gzFile file, const char *s));
/*
     Writes the given null-terminated string to the compressed file, excluding
   the terminating null character.

     gzputs returns the number of characters written, or -1 in case of error.
*/
```

```
ZEXTERN char * ZEXPORT gzgets OF((gzFile file, char *buf, int len));
/*
     Reads bytes from the compressed file until len-1 characters are read, or a
   newline character is read and transferred to buf, or an end-of-file
   condition is encountered.  If any characters are read or if len == 1, the
   string is terminated with a null character.  If no characters are read due
   to an end-of-file or len < 1, then the buffer is left untouched.

     gzgets returns buf which is a null-terminated string, or it returns NULL
   for end-of-file or in case of error.  If there was an error, the contents at
   buf are indeterminate.
*/

ZEXTERN int ZEXPORT gzputc OF((gzFile file, int c));
/*
     Writes c, converted to an unsigned char, into the compressed file.  gzputc
   returns the value that was written, or -1 in case of error.
*/

ZEXTERN int ZEXPORT gzgetc OF((gzFile file));
/*
     Reads one byte from the compressed file.  gzgetc returns this byte or -1
   in case of end of file or error.  This is implemented as a macro for speed.
   As such, it does not do all of the checking the other functions do.  I.e.
   it does not check to see if file is NULL, nor whether the structure file
   points to has been clobbered or not.
*/

ZEXTERN int ZEXPORT gzungetc OF((int c, gzFile file));
/*
     Push one character back onto the stream to be read as the first character
   on the next read.  At least one character of push-back is allowed.
   gzungetc() returns the character pushed, or -1 on failure.  gzungetc() will
   fail if c is -1, and may fail if a character has been pushed but not read
   yet.  If gzungetc is used immediately after gzopen or gzdopen, at least the
   output buffer size of pushed characters is allowed.  (See gzbuffer above.)
   The pushed character will be discarded if the stream is repositioned with
   gzseek() or gzrewind().
*/

ZEXTERN int ZEXPORT gzflush OF((gzFile file, int flush));
/*
     Flushes all pending output into the compressed file.  The parameter flush
   is as in the deflate() function.  The return value is the zlib error number
   (see function gzerror below).  gzflush is only permitted when writing.

     If the flush parameter is Z_FINISH, the remaining data is written and the
   gzip stream is completed in the output.  If gzwrite() is called again, a new
   gzip stream will be started in the output.  gzread() is able to read such
   concatented gzip streams.

     gzflush should be called only when strictly necessary because it will
   degrade compression if called too often.
*/

/*
ZEXTERN z_off_t ZEXPORT gzseek OF((gzFile file,
                                   z_off_t offset, int whence));

     Sets the starting position for the next gzread or gzwrite on the given
   compressed file.  The offset represents a number of bytes in the
   uncompressed data stream.  The whence parameter is defined as in lseek(2);
   the value SEEK_END is not supported.

     If the file is opened for reading, this function is emulated but can be
   extremely slow.  If the file is opened for writing, only forward seeks are
   supported; gzseek then compresses a sequence of zeroes up to the new
   starting position.

     gzseek returns the resulting offset location as measured in bytes from
   the beginning of the uncompressed stream, or -1 in case of error, in
   particular if the file is opened for writing and the new starting position
   would be before the current position.
*/

ZEXTERN int ZEXPORT    gzrewind OF((gzFile file));
/*
     Rewinds the given file. This function is supported only for reading.

     gzrewind(file) is equivalent to (int)gzseek(file, 0L, SEEK_SET)
*/

/*
ZEXTERN z_off_t ZEXPORT    gztell OF((gzFile file));

     Returns the starting position for the next gzread or gzwrite on the given
   compressed file.  This position represents a number of bytes in the
   uncompressed data stream, and is zero when starting, even if appending or
   reading a gzip stream from the middle of a file using gzdopen().

     gztell(file) is equivalent to gzseek(file, 0L, SEEK_CUR)
*/

/*
ZEXTERN z_off_t ZEXPORT gzoffset OF((gzFile file));

     Returns the current offset in the file being read or written.  This offset
   includes the count of bytes that precede the gzip stream, for example when
```

```
   appending or when using gzdopen() for reading.  When reading, the offset
   does not include as yet unused buffered input.  This information can be used
   for a progress indicator.  On error, gzoffset() returns -1.
*/

ZEXTERN int ZEXPORT gzeof OF((gzFile file));
/*
     Returns true (1) if the end-of-file indicator has been set while reading,
   false (0) otherwise.  Note that the end-of-file indicator is set only if the
   read tried to go past the end of the input, but came up short.  Therefore,
   just like feof(), gzeof() may return false even if there is no more data to
   read, in the event that the last read request was for the exact number of
   bytes remaining in the input file.  This will happen if the input file size
   is an exact multiple of the buffer size.

     If gzeof() returns true, then the read functions will return no more data,
   unless the end-of-file indicator is reset by gzclearerr() and the input file
   has grown since the previous end of file was detected.
*/

ZEXTERN int ZEXPORT gzdirect OF((gzFile file));
/*
     Returns true (1) if file is being copied directly while reading, or false
   (0) if file is a gzip stream being decompressed.

     If the input file is empty, gzdirect() will return true, since the input
   does not contain a gzip stream.

     If gzdirect() is used immediately after gzopen() or gzdopen() it will
   cause buffers to be allocated to allow reading the file to determine if it
   is a gzip file.  Therefore if gzbuffer() is used, it should be called before
   gzdirect().

     When writing, gzdirect() returns true (1) if transparent writing was
   requested ("wT" for the gzopen() mode), or false (0) otherwise.  (Note:
   gzdirect() is not needed when writing.  Transparent writing must be
   explicitly requested, so the application already knows the answer.  When
   linking statically, using gzdirect() will include all of the zlib code for
   gzip file reading and decompression, which may not be desired.)
*/

ZEXTERN int ZEXPORT    gzclose OF((gzFile file));
/*
     Flushes all pending output if necessary, closes the compressed file and
   deallocates the (de)compression state.  Note that once file is closed, you
   cannot call gzerror with file, since its structures have been deallocated.
   gzclose must not be called more than once on the same file, just as free
   must not be called more than once on the same allocation.

     gzclose will return Z_STREAM_ERROR if file is not valid, Z_ERRNO on a
   file operation error, Z_MEM_ERROR if out of memory, Z_BUF_ERROR if the
   last read ended in the middle of a gzip stream, or Z_OK on success.
*/

ZEXTERN int ZEXPORT gzclose_r OF((gzFile file));
ZEXTERN int ZEXPORT gzclose_w OF((gzFile file));
/*
     Same as gzclose(), but gzclose_r() is only for use when reading, and
   gzclose_w() is only for use when writing or appending.  The advantage to
   using these instead of gzclose() is that they avoid linking in zlib
   compression or decompression code that is not used when only reading or only
   writing respectively.  If gzclose() is used, then both compression and
   decompression code will be included the application when linking to a static
   zlib library.
*/

ZEXTERN const char * ZEXPORT gzerror OF((gzFile file, int *errnum));
/*
     Returns the error message for the last error which occurred on the given
   compressed file.  errnum is set to zlib error number.  If an error occurred
   in the file system and not in the compression library, errnum is set to
   Z_ERRNO and the application may consult errno to get the exact error code.

     The application must not modify the returned string.  Future calls to
   this function may invalidate the previously returned string.  If file is
   closed, then the string previously returned by gzerror will no longer be
   available.

     gzerror() should be used to distinguish errors from end-of-file for those
   functions above that do not distinguish those cases in their return values.
*/

ZEXTERN void ZEXPORT gzclearerr OF((gzFile file));
/*
     Clears the error and end-of-file flags for file.  This is analogous to the
   clearerr() function in stdio.  This is useful for continuing to read a gzip
   file that is being written concurrently.
*/

#endif /* !Z_SOLO */

                        /* checksum functions */

/*
     These functions are not related to compression but are exported
   anyway because they might be useful in applications using the compression
   library.
*/
```

```
ZEXTERN uLong ZEXPORT adler32 OF((uLong adler, const Bytef *buf, uInt len));
/*
     Update a running Adler-32 checksum with the bytes buf[0..len-1] and
   return the updated checksum.  If buf is Z_NULL, this function returns the
   required initial value for the checksum.

     An Adler-32 checksum is almost as reliable as a CRC32 but can be computed
   much faster.

   Usage example:

     uLong adler = adler32(0L, Z_NULL, 0);

     while (read_buffer(buffer, length) != EOF) {
       adler = adler32(adler, buffer, length);
     }
     if (adler != original_adler) error();
*/

/*
ZEXTERN uLong ZEXPORT adler32_combine OF((uLong adler1, uLong adler2,
                                          z_off_t len2));

     Combine two Adler-32 checksums into one.  For two sequences of bytes, seq1
   and seq2 with lengths len1 and len2, Adler-32 checksums were calculated for
   each, adler1 and adler2.  adler32_combine() returns the Adler-32 checksum of
   seq1 and seq2 concatenated, requiring only adler1, adler2, and len2.  Note
   that the z_off_t type (like off_t) is a signed integer.  If len2 is
   negative, the result has no meaning or utility.
*/

ZEXTERN uLong ZEXPORT crc32   OF((uLong crc, const Bytef *buf, uInt len));
/*
     Update a running CRC-32 with the bytes buf[0..len-1] and return the
   updated CRC-32.  If buf is Z_NULL, this function returns the required
   initial value for the crc.  Pre- and post-conditioning (one's complement) is
   performed within this function so it shouldn't be done by the application.

   Usage example:

     uLong crc = crc32(0L, Z_NULL, 0);

     while (read_buffer(buffer, length) != EOF) {
       crc = crc32(crc, buffer, length);
     }
     if (crc != original_crc) error();
*/

/*
ZEXTERN uLong ZEXPORT crc32_combine OF((uLong crc1, uLong crc2, z_off_t len2));

     Combine two CRC-32 check values into one.  For two sequences of bytes,
   seq1 and seq2 with lengths len1 and len2, CRC-32 check values were
   calculated for each, crc1 and crc2.  crc32_combine() returns the CRC-32
   check value of seq1 and seq2 concatenated, requiring only crc1, crc2, and
   len2.
*/


                        /* various hacks, don't look :) */

/* deflateInit and inflateInit are macros to allow checking the zlib version
 * and the compiler's view of z_stream:
 */
ZEXTERN int ZEXPORT deflateInit_ OF((z_streamp strm, int level,
                                     const char *version, int stream_size));
ZEXTERN int ZEXPORT inflateInit_ OF((z_streamp strm,
                                     const char *version, int stream_size));
ZEXTERN int ZEXPORT deflateInit2_ OF((z_streamp strm, int  level, int  method,
                                      int windowBits, int memLevel,
                                      int strategy, const char *version,
                                      int stream_size));
ZEXTERN int ZEXPORT inflateInit2_ OF((z_streamp strm, int  windowBits,
                                      const char *version, int stream_size));
ZEXTERN int ZEXPORT inflateBackInit_ OF((z_streamp strm, int windowBits,
                                         unsigned char FAR *window,
                                         const char *version,
                                         int stream_size));
#define deflateInit(strm, level) \
        deflateInit_((strm), (level), ZLIB_VERSION, (int)sizeof(z_stream))
#define inflateInit(strm) \
        inflateInit_((strm), ZLIB_VERSION, (int)sizeof(z_stream))
#define deflateInit2(strm, level, method, windowBits, memLevel, strategy) \
        deflateInit2_((strm),(level),(method),(windowBits),(memLevel),\
                      (strategy), ZLIB_VERSION, (int)sizeof(z_stream))
#define inflateInit2(strm, windowBits) \
        inflateInit2_((strm), (windowBits), ZLIB_VERSION, \
                      (int)sizeof(z_stream))
#define inflateBackInit(strm, windowBits, window) \
        inflateBackInit_((strm), (windowBits), (window), \
                      ZLIB_VERSION, (int)sizeof(z_stream))

#ifndef Z_SOLO

/* gzgetc() macro and its supporting function and exposed data structure.  Note
 * that the real internal state is much larger than the exposed structure.
 * This abbreviated structure exposes just enough for the gzgetc() macro.  The
```

```
 * user should not mess with these exposed elements, since their names or
 * behavior could change in the future, perhaps even capriciously.  They can
 * only be used by the gzgetc() macro.  You have been warned.
 */
struct gzFile_s {
    unsigned have;
    unsigned char *next;
    z_off64_t pos;
};
ZEXTERN int ZEXPORT gzgetc_ OF((gzFile file));  /* backward compatibility */
#ifdef Z_PREFIX_SET
#  undef z_gzgetc
#  define z_gzgetc(g) \
          ((g)->have ? ((g)->have--, (g)->pos++, *((g)->next)++) : gzgetc(g))
#else
#  define gzgetc(g) \
          ((g)->have ? ((g)->have--, (g)->pos++, *((g)->next)++) : gzgetc(g))
#endif

/* provide 64-bit offset functions if _LARGEFILE64_SOURCE defined, and/or
 * change the regular functions to 64 bits if _FILE_OFFSET_BITS is 64 (if
 * both are true, the application gets the *64 functions, and the regular
 * functions are changed to 64 bits) -- in case these are set on systems
 * without large file support, _LFS64_LARGEFILE must also be true
 */
#ifdef Z_LARGE64
   ZEXTERN gzFile ZEXPORT gzopen64 OF((const char *, const char *));
   ZEXTERN z_off64_t ZEXPORT gzseek64 OF((gzFile, z_off64_t, int));
   ZEXTERN z_off64_t ZEXPORT gztell64 OF((gzFile));
   ZEXTERN z_off64_t ZEXPORT gzoffset64 OF((gzFile));
   ZEXTERN uLong ZEXPORT adler32_combine64 OF((uLong, uLong, z_off64_t));
   ZEXTERN uLong ZEXPORT crc32_combine64 OF((uLong, uLong, z_off64_t));
#endif

#if !defined(ZLIB_INTERNAL) && defined(Z_WANT64)
#  ifdef Z_PREFIX_SET
#    define z_gzopen z_gzopen64
#    define z_gzseek z_gzseek64
#    define z_gztell z_gztell64
#    define z_gzoffset z_gzoffset64
#    define z_adler32_combine z_adler32_combine64
#    define z_crc32_combine z_crc32_combine64
#  else
#    define gzopen gzopen64
#    define gzseek gzseek64
#    define gztell gztell64
#    define gzoffset gzoffset64
#    define adler32_combine adler32_combine64
#    define crc32_combine crc32_combine64
#  endif
#  ifndef Z_LARGE64
     ZEXTERN gzFile ZEXPORT gzopen64 OF((const char *, const char *));
     ZEXTERN z_off_t ZEXPORT gzseek64 OF((gzFile, z_off_t, int));
     ZEXTERN z_off_t ZEXPORT gztell64 OF((gzFile));
     ZEXTERN z_off_t ZEXPORT gzoffset64 OF((gzFile));
     ZEXTERN uLong ZEXPORT adler32_combine64 OF((uLong, uLong, z_off_t));
     ZEXTERN uLong ZEXPORT crc32_combine64 OF((uLong, uLong, z_off_t));
#  endif
#else
   ZEXTERN gzFile ZEXPORT gzopen OF((const char *, const char *));
   ZEXTERN z_off_t ZEXPORT gzseek OF((gzFile, z_off_t, int));
   ZEXTERN z_off_t ZEXPORT gztell OF((gzFile));
   ZEXTERN z_off_t ZEXPORT gzoffset OF((gzFile));
   ZEXTERN uLong ZEXPORT adler32_combine OF((uLong, uLong, z_off_t));
   ZEXTERN uLong ZEXPORT crc32_combine OF((uLong, uLong, z_off_t));
#endif

#else /* Z_SOLO */

   ZEXTERN uLong ZEXPORT adler32_combine OF((uLong, uLong, z_off_t));
   ZEXTERN uLong ZEXPORT crc32_combine OF((uLong, uLong, z_off_t));

#endif /* !Z_SOLO */

/* hack for buggy compilers */
#if !defined(ZUTIL_H) && !defined(NO_DUMMY_DECL)
    struct internal_state {int dummy;};
#endif

/* undocumented functions */
ZEXTERN const char   * ZEXPORT zError           OF((int));
ZEXTERN int            ZEXPORT inflateSyncPoint OF((z_streamp));
ZEXTERN const z_crc_t FAR * ZEXPORT get_crc_table    OF((void));
ZEXTERN int            ZEXPORT inflateUndermine OF((z_streamp, int));
ZEXTERN int            ZEXPORT inflateResetKeep OF((z_streamp));
ZEXTERN int            ZEXPORT deflateResetKeep OF((z_streamp));
#if defined(_WIN32) && !defined(Z_SOLO)
ZEXTERN gzFile         ZEXPORT gzopen_w OF((const wchar_t *path,
                                            const char *mode));
#endif
#if defined(STDC) || defined(Z_HAVE_STDARG_H)
#  ifndef Z_SOLO
ZEXTERN int            ZEXPORTVA gzvprintf Z_ARG((gzFile file,
                                                  const char *format,
                                                  va_list va));
#  endif
#endif
```

```
#ifdef __cplusplus
}
#endif

#endif /* ZLIB_H */
```

Notice for package(s)

zlib

```
zlib License


  This software is provided `as-is`, without any express or implied
  warranty.  In no event will the authors be held liable for any damages
  arising from the use of this software.

  Permission is granted to anyone to use this software for any purpose,
  including commercial applications, and to alter it and redistribute it
  freely, subject to the following restrictions:

  1. The origin of this software must not be misrepresented; you must not
     claim that you wrote the original software. If you use this software
     in a product, an acknowledgment in the product documentation would be
     appreciated but is not required.
  2. Altered source versions must be plainly marked as such, and must not be
     misrepresented as being the original software.
  3. This notice may not be removed or altered from any source distribution.
```

Notice for package(s)

base-files
bc
opkg-utils
stat
usbutils

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
     59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
```

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

  10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the

library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

Notice for package(s)

acl
arptables
attr
babeltrace
base-files
base-passwd
bc
bridge-utils
busybox
cryptodev-linux
e2fsprogs
ethtool
gmp
init-ifupdown
initscripts
iproute2
iptables
iputils
kbd
kexec-tools
keymaps
kmod
libcap
libnftnl
libtool
lsb
lsbinitscripts
lttng-modules
lttng-tools
lttng-ust
lzo
mtd-utils
netbase
nettle
nfs-utils
nftables
opkg-utils
pciutils
pps-tools
procps
psmisc
quota
stat
sysfsutils
sysvinit
sysvinit-inittab
udev
update-rc.d
usbutils
util-linux
xz

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is i

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have th

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you hav

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, di

Also, for each author`s protection and ours, we want to make certain that everyone understands that there is no warranty for this free softwar

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running t

1. You may copy and distribute verbatim copies of the Program`s source code as you receive it, in any medium, provided that you conspicuously

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute s

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part therec
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be r

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to ex

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Secti

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 ab
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physical
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only f
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code m

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to co

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to c

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distrib

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), condi

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to a

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the origin

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions wil

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WH

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it fre

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

one line to give the program`s name and an idea of what it does.
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w`. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c`
for details.
The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you u

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if n

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision`
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you
```

Notice for package(s)

update-rc.d

```
#!/bin/sh
#
# update-rc.d   Update the links in /etc/rc[0-9S].d/
#
# (c) 2003, 2004 Phil Blundell <pb@handhelds.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

initd="/etc/init.d"
etcd="/etc/rc"
notreally=0
force=0
dostart=0
verbose=0

usage()
{
        cat >&2 <<EOF
usage: update-rc.d [-n] [-f] [-r <root>] <basename> remove
       update-rc.d [-n] [-r <root>] [-s] <basename> defaults [NN | sNN kNN]
       update-rc.d [-n] [-r <root>] [-s] <basename> start|stop NN runlvl [runlvl] [...] .
                -n: not really
                -f: force
                -v: verbose
                -r: alternate root path (default is /)
                -s: invoke start methods if appropriate to current runlevel
EOF
}

checklinks()
{
        local i dn fn remove=0
        if [ "x$1" = "xremove" ]; then
                echo " Removing any system startup links for $bn ..."
                remove=1
        fi

        for i in 0 1 2 3 4 5 6 7 8 9 S; do
                dn="${etcd}${i}.d"
                if [ ! -d $dn ]; then
                        continue;
                fi
                for f in ${dn}/[SK]??${bn}; do
                        if [ -L $f ]; then
                                if [ $remove -eq 0 ]; then
                                        return 1
                                fi
                                echo "  $f"
                                if [ $notreally -eq 1 ]; then
                                        continue
                                fi
                                rm $f
                        fi
                done
        done

        return 0
}

dolink()
{
        startstop=$1
        lev=`echo $2 | cut -d/ -f1`
        nn=`echo $2 | cut -d/ -f2`
        fn="${etcd}${lev}.d/${startstop}${nn}${bn}"
        [ $verbose -eq 1 ] && echo "  $fn -> ../init.d/$bn"
        if [ $notreally -eq 0 ]; then
                mkdir -p `dirname $fn`
```

```
                ln -s ../init.d/$bn $fn
        fi
        if [ $dostart -eq 1 ] && [ $startstop = "S" ] && [ $lev = $RUNLEVEL ]; then
                $fn start || true
        fi
}

makelinks()
{
        if ! checklinks; then
                echo " System startup links for $initd/$bn already exist."
                if [ $dostart -eq 1 ] && [ $notreally -eq 0 ] && [ -L ${etcd}${RUNLEVEL}.d/S??${bn} ]; then
                        ${etcd}${RUNLEVEL}.d/S??${bn} restart || true
                fi
                exit 0
        fi

        echo " Adding system startup for $initd/$bn."

        for i in $startlinks; do
                dolink S $i
        done
        for i in $stoplinks; do
                dolink K $i
        done
}

while [ $# -gt 0 ]; do
        case $1 in
                -n)     notreally=1
                        shift
                        continue
                        ;;
                -v)     verbose=1
                        shift
                        continue
                        ;;
                -f)     force=1
                        shift
                        continue
                        ;;
                -s)     dostart=1
                        shift
                        continue
                        ;;
                -r)     shift
                        root=$1
                        initd="${root}${initd}"
                        etcd="${root}${etcd}"
                        shift
                        ;;
                -h | --help)
                        usage
                        exit 0
                        ;;
                -*)
                        usage
                        exit 1
                        ;;
                *)
                        break
                        ;;
        esac
done

if [ $# -lt 2 ]; then
        usage
        exit 1
fi

bn=$1
shift

sn=$initd/$bn
if [ -L "$sn" -a -n "$root" ]; then
        if which readlink >/dev/null; then
                while true; do
                        linksn="$(readlink "$sn")"
                        if [ -z "$linksn" ]; then
                                break
                        fi

                        sn="$linksn"
                        case "$sn" in
                                /*) sn="$root$sn" ;;
                                *)  sn="$initd/$sn" ;;
                        esac
                done
        else
                echo "update-rc.d: readlink tool not present, cannot check whether \
                        $sn symlink points to a valid file." >&2
        fi
fi

if [ $1 != "remove" ]; then
        if [ ! -f "$sn" ]; then
                echo "update-rc.d: $initd/$bn: file does not exist" >&2
                exit 1
```

```
        fi
else
        if [ -f "$sn" ]; then
                if [ $force -eq 1 ]; then
                        echo "update-rc.d: $initd/$bn exists during rc.d purge (continuing)" >&2
                else
                        echo "update-rc.d: $initd/$bn exists during rc.d purge (use -f to force)" >&2
                        exit 1
                fi
        fi
fi

if [ $dostart -eq 1 ]; then
        #RUNLEVEL=`sed 's/.*\[\(.*\)\]/\1/' < /proc/1/cmdline`
        RUNLEVEL=`runlevel | cut -d" " -f2`
        if [ "x$RUNLEVEL" = "x" ]; then
                echo "Unable to determine current runlevel" >&2
                exit 1
        fi
fi

case $1 in
        remove)
                checklinks "remove"
                ;;

        defaults)
                if [ $# -gt 3 ]; then
                        echo "defaults takes only one or two arguments" >&2
                        usage
                        exit 1
                fi
                start=20
                stop=20
                if [ $# -gt 1 ]; then
                        start=$2
                        stop=$2
                fi
                if [ $# -gt 2 ]; then
                        stop=$3
                fi
                start=`printf %02d $start`
                stop=`printf %02d $stop`
                stoplinks="0/$stop 1/$stop 6/$stop"
                startlinks="2/$start 3/$start 4/$start 5/$start"
                makelinks
                ;;

        start | stop)
                while [ $# -gt 0 ]; do
                        if [ $1 = "start" ]; then
                                letter=S
                        elif [ $1 = "stop" ]; then
                                letter=K
                        else
                                echo "expected start or stop" >&2
                                usage
                                exit 1
                        fi
                        shift
                        NN=`printf %02d $(expr $1 + 0)`
                        shift
                        while [ "x$1" != "x." ]; do
                                if [ $# -eq 0 ]; then
                                        echo "action with list of runlevels not terminated by \`.'" >&2
                                        exit 1
                                fi
                                level=$1
                                shift
                                case $letter in
                                        S) startlinks="$startlinks $level/$NN" ;;
                                        K) stoplinks="$stoplinks $level/$NN" ;;
                                esac
                        done
                        shift
                done
                makelinks
                ;;

        *)
                usage
                exit 1
                ;;
esac
```

Notice for package(s)

run-postinsts
shadow-securetty
udev-extraconf

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Notice for package(s)

babeltrace
e2fsprogs
libffi
liburcu
libxml2
lttng-modules
lttng-ust
ncurses
nfs-utils
popt
run-postinsts
shadow-securetty
udev-extraconf
udev-rules-qoriq

MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Notice for package(s)

base-passwd
iproute2

                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code. (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

  5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

  7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any

patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this

```
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

shadow

```
NOTE:
  This license has been obsoleted by the change to the BSD-style copyright.
  You may continue to use this license if you wish, but you are under no
  obligation to do so.

(*
This document is freely plagiarised from the 'Artistic Licence',
distributed as part of the Perl v4.0 kit by Larry Wall, which is
available from most major archive sites.  I stole it from CrackLib.

        $Id$
*)

This documents purpose is to state the conditions under which this
Package (See definition below) viz: "Shadow", the Shadow Password Suite
which is held by Julianne Frances Haugh, may be copied, such that the
copyright holder maintains some semblance of artistic control over the
development of the package, while giving the users of the package the
right to use and distribute the Package in a more-or-less customary
fashion, plus the right to make reasonable modifications.

So there.

*****************************************************************************

Definitions:


A "Package" refers to the collection of files distributed by the
Copyright Holder, and derivatives of that collection of files created
through textual modification, or segments thereof.

"Standard Version" refers to such a Package if it has not been modified,
or has been modified in accordance with the wishes of the Copyright
Holder.

"Copyright Holder" is whoever is named in the copyright or copyrights
for the package.

"You" is you, if you're thinking about copying or distributing this
Package.

"Reasonable copying fee" is whatever you can justify on the basis of
media cost, duplication charges, time of people involved, and so on.
(You will not be required to justify it to the Copyright Holder, but
only to the computing community at large as a market that must bear the
fee.)

"Freely Available" means that no fee is charged for the item itself,
though there may be fees involved in handling the item.  It also means
that recipients of the item may redistribute it under the same
conditions they received it.


1.  You may make and give away verbatim copies of the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated
disclaimers.
```

2. You may apply bug fixes, portability fixes and other modifications
derived from the Public Domain or from the Copyright Holder. A Package
modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way,
provided that you insert a prominent notice in each changed file stating
how and when AND WHY you changed that file, and provided that you do at
least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or an
equivalent medium, or placing the modifications on a major archive site
such as uunet.uu.net, or by allowing the Copyright Holder to include
your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with
standard executables, which must also be provided, and provide separate
documentation for each non-standard executable that clearly documents
how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or
executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files,
together with instructions (in the manual page or equivalent) on where
to get the Standard Version.

b) accompany the distribution with the machine-readable source of the
Package with your modifications.

c) accompany any non-standard executables with their corresponding
Standard Version executables, giving the non-standard executables
non-standard names, and clearly documenting the differences in manual
pages (or equivalent), together with instructions on where to get the
Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this
Package. You may charge any fee you choose for support of this Package.
YOU MAY NOT CHARGE A FEE FOR THIS PACKAGE ITSELF. However, you may
distribute this Package in aggregate with other (possibly commercial)
programs as part of a larger (possibly commercial) software distribution
provided that YOU DO NOT ADVERTISE this package as a product of your
own.

6. The name of the Copyright Holder may not be used to endorse or
promote products derived from this software without specific prior
written permission.

7. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

                                The End

Notice for package(s)

shadow

```
/*
 * Copyright (c) 1989 - 1994, Julianne Frances Haugh
 * Copyright (c) 1996 - 2000, Marek Michałkiewicz
 * Copyright (c) 2001 - 2006, Tomasz Kłoczko
 * Copyright (c) 2007 - 2011, Nicolas François
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the copyright holders or contributors may not be used to
 *    endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
 * HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
```

```
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

#include <config.h>

#ident "$Id$"

#include <errno.h>
#include <fcntl.h>
#include <getopt.h>
#include <pwd.h>
#include <signal.h>
#include <stdio.h>
#include <sys/types.h>
#ifdef WITH_SELINUX
#include <selinux/selinux.h>
#include <selinux/flask.h>
#include <selinux/av_permissions.h>
#include <selinux/context.h>
#endif                          /* WITH_SELINUX */
#include <time.h>
#include "defines.h"
#include "getdef.h"
#include "nscd.h"
#include "prototypes.h"
#include "pwauth.h"
#include "pwio.h"
#include "shadowio.h"

/*
 * exit status values
 */
/*@-exitarg@*/
#define E_SUCCESS       0       /* success */
#define E_NOPERM        1       /* permission denied */
#define E_USAGE         2       /* invalid combination of options */
#define E_FAILURE       3       /* unexpected failure, nothing done */
#define E_MISSING       4       /* unexpected failure, passwd file missing */
#define E_PWDBUSY       5       /* passwd file busy, try again later */
#define E_BAD_ARG       6       /* invalid argument to option */
/*
 * Global variables
 */
const char *Prog;               /* Program name */

static char *name;              /* The name of user whose password is being changed */
static char *myname;            /* The current user's name */
static bool amroot;             /* The caller's real UID was 0 */

static bool
    aflg = false,                       /* -a - show status for all users */
    dflg = false,                       /* -d - delete password */
    eflg = false,                       /* -e - force password change */
    iflg = false,                       /* -i - set inactive days */
    kflg = false,                       /* -k - change only if expired */
    lflg = false,                       /* -l - lock the user's password */
    nflg = false,                       /* -n - set minimum days */
    qflg = false,                       /* -q - quiet mode */
    Sflg = false,                       /* -S - show password status */
    uflg = false,                       /* -u - unlock the user's password */
    wflg = false,                       /* -w - set warning days */
    xflg = false;                       /* -x - set maximum days */

/*
 * set to 1 if there are any flags which require root privileges,
 * and require username to be specified
 */
static bool anyflag = false;

static long age_min = 0;        /* Minimum days before change   */
static long age_max = 0;        /* Maximum days until change     */
static long warn = 0;           /* Warning days before change   */
static long inact = 0;          /* Days without change before locked */

#ifndef USE_PAM
static bool do_update_age = false;
#endif                          /* ! USE_PAM */

static bool pw_locked = false;
static bool spw_locked = false;

#ifndef USE_PAM
/*
 * Size of the biggest passwd:
 *   $6$        3
 *   rounds=    7
 *   999999999  9
 *   $          1
 *   salt       16
 *   $          1
 *   SHA512     123
 *   nul        1
 *
 *   total      161
 */
static char crypt_passwd[256];
```

```
static bool do_update_pwd = false;
#endif                          /* !USE_PAM */

/*
 * External identifiers
 */

/* local function prototypes */
static /*@noreturn@*/void usage (int);

#ifndef USE_PAM
static bool reuse (const char *, const struct passwd *);
static int new_password (const struct passwd *);

static void check_password (const struct passwd *, const struct spwd *);
#endif                          /* !USE_PAM */
static /*@observer@*/const char *date_to_str (time_t);
static /*@observer@*/const char *pw_status (const char *);
static void print_status (const struct passwd *);
static /*@noreturn@*/void fail_exit (int);
static /*@noreturn@*/void oom (void);
static char *update_crypt_pw (char *);
static void update_noshadow (void);

static void update_shadow (void);
#ifdef WITH_SELINUX
static int check_selinux_access (const char *changed_user,
                                 uid_t changed_uid,
                                 access_vector_t requested_access);
#endif                          /* WITH_SELINUX */

/*
 * usage - print command usage and exit
 */
static /*@noreturn@*/void usage (int status)
{
        FILE *usageout = (E_SUCCESS != status) ? stderr : stdout;
        (void) fprintf (usageout,
                        _("Usage: %s [options] [LOGIN]\n"
                          "\n"
                          "Options:\n"),
                        Prog);
        (void) fputs (_("  -a, --all                    report password status on all accounts\n"), usageout);
        (void) fputs (_("  -d, --delete                 delete the password for the named account\n"), usageout);
        (void) fputs (_("  -e, --expire                 force expire the password for the named account\n"), usageout);
        (void) fputs (_("  -h, --help                   display this help message and exit\n"), usageout);
        (void) fputs (_("  -k, --keep-tokens            change password only if expired\n"), usageout);
        (void) fputs (_("  -i, --inactive INACTIVE      set password inactive after expiration\n"
                        "                               to INACTIVE\n"), usageout);
        (void) fputs (_("  -l, --lock                   lock the password of the named account\n"), usageout);
        (void) fputs (_("  -n, --mindays MIN_DAYS       set minimum number of days before password\n"
                        "                               change to MIN_DAYS\n"), usageout);
        (void) fputs (_("  -q, --quiet                  quiet mode\n"), usageout);
        (void) fputs (_("  -r, --repository REPOSITORY  change password in REPOSITORY repository\n"), usageout);
        (void) fputs (_("  -R, --root CHROOT_DIR        directory to chroot into\n"), usageout);
        (void) fputs (_("  -S, --status                 report password status on the named account\n"), usageout);
        (void) fputs (_("  -u, --unlock                 unlock the password of the named account\n"), usageout);
        (void) fputs (_("  -w, --warndays WARN_DAYS     set expiration warning days to WARN_DAYS\n"), usageout);
        (void) fputs (_("  -x, --maxdays MAX_DAYS       set maximum number of days before password\n"
                        "                               change to MAX_DAYS\n"), usageout);
        (void) fputs ("\n", usageout);
        exit (status);
}

#ifndef USE_PAM
static bool reuse (const char *pass, const struct passwd *pw)
{
#ifdef HAVE_LIBCRACK_HIST
        const char *reason;

#ifdef HAVE_LIBCRACK_PW
        const char *FascistHistoryPw (const char *, const struct passwd *);

        reason = FascistHistory (pass, pw);
#else                           /* !HAVE_LIBCRACK_PW */
        const char *FascistHistory (const char *, int);

        reason = FascistHistory (pass, pw->pw_uid);
#endif                          /* !HAVE_LIBCRACK_PW */
        if (NULL != reason) {
                (void) printf (_("Bad password: %s.  "), reason);
                return true;
        }
#endif                          /* HAVE_LIBCRACK_HIST */
        return false;
}

/*
 * new_password - validate old password and replace with new (both old and
 * new in global "char crypt_passwd[128]")
 */
static int new_password (const struct passwd *pw)
{
        char *clear;            /* Pointer to clear text */
        char *cipher;           /* Pointer to cipher text */
        const char *salt;       /* Pointer to new salt */
        char *cp;               /* Pointer to getpass() response */
        char orig[200];         /* Original password */
```

```
        char pass[200];         /* New password */
        int i;                  /* Counter for retries */
        bool warned;
        int pass_max_len = -1;
        const char *method;

#ifdef HAVE_LIBCRACK_HIST
        int HistUpdate (const char *, const char *);
#endif                          /* HAVE_LIBCRACK_HIST */

        /*
         * Authenticate the user. The user will be prompted for their own
         * password.
         */

        if (!amroot && ('\0' != crypt_passwd[0])) {
                clear = getpass (_("Old password: "));
                if (NULL == clear) {
                        return -1;
                }

                cipher = pw_encrypt (clear, crypt_passwd);

                if (NULL == cipher) {
                        strzero (clear);
                        fprintf (stderr,
                                 _("%s: failed to crypt password with previous salt: %s\n"),
                                 Prog, strerror (errno));
                        SYSLOG ((LOG_INFO,
                                 "Failed to crypt password with previous salt of user '%s'",
                                 pw->pw_name));
                        return -1;
                }

                if (strcmp (cipher, crypt_passwd) != 0) {
                        strzero (clear);
                        strzero (cipher);
                        SYSLOG ((LOG_WARN, "incorrect password for %s",
                                 pw->pw_name));
                        (void) sleep (1);
                        (void) fprintf (stderr,
                                        _("Incorrect password for %s.\n"),
                                        pw->pw_name);
                        return -1;
                }
                STRFCPY (orig, clear);
                strzero (clear);
                strzero (cipher);
        } else {
                orig[0] = '\0';
        }

        /*
         * Get the new password. The user is prompted for the new password
         * and has five tries to get it right. The password will be tested
         * for strength, unless it is the root user. This provides an escape
         * for initial login passwords.
         */
        method = getdef_str ("ENCRYPT_METHOD");
        if (NULL == method) {
                if (!getdef_bool ("MD5_CRYPT_ENAB")) {
                        pass_max_len = getdef_num ("PASS_MAX_LEN", 8);
                }
        } else {
                if (   (strcmp (method, "MD5")    == 0)
#ifdef USE_SHA_CRYPT
                    || (strcmp (method, "SHA256") == 0)
                    || (strcmp (method, "SHA512") == 0)
#endif                          /* USE_SHA_CRYPT */
                    ) {
                        pass_max_len = -1;
                } else {
                        pass_max_len = getdef_num ("PASS_MAX_LEN", 8);
                }
        }
        if (!qflg) {
                if (pass_max_len == -1) {
                        (void) printf (_(
"Enter the new password (minimum of %d characters)\n"
"Please use a combination of upper and lower case letters and numbers.\n"),
                                getdef_num ("PASS_MIN_LEN", 5));
                } else {
                        (void) printf (_(
"Enter the new password (minimum of %d, maximum of %d characters)\n"
"Please use a combination of upper and lower case letters and numbers.\n"),
                                getdef_num ("PASS_MIN_LEN", 5), pass_max_len);
                }
        }

        warned = false;
        for (i = getdef_num ("PASS_CHANGE_TRIES", 5); i > 0; i--) {
                cp = getpass (_("New password: "));
                if (NULL == cp) {
                        memzero (orig, sizeof orig);
                        return -1;
                }
                if (warned && (strcmp (pass, cp) != 0)) {
                        warned = false;
```

```
                }
                STRFCPY (pass, cp);
                strzero (cp);

                if (!amroot && (!obscure (orig, pass, pw) || reuse (pass, pw))) {
                        (void) puts (_("Try again."));
                        continue;
                }

                /*
                 * If enabled, warn about weak passwords even if you are
                 * root (enter this password again to use it anyway).
                 * --marekm
                 */
                if (amroot && !warned && getdef_bool ("PASS_ALWAYS_WARN")
                    && (!obscure (orig, pass, pw) || reuse (pass, pw))) {
                        (void) puts (_("\nWarning: weak password (enter it again to use it anyway)."));
                        warned = true;
                        continue;
                }
                cp = getpass (_("Re-enter new password: "));
                if (NULL == cp) {
                        memzero (orig, sizeof orig);
                        return -1;
                }
                if (strcmp (cp, pass) != 0) {
                        (void) fputs (_("They don't match; try again.\n"), stderr);
                } else {
                        strzero (cp);
                        break;
                }
        }
        memzero (orig, sizeof orig);

        if (i == 0) {
                memzero (pass, sizeof pass);
                return -1;
        }

        /*
         * Encrypt the password, then wipe the cleartext password.
         */
        salt = crypt_make_salt (NULL, NULL);
        cp = pw_encrypt (pass, salt);
        memzero (pass, sizeof pass);

        if (NULL == cp) {
                fprintf (stderr,
                         _("%s: failed to crypt password with salt '%s': %s\n"),
                         Prog, salt, strerror (errno));
                return -1;
        }

#ifdef HAVE_LIBCRACK_HIST
        HistUpdate (pw->pw_name, crypt_passwd);
#endif                          /* HAVE_LIBCRACK_HIST */
        STRFCPY (crypt_passwd, cp);
        return 0;
}

/*
 * check_password - test a password to see if it can be changed
 *
 *      check_password() sees if the invoker has permission to change the
 *      password for the given user.
 */
static void check_password (const struct passwd *pw, const struct spwd *sp)
{
        time_t now;
        int exp_status;

        exp_status = isexpired (pw, sp);

        /*
         * If not expired and the "change only if expired" option (idea from
         * PAM) was specified, do nothing. --marekm
         */
        if (kflg && (0 == exp_status)) {
                exit (E_SUCCESS);
        }

        /*
         * Root can change any password any time.
         */
        if (amroot) {
                return;
        }

        (void) time (&now);

        /*
         * Expired accounts cannot be changed ever. Passwords which are
         * locked may not be changed. Passwords where min > max may not be
         * changed. Passwords which have been inactive too long cannot be
         * changed.
         */
        if (   (sp->sp_pwdp[0] == '!')
            || (exp_status > 1)
```

```
            || (   (sp->sp_max >= 0)
                && (sp->sp_min > sp->sp_max))) {
                (void) fprintf (stderr,
                                _("The password for %s cannot be changed.\n"),
                                sp->sp_namp);
                SYSLOG ((LOG_WARN, "password locked for '%s'", sp->sp_namp));
                closelog ();
                exit (E_NOPERM);
        }

        /*
         * Passwords may only be changed after sp_min time is up.
         */
        if (sp->sp_lstchg > 0) {
                time_t ok;
                ok = (time_t) sp->sp_lstchg * SCALE;
                if (sp->sp_min > 0) {
                        ok += (time_t) sp->sp_min * SCALE;
                }

                if (now < ok) {
                        (void) fprintf (stderr,
                                        _("The password for %s cannot be changed yet.\n"),
                                        pw->pw_name);
                        SYSLOG ((LOG_WARN, "now < minimum age for '%s'", pw->pw_name));
                        closelog ();
                        exit (E_NOPERM);
                }
        }
}
#endif                          /* !USE_PAM */

static /*@observer@*/const char *date_to_str (time_t t)
{
        static char buf[80];
        struct tm *tm;

        tm = gmtime (&t);
#ifdef HAVE_STRFTIME
        (void) strftime (buf, sizeof buf, "%m/%d/%Y", tm);
#else                           /* !HAVE_STRFTIME */
        (void) snprintf (buf, sizeof buf, "%02d/%02d/%04d",
                         tm->tm_mon + 1, tm->tm_mday, tm->tm_year + 1900);
#endif                          /* !HAVE_STRFTIME */
        return buf;
}

static /*@observer@*/const char *pw_status (const char *pass)
{
        if (*pass == '*' || *pass == '!') {
                return "L";
        }
        if (*pass == '\0') {
                return "NP";
        }
        return "P";
}

/*
 * print_status - print current password status
 */
static void print_status (const struct passwd *pw)
{
        struct spwd *sp;

        sp = getspnam (pw->pw_name); /* local, no need for xgetspnam */
        if (NULL != sp) {
                (void) printf ("%s %s %s %lld %lld %lld %lld\n",
                               pw->pw_name,
                               pw_status (sp->sp_pwdp),
                               date_to_str (sp->sp_lstchg * SCALE),
                               ((long long)sp->sp_min * SCALE) / DAY,
                               ((long long)sp->sp_max * SCALE) / DAY,
                               ((long long)sp->sp_warn * SCALE) / DAY,
                               ((long long)sp->sp_inact * SCALE) / DAY);
        } else {
                (void) printf ("%s %s\n",
                               pw->pw_name, pw_status (pw->pw_passwd));
        }
}


static /*@noreturn@*/void fail_exit (int status)
{
        if (pw_locked) {
                if (pw_unlock () == 0) {
                        (void) fprintf (stderr, _("%s: failed to unlock %s\n"), Prog, pw_dbname ());
                        SYSLOG ((LOG_ERR, "failed to unlock %s", pw_dbname ()));
                        /* continue */
                }
        }

        if (spw_locked) {
                if (spw_unlock () == 0) {
                        (void) fprintf (stderr, _("%s: failed to unlock %s\n"), Prog, spw_dbname ());
                        SYSLOG ((LOG_ERR, "failed to unlock %s", spw_dbname ()));
                        /* continue */
                }
```

```
        }

        exit (status);
}

static /*@noreturn@*/void oom (void)
{
        (void) fprintf (stderr, _("%s: out of memory\n"), Prog);
        fail_exit (E_FAILURE);
}

static char *update_crypt_pw (char *cp)
{
#ifndef USE_PAM
        if (do_update_pwd) {
                cp = xstrdup (crypt_passwd);
        }
#endif                          /* !USE_PAM */

        if (dflg) {
                *cp = '\0';
        }

        if (uflg && *cp == '!') {
                if (cp[1] == '\0') {
                        (void) fprintf (stderr,
                                        _("%s: unlocking the password would result in a passwordless account.\n"
                                          "You should set a password with usermod -p to unlock the password of this account.\n"),
                                        Prog);
                        fail_exit (E_FAILURE);
                } else {
                        cp++;
                }
        }

        if (lflg && *cp != '!') {
                char *newpw = xmalloc (strlen (cp) + 2);

                strcpy (newpw, "!");
                strcat (newpw, cp);
                cp = newpw;
        }
        return cp;
}


static void update_noshadow (void)
{
        const struct passwd *pw;
        struct passwd *npw;

        if (pw_lock () == 0) {
                (void) fprintf (stderr,
                                _("%s: cannot lock %s; try again later.\n"),
                                Prog, pw_dbname ());
                exit (E_PWDBUSY);
        }
        pw_locked = true;
        if (pw_open (O_RDWR) == 0) {
                (void) fprintf (stderr,
                                _("%s: cannot open %s\n"),
                                Prog, pw_dbname ());
                SYSLOG ((LOG_WARN, "cannot open %s", pw_dbname ()));
                fail_exit (E_MISSING);
        }
        pw = pw_locate (name);
        if (NULL == pw) {
                (void) fprintf (stderr,
                                _("%s: user '%s' does not exist in %s\n"),
                                Prog, name, pw_dbname ());
                fail_exit (E_NOPERM);
        }
        npw = __pw_dup (pw);
        if (NULL == npw) {
                oom ();
        }
        npw->pw_passwd = update_crypt_pw (npw->pw_passwd);
        if (pw_update (npw) == 0) {
                (void) fprintf (stderr,
                                _("%s: failed to prepare the new %s entry '%s'\n"),
                                Prog, pw_dbname (), npw->pw_name);
                fail_exit (E_FAILURE);
        }
        if (pw_close () == 0) {
                (void) fprintf (stderr,
                                _("%s: failure while writing changes to %s\n"),
                                Prog, pw_dbname ());
                SYSLOG ((LOG_ERR, "failure while writing changes to %s", pw_dbname ()));
                fail_exit (E_FAILURE);
        }
        if (pw_unlock () == 0) {
                (void) fprintf (stderr,
                                _("%s: failed to unlock %s\n"),
                                Prog, pw_dbname ());
                SYSLOG ((LOG_ERR, "failed to unlock %s", pw_dbname ()));
                /* continue */
        }
        pw_locked = false;
```

```
}

static void update_shadow (void)
{
        const struct spwd *sp;
        struct spwd *nsp;

        if (spw_lock () == 0) {
                (void) fprintf (stderr,
                                _("%s: cannot lock %s; try again later.\n"),
                                Prog, spw_dbname ());
                exit (E_PWDBUSY);
        }
        spw_locked = true;
        if (spw_open (O_RDWR) == 0) {
                (void) fprintf (stderr,
                                _("%s: cannot open %s\n"),
                                Prog, spw_dbname ());
                SYSLOG ((LOG_WARN, "cannot open %s", spw_dbname ()));
                fail_exit (E_FAILURE);
        }
        sp = spw_locate (name);
        if (NULL == sp) {
                /* Try to update the password in /etc/passwd instead. */
                (void) spw_close ();
                update_noshadow ();
                if (spw_unlock () == 0) {
                        (void) fprintf (stderr,
                                        _("%s: failed to unlock %s\n"),
                                        Prog, spw_dbname ());
                        SYSLOG ((LOG_ERR, "failed to unlock %s", spw_dbname ()));
                        /* continue */
                }
                spw_locked = false;
                return;
        }
        nsp = __spw_dup (sp);
        if (NULL == nsp) {
                oom ();
        }
        nsp->sp_pwdp = update_crypt_pw (nsp->sp_pwdp);
        if (xflg) {
                nsp->sp_max = (age_max * DAY) / SCALE;
        }
        if (nflg) {
                nsp->sp_min = (age_min * DAY) / SCALE;
        }
        if (wflg) {
                nsp->sp_warn = (warn * DAY) / SCALE;
        }
        if (iflg) {
                nsp->sp_inact = (inact * DAY) / SCALE;
        }
#ifndef USE_PAM
        if (do_update_age) {
                nsp->sp_lstchg = (long) time ((time_t *) 0) / SCALE;
                if (0 == nsp->sp_lstchg) {
                        /* Better disable aging than requiring a password
                         * change */
                        nsp->sp_lstchg = -1;
                }
        }
#endif                          /* !USE_PAM */

        /*
         * Force change on next login, like SunOS 4.x passwd -e or Solaris
         * 2.x passwd -f. Solaris 2.x seems to do the same thing (set
         * sp_lstchg to 0).
         */
        if (eflg) {
                nsp->sp_lstchg = 0;
        }

        if (spw_update (nsp) == 0) {
                (void) fprintf (stderr,
                                _("%s: failed to prepare the new %s entry '%s'\n"),
                                Prog, spw_dbname (), nsp->sp_namp);
                fail_exit (E_FAILURE);
        }
        if (spw_close () == 0) {
                (void) fprintf (stderr,
                                _("%s: failure while writing changes to %s\n"),
                                Prog, spw_dbname ());
                SYSLOG ((LOG_ERR, "failure while writing changes to %s", spw_dbname ()));
                fail_exit (E_FAILURE);
        }
        if (spw_unlock () == 0) {
                (void) fprintf (stderr,
                                _("%s: failed to unlock %s\n"),
                                Prog, spw_dbname ());
                SYSLOG ((LOG_ERR, "failed to unlock %s", spw_dbname ()));
                /* continue */
        }
        spw_locked = false;
}

#ifdef WITH_SELINUX
static int check_selinux_access (const char *changed_user,
```

```
                                uid_t changed_uid,
                                access_vector_t requested_access)
{
        int status = -1;
        security_context_t user_context;
        context_t c;
        const char *user;

        /* if in permissive mode then allow the operation */
        if (security_getenforce() == 0) {
                return 0;
        }

        /* get the context of the process which executed passwd */
        if (getprevcon(&user_context) != 0) {
                return -1;
        }

        /* get the "user" portion of the context (the part before the first
           colon) */
        c = context_new(user_context);
        user = context_user_get(c);

        /* if changing a password for an account with UID==0 or for an account
           where the identity matches then return success */
        if (changed_uid != 0 && strcmp(changed_user, user) == 0) {
                status = 0;
        } else {
                struct av_decision avd;
                int retval;
                retval = security_compute_av(user_context,
                                             user_context,
                                             SECCLASS_PASSWD,
                                             requested_access,
                                             &avd);
                if ((retval == 0) &&
                    ((requested_access & avd.allowed) == requested_access)) {
                        status = 0;
                }
        }
        context_free(c);
        freecon(user_context);
        return status;
}

#endif                          /* WITH_SELINUX */

/*
 * passwd - change a user's password file information
 *
 *      This command controls the password file and commands which are used
 *      to modify it.
 *
 *      The valid options are
 *
 *      -d      delete the password for the named account (*)
 *      -e      expire the password for the named account (*)
 *      -f      execute chfn command to interpret flags
 *      -g      execute gpasswd command to interpret flags
 *      -i #    set sp_inact to # days (*)
 *      -k      change password only if expired
 *      -l      lock the password of the named account (*)
 *      -n #    set sp_min to # days (*)
 *      -r #    change password in # repository
 *      -s      execute chsh command to interpret flags
 *      -S      show password status of named account
 *      -u      unlock the password of the named account (*)
 *      -w #    set sp_warn to # days (*)
 *      -x #    set sp_max to # days (*)
 *
 *      (*) requires root permission to execute.
 *
 *      All of the time fields are entered in days and converted to the
 *      appropriate internal format. For finer resolute the chage
 *      command must be used.
 */
int main (int argc, char **argv)
{
        const struct passwd *pw;        /* Password file entry for user      */

#ifndef USE_PAM
        char *cp;               /* Miscellaneous character pointing  */

        const struct spwd *sp;  /* Shadow file entry for user   */
#endif                          /* !USE_PAM */

        sanitize_env ();

        /*
         * Get the program name. The program name is used as a prefix to
         * most error messages.
         */
        Prog = Basename (argv[0]);

        (void) setlocale (LC_ALL, "");
        (void) bindtextdomain (PACKAGE, LOCALEDIR);
        (void) textdomain (PACKAGE);
```

```
process_root_flag ("-R", argc, argv);

/*
 * The program behaves differently when executed by root than when
 * executed by a normal user.
 */
amroot = (getuid () == 0);

OPENLOG ("passwd");

{
        /*
         * Parse the command line options.
         */
        int c;
        static struct option long_options[] = {
                {"all",         no_argument,       NULL, 'a'},
                {"delete",      no_argument,       NULL, 'd'},
                {"expire",      no_argument,       NULL, 'e'},
                {"help",        no_argument,       NULL, 'h'},
                {"inactive",    required_argument, NULL, 'i'},
                {"keep-tokens", no_argument,       NULL, 'k'},
                {"lock",        no_argument,       NULL, 'l'},
                {"mindays",     required_argument, NULL, 'n'},
                {"quiet",       no_argument,       NULL, 'q'},
                {"repository",  required_argument, NULL, 'r'},
                {"root",        required_argument, NULL, 'R'},
                {"status",      no_argument,       NULL, 'S'},
                {"unlock",      no_argument,       NULL, 'u'},
                {"warndays",    required_argument, NULL, 'w'},
                {"maxdays",     required_argument, NULL, 'x'},
                {NULL, 0, NULL, '\0'}
        };

        while ((c = getopt_long (argc, argv, "adehi:kln:qr:R:Suw:x:",
                                 long_options, NULL)) != -1) {
                switch (c) {
                case 'a':
                        aflg = true;
                        break;
                case 'd':
                        dflg = true;
                        anyflag = true;
                        break;
                case 'e':
                        eflg = true;
                        anyflag = true;
                        break;
                case 'h':
                        usage (E_SUCCESS);
                        /*@notreached@*/break;
                case 'i':
                        if (   (getlong (optarg, &inact) == 0)
                            || (inact < -1)) {
                                fprintf (stderr,
                                         _("%s: invalid numeric argument '%s'\n"),
                                         Prog, optarg);
                                usage (E_BAD_ARG);
                        }
                        iflg = true;
                        anyflag = true;
                        break;
                case 'k':
                        /* change only if expired, like Linux-PAM passwd -k. */
                        kflg = true;    /* ok for users */
                        break;
                case 'l':
                        lflg = true;
                        anyflag = true;
                        break;
                case 'n':
                        if (   (getlong (optarg, &age_min) == 0)
                            || (age_min < -1)) {
                                fprintf (stderr,
                                         _("%s: invalid numeric argument '%s'\n"),
                                         Prog, optarg);
                                usage (E_BAD_ARG);
                        }
                        nflg = true;
                        anyflag = true;
                        break;
                case 'q':
                        qflg = true;    /* ok for users */
                        break;
                case 'r':
                        /* -r repository (files|nis|nisplus) */
                        /* only "files" supported for now */
                        if (strcmp (optarg, "files") != 0) {
                                fprintf (stderr,
                                         _("%s: repository %s not supported\n"),
                                         Prog, optarg);
                                exit (E_BAD_ARG);
                        }
                        break;
                case 'R': /* no-op, handled in process_root_flag () */
                        break;
                case 'S':
                        Sflg = true;    /* ok for users */
```

```
                        break;
                case 'u':
                        uflg = true;
                        anyflag = true;
                        break;
                case 'w':
                        if (   (getlong (optarg, &warn) == 0)
                            || (warn < -1)) {
                                (void) fprintf (stderr,
                                                _("%s: invalid numeric argument '%s'\n"),
                                                Prog, optarg);
                                usage (E_BAD_ARG);
                        }
                        wflg = true;
                        anyflag = true;
                        break;
                case 'x':
                        if (   (getlong (optarg, &age_max) == 0)
                            || (age_max < -1)) {
                                (void) fprintf (stderr,
                                                _("%s: invalid numeric argument '%s'\n"),
                                                Prog, optarg);
                                usage (E_BAD_ARG);
                        }
                        xflg = true;
                        anyflag = true;
                        break;
                default:
                        usage (E_BAD_ARG);
                }
        }
    }

    /*
     * Now I have to get the user name. The name will be gotten from the
     * command line if possible. Otherwise it is figured out from the
     * environment.
     */
    pw = get_my_pwent ();
    if (NULL == pw) {
            (void) fprintf (stderr,
                            _("%s: Cannot determine your user name.\n"),
                            Prog);
            SYSLOG ((LOG_WARN, "Cannot determine the user name of the caller (UID %lu)",
                     (unsigned long) getuid ()));
            exit (E_NOPERM);
    }
    myname = xstrdup (pw->pw_name);
    if (optind < argc) {
            name = argv[optind];
    } else {
            name = myname;
    }

    /*
     * Make sure that at most one username was specified.
     */
    if (argc > (optind+1)) {
            usage (E_USAGE);
    }

    /*
     * The -a flag requires -S, no other flags, no username, and
     * you must be root.  --marekm
     */
    if (aflg) {
            if (anyflag || !Sflg || (optind < argc)) {
                    usage (E_USAGE);
            }
            if (!amroot) {
                    (void) fprintf (stderr,
                                    _("%s: Permission denied.\n"),
                                    Prog);
                    exit (E_NOPERM);
            }
            setpwent ();
            while ( (pw = getpwent ()) != NULL ) {
                    print_status (pw);
            }
            endpwent ();
            exit (E_SUCCESS);
    }
#if 0
    /*
     * Allow certain users (administrators) to change passwords of
     * certain users. Not implemented yet. --marekm
     */
    if (may_change_passwd (myname, name))
            amroot = 1;
#endif

    /*
     * If any of the flags were given, a user name must be supplied on
     * the command line. Only an unadorned command line doesn't require
     * the user's name be given. Also, -x, -n, -w, -i, -e, -d,
     * -l, -u may appear with each other. -S, -k must appear alone.
     */
```

```
        /*
         * -S now ok for normal users (check status of my own account), and
         * doesn't require username.  --marekm
         */
        if (anyflag && optind >= argc) {
                usage (E_USAGE);
        }

        if (   (Sflg && kflg)
            || (anyflag && (Sflg || kflg))) {
                usage (E_USAGE);
        }

        if (anyflag && !amroot) {
                (void) fprintf (stderr, _("%s: Permission denied.\n"), Prog);
                exit (E_NOPERM);
        }

        pw = xgetpwnam (name);
        if (NULL == pw) {
                (void) fprintf (stderr,
                                _("%s: user '%s' does not exist\n"),
                                Prog, name);
                exit (E_NOPERM);
        }
#ifdef WITH_SELINUX
        /* only do this check when getuid()==0 because it's a pre-condition for
           changing a password without entering the old one */
        if ((is_selinux_enabled() > 0) && (getuid() == 0) &&
            (check_selinux_access (name, pw->pw_uid, PASSWD__PASSWD) != 0)) {
                security_context_t user_context = NULL;
                const char *user = "Unknown user context";
                if (getprevcon (&user_context) == 0) {
                        user = user_context; /* FIXME: use context_user_get? */
                }
                SYSLOG ((LOG_ALERT,
                         "%s is not authorized to change the password of %s",
                         user, name));
                (void) fprintf(stderr,
                               _("%s: %s is not authorized to change the password of %s\n"),
                               Prog, user, name);
                if (NULL != user_context) {
                        freecon (user_context);
                }
                exit (E_NOPERM);
        }
#endif                          /* WITH_SELINUX */

        /*
         * If the UID of the user does not match the current real UID,
         * check if I'm root.
         */
        if (!amroot && (pw->pw_uid != getuid ())) {
                (void) fprintf (stderr,
                                _("%s: You may not view or modify password information for %s.\n"),
                                Prog, name);
                SYSLOG ((LOG_WARN,
                         "%s: can't view or modify password information for %s",
                         Prog, name));
                closelog ();
                exit (E_NOPERM);
        }

        if (Sflg) {
                print_status (pw);
                exit (E_SUCCESS);
        }
#ifndef USE_PAM
        /*
         * The user name is valid, so let's get the shadow file entry.
         */
        sp = getspnam (name); /* !USE_PAM, no need for xgetspnam */
        if (NULL == sp) {
                if (errno == EACCES) {
                        (void) fprintf (stderr,
                                        _("%s: Permission denied.\n"),
                                        Prog);
                        exit (E_NOPERM);
                }
                sp = pwd_to_spwd (pw);
        }

        cp = sp->sp_pwdp;

        /*
         * If there are no other flags, just change the password.
         */
        if (!anyflag) {
                STRFCPY (crypt_passwd, cp);

                /*
                 * See if the user is permitted to change the password.
                 * Otherwise, go ahead and set a new password.
                 */
                check_password (pw, sp);

                /*
                 * Let the user know whose password is being changed.
```

```
                 */
                if (!qflg) {
                        (void) printf (_("Changing password for %s\n"), name);
                }

                if (new_password (pw) != 0) {
                        (void) fprintf (stderr,
                                        _("The password for %s is unchanged.\n"),
                                        name);
                        closelog ();
                        exit (E_NOPERM);
                }
                do_update_pwd = true;
                do_update_age = true;
        }
#endif                          /* !USE_PAM */
        /*
         * Before going any further, raise the ulimit to prevent colliding
         * into a lowered ulimit, and set the real UID to root to protect
         * against unexpected signals. Any keyboard signals are set to be
         * ignored.
         */
        pwd_init ();

#ifdef USE_PAM
        /*
         * Don't set the real UID for PAM...
         */
        if (!anyflag) {
                do_pam_passwd (name, qflg, kflg);
                exit (E_SUCCESS);
        }
#endif                          /* USE_PAM */
        if (setuid (0) != 0) {
                (void) fputs (_("Cannot change ID to root.\n"), stderr);
                SYSLOG ((LOG_ERR, "can't setuid(0)"));
                closelog ();
                exit (E_NOPERM);
        }
        if (spw_file_present ()) {
                update_shadow ();
        } else {
                update_noshadow ();
        }

        nscd_flush_cache ("passwd");
        nscd_flush_cache ("group");

        SYSLOG ((LOG_INFO, "password for '%s' changed by '%s'", name, myname));
        closelog ();
        if (!qflg) {
                if (!anyflag) {
#ifndef USE_PAM
                        (void) printf (_("%s: password changed.\n"), Prog);
#endif                          /* USE_PAM */
                } else {
                        (void) printf (_("%s: password expiry information changed.\n"), Prog);
                }
        }

        return E_SUCCESS;
}
```

Notice for package(s)

e2fsprogs
flex
glib-2.0
hdparm
iputils
libcap
libevent
libnfsidmap
libpcap
libtirpc
nfs-utils
openssh
quota
rpcbind
shadow
tcpdump
util-linux

```
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

Notice for package(s)

perl
shadow

```
The Artistic License
Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semb

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through t
"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright H
"Copyright Holder" is whoever is named in the copyright or copyrights for the package.
"You" is you, if you`re thinking about copying or distributing this Package.
"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You
"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means tha
1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that yo

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package m

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or a

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a sepa

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-stand

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written p

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF ME

The End
```

Notice for package(s)

openssh

```
This file is part of the OpenSSH software.
```

The licences which components of this software fall under are as
follows. First, we will summarize and say that all components
are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)
```
 * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
 *                    All rights reserved
 *
 * As far as I am concerned, the code I have written for this software
 * can be used freely for any purpose.  Any derived versions of this
 * software must be clearly marked as such, and if the derived work is
 * incompatible with the protocol description in the RFC file, it must be
 * called by a name other than "ssh" or "Secure Shell".
```

[Tatu continues]
```
 *  However, I am not implying to give any licenses to any patents or
 * copyrights held by third parties, and the software includes parts that
 * are not under my direct control.  As far as I know, all included
 * source code is used in accordance with the relevant license agreements
 * and can be used freely for any purpose (the GNU license being the most
 * restrictive); see below for details.
```

[However, none of that term is relevant at this point in time. All of
these restrictively licenced software components which he talks about
have been removed from OpenSSH, i.e.,

- RSA is no longer included, found in the OpenSSL library
- IDEA is no longer included, its use is deprecated
- DES is now external, in the OpenSSL library
- GMP is no longer used, and instead we call BN code from OpenSSL
- Zlib is now external, in a library
- The make-ssh-known-hosts script is no longer included
- TSS has been removed
- MD5 is now external, in the OpenSSL library
- RC4 support has been replaced with ARC4 support from OpenSSL
- Blowfish is now external, in the OpenSSL library

[The licence continues]

Note that any information and cryptographic algorithms used in this
software are publicly available on the Internet and at any major
bookstore, scientific library, and patent office worldwide. More
information can be found e.g. at "http://www.cs.hut.fi/crypto".

The legal status of this program is some combination of all these
permissions and restrictions. Use only at your own responsibility.
You will be responsible for any legal consequences yourself; I am not
making any claims whether possessing or using this is legal or not in
your country, and I am not taking any responsibility on your behalf.

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

2)
The 32-bit CRC compensation attack detector in deattack.c was
contributed by CORE SDI S.A. under a BSD-style license.

```
 * Cryptographic attack detector for ssh - source code
 *
 * Copyright (c) 1998 CORE SDI S.A., Buenos Aires, Argentina.
 *
 * All rights reserved. Redistribution and use in source and binary
 * forms, with or without modification, are permitted provided that
 * this copyright notice is retained.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
 * WARRANTIES ARE DISCLAIMED. IN NO EVENT SHALL CORE SDI S.A. BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR
 * CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OR MISUSE OF THIS
 * SOFTWARE.
 *
 * Ariel Futoransky <futo@core-sdi.com>
 * <http://www.core-sdi.com>
```

3)
ssh-keyscan was contributed by David Mazieres under a BSD-style

license.

```
 * Copyright 1995, 1996 by David Mazieres <dm@lcs.mit.edu>.
 *
 * Modification and redistribution in source and binary forms is
 * permitted provided that due credit is given to the author and the
 * OpenBSD project by leaving this copyright notice intact.
```

4)
The Rijndael implementation by Vincent Rijmen, Antoon Bosselaers
and Paulo Barreto is in the public domain and distributed
with the following license:

```
 * @version 3.0 (December 2000)
 *
 * Optimised ANSI C code for the Rijndael cipher (now AES)
 *
 * @author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be>
 * @author Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be>
 * @author Paulo Barreto <paulo.barreto@terra.com.br>
 *
 * This code is hereby placed in the public domain.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHORS ''AS IS'' AND ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

5)
One component of the ssh source code is under a 3-clause BSD license,
held by the University of California, since we pulled these parts from
original Berkeley code.

```
 * Copyright (c) 1983, 1990, 1992, 1993, 1995
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
```

6)
Remaining components of the software are provided under a standard
2-term BSD licence with the following names as copyright holders:

```
    Markus Friedl
    Theo de Raadt
    Niels Provos
    Dug Song
    Aaron Campbell
    Damien Miller
    Kevin Steves
    Daniel Kouril
    Wesley Griffin
    Per Allansson
    Nils Nordman
    Simon Wilkinson
```

Portable OpenSSH additionally includes code from the following copyright
holders, also under the 2-term BSD license:

```
    Ben Lindstrom
    Tim Rice
    Andre Lucas
    Chris Adams
    Corinna Vinschen
    Cray Inc.
    Denis Parker
    Gert Doering
    Jakob Schlyter
    Jason Downs
    Juha Yrj�l�
```

Michael Stone
Networks Associates Technology, Inc.
Solar Designer
Todd C. Miller
Wayne Schroeder
William Jones
Darren Tucker
Sun Microsystems
The SCO Group
Daniel Walsh
Red Hat, Inc
Simon Vallet / Genoscope

```
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

8) Portable OpenSSH contains the following additional licenses:

a) md5crypt.c, md5crypt.h

```
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <phk@login.dknet.dk> wrote this file.  As long as you retain this
 * notice you can do whatever you want with this stuff. If we meet
 * some day, and you think this stuff is worth it, you can buy me a
 * beer in return.   Poul-Henning Kamp
```

b) snprintf replacement

```
 * Copyright Patrick Powell 1995
 * This code is based on code written by Patrick Powell
 * (papowell@astart.com) It may be used for any purpose as long as this
 * notice remains intact on all source code distributions
```

c) Compatibility code (openbsd-compat)

Apart from the previously mentioned licenses, various pieces of code
in the openbsd-compat/ subdirectory are licensed as follows:

Some code is licensed under a 3-term BSD license, to the following
copyright holders:

Todd C. Miller
Theo de Raadt
Damien Miller
Eric P. Allman
The Regents of the University of California
Constantin S. Svintsoff

```
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
```

Some code is licensed under an ISC-style license, to the following
copyright holders:

Internet Software Consortium.
Todd C. Miller
Reyk Floeter
Chad Mynhier

```
 * Permission to use, copy, modify, and distribute this software for any
```

```
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER DISCLAIMS ALL
 * WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE
 * FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
 * OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
 * CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Some code is licensed under a MIT-style license to the following
copyright holders:

 Free Software Foundation, Inc.

 * Permission is hereby granted, free of charge, to any person obtaining a  *
 * copy of this software and associated documentation files (the            *
 * "Software"), to deal in the Software without restriction, including      *
 * without limitation the rights to use, copy, modify, merge, publish,      *
 * distribute, distribute with modifications, sublicense, and/or sell       *
 * copies of the Software, and to permit persons to whom the Software is    *
 * furnished to do so, subject to the following conditions:                 *
 *                                                                          *
 * The above copyright notice and this permission notice shall be included  *
 * in all copies or substantial portions of the Software.                   *
 *                                                                          *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  *
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF               *
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.   *
 * IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,   *
 * DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR    *
 * OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR    *
 * THE USE OR OTHER DEALINGS IN THE SOFTWARE.                               *
 *                                                                          *
 * Except as contained in this notice, the name(s) of the above copyright   *
 * holders shall not be used in advertising or otherwise to promote the     *
 * sale, use or other dealings in this Software without prior written       *
 * authorization.                                                           *
 ****************************************************************************/


------
$OpenBSD: LICENCE,v 1.19 2004/08/30 09:18:08 markus Exp $
```

Notice for package(s)

kexec-tools

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.
```

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections

1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions

of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

  10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

```
This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

kexec-tools

```
/*
 * kexec: Linux boots Linux
 *
 * Copyright (C) 2003-2005  Eric Biederman (ebiederm@xmission.com)
 *
 * Modified (2007-05-15) by Francesco Chiechi to rudely handle mips platform
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation (version 2 of the License).
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#define _GNU_SOURCE
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <limits.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/reboot.h>
#include <unistd.h>
#include <fcntl.h>
#ifndef _O_BINARY
#define _O_BINARY 0
#endif
#include <getopt.h>
#include <ctype.h>

#include "config.h"

#include <sha256.h>
#include "kexec.h"
#include "kexec-syscall.h"
#include "kexec-elf.h"
#include "kexec-sha256.h"
#include "kexec-zlib.h"
#include "kexec-lzma.h"
#include <arch/options.h>

unsigned long long mem_min = 0;
unsigned long long mem_max = ULONG_MAX;
static unsigned long kexec_flags = 0;
/* Flags for kexec file (fd) based syscall */
static unsigned long kexec_file_flags = 0;
int kexec_debug = 0;

void dbgprint_mem_range(const char *prefix, struct memory_range *mr, int nr_mr)
{
        int i;
        dbgprintf("%s\n", prefix);
        for (i = 0; i < nr_mr; i++) {
                dbgprintf("%016llx-%016llx (%d)\n", mr[i].start,
                          mr[i].end, mr[i].type);
        }
}

void die(const char *fmt, ...)
{
        va_list args;
        va_start(args, fmt);
        vfprintf(stderr, fmt, args);
        va_end(args);
        fflush(stdout);
        fflush(stderr);
        exit(1);
}

static char *xstrdup(const char *str)
{
        char *new = strdup(str);
```

```
        if (!new)
                die("Cannot strdup \"%s\": %s\n",
                        str, strerror(errno));
        return new;
}

void *xmalloc(size_t size)
{
        void *buf;
        if (!size)
                return NULL;
        buf = malloc(size);
        if (!buf) {
                die("Cannot malloc %ld bytes: %s\n",
                        size + 0UL, strerror(errno));
        }
        return buf;
}

void *xrealloc(void *ptr, size_t size)
{
        void *buf;
        buf = realloc(ptr, size);
        if (!buf) {
                die("Cannot realloc %ld bytes: %s\n",
                        size + 0UL, strerror(errno));
        }
        return buf;
}

int valid_memory_range(struct kexec_info *info,
                       unsigned long sstart, unsigned long send)
{
        int i;
        if (sstart > send) {
                return 0;
        }
        if ((send > mem_max) || (sstart < mem_min)) {
                return 0;
        }
        for (i = 0; i < info->memory_ranges; i++) {
                unsigned long mstart, mend;
                /* Only consider memory ranges */
                if (info->memory_range[i].type != RANGE_RAM)
                        continue;
                mstart = info->memory_range[i].start;
                mend = info->memory_range[i].end;
                if (i < info->memory_ranges - 1
                    && mend == info->memory_range[i+1].start
                    && info->memory_range[i+1].type == RANGE_RAM)
                        mend = info->memory_range[i+1].end;

                /* Check to see if we are fully contained */
                if ((mstart <= sstart) && (mend >= send)) {
                        return 1;
                }
        }
        return 0;
}

static int valid_memory_segment(struct kexec_info *info,
                                struct kexec_segment *segment)
{
        unsigned long sstart, send;
        sstart = (unsigned long)segment->mem;
        send   = sstart + segment->memsz - 1;

        return valid_memory_range(info, sstart, send);
}

void print_segments(FILE *f, struct kexec_info *info)
{
        int i;

        fprintf(f, "nr_segments = %d\n", info->nr_segments);
        for (i = 0; i < info->nr_segments; i++) {
                fprintf(f, "segment[%d].buf   = %p\n",  i,
                        info->segment[i].buf);
                fprintf(f, "segment[%d].bufsz = 0x%zx\n", i,
                        info->segment[i].bufsz);
                fprintf(f, "segment[%d].mem   = %p\n",  i,
                        info->segment[i].mem);
                fprintf(f, "segment[%d].memsz = 0x%zx\n", i,
                        info->segment[i].memsz);
        }
}

int sort_segments(struct kexec_info *info)
{
        int i, j;
        void *end;

        /* Do a stupid insertion sort... */
        for (i = 0; i < info->nr_segments; i++) {
                int tidx;
                struct kexec_segment temp;
                tidx = i;
                for (j = i +1; j < info->nr_segments; j++) {
```

```
                        if (info->segment[j].mem < info->segment[tidx].mem) {
                                tidx = j;
                        }
                }
                if (tidx != i) {
                        temp = info->segment[tidx];
                        info->segment[tidx] = info->segment[i];
                        info->segment[i] = temp;
                }
        }
        /* Now see if any of the segments overlap */
        end = 0;
        for (i = 0; i < info->nr_segments; i++) {
                if (end > info->segment[i].mem) {
                        fprintf(stderr, "Overlapping memory segments at %p\n",
                                end);
                        return -1;
                }
                end = ((char *)info->segment[i].mem) + info->segment[i].memsz;
        }
        return 0;
}

unsigned long locate_hole(struct kexec_info *info,
        unsigned long hole_size, unsigned long hole_align,
        unsigned long hole_min, unsigned long hole_max,
        int hole_end)
{
        int i, j;
        struct memory_range *mem_range;
        int max_mem_ranges, mem_ranges;
        unsigned long hole_base;

        if (hole_end == 0) {
                die("Invalid hole end argument of 0 specified to locate_hole");
        }

        /* Set an initial invalid value for the hole base */
        hole_base = ULONG_MAX;

        /* Align everything to at least a page size boundary */
        if (hole_align < (unsigned long)getpagesize()) {
                hole_align = getpagesize();
        }

        /* Compute the free memory ranges */
        max_mem_ranges = info->memory_ranges + info->nr_segments;
        mem_range = xmalloc(max_mem_ranges *sizeof(struct memory_range));
        mem_ranges = 0;

        /* Perform a merge on the 2 sorted lists of memory ranges  */
        for (j = 0, i = 0; i < info->memory_ranges; i++) {
                unsigned long long sstart, send;
                unsigned long long mstart, mend;
                mstart = info->memory_range[i].start;
                mend = info->memory_range[i].end;
                if (info->memory_range[i].type != RANGE_RAM)
                        continue;
                while ((j < info->nr_segments) &&
                       (((unsigned long)info->segment[j].mem) <= mend)) {
                        sstart = (unsigned long)info->segment[j].mem;
                        send = sstart + info->segment[j].memsz -1;
                        if (mstart < sstart) {
                                mem_range[mem_ranges].start = mstart;
                                mem_range[mem_ranges].end = sstart -1;
                                mem_range[mem_ranges].type = RANGE_RAM;
                                mem_ranges++;
                        }
                        mstart = send +1;
                        j++;
                }
                if (mstart < mend) {
                        mem_range[mem_ranges].start = mstart;
                        mem_range[mem_ranges].end = mend;
                        mem_range[mem_ranges].type = RANGE_RAM;
                        mem_ranges++;
                }
        }
        /* Now find the end of the last memory_range I can use */
        for (i = 0; i < mem_ranges; i++) {
                unsigned long long start, end, size;
                start = mem_range[i].start;
                end   = mem_range[i].end;
                /* First filter the range start and end values
                 * through the lens of mem_min, mem_max and hole_align.
                 */
                if (start < mem_min) {
                        start = mem_min;
                }
                if (start < hole_min) {
                        start = hole_min;
                }
                start = _ALIGN(start, hole_align);
                if (end > mem_max) {
                        end = mem_max;
                }
                if (end > hole_max) {
                        end = hole_max;
```

```
                }
                /* Is this still a valid memory range? */
                if ((start >= end) || (start >= mem_max) || (end <= mem_min)) {
                        continue;
                }
                /* Is there enough space left so we can use it? */
                size = end - start;
                if (!hole_size || size >= hole_size - 1) {
                        if (hole_end > 0) {
                                hole_base = start;
                                break;
                        } else {
                                hole_base = _ALIGN_DOWN(end - hole_size + 1,
                                        hole_align);
                        }
                }
        }
        free(mem_range);
        if (hole_base == ULONG_MAX) {
                fprintf(stderr, "Could not find a free area of memory of "
                        "0x%lx bytes...\n", hole_size);
                return ULONG_MAX;
        }
        if (hole_size && (hole_base + hole_size - 1)  > hole_max) {
                fprintf(stderr, "Could not find a free area of memory below: "
                        "0x%lx...\n", hole_max);
                return ULONG_MAX;
        }
        return hole_base;
}

void add_segment_phys_virt(struct kexec_info *info,
        const void *buf, size_t bufsz,
        unsigned long base, size_t memsz, int phys)
{
        unsigned long last;
        size_t size;
        int pagesize;

        if (bufsz > memsz) {
                bufsz = memsz;
        }
        /* Forget empty segments */
        if (memsz == 0) {
                return;
        }

        /* Round memsz up to a multiple of pagesize */
        pagesize = getpagesize();
        memsz = _ALIGN(memsz, pagesize);

        /* Verify base is pagesize aligned.
         * Finding a way to cope with this problem
         * is important but for now error so at least
         * we are not surprised by the code doing the wrong
         * thing.
         */
        if (base & (pagesize -1)) {
                die("Base address: 0x%lx is not page aligned\n", base);
        }

        if (phys)
                base = virt_to_phys(base);

        last = base + memsz -1;
        if (!valid_memory_range(info, base, last)) {
                die("Invalid memory segment %p - %p\n",
                        (void *)base, (void *)last);
        }

        size = (info->nr_segments + 1) * sizeof(info->segment[0]);
        info->segment = xrealloc(info->segment, size);
        info->segment[info->nr_segments].buf   = buf;
        info->segment[info->nr_segments].bufsz = bufsz;
        info->segment[info->nr_segments].mem   = (void *)base;
        info->segment[info->nr_segments].memsz = memsz;
        info->nr_segments++;
        if (info->nr_segments > KEXEC_MAX_SEGMENTS) {
                fprintf(stderr, "Warning: kernel segment limit reached. "
                        "This will likely fail\n");
        }
}

unsigned long add_buffer_phys_virt(struct kexec_info *info,
        const void *buf, unsigned long bufsz, unsigned long memsz,
        unsigned long buf_align, unsigned long buf_min, unsigned long buf_max,
        int buf_end, int phys)
{
        unsigned long base;
        int result;
        int pagesize;

        result = sort_segments(info);
        if (result < 0) {
                die("sort_segments failed\n");
        }

        /* Round memsz up to a multiple of pagesize */
```

```
        pagesize = getpagesize();
        memsz = _ALIGN(memsz, pagesize);

        base = locate_hole(info, memsz, buf_align, buf_min, buf_max, buf_end);
        if (base == ULONG_MAX) {
                die("locate_hole failed\n");
        }

        add_segment_phys_virt(info, buf, bufsz, base, memsz, phys);
        return base;
}

unsigned long add_buffer_virt(struct kexec_info *info, const void *buf,
                              unsigned long bufsz, unsigned long memsz,
                              unsigned long buf_align, unsigned long buf_min,
                              unsigned long buf_max, int buf_end)
{
        return add_buffer_phys_virt(info, buf, bufsz, memsz, buf_align,
                                    buf_min, buf_max, buf_end, 0);
}

static int find_memory_range(struct kexec_info *info,
                             unsigned long *base, unsigned long *size)
{
        int i;
        unsigned long start, end;

        for (i = 0; i < info->memory_ranges; i++) {
                if (info->memory_range[i].type != RANGE_RAM)
                        continue;
                start = info->memory_range[i].start;
                end = info->memory_range[i].end;
                if (end > *base && start < *base + *size) {
                        if (start > *base) {
                                *size = *base + *size - start;
                                *base = start;
                        }
                        if (end < *base + *size)
                                *size = end - *base;
                        return 1;
                }
        }
        return 0;
}

static int find_segment_hole(struct kexec_info *info,
                             unsigned long *base, unsigned long *size)
{
        int i;
        unsigned long seg_base, seg_size;

        for (i = 0; i < info->nr_segments; i++) {
                seg_base = (unsigned long)info->segment[i].mem;
                seg_size = info->segment[i].memsz;

                if (seg_base + seg_size <= *base)
                        continue;
                else if (seg_base >= *base + *size)
                        break;
                else if (*base < seg_base) {
                        *size = seg_base - *base;
                        break;
                } else if (seg_base + seg_size < *base + *size) {
                        *size = *base + *size - (seg_base + seg_size);
                        *base = seg_base + seg_size;
                } else {
                        *size = 0;
                        break;
                }
        }
        return *size;
}

static int add_backup_segments(struct kexec_info *info,
                               unsigned long backup_base,
                               unsigned long backup_size)
{
        unsigned long mem_base, mem_size, bkseg_base, bkseg_size, start, end;
        unsigned long pagesize;

        pagesize = getpagesize();
        while (backup_size) {
                mem_base = backup_base;
                mem_size = backup_size;
                if (!find_memory_range(info, &mem_base, &mem_size))
                        break;
                backup_size = backup_base + backup_size - \
                        (mem_base + mem_size);
                backup_base = mem_base + mem_size;
                while (mem_size) {
                        bkseg_base = mem_base;
                        bkseg_size = mem_size;
                        if (sort_segments(info) < 0)
                                return -1;
                        if (!find_segment_hole(info, &bkseg_base, &bkseg_size))
                                break;
                        start = _ALIGN(bkseg_base, pagesize);
                        end = _ALIGN_DOWN(bkseg_base + bkseg_size, pagesize);
```

```
                        add_segment_phys_virt(info, NULL, 0,
                                              start, end-start, 0);
                        mem_size = mem_base + mem_size - \
                                (bkseg_base + bkseg_size);
                        mem_base = bkseg_base + bkseg_size;
                }
        }
        return 0;
}

static char *slurp_fd(int fd, const char *filename, off_t size, off_t *nread)
{
        char *buf;
        off_t progress;
        ssize_t result;

        buf = xmalloc(size);
        progress = 0;
        while (progress < size) {
                result = read(fd, buf + progress, size - progress);
                if (result < 0) {
                        if ((errno == EINTR) || (errno == EAGAIN))
                                continue;
                        fprintf(stderr, "Read on %s failed: %s\n", filename,
                                strerror(errno));
                        free(buf);
                        close(fd);
                        return NULL;
                }
                if (result == 0)
                        /* EOF */
                        break;
                progress += result;
        }
        result = close(fd);
        if (result < 0)
                die("Close of %s failed: %s\n", filename, strerror(errno));

        if (nread)
                *nread = progress;
        return buf;
}

char *slurp_file(const char *filename, off_t *r_size)
{
        int fd;
        char *buf;
        off_t size, err, nread;
        ssize_t result;
        struct stat stats;

        if (!filename) {
                *r_size = 0;
                return 0;
        }
        fd = open(filename, O_RDONLY | _O_BINARY);
        if (fd < 0) {
                die("Cannot open `%s': %s\n",
                        filename, strerror(errno));
        }
        result = fstat(fd, &stats);
        if (result < 0) {
                die("Cannot stat: %s: %s\n",
                        filename, strerror(errno));
        }
        /*
         * Seek in case the kernel is a character node like /dev/ubi0_0.
         * This does not work on regular files which live in /proc and
         * we need this for some /proc/device-tree entries
         */
        if (S_ISCHR(stats.st_mode)) {

                size = lseek(fd, 0, SEEK_END);
                if (size < 0)
                        die("Can not seek file %s: %s\n", filename,
                                        strerror(errno));

                err = lseek(fd, 0, SEEK_SET);
                if (err < 0)
                        die("Can not seek to the begin of file %s: %s\n",
                                        filename, strerror(errno));
        } else {
                size = stats.st_size;
        }

        buf = slurp_fd(fd, filename, size, &nread);
        if (!buf)
                die("Cannot read %s", filename);

        if (nread != size)
                die("Read on %s ended before stat said it should\n", filename);

        *r_size = size;
        return buf;
}

/* This functions reads either specified number of bytes from the file or
   lesser if EOF is met. */
```

```
char *slurp_file_len(const char *filename, off_t size, off_t *nread)
{
        int fd;

        if (!filename)
                return 0;
        fd = open(filename, O_RDONLY | _O_BINARY);
        if (fd < 0) {
                fprintf(stderr, "Cannot open %s: %s\n", filename,
                                strerror(errno));
                return 0;
        }

        return slurp_fd(fd, filename, size, nread);
}

char *slurp_decompress_file(const char *filename, off_t *r_size)
{
        char *kernel_buf;

        kernel_buf = zlib_decompress_file(filename, r_size);
        if (!kernel_buf) {
                kernel_buf = lzma_decompress_file(filename, r_size);
                if (!kernel_buf)
                        return slurp_file(filename, r_size);
        }
        return kernel_buf;
}

static void update_purgatory(struct kexec_info *info)
{
        static const uint8_t null_buf[256];
        sha256_context ctx;
        sha256_digest_t digest;
        struct sha256_region region[SHA256_REGIONS];
        int i, j;
        /* Don't do anything if we are not using purgatory */
        if (!info->rhdr.e_shdr) {
                return;
        }
        arch_update_purgatory(info);
        memset(region, 0, sizeof(region));
        sha256_starts(&ctx);
        /* Compute a hash of the loaded kernel */
        for(j = i = 0; i < info->nr_segments; i++) {
                unsigned long nullsz;
                /* Don't include purgatory in the checksum.  The stack
                 * in the bss will definitely change, and the .data section
                 * will also change when we poke the sha256_digest in there.
                 * A very clever/careful person could probably improve this.
                 */
                if (info->segment[i].mem == (void *)info->rhdr.rel_addr) {
                        continue;
                }
                sha256_update(&ctx, info->segment[i].buf,
                              info->segment[i].bufsz);
                nullsz = info->segment[i].memsz - info->segment[i].bufsz;
                while(nullsz) {
                        unsigned long bytes = nullsz;
                        if (bytes > sizeof(null_buf)) {
                                bytes = sizeof(null_buf);
                        }
                        sha256_update(&ctx, null_buf, bytes);
                        nullsz -= bytes;
                }
                region[j].start = (unsigned long) info->segment[i].mem;
                region[j].len   = info->segment[i].memsz;
                j++;
        }
        sha256_finish(&ctx, digest);
        elf_rel_set_symbol(&info->rhdr, "sha256_regions", &region,
                           sizeof(region));
        elf_rel_set_symbol(&info->rhdr, "sha256_digest", &digest,
                           sizeof(digest));
}

/*
 *      Load the new kernel
 */
static int my_load(const char *type, int fileind, int argc, char **argv,
                   unsigned long kexec_flags, void *entry)
{
        char *kernel;
        char *kernel_buf;
        off_t kernel_size;
        int i = 0;
        int result;
        struct kexec_info info;
        long native_arch;
        int guess_only = 0;

        memset(&info, 0, sizeof(info));
        info.kexec_flags = kexec_flags;

        fprintf(stderr, "%s:%d: do\n", __func__, __LINE__);

        result = 0;
```

```
if (argc - fileind <= 0) {
        fprintf(stderr, "No kernel specified\n");
        usage();
        return -1;
}
kernel = argv[fileind];
/* slurp in the input kernel */
kernel_buf = slurp_decompress_file(kernel, &kernel_size);

dbgprintf("kernel: %p kernel_size: 0x%lx\n",
          kernel_buf, kernel_size);

if (get_memory_ranges(&info.memory_range, &info.memory_ranges,
        info.kexec_flags) < 0 || info.memory_ranges == 0) {
        fprintf(stderr, "Could not get memory layout\n");
        return -1;
}
/* if a kernel type was specified, try to honor it */
if (type) {
        for (i = 0; i < file_types; i++) {
                if (strcmp(type, file_type[i].name) == 0)
                        break;
        }
        if (i == file_types) {
                fprintf(stderr, "Unsupported kernel type %s\n", type);
                return -1;
        } else {
                /* make sure our file is really of that type */
                if (file_type[i].probe(kernel_buf, kernel_size) < 0)
                        guess_only = 1;
        }
}
if (!type || guess_only) {
        for (i = 0; i < file_types; i++) {
                if (file_type[i].probe(kernel_buf, kernel_size) == 0)
                        break;
        }
        if (i == file_types) {
                fprintf(stderr, "Cannot determine the file type "
                                "of %s\n", kernel);
                return -1;
        } else {
                if (guess_only) {
                        fprintf(stderr, "Wrong file type %s, "
                                "file matches type %s\n",
                                type, file_type[i].name);
                        return -1;
                }
        }
}
/* Figure out our native architecture before load */
native_arch = physical_arch();
if (native_arch < 0) {
        return -1;
}
info.kexec_flags |= native_arch;

result = file_type[i].load(argc, argv, kernel_buf, kernel_size, &info);
if (result < 0) {
        switch (result) {
        case ENOCRASHKERNEL:
                fprintf(stderr,
                        "No crash kernel segment found in /proc/iomem\n"
                        "Please check the crashkernel= boot parameter.\n");
                break;
        case EFAILED:
        default:
                fprintf(stderr, "Cannot load %s\n", kernel);
                break;
        }
        return result;
}
/* If we are not in native mode setup an appropriate trampoline */
if (arch_compat_trampoline(&info) < 0) {
        return -1;
}
if (info.kexec_flags & KEXEC_PRESERVE_CONTEXT) {
        add_backup_segments(&info, mem_min, mem_max - mem_min + 1);
}
/* Verify all of the segments load to a valid location in memory */
for (i = 0; i < info.nr_segments; i++) {
        if (!valid_memory_segment(&info, info.segment +i)) {
                fprintf(stderr, "Invalid memory segment %p - %p\n",
                        info.segment[i].mem,
                        ((char *)info.segment[i].mem) +
                        info.segment[i].memsz);
                return -1;
        }
}
/* Sort the segments and verify we don't have overlaps */
if (sort_segments(&info) < 0) {
        return -1;
}
/* if purgatory is loaded update it */
update_purgatory(&info);
if (entry)
        info.entry = entry;
```

```
        dbgprintf("kexec_load: entry = %p flags = 0x%lx\n",
                  info.entry, info.kexec_flags);
        if (kexec_debug)
                print_segments(stderr, &info);

        if (xen_present())
                result = xen_kexec_load(&info);
        else
                result = kexec_load(info.entry,
                                    info.nr_segments, info.segment,
                                    info.kexec_flags);
        if (result != 0) {
                /* The load failed, print some debugging information */
                fprintf(stderr, "kexec_load failed: %s\n",
                        strerror(errno));
                fprintf(stderr, "entry       = %p flags = 0x%lx\n",
                        info.entry, info.kexec_flags);
                print_segments(stderr, &info);
        }
        return result;
}

static int kexec_file_unload(unsigned long kexec_file_flags)
{
        int ret = 0;

        ret = kexec_file_load(-1, -1, 0, NULL, kexec_file_flags);
        if (ret != 0) {
                /* The unload failed, print some debugging information */
                fprintf(stderr, "kexec_file_load(unload) failed\n: %s\n",
                        strerror(errno));
        }
        return ret;
}

static int k_unload (unsigned long kexec_flags)
{
        int result;
        long native_arch;

        /* set the arch */
        native_arch = physical_arch();
        if (native_arch < 0) {
                return -1;
        }
        kexec_flags |= native_arch;

        if (xen_present())
                result = xen_kexec_unload(kexec_flags);
        else
                result = kexec_load(NULL, 0, NULL, kexec_flags);
        if (result != 0) {
                /* The unload failed, print some debugging information */
                fprintf(stderr, "kexec unload failed: %s\n",
                        strerror(errno));
        }
        return result;
}

/*
 *      Start a reboot.
 */
static int my_shutdown(void)
{
        char *args[] = {
                "shutdown",
                "-r",
                "now",
                NULL
        };

        execv("/sbin/shutdown", args);
        execv("/etc/shutdown", args);
        execv("/bin/shutdown", args);

        perror("shutdown");
        return -1;
}

/*
 *      Exec the new kernel (reboot)
 */
static int my_exec(void)
{
        if (xen_present())
                xen_kexec_exec();
        else
                reboot(LINUX_REBOOT_CMD_KEXEC);
        /* I have failed if I make it here */
        fprintf(stderr, "kexec failed: %s\n",
                strerror(errno));
        return -1;
}

static int kexec_loaded(void);

static int load_jump_back_helper_image(unsigned long kexec_flags, void *entry)
{
```

```
        int result;
        struct kexec_segment seg;

        memset(&seg, 0, sizeof(seg));
        result = kexec_load(entry, 1, &seg, kexec_flags);
        return result;
}

/*
 *      Jump back to the original kernel
 */
static int my_load_jump_back_helper(unsigned long kexec_flags, void *entry)
{
        int result;

        if (kexec_loaded()) {
                fprintf(stderr, "There is kexec kernel loaded, make sure "
                        "you are in kexeced kernel.\n");
                return -1;
        }
        if (!entry) {
                fprintf(stderr, "Please specify jump back entry "
                        "in command line\n");
                return -1;
        }
        result = load_jump_back_helper_image(kexec_flags, entry);
        if (result) {
                fprintf(stderr, "load jump back kernel failed: %s\n",
                        strerror(errno));
                return result;
        }
        return result;
}

static void version(void)
{
        printf(PACKAGE_STRING " released " PACKAGE_DATE "\n");
}

void usage(void)
{
        int i;

        version();
        printf("Usage: kexec [OPTION]... [kernel]\n"
               "Directly reboot into a new kernel\n"
               "\n"
               " -h, --help           Print this help.\n"
               " -v, --version        Print the version of kexec.\n"
               " -f, --force          Force an immediate kexec,\n"
               "                      don't call shutdown.\n"
               " -x, --no-ifdown      Don't bring down network interfaces.\n"
               " -y, --no-sync        Don't sync filesystems before kexec.\n"
               " -l, --load           Load the new kernel into the\n"
               "                      current kernel.\n"
               " -p, --load-panic     Load the new kernel for use on panic.\n"
               " -u, --unload         Unload the current kexec target kernel.\n"
               "                      If capture kernel is being unloaded\n"
               "                      specify -p with -u.\n"
               " -e, --exec           Execute a currently loaded kernel.\n"
               " -t, --type=TYPE      Specify the new kernel is of this type.\n"
               "     --mem-min=<addr> Specify the lowest memory address to\n"
               "                      load code into.\n"
               "     --mem-max=<addr> Specify the highest memory address to\n"
               "                      load code into.\n"
               "     --reuseinitrd    Reuse initrd from first boot.\n"
               "     --load-preserve-context Load the new kernel and preserve\n"
               "                      context of current kernel during kexec.\n"
               "     --load-jump-back-helper Load a helper image to jump back\n"
               "                      to original kernel.\n"
               "     --entry=<addr>   Specify jump back address.\n"
               "                      (0 means it's not jump back or\n"
               "                      preserve context)\n"
               "                      to original kernel.\n"
               " -s, --kexec-file-syscall Use file based syscall for kexec operation\n"
               " -d, --debug           Enable debugging to help spot a failure.\n"
               "\n"
               "Supported kernel file types and options: \n");
        for (i = 0; i < file_types; i++) {
                printf("%s\n", file_type[i].name);
                file_type[i].usage();
        }
        printf( "Architecture options: \n");
        arch_usage();
        printf("\n");
}

static int kexec_loaded(void)
{
        long ret = -1;
        FILE *fp;
        char *p;
        char line[3];

        /* No way to tell if an image is loaded under Xen, assume it is. */
        if (xen_present())
                return 1;
```

```
        fp = fopen("/sys/kernel/kexec_loaded", "r");
        if (fp == NULL)
                return -1;

        p = fgets(line, sizeof(line), fp);
        fclose(fp);

        if (p == NULL)
                return -1;

        ret = strtol(line, &p, 10);

        /* Too long */
        if (ret > INT_MAX)
                return -1;

        /* No digits were found */
        if (p == line)
                return -1;

        return (int)ret;
}

/*
 * Remove parameter from a kernel command line. Helper function by get_command_line().
 */
static void remove_parameter(char *line, const char *param_name)
{
        char *start, *end;

        start = strstr(line, param_name);

        /* parameter not found */
        if (!start)
                return;

        /*
         * check if that's really the start of a parameter and not in
         * the middle of the word
         */
        if (start != line && !isspace(*(start-1)))
                return;

        end = strstr(start, " ");
        if (!end)
                *start = 0;
        else {
                memmove(start, end+1, strlen(end));
                *(end + strlen(end)) = 0;
        }
}

/*
 * Returns the contents of the current command line to be used with
 * --reuse-cmdline option.  The function gets called from architecture specific
 * code. If we load a panic kernel, that function will strip the
 * "crashkernel=" option because it does not make sense that the crashkernel
 * reserves memory for a crashkernel (well, it would not boot since the
 * amount is exactly the same as the crashkernel has overall memory). Also,
 * remove the BOOT_IMAGE from lilo (and others) since that doesn't make
 * sense here any more. The kernel could be different even if we reuse the
 * commandline.
 *
 * The function returns dynamically allocated memory.
 */
char *get_command_line(void)
{
        FILE *fp;
        char *line;
        const int sizeof_line = 2048;

        line = malloc(sizeof_line);
        if (line == NULL)
                die("Could not allocate memory to read /proc/cmdline.");

        fp = fopen("/proc/cmdline", "r");
        if (!fp)
                die("Could not open /proc/cmdline.");

        if (fgets(line, sizeof_line, fp) == NULL)
                die("Can't read /proc/cmdline.");

        fclose(fp);

        /* strip newline */
        line[strlen(line) - 1] = '\0';

        remove_parameter(line, "BOOT_IMAGE");
        if (kexec_flags & KEXEC_ON_CRASH)
                remove_parameter(line, "crashkernel");

        return line;
}

/* check we retained the initrd */
static void check_reuse_initrd(void)
{
        char *str = NULL;
```

```
        char *line = get_command_line();

        str = strstr(line, "retain_initrd");
        free(line);

        if (str == NULL)
                die("unrecoverable error: current boot didn't "
                    "retain the initrd for reuse.\n");
}

char *concat_cmdline(const char *base, const char *append)
{
        char *cmdline;
        if (!base && !append)
                return NULL;
        if (append && !base)
                return xstrdup(append);
        if (base && !append)
                return xstrdup(base);
        cmdline = xmalloc(strlen(base) + 1 + strlen(append) + 1);
        strcpy(cmdline, base);
        strcat(cmdline, " ");
        strcat(cmdline, append);
        return cmdline;
}

/* New file based kexec system call related code */
static int do_kexec_file_load(int fileind, int argc, char **argv,
                        unsigned long flags) {

        char *kernel;
        int kernel_fd, i;
        struct kexec_info info;
        int ret = 0;
        char *kernel_buf;
        off_t kernel_size;

        memset(&info, 0, sizeof(info));
        info.segment = NULL;
        info.nr_segments = 0;
        info.entry = NULL;
        info.backup_start = 0;
        info.kexec_flags = flags;

        info.file_mode = 1;
        info.initrd_fd = -1;

        if (!is_kexec_file_load_implemented()) {
                fprintf(stderr, "syscall kexec_file_load not available.\n");
                return -1;
        }

        if (argc - fileind <= 0) {
                fprintf(stderr, "No kernel specified\n");
                usage();
                return -1;
        }

        kernel = argv[fileind];

        kernel_fd = open(kernel, O_RDONLY);
        if (kernel_fd == -1) {
                fprintf(stderr, "Failed to open file %s:%s\n", kernel,
                                strerror(errno));
                return -1;
        }

        /* slurp in the input kernel */
        kernel_buf = slurp_decompress_file(kernel, &kernel_size);

        for (i = 0; i < file_types; i++) {
                if (file_type[i].probe(kernel_buf, kernel_size) >= 0)
                        break;
        }

        if (i == file_types) {
                fprintf(stderr, "Cannot determine the file type " "of %s\n",
                                kernel);
                return -1;
        }

        ret = file_type[i].load(argc, argv, kernel_buf, kernel_size, &info);
        if (ret < 0) {
                fprintf(stderr, "Cannot load %s\n", kernel);
                return ret;
        }

        /*
         * If there is no initramfs, set KEXEC_FILE_NO_INITRAMFS flag so that
         * kernel does not return error with negative initrd_fd.
         */
        if (info.initrd_fd == -1)
                info.kexec_flags |= KEXEC_FILE_NO_INITRAMFS;

        ret = kexec_file_load(kernel_fd, info.initrd_fd, info.command_line_len,
                        info.command_line, info.kexec_flags);
        if (ret != 0)
                fprintf(stderr, "kexec_file_load failed: %s\n",
```

```
                                        strerror(errno));
        return ret;
}


int main(int argc, char *argv[])
{
        int do_load = 1;
        int do_exec = 0;
        int do_load_jump_back_helper = 0;
        int do_shutdown = 1;
        int do_sync = 1, skip_sync = 0;
        int do_ifdown = 0, skip_ifdown = 0;
        int do_unload = 0;
        int do_reuse_initrd = 0;
        int do_kexec_file_syscall = 0;
        void *entry = 0;
        char *type = 0;
        char *endptr;
        int opt;
        int result = 0;
        int fileind;
        static const struct option options[] = {
                KEXEC_ALL_OPTIONS
                { 0, 0, 0, 0},
        };
        static const char short_options[] = KEXEC_ALL_OPT_STR;

        /*
         * First check if --use-kexec-file-syscall is set. That changes lot of
         * things
         */
        while ((opt = getopt_long(argc, argv, short_options,
                                  options, 0)) != -1) {
                switch(opt) {
                case OPT_KEXEC_FILE_SYSCALL:
                        do_kexec_file_syscall = 1;
                        break;
                }
        }

        /* Reset getopt for the next pass. */
        opterr = 1;
        optind = 1;

        while ((opt = getopt_long(argc, argv, short_options,
                                  options, 0)) != -1) {
                switch(opt) {
                case '?':
                        usage();
                        return 1;
                case OPT_HELP:
                        usage();
                        return 0;
                case OPT_VERSION:
                        version();
                        return 0;
                case OPT_DEBUG:
                        kexec_debug = 1;
                case OPT_NOIFDOWN:
                        skip_ifdown = 1;
                        break;
                case OPT_NOSYNC:
                        skip_sync = 1;
                        break;
                case OPT_FORCE:
                        do_load = 1;
                        do_shutdown = 0;
                        do_sync = 1;
                        do_ifdown = 1;
                        do_exec = 1;
                        break;
                case OPT_LOAD:
                        do_load = 1;
                        do_exec = 0;
                        do_shutdown = 0;
                        break;
                case OPT_UNLOAD:
                        do_load = 0;
                        do_shutdown = 0;
                        do_sync = 0;
                        do_unload = 1;
                        if (do_kexec_file_syscall)
                                kexec_file_flags |= KEXEC_FILE_UNLOAD;
                        break;
                case OPT_EXEC:
                        do_load = 0;
                        do_shutdown = 0;
                        do_sync = 1;
                        do_ifdown = 1;
                        do_exec = 1;
                        break;
                case OPT_LOAD_JUMP_BACK_HELPER:
                        do_load = 0;
                        do_shutdown = 0;
                        do_sync = 1;
                        do_ifdown = 1;
                        do_exec = 0;
```

```
                do_load_jump_back_helper = 1;
                kexec_flags = KEXEC_PRESERVE_CONTEXT;
                break;
        case OPT_ENTRY:
                entry = (void *)strtoul(optarg, &endptr, 0);
                if (*endptr) {
                        fprintf(stderr,
                                "Bad option value in --entry=%s\n",
                                optarg);
                        usage();
                        return 1;
                }
                break;
        case OPT_LOAD_PRESERVE_CONTEXT:
                do_load = 1;
                do_exec = 0;
                do_shutdown = 0;
                do_sync = 1;
                kexec_flags = KEXEC_PRESERVE_CONTEXT;
                break;
        case OPT_TYPE:
                type = optarg;
                break;
        case OPT_PANIC:
                do_load = 1;
                do_exec = 0;
                do_shutdown = 0;
                do_sync = 0;
                if (do_kexec_file_syscall)
                        kexec_file_flags |= KEXEC_FILE_ON_CRASH;
                else
                        kexec_flags = KEXEC_ON_CRASH;
                break;
        case OPT_MEM_MIN:
                mem_min = strtoul(optarg, &endptr, 0);
                if (*endptr) {
                        fprintf(stderr,
                                "Bad option value in --mem-min=%s\n",
                                optarg);
                        usage();
                        return 1;
                }
                break;
        case OPT_MEM_MAX:
                mem_max = strtoul(optarg, &endptr, 0);
                if (*endptr) {
                        fprintf(stderr,
                                "Bad option value in --mem-max=%s\n",
                                optarg);
                        usage();
                        return 1;
                }
                break;
        case OPT_REUSE_INITRD:
                do_reuse_initrd = 1;
                break;
        case OPT_KEXEC_FILE_SYSCALL:
                /* We already parsed it. Nothing to do. */
                break;
        default:
                break;
        }
}

if (skip_ifdown)
        do_ifdown = 0;
if (skip_sync)
        do_sync = 0;

if (do_load && (kexec_flags & KEXEC_ON_CRASH) &&
    !is_crashkernel_mem_reserved()) {
        die("Memory for crashkernel is not reserved\n"
            "Please reserve memory by passing"
            "\"crashkernel=X@Y\" parameter to kernel\n"
            "Then try to loading kdump kernel\n");
}

if (do_load && (kexec_flags & KEXEC_PRESERVE_CONTEXT) &&
    mem_max == ULONG_MAX) {
        die("Please specify memory range used by kexeced kernel\n"
            "to preserve the context of original kernel with \n"
            "\"--mem-max\" parameter\n");
}

fileind = optind;
/* Reset getopt for the next pass; called in other source modules */
opterr = 1;
optind = 1;

result = arch_process_options(argc, argv);

/* Check for bogus options */
if (!do_load) {
        while((opt = getopt_long(argc, argv, short_options,
                                 options, 0)) != -1) {
                if ((opt == '?') || (opt >= OPT_ARCH_MAX)) {
                        usage();
                        return 1;
```

```
                        }
                }
        }

        if (do_reuse_initrd){
                check_reuse_initrd();
                arch_reuse_initrd();
        }

        if (do_unload) {
                if (do_kexec_file_syscall)
                        result = kexec_file_unload(kexec_file_flags);
                else
                        result = k_unload(kexec_flags);
        }
        if (do_load && (result == 0)) {
                if (do_kexec_file_syscall)
                        result = do_kexec_file_load(fileind, argc, argv,
                                                    kexec_file_flags);
                else
                        result = my_load(type, fileind, argc, argv,
                                                   kexec_flags, entry);
        }
        /* Don't shutdown unless there is something to reboot to! */
        if ((result == 0) && (do_shutdown || do_exec) && !kexec_loaded()) {
                die("Nothing has been loaded!\n");
        }
        if ((result == 0) && do_shutdown) {
                result = my_shutdown();
        }
        if ((result == 0) && do_sync) {
                sync();
        }
        if ((result == 0) && do_ifdown) {
                ifdown();
        }
        if ((result == 0) && do_exec) {
                result = my_exec();
        }
        if ((result == 0) && do_load_jump_back_helper) {
                result = my_load_jump_back_helper(kexec_flags, entry);
        }

        fflush(stdout);
        fflush(stderr);
        return result;
}
```

Notice for package(s)

kbd

```
The file
        kbdrate.c
is Copyright (C) 1992 Rickard E. Faith.

The files
        dumpkeys.c loadkeys.y loadkeys.analyze.l ksyms.c ksyms.h
        showkey.c keymaps.5 loadkeys.1 dumpkeys.1 showkey.1
are Copyright (C) 1993 Risto Kankkunen.

The files
        setfont.c setfont.8 mapscrn.c mapscrn.8
(and changes to loadkeys.y)
are Copyright (C) 1993 Eugene G. Crosser.

The files
        psfaddtable.1 psfgettable.1 psfstriptable.1
are Copyright (C) 1994 H. Peter Anvin.

The files
        chvt.c clrunimap.c deallocvt.c fgconsole.c findfile.c
        getfd.c getkeycodes.c getunimap.c kbd_mode.c loadunimap.c
        psffontop.c psfxtable.c resizecons.c screendump.c setkeycodes.c
        setleds.c setmetamode.c setvesablank.c showconsolefont.c
        spawn_console.c spawn_login.c
        chvt.1 deallocvt.1 kbd_mode.1 setleds.1 setmetamode.1
        getkeycodes.8 loadunimap.8 resizecons.8 setkeycodes.8
        kbd.FAQ.*
(and changes to earlier mentioned programs)
are Copyright (C) 1994-1999 Andries E. Brouwer.

The file
        openvt.c
is Copyright (C) 1994 by Jon Tombs <jon@gtex02.us.es>
portions also authors: Andries E. Brouwer <aeb@cwi.nl>
                       Joshua Spoerri <josh@cooper.edu>
                       Chuck Martin <cmartin@bigfoot.com>
                       Alexey Gladkov <gladkov.alexey@gmail.com>

The file
```

```
       kbdinfo.c
is Copyright (C) 2011 Alexey Gladkov.

The file
       setvtrgb.c
is Copyright (C) 2011 Alexey Gladkov,
portions also Copyright (C) 2011 Canonical Ltd.
              Authors: Seth Forshee <seth.forshee@canonical.com>
                       Dustin Kirkland <kirkland@canonical.com>

The files
       auth.c pam_auth.h pam.c parse.c screen.c username.c vlock.c vlock.h vt.c
is Copyright (C) 1994-1998 Michael K. Johnson <johnsonm@redhat.com>
   Copyright (C) 2002-2006 Dmitry V. Levin <ldv@altlinux.org>

All files in this package may be freely copied under the terms
of the GNU General Public License (GPL), version 2, or at your
option any later version - except possibly for the restrictions
mentioned in the directory consolefonts.
```

Notice for package(s)

initscripts
keymaps
lttng-modules
pciutils
pps-tools
sysvinit

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
```

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a

special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Notice for package(s)

libtirpc

```
/*
 * Copyright (c) Copyright (c) Bull S.A.  2005  All Rights Reserved.
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
```

```
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

Notice for package(s)

libtirpc

```
/*
 * Copyright (c) 2009, Sun Microsystems, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice,
 *   this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice,
 *   this list of conditions and the following disclaimer in the documentation
 *   and/or other materials provided with the distribution.
 * - Neither the name of Sun Microsystems, Inc. nor the names of its
 *   contributors may be used to endorse or promote products derived
 *   from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

/*
 * netname utility routines
 * convert from unix names to network names and vice-versa
 * This module is operating system dependent!
 * What we define here will work with any unix system that has adopted
 * the sun NIS domain architecture.
 */

#include <sys/param.h>
#include <rpc/rpc.h>
#include "rpc_com.h"
#ifdef YP
#include <rpcsvc/yp_prot.h>
#include <rpcsvc/ypclnt.h>
#endif
#include <ctype.h>
#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#ifndef MAXHOSTNAMELEN
#define MAXHOSTNAMELEN 256
#endif
#ifndef NGROUPS
#define NGROUPS 16
#endif

#define TYPE_BIT(type)  (sizeof (type) * CHAR_BIT)

#define TYPE_SIGNED(type) (((type) -1) < 0)

/*
** 302 / 1000 is log10(2.0) rounded up.
** Subtract one for the sign bit if the type is signed;
** add one for integer division truncation;
```

```
** add one more for a minus sign if the type is signed.
*/
#define INT_STRLEN_MAXIMUM(type) \
    ((TYPE_BIT(type) - TYPE_SIGNED(type)) * 302 / 1000 + 1 + TYPE_SIGNED(type))

static char *OPSYS = "unix";

/*
 * Figure out my fully qualified network name
 */
int
getnetname(name)
        char name[MAXNETNAMELEN+1];
{
        uid_t uid;

        uid = geteuid();
        if (uid == 0) {
                return (host2netname(name, (char *) NULL, (char *) NULL));
        } else {
                return (user2netname(name, uid, (char *) NULL));
        }
}


/*
 * Convert unix cred to network-name
 */
int
user2netname(netname, uid, domain)
        char netname[MAXNETNAMELEN + 1];
        const uid_t uid;
        const char *domain;
{
        char *dfltdom;

        if (domain == NULL) {
                if (__rpc_get_default_domain(&dfltdom) != 0) {
                        return (0);
                }
                domain = dfltdom;
        }
        if (strlen(domain) + 1 + INT_STRLEN_MAXIMUM(u_long) + 1 + strlen(OPSYS) > MAXNETNAMELEN) {
                return (0);
        }
        (void) sprintf(netname, "%s.%ld@%s", OPSYS, (u_long)uid, domain);
        return (1);
}


/*
 * Convert host to network-name
 */
int
host2netname(netname, host, domain)
        char netname[MAXNETNAMELEN + 1];
        const char *host;
        const char *domain;
{
        char *dfltdom;
        char hostname[MAXHOSTNAMELEN+1];

        if (domain == NULL) {
                if (__rpc_get_default_domain(&dfltdom) != 0) {
                        return (0);
                }
                domain = dfltdom;
        }
        if (host == NULL) {
                (void) gethostname(hostname, sizeof(hostname));
                host = hostname;
        }
        if (strlen(domain) + 1 + strlen(host) + 1 + strlen(OPSYS) > MAXNETNAMELEN) {
                return (0);
        }
        (void) sprintf(netname, "%s.%s@%s", OPSYS, host, domain);
        return (1);
}
```

Notice for package(s)

rpcbind

```
/*
 * Copyright (c) Copyright (c) Bull S.A.  2005  All Rights Reserved.
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
```

```
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

Notice for package(s)

rpcbind

```
/*
 * Copyright (c) 2009, Sun Microsystems, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice,
 *   this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice,
 *   this list of conditions and the following disclaimer in the documentation
 *   and/or other materials provided with the distribution.
 * - Neither the name of Sun Microsystems, Inc. nor the names of its
 *   contributors may be used to endorse or promote products derived
 *   from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
/*
 * Copyright (c) 1986 - 1991 by Sun Microsystems, Inc.
 */
/*
 * rpcinfo: ping a particular rpc program
 *      or dump the the registered programs on the remote machine.
 */

/*
 * We are for now defining PORTMAP here.  It doesnt even compile
 * unless it is defined.
 */
#ifndef PORTMAP
#define PORTMAP
#endif

/*
 * If PORTMAP is defined, rpcinfo will talk to both portmapper and
 * rpcbind programs; else it talks only to rpcbind. In the latter case
 * all the portmapper specific options such as -u, -t, -p become void.
 */
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <rpc/rpc.h>
#include <stdio.h>
#include <rpc/rpcb_prot.h>
#include <rpc/rpcent.h>
#include <rpc/nettype.h>
#include <rpc/rpc_com.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <err.h>
#include <ctype.h>

#ifdef PORTMAP                  /* Support for version 2 portmapper */
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <rpc/pmap_prot.h>
#include <rpc/pmap_clnt.h>
#endif
```

```
#define max(a,b) ((a) > (b) ? (a) : (b))

#define MIN_VERS        ((u_long)0)
#define MAX_VERS        ((u_long)4294967295UL)
#define UNKNOWN         "unknown"

/*
 * Functions to be performed.
 */
#define NONE            0       /* no function */
#define PMAPDUMP        1       /* dump portmapper registrations */
#define TCPPING         2       /* ping TCP service */
#define UDPPING         3       /* ping UDP service */
#define BROADCAST       4       /* ping broadcast service */
#define DELETES         5       /* delete registration for the service */
#define ADDRPING        6       /* pings at the given address */
#define PROGPING        7       /* pings a program on a given host */
#define RPCBDUMP        8       /* dump rpcbind registrations */
#define RPCBDUMP_SHORT  9       /* dump rpcbind registrations - short version */
#define RPCBADDRLIST    10      /* dump addr list about one prog */
#define RPCBGETSTAT     11      /* Get statistics */

struct netidlist
{
  char *netid;
  struct netidlist *next;
};

struct verslist
{
  int vers;
  struct verslist *next;
};

struct rpcbdump_short
{
  u_long prog;
  struct verslist *vlist;
  struct netidlist *nlist;
  struct rpcbdump_short *next;
  char *owner;
};


#ifdef PORTMAP
static void ip_ping (u_short, char *, int, char **);
static void pmapdump (int, char **);
static CLIENT *ip_getclient(const char *hostname, rpcprog_t prognum, rpcvers_t versnum, const char *proto);
#endif

static bool_t reply_proc (void *, struct netbuf *, struct netconfig *);
static void brdcst (int, char **);
static void addrping (char *, char *, int, char **);
static void progping (char *, int, char **);
static CLIENT *clnt_addr_create (char *, struct netconfig *, u_long, u_long);
static CLIENT *clnt_rpcbind_create (char *, int, struct netbuf **);
static CLIENT *getclnthandle (char *, struct netconfig *, u_long,
                              struct netbuf **);
static int pstatus (CLIENT *, u_long, u_long);
static void rpcbdump (int, char *, int, char **);
static void rpcbgetstat (int, char **);
static void rpcbaddrlist (char *, int, char **);
static void deletereg (char *, int, char **);
static void print_rmtcallstat (int, rpcb_stat *);
static void print_getaddrstat (int, rpcb_stat *);
static void usage (void);
static u_long getprognum (char *);
static u_long getvers (char *);
static char *spaces (int);
static bool_t add_version (struct rpcbdump_short *, u_long);
static bool_t add_netid (struct rpcbdump_short *, char *);

int main (int argc, char **argv);

int
main (int argc, char **argv)
{
  register int c;
  int errflg;
  int function;
  char *netid = NULL;
  char *address = NULL;
#ifdef PORTMAP
  char *strptr;
  u_short portnum = 0;
#endif

  function = NONE;
  errflg = 0;
#ifdef PORTMAP
  while ((c = getopt (argc, argv, "a:bdlmn:pstT:u")) != -1)
#else
  while ((c = getopt (argc, argv, "a:bdlmn:sT:")) != -1)
#endif
    {
      switch (c)
        {
```

```
#ifdef PORTMAP
        case 'p':
          if (function != NONE)
            errflg = 1;
          else
            function = PMAPDUMP;
          break;

        case 't':
          if (function != NONE)
            errflg = 1;
          else
            function = TCPPING;
          break;

        case 'u':
          if (function != NONE)
            errflg = 1;
          else
            function = UDPPING;
          break;

        case 'n':
          portnum = (u_short) strtol (optarg, &strptr, 10);
          if (strptr == optarg || *strptr != '\0')
            {
              fprintf (stderr, "rpcinfo: %s is illegal port number\n",
                       optarg);
              exit (1);
            }
          break;
#endif

        case 'a':
          address = optarg;
          if (function != NONE)
            errflg = 1;
          else
            function = ADDRPING;
          break;

        case 'b':
          if (function != NONE)
            errflg = 1;
          else
            function = BROADCAST;
          break;

        case 'd':
          if (function != NONE)
            errflg = 1;
          else
            function = DELETES;
          break;

        case 'l':
          if (function != NONE)
            errflg = 1;
          else
            function = RPCBADDRLIST;
          break;

        case 'm':
          if (function != NONE)
            errflg = 1;
          else
            function = RPCBGETSTAT;
          break;

        case 's':
          if (function != NONE)
            errflg = 1;
          else
            function = RPCBDUMP_SHORT;
          break;

        case 'T':
          netid = optarg;
          break;
        case '?':
          errflg = 1;
          break;
        }
    }

  if (errflg || ((function == ADDRPING) && !netid))
    {
      usage ();
      return 1;
    }

  if (function == NONE)
    {
      if (argc - optind > 1)
        function = PROGPING;
      else
        function = RPCBDUMP;
    }
```

```
  switch (function)
    {
#ifdef PORTMAP
    case PMAPDUMP:
      if (portnum != 0)
        {
          usage ();
          return 1;
        }
      pmapdump (argc - optind, argv + optind);
      break;

    case UDPPING:
      ip_ping (portnum, "udp", argc - optind, argv + optind);
      break;

    case TCPPING:
      ip_ping (portnum, "tcp", argc - optind, argv + optind);
      break;
#endif
    case BROADCAST:
      brdcst (argc - optind, argv + optind);
      break;
    case DELETES:
      deletereg (netid, argc - optind, argv + optind);
      break;
    case ADDRPING:
      addrping (address, netid, argc - optind, argv + optind);
      break;
    case PROGPING:
      progping (netid, argc - optind, argv + optind);
      break;
    case RPCBDUMP:
    case RPCBDUMP_SHORT:
      rpcbdump (function, netid, argc - optind, argv + optind);
      break;
    case RPCBGETSTAT:
      rpcbgetstat (argc - optind, argv + optind);
      break;
    case RPCBADDRLIST:
      rpcbaddrlist (netid, argc - optind, argv + optind);
      break;
    }
  return (0);
}

static CLIENT *
local_rpcb (rpcprog_t prog, rpcvers_t vers)
{
#if 0
  void *localhandle;
  struct netconfig *nconf;
  CLIENT *clnt;

  localhandle = setnetconfig();
  while ((nconf = getnetconfig(localhandle)) != NULL) {
    if (nconf->nc_protofmly != NULL &&
        strcmp(nconf->nc_protofmly, NC_LOOPBACK) == 0)
      break;
  }
  if (nconf == NULL) {
    warnx("getnetconfig: %s", nc_sperror());
    return (NULL);
  }

  clnt = clnt_tp_create (/* "localhost"*/ NULL, prog, vers, nconf);
  endnetconfig(localhandle);
  return clnt;
#else
  struct netbuf nbuf;
  struct sockaddr_un sun;
  int sock;

  memset (&sun, 0, sizeof sun);
  sock = socket (AF_LOCAL, SOCK_STREAM, 0);
  if (sock < 0)
    return NULL;

  sun.sun_family = AF_LOCAL;
  strcpy (sun.sun_path, _PATH_RPCBINDSOCK);
  nbuf.len = SUN_LEN (&sun);
  nbuf.maxlen = sizeof (struct sockaddr_un);
  nbuf.buf = &sun;

  return clnt_vc_create (sock, &nbuf, prog, vers, 0, 0);
#endif
}

#ifdef PORTMAP
static enum clnt_stat
ip_ping_one(client, vers)
     CLIENT *client;
     u_int32_t vers;
{
  struct timeval to = { .tv_sec = 10, .tv_usec = 0 };

  (void) CLNT_CONTROL (client, CLSET_VERS, &vers);
```

```
  return CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                    (char *) NULL, (xdrproc_t) xdr_void, (char *) NULL,
                    to);
}

/*
 * If portnum is 0, then go and get the address from portmapper, which happens
 * transparently through clnt*_create(); If version number is not given, it
 * tries to find out the version number by making a call to version 0 and if
 * that fails, it obtains the high order and the low order version number. If
 * version 0 calls succeeds, it tries for MAXVERS call and repeats the same.
 */
static void
ip_ping (portnum, proto, argc, argv)
     u_short portnum;
     char *proto;
     int argc;
     char **argv;
{
  CLIENT *client;
  enum clnt_stat rpc_stat;
  const char *hostname;
  u_long prognum, vers, minvers, maxvers;
  struct rpc_err rpcerr;
  int failure = 0;

  if (argc < 2 || argc > 3)
    {
      usage ();
      exit (1);
    }

  hostname = argv[0];
  prognum = getprognum (argv[1]);
  if (argc == 2)
    {                           /* Version number not known */
      /*
       * A call to version 0 should fail with a program/version
       * mismatch, and give us the range of versions supported.
       */
      vers = MIN_VERS;
    }
  else
    {
      vers = getvers (argv[2]);
    }

  client = ip_getclient(hostname, prognum, vers, proto);

  rpc_stat = ip_ping_one(client, vers);
  if (argc != 2)
    {
      /* Version number was known */
      if (pstatus (client, prognum, vers) < 0)
        exit (1);
      (void) CLNT_DESTROY (client);
      return;
    }

  /* Version number not known */
  if (rpc_stat == RPC_PROGVERSMISMATCH)
    {
      clnt_geterr (client, &rpcerr);
      minvers = rpcerr.re_vers.low;
      maxvers = rpcerr.re_vers.high;
    }
  else if (rpc_stat == RPC_SUCCESS)
    {
      /*
       * Oh dear, it DOES support version 0.
       * Let's try version MAX_VERS.
       */
      rpc_stat = ip_ping_one(client, MAX_VERS);
      if (rpc_stat == RPC_PROGVERSMISMATCH)
        {
          clnt_geterr (client, &rpcerr);
          minvers = rpcerr.re_vers.low;
          maxvers = rpcerr.re_vers.high;
        }
      else if (rpc_stat == RPC_SUCCESS)
        {
          /*
           * It also supports version MAX_VERS.
           * Looks like we have a wise guy.
           * OK, we give them information on all
           * 4 billion versions they support...
           */
          minvers = 0;
          maxvers = MAX_VERS;
        }
      else
        {
          (void) pstatus (client, prognum, MAX_VERS);
          exit (1);
        }
    }
  else
    {
```

```
      (void) pstatus (client, prognum, (u_long) 0);
      exit (1);
    }
  for (vers = minvers; vers <= maxvers; vers++)
    {
      rpc_stat = ip_ping_one(client, vers);
      if (pstatus (client, prognum, vers) < 0)
        failure = 1;
    }
  if (failure)
    exit (1);
  (void) CLNT_DESTROY (client);
  return;
}

/*
 * Dump all the portmapper registerations
 */
static void
pmapdump (argc, argv)
     int argc;
     char **argv;
{
  struct pmaplist *head = NULL;
  struct timeval minutetimeout;
  register CLIENT *client;
  struct rpcent *rpc;
  enum clnt_stat clnt_st;
  struct rpc_err err;
  char *host = NULL;

  if (argc > 1)
    {
      usage ();
      exit (1);
    }
  if (argc == 1)
    {
      host = argv[0];

      /* This is a little bit more complicated than it should be.
       * ip_getclient will do an rpcb_getaddr call to identify the
       * port of the portmapper - but it works, and it's easier than
       * creating a copy of ip_getclient that avoids the getaddr call.
       */
      client = ip_getclient(host, PMAPPROG, PMAPVERS, "tcp");
    }
  else
    client = local_rpcb (PMAPPROG, PMAPVERS);

  if (client == NULL)
    {
      if (rpc_createerr.cf_stat == RPC_TLIERROR)
        {
          /*
           * "Misc. TLI error" is not too helpful. Most likely
           * the connection to the remote server timed out, so
           * this error is at least less perplexing.
           */
          rpc_createerr.cf_stat = RPC_PMAPFAILURE;
          rpc_createerr.cf_error.re_status = RPC_FAILED;
        }
      clnt_pcreateerror ("rpcinfo: can't contact portmapper");
      exit (1);
    }

  minutetimeout.tv_sec = 60;
  minutetimeout.tv_usec = 0;

  clnt_st = CLNT_CALL (client, PMAPPROC_DUMP, (xdrproc_t) xdr_void,
                       NULL, (xdrproc_t) xdr_pmaplist_ptr, (char *) &head,
                       minutetimeout);
  if (clnt_st != RPC_SUCCESS)
    {
      if ((clnt_st == RPC_PROGVERSMISMATCH) || (clnt_st == RPC_PROGUNAVAIL))
        {
          CLNT_GETERR (client, &err);
          if (err.re_vers.low > PMAPVERS)
            fprintf (stderr,
                     "%s does not support portmapper.  Try rpcinfo %s instead\n",
                     host, host);
          exit (1);
        }
      clnt_perror (client, "rpcinfo: can't contact portmapper");
      exit (1);
    }
  if (head == NULL)
    {
      printf ("No remote programs registered.\n");
    }
  else
    {
      printf ("   program vers proto   port  service\n");
      for (; head != NULL; head = head->pml_next)
        {
          printf ("%10ld%5ld", head->pml_map.pm_prog, head->pml_map.pm_vers);
          if (head->pml_map.pm_prot == IPPROTO_UDP)
            printf ("%6s", "udp");
```

```
          else if (head->pml_map.pm_prot == IPPROTO_TCP)
            printf ("%6s", "tcp");
          else
            printf ("%6ld", head->pml_map.pm_prot);
          printf ("%7ld", head->pml_map.pm_port);
          rpc = getrpcbynumber (head->pml_map.pm_prog);
          if (rpc)
            printf ("  %s\n", rpc->r_name);
          else
            printf ("\n");
        }
    }
}

/*
 * Try to obtain the address of a given host/program/version, using the
 * specified protocol (one of udp or tcp).
 * This loops over all netconfig entries (according to the order given by
 * netpath and the config file), and tries to resolve the hostname, and obtain
 * the address using rpcb_getaddr.
 */
CLIENT *
ip_getclient(hostname, prognum, versnum, proto)
     const char *hostname;
     rpcprog_t prognum;
     rpcvers_t versnum;
     const char *proto;
{
  void *handle;
  enum clnt_stat saved_stat = RPC_SUCCESS;
  struct netconfig *nconf, *result = NULL;
  struct netbuf bind_address;
  struct sockaddr_storage __sa;
  CLIENT *client;

  memset(&bind_address, 0, sizeof(bind_address));
  bind_address.maxlen = sizeof(__sa);
  bind_address.buf = &__sa;

  handle = setnetconfig();
  while ((nconf = getnetconfig(handle)) != NULL)
    {
      if (!strcmp(nconf->nc_proto, proto)) {
        if (rpcb_getaddr(prognum, versnum, nconf, &bind_address, hostname))
          {
            result = getnetconfigent(nconf->nc_netid);
            endnetconfig(handle);
            break;
          }

        if (rpc_createerr.cf_stat != RPC_UNKNOWNHOST)
          {
            clnt_pcreateerror (hostname);
            exit (1);
          }

        saved_stat = rpc_createerr.cf_stat;
      }
    }

  if (result == NULL)
    {
      if (saved_stat != RPC_SUCCESS)
        {
          rpc_createerr.cf_stat = saved_stat;
          clnt_pcreateerror (hostname);
        }
      else
        fprintf (stderr, "Cannot find suitable transport for protocol %s\n", proto);

      exit (1);
    }

  client = clnt_tli_create(RPC_ANYFD, result, &bind_address, prognum, versnum, 0, 0);
  if (client == NULL)
    {
      clnt_pcreateerror(hostname);
      exit (1);
    }

  freenetconfigent(result);
  return client;
}
#endif /* PORTMAP */

static int
sa_len(struct sockaddr *sa)
{
    socklen_t salen;

    switch (sa->sa_family)
    {
        case AF_LOCAL:
            salen = sizeof (struct sockaddr_un);
            break;
        case AF_INET:
            salen = sizeof (struct sockaddr_in);
            break;
```

```
        case AF_INET6:
            salen = sizeof (struct sockaddr_in6);
            break;
        default:
            salen = 0;
            break;
    }
    return salen;
}

/*
 * reply_proc collects replies from the broadcast.
 * to get a unique list of responses the output of rpcinfo should
 * be piped through sort(1) and then uniq(1).
 */

 /*ARGSUSED*/ static bool_t
reply_proc (res, who, nconf)
     void *res;                 /* Nothing comes back */
     struct netbuf *who;        /* Who sent us the reply */
     struct netconfig *nconf;   /* On which transport the reply came */
{
  char *uaddr;
  char hostbuf[NI_MAXHOST];
  char *hostname;
  struct sockaddr *sa = (struct sockaddr *) who->buf;

  if (getnameinfo (sa, sa_len (sa), hostbuf, NI_MAXHOST, NULL, 0, 0))
    {
      hostname = UNKNOWN;
    }
  else
    {
      hostname = hostbuf;
    }
  if (!(uaddr = taddr2uaddr (nconf, who)))
    {
      uaddr = UNKNOWN;
    }
  printf ("%s\t%s\n", uaddr, hostname);
  if (strcmp (uaddr, UNKNOWN))
    free ((char *) uaddr);
  return (FALSE);
}

static void
brdcst (argc, argv)
     int argc;
     char **argv;
{
  enum clnt_stat rpc_stat;
  u_long prognum, vers;

  if (argc != 2)
    {
      usage ();
      exit (1);
    }
  prognum = getprognum (argv[0]);
  vers = getvers (argv[1]);
  rpc_stat = rpc_broadcast (prognum, vers, NULLPROC,
                            (xdrproc_t) xdr_void, (char *) NULL,
                            (xdrproc_t) xdr_void, (char *) NULL,
                            (resultproc_t) reply_proc, NULL);
  if ((rpc_stat != RPC_SUCCESS) && (rpc_stat != RPC_TIMEDOUT))
    {
      fprintf (stderr, "rpcinfo: broadcast failed: %s\n",
               clnt_sperrno (rpc_stat));
      exit (1);
    }
  exit (0);
}

static bool_t
add_version (rs, vers)
     struct rpcbdump_short *rs;
     u_long vers;
{
  struct verslist *vl;

  for (vl = rs->vlist; vl; vl = vl->next)
    if (vl->vers == vers)
      break;
  if (vl)
    return (TRUE);
  vl = (struct verslist *) malloc (sizeof (struct verslist));
  if (vl == NULL)
    return (FALSE);
  vl->vers = vers;
  vl->next = rs->vlist;
  rs->vlist = vl;
  return (TRUE);
}

static bool_t
add_netid (rs, netid)
     struct rpcbdump_short *rs;
     char *netid;
```

```
{
  struct netidlist *nl;

  for (nl = rs->nlist; nl; nl = nl->next)
    if (strcmp (nl->netid, netid) == 0)
      break;
  if (nl)
    return (TRUE);
  nl = (struct netidlist *) malloc (sizeof (struct netidlist));
  if (nl == NULL)
    return (FALSE);
  nl->netid = netid;
  nl->next = rs->nlist;
  rs->nlist = nl;
  return (TRUE);
}

static void
rpcbdump (dumptype, netid, argc, argv)
     int dumptype;
     char *netid;
     int argc;
     char **argv;
{
  rpcblist_ptr head = NULL;
  struct timeval minutetimeout;
  register CLIENT *client;
  struct rpcent *rpc;
  char *host;
  struct netidlist *nl;
  struct verslist *vl;
  struct rpcbdump_short *rs, *rs_tail = NULL;
  char buf[256];
  enum clnt_stat clnt_st;
  struct rpc_err err;
  struct rpcbdump_short *rs_head = NULL;

  if (argc > 1)
    {
      usage ();
      exit (1);
    }
  if (argc == 1)
    {
      host = argv[0];
      if (netid == NULL)
        {
          client = clnt_rpcbind_create (host, RPCBVERS, NULL);
        }
      else
        {
          struct netconfig *nconf;

          nconf = getnetconfigent (netid);
          if (nconf == NULL)
            {
              nc_perror ("rpcinfo: invalid transport");
              exit (1);
            }
          client = getclnthandle (host, nconf, RPCBVERS, NULL);
          if (nconf)
            (void) freenetconfigent (nconf);
        }
    }
  else
    client = local_rpcb (PMAPPROG, RPCBVERS);

  if (client == (CLIENT *) NULL)
    {
      clnt_pcreateerror ("rpcinfo: can't contact rpcbind");
      exit (1);
    }
  minutetimeout.tv_sec = 60;
  minutetimeout.tv_usec = 0;
  clnt_st = CLNT_CALL (client, RPCBPROC_DUMP, (xdrproc_t) xdr_void,
                       NULL, (xdrproc_t) xdr_rpcblist_ptr, (char *) &head,
                       minutetimeout);
  if (clnt_st != RPC_SUCCESS)
    {
      if ((clnt_st == RPC_PROGVERSMISMATCH) || (clnt_st == RPC_PROGUNAVAIL))
        {
          int vers;

          CLNT_GETERR (client, &err);
          if (err.re_vers.low == RPCBVERS4)
            {
              vers = RPCBVERS4;
              clnt_control (client, CLSET_VERS, (char *) &vers);
              clnt_st = CLNT_CALL (client, RPCBPROC_DUMP,
                                   (xdrproc_t) xdr_void, NULL,
                                   (xdrproc_t) xdr_rpcblist_ptr,
                                   (char *) &head, minutetimeout);
              if (clnt_st != RPC_SUCCESS)
                goto failed;
            }
          else
            {
              if (err.re_vers.high == PMAPVERS)
```

```
               {
                 int high, low;
                 struct pmaplist *pmaphead = NULL;
                 rpcblist_ptr list, prev = NULL;

                 vers = PMAPVERS;
                 clnt_control (client, CLSET_VERS, (char *) &vers);
                 clnt_st = CLNT_CALL (client, PMAPPROC_DUMP,
                                      (xdrproc_t) xdr_void, NULL,
                                      (xdrproc_t) xdr_pmaplist_ptr,
                                      (char *) &pmaphead, minutetimeout);
                 if (clnt_st != RPC_SUCCESS)
                   goto failed;
                 /*
                  * convert to rpcblist_ptr format
                  */
                 for (head = NULL; pmaphead != NULL;
                      pmaphead = pmaphead->pml_next)
                   {
                     list = (rpcblist *) malloc (sizeof (rpcblist));
                     if (list == NULL)
                       goto error;
                     if (head == NULL)
                       head = list;
                     else
                       prev->rpcb_next = (rpcblist_ptr) list;

                     list->rpcb_next = NULL;
                     list->rpcb_map.r_prog = pmaphead->pml_map.pm_prog;
                     list->rpcb_map.r_vers = pmaphead->pml_map.pm_vers;
                     if (pmaphead->pml_map.pm_prot == IPPROTO_UDP)
                       list->rpcb_map.r_netid = "udp";
                     else if (pmaphead->pml_map.pm_prot == IPPROTO_TCP)
                       list->rpcb_map.r_netid = "tcp";
                     else
                       {
#define MAXLONG_AS_STRING       "2147483648"
                         list->rpcb_map.r_netid =
                           malloc (strlen (MAXLONG_AS_STRING) + 1);
                         if (list->rpcb_map.r_netid == NULL)
                           goto error;
                         sprintf (list->rpcb_map.r_netid, "%6ld",
                                  pmaphead->pml_map.pm_prot);
                       }
                     list->rpcb_map.r_owner = UNKNOWN;
                     low = pmaphead->pml_map.pm_port & 0xff;
                     high = (pmaphead->pml_map.pm_port >> 8) & 0xff;
                     list->rpcb_map.r_addr = strdup ("0.0.0.0.XXX.XXX");
                     sprintf (&list->rpcb_map.r_addr[8], "%d.%d", high, low);
                     prev = list;
                   }
               }
           }
       }
     else
       {                        /* any other error */
       failed:
         clnt_perror (client, "rpcinfo: can't contact rpcbind: ");
         exit (1);
       }
   }
 if (head == NULL)
   {
     printf ("No remote programs registered.\n");
   }
 else if (dumptype == RPCBDUMP)
   {
     printf
       ("   program version netid     address                service    owner\n");
     for (; head != NULL; head = head->rpcb_next)
       {
         printf ("%10u%5u    ",
                 head->rpcb_map.r_prog, head->rpcb_map.r_vers);
         printf ("%-9s ", head->rpcb_map.r_netid);
         printf ("%-22s", head->rpcb_map.r_addr);
         rpc = getrpcbynumber (head->rpcb_map.r_prog);
         if (rpc)
           printf (" %-10s", rpc->r_name);
         else
           printf (" %-10s", "-");
         printf (" %s\n", head->rpcb_map.r_owner);
       }
   }
 else if (dumptype == RPCBDUMP_SHORT)
   {
     for (; head != NULL; head = head->rpcb_next)
       {
         for (rs = rs_head; rs; rs = rs->next)
           if (head->rpcb_map.r_prog == rs->prog)
             break;
         if (rs == NULL)
           {
             rs = (struct rpcbdump_short *)
               malloc (sizeof (struct rpcbdump_short));
             if (rs == NULL)
               goto error;
             rs->next = NULL;
             if (rs_head == NULL)
```

```
                {
                  rs_head = rs;
                  rs_tail = rs;
                }
              else
                {
                  rs_tail->next = rs;
                  rs_tail = rs;
                }
              rs->prog = head->rpcb_map.r_prog;
              rs->owner = head->rpcb_map.r_owner;
              rs->nlist = NULL;
              rs->vlist = NULL;
            }
          if (add_version (rs, head->rpcb_map.r_vers) == FALSE)
            goto error;
          if (add_netid (rs, head->rpcb_map.r_netid) == FALSE)
            goto error;
        }
      printf
        ("   program version(s) netid(s)                         service     owner\n");
      for (rs = rs_head; rs; rs = rs->next)
        {
          char *p = buf;

          printf ("%10ld  ", rs->prog);
          for (vl = rs->vlist; vl; vl = vl->next)
            {
              sprintf (p, "%d", vl->vers);
              p = p + strlen (p);
              if (vl->next)
                sprintf (p++, ",");
            }
          printf ("%-10s", buf);
          buf[0] = '\0';
          for (nl = rs->nlist; nl; nl = nl->next)
            {
              strcat (buf, nl->netid);
              if (nl->next)
                strcat (buf, ",");
            }
          printf ("%-32s", buf);
          rpc = getrpcbynumber (rs->prog);
          if (rpc)
            printf (" %-11s", rpc->r_name);
          else
            printf (" %-11s", "-");
          printf (" %s\n", rs->owner);
        }
    }
  clnt_destroy (client);
  return;
error:fprintf (stderr, "rpcinfo: no memory\n");
  return;
}

static char nullstring[] = "\000";

static void
rpcbaddrlist (netid, argc, argv)
     char *netid;
     int argc;
     char **argv;
{
  rpcb_entry_list_ptr head = NULL;
  struct timeval minutetimeout;
  register CLIENT *client;
  struct rpcent *rpc;
  char *host;
  RPCB parms;
  struct netbuf *targaddr;

  if (argc != 3)
    {
      usage ();
      exit (1);
    }
  host = argv[0];
  if (netid == NULL)
    {
      client = clnt_rpcbind_create (host, RPCBVERS4, &targaddr);
    }
  else
    {
      struct netconfig *nconf;

      nconf = getnetconfigent (netid);
      if (nconf == NULL)
        {
          nc_perror ("rpcinfo: invalid transport");
          exit (1);
        }
      client = getclnthandle (host, nconf, RPCBVERS4, &targaddr);
      if (nconf)
        (void) freenetconfigent (nconf);
    }
  if (client == (CLIENT *) NULL)
    {
```

```
      clnt_pcreateerror ("rpcinfo: can't contact rpcbind");
      exit (1);
    }
  minutetimeout.tv_sec = 60;
  minutetimeout.tv_usec = 0;

  parms.r_prog = getprognum (argv[1]);
  parms.r_vers = getvers (argv[2]);
  parms.r_netid = client->cl_netid;
  if (targaddr == NULL)
    {
      parms.r_addr = nullstring;        /* for XDRing */
    }
  else
    {
      /*
       * We also send the remote system the address we
       * used to contact it in case it can help it
       * connect back with us
       */
      struct netconfig *nconf;

      nconf = getnetconfigent (client->cl_netid);
      if (nconf != NULL)
        {
          parms.r_addr = taddr2uaddr (nconf, targaddr);
          if (parms.r_addr == NULL)
            parms.r_addr = nullstring;
          freenetconfigent (nconf);
        }
      else
        {
          parms.r_addr = nullstring;    /* for XDRing */
        }
      free (targaddr->buf);
      free (targaddr);
    }
  parms.r_owner = nullstring;

  if (CLNT_CALL (client, RPCBPROC_GETADDRLIST, (xdrproc_t) xdr_rpcb,
                 (char *) &parms, (xdrproc_t) xdr_rpcb_entry_list_ptr,
                 (char *) &head, minutetimeout) != RPC_SUCCESS)
    {
      clnt_perror (client, "rpcinfo: can't contact rpcbind: ");
      exit (1);
    }
  if (head == NULL)
    {
      printf ("No remote programs registered.\n");
    }
  else
    {
      printf
        ("   program vers  tp_family/name/class    address\t\t  service\n");
      for (; head != NULL; head = head->rpcb_entry_next)
        {
          rpcb_entry *re;
          char buf[128];

          re = &head->rpcb_entry_map;
          printf ("%10u%3u    ", parms.r_prog, parms.r_vers);
          sprintf (buf, "%s/%s/%s ",
                   re->r_nc_protofmly, re->r_nc_proto,
                   re->r_nc_semantics == NC_TPI_CLTS ? "clts" :
                   re->r_nc_semantics == NC_TPI_COTS ? "cots" : "cots_ord");
          printf ("%-24s", buf);
          printf ("%-24s", re->r_maddr);
          rpc = getrpcbynumber (parms.r_prog);
          if (rpc)
            printf (" %-13s", rpc->r_name);
          else
            printf (" %-13s", "-");
          printf ("\n");
        }
    }
  clnt_destroy (client);
  return;
}

/*
 * monitor rpcbind
 */
static void
rpcbgetstat (argc, argv)
     int argc;
     char **argv;
{
  rpcb_stat_byvers inf;
  struct timeval minutetimeout;
  register CLIENT *client;
  char *host;
  int i, j;
  rpcbs_addrlist *pa;
  rpcbs_rmtcalllist *pr;
  int cnt, flen;
#define MAXFIELD        64
  char fieldbuf[MAXFIELD];
#define MAXLINE         256
```

```
  char linebuf[MAXLINE];
  char *cp, *lp;
  char *pmaphdr[] = {
    "NULL", "SET", "UNSET", "GETPORT",
    "DUMP", "CALLIT"
  };
  char *rpcb3hdr[] = {
    "NULL", "SET", "UNSET", "GETADDR", "DUMP", "CALLIT", "TIME",
    "U2T", "T2U"
  };
  char *rpcb4hdr[] = {
    "NULL", "SET", "UNSET", "GETADDR", "DUMP", "CALLIT", "TIME",
    "U2T", "T2U", "VERADDR", "INDRECT", "GETLIST", "GETSTAT"
  };

#define TABSTOP 8

  if (argc >= 1)
    {
      host = argv[0];
      client = clnt_rpcbind_create (host, RPCBVERS4, NULL);
    }
  else
    client = local_rpcb (PMAPPROG, RPCBVERS4);
  if (client == (CLIENT *) NULL)
    {
      clnt_pcreateerror ("rpcinfo: can't contact rpcbind");
      exit (1);
    }
  minutetimeout.tv_sec = 60;
  minutetimeout.tv_usec = 0;
  memset ((char *) &inf, 0, sizeof (rpcb_stat_byvers));
  if (CLNT_CALL (client, RPCBPROC_GETSTAT, (xdrproc_t) xdr_void, NULL,
                 (xdrproc_t) xdr_rpcb_stat_byvers, (char *) &inf,
                 minutetimeout) != RPC_SUCCESS)
    {
      clnt_perror (client, "rpcinfo: can't contact rpcbind: ");
      exit (1);
    }
  printf ("PORTMAP (version 2) statistics\n");
  lp = linebuf;
  for (i = 0; i <= rpcb_highproc_2; i++)
    {
      fieldbuf[0] = '\0';
      switch (i)
        {
        case PMAPPROC_SET:
          sprintf (fieldbuf, "%d/", inf[RPCBVERS_2_STAT].setinfo);
          break;
        case PMAPPROC_UNSET:
          sprintf (fieldbuf, "%d/", inf[RPCBVERS_2_STAT].unsetinfo);
          break;
        case PMAPPROC_GETPORT:
          cnt = 0;
          for (pa = inf[RPCBVERS_2_STAT].addrinfo; pa; pa = pa->next)
            cnt += pa->success;
          sprintf (fieldbuf, "%d/", cnt);
          break;
        case PMAPPROC_CALLIT:
          cnt = 0;
          for (pr = inf[RPCBVERS_2_STAT].rmtinfo; pr; pr = pr->next)
            cnt += pr->success;
          sprintf (fieldbuf, "%d/", cnt);
          break;
        default:
          break;                /* For the remaining ones */
        }
      cp = &fieldbuf[0] + strlen (fieldbuf);
      sprintf (cp, "%d", inf[RPCBVERS_2_STAT].info[i]);
      flen = strlen (fieldbuf);
      printf ("%s%s", pmaphdr[i],
              spaces ((TABSTOP * (1 + flen / TABSTOP))
                      - strlen (pmaphdr[i])));
      sprintf (lp, "%s%s", fieldbuf,
               spaces (cnt = ((TABSTOP * (1 + flen / TABSTOP)) - flen)));
      lp += (flen + cnt);
    }
  printf ("\n%s\n\n", linebuf);

  if (inf[RPCBVERS_2_STAT].info[PMAPPROC_CALLIT])
    {
      printf ("PMAP_RMTCALL call statistics\n");
      print_rmtcallstat (RPCBVERS_2_STAT, &inf[RPCBVERS_2_STAT]);
      printf ("\n");
    }

  if (inf[RPCBVERS_2_STAT].info[PMAPPROC_GETPORT])
    {
      printf ("PMAP_GETPORT call statistics\n");
      print_getaddrstat (RPCBVERS_2_STAT, &inf[RPCBVERS_2_STAT]);
      printf ("\n");
    }

  printf ("RPCBIND (version 3) statistics\n");
  lp = linebuf;
  for (i = 0; i <= rpcb_highproc_3; i++)
    {
      fieldbuf[0] = '\0';
```

```
    switch (i)
      {
      case RPCBPROC_SET:
        sprintf (fieldbuf, "%d/", inf[RPCBVERS_3_STAT].setinfo);
        break;
      case RPCBPROC_UNSET:
        sprintf (fieldbuf, "%d/", inf[RPCBVERS_3_STAT].unsetinfo);
        break;
      case RPCBPROC_GETADDR:
        cnt = 0;
        for (pa = inf[RPCBVERS_3_STAT].addrinfo; pa; pa = pa->next)
          cnt += pa->success;
        sprintf (fieldbuf, "%d/", cnt);
        break;
      case RPCBPROC_CALLIT:
        cnt = 0;
        for (pr = inf[RPCBVERS_3_STAT].rmtinfo; pr; pr = pr->next)
          cnt += pr->success;
        sprintf (fieldbuf, "%d/", cnt);
        break;
      default:
        break;                /* For the remaining ones */
      }
    cp = &fieldbuf[0] + strlen (fieldbuf);
    sprintf (cp, "%d", inf[RPCBVERS_3_STAT].info[i]);
    flen = strlen (fieldbuf);
    printf ("%s%s", rpcb3hdr[i],
            spaces ((TABSTOP * (1 + flen / TABSTOP))
                    - strlen (rpcb3hdr[i])));
    sprintf (lp, "%s%s", fieldbuf,
             spaces (cnt = ((TABSTOP * (1 + flen / TABSTOP)) - flen)));
    lp += (flen + cnt);
  }
printf ("\n%s\n\n", linebuf);

if (inf[RPCBVERS_3_STAT].info[RPCBPROC_CALLIT])
  {
    printf ("RPCB_RMTCALL (version 3) call statistics\n");
    print_rmtcallstat (RPCBVERS_3_STAT, &inf[RPCBVERS_3_STAT]);
    printf ("\n");
  }

if (inf[RPCBVERS_3_STAT].info[RPCBPROC_GETADDR])
  {
    printf ("RPCB_GETADDR (version 3) call statistics\n");
    print_getaddrstat (RPCBVERS_3_STAT, &inf[RPCBVERS_3_STAT]);
    printf ("\n");
  }

printf ("RPCBIND (version 4) statistics\n");

for (j = 0; j <= 9; j += 9)
  {                           /* Just two iterations for printing */
    lp = linebuf;
    for (i = j; i <= max (8, rpcb_highproc_4 - 9 + j); i++)
      {
        fieldbuf[0] = '\0';
        switch (i)
          {
          case RPCBPROC_SET:
            sprintf (fieldbuf, "%d/", inf[RPCBVERS_4_STAT].setinfo);
            break;
          case RPCBPROC_UNSET:
            sprintf (fieldbuf, "%d/", inf[RPCBVERS_4_STAT].unsetinfo);
            break;
          case RPCBPROC_GETADDR:
            cnt = 0;
            for (pa = inf[RPCBVERS_4_STAT].addrinfo; pa; pa = pa->next)
              cnt += pa->success;
            sprintf (fieldbuf, "%d/", cnt);
            break;
          case RPCBPROC_CALLIT:
            cnt = 0;
            for (pr = inf[RPCBVERS_4_STAT].rmtinfo; pr; pr = pr->next)
              cnt += pr->success;
            sprintf (fieldbuf, "%d/", cnt);
            break;
          default:
            break;            /* For the remaining ones */
          }
        cp = &fieldbuf[0] + strlen (fieldbuf);
        /*
         * XXX: We also add RPCBPROC_GETADDRLIST queries to
         * RPCB_GETADDR because rpcbind includes the
         * RPCB_GETADDRLIST successes in RPCB_GETADDR.
         */
        if (i != RPCBPROC_GETADDR)
          sprintf (cp, "%d", inf[RPCBVERS_4_STAT].info[i]);
        else
          sprintf (cp, "%d", inf[RPCBVERS_4_STAT].info[i] +
                   inf[RPCBVERS_4_STAT].info[RPCBPROC_GETADDRLIST]);
        flen = strlen (fieldbuf);
        printf ("%s%s", rpcb4hdr[i],
                spaces ((TABSTOP * (1 + flen / TABSTOP))
                        - strlen (rpcb4hdr[i])));
        sprintf (lp, "%s%s", fieldbuf,
                 spaces (cnt = ((TABSTOP * (1 + flen / TABSTOP)) - flen)));
        lp += (flen + cnt);
```

```
        }
      printf ("\n%s\n", linebuf);
    }

  if (inf[RPCBVERS_4_STAT].info[RPCBPROC_CALLIT] ||
      inf[RPCBVERS_4_STAT].info[RPCBPROC_INDIRECT])
    {
      printf ("\n");
      printf ("RPCB_RMTCALL (version 4) call statistics\n");
      print_rmtcallstat (RPCBVERS_4_STAT, &inf[RPCBVERS_4_STAT]);
    }

  if (inf[RPCBVERS_4_STAT].info[RPCBPROC_GETADDR])
    {
      printf ("\n");
      printf ("RPCB_GETADDR (version 4) call statistics\n");
      print_getaddrstat (RPCBVERS_4_STAT, &inf[RPCBVERS_4_STAT]);
    }
  clnt_destroy (client);
}

/*
 * Delete registeration for this (prog, vers, netid)
 */
static void
deletereg (netid, argc, argv)
     char *netid;
     int argc;
     char **argv;
{
  struct netconfig *nconf = NULL;

  if (argc != 2)
    {
      usage ();
      exit (1);
    }
  if (netid)
    {
      nconf = getnetconfigent (netid);
      if (nconf == NULL)
        {
          fprintf (stderr, "rpcinfo: netid %s not supported\n", netid);
          exit (1);
        }
    }
  if ((rpcb_unset (getprognum (argv[0]), getvers (argv[1]), nconf)) == 0)
    {
      fprintf (stderr,
               "rpcinfo: Could not delete registration for prog %s version %s\n",
               argv[0], argv[1]);
      exit (1);
    }
}

/*
 * Create and return a handle for the given nconf.
 * Exit if cannot create handle.
 */
static CLIENT *
clnt_addr_create (address, nconf, prog, vers)
     char *address;
     struct netconfig *nconf;
     u_long prog;
     u_long vers;
{
  CLIENT *client;
  static struct netbuf *nbuf;
  static int fd = RPC_ANYFD;

  if (fd == RPC_ANYFD)
    {
      if ((fd = __rpc_nconf2fd (nconf)) == -1)
        {
          rpc_createerr.cf_stat = RPC_TLIERROR;
          clnt_pcreateerror ("rpcinfo");
          exit (1);
        }
      /* Convert the uaddr to taddr */
      nbuf = uaddr2taddr (nconf, address);
      if (nbuf == NULL)
        {
          errx (1, "rpcinfo: no address for client handle");
          exit (1);
        }
    }
  client = clnt_tli_create (fd, nconf, nbuf, prog, vers, 0, 0);
  if (client == (CLIENT *) NULL)
    {
      clnt_pcreateerror ("rpcinfo");
      exit (1);
    }
  return (client);
}

/*
 * If the version number is given, ping that (prog, vers); else try to find
 * the version numbers supported for that prog and ping all the versions.
```

```
 * Remote rpcbind is not contacted for this service. The requests are
 * sent directly to the services themselves.
 */
static void
addrping (address, netid, argc, argv)
     char *address;
     char *netid;
     int argc;
     char **argv;
{
  CLIENT *client;
  struct timeval to;
  enum clnt_stat rpc_stat;
  u_int32_t prognum, versnum, minvers, maxvers;
  struct rpc_err rpcerr;
  int failure = 0;
  struct netconfig *nconf;
  int fd;

  if (argc < 1 || argc > 2 || (netid == NULL))
    {
      usage ();
      exit (1);
    }
  nconf = getnetconfigent (netid);
  if (nconf == (struct netconfig *) NULL)
    {
      fprintf (stderr, "rpcinfo: Could not find %s\n", netid);
      exit (1);
    }
  to.tv_sec = 10;
  to.tv_usec = 0;
  prognum = getprognum (argv[0]);
  if (argc == 1)
    {                           /* Version number not known */
      /*
       * A call to version 0 should fail with a program/version
       * mismatch, and give us the range of versions supported.
       */
      versnum = MIN_VERS;
    }
  else
    {
      versnum = getvers (argv[1]);
    }
  client = clnt_addr_create (address, nconf, prognum, versnum);
  rpc_stat = CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                        (char *) NULL, (xdrproc_t) xdr_void,
                        (char *) NULL, to);
  if (argc == 2)
    {
      /* Version number was known */
      if (pstatus (client, prognum, versnum) < 0)
        failure = 1;
      (void) CLNT_DESTROY (client);
      if (failure)
        exit (1);
      return;
    }
  /* Version number not known */
  (void) CLNT_CONTROL (client, CLSET_FD_NCLOSE, (char *) NULL);
  (void) CLNT_CONTROL (client, CLGET_FD, (char *) &fd);
  if (rpc_stat == RPC_PROGVERSMISMATCH)
    {
      clnt_geterr (client, &rpcerr);
      minvers = rpcerr.re_vers.low;
      maxvers = rpcerr.re_vers.high;
    }
  else if (rpc_stat == RPC_SUCCESS)
    {
      /*
       * Oh dear, it DOES support version 0.
       * Let's try version MAX_VERS.
       */
      (void) CLNT_DESTROY (client);
      client = clnt_addr_create (address, nconf, prognum, MAX_VERS);
      rpc_stat = CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                            (char *) NULL, (xdrproc_t) xdr_void,
                            (char *) NULL, to);
      if (rpc_stat == RPC_PROGVERSMISMATCH)
        {
          clnt_geterr (client, &rpcerr);
          minvers = rpcerr.re_vers.low;
          maxvers = rpcerr.re_vers.high;
        }
      else if (rpc_stat == RPC_SUCCESS)
        {
          /*
           * It also supports version MAX_VERS.
           * Looks like we have a wise guy.
           * OK, we give them information on all
           * 4 billion versions they support...
           */
          minvers = 0;
          maxvers = MAX_VERS;
        }
      else
        {
```

```
          (void) pstatus (client, prognum, MAX_VERS);
          exit (1);
        }
    }
  else
    {
      (void) pstatus (client, prognum, (u_long) 0);
      exit (1);
    }
  (void) CLNT_DESTROY (client);
  for (versnum = minvers; versnum <= maxvers; versnum++)
    {
      client = clnt_addr_create (address, nconf, prognum, versnum);
      rpc_stat = CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                            (char *) NULL, (xdrproc_t) xdr_void,
                            (char *) NULL, to);
      if (pstatus (client, prognum, versnum) < 0)
        failure = 1;
      (void) CLNT_DESTROY (client);
    }
  (void) close (fd);
  if (failure)
    exit (1);
  return;
}

/*
 * If the version number is given, ping that (prog, vers); else try to find
 * the version numbers supported for that prog and ping all the versions.
 * Remote rpcbind is *contacted* for this service. The requests are
 * then sent directly to the services themselves.
 */
static void
progping (netid, argc, argv)
     char *netid;
     int argc;
     char **argv;
{
  CLIENT *client;
  struct timeval to;
  enum clnt_stat rpc_stat;
  u_int32_t prognum, versnum, minvers, maxvers;
  struct rpc_err rpcerr;
  int failure = 0;
  struct netconfig *nconf;

  if (argc < 2 || argc > 3 || (netid == NULL))
    {
      usage ();
      exit (1);
    }
  prognum = getprognum (argv[1]);
  if (argc == 2)
    {                           /* Version number not known */
      /*
       * A call to version 0 should fail with a program/version
       * mismatch, and give us the range of versions supported.
       */
      versnum = MIN_VERS;
    }
  else
    {
      versnum = getvers (argv[2]);
    }
  if (netid)
    {
      nconf = getnetconfigent (netid);
      if (nconf == (struct netconfig *) NULL)
        {
          fprintf (stderr, "rpcinfo: Could not find %s\n", netid);
          exit (1);
        }
      client = clnt_tp_create (argv[0], prognum, versnum, nconf);
    }
  else
    {
      client = clnt_create (argv[0], prognum, versnum, "NETPATH");
    }
  if (client == (CLIENT *) NULL)
    {
      clnt_pcreateerror ("rpcinfo");
      exit (1);
    }
  to.tv_sec = 10;
  to.tv_usec = 0;
  rpc_stat = CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                        (char *) NULL, (xdrproc_t) xdr_void,
                        (char *) NULL, to);
  if (argc == 3)
    {
      /* Version number was known */
      if (pstatus (client, prognum, versnum) < 0)
        failure = 1;
      (void) CLNT_DESTROY (client);
      if (failure)
        exit (1);
      return;
    }
```

```
  /* Version number not known */
  if (rpc_stat == RPC_PROGVERSMISMATCH)
    {
      clnt_geterr (client, &rpcerr);
      minvers = rpcerr.re_vers.low;
      maxvers = rpcerr.re_vers.high;
    }
  else if (rpc_stat == RPC_SUCCESS)
    {
      /*
       * Oh dear, it DOES support version 0.
       * Let's try version MAX_VERS.
       */
      versnum = MAX_VERS;
      (void) CLNT_CONTROL (client, CLSET_VERS, (char *) &versnum);
      rpc_stat = CLNT_CALL (client, NULLPROC,
                            (xdrproc_t) xdr_void, (char *) NULL,
                            (xdrproc_t) xdr_void, (char *) NULL, to);
      if (rpc_stat == RPC_PROGVERSMISMATCH)
        {
          clnt_geterr (client, &rpcerr);
          minvers = rpcerr.re_vers.low;
          maxvers = rpcerr.re_vers.high;
        }
      else if (rpc_stat == RPC_SUCCESS)
        {
          /*
           * It also supports version MAX_VERS.
           * Looks like we have a wise guy.
           * OK, we give them information on all
           * 4 billion versions they support...
           */
          minvers = 0;
          maxvers = MAX_VERS;
        }
      else
        {
          (void) pstatus (client, prognum, MAX_VERS);
          exit (1);
        }
    }
  else
    {
      (void) pstatus (client, prognum, (u_long) 0);
      exit (1);
    }
  for (versnum = minvers; versnum <= maxvers; versnum++)
    {
      (void) CLNT_CONTROL (client, CLSET_VERS, (char *) &versnum);
      rpc_stat = CLNT_CALL (client, NULLPROC, (xdrproc_t) xdr_void,
                            (char *) NULL, (xdrproc_t) xdr_void,
                            (char *) NULL, to);
      if (pstatus (client, prognum, versnum) < 0)
        failure = 1;
    }
  (void) CLNT_DESTROY (client);
  if (failure)
    exit (1);
  return;
}

static void
usage ()
{
  fprintf (stderr, "Usage: rpcinfo [-m | -s] [host]\n");
#ifdef PORTMAP
  fprintf (stderr, "       rpcinfo -p [host]\n");
#endif
  fprintf (stderr, "       rpcinfo -T netid host prognum [versnum]\n");
  fprintf (stderr, "       rpcinfo -l host prognum versnum\n");
#ifdef PORTMAP
  fprintf (stderr,
           "       rpcinfo [-n portnum] -u | -t host prognum [versnum]\n");
#endif
  fprintf (stderr,
           "       rpcinfo -a serv_address -T netid prognum [version]\n");
  fprintf (stderr, "       rpcinfo -b prognum versnum\n");
  fprintf (stderr, "       rpcinfo -d [-T netid] prognum versnum\n");
}

static u_long
getprognum (arg)
     char *arg;
{
  char *strptr;
  register struct rpcent *rpc;
  register u_long prognum;
  char *tptr = arg;

  while (*tptr && isdigit (*tptr++));
  if (*tptr || isalpha (*(tptr - 1)))
    {
      rpc = getrpcbyname (arg);
      if (rpc == NULL)
        {
          fprintf (stderr, "rpcinfo: %s is unknown service\n", arg);
          exit (1);
        }
```

```
      prognum = rpc->r_number;
    }
  else
    {
      prognum = strtol (arg, &strptr, 10);
      if (strptr == arg || *strptr != '\0')
        {
          fprintf (stderr, "rpcinfo: %s is illegal program number\n", arg);
          exit (1);
        }
    }
  return (prognum);
}

static u_long
getvers (arg)
     char *arg;
{
  char *strptr;
  register u_long vers;

  vers = (int) strtol (arg, &strptr, 10);
  if (strptr == arg || *strptr != '\0')
    {
      fprintf (stderr, "rpcinfo: %s is illegal version number\n", arg);
      exit (1);
    }
  return (vers);
}

/*
 * This routine should take a pointer to an "rpc_err" structure, rather than
 * a pointer to a CLIENT structure, but "clnt_perror" takes a pointer to
 * a CLIENT structure rather than a pointer to an "rpc_err" structure.
 * As such, we have to keep the CLIENT structure around in order to print
 * a good error message.
 */
static int
pstatus (client, prog, vers)
     register CLIENT *client;
     u_long prog;
     u_long vers;
{
  struct rpc_err rpcerr;

  clnt_geterr (client, &rpcerr);
  if (rpcerr.re_status != RPC_SUCCESS)
    {
      clnt_perror (client, "rpcinfo");
      printf ("program %lu version %lu is not available\n", prog, vers);
      return (-1);
    }
  else
    {
      printf ("program %lu version %lu ready and waiting\n", prog, vers);
      return (0);
    }
}

static CLIENT *
clnt_rpcbind_create (host, rpcbversnum, targaddr)
     char *host;
     int rpcbversnum;
     struct netbuf **targaddr;
{
  static char *tlist[3] = {
    "circuit_n", "circuit_v", "datagram_v"
  };
  int i;
  struct netconfig *nconf;
  CLIENT *clnt = NULL;
  void *handle;

  rpc_createerr.cf_stat = RPC_SUCCESS;
  for (i = 0; i < 3; i++)
    {
      if ((handle = __rpc_setconf (tlist[i])) == NULL)
        continue;
      while (clnt == (CLIENT *) NULL)
        {
          if ((nconf = __rpc_getconf (handle)) == NULL)
            {
              if (rpc_createerr.cf_stat == RPC_SUCCESS)
                rpc_createerr.cf_stat = RPC_UNKNOWNPROTO;
              break;
            }
          clnt = getclnthandle (host, nconf, rpcbversnum, targaddr);
        }
      if (clnt)
        break;
      __rpc_endconf (handle);
    }
  return (clnt);
}

static CLIENT *
getclnthandle (host, nconf, rpcbversnum, targaddr)
     char *host;
```

```
     struct netconfig *nconf;
     u_long rpcbversnum;
     struct netbuf **targaddr;
{
  struct netbuf addr;
  struct addrinfo hints, *res;
  CLIENT *client = NULL;

  /* Get the address of the rpcbind */
  memset (&hints, 0, sizeof hints);
  if ((getaddrinfo (host, "rpcbind", &hints, &res) != 0) &&
      (getaddrinfo (host, "portmapper",&hints, &res) != 0))
    {
      rpc_createerr.cf_stat = RPC_N2AXLATEFAILURE;
      return (NULL);
    }
  addr.len = addr.maxlen = res->ai_addrlen;
  addr.buf = res->ai_addr;
  client = clnt_tli_create (RPC_ANYFD, nconf, &addr, RPCBPROG,
                            rpcbversnum, 0, 0);
  if (client)
    {
      if (targaddr != NULL)
        {
          *targaddr = (struct netbuf *) malloc (sizeof (struct netbuf));
          if (*targaddr != NULL)
            {
              (*targaddr)->maxlen = addr.maxlen;
              (*targaddr)->len = addr.len;
              (*targaddr)->buf = (char *) malloc (addr.len);
              if ((*targaddr)->buf != NULL)
                {
                  memcpy ((*targaddr)->buf, addr.buf, addr.len);
                }
            }
        }
    }
  else
    {
      if (rpc_createerr.cf_stat == RPC_TLIERROR)
        {
          /*
           * Assume that the other system is dead; this is a
           * better error to display to the user.
           */
          rpc_createerr.cf_stat = RPC_RPCBFAILURE;
          rpc_createerr.cf_error.re_status = RPC_FAILED;
        }
    }
  freeaddrinfo (res);
  return (client);
}

static void
print_rmtcallstat (rtype, infp)
     int rtype;
     rpcb_stat *infp;
{
  register rpcbs_rmtcalllist_ptr pr;
  struct rpcent *rpc;

  if (rtype == RPCBVERS_4_STAT)
    printf ("prog\t\tvers\tproc\tnetid\tindirect success failure\n");
  else
    printf ("prog\t\tvers\tproc\tnetid\tsuccess\tfailure\n");
  for (pr = infp->rmtinfo; pr; pr = pr->next)
    {
      rpc = getrpcbynumber (pr->prog);
      if (rpc)
        printf ("%-16s", rpc->r_name);
      else
        printf ("%-16d", pr->prog);
      printf ("%d\t%d\t%s\t", pr->vers, pr->proc, pr->netid);
      if (rtype == RPCBVERS_4_STAT)
        printf ("%d\t ", pr->indirect);
      printf ("%d\t%d\n", pr->success, pr->failure);
    }
}

static void
print_getaddrstat (rtype, infp)
     int rtype;
     rpcb_stat *infp;
{
  rpcbs_addrlist_ptr al;
  register struct rpcent *rpc;

  printf ("prog\t\tvers\tnetid\t  success\tfailure\n");
  for (al = infp->addrinfo; al; al = al->next)
    {
      rpc = getrpcbynumber (al->prog);
      if (rpc)
        printf ("%-16s", rpc->r_name);
      else
        printf ("%-16d", al->prog);
      printf ("%d\t%s\t  %-12d\t%d\n",
              al->vers, al->netid, al->success, al->failure);
    }
```

```
}

static char *
spaces (howmany)
     int howmany;
{
  static char space_array[] =   /* 64 spaces */
    "                                                                ";

  if (howmany <= 0 || howmany > sizeof (space_array))
    {
      return ("");
    }
  return (&space_array[sizeof (space_array) - howmany - 1]);
}
```

Notice for package(s)

hdparm

```
BSD-Style Open Source License:

You may freely use, modify, and redistribute the hdparm program,
as either binary or source, or both.

The only condition is that my name and copyright notice
remain in the source code as-is.

Mark Lord (mlord@pobox.com)
```

Notice for package(s)

hdparm

```
This package was debianized by Christopher L Cheney <ccheney@debian.org> on
Wed, 21 Nov 2001 15:51:14 -0600.

It was downloaded from http://www.ibiblio.org/pub/Linux/system/hardware

Upstream Author: Mark S. Lord <mlord@pobox.com>

Copyright:

/* hdparm.c - Command line interface to get/set hard disk parameters */
/*          - by Mark Lord © 1994-2008 -- freely distributable */

You are free to distribute this software under the terms of the BSD License.
On Debian systems, the complete text of the BSD License can be found in
/usr/share/common-licenses/BSD
```

Notice for package(s)

hdparm

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
```

have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author

```
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

hdparm

```
#!/bin/bash
#
# SATA SSD free-space TRIM utility, by Mark Lord <mlord@pobox.com>

VERSION=3.6

# Copyright (C) 2009-2010 Mark Lord.  All rights reserved.
#
# Contains hfsplus and ntfs code contributed by Heiko Wegeler <heiko.wegeler@googlemail.com>.
# Package sleuthkit version >=3.1.1 is required for HFS+. Package ntfs-3g and ntfsprogs is required for NTFS.
#
# Requires gawk, a really-recent hdparm, and various other programs.
# This needs to be redone entirely in C, for 64-bit math, someday.
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License Version 2,
# as published by the Free Software Foundation.
#
# This program is distributed in the hope that it would be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software Foundation,
# Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#
# Note for OCZ Vertex-LE users:  the drive firmware will error when
# attempting to trim the final sector of the drive.  To avoid this,
# partition the drive such that the final sector is not used.

export LANG=C

## The usual terse usage information:
##
function usage_error(){
        echo >&2
        echo "Linux tune-up (TRIM) utility for SATA SSDs"
        echo "Usage:  $0 [--verbose] [--commit] <mount_point|block_device>" >&2
        echo "   Eg:  $0 /dev/sda1" >&2
        echo >&2
        exit 1
}

## Parameter parsing for the main script.
## Yeah, we could use getopt here instead, but what fun would that be?
##

echo
echo "${0##*/}: Linux SATA SSD TRIM utility, version $VERSION, by Mark Lord."

export verbose=0
commit=""
destroy_me=""
argc=$#
arg=""
while [ $argc -gt 0 ]; do
        if [ "$1" = "--commit" ]; then
                commit=yes
        elif [ "$1" = "--please-prematurely-wear-out-my-ssd" ]; then
                destroy_me=yes
        elif [ "$1" = "--verbose" ]; then
```

```
                verbose=$((verbose + 1))
        elif [ "$1" = "" ]; then
                usage_error
        else
                if [ "$arg" != "" ]; then
                        echo "$1: too many arguments, aborting." >&2
                        exit 1
                fi
                arg="$1"
        fi
        argc=$((argc - 1))
        shift
done
[ "$arg" = "" ] && usage_error

## Find a required program, or else give a nicer error message than we'd otherwise see:
##
function find_prog(){
        prog="$1"
        if [ ! -x "$prog" ]; then
                prog="${prog##*/}"
                p=`type -f -P "$prog" 2>/dev/null`
                if [ "$p" = "" ]; then
                        [ "$2" != "quiet" ] && echo "$1: needed but not found, aborting." >&2
                        exit 1
                fi
                prog="$p"
                [ $verbose -gt 0 ] && echo "  --> using $prog instead of $1" >&2
        fi
        echo "$prog"
}

## Ensure we have most of the necessary utilities available before trying to proceed:
##
hash -r  ## Refresh bash's cached PATH entries
HDPARM=`find_prog /sbin/hdparm` || exit 1
FIND=`find_prog /usr/bin/find`  || exit 1
STAT=`find_prog /usr/bin/stat`  || exit 1
GAWK=`find_prog /usr/bin/gawk`  || exit 1
BLKID=`find_prog /sbin/blkid`   || exit 1
GREP=`find_prog /bin/grep`      || exit 1
ID=`find_prog /usr/bin/id`      || exit 1
LS=`find_prog /bin/ls`          || exit 1
DF=`find_prog /bin/df`          || exit 1
RM=`find_prog /bin/rm`          || exit 1
STAT=`find_prog /usr/bin/stat`  || exit 1

[ $verbose -gt 1 ] && HDPARM="$HDPARM --verbose"

## I suppose this will confuse the three SELinux users out there:
##
if [ `$ID -u` -ne 0 ]; then
        echo "Only the super-user can use this (try \"sudo $0\" instead), aborting." >&2
        exit 1
fi

## We need a very modern hdparm, for its --fallocate and --trim-sector-ranges-stdin flags:
## Version 9.25 added automatic determination of safe max-size of TRIM commands.
##
HDPVER=`$HDPARM -V | $GAWK '{gsub("[^0-9.]","",$2); if ($2 > 0) print ($2 * 100); else print 0; exit(0)}'`
if [ $HDPVER -lt 925 ]; then
        echo "$HDPARM: version >= 9.25 is required, aborting." >&2
        exit 1
fi

## Convert relative path "$1" into an absolute pathname, resolving all symlinks:
##
function get_realpath(){
        iter=0
        p="$1"
        while [ -e "$p" -a $iter -lt 100 ]; do
                ## Strip trailing slashes:
                while [ "$p" != "/" -a "$p" != "${p%%/}" ]; do
                        p="${p%%/}"
                done
                ## Split into directory:leaf portions:
                d="${p%/*}"
                t="${p##*/}"
                ## If the split worked, then cd into the directory portion:
                if [ "$d" != "" -a "$d" != "$p" ]; then
                        cd -P "$d" || exit
                        p="$t"
                fi
                ## If what we have left is a directory, then cd to it and print realpath:
                if [ -d "$p" ]; then
                        cd -P "$p" || exit
                        pwd -P
                        exit
                ## Otherwise if it is a symlink, read the link and loop again:
                elif [ -h "$p" ]; then
                        p="`$LS -ld "$p" | awk '{sub("^[^>]*->[>] *",""); print}'`"
                ## Otherwise, prefix $p with the cwd path and print it:
                elif [ -e "$p" ]; then
                        [ "${p:0:1}" = "/" ] || p="`pwd -P`/$p"
                        echo "$p"
                        exit
                fi
                iter=$((iter + 1))
```

```
        done
}

function get_devpath(){
        dir="$1"
        kdev=`$STAT --format="%04D" "$dir" 2>/dev/null`
        [ "$kdev" = "" ] && exit 1
        major=$((0x${kdev:0:2}))
        minor=$((0x${kdev:2:2}))
        $FIND /dev -xdev -type b -exec $LS -ln {} \; | $GAWK -v major="$major," -v minor="$minor" \
                '($5 == major && $6 == minor){r=$NF}END{print r}'
}

## Convert "$arg" into an absolute pathname target, with no symlinks or embedded blanks:
target="`get_realpath "$arg"`"
if [ "$target" = "" ]; then
        [ "$arg" = "/dev/root" ] && target="`get_devpath /`"
        if [ "$target" = "" ]; then
                echo "$arg: unable to determine full pathname, aborting." >&2
                exit 1
        fi
fi
if [ "$target" != "${target##* }" ]; then
        echo "\"$target\": pathname has embedded blanks, aborting." >&2
        exit 1
fi

## Take a first cut at online/offline determination, based on the target:
##
if [ -d "$target" ]; then
        method=online
elif [ -b "$target" ]; then
        method=offline
else
        echo "$target: not a block device or mount point, aborting." >&2
        exit 1
fi

## Find the active mount-point (fsdir) associated with a device ($1: fsdev).
## This is complicated, and probably still buggy, because a single
## device can show up under *multiple* mount points in /proc/mounts.
##
function get_fsdir(){
        rw=""
        r=""
        while read -a m ; do
                pdev="${m[0]}"
                [ "$pdev" = "$1" ] || pdev="`get_realpath "$pdev"`"
                if [ "$pdev" = "$1" ]; then
                        if [ "$rw" != "rw" ]; then
                                rw="${m[3]:0:2}"
                                r="${m[1]}"
                        fi
                fi
                #echo "$pdev ${m[1]} ${m[2]} ${m[3]}"
        done
        echo -n "$r"
}

## Find the device (fsdev) associated with a mount point ($1: fsdir).
## Since mounts can be stacked on top of each other, we return the
## one from the last occurance in the list from /proc/mounts.
##
function get_fsdev(){   ## from fsdir
        get_realpath "`$GAWK -v p="$1" '{if ($2 == p) r=$1} END{print r}' < /proc/mounts`"
}

## Find the r/w or r/o status (fsmode) of a filesystem mount point  ($1: fsdir)
## We get it from the last occurance of the mount point in the list from /proc/mounts,
## and convert it to a longer human-readable string.
##
function get_fsmode(){  ## from fsdir
        mode="`$GAWK -v p="$1" '{if ($2 == p) r=substr($4,1,2)} END{print r}' < /proc/mounts`"
        if [ "$mode" = "ro" ]; then
                echo "read-only"
        elif [ "$mode" = "rw" ]; then
                echo "read-write"
        else
                echo "$fsdir: unable to determine mount status, aborting." >&2
                exit 1
        fi
}

## Try and determine the device name associated with the root filesystem.
## This is nearly impossible to do in any perfect fashion.
##
## Redhat/Fedora no longer have an rdev command.  Silly them.
## So we now implement it internally, below.
##
## match_rootdev *should* work, but on some distros it may find only "/dev/root",
## and "/dev/root" is not usually a real device.  We leave it like that for now,
## because that's the pattern such systems also use in /proc/mounts.
## Later, at time of use, we'll try harder to find the real rootdev.
##
## FIXME: apparently this doesn't work on SuSE Linux, though.
## So for there, we'll likely need to read /etc/mtab,
## or be a lot more clever and get it somehow from statfs or something.
## FIXME: or use target from /dev/root symlink for Gentoo as well.
```

```
##
function match_rootdev() {
        rdev=""
        rdevno="$1"
        while read bdev ; do
                if [ "$rdev" = "" -o "$bdev" != "/dev/root" ]; then
                        devno=$($STAT -c "0x%t%02T" "$bdev" 2>/dev/null)
                        [ "$devno" = "$rdevno" ] && rdev="$bdev"
                fi
        done
        echo -n "$rdev"
}

rootdev=$($FIND /dev/ -type b 2>/dev/null | match_rootdev $($STAT -c "0x%D" '/'))
[ $verbose -gt 0 ] && echo "rootdev=$rootdev"

## The user gave us a directory (mount point) to TRIM,
## which implies that we will be doing an online TRIM
## using --fallocate and --fibmap to find the free extents.
## Do some preliminary correctness/feasibility checks on fsdir:
##
if [ "$method" = "online" ]; then
        ## Ensure fsdir exists and is accessible to us:
        fsdir="$target"
        cd "$fsdir" || exit 1

        if [ "$fsdir" = "/" ]; then
                fsdev="$rootdev"
        else
                ## Figure out what device holds the filesystem.
                fsdev="`get_fsdev $fsdir`"
                if [ "$fsdev" = "" ]; then
                        echo "$fsdir: not found in /proc/mounts, aborting." >&2
                        exit 1
                fi
        fi

        ## The root filesystem may show up as the phoney "/dev/root" device
        ## in /proc/mounts (ugh).  So if we see that, then substitute the rootdev
        ## that $DF gave us earlier.  But $DF may have the same problem (double ugh).
        ##
        [ ! -e "$fsdev" -a "$fsdev" = "/dev/root" ] && fsdev="$rootdev"

        ## Ensure that fsdev exists and is a block device:
        if [ ! -e "$fsdev" ]; then
                if [ "$fsdev" != "/dev/root" ]; then
                        echo "$fsdev: not found" >&2
                        exit 1
                fi
                if [ "$rootdev" = "" ]; then
                        echo "$fsdev: not found" >&2
                        exit 1
                fi
                fsdev="$rootdev"
        fi
        if [ ! -b "$fsdev" ]; then
                echo "$fsdev: not a block device" >&2
                exit 1
        fi

        ## If it is mounted read-only, we must switch to doing an "offline" trim of fsdev:
        fsmode="`get_fsmode $fsdir`" || exit 1
        [ $verbose -gt 0 ] && echo "fsmode1: fsmode=$fsmode"
        [ "$fsmode" = "read-only" ] && method=offline
fi

## This is not an "else" clause from the above, because "method" may have changed.
## For offline TRIM, we need the block device, and it cannot be mounted read-write:
##
if [ "$method" = "offline" ]; then
        ## We might already have fsdev/fsdir from above; if not, we need to find them.
        if [ "$fsdev" = "" -o "$fsdir" = "" ]; then
                fsdev="$target"
                fsdir="`get_fsdir "$fsdev" < /proc/mounts`"
                ## More weirdness for /dev/root in /proc/mounts:
                if [ "$fsdir" = "" -a "$fsdev" = "$rootdev" ]; then
                        fsdir="`get_fsdir /dev/root < /proc/mounts`"
                        if [ "$fsdir" = "" ]; then
                                rdev="`get_devpath /`"
                                [ "$rdev" != "" ] && fsdir="`get_fsdir "$rdev" < /proc/mounts`"
                        fi
                fi
        fi

        ## If the filesystem is truly not-mounted, then fsdir will still be empty here.
        ## It could be mounted, though.  Read-only is fine, but read-write means we need
        ## to switch gears and do an "online" TRIM instead of an "offline" TRIM.
        ##
        if [ "$fsdir" != "" ]; then
                fsmode="`get_fsmode $fsdir`" || exit 1
                [ $verbose -gt 0 ] && echo "fsmode2: fsmode=$fsmode"
                if [ "$fsmode" = "read-write" ]; then
                        method=online
                        cd "$fsdir" || exit 1
                fi
        fi
fi
```

```
## Use $LS to find the major number of a block device:
##
function get_major(){
        $LS -ln "$1" | $GAWK '{print gensub(",","",1,$5)}'
}

## At this point, we have finalized our selection of online vs. offline,
## and we definitely know the fsdev, as well as the fsdir (fsdir="" if not-mounted).
##
## Now guess at the underlying rawdev name, which could be exactly the same as fsdev.
## Then determine whether or not rawdev claims support for TRIM commands.
## Note that some devices lie about support, and later reject the TRIM commands.
##
rawdev=`echo $fsdev | $GAWK '{print gensub("[0-9]*$","","g")}'`
rawdev="`get_realpath "$rawdev"`"
if [ ! -e "$rawdev" ]; then
        rawdev=""
elif [ ! -b "$rawdev" ]; then
        rawdev=""
elif [ "`get_major $fsdev`" -ne "`get_major $rawdev`" ]; then  ## sanity check
        rawdev=""
else
        ## "SCSI" drives only; no LVM confusion for now:
        maj="$(get_major $fsdev)"
        maj_ok=0
        for scsi_major in 8 65 66 67 68 69 70 71 ; do
                [ "$maj" = "$scsi_major" ] && maj_ok=1
        done
        if [ $maj_ok -eq 0 ]; then
                echo "$rawdev: does not appear to be a SCSI/SATA SSD, aborting." >&2
                exit 1
        fi
        if ! $HDPARM -I $rawdev | $GREP -i '[    ][*][    ]*Data Set Management TRIM supported' &>/dev/null ; then
                if [ "$commit" = "yes" ]; then
                        echo "$rawdev: DSM/TRIM command not supported, aborting." >&2
                        exit 1
                fi
                echo "$rawdev: DSM/TRIM command not supported (continuing with dry-run)." >&2
        fi
fi
if [ "$rawdev" = "" ]; then
        echo "$fsdev: unable to reliably determine the underlying physical device name, aborting" >&2
        exit 1
fi

## We also need to know the offset of fsdev from the beginning of rawdev,
## because TRIM requires absolute sector numbers within rawdev:
##
fsoffset=`$HDPARM -g "$fsdev" | $GAWK 'END {print $NF}'`

## Next step is to determine what type of filesystem we are dealing with (fstype):
##
if [ "$fsdir" = "" ]; then
        ## Not mounted: use $BLKID to determine the fstype of fsdev:
        fstype=`$BLKID -w /dev/null -c /dev/null $fsdev 2>/dev/null | \
                 $GAWK '/ TYPE=".*"/{sub("^.* TYPE=\"",""); sub("[\" ][\" ]*.*$",""); print}'`
        [ $verbose -gt 0 ] && echo "$fsdev: fstype=$fstype"
else
        ## Mounted: we could just use $BLKID here, too, but it's safer to use /proc/mounts directly:
        fstype="`$GAWK -v p="$fsdir" '{if ($2 == p) r=$3} END{print r}' < /proc/mounts`"
        [ $verbose -gt 0 ] && echo "$fsdir: fstype=$fstype"
fi
if [ "$fstype" = "" ]; then
        echo "$fsdev: unable to determine filesystem type, aborting." >&2
        exit 1
fi

## Some helper funcs and vars for use with the xfs filesystem tools:
##
function xfs_abort(){
        echo "$fsdev: unable to determine xfs filesystem ${1-parameters}, aborting." >&2
        exit 1
}
function xfs_trimlist(){
        $XFS_DB -r -c "freesp -d" "$fsdev"  ## couldn't get this to work inline
}
xfs_agoffsets=""
xfs_blksects=0

## We used to allow single-drive btrfs here, but it stopped working in linux-2.6.31,
## and Chris Mason says "unsafe at any speed" really.  So it's been dropped now.
##
if [ "$fstype" = "btrfs" ]; then  ## hdparm --fibmap fails, due to fake 0:xx device nodes
        echo "$target: btrfs filesystem type not supported (cannot determine physical devices), aborting." >&2
        exit 1
fi

## Now figure out whether we can actually do TRIM on this type of filesystem:
##
if [ "$method" = "online" ]; then
        ## Print sensible error messages for some common situations,
        ## rather than failing with more confusing messages later on..
        ##
        if [ "$fstype" = "ext2" -o "$fstype" = "ext3" ]; then  ## No --fallocate support
                echo "$target: cannot TRIM $fstype filesystem when mounted read-write, aborting." >&2
                exit 1
        fi
```

```
        ## Figure out if we have enough free space to even attempt TRIM:
        ##
        freesize=`$DF -P -B 1024 . | $GAWK '{r=$4}END{print r}'`
        if [ "$freesize" = "" ]; then
                echo "$fsdev: unknown to '$DF'"
                exit 1
        fi
        if [ $freesize -lt 15000 ]; then
                echo "$target: filesystem too full for TRIM, aborting." >&2
                exit 1
        fi

        ## Figure out how much space to --fallocate (later), keeping in mind
        ## that this is a live filesystem, and we need to leave some space for
        ## other concurrent activities, as well as for filesystem overhead (metadata).
        ## So, reserve at least 1% or 7500 KB, whichever is larger:
        ##
        reserved=$((freesize / 100))
        [ $reserved -lt 7500 ] && reserved=7500
        [ $verbose -gt 0 ] && echo "freesize = ${freesize} KB, reserved = ${reserved} KB"
        tmpsize=$((freesize - reserved))
        tmpfile="WIPER_TMPFILE.$$"
        get_trimlist="$HDPARM --fibmap $tmpfile"
else
        ## We can only do offline TRIM on filesystems that we "know" about here.
        ## Currently, this includes the ext2/3/4 family, xfs, and reiserfs.
        ## The first step for any of these is to ensure that the filesystem is "clean",
        ## and immediately abort if it is not.
        ##
        get_trimlist=""
        if [ "$fstype" = "ext2" -o "$fstype" = "ext3" -o "$fstype" = "ext4" ]; then
                DUMPE2FS=`find_prog /sbin/dumpe2fs` || exit 1
                fstate="`$DUMPE2FS $fsdev 2>/dev/null | $GAWK '/^[Ff]ilesystem state:/{print $NF}' 2>/dev/null`"
                if [ "$fstate" != "clean" ]; then
                        echo "$target: filesystem not clean, please run \"e2fsck $fsdev\" first, aborting." >&2
                        exit 1
                fi
                get_trimlist="$DUMPE2FS $fsdev"
        elif [ "$fstype" = "xfs" ]; then
                XFS_DB=`find_prog /sbin/xfs_db` || exit 1
                XFS_REPAIR=`find_prog /sbin/xfs_repair` || exit 1
                if ! $XFS_REPAIR -n "$fsdev" &>/dev/null ; then
                        echo "$fsdev: filesystem not clean, please run \"xfs_repair $fsdev\" first, aborting." >&2
                        exit 1
                fi

                ## For xfs, life is more complex than with ext2/3/4 above.
                ## The $XFS_DB tool does not return absolute block numbers for freespace,
                ## but rather gives them as relative to it's allocation groups (ag's).
                ## So, we'll need to interogate it for the offset of each ag within the filesystem.
                ## The agoffsets are extracted from $XFS_DB as sector offsets within the fsdev.
                ##
                agcount=`$XFS_DB -r -c "sb" -c "print agcount" "$fsdev" | $GAWK '{print 0 + $NF}'`
                [ "$agcount" = "" -o "$agcount" = "0" ] && xfs_abort "agcount"
                xfs_agoffsets=
                i=0
                while [ $i -lt $agcount ]; do
                        agoffset=`$XFS_DB -r -c "sb" -c "convert agno $i daddr" "$fsdev" \
                                | $GAWK '{print 0 + gensub("[( )]","","g",$2)}'`
                        [ "$agoffset" = "" ] && xfs_abort "agoffset-$i"
                        [ $i -gt 0 ] && [ $agoffset -le ${xfs_agoffsets##* } ] && xfs_abort "agoffset[$i]"
                        xfs_agoffsets="$xfs_agoffsets $agoffset"
                        i=$((i + 1))
                done
                xfs_agoffsets="${xfs_agoffsets:1}"      ## strip leading space

                ## We also need xfs_blksects for later, because freespace gets listed as block numbers.
                ##
                blksize=`$XFS_DB -r -c "sb" -c "print blocksize" "$fsdev" | $GAWK '{print 0 + $NF}'`
                [ "$blksize" = "" -o "$blksize" = "0" ] && xfs_abort "block size"
                xfs_blksects=$((blksize/512))
                get_trimlist="xfs_trimlist"
        elif [ "$fstype" = "reiserfs" ]; then
                DEBUGREISERFS=`find_prog /sbin/debugreiserfs` || exit 1
                ( $DEBUGREISERFS $fsdev | $GREP '^Filesystem state:.consistent' ) &> /dev/null
                if [ $? -ne 0 ]; then
                        echo "Please run fsck.reiserfs first, aborting." >&2
                        exit 1
                fi
                get_trimlist="$DEBUGREISERFS -m $fsdev"
        elif [ "$fstype" = "hfsplus" ]; then
                OD=`find_prog /usr/bin/od` || exit 1
                TR=`find_prog /usr/bin/tr` || exit 1
                #check sleuthkit
                FSSTAT=`find_prog /usr/local/bin/fsstat`
                if [ "$?" = "1" ]; then
                        echo "fsstat and icat from package sleuthkit >= 3.1.1 is required for hfsplus."
                        exit 1
                fi
                ICAT=`find_prog /usr/local/bin/icat`
                if [ "`$ICAT -f list 2>/dev/stdout|$GREP HFS+`" = "" ]; then
                        echo "Wrong icat, version from package sleuthkit >= 3.1.1 is required for hfsplus."
                        exit 1
                fi
                #check for unmounted properly
                if [ "`$FSSTAT -f hfs $fsdev | $GREP "Volume Unmounted Properly"`" = ""  ]; then
                        echo "Hfsplus volume unmounted improperly!"
                        exit 1
```

```
                fi
                #check $AllocationFile inode
                FFIND=`find_prog /usr/local/bin/ffind`
                if [ "`$FFIND -f hfs $fsdev 6`" != "/\$AllocationFile" ]; then
                        echo "Hfsplus bitmap \$AllocationFile is not inode 6!"
                        exit 1
                fi
                #get offset for hfsplus with a wrapper
                hfsoffset=`$FSSTAT -f hfs $fsdev | $GREP "File system is embedded in an HFS wrapper at offset "|$TR -d "\t"`
                if [ -n "$hfsoffset" ]; then
                        hfsoffset=${hfsoffset:52}
                        ((fsoffset=fsoffset+hfsoffset))
                        echo "File system is embedded in an HFS wrapper at offset $hfsoffset"
                fi
                blksize=`$FSSTAT -f hfs $fsdev | $GREP "Allocation Block Size: "|$TR -d "\t"`
                blksize=${blksize:23}
                blksects=$((blksize / 512))
                #get count of used bytes in $AllocationFile
                blkcount=`$FSSTAT -f hfs $fsdev | $GREP "Block Range: 0 - "`
                blkcount=${blkcount:17}
                bytecount=$((blkcount/blksects))

                method="bitmap_offline"
                get_trimlist="echo $blksects hfsplus `$ICAT -f hfs $fsdev 6 | $OD -N $bytecount -An -vtu1 -j0 -w1`"
        elif [ "$fstype" = "ntfs" ]; then
                NTFSINFO=`find_prog /usr/bin/ntfsinfo` || exit 1
                NTFSCAT=`find_prog /usr/bin/ntfscat` || exit 1
                NTFSPROBE=`find_prog /usr/bin/ntfs-3g.probe` || exit 1
                OD=`find_prog /usr/bin/od` || exit 1
                TR=`find_prog /usr/bin/tr` || exit 1
                #check for unmounted properly
                $NTFSPROBE -w $fsdev 2>/dev/null
                if [ $? -ne 0 ]; then
                        echo "$fsdev contains an unclean file system!"
                        exit 1
                fi
                #check for volume version
                if [ "`$NTFSINFO -m -f $fsdev | $GREP "Volume Version: 3.1"`" = "" ]; then
                        echo "NTFS volume version must be 3.1!"
                        exit 1
                fi
                blksize=`$NTFSINFO -m -f $fsdev | $GREP "Cluster Size: " | $TR -d "\t"`
                blksize=${blksize:14}
                blksects=$((blksize / 512))
                #get count of used bytes in $Bitmap
                blkcount=`$NTFSINFO -m -f $fsdev | $GREP "Volume Size in Clusters: " | $TR -d "\t"`
                blkcount=${blkcount:25}
                bytecount=$((blkcount/blksects))

                method="bitmap_offline"
                get_trimlist="echo $blksects ntfs `$NTFSCAT $fsdev \\\$Bitmap | $OD -N $bytecount -An -vtu1 -j0 -w1`"
        fi
        if [ "$get_trimlist" = "" ]; then
                echo "$target: offline TRIM not supported for $fstype filesystems, aborting." >&2
                exit 1
        fi
fi

## All ready.  Now let the user know exactly what we intend to do:
##
mountstatus="$fstype non-mounted"
[ "$fsdir" = "" ] || mountstatus="$fstype mounted $fsmode at $fsdir"
echo "Preparing for $method TRIM of free space on $fsdev ($mountstatus)."

## If they specified "--commit" on the command line, then prompt for confirmation first:
##
if [ "$commit" = "yes" ]; then
        if [ "$destroy_me" = "" ]; then
                echo >/dev/tty
                echo -n "This operation could silently destroy your data.  Are you sure (y/N)? " >/dev/tty
                read yn < /dev/tty
                if [ "$yn" != "y" -a "$yn" != "Y" ]; then
                        echo "Aborting." >&2
                        exit 1
                fi
        fi
        TRIM="$HDPARM --please-destroy-my-drive --trim-sector-ranges-stdin $rawdev"
else
        echo "This will be a DRY-RUN only.  Use --commit to do it for real."
        TRIM="$GAWK {}"
fi

## Useful in a few places later on:
##
function sync_disks(){
        echo -n "Syncing disks.. "
        sync
        echo
}

## Clean up tmpfile (if any) and exit:
##
function do_cleanup(){
        if [ "$method" = "online" ]; then
                if [ -e $tmpfile ]; then
                        echo "Removing temporary file.."
                        $RM -f $tmpfile
                fi
```

```
                sync_disks
        fi
        [ $1 -eq 0 ] && echo "Done."
        [ $1 -eq 0 ] || echo "Aborted." >&2
        exit $1
}

## Prepare signal handling, in case we get interrupted while $tmpfile exists:
##
function do_abort(){
        echo
        do_cleanup 1
}
trap do_abort SIGTERM
trap do_abort SIGQUIT
trap do_abort SIGINT
trap do_abort SIGHUP
trap do_abort SIGPIPE

## For online TRIM, go ahead and create the huge temporary file.
## This is where we finally discover whether the filesystem actually
## supports --fallocate or not.  Some folks will be disappointed here.
##
## Note that --fallocate does not actually write any file data to fsdev,
## but rather simply allocates formerly-free space to the tmpfile.
##
if [ "$method" = "online" ]; then
        if [ -e "$tmpfile" ]; then
                if ! $RM -f "$tmpfile" ; then
                        echo "$tmpfile: already exists and could not be removed, aborting." >&2
                        exit 1
                fi
        fi
        echo -n "Allocating temporary file (${tmpsize} KB).. "
        if ! $HDPARM --fallocate "${tmpsize}" $tmpfile ; then
                echo "$target: this kernel may not support 'fallocate' on a $fstype filesystem, aborting." >&2
                exit 1
        fi
        echo
fi

## Finally, we are now ready to TRIM something!
##
## Feed the "get_trimlist" output into a gawk program which will
## extract the trimable lba-ranges (extents) and batch them together
## into huge --trim-sector-ranges calls.
##
## We are limited by at least one thing when doing this:
##   1. Some device drivers may not support more than 255 sectors
##      full of lba:count range data per TRIM command.
## The latest hdparm versions now take care of that automatically.
##
sync_disks
if [ "$commit" = "yes" ]; then
        echo "Beginning TRIM operations.."
else
        echo "Simulating TRIM operations.."
fi
[ $verbose -gt 0 ] && echo "get_trimlist=$get_trimlist"

## Begin gawk program
GAWKPROG='
        BEGIN {
                if (xfs_agoffsets != "") {
                        method = "xfs_offline"
                        agcount = split(xfs_agoffsets,agoffset," ");
                }
        }
        function append_range (lba,count  ,this_count){
                nsectors += count;
                while (count > 0) {
                        this_count  = (count > 65535) ? 65535 : count
                        printf "%u:%u ", lba, this_count
                        if (verbose > 1)
                                printf "%u:%u ", lba, this_count > "/dev/stderr"
                        lba        += this_count
                        count      -= this_count
                        nranges++;
                }
        }
        (method == "online") {  ## Output from "hdparm --fibmap", in absolute sectors:
                if (NF == 4 && $2 ~ "^[1-9][0-9]*$")
                        append_range($2,$4)
                next
        }
        (method == "xfs_offline") { ## Output from xfs_db:
                if (NF == 3 && gensub("[0-9 ]","","g",$0) == "" && $1 < agcount) {
                        lba   = agoffset[1 + $1] + ($2 * xfs_blksects) + fsoffset
                        count = $3 * xfs_blksects
                        append_range(lba,count)
                }
                next
        }
        (method == "bitmap_offline") {
                n = split($0,f)
                blksects = f[1]
                fstype = f[2]
                bitmap_start = 3
```

```
        range_first = -1 #clusters
        range_last = -1
        for (i = bitmap_start; i <= n-1; i++) {
                if (f[i] == 0) {
                        if (range_first == -1)
                                range_first = (i-bitmap_start) * 8
                        range_last = (i-bitmap_start) * 8 + 7
                } else if (f[i] == 255 && range_first > -1){
                        #printf range_first "-" range_last "\n" > "/dev/stderr"
                        lba = (range_first * blksects) + fsoffset
                        count = (range_last - range_first + 1) * blksects
                        append_range(lba,count)
                        range_first = -1
                        range_last = -1
                } else {
                        for (b = 0; b < 8; b++) {
                                if (fstype == "ntfs")
                                        bit = and(f[i], lshift(1, b)) ? 1 : 0
                                else #hfsplus
                                        bit = and(f[i], lshift(1, 7-b)) ? 1 : 0
                                if (bit == 0) {
                                        if (range_first == -1) {
                                                range_first = (i-bitmap_start) * 8 + b
                                                range_last = (i-bitmap_start) * 8 + b
                                        } else
                                                range_last += 1
                                } else if (range_first > -1) {
                                        #printf range_first "-" range_last " " > "/dev/stderr"
                                        lba = (range_first * blksects) + fsoffset
                                        count = (range_last - range_first + 1) * blksects
                                        if (fstype == "ntfs")
                                                append_range(lba,count)
                                        else if (count > (2 * blksects)) #faster for hfsplus
                                                append_range(lba,count)
                                        range_first = -1
                                        range_last = -1
                                }
                        }
                }
        }
        if (range_first > -1){
                #printf range_first "-" range_last " " > "/dev/stderr"
                lba = (range_first * blksects) + fsoffset
                count = (range_last - range_first + 1) * blksects
                append_range(lba,count)
        }
        next
}
/^Block size: *[1-9]/ { ## First stage output from dumpe2fs:
        blksects = $NF / 512
        next
}
/^Group [0-9][0-9]*:/ { ## Second stage output from dumpe2fs:
        in_groups = 1
        next
}
/^ *Free blocks: [0-9]/ { ## Bulk of output from dumpe2fs:
        if (blksects && in_groups) {
                n = split(substr($0,16),f,",*  *")
                for (i = 1; i <= n; ++i) {
                        if (f[i] ~ "^[1-9][0-9]*-[1-9][0-9]*$") {
                                split(f[i],b,"-")
                                lba   = (b[1] * blksects) + fsoffset
                                count = (b[2] - b[1] + 1) * blksects
                                append_range(lba,count)
                        } else if (f[i] ~ "^[1-9][0-9]*$") {
                                lba   = (f[i] * blksects) + fsoffset
                                count = blksects
                                append_range(lba,count)
                        }
                }
                next
        }
}
/^Reiserfs super block/ {
        method = "reiserfs"
        next
}
/^Blocksize: / {
        if (method == "reiserfs") {
                blksects = $2 / 512
                next
        }
}
/^#[0-9][0-9]*:.*Free[(]/ { ## debugreiserfs
        if (method == "reiserfs" && blksects > 0) {
                n = split($0,f)
                for (i = 4; i <= n; ++i) {
                        if (f[i] ~ "^ *Free[(]") {
                                if (2 == split(gensub("[^-0-9]","","g",f[i]),b,"-")) {
                                        lba = (b[1] * blksects) + fsoffset
                                        count = (b[2] - b[1] + 1) * blksects
                                        append_range(lba, count)
                                }
                        }
                }
                next
        }
```

```
        }
        END {
                if (err == 0 && commit != "yes")
                        printf "(dry-run) trimming %u sectors from %u ranges\n", nsectors, nranges > "/dev/stderr"
                exit err
        }'
## End gawk program

$get_trimlist 2>/dev/null | $GAWK               \
        -v commit="$commit"                     \
        -v method="$method"                     \
        -v rawdev="$rawdev"                     \
        -v fsoffset="$fsoffset"                 \
        -v verbose="$verbose"                   \
        -v xfs_blksects="$xfs_blksects"         \
        -v xfs_agoffsets="$xfs_agoffsets"       \
        "$GAWKPROG" | $TRIM

do_cleanup $?
```

Notice for package(s)

e2fsprogs

```
/*
 * ext2fs.h --- ext2fs
 *
 * Copyright (C) 1993, 1994, 1995, 1996 Theodore Ts'o.
 *
 * %Begin-Header%
 * This file may be redistributed under the terms of the GNU Library
 * General Public License, version 2.
 * %End-Header%
 */

#ifndef _EXT2FS_EXT2FS_H
#define _EXT2FS_EXT2FS_H

#ifdef __GNUC__
#define EXT2FS_ATTR(x) __attribute__(x)
#else
#define EXT2FS_ATTR(x)
#endif

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Non-GNU C compilers won't necessarily understand inline
 */
#if (!defined(__GNUC__) && !defined(__WATCOMC__))
#define NO_INLINE_FUNCS
#endif

/*
 * Where the master copy of the superblock is located, and how big
 * superblocks are supposed to be.  We define SUPERBLOCK_SIZE because
 * the size of the superblock structure is not necessarily trustworthy
 * (some versions have the padding set up so that the superblock is
 * 1032 bytes long).
 */
#define SUPERBLOCK_OFFSET       1024
#define SUPERBLOCK_SIZE         1024

/*
 * The last ext2fs revision level that this version of the library is
 * able to support.
 */
#define EXT2_LIB_CURRENT_REV    EXT2_DYNAMIC_REV

#ifdef HAVE_SYS_TYPES_H
#include <sys/types.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>

#if EXT2_FLAT_INCLUDES
#include "e2_types.h"
#include "ext2_fs.h"
#include "ext3_extents.h"
#else
#include <ext2fs/ext2_types.h>
#include <ext2fs/ext2_fs.h>
#include <ext2fs/ext3_extents.h>
#endif /* EXT2_FLAT_INCLUDES */

#ifdef __CHECK_ENDIAN__
```

```
#define __bitwise __attribute__((bitwise))
#else
#define __bitwise
#endif

typedef __u32 __bitwise         ext2_ino_t;
typedef __u32 __bitwise         blk_t;
typedef __u64 __bitwise         blk64_t;
typedef __u32 __bitwise         dgrp_t;
typedef __u32 __bitwise         ext2_off_t;
typedef __u64 __bitwise         ext2_off64_t;
typedef __s64 __bitwise         e2_blkcnt_t;
typedef __u32 __bitwise         ext2_dirhash_t;

#if EXT2_FLAT_INCLUDES
#include "com_err.h"
#include "ext2_io.h"
#include "ext2_err.h"
#include "ext2_ext_attr.h"
#else
#include <et/com_err.h>
#include <ext2fs/ext2_io.h>
#include <ext2fs/ext2_err.h>
#include <ext2fs/ext2_ext_attr.h>
#endif

/*
 * Portability help for Microsoft Visual C++
 */
#ifdef _MSC_VER
#define EXT2_QSORT_TYPE int __cdecl
#else
#define EXT2_QSORT_TYPE int
#endif

typedef struct struct_ext2_filsys *ext2_filsys;

#define EXT2FS_MARK_ERROR       0
#define EXT2FS_UNMARK_ERROR     1
#define EXT2FS_TEST_ERROR       2

typedef struct ext2fs_struct_generic_bitmap *ext2fs_generic_bitmap;
typedef struct ext2fs_struct_generic_bitmap *ext2fs_inode_bitmap;
typedef struct ext2fs_struct_generic_bitmap *ext2fs_block_bitmap;

#define EXT2_FIRST_INODE(s)     EXT2_FIRST_INO(s)


/*
 * Badblocks list definitions
 */

typedef struct ext2_struct_u32_list *ext2_badblocks_list;
typedef struct ext2_struct_u32_iterate *ext2_badblocks_iterate;

typedef struct ext2_struct_u32_list *ext2_u32_list;
typedef struct ext2_struct_u32_iterate *ext2_u32_iterate;

/* old */
typedef struct ext2_struct_u32_list *badblocks_list;
typedef struct ext2_struct_u32_iterate *badblocks_iterate;

#define BADBLOCKS_FLAG_DIRTY    1

/*
 * ext2_dblist structure and abstractions (see dblist.c)
 */
struct ext2_db_entry2 {
        ext2_ino_t      ino;
        blk64_t blk;
        e2_blkcnt_t     blockcnt;
};

/* Ye Olde 32-bit version */
struct ext2_db_entry {
        ext2_ino_t      ino;
        blk_t   blk;
        int     blockcnt;
};

typedef struct ext2_struct_dblist *ext2_dblist;

#define DBLIST_ABORT    1

/*
 * ext2_fileio definitions
 */

#define EXT2_FILE_WRITE         0x0001
#define EXT2_FILE_CREATE        0x0002

#define EXT2_FILE_MASK          0x00FF

#define EXT2_FILE_BUF_DIRTY     0x4000
#define EXT2_FILE_BUF_VALID     0x2000

typedef struct ext2_file *ext2_file_t;
```

```
#define EXT2_SEEK_SET   0
#define EXT2_SEEK_CUR   1
#define EXT2_SEEK_END   2

/*
 * Flags for the ext2_filsys structure and for ext2fs_open()
 */
#define EXT2_FLAG_RW                    0x01
#define EXT2_FLAG_CHANGED               0x02
#define EXT2_FLAG_DIRTY                 0x04
#define EXT2_FLAG_VALID                 0x08
#define EXT2_FLAG_IB_DIRTY              0x10
#define EXT2_FLAG_BB_DIRTY              0x20
#define EXT2_FLAG_SWAP_BYTES            0x40
#define EXT2_FLAG_SWAP_BYTES_READ       0x80
#define EXT2_FLAG_SWAP_BYTES_WRITE      0x100
#define EXT2_FLAG_MASTER_SB_ONLY        0x200
#define EXT2_FLAG_FORCE                 0x400
#define EXT2_FLAG_SUPER_ONLY            0x800
#define EXT2_FLAG_JOURNAL_DEV_OK        0x1000
#define EXT2_FLAG_IMAGE_FILE            0x2000
#define EXT2_FLAG_EXCLUSIVE             0x4000
#define EXT2_FLAG_SOFTSUPP_FEATURES     0x8000
#define EXT2_FLAG_NOFREE_ON_ERROR       0x10000
#define EXT2_FLAG_64BITS                0x20000
#define EXT2_FLAG_PRINT_PROGRESS        0x40000
#define EXT2_FLAG_DIRECT_IO             0x80000
#define EXT2_FLAG_SKIP_MMP              0x100000

/*
 * Special flag in the ext2 inode i_flag field that means that this is
 * a new inode.  (So that ext2_write_inode() can clear extra fields.)
 */
#define EXT2_NEW_INODE_FL       0x80000000

/*
 * Flags for mkjournal
 */
#define EXT2_MKJOURNAL_V1_SUPER 0x0000001 /* create V1 superblock (deprecated) */
#define EXT2_MKJOURNAL_LAZYINIT 0x0000002 /* don't zero journal inode before use*/
#define EXT2_MKJOURNAL_NO_MNT_CHECK 0x0000004 /* don't check mount status */

struct opaque_ext2_group_desc;

struct struct_ext2_filsys {
        errcode_t                       magic;
        io_channel                      io;
        int                             flags;
        char *                          device_name;
        struct ext2_super_block *       super;
        unsigned int                    blocksize;
        int                             fragsize;
        dgrp_t                          group_desc_count;
        unsigned long                   desc_blocks;
        struct opaque_ext2_group_desc * group_desc;
        unsigned int                    inode_blocks_per_group;
        ext2fs_inode_bitmap             inode_map;
        ext2fs_block_bitmap             block_map;
        /* XXX FIXME-64: not 64-bit safe, but not used? */
        errcode_t (*get_blocks)(ext2_filsys fs, ext2_ino_t ino, blk_t *blocks);
        errcode_t (*check_directory)(ext2_filsys fs, ext2_ino_t ino);
        errcode_t (*write_bitmaps)(ext2_filsys fs);
        errcode_t (*read_inode)(ext2_filsys fs, ext2_ino_t ino,
                                struct ext2_inode *inode);
        errcode_t (*write_inode)(ext2_filsys fs, ext2_ino_t ino,
                                 struct ext2_inode *inode);
        ext2_badblocks_list             badblocks;
        ext2_dblist                     dblist;
        __u32                           stride; /* for mke2fs */
        struct ext2_super_block *       orig_super;
        struct ext2_image_hdr *         image_header;
        __u32                           umask;
        time_t                          now;
        int                             cluster_ratio_bits;
        __u16                           default_bitmap_type;
        __u16                           pad;
        /*
         * Reserved for future expansion
         */
        __u32                           reserved[5];

        /*
         * Reserved for the use of the calling application.
         */
        void *                          priv_data;

        /*
         * Inode cache
         */
        struct ext2_inode_cache         *icache;
        io_channel                      image_io;

        /*
         * More callback functions
         */
        errcode_t (*get_alloc_block)(ext2_filsys fs, blk64_t goal,
                                     blk64_t *ret);
        void (*block_alloc_stats)(ext2_filsys fs, blk64_t blk, int inuse);
```

```
        /*
         * Buffers for Multiple mount protection(MMP) block.
         */
        void *mmp_buf;
        void *mmp_cmp;
        int mmp_fd;

        /*
         * Time at which e2fsck last updated the MMP block.
         */
        long mmp_last_written;
};

#if EXT2_FLAT_INCLUDES
#include "e2_bitops.h"
#else
#include <ext2fs/bitops.h>
#endif

/*
 * 64-bit bitmap backend types
 */
#define EXT2FS_BMAP64_BITARRAY  1
#define EXT2FS_BMAP64_RBTREE    2
#define EXT2FS_BMAP64_AUTODIR   3

/*
 * Return flags for the block iterator functions
 */
#define BLOCK_CHANGED   1
#define BLOCK_ABORT     2
#define BLOCK_ERROR     4

/*
 * Block interate flags
 *
 * BLOCK_FLAG_APPEND, or BLOCK_FLAG_HOLE, indicates that the interator
 * function should be called on blocks where the block number is zero.
 * This is used by ext2fs_expand_dir() to be able to add a new block
 * to an inode.  It can also be used for programs that want to be able
 * to deal with files that contain "holes".
 *
 * BLOCK_FLAG_DEPTH_TRAVERSE indicates that the iterator function for
 * the indirect, doubly indirect, etc. blocks should be called after
 * all of the blocks containined in the indirect blocks are processed.
 * This is useful if you are going to be deallocating blocks from an
 * inode.
 *
 * BLOCK_FLAG_DATA_ONLY indicates that the iterator function should be
 * called for data blocks only.
 *
 * BLOCK_FLAG_READ_ONLY is a promise by the caller that it will not
 * modify returned block number.
 *
 * BLOCK_FLAG_NO_LARGE is for internal use only.  It informs
 * ext2fs_block_iterate2 that large files won't be accepted.
 */
#define BLOCK_FLAG_APPEND       1
#define BLOCK_FLAG_HOLE         1
#define BLOCK_FLAG_DEPTH_TRAVERSE       2
#define BLOCK_FLAG_DATA_ONLY    4
#define BLOCK_FLAG_READ_ONLY    8

#define BLOCK_FLAG_NO_LARGE     0x1000

/*
 * Magic "block count" return values for the block iterator function.
 */
#define BLOCK_COUNT_IND         (-1)
#define BLOCK_COUNT_DIND        (-2)
#define BLOCK_COUNT_TIND        (-3)
#define BLOCK_COUNT_TRANSLATOR  (-4)

#if 0
/*
 * Flags for ext2fs_move_blocks
 */
#define EXT2_BMOVE_GET_DBLIST   0x0001
#define EXT2_BMOVE_DEBUG        0x0002
#endif

/*
 * Generic (non-filesystem layout specific) extents structure
 */

#define EXT2_EXTENT_FLAGS_LEAF          0x0001
#define EXT2_EXTENT_FLAGS_UNINIT        0x0002
#define EXT2_EXTENT_FLAGS_SECOND_VISIT  0x0004

struct ext2fs_extent {
        blk64_t e_pblk;         /* first physical block */
        blk64_t e_lblk;         /* first logical block extent covers */
        __u32   e_len;          /* number of blocks covered by extent */
        __u32   e_flags;        /* extent flags */
};

typedef struct ext2_extent_handle *ext2_extent_handle_t;
```

```
typedef struct ext2_extent_path *ext2_extent_path_t;

/*
 * Flags used by ext2fs_extent_get()
 */
#define EXT2_EXTENT_CURRENT     0x0000
#define EXT2_EXTENT_MOVE_MASK   0x000F
#define EXT2_EXTENT_ROOT        0x0001
#define EXT2_EXTENT_LAST_LEAF   0x0002
#define EXT2_EXTENT_FIRST_SIB   0x0003
#define EXT2_EXTENT_LAST_SIB    0x0004
#define EXT2_EXTENT_NEXT_SIB    0x0005
#define EXT2_EXTENT_PREV_SIB    0x0006
#define EXT2_EXTENT_NEXT_LEAF   0x0007
#define EXT2_EXTENT_PREV_LEAF   0x0008
#define EXT2_EXTENT_NEXT        0x0009
#define EXT2_EXTENT_PREV        0x000A
#define EXT2_EXTENT_UP          0x000B
#define EXT2_EXTENT_DOWN        0x000C
#define EXT2_EXTENT_DOWN_AND_LAST 0x000D

/*
 * Flags used by ext2fs_extent_insert()
 */
#define EXT2_EXTENT_INSERT_AFTER        0x0001 /* insert after handle loc'n */
#define EXT2_EXTENT_INSERT_NOSPLIT      0x0002 /* insert may not cause split */

/*
 * Flags used by ext2fs_extent_delete()
 */
#define EXT2_EXTENT_DELETE_KEEP_EMPTY   0x001 /* keep node if last extnt gone */

/*
 * Flags used by ext2fs_extent_set_bmap()
 */
#define EXT2_EXTENT_SET_BMAP_UNINIT     0x0001

/*
 * Data structure returned by ext2fs_extent_get_info()
 */
struct ext2_extent_info {
        int             curr_entry;
        int             curr_level;
        int             num_entries;
        int             max_entries;
        int             max_depth;
        int             bytes_avail;
        blk64_t         max_lblk;
        blk64_t         max_pblk;
        __u32           max_len;
        __u32           max_uninit_len;
};

/*
 * Flags for directory block reading and writing functions
 */
#define EXT2_DIRBLOCK_V2_STRUCT 0x0001

/*
 * Return flags for the directory iterator functions
 */
#define DIRENT_CHANGED  1
#define DIRENT_ABORT    2
#define DIRENT_ERROR    3

/*
 * Directory iterator flags
 */

#define DIRENT_FLAG_INCLUDE_EMPTY       1
#define DIRENT_FLAG_INCLUDE_REMOVED     2

#define DIRENT_DOT_FILE         1
#define DIRENT_DOT_DOT_FILE     2
#define DIRENT_OTHER_FILE       3
#define DIRENT_DELETED_FILE     4

/*
 * Inode scan definitions
 */
typedef struct ext2_struct_inode_scan *ext2_inode_scan;

/*
 * ext2fs_scan flags
 */
#define EXT2_SF_CHK_BADBLOCKS   0x0001
#define EXT2_SF_BAD_INODE_BLK   0x0002
#define EXT2_SF_BAD_EXTRA_BYTES 0x0004
#define EXT2_SF_SKIP_MISSING_ITABLE     0x0008
#define EXT2_SF_DO_LAZY         0x0010

/*
 * ext2fs_check_if_mounted flags
 */
#define EXT2_MF_MOUNTED         1
#define EXT2_MF_ISROOT          2
#define EXT2_MF_READONLY        4
#define EXT2_MF_SWAP            8
```

```
#define EXT2_MF_BUSY            16

/*
 * Ext2/linux mode flags.  We define them here so that we don't need
 * to depend on the OS's sys/stat.h, since we may be compiling on a
 * non-Linux system.
 */
#define LINUX_S_IFMT  00170000
#define LINUX_S_IFSOCK 0140000
#define LINUX_S_IFLNK    0120000
#define LINUX_S_IFREG  0100000
#define LINUX_S_IFBLK  0060000
#define LINUX_S_IFDIR  0040000
#define LINUX_S_IFCHR  0020000
#define LINUX_S_IFIFO  0010000
#define LINUX_S_ISUID  0004000
#define LINUX_S_ISGID  0002000
#define LINUX_S_ISVTX  0001000

#define LINUX_S_IRWXU 00700
#define LINUX_S_IRUSR 00400
#define LINUX_S_IWUSR 00200
#define LINUX_S_IXUSR 00100

#define LINUX_S_IRWXG 00070
#define LINUX_S_IRGRP 00040
#define LINUX_S_IWGRP 00020
#define LINUX_S_IXGRP 00010

#define LINUX_S_IRWXO 00007
#define LINUX_S_IROTH 00004
#define LINUX_S_IWOTH 00002
#define LINUX_S_IXOTH 00001

#define LINUX_S_ISLNK(m)        (((m) & LINUX_S_IFMT) == LINUX_S_IFLNK)
#define LINUX_S_ISREG(m)        (((m) & LINUX_S_IFMT) == LINUX_S_IFREG)
#define LINUX_S_ISDIR(m)        (((m) & LINUX_S_IFMT) == LINUX_S_IFDIR)
#define LINUX_S_ISCHR(m)        (((m) & LINUX_S_IFMT) == LINUX_S_IFCHR)
#define LINUX_S_ISBLK(m)        (((m) & LINUX_S_IFMT) == LINUX_S_IFBLK)
#define LINUX_S_ISFIFO(m)       (((m) & LINUX_S_IFMT) == LINUX_S_IFIFO)
#define LINUX_S_ISSOCK(m)       (((m) & LINUX_S_IFMT) == LINUX_S_IFSOCK)

/*
 * ext2 size of an inode
 */
#define EXT2_I_SIZE(i)  ((i)->i_size | ((__u64) (i)->i_size_high << 32))

/*
 * ext2_icount_t abstraction
 */
#define EXT2_ICOUNT_OPT_INCREMENT       0x01

typedef struct ext2_icount *ext2_icount_t;

/*
 * Flags for ext2fs_bmap
 */
#define BMAP_ALLOC      0x0001
#define BMAP_SET        0x0002

/*
 * Returned flags from ext2fs_bmap
 */
#define BMAP_RET_UNINIT 0x0001

/*
 * Flags for imager.c functions
 */
#define IMAGER_FLAG_INODEMAP    1
#define IMAGER_FLAG_SPARSEWRITE 2

/*
 * For checking structure magic numbers...
 */

#define EXT2_CHECK_MAGIC(struct, code) \
          if ((struct)->magic != (code)) return (code)


/*
 * For ext2 compression support
 */
#define EXT2FS_COMPRESSED_BLKADDR ((blk_t) -1)
#define HOLE_BLKADDR(_b) ((_b) == 0 || (_b) == EXT2FS_COMPRESSED_BLKADDR)

/*
 * Features supported by this version of the library
 */
#define EXT2_LIB_FEATURE_COMPAT_SUPP    (EXT2_FEATURE_COMPAT_DIR_PREALLOC|\
                                         EXT2_FEATURE_COMPAT_IMAGIC_INODES|\
                                         EXT3_FEATURE_COMPAT_HAS_JOURNAL|\
                                         EXT2_FEATURE_COMPAT_RESIZE_INODE|\
                                         EXT2_FEATURE_COMPAT_DIR_INDEX|\
                                         EXT2_FEATURE_COMPAT_EXT_ATTR)

/* This #ifdef is temporary until compression is fully supported */
#ifdef ENABLE_COMPRESSION
#ifndef I_KNOW_THAT_COMPRESSION_IS_EXPERIMENTAL
```

```
/* If the below warning bugs you, then have
   `CPPFLAGS=-DI_KNOW_THAT_COMPRESSION_IS_EXPERIMENTAL' in your
   environment at configure time. */
 #warning "Compression support is experimental"
#endif
#define EXT2_LIB_FEATURE_INCOMPAT_SUPP  (EXT2_FEATURE_INCOMPAT_FILETYPE|\
                                         EXT2_FEATURE_INCOMPAT_COMPRESSION|\
                                         EXT3_FEATURE_INCOMPAT_JOURNAL_DEV|\
                                         EXT2_FEATURE_INCOMPAT_META_BG|\
                                         EXT3_FEATURE_INCOMPAT_RECOVER|\
                                         EXT3_FEATURE_INCOMPAT_EXTENTS|\
                                         EXT4_FEATURE_INCOMPAT_FLEX_BG|\
                                         EXT4_FEATURE_INCOMPAT_MMP|\
                                         EXT4_FEATURE_INCOMPAT_64BIT)
#else
#define EXT2_LIB_FEATURE_INCOMPAT_SUPP  (EXT2_FEATURE_INCOMPAT_FILETYPE|\
                                         EXT3_FEATURE_INCOMPAT_JOURNAL_DEV|\
                                         EXT2_FEATURE_INCOMPAT_META_BG|\
                                         EXT3_FEATURE_INCOMPAT_RECOVER|\
                                         EXT3_FEATURE_INCOMPAT_EXTENTS|\
                                         EXT4_FEATURE_INCOMPAT_FLEX_BG|\
                                         EXT4_FEATURE_INCOMPAT_MMP|\
                                         EXT4_FEATURE_INCOMPAT_64BIT)
#endif
#ifdef CONFIG_QUOTA
#define EXT2_LIB_FEATURE_RO_COMPAT_SUPP (EXT2_FEATURE_RO_COMPAT_SPARSE_SUPER|\
                                         EXT4_FEATURE_RO_COMPAT_HUGE_FILE|\
                                         EXT2_FEATURE_RO_COMPAT_LARGE_FILE|\
                                         EXT4_FEATURE_RO_COMPAT_DIR_NLINK|\
                                         EXT4_FEATURE_RO_COMPAT_EXTRA_ISIZE|\
                                         EXT4_FEATURE_RO_COMPAT_GDT_CSUM|\
                                         EXT4_FEATURE_RO_COMPAT_BIGALLOC|\
                                         EXT4_FEATURE_RO_COMPAT_QUOTA)
#else
#define EXT2_LIB_FEATURE_RO_COMPAT_SUPP (EXT2_FEATURE_RO_COMPAT_SPARSE_SUPER|\
                                         EXT4_FEATURE_RO_COMPAT_HUGE_FILE|\
                                         EXT2_FEATURE_RO_COMPAT_LARGE_FILE|\
                                         EXT4_FEATURE_RO_COMPAT_DIR_NLINK|\
                                         EXT4_FEATURE_RO_COMPAT_EXTRA_ISIZE|\
                                         EXT4_FEATURE_RO_COMPAT_GDT_CSUM|\
                                         EXT4_FEATURE_RO_COMPAT_BIGALLOC)
#endif

/*
 * These features are only allowed if EXT2_FLAG_SOFTSUPP_FEATURES is passed
 * to ext2fs_openfs()
 */
#define EXT2_LIB_SOFTSUPP_INCOMPAT      (0)
#define EXT2_LIB_SOFTSUPP_RO_COMPAT     (EXT4_FEATURE_RO_COMPAT_REPLICA)


/* Translate a block number to a cluster number */
#define EXT2FS_CLUSTER_RATIO(fs)        (1 << (fs)->cluster_ratio_bits)
#define EXT2FS_CLUSTER_MASK(fs)         (EXT2FS_CLUSTER_RATIO(fs) - 1)
#define EXT2FS_B2C(fs, blk)             ((blk) >> (fs)->cluster_ratio_bits)
/* Translate a cluster number to a block number */
#define EXT2FS_C2B(fs, cluster)         ((cluster) << (fs)->cluster_ratio_bits)
/* Translate # of blks to # of clusters */
#define EXT2FS_NUM_B2C(fs, blks)        (((blks) + EXT2FS_CLUSTER_MASK(fs)) >> \
                                         (fs)->cluster_ratio_bits)

#if defined(HAVE_FSTAT64) && !defined(__OSX_AVAILABLE_BUT_DEPRECATED)
typedef struct stat64 ext2fs_struct_stat;
#else
typedef struct stat ext2fs_struct_stat;
#endif

/*
 * For ext2fs_close2() and ext2fs_flush2(), this flag allows you to
 * avoid the fsync call.
 */
#define EXT2_FLAG_FLUSH_NO_SYNC          1

/*
 * function prototypes
 */

/* The LARGE_FILE feature should be set if we have stored files 2GB+ in size */
static inline int ext2fs_needs_large_file_feature(unsigned long long file_size)
{
        return file_size >= 0x80000000ULL;
}

/* alloc.c */
extern errcode_t ext2fs_new_inode(ext2_filsys fs, ext2_ino_t dir, int mode,
                                  ext2fs_inode_bitmap map, ext2_ino_t *ret);
extern errcode_t ext2fs_new_block(ext2_filsys fs, blk_t goal,
                                  ext2fs_block_bitmap map, blk_t *ret);
extern errcode_t ext2fs_new_block2(ext2_filsys fs, blk64_t goal,
                                   ext2fs_block_bitmap map, blk64_t *ret);
extern errcode_t ext2fs_get_free_blocks(ext2_filsys fs, blk_t start,
                                        blk_t finish, int num,
                                        ext2fs_block_bitmap map,
                                        blk_t *ret);
extern errcode_t ext2fs_get_free_blocks2(ext2_filsys fs, blk64_t start,
                                         blk64_t finish, int num,
                                         ext2fs_block_bitmap map,
                                         blk64_t *ret);
```

```
extern errcode_t ext2fs_alloc_block(ext2_filsys fs, blk_t goal,
                                    char *block_buf, blk_t *ret);
extern errcode_t ext2fs_alloc_block2(ext2_filsys fs, blk64_t goal,
                                     char *block_buf, blk64_t *ret);
extern void ext2fs_set_alloc_block_callback(ext2_filsys fs,
                                            errcode_t (*func)(ext2_filsys fs,
                                                              blk64_t goal,
                                                              blk64_t *ret),
                                            errcode_t (**old)(ext2_filsys fs,
                                                              blk64_t goal,
                                                              blk64_t *ret));

/* alloc_sb.c */
extern int ext2fs_reserve_super_and_bgd(ext2_filsys fs,
                                        dgrp_t group,
                                        ext2fs_block_bitmap bmap);
extern void ext2fs_set_block_alloc_stats_callback(ext2_filsys fs,
                                                  void (*func)(ext2_filsys fs,
                                                               blk64_t blk,
                                                               int inuse),
                                                  void (**old)(ext2_filsys fs,
                                                               blk64_t blk,
                                                               int inuse));

/* alloc_stats.c */
void ext2fs_inode_alloc_stats(ext2_filsys fs, ext2_ino_t ino, int inuse);
void ext2fs_inode_alloc_stats2(ext2_filsys fs, ext2_ino_t ino,
                               int inuse, int isdir);
void ext2fs_block_alloc_stats(ext2_filsys fs, blk_t blk, int inuse);
void ext2fs_block_alloc_stats2(ext2_filsys fs, blk64_t blk, int inuse);

/* alloc_tables.c */
extern errcode_t ext2fs_allocate_tables(ext2_filsys fs);
extern errcode_t ext2fs_allocate_group_table(ext2_filsys fs, dgrp_t group,
                                             ext2fs_block_bitmap bmap);

/* badblocks.c */
extern errcode_t ext2fs_u32_list_create(ext2_u32_list *ret, int size);
extern errcode_t ext2fs_u32_list_add(ext2_u32_list bb, __u32 blk);
extern int ext2fs_u32_list_find(ext2_u32_list bb, __u32 blk);
extern int ext2fs_u32_list_test(ext2_u32_list bb, blk_t blk);
extern errcode_t ext2fs_u32_list_iterate_begin(ext2_u32_list bb,
                                               ext2_u32_iterate *ret);
extern int ext2fs_u32_list_iterate(ext2_u32_iterate iter, blk_t *blk);
extern void ext2fs_u32_list_iterate_end(ext2_u32_iterate iter);
extern errcode_t ext2fs_u32_copy(ext2_u32_list src, ext2_u32_list *dest);
extern int ext2fs_u32_list_equal(ext2_u32_list bb1, ext2_u32_list bb2);

extern errcode_t ext2fs_badblocks_list_create(ext2_badblocks_list *ret,
                                              int size);
extern errcode_t ext2fs_badblocks_list_add(ext2_badblocks_list bb,
                                           blk_t blk);
extern int ext2fs_badblocks_list_test(ext2_badblocks_list bb,
                                      blk_t blk);
extern int ext2fs_u32_list_del(ext2_u32_list bb, __u32 blk);
extern void ext2fs_badblocks_list_del(ext2_u32_list bb, __u32 blk);
extern errcode_t
       ext2fs_badblocks_list_iterate_begin(ext2_badblocks_list bb,
                                           ext2_badblocks_iterate *ret);
extern int ext2fs_badblocks_list_iterate(ext2_badblocks_iterate iter,
                                         blk_t *blk);
extern void ext2fs_badblocks_list_iterate_end(ext2_badblocks_iterate iter);
extern errcode_t ext2fs_badblocks_copy(ext2_badblocks_list src,
                                       ext2_badblocks_list *dest);
extern int ext2fs_badblocks_equal(ext2_badblocks_list bb1,
                                  ext2_badblocks_list bb2);
extern int ext2fs_u32_list_count(ext2_u32_list bb);

/* bb_compat */
extern errcode_t badblocks_list_create(badblocks_list *ret, int size);
extern errcode_t badblocks_list_add(badblocks_list bb, blk_t blk);
extern int badblocks_list_test(badblocks_list bb, blk_t blk);
extern errcode_t badblocks_list_iterate_begin(badblocks_list bb,
                                              badblocks_iterate *ret);
extern int badblocks_list_iterate(badblocks_iterate iter, blk_t *blk);
extern void badblocks_list_iterate_end(badblocks_iterate iter);
extern void badblocks_list_free(badblocks_list bb);

/* bb_inode.c */
extern errcode_t ext2fs_update_bb_inode(ext2_filsys fs,
                                        ext2_badblocks_list bb_list);

/* bitmaps.c */
extern void ext2fs_free_block_bitmap(ext2fs_block_bitmap bitmap);
extern void ext2fs_free_inode_bitmap(ext2fs_inode_bitmap bitmap);
extern errcode_t ext2fs_copy_bitmap(ext2fs_generic_bitmap src,
                                    ext2fs_generic_bitmap *dest);
extern errcode_t ext2fs_write_inode_bitmap(ext2_filsys fs);
extern errcode_t ext2fs_write_block_bitmap (ext2_filsys fs);
extern errcode_t ext2fs_read_inode_bitmap (ext2_filsys fs);
extern errcode_t ext2fs_read_block_bitmap(ext2_filsys fs);
extern errcode_t ext2fs_allocate_block_bitmap(ext2_filsys fs,
                                              const char *descr,
                                              ext2fs_block_bitmap *ret);
extern errcode_t ext2fs_allocate_subcluster_bitmap(ext2_filsys fs,
                                                   const char *descr,
                                                   ext2fs_block_bitmap *ret);
extern int ext2fs_get_bitmap_granularity(ext2fs_block_bitmap bitmap);
```

```
extern errcode_t ext2fs_allocate_inode_bitmap(ext2_filsys fs,
                                              const char *descr,
                                              ext2fs_inode_bitmap *ret);
extern errcode_t ext2fs_fudge_inode_bitmap_end(ext2fs_inode_bitmap bitmap,
                                               ext2_ino_t end, ext2_ino_t *oend);
extern errcode_t ext2fs_fudge_block_bitmap_end(ext2fs_block_bitmap bitmap,
                                               blk_t end, blk_t *oend);
extern errcode_t ext2fs_fudge_block_bitmap_end2(ext2fs_block_bitmap bitmap,
                                        blk64_t end, blk64_t *oend);
extern void ext2fs_clear_inode_bitmap(ext2fs_inode_bitmap bitmap);
extern void ext2fs_clear_block_bitmap(ext2fs_block_bitmap bitmap);
extern errcode_t ext2fs_read_bitmaps(ext2_filsys fs);
extern errcode_t ext2fs_write_bitmaps(ext2_filsys fs);
extern errcode_t ext2fs_resize_inode_bitmap(__u32 new_end, __u32 new_real_end,
                                            ext2fs_inode_bitmap bmap);
extern errcode_t ext2fs_resize_inode_bitmap2(__u64 new_end,
                                             __u64 new_real_end,
                                             ext2fs_inode_bitmap bmap);
extern errcode_t ext2fs_resize_block_bitmap(__u32 new_end, __u32 new_real_end,
                                            ext2fs_block_bitmap bmap);
extern errcode_t ext2fs_resize_block_bitmap2(__u64 new_end,
                                             __u64 new_real_end,
                                             ext2fs_block_bitmap bmap);
extern errcode_t ext2fs_compare_block_bitmap(ext2fs_block_bitmap bm1,
                                             ext2fs_block_bitmap bm2);
extern errcode_t ext2fs_compare_inode_bitmap(ext2fs_inode_bitmap bm1,
                                             ext2fs_inode_bitmap bm2);
extern errcode_t ext2fs_set_inode_bitmap_range(ext2fs_inode_bitmap bmap,
                                       ext2_ino_t start, unsigned int num,
                                       void *in);
extern errcode_t ext2fs_set_inode_bitmap_range2(ext2fs_inode_bitmap bmap,
                                        __u64 start, size_t num,
                                        void *in);
extern errcode_t ext2fs_get_inode_bitmap_range(ext2fs_inode_bitmap bmap,
                                       ext2_ino_t start, unsigned int num,
                                       void *out);
extern errcode_t ext2fs_get_inode_bitmap_range2(ext2fs_inode_bitmap bmap,
                                        __u64 start, size_t num,
                                        void *out);
extern errcode_t ext2fs_set_block_bitmap_range(ext2fs_block_bitmap bmap,
                                       blk_t start, unsigned int num,
                                       void *in);
extern errcode_t ext2fs_set_block_bitmap_range2(ext2fs_block_bitmap bmap,
                                        blk64_t start, size_t num,
                                        void *in);
extern errcode_t ext2fs_get_block_bitmap_range(ext2fs_block_bitmap bmap,
                                       blk_t start, unsigned int num,
                                       void *out);
extern errcode_t ext2fs_get_block_bitmap_range2(ext2fs_block_bitmap bmap,
                                        blk64_t start, size_t num,
                                        void *out);

/* blknum.c */
extern dgrp_t ext2fs_group_of_blk2(ext2_filsys fs, blk64_t);
extern blk64_t ext2fs_group_first_block2(ext2_filsys fs, dgrp_t group);
extern blk64_t ext2fs_group_last_block2(ext2_filsys fs, dgrp_t group);
extern int ext2fs_group_blocks_count(ext2_filsys fs, dgrp_t group);
extern blk64_t ext2fs_inode_data_blocks2(ext2_filsys fs,
                                         struct ext2_inode *inode);
extern blk64_t ext2fs_inode_i_blocks(ext2_filsys fs,
                                     struct ext2_inode *inode);
extern blk64_t ext2fs_blocks_count(struct ext2_super_block *super);
extern void ext2fs_blocks_count_set(struct ext2_super_block *super,
                                    blk64_t blk);
extern void ext2fs_blocks_count_add(struct ext2_super_block *super,
                                    blk64_t blk);
extern blk64_t ext2fs_r_blocks_count(struct ext2_super_block *super);
extern void ext2fs_r_blocks_count_set(struct ext2_super_block *super,
                                      blk64_t blk);
extern void ext2fs_r_blocks_count_add(struct ext2_super_block *super,
                                      blk64_t blk);
extern blk64_t ext2fs_free_blocks_count(struct ext2_super_block *super);
extern void ext2fs_free_blocks_count_set(struct ext2_super_block *super,
                                         blk64_t blk);
extern void ext2fs_free_blocks_count_add(struct ext2_super_block *super,
                                         blk64_t blk);
/* Block group descriptor accessor functions */
extern struct ext2_group_desc *ext2fs_group_desc(ext2_filsys fs,
                                          struct opaque_ext2_group_desc *gdp,
                                          dgrp_t group);
extern blk64_t ext2fs_block_bitmap_loc(ext2_filsys fs, dgrp_t group);
extern void ext2fs_block_bitmap_loc_set(ext2_filsys fs, dgrp_t group,
                                        blk64_t blk);
extern blk64_t ext2fs_inode_bitmap_loc(ext2_filsys fs, dgrp_t group);
extern void ext2fs_inode_bitmap_loc_set(ext2_filsys fs, dgrp_t group,
                                        blk64_t blk);
extern blk64_t ext2fs_inode_table_loc(ext2_filsys fs, dgrp_t group);
extern void ext2fs_inode_table_loc_set(ext2_filsys fs, dgrp_t group,
                                       blk64_t blk);
extern __u32 ext2fs_bg_free_blocks_count(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_free_blocks_count_set(ext2_filsys fs, dgrp_t group,
                                            __u32 n);
extern __u32 ext2fs_bg_free_inodes_count(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_free_inodes_count_set(ext2_filsys fs, dgrp_t group,
                                            __u32 n);
extern __u32 ext2fs_bg_used_dirs_count(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_used_dirs_count_set(ext2_filsys fs, dgrp_t group,
                                          __u32 n);
```

```
extern __u32 ext2fs_bg_itable_unused(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_itable_unused_set(ext2_filsys fs, dgrp_t group,
                                        __u32 n);
extern __u16 ext2fs_bg_flags(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_flags_zap(ext2_filsys fs, dgrp_t group);
extern int ext2fs_bg_flags_test(ext2_filsys fs, dgrp_t group, __u16 bg_flag);
extern void ext2fs_bg_flags_set(ext2_filsys fs, dgrp_t group, __u16 bg_flags);
extern void ext2fs_bg_flags_clear(ext2_filsys fs, dgrp_t group, __u16 bg_flags);
extern __u16 ext2fs_bg_checksum(ext2_filsys fs, dgrp_t group);
extern void ext2fs_bg_checksum_set(ext2_filsys fs, dgrp_t group, __u16 checksum);
extern blk64_t ext2fs_file_acl_block(ext2_filsys fs,
                                     const struct ext2_inode *inode);
extern void ext2fs_file_acl_block_set(ext2_filsys fs,
                                      struct ext2_inode *inode, blk64_t blk);

/* block.c */
extern errcode_t ext2fs_block_iterate(ext2_filsys fs,
                                      ext2_ino_t        ino,
                                      int       flags,
                                      char *block_buf,
                                      int (*func)(ext2_filsys fs,
                                                  blk_t *blocknr,
                                                  int   blockcnt,
                                                  void  *priv_data),
                                      void *priv_data);
errcode_t ext2fs_block_iterate2(ext2_filsys fs,
                                ext2_ino_t      ino,
                                int     flags,
                                char *block_buf,
                                int (*func)(ext2_filsys fs,
                                            blk_t       *blocknr,
                                            e2_blkcnt_t blockcnt,
                                            blk_t       ref_blk,
                                            int         ref_offset,
                                            void        *priv_data),
                                void *priv_data);
errcode_t ext2fs_block_iterate3(ext2_filsys fs,
                                ext2_ino_t ino,
                                int     flags,
                                char *block_buf,
                                int (*func)(ext2_filsys fs,
                                            blk64_t     *blocknr,
                                            e2_blkcnt_t blockcnt,
                                            blk64_t     ref_blk,
                                            int         ref_offset,
                                            void        *priv_data),
                                void *priv_data);

/* bmap.c */
extern errcode_t ext2fs_bmap(ext2_filsys fs, ext2_ino_t ino,
                             struct ext2_inode *inode,
                             char *block_buf, int bmap_flags,
                             blk_t block, blk_t *phys_blk);
extern errcode_t ext2fs_bmap2(ext2_filsys fs, ext2_ino_t ino,
                              struct ext2_inode *inode,
                              char *block_buf, int bmap_flags, blk64_t block,
                              int *ret_flags, blk64_t *phys_blk);
errcode_t ext2fs_map_cluster_block(ext2_filsys fs, ext2_ino_t ino,
                                   struct ext2_inode *inode, blk64_t lblk,
                                   blk64_t *pblk);

#if 0
/* bmove.c */
extern errcode_t ext2fs_move_blocks(ext2_filsys fs,
                                    ext2fs_block_bitmap reserve,
                                    ext2fs_block_bitmap alloc_map,
                                    int flags);
#endif

/* check_desc.c */
extern errcode_t ext2fs_check_desc(ext2_filsys fs);

/* closefs.c */
extern errcode_t ext2fs_close(ext2_filsys fs);
extern errcode_t ext2fs_close2(ext2_filsys fs, int flags);
extern errcode_t ext2fs_flush(ext2_filsys fs);
extern errcode_t ext2fs_flush2(ext2_filsys fs, int flags);
extern int ext2fs_bg_has_super(ext2_filsys fs, dgrp_t group_block);
extern errcode_t ext2fs_super_and_bgd_loc2(ext2_filsys fs,
                                    dgrp_t group,
                                    blk64_t *ret_super_blk,
                                    blk64_t *ret_old_desc_blk,
                                    blk64_t *ret_new_desc_blk,
                                    blk_t *ret_used_blks);
extern int ext2fs_super_and_bgd_loc(ext2_filsys fs,
                                    dgrp_t group,
                                    blk_t *ret_super_blk,
                                    blk_t *ret_old_desc_blk,
                                    blk_t *ret_new_desc_blk,
                                    int *ret_meta_bg);
extern void ext2fs_update_dynamic_rev(ext2_filsys fs);

/* crc32c.c */
extern __u32 ext2fs_crc32c_be(__u32 crc, unsigned char const *p, size_t len);
extern __u32 ext2fs_crc32c_le(__u32 crc, unsigned char const *p, size_t len);

/* csum.c */
extern void ext2fs_group_desc_csum_set(ext2_filsys fs, dgrp_t group);
```

```
extern int ext2fs_group_desc_csum_verify(ext2_filsys fs, dgrp_t group);
extern errcode_t ext2fs_set_gdt_csum(ext2_filsys fs);
extern __u16 ext2fs_group_desc_csum(ext2_filsys fs, dgrp_t group);

/* dblist.c */

extern errcode_t ext2fs_get_num_dirs(ext2_filsys fs, ext2_ino_t *ret_num_dirs);
extern errcode_t ext2fs_init_dblist(ext2_filsys fs, ext2_dblist *ret_dblist);
extern errcode_t ext2fs_add_dir_block(ext2_dblist dblist, ext2_ino_t ino,
                                      blk_t blk, int blockcnt);
extern errcode_t ext2fs_add_dir_block2(ext2_dblist dblist, ext2_ino_t ino,
                                       blk64_t blk, e2_blkcnt_t blockcnt);
extern void ext2fs_dblist_sort(ext2_dblist dblist,
                               EXT2_QSORT_TYPE (*sortfunc)(const void *,
                                                           const void *));
extern void ext2fs_dblist_sort2(ext2_dblist dblist,
                                EXT2_QSORT_TYPE (*sortfunc)(const void *,
                                                            const void *));
extern errcode_t ext2fs_dblist_iterate(ext2_dblist dblist,
       int (*func)(ext2_filsys fs, struct ext2_db_entry *db_info,
                   void        *priv_data),
       void *priv_data);
extern errcode_t ext2fs_dblist_iterate2(ext2_dblist dblist,
       int (*func)(ext2_filsys fs, struct ext2_db_entry2 *db_info,
                   void        *priv_data),
       void *priv_data);
extern errcode_t ext2fs_set_dir_block(ext2_dblist dblist, ext2_ino_t ino,
                                      blk_t blk, int blockcnt);
extern errcode_t ext2fs_set_dir_block2(ext2_dblist dblist, ext2_ino_t ino,
                                       blk64_t blk, e2_blkcnt_t blockcnt);
extern errcode_t ext2fs_copy_dblist(ext2_dblist src,
                                    ext2_dblist *dest);
extern int ext2fs_dblist_count(ext2_dblist dblist);
extern blk64_t ext2fs_dblist_count2(ext2_dblist dblist);
extern errcode_t ext2fs_dblist_get_last(ext2_dblist dblist,
                                        struct ext2_db_entry **entry);
extern errcode_t ext2fs_dblist_get_last2(ext2_dblist dblist,
                                         struct ext2_db_entry2 **entry);
extern errcode_t ext2fs_dblist_drop_last(ext2_dblist dblist);

/* dblist_dir.c */
extern errcode_t
       ext2fs_dblist_dir_iterate(ext2_dblist dblist,
                                 int   flags,
                                 char  *block_buf,
                                 int (*func)(ext2_ino_t        dir,
                                             int               entry,
                                             struct ext2_dir_entry *dirent,
                                             int       offset,
                                             int       blocksize,
                                             char      *buf,
                                             void      *priv_data),
                                 void *priv_data);

/* dirblock.c */
extern errcode_t ext2fs_read_dir_block(ext2_filsys fs, blk_t block,
                                       void *buf);
extern errcode_t ext2fs_read_dir_block2(ext2_filsys fs, blk_t block,
                                        void *buf, int flags);
extern errcode_t ext2fs_read_dir_block3(ext2_filsys fs, blk64_t block,
                                        void *buf, int flags);
extern errcode_t ext2fs_write_dir_block(ext2_filsys fs, blk_t block,
                                        void *buf);
extern errcode_t ext2fs_write_dir_block2(ext2_filsys fs, blk_t block,
                                         void *buf, int flags);
extern errcode_t ext2fs_write_dir_block3(ext2_filsys fs, blk64_t block,
                                         void *buf, int flags);

/* dirhash.c */
extern errcode_t ext2fs_dirhash(int version, const char *name, int len,
                                const __u32 *seed,
                                ext2_dirhash_t *ret_hash,
                                ext2_dirhash_t *ret_minor_hash);


/* dir_iterate.c */
extern errcode_t ext2fs_get_rec_len(ext2_filsys fs,
                                    struct ext2_dir_entry *dirent,
                                    unsigned int *rec_len);
extern errcode_t ext2fs_set_rec_len(ext2_filsys fs,
                                    unsigned int len,
                                    struct ext2_dir_entry *dirent);
extern errcode_t ext2fs_dir_iterate(ext2_filsys fs,
                                ext2_ino_t dir,
                                int flags,
                                char *block_buf,
                                int (*func)(struct ext2_dir_entry *dirent,
                                            int   offset,
                                            int   blocksize,
                                            char  *buf,
                                            void  *priv_data),
                                void *priv_data);
extern errcode_t ext2fs_dir_iterate2(ext2_filsys fs,
                                ext2_ino_t dir,
                                int flags,
                                char *block_buf,
                                int (*func)(ext2_ino_t    dir,
                                            int   entry,
```

```
                                        struct ext2_dir_entry *dirent,
                                        int   offset,
                                        int   blocksize,
                                        char  *buf,
                                        void  *priv_data),
                                void *priv_data);

/* dupfs.c */
extern errcode_t ext2fs_dup_handle(ext2_filsys src, ext2_filsys *dest);

/* expanddir.c */
extern errcode_t ext2fs_expand_dir(ext2_filsys fs, ext2_ino_t dir);

/* ext_attr.c */
extern __u32 ext2fs_ext_attr_hash_entry(struct ext2_ext_attr_entry *entry,
                                        void *data);
extern errcode_t ext2fs_read_ext_attr(ext2_filsys fs, blk_t block, void *buf);
extern errcode_t ext2fs_read_ext_attr2(ext2_filsys fs, blk64_t block,
                                       void *buf);
extern errcode_t ext2fs_write_ext_attr(ext2_filsys fs, blk_t block,
                                       void *buf);
extern errcode_t ext2fs_write_ext_attr2(ext2_filsys fs, blk64_t block,
                                       void *buf);
extern errcode_t ext2fs_adjust_ea_refcount(ext2_filsys fs, blk_t blk,
                                           char *block_buf,
                                           int adjust, __u32 *newcount);
extern errcode_t ext2fs_adjust_ea_refcount2(ext2_filsys fs, blk64_t blk,
                                           char *block_buf,
                                           int adjust, __u32 *newcount);

/* extent.c */
extern errcode_t ext2fs_extent_header_verify(void *ptr, int size);
extern errcode_t ext2fs_extent_open(ext2_filsys fs, ext2_ino_t ino,
                                    ext2_extent_handle_t *handle);
extern errcode_t ext2fs_extent_open2(ext2_filsys fs, ext2_ino_t ino,
                                     struct ext2_inode *inode,
                                     ext2_extent_handle_t *ret_handle);
extern void ext2fs_extent_free(ext2_extent_handle_t handle);
extern errcode_t ext2fs_extent_get(ext2_extent_handle_t handle,
                                   int flags, struct ext2fs_extent *extent);
extern errcode_t ext2fs_extent_node_split(ext2_extent_handle_t handle);
extern errcode_t ext2fs_extent_replace(ext2_extent_handle_t handle, int flags,
                                       struct ext2fs_extent *extent);
extern errcode_t ext2fs_extent_insert(ext2_extent_handle_t handle, int flags,
                                      struct ext2fs_extent *extent);
extern errcode_t ext2fs_extent_set_bmap(ext2_extent_handle_t handle,
                                        blk64_t logical, blk64_t physical,
                                        int flags);
extern errcode_t ext2fs_extent_delete(ext2_extent_handle_t handle, int flags);
extern errcode_t ext2fs_extent_get_info(ext2_extent_handle_t handle,
                                        struct ext2_extent_info *info);
extern errcode_t ext2fs_extent_goto(ext2_extent_handle_t handle,
                                    blk64_t blk);
extern errcode_t ext2fs_extent_goto2(ext2_extent_handle_t handle,
                                     int leaf_level, blk64_t blk);
extern errcode_t ext2fs_extent_fix_parents(ext2_extent_handle_t handle);

/* fileio.c */
extern errcode_t ext2fs_file_open2(ext2_filsys fs, ext2_ino_t ino,
                                   struct ext2_inode *inode,
                                   int flags, ext2_file_t *ret);
extern errcode_t ext2fs_file_open(ext2_filsys fs, ext2_ino_t ino,
                                  int flags, ext2_file_t *ret);
extern ext2_filsys ext2fs_file_get_fs(ext2_file_t file);
struct ext2_inode *ext2fs_file_get_inode(ext2_file_t file);
extern ext2_ino_t ext2fs_file_get_inode_num(ext2_file_t file);
extern errcode_t ext2fs_file_close(ext2_file_t file);
extern errcode_t ext2fs_file_flush(ext2_file_t file);
extern errcode_t ext2fs_file_read(ext2_file_t file, void *buf,
                                  unsigned int wanted, unsigned int *got);
extern errcode_t ext2fs_file_write(ext2_file_t file, const void *buf,
                                   unsigned int nbytes, unsigned int *written);
extern errcode_t ext2fs_file_llseek(ext2_file_t file, __u64 offset,
                                   int whence, __u64 *ret_pos);
extern errcode_t ext2fs_file_lseek(ext2_file_t file, ext2_off_t offset,
                                   int whence, ext2_off_t *ret_pos);
errcode_t ext2fs_file_get_lsize(ext2_file_t file, __u64 *ret_size);
extern ext2_off_t ext2fs_file_get_size(ext2_file_t file);
extern errcode_t ext2fs_file_set_size(ext2_file_t file, ext2_off_t size);
extern errcode_t ext2fs_file_set_size2(ext2_file_t file, ext2_off64_t size);

/* finddev.c */
extern char *ext2fs_find_block_device(dev_t device);

/* flushb.c */
extern errcode_t ext2fs_sync_device(int fd, int flushb);

/* freefs.c */
extern void ext2fs_free(ext2_filsys fs);
extern void ext2fs_free_dblist(ext2_dblist dblist);
extern void ext2fs_badblocks_list_free(ext2_badblocks_list bb);
extern void ext2fs_u32_list_free(ext2_u32_list bb);

/* gen_bitmap.c */
extern void ext2fs_free_generic_bitmap(ext2fs_inode_bitmap bitmap);
extern errcode_t ext2fs_make_generic_bitmap(errcode_t magic, ext2_filsys fs,
                                            __u32 start, __u32 end,
                                            __u32 real_end,
```

```
                                        const char *descr, char *init_map,
                                        ext2fs_generic_bitmap *ret);
extern errcode_t ext2fs_allocate_generic_bitmap(__u32 start,
                                          __u32 end,
                                          __u32 real_end,
                                          const char *descr,
                                          ext2fs_generic_bitmap *ret);
extern errcode_t ext2fs_copy_generic_bitmap(ext2fs_generic_bitmap src,
                                        ext2fs_generic_bitmap *dest);
extern void ext2fs_clear_generic_bitmap(ext2fs_generic_bitmap bitmap);
extern errcode_t ext2fs_fudge_generic_bitmap_end(ext2fs_inode_bitmap bitmap,
                                           errcode_t magic,
                                           errcode_t neq,
                                           ext2_ino_t end,
                                           ext2_ino_t *oend);
extern void ext2fs_set_generic_bitmap_padding(ext2fs_generic_bitmap map);
extern errcode_t ext2fs_resize_generic_bitmap(errcode_t magic,
                                        __u32 new_end,
                                        __u32 new_real_end,
                                        ext2fs_generic_bitmap bmap);
extern errcode_t ext2fs_compare_generic_bitmap(errcode_t magic, errcode_t neq,
                                         ext2fs_generic_bitmap bm1,
                                         ext2fs_generic_bitmap bm2);
extern errcode_t ext2fs_get_generic_bitmap_range(ext2fs_generic_bitmap bmap,
                                           errcode_t magic,
                                           __u32 start, __u32 num,
                                           void *out);
extern errcode_t ext2fs_set_generic_bitmap_range(ext2fs_generic_bitmap bmap,
                                           errcode_t magic,
                                           __u32 start, __u32 num,
                                           void *in);
extern errcode_t ext2fs_find_first_zero_generic_bitmap(ext2fs_generic_bitmap bitmap,
                                                 __u32 start, __u32 end,
                                                 __u32 *out);

/* gen_bitmap64.c */

/* Generate and print bitmap usage statistics */
#define BMAP_STATS

void ext2fs_free_generic_bmap(ext2fs_generic_bitmap bmap);
errcode_t ext2fs_alloc_generic_bmap(ext2_filsys fs, errcode_t magic,
                                    int type, __u64 start, __u64 end,
                                    __u64 real_end,
                                    const char *descr,
                                    ext2fs_generic_bitmap *ret);
errcode_t ext2fs_copy_generic_bmap(ext2fs_generic_bitmap src,
                                   ext2fs_generic_bitmap *dest);
void ext2fs_clear_generic_bmap(ext2fs_generic_bitmap bitmap);
errcode_t ext2fs_fudge_generic_bmap_end(ext2fs_generic_bitmap bitmap,
                                        errcode_t neq,
                                        __u64 end, __u64 *oend);
void ext2fs_set_generic_bmap_padding(ext2fs_generic_bitmap bmap);
errcode_t ext2fs_resize_generic_bmap(ext2fs_generic_bitmap bmap,
                                     __u64 new_end,
                                     __u64 new_real_end);
errcode_t ext2fs_compare_generic_bmap(errcode_t neq,
                                      ext2fs_generic_bitmap bm1,
                                      ext2fs_generic_bitmap bm2);
errcode_t ext2fs_get_generic_bmap_range(ext2fs_generic_bitmap bmap,
                                        __u64 start, unsigned int num,
                                        void *out);
errcode_t ext2fs_set_generic_bmap_range(ext2fs_generic_bitmap bmap,
                                        __u64 start, unsigned int num,
                                        void *in);
errcode_t ext2fs_convert_subcluster_bitmap(ext2_filsys fs,
                                           ext2fs_block_bitmap *bitmap);

/* getsize.c */
extern errcode_t ext2fs_get_device_size(const char *file, int blocksize,
                                        blk_t *retblocks);
extern errcode_t ext2fs_get_device_size2(const char *file, int blocksize,
                                        blk64_t *retblocks);

/* getsectsize.c */
extern int ext2fs_get_dio_alignment(int fd);
errcode_t ext2fs_get_device_sectsize(const char *file, int *sectsize);
errcode_t ext2fs_get_device_phys_sectsize(const char *file, int *sectsize);

/* i_block.c */
errcode_t ext2fs_iblk_add_blocks(ext2_filsys fs, struct ext2_inode *inode,
                                 blk64_t num_blocks);
errcode_t ext2fs_iblk_sub_blocks(ext2_filsys fs, struct ext2_inode *inode,
                                 blk64_t num_blocks);
errcode_t ext2fs_iblk_set(ext2_filsys fs, struct ext2_inode *inode, blk64_t b);

/* imager.c */
extern errcode_t ext2fs_image_inode_write(ext2_filsys fs, int fd, int flags);
extern errcode_t ext2fs_image_inode_read(ext2_filsys fs, int fd, int flags);
extern errcode_t ext2fs_image_super_write(ext2_filsys fs, int fd, int flags);
extern errcode_t ext2fs_image_super_read(ext2_filsys fs, int fd, int flags);
extern errcode_t ext2fs_image_bitmap_write(ext2_filsys fs, int fd, int flags);
extern errcode_t ext2fs_image_bitmap_read(ext2_filsys fs, int fd, int flags);

/* ind_block.c */
errcode_t ext2fs_read_ind_block(ext2_filsys fs, blk_t blk, void *buf);
errcode_t ext2fs_write_ind_block(ext2_filsys fs, blk_t blk, void *buf);
```

```
/* initialize.c */
extern errcode_t ext2fs_initialize(const char *name, int flags,
                                   struct ext2_super_block *param,
                                   io_manager manager, ext2_filsys *ret_fs);

/* icount.c */
extern void ext2fs_free_icount(ext2_icount_t icount);
extern errcode_t ext2fs_create_icount_tdb(ext2_filsys fs, char *tdb_dir,
                                          int flags, ext2_icount_t *ret);
extern errcode_t ext2fs_create_icount2(ext2_filsys fs, int flags,
                                       unsigned int size,
                                       ext2_icount_t hint, ext2_icount_t *ret);
extern errcode_t ext2fs_create_icount(ext2_filsys fs, int flags,
                                      unsigned int size,
                                      ext2_icount_t *ret);
extern errcode_t ext2fs_icount_fetch(ext2_icount_t icount, ext2_ino_t ino,
                                     __u16 *ret);
extern errcode_t ext2fs_icount_increment(ext2_icount_t icount, ext2_ino_t ino,
                                         __u16 *ret);
extern errcode_t ext2fs_icount_decrement(ext2_icount_t icount, ext2_ino_t ino,
                                         __u16 *ret);
extern errcode_t ext2fs_icount_store(ext2_icount_t icount, ext2_ino_t ino,
                                     __u16 count);
extern ext2_ino_t ext2fs_get_icount_size(ext2_icount_t icount);
errcode_t ext2fs_icount_validate(ext2_icount_t icount, FILE *);

/* inline.c */

extern errcode_t ext2fs_get_memalign(unsigned long size,
                                     unsigned long align, void *ptr);

/* inode.c */
extern errcode_t ext2fs_flush_icache(ext2_filsys fs);
extern errcode_t ext2fs_get_next_inode_full(ext2_inode_scan scan,
                                            ext2_ino_t *ino,
                                            struct ext2_inode *inode,
                                            int bufsize);
extern errcode_t ext2fs_open_inode_scan(ext2_filsys fs, int buffer_blocks,
                                ext2_inode_scan *ret_scan);
extern void ext2fs_close_inode_scan(ext2_inode_scan scan);
extern errcode_t ext2fs_get_next_inode(ext2_inode_scan scan, ext2_ino_t *ino,
                               struct ext2_inode *inode);
extern errcode_t ext2fs_inode_scan_goto_blockgroup(ext2_inode_scan scan,
                                                   int  group);
extern void ext2fs_set_inode_callback
        (ext2_inode_scan scan,
         errcode_t (*done_group)(ext2_filsys fs,
                                 ext2_inode_scan scan,
                                 dgrp_t group,
                                 void * priv_data),
         void *done_group_data);
extern int ext2fs_inode_scan_flags(ext2_inode_scan scan, int set_flags,
                                   int clear_flags);
extern errcode_t ext2fs_read_inode_full(ext2_filsys fs, ext2_ino_t ino,
                                        struct ext2_inode * inode,
                                        int bufsize);
extern errcode_t ext2fs_read_inode (ext2_filsys fs, ext2_ino_t ino,
                            struct ext2_inode * inode);
extern errcode_t ext2fs_write_inode_full(ext2_filsys fs, ext2_ino_t ino,
                                         struct ext2_inode * inode,
                                         int bufsize);
extern errcode_t ext2fs_write_inode(ext2_filsys fs, ext2_ino_t ino,
                            struct ext2_inode * inode);
extern errcode_t ext2fs_write_new_inode(ext2_filsys fs, ext2_ino_t ino,
                            struct ext2_inode * inode);
extern errcode_t ext2fs_get_blocks(ext2_filsys fs, ext2_ino_t ino, blk_t *blocks);
extern errcode_t ext2fs_check_directory(ext2_filsys fs, ext2_ino_t ino);

/* inode_io.c */
extern io_manager inode_io_manager;
extern errcode_t ext2fs_inode_io_intern(ext2_filsys fs, ext2_ino_t ino,
                                        char **name);
extern errcode_t ext2fs_inode_io_intern2(ext2_filsys fs, ext2_ino_t ino,
                                         struct ext2_inode *inode,
                                         char **name);

/* ismounted.c */
extern errcode_t ext2fs_check_if_mounted(const char *file, int *mount_flags);
extern errcode_t ext2fs_check_mount_point(const char *device, int *mount_flags,
                                          char *mtpt, int mtlen);

/* punch.c */
/*
 * NOTE: This function removes from an inode the blocks "start", "end", and
 * every block in between.
 */
extern errcode_t ext2fs_punch(ext2_filsys fs, ext2_ino_t ino,
                              struct ext2_inode *inode,
                              char *block_buf, blk64_t start,
                              blk64_t end);

/* namei.c */
extern errcode_t ext2fs_lookup(ext2_filsys fs, ext2_ino_t dir, const char *name,
                         int namelen, char *buf, ext2_ino_t *inode);
extern errcode_t ext2fs_namei(ext2_filsys fs, ext2_ino_t root, ext2_ino_t cwd,
                        const char *name, ext2_ino_t *inode);
errcode_t ext2fs_namei_follow(ext2_filsys fs, ext2_ino_t root, ext2_ino_t cwd,
                              const char *name, ext2_ino_t *inode);
```

```
extern errcode_t ext2fs_follow_link(ext2_filsys fs, ext2_ino_t root, ext2_ino_t cwd,
                        ext2_ino_t inode, ext2_ino_t *res_inode);

/* native.c */
int ext2fs_native_flag(void);

/* newdir.c */
extern errcode_t ext2fs_new_dir_block(ext2_filsys fs, ext2_ino_t dir_ino,
                              ext2_ino_t parent_ino, char **block);

/* mkdir.c */
extern errcode_t ext2fs_mkdir(ext2_filsys fs, ext2_ino_t parent, ext2_ino_t inum,
                              const char *name);

/* mkjournal.c */
extern errcode_t ext2fs_zero_blocks(ext2_filsys fs, blk_t blk, int num,
                                    blk_t *ret_blk, int *ret_count);
extern errcode_t ext2fs_zero_blocks2(ext2_filsys fs, blk64_t blk, int num,
                                     blk64_t *ret_blk, int *ret_count);
extern errcode_t ext2fs_create_journal_superblock(ext2_filsys fs,
                                                  __u32 num_blocks, int flags,
                                                  char  **ret_jsb);
extern errcode_t ext2fs_add_journal_device(ext2_filsys fs,
                                           ext2_filsys journal_dev);
extern errcode_t ext2fs_add_journal_inode(ext2_filsys fs, blk_t num_blocks,
                                          int flags);
extern int ext2fs_default_journal_size(__u64 num_blocks);

/* openfs.c */
extern errcode_t ext2fs_open(const char *name, int flags, int superblock,
                             unsigned int block_size, io_manager manager,
                             ext2_filsys *ret_fs);
extern errcode_t ext2fs_open2(const char *name, const char *io_options,
                              int flags, int superblock,
                              unsigned int block_size, io_manager manager,
                              ext2_filsys *ret_fs);
extern blk64_t ext2fs_descriptor_block_loc2(ext2_filsys fs,
                                            blk64_t group_block, dgrp_t i);
extern blk_t ext2fs_descriptor_block_loc(ext2_filsys fs, blk_t group_block,
                                         dgrp_t i);
errcode_t ext2fs_get_data_io(ext2_filsys fs, io_channel *old_io);
errcode_t ext2fs_set_data_io(ext2_filsys fs, io_channel new_io);
errcode_t ext2fs_rewrite_to_io(ext2_filsys fs, io_channel new_io);

/* get_pathname.c */
extern errcode_t ext2fs_get_pathname(ext2_filsys fs, ext2_ino_t dir, ext2_ino_t ino,
                                     char **name);

/* link.c */
errcode_t ext2fs_link(ext2_filsys fs, ext2_ino_t dir, const char *name,
                      ext2_ino_t ino, int flags);
errcode_t ext2fs_unlink(ext2_filsys fs, ext2_ino_t dir, const char *name,
                        ext2_ino_t ino, int flags);

/* symlink.c */
errcode_t ext2fs_symlink(ext2_filsys fs, ext2_ino_t parent, ext2_ino_t ino,
                         const char *name, char *target);

/* mmp.c */
errcode_t ext2fs_mmp_read(ext2_filsys fs, blk64_t mmp_blk, void *buf);
errcode_t ext2fs_mmp_write(ext2_filsys fs, blk64_t mmp_blk, void *buf);
errcode_t ext2fs_mmp_clear(ext2_filsys fs);
errcode_t ext2fs_mmp_init(ext2_filsys fs);
errcode_t ext2fs_mmp_start(ext2_filsys fs);
errcode_t ext2fs_mmp_update(ext2_filsys fs);
errcode_t ext2fs_mmp_stop(ext2_filsys fs);
unsigned ext2fs_mmp_new_seq(void);

/* read_bb.c */
extern errcode_t ext2fs_read_bb_inode(ext2_filsys fs,
                                      ext2_badblocks_list *bb_list);

/* read_bb_file.c */
extern errcode_t ext2fs_read_bb_FILE2(ext2_filsys fs, FILE *f,
                                      ext2_badblocks_list *bb_list,
                                      void *priv_data,
                                      void (*invalid)(ext2_filsys fs,
                                                      blk_t blk,
                                                      char *badstr,
                                                      void *priv_data));
extern errcode_t ext2fs_read_bb_FILE(ext2_filsys fs, FILE *f,
                                     ext2_badblocks_list *bb_list,
                                     void (*invalid)(ext2_filsys fs,
                                                     blk_t blk));

/* res_gdt.c */
extern errcode_t ext2fs_create_resize_inode(ext2_filsys fs);

/* swapfs.c */
extern void ext2fs_swap_ext_attr(char *to, char *from, int bufsize,
                                 int has_header);
extern void ext2fs_swap_ext_attr_header(struct ext2_ext_attr_header *to_header,
                                        struct ext2_ext_attr_header *from_hdr);
extern void ext2fs_swap_ext_attr_entry(struct ext2_ext_attr_entry *to_entry,
                                       struct ext2_ext_attr_entry *from_entry);
extern void ext2fs_swap_super(struct ext2_super_block * super);
extern void ext2fs_swap_group_desc(struct ext2_group_desc *gdp);
extern void ext2fs_swap_group_desc2(ext2_filsys, struct ext2_group_desc *gdp);
```

```
extern void ext2fs_swap_inode_full(ext2_filsys fs, struct ext2_inode_large *t,
                                   struct ext2_inode_large *f, int hostorder,
                                   int bufsize);
extern void ext2fs_swap_inode(ext2_filsys fs,struct ext2_inode *t,
                              struct ext2_inode *f, int hostorder);
extern void ext2fs_swap_mmp(struct mmp_struct *mmp);

/* unix_io.c */
extern int ext2fs_open_file(const char *pathname, int flags, mode_t mode);
extern int ext2fs_stat(const char *path, ext2fs_struct_stat *buf);
extern int ext2fs_fstat(int fd, ext2fs_struct_stat *buf);

/* valid_blk.c */
extern int ext2fs_inode_has_valid_blocks(struct ext2_inode *inode);
extern int ext2fs_inode_has_valid_blocks2(ext2_filsys fs,
                                          struct ext2_inode *inode);

/* version.c */
extern int ext2fs_parse_version_string(const char *ver_string);
extern int ext2fs_get_library_version(const char **ver_string,
                                      const char **date_string);

/* write_bb_file.c */
extern errcode_t ext2fs_write_bb_FILE(ext2_badblocks_list bb_list,
                                      unsigned int flags,
                                      FILE *f);


/* inline functions */
#ifdef NO_INLINE_FUNCS
extern errcode_t ext2fs_get_mem(unsigned long size, void *ptr);
extern errcode_t ext2fs_get_memzero(unsigned long size, void *ptr);
extern errcode_t ext2fs_get_array(unsigned long count,
                                  unsigned long size, void *ptr);
extern errcode_t ext2fs_get_arrayzero(unsigned long count,
                                      unsigned long size, void *ptr);
extern errcode_t ext2fs_free_mem(void *ptr);
extern errcode_t ext2fs_resize_mem(unsigned long old_size,
                                   unsigned long size, void *ptr);
extern void ext2fs_mark_super_dirty(ext2_filsys fs);
extern void ext2fs_mark_changed(ext2_filsys fs);
extern int ext2fs_test_changed(ext2_filsys fs);
extern void ext2fs_mark_valid(ext2_filsys fs);
extern void ext2fs_unmark_valid(ext2_filsys fs);
extern int ext2fs_test_valid(ext2_filsys fs);
extern void ext2fs_mark_ib_dirty(ext2_filsys fs);
extern void ext2fs_mark_bb_dirty(ext2_filsys fs);
extern int ext2fs_test_ib_dirty(ext2_filsys fs);
extern int ext2fs_test_bb_dirty(ext2_filsys fs);
extern dgrp_t ext2fs_group_of_blk(ext2_filsys fs, blk_t blk);
extern dgrp_t ext2fs_group_of_ino(ext2_filsys fs, ext2_ino_t ino);
extern blk_t ext2fs_group_first_block(ext2_filsys fs, dgrp_t group);
extern blk_t ext2fs_group_last_block(ext2_filsys fs, dgrp_t group);
extern blk_t ext2fs_inode_data_blocks(ext2_filsys fs,
                                      struct ext2_inode *inode);
extern unsigned int ext2fs_div_ceil(unsigned int a, unsigned int b);
extern __u64 ext2fs_div64_ceil(__u64 a, __u64 b);
#endif

/*
 * The actual inlined functions definitions themselves...
 *
 * If NO_INLINE_FUNCS is defined, then we won't try to do inline
 * functions at all!
 */
#if (defined(INCLUDE_INLINE_FUNCS) || !defined(NO_INLINE_FUNCS))
#ifdef INCLUDE_INLINE_FUNCS
#define _INLINE_ extern
#else
#if (__STDC_VERSION__ >= 199901L)
#define _INLINE_ inline
#else
#ifdef __GNUC__
#define _INLINE_ extern __inline__
#else                           /* For Watcom C */
#define _INLINE_ extern inline
#endif /* __GNUC__ */
#endif /* __STDC_VERSION__ >= 199901L */
#endif

#ifndef EXT2_CUSTOM_MEMORY_ROUTINES
#include <string.h>
/*
 *  Allocate memory.  The 'ptr' arg must point to a pointer.
 */
_INLINE_ errcode_t ext2fs_get_mem(unsigned long size, void *ptr)
{
        void *pp;

        pp = malloc(size);
        if (!pp)
                return EXT2_ET_NO_MEMORY;
        memcpy(ptr, &pp, sizeof (pp));
        return 0;
}

_INLINE_ errcode_t ext2fs_get_memzero(unsigned long size, void *ptr)
{
```

```
        void *pp;

        pp = malloc(size);
        if (!pp)
                return EXT2_ET_NO_MEMORY;
        memset(pp, 0, size);
        memcpy(ptr, &pp, sizeof(pp));
        return 0;
}

_INLINE_ errcode_t ext2fs_get_array(unsigned long count, unsigned long size, void *ptr)
{
        if (count && (-1UL)/count<size)
                return EXT2_ET_NO_MEMORY;
        return ext2fs_get_mem(count*size, ptr);
}

_INLINE_ errcode_t ext2fs_get_arrayzero(unsigned long count,
                                        unsigned long size, void *ptr)
{
        void *pp;

        if (count && (-1UL)/count<size)
                return EXT2_ET_NO_MEMORY;
        pp = calloc(count, size);
        if (!pp)
                return EXT2_ET_NO_MEMORY;
        memcpy(ptr, &pp, sizeof(pp));
        return 0;
}

/*
 * Free memory.  The 'ptr' arg must point to a pointer.
 */
_INLINE_ errcode_t ext2fs_free_mem(void *ptr)
{
        void *p;

        memcpy(&p, ptr, sizeof(p));
        free(p);
        p = 0;
        memcpy(ptr, &p, sizeof(p));
        return 0;
}

/*
 *  Resize memory.  The 'ptr' arg must point to a pointer.
 */
_INLINE_ errcode_t ext2fs_resize_mem(unsigned long EXT2FS_ATTR((unused)) old_size,
                                     unsigned long size, void *ptr)
{
        void *p;

        /* Use "memcpy" for pointer assignments here to avoid problems
         * with C99 strict type aliasing rules. */
        memcpy(&p, ptr, sizeof(p));
        p = realloc(p, size);
        if (!p)
                return EXT2_ET_NO_MEMORY;
        memcpy(ptr, &p, sizeof(p));
        return 0;
}
#endif  /* Custom memory routines */

/*
 * Mark a filesystem superblock as dirty
 */
_INLINE_ void ext2fs_mark_super_dirty(ext2_filsys fs)
{
        fs->flags |= EXT2_FLAG_DIRTY | EXT2_FLAG_CHANGED;
}

/*
 * Mark a filesystem as changed
 */
_INLINE_ void ext2fs_mark_changed(ext2_filsys fs)
{
        fs->flags |= EXT2_FLAG_CHANGED;
}

/*
 * Check to see if a filesystem has changed
 */
_INLINE_ int ext2fs_test_changed(ext2_filsys fs)
{
        return (fs->flags & EXT2_FLAG_CHANGED);
}

/*
 * Mark a filesystem as valid
 */
_INLINE_ void ext2fs_mark_valid(ext2_filsys fs)
{
        fs->flags |= EXT2_FLAG_VALID;
}

/*
 * Mark a filesystem as NOT valid
```

```
 */
_INLINE_ void ext2fs_unmark_valid(ext2_filsys fs)
{
        fs->flags &= ~EXT2_FLAG_VALID;
}

/*
 * Check to see if a filesystem is valid
 */
_INLINE_ int ext2fs_test_valid(ext2_filsys fs)
{
        return (fs->flags & EXT2_FLAG_VALID);
}

/*
 * Mark the inode bitmap as dirty
 */
_INLINE_ void ext2fs_mark_ib_dirty(ext2_filsys fs)
{
        fs->flags |= EXT2_FLAG_IB_DIRTY | EXT2_FLAG_CHANGED;
}

/*
 * Mark the block bitmap as dirty
 */
_INLINE_ void ext2fs_mark_bb_dirty(ext2_filsys fs)
{
        fs->flags |= EXT2_FLAG_BB_DIRTY | EXT2_FLAG_CHANGED;
}

/*
 * Check to see if a filesystem's inode bitmap is dirty
 */
_INLINE_ int ext2fs_test_ib_dirty(ext2_filsys fs)
{
        return (fs->flags & EXT2_FLAG_IB_DIRTY);
}

/*
 * Check to see if a filesystem's block bitmap is dirty
 */
_INLINE_ int ext2fs_test_bb_dirty(ext2_filsys fs)
{
        return (fs->flags & EXT2_FLAG_BB_DIRTY);
}

/*
 * Return the group # of a block
 */
_INLINE_ dgrp_t ext2fs_group_of_blk(ext2_filsys fs, blk_t blk)
{
        return ext2fs_group_of_blk2(fs, blk);
}
/*
 * Return the group # of an inode number
 */
_INLINE_ dgrp_t ext2fs_group_of_ino(ext2_filsys fs, ext2_ino_t ino)
{
        return (ino - 1) / fs->super->s_inodes_per_group;
}

/*
 * Return the first block (inclusive) in a group
 */
_INLINE_ blk_t ext2fs_group_first_block(ext2_filsys fs, dgrp_t group)
{
        return (blk_t) ext2fs_group_first_block2(fs, group);
}

/*
 * Return the last block (inclusive) in a group
 */
_INLINE_ blk_t ext2fs_group_last_block(ext2_filsys fs, dgrp_t group)
{
        return (blk_t) ext2fs_group_last_block2(fs, group);
}

_INLINE_ blk_t ext2fs_inode_data_blocks(ext2_filsys fs,
                                        struct ext2_inode *inode)
{
        return (blk_t) ext2fs_inode_data_blocks2(fs, inode);
}

/*
 * This is an efficient, overflow safe way of calculating ceil((1.0 * a) / b)
 */
_INLINE_ unsigned int ext2fs_div_ceil(unsigned int a, unsigned int b)
{
        if (!a)
                return 0;
        return ((a - 1) / b) + 1;
}

_INLINE_ __u64 ext2fs_div64_ceil(__u64 a, __u64 b)
{
        if (!a)
                return 0;
        return ((a - 1) / b) + 1;
```

```
}

#undef _INLINE_
#endif

#ifdef __cplusplus
}
#endif

#endif /* _EXT2FS_EXT2FS_H */
```

Notice for package(s)

e2fsprogs

```
/*
 * e2p.h --- header file for the e2p library
 *
 * %Begin-Header%
 * This file may be redistributed under the terms of the GNU Library
 * General Public License, version 2.
 * %End-Header%
 */

#include <sys/types.h>          /* Needed by dirent.h on netbsd */
#include <stdio.h>
#include <dirent.h>

#include <ext2fs/ext2_fs.h>

#define E2P_FEATURE_COMPAT      0
#define E2P_FEATURE_INCOMPAT    1
#define E2P_FEATURE_RO_INCOMPAT 2
#define E2P_FEATURE_TYPE_MASK   0x03

#define E2P_FEATURE_NEGATE_FLAG 0x80

#define E2P_FS_FEATURE          0
#define E2P_JOURNAL_FEATURE     1

/* `options' for print_flags() */

#define PFOPT_LONG  1 /* Must be 1 for compatibility with `int long_format'. */


int fgetflags (const char * name, unsigned long * flags);
int fgetversion (const char * name, unsigned long * version);
int fsetflags (const char * name, unsigned long flags);
int fsetversion (const char * name, unsigned long version);
int getflags (int fd, unsigned long * flags);
int getversion (int fd, unsigned long * version);
int iterate_on_dir (const char * dir_name,
                    int (*func) (const char *, struct dirent *, void *),
                    void * private);
void list_super(struct ext2_super_block * s);
void list_super2(struct ext2_super_block * s, FILE *f);
void print_fs_errors (FILE * f, unsigned short errors);
void print_flags (FILE * f, unsigned long flags, unsigned options);
void print_fs_state (FILE * f, unsigned short state);
int setflags (int fd, unsigned long flags);
int setversion (int fd, unsigned long version);

const char *e2p_feature2string(int compat, unsigned int mask);
const char *e2p_jrnl_feature2string(int compat, unsigned int mask);
int e2p_string2feature(char *string, int *compat, unsigned int *mask);
int e2p_jrnl_string2feature(char *string, int *compat_type, unsigned int *mask);
int e2p_edit_feature(const char *str, __u32 *compat_array, __u32 *ok_array);
int e2p_edit_feature2(const char *str, __u32 *compat_array, __u32 *ok_array,
                      __u32 *clear_ok_array, int *type_err,
                      unsigned int *mask_err);

int e2p_is_null_uuid(void *uu);
void e2p_uuid_to_str(void *uu, char *out);
const char *e2p_uuid2str(void *uu);

const char *e2p_hash2string(int num);
int e2p_string2hash(char *string);

const char *e2p_mntopt2string(unsigned int mask);
int e2p_string2mntopt(char *string, unsigned int *mask);
int e2p_edit_mntopts(const char *str, __u32 *mntopts, __u32 ok);

unsigned long parse_num_blocks(const char *arg, int log_block_size);
unsigned long long parse_num_blocks2(const char *arg, int log_block_size);

char *e2p_os2string(int os_type);
int e2p_string2os(char *str);

unsigned int e2p_percent(int percent, unsigned int base);
```

Notice for package(s)

e2fsprogs

```
/*
 * Public include file for the UUID library
 *
 * Copyright (C) 1996, 1997, 1998 Theodore Ts'o.
 *
 * %Begin-Header%
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, and the entire permission notice in its entirety,
 *    including the disclaimer of warranties.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote
 *    products derived from this software without specific prior
 *    written permission.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF
 * WHICH ARE HEREBY DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
 * OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
 * USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 * %End-Header%
 */

#ifndef _UUID_UUID_H
#define _UUID_UUID_H

#include <sys/types.h>
#ifndef _WIN32
#include <sys/time.h>
#endif
#include <time.h>

typedef unsigned char uuid_t[16];

/* UUID Variant definitions */
#define UUID_VARIANT_NCS        0
#define UUID_VARIANT_DCE        1
#define UUID_VARIANT_MICROSOFT  2
#define UUID_VARIANT_OTHER      3

/* UUID Type definitions */
#define UUID_TYPE_DCE_TIME   1
#define UUID_TYPE_DCE_RANDOM 4

/* Allow UUID constants to be defined */
#ifdef __GNUC__
#define UUID_DEFINE(name,u0,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15) \
        static const uuid_t name __attribute__ ((unused)) = {u0,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15}
#else
#define UUID_DEFINE(name,u0,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15) \
        static const uuid_t name = {u0,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15}
#endif

#ifdef __cplusplus
extern "C" {
#endif

/* clear.c */
void uuid_clear(uuid_t uu);

/* compare.c */
int uuid_compare(const uuid_t uu1, const uuid_t uu2);

/* copy.c */
void uuid_copy(uuid_t dst, const uuid_t src);

/* gen_uuid.c */
void uuid_generate(uuid_t out);
void uuid_generate_random(uuid_t out);
void uuid_generate_time(uuid_t out);

/* isnull.c */
int uuid_is_null(const uuid_t uu);

/* parse.c */
int uuid_parse(const char *in, uuid_t uu);

/* unparse.c */
void uuid_unparse(const uuid_t uu, char *out);
```

```
void uuid_unparse_lower(const uuid_t uu, char *out);
void uuid_unparse_upper(const uuid_t uu, char *out);

/* uuid_time.c */
time_t uuid_time(const uuid_t uu, struct timeval *ret_tv);
int uuid_type(const uuid_t uu);
int uuid_variant(const uuid_t uu);

#ifdef __cplusplus
}
#endif

#endif /* _UUID_UUID_H */
```

Notice for package(s)

e2fsprogs
util-linux

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, and the entire permission notice in its entirety,
   including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior
   written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF
WHICH ARE HEREBY DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

Notice for package(s)

e2fsprogs

```
/*
 * Copyright 1987 by MIT Student Information Processing Board
 *
 * Permission to use, copy, modify, and distribute this software and
 * its documentation for any purpose is hereby granted, provided that
 * the names of M.I.T. and the M.I.T. S.I.P.B. not be used in
 * advertising or publicity pertaining to distribution of the software
 * without specific, written prior permission.  M.I.T. and the
 * M.I.T. S.I.P.B. make no representations about the suitability of
 * this software for any purpose.  It is provided "as is" without
 * express or implied warranty.
 */

#include "config.h"
#include "com_err.h"
#include "error_table.h"
#include "internal.h"

static const char char_set[] =
        "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_";

static char buf[6];

const char * error_table_name(errcode_t num)
{
    int ch;
    int i;
    char *p;

    /* num = aa aaa abb bbb bcc ccc cdd ddd d?? ??? ??? */
    p = buf;
    num >>= ERRCODE_RANGE;
    /* num = ?? ??? ??? aaa aaa bbb bbb ccc ccc ddd ddd */
    num &= 077777777L;
    /* num = 00 000 000 aaa aaa bbb bbb ccc ccc ddd ddd */
    for (i = 4; i >= 0; i--) {
```

```
        ch = (int)((num >> BITS_PER_CHAR * i) & ((1 << BITS_PER_CHAR) - 1));
        if (ch != 0)
            *p++ = char_set[ch-1];
    }
    *p = '\0';
    return(buf);
}
```

Notice for package(s)

e2fsprogs

```
/*
 * Copyright 1987, 1988 by MIT Student Information Processing Board
 *
 * Permission to use, copy, modify, and distribute this software and
 * its documentation for any purpose is hereby granted, provided that
 * the names of M.I.T. and the M.I.T. S.I.P.B. not be used in
 * advertising or publicity pertaining to distribution of the software
 * without specific, written prior permission.  M.I.T. and the
 * M.I.T. S.I.P.B. make no representations about the suitability of
 * this software for any purpose.  It is provided "as is" without
 * express or implied warranty.
 *
 * This quote is just too good to not pass on:
 *
 *      "BTW, I would have rejected the name Story Server because its
 *      initials are SS, the name of the secret police in Nazi
 *      Germany, probably the most despised pair of letters in western
 *      culture."  --- http://scriptingnewsarchive.userland.com/1999/12/13
 *
 * Let no one say political correctness isn't dead....
 */

#ifndef _ss_h
#define _ss_h __FILE__

#include <ss/ss_err.h>

#define __SS_CONST const
#define __SS_PROTO (int, const char * const *, int, void *)

#ifdef __GNUC__
#define __SS_ATTR(x) __attribute__(x)
#else
#define __SS_ATTR(x)
#endif


typedef __SS_CONST struct _ss_request_entry {
    __SS_CONST char * __SS_CONST *command_names; /* whatever */
    void (* __SS_CONST function) __SS_PROTO; /* foo */
    __SS_CONST char * __SS_CONST info_string;   /* NULL */
    int flags;                  /* 0 */
} ss_request_entry;

typedef __SS_CONST struct _ss_request_table {
    int version;
    ss_request_entry *requests;
} ss_request_table;

#define SS_RQT_TBL_V2   2

typedef struct _ss_rp_options { /* DEFAULT VALUES */
    int version;                /* SS_RP_V1 */
    void (*unknown) __SS_PROTO; /* call for unknown command */
    int allow_suspend;
    int catch_int;
} ss_rp_options;

#define SS_RP_V1 1

#define SS_OPT_DONT_LIST        0x0001
#define SS_OPT_DONT_SUMMARIZE   0x0002

void ss_help __SS_PROTO;
#if 0
char *ss_current_request();     /* This is actually a macro */
#endif

char *ss_name(int sci_idx);
void ss_error (int, long, char const *, ...)
        __SS_ATTR((format(printf, 3, 4)));
void ss_perror (int, long, char const *);

int ss_create_invocation(const char *, const char *, void *,
                         ss_request_table *, int *);
void ss_delete_invocation(int);
int ss_listen(int);
int ss_execute_line(int, char *);
void ss_add_request_table(int, ss_request_table *, int, int *);
```

```
void ss_delete_request_table(int, ss_request_table *, int *);
void ss_abort_subsystem(int sci_idx, int code);
void ss_quit(int argc, const char * const *argv, int sci_idx, void *infop);
void ss_self_identify(int argc, const char * const *argv, int sci_idx, void *infop);
void ss_subsystem_name(int argc, const char * const *argv,
                       int sci_idx, void *infop);
void ss_subsystem_version(int argc, const char * const *argv,
                          int sci_idx, void *infop);
void ss_unimplemented(int argc, const char * const *argv,
                      int sci_idx, void *infop);
void ss_set_prompt(int sci_idx, char *new_prompt);
char *ss_get_prompt(int sci_idx);
void ss_get_readline(int sci_idx);
char *ss_safe_getenv(const char *arg);

extern ss_request_table ss_std_requests;
#endif /* _ss_h */
```

Notice for package(s)

e2fsprogs
glib-2.0
procps

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

[This is the first released version of the library GPL. It is

 numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other lib

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have th

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave yo

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permissior

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. T

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, bec

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the librar

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work base

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party say

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library"

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all th

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspi

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute s

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facili
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Ther

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be r

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to ex

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do thi

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subseque

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the ter

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked w

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contair

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a deriva

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any execu

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work conta

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the
b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, ab
c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified
d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the ope

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library faciliti

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must b
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find th
8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt oth

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distrib

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the origina

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conc

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to a

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the origi

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new ver

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that ever

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effect

one line to give the library's name and an idea of what it does.
Copyright (C) year name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the
Free Software Foundation, Inc., 51 Franklin St, Fifth Floor,
Boston, MA 02110-1301, USA.
Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if r

Yoyodyne, Inc., hereby disclaims all copyright interest in
the library `Frob' (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
That's all there is to it!

Notice for package(s)

busybox

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included
in its entirety, below). Version 2 is the only version of this license which
this version of BusyBox (or modified versions derived from this one) may be
distributed under.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

```
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA


Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year  name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

busybox
bzip2

```
--------------------------------------------------------------------------

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2010 Julian R Seward.  All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
   not claim that you wrote the original software.  If you use this
   software in a product, an acknowledgment in the product
   documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
   not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org
```

bzip2/libbzip2 version 1.0.6 of 6 September 2010

Notice for package(s)

modutils-initscripts

Public Domain

Notice for package(s)

glib-2.0
modutils-initscripts
xz

This is a placeholder for the Public Domain License

Notice for package(s)

udev

                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
     51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the

Library.

  You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a

derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or

distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU

```
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

sysvinit

```
Sysvinit is Copyright (C) 1991-2004 Miquel van Smoorenburg

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301 USA

On Debian GNU/Linux systems, the complete text of the GNU General
Public License can be found in `/usr/share/common-licenses/GPL-2'.

Send patches to sysvinit-devel@nongnu.org

The of the start-stop-daemon

 * A rewrite of the original Debian's start-stop-daemon Perl script
 * in C (faster - it is executed many times during system startup).
 *
 * Written by Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl>,
 * public domain.
```

Notice for package(s)

init-ifupdown
netbase

```
This package was created by Peter Tobias tobias@et-inf.fho-emden.de on
Wed, 24 Aug 1994 21:33:28 +0200 and maintained by Anthony Towns
<ajt@debian.org> until 2001.
It is currently maintained by Marco d'Itri <md@linux.it>.

Copyright 1994-2010 Peter Tobias, Anthony Towns and Marco d'Itri

The programs in this package are distributed under the terms of the GNU
General Public License, version 2 as distributed by the Free Software
Foundation. On Debian systems, a copy of this license may be found in
/usr/share/common-licenses/GPL-2.
```

Notice for package(s)

gnutls
kmod

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
     51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

  Finally, software patents pose a constant threat to the existence of
any free program.  We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder.  Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License.  This license, the GNU Lesser
```

General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices
    stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no
    charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application. Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the

Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

  It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

  7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

    b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the

Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

```
           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

sysfsutils

```
The commands and utilities under the "test" directory are licensed under the
GNU General Public License (GPL) Version 2, June 1991. The full text of the
GPL is located at:

sysfsutils/cmd/GPL

The sysfs library is licensed under the GNU Lesser Public License (LGPL)
Version 2.1, February 1999. The full text of the LGPL is located at:

sysfsutils/lib/LGPL
```

Notice for package(s)

sysfsutils

```
      The GNU General Public License (GPL)
      Version 2, June 1991

      Copyright (C) 1989, 1991 Free Software Foundation, Inc.
      59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
      Everyone is permitted to copy and distribute verbatim copies
      of this license document, but changing it is not allowed.

      Preamble

      The licenses for most software are designed to take away your freedom to
      share and change it. By contrast, the GNU General Public License is
      intended to guarantee your freedom to share and change free software--to
      make sure the software is free for all its users. This General Public
      License applies to most of the Free Software Foundation's software and to
      any other program whose authors commit to using it. (Some other Free
      Software Foundation software is covered by the GNU Library General Public
      License instead.) You can apply it to your programs, too.
```

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

  b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

  c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to

each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

  a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

  b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

  c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Notice for package(s)

sysfsutils

GNU Lesser Public License
Version 2.1, February 1999

  [This is the first released version of the Lesser GPL. It also counts
  as the successor of the GNU Library Public License, version 2, hence
  the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have

the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either

verbatim or with modifications and/or translated straightforwardly into
another language. (Hereinafter, translation is included without limitation
in the term "modification".)

"Source code" for a work means the preferred form of the work for making
modifications to it. For a library, complete source code means all the
source code for all modules it contains, plus any associated interface
definition files, plus the scripts used to control compilation and
installation of the library.

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of running a
program using the Library is not restricted, and output from such a
program is covered only if its contents constitute a work based on the
Library (independent of the use of the Library in a tool for writing it).
Whether that is true depends on what the Library does and what the program
that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the notices
that refer to this License and to the absence of any warranty; and
distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you
may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it,
thus forming a work based on the Library, and copy and distribute such
modifications or work under the terms of Section 1 above, provided that
you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices stating
  that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no charge to
  all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a table
  of data to be supplied by an application program that uses the facility,
  other than as an argument passed when the facility is invoked, then you
  must make a good faith effort to ensure that, in the event an
  application does not supply such function or table, the facility still
  operates, and performs whatever part of its purpose remains meaningful.

  (For example, a function in a library to compute square roots has a
  purpose that is entirely well-defined independent of the application.
  Therefore, Subsection 2d requires that any application-supplied function
  or table used by this function must be optional: if the application does
  not supply it, the square root function must still compute square
  roots.)

  These requirements apply to the modified work as a whole. If
  identifiable sections of that work are not derived from the Library, and
  can be reasonably considered independent and separate works in
  themselves, then this License, and its terms, do not apply to those
  sections when you distribute them as separate works. But when you
  distribute the same sections as part of a whole which is a work based on
  the Library, the distribution of the whole must be on the terms of this
  License, whose permissions for other licensees extend to the entire
  whole, and thus to each and every part regardless of who wrote it.

  Thus, it is not the intent of this section to claim rights or contest
  your rights to work written entirely by you; rather, the intent is to
  exercise the right to control the distribution of derivative or
  collective works based on the Library.

  In addition, mere aggregation of another work not based on the Library
  with the Library (or with a work based on the Library) on a volume of a
  storage or distribution medium does not bring the other work under the
  scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so that
they refer to the ordinary GNU General Public License, version 2, instead
of to this License. (If a newer version than version 2 of the ordinary GNU
General Public License has appeared, then you can specify that version
instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that
copy, so the ordinary GNU General Public License applies to all subsequent
copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the
Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of
it, under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you accompany it with the complete
corresponding machine-readable source code, which must be distributed
under the terms of Sections 1 and 2 above on a medium customarily used for
software interchange.

If distribution of object code is made by offering access to copy from a

designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

  a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

  b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

  c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

  d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

  e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library

and of the other library facilities is otherwise permitted, and provided that you do these two things:

  a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

  b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE
LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT
LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES
SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE
WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN
ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Notice for package(s)

run-postinsts
udev-extraconf
udev-rules-qoriq

Different components of OpenEmbedded are under different licenses (a mix
of MIT and GPLv2). Please see:

meta/COPYING.GPLv2 (GPLv2)
meta/COPYING.MIT (MIT)
meta-selftest/COPYING.MIT (MIT)
meta-skeleton/COPYING.MIT (MIT)

All metadata is MIT licensed unless otherwise stated. Source code
included in tree for individual recipes is under the LICENSE stated in
the associated recipe (.bb file) unless otherwise stated.

License information for any other files is either explicitly stated
or defaults to GPL version 2.

Notice for package(s)

bash
coreutils
dosfstools
elfutils
gawk
gmp
gnutls
gzip
readline
rsync
xz

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This

License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a

fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information. But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law. If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it. (Additional permissions may be written to require their own
removal in certain cases when you modify the work.) You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10. If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term. If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License. Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means

prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license

you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms,

reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program> Copyright (C) <year> <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

Notice for package(s)

bash
coreutils
dosfstools
elfutils
gawk
gdbm
gnutls
grep
gzip
m4
readline
rsync
sed

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, th

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have th

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have cer

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms tha

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License givir

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and au

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of softwa

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients”

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of a

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement ur

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of inter

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of pack

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work)

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. Y

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes i

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention i

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriatel

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
* b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This r
* c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will the
* d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive int

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and w

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corr

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Correspo
* b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written of
* c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is
* d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corres
* e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Sourc

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, o

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or update

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documer

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additior

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Ad

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders o

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
* b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Noti
* c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasc
* d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
* e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
* f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) w

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you receiv

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional te

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above
8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify i

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unl

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by s

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under thi
9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurrir
10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organi

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not imp
11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work t

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make,

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (su

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to c

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise b
12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not
13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under versic
14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions wi

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public Lice

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public stat

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyr
15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS
16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEY
17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewir

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it fre

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License

```
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's com

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necess

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library,
```

Notice for package(s)

mtd-utils
psmisc

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License.  The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language.  (Hereinafter, translation is included without limitation in
```

the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

  5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

  7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

Notice for package(s)

mtd-utils

```
/*
 * Copyright (c) Artem Bityutskiy, 2007, 2008
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
```

```
 * the GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#ifndef __MTD_UTILS_COMMON_H__
#define __MTD_UTILS_COMMON_H__

#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <features.h>
#include <inttypes.h>
#include "version.h"

#ifndef PROGRAM_NAME
# error "You must define PROGRAM_NAME before including this header"
#endif

#ifdef __cplusplus
extern "C" {
#endif

#ifndef MIN     /* some C lib headers define this for us */
#define MIN(a, b) ((a) < (b) ? (a) : (b))
#endif
#ifndef MAX
#define MAX(a, b) ((a) > (b) ? (a) : (b))
#endif
#define min(a, b) MIN(a, b) /* glue for linux kernel source */
#define ARRAY_SIZE(a) (sizeof(a) / sizeof((a)[0]))

#ifndef O_CLOEXEC
#define O_CLOEXEC 0
#endif

/* define a print format specifier for off_t */
#ifdef __USE_FILE_OFFSET64
#define PRIxoff_t PRIx64
#define PRIdoff_t PRId64
#else
#define PRIxoff_t "l"PRIx32
#define PRIdoff_t "l"PRId32
#endif

/* Verbose messages */
#define bareverbose(verbose, fmt, ...) do {                     \
        if (verbose)                                            \
                printf(fmt, ##__VA_ARGS__);                     \
} while(0)
#define verbose(verbose, fmt, ...) \
        bareverbose(verbose, "%s: " fmt "\n", PROGRAM_NAME, ##__VA_ARGS__)

/* Normal messages */
#define normsg_cont(fmt, ...) do {                              \
        printf("%s: " fmt, PROGRAM_NAME, ##__VA_ARGS__);        \
} while(0)
#define normsg(fmt, ...) do {                                   \
        normsg_cont(fmt "\n", ##__VA_ARGS__);                   \
} while(0)

/* Error messages */
#define errmsg(fmt, ...)  ({                                                \
        fprintf(stderr, "%s: error!: " fmt "\n", PROGRAM_NAME, ##__VA_ARGS__); \
        -1;                                                                 \
})
#define errmsg_die(fmt, ...) do {                                           \
        exit(errmsg(fmt, ##__VA_ARGS__));                                   \
} while(0)

/* System error messages */
#define sys_errmsg(fmt, ...)  ({                                            \
        int _err = errno;                                                   \
        errmsg(fmt, ##__VA_ARGS__);                                         \
        fprintf(stderr, "%*serror %d (%s)\n", (int)sizeof(PROGRAM_NAME) + 1,\
                "", _err, strerror(_err));                                  \
        -1;                                                                 \
})
#define sys_errmsg_die(fmt, ...) do {                                       \
        exit(sys_errmsg(fmt, ##__VA_ARGS__));                               \
} while(0)

/* Warnings */
#define warnmsg(fmt, ...) do {                                          \
        fprintf(stderr, "%s: warning!: " fmt "\n", PROGRAM_NAME, ##__VA_ARGS__); \
} while(0)

/**
 * prompt the user for confirmation
 */
static inline bool prompt(const char *msg, bool def)
{
```

```
        char *line = NULL;
        size_t len;
        bool ret = def;

        do {
                normsg_cont("%s (%c/%c) ", msg, def ? 'Y' : 'y', def ? 'n' : 'N');
                fflush(stdout);

                while (getline(&line, &len, stdin) == -1) {
                        printf("failed to read prompt; assuming '%s'\n",
                                def ? "yes" : "no");
                        break;
                }

                if (strcmp("\n", line) != 0) {
                        switch (rpmatch(line)) {
                        case 0: ret = false; break;
                        case 1: ret = true; break;
                        case -1:
                                puts("unknown response; please try again");
                                continue;
                        }
                }
                break;
        } while (1);

        free(line);

        return ret;
}

static inline int is_power_of_2(unsigned long long n)
{
        return (n != 0 && ((n & (n - 1)) == 0));
}

/**
 * simple_strtoX - convert a hex/dec/oct string into a number
 * @snum: buffer to convert
 * @error: set to 1 when buffer isn't fully consumed
 *
 * These functions are similar to the standard strtoX() functions, but they are
 * a little bit easier to use if you want to convert full string of digits into
 * the binary form. The typical usage:
 *
 * int error = 0;
 * unsigned long num;
 *
 * num = simple_strtoul(str, &error);
 * if (error || ... if needed, your check that num is not out of range ...)
 *      error_happened();
 */
#define simple_strtoX(func, type) \
static inline type simple_##func(const char *snum, int *error) \
{ \
        char *endptr; \
        type ret = func(snum, &endptr, 0); \
 \
        if (error && (!*snum || *endptr)) { \
                errmsg("%s: unable to parse the number '%s'", #func, snum); \
                *error = 1; \
        } \
 \
        return ret; \
}
simple_strtoX(strtol, long int)
simple_strtoX(strtoll, long long int)
simple_strtoX(strtoul, unsigned long int)
simple_strtoX(strtoull, unsigned long long int)

/* Simple version-printing for utils */
#define common_print_version() \
do { \
        printf("%s %s\n", PROGRAM_NAME, VERSION); \
} while (0)

#include "xalloc.h"

#ifdef __cplusplus
}
#endif

#endif /* !__MTD_UTILS_COMMON_H__ */
```

Notice for package(s)

lttng-modules

```
LTTng modules licensing
Mathieu Desnoyers
June 2, 2011
```

```
* LGPLv2.1/GPLv2 dual-license

The files contained within this package are licensed under
LGPLv2.1/GPLv2 dual-license (see lgpl-2.1.txt and gpl-2.0.txt for
details), except for files identified by the following sections.

* GPLv2 license

These files are licensed exclusively under the GPLv2 license. See
gpl-2.0.txt for details.

lib/ringbuffer/ring_buffer_splice.c
lib/ringbuffer/ring_buffer_mmap.c
instrumentation/events/mainline/*.h
instrumentation/events/lttng-modules/*.h
wrapper/list.h

* MIT-style license

These files are licensed under an MIT-style license. See mit-license.txt
for details.

lib/prio_heap/lttng_prio_heap.h
lib/prio_heap/lttng_prio_heap.c
lib/bitfield.h
```

Notice for package(s)

lttng-modules

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

  Finally, software patents pose a constant threat to the existence of
any free program.  We wish to make sure that a company cannot
```

effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

  When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

  We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

  For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

  Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

                  GNU LESSER GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,

and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The

threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses

terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING

RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!

Notice for package(s)

libxml2

Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

 Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is fur-
nished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-
NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Notice for package(s)

libxml2

```
/*
 * hash.c: chained hash tables
 *
 * Reference: Your favorite introductory book on algorithms
 *
 * Copyright (C) 2000,2012 Bjorn Reese and Daniel Veillard.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
 * MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND
 * CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.
 *
 * Author: breese@users.sourceforge.net
 */

#define IN_LIBXML
#include "libxml.h"

#include <string.h>
#ifdef HAVE_STDLIB_H
#include <stdlib.h>
#endif
#ifdef HAVE_TIME_H
#include <time.h>
#endif

/*
 * Following http://www.ocert.org/advisories/ocert-2011-003.html
 * it seems that having hash randomization might be a good idea
 * when using XML with untrusted data
 */
#if defined(HAVE_RAND) && defined(HAVE_SRAND) && defined(HAVE_TIME)
#define HASH_RANDOMIZATION
#endif

#include <libxml/parser.h>
#include <libxml/hash.h>
#include <libxml/xmlmemory.h>
#include <libxml/xmlerror.h>
#include <libxml/globals.h>

#define MAX_HASH_LEN 8

/* #define DEBUG_GROW */

/*
 * A single entry in the hash table
 */
typedef struct _xmlHashEntry xmlHashEntry;
typedef xmlHashEntry *xmlHashEntryPtr;
struct _xmlHashEntry {
    struct _xmlHashEntry *next;
    xmlChar *name;
    xmlChar *name2;
    xmlChar *name3;
    void *payload;
    int valid;
};

/*
 * The entire hash table
 */
struct _xmlHashTable {
    struct _xmlHashEntry *table;
    int size;
    int nbElems;
    xmlDictPtr dict;
#ifdef HASH_RANDOMIZATION
    int random_seed;
#endif
};

/*
 * xmlHashComputeKey:
 * Calculate the hash key
 */
static unsigned long
xmlHashComputeKey(xmlHashTablePtr table, const xmlChar *name,
                  const xmlChar *name2, const xmlChar *name3) {
    unsigned long value = 0L;
    char ch;

#ifdef HASH_RANDOMIZATION
    value = table->random_seed;
#endif
    if (name != NULL) {
        value += 30 * (*name);
        while ((ch = *name++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    value = value ^ ((value << 5) + (value >> 3));
    if (name2 != NULL) {
        while ((ch = *name2++) != 0) {
```

```
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    value = value ^ ((value << 5) + (value >> 3));
    if (name3 != NULL) {
        while ((ch = *name3++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    return (value % table->size);
}

static unsigned long
xmlHashComputeQKey(xmlHashTablePtr table,
                   const xmlChar *prefix, const xmlChar *name,
                   const xmlChar *prefix2, const xmlChar *name2,
                   const xmlChar *prefix3, const xmlChar *name3) {
    unsigned long value = 0L;
    char ch;

#ifdef HASH_RANDOMIZATION
    value = table->random_seed;
#endif
    if (prefix != NULL)
        value += 30 * (*prefix);
    else
        value += 30 * (*name);

    if (prefix != NULL) {
        while ((ch = *prefix++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
        value = value ^ ((value << 5) + (value >> 3) + (unsigned long)':');
    }
    if (name != NULL) {
        while ((ch = *name++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    value = value ^ ((value << 5) + (value >> 3));
    if (prefix2 != NULL) {
        while ((ch = *prefix2++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
        value = value ^ ((value << 5) + (value >> 3) + (unsigned long)':');
    }
    if (name2 != NULL) {
        while ((ch = *name2++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    value = value ^ ((value << 5) + (value >> 3));
    if (prefix3 != NULL) {
        while ((ch = *prefix3++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
        value = value ^ ((value << 5) + (value >> 3) + (unsigned long)':');
    }
    if (name3 != NULL) {
        while ((ch = *name3++) != 0) {
            value = value ^ ((value << 5) + (value >> 3) + (unsigned long)ch);
        }
    }
    return (value % table->size);
}

/**
 * xmlHashCreate:
 * @size: the size of the hash table
 *
 * Create a new xmlHashTablePtr.
 *
 * Returns the newly created object, or NULL if an error occured.
 */
xmlHashTablePtr
xmlHashCreate(int size) {
    xmlHashTablePtr table;

    if (size <= 0)
        size = 256;

    table = xmlMalloc(sizeof(xmlHashTable));
    if (table) {
        table->dict = NULL;
        table->size = size;
        table->nbElems = 0;
        table->table = xmlMalloc(size * sizeof(xmlHashEntry));
        if (table->table) {
            memset(table->table, 0, size * sizeof(xmlHashEntry));
#ifdef HASH_RANDOMIZATION
            table->random_seed = __xmlRandom();
#endif
            return(table);
        }
        xmlFree(table);
    }
    return(NULL);
}
```

```
/**
 * xmlHashCreateDict:
 * @size: the size of the hash table
 * @dict: a dictionary to use for the hash
 *
 * Create a new xmlHashTablePtr which will use @dict as the internal dictionary
 *
 * Returns the newly created object, or NULL if an error occured.
 */
xmlHashTablePtr
xmlHashCreateDict(int size, xmlDictPtr dict) {
    xmlHashTablePtr table;

    table = xmlHashCreate(size);
    if (table != NULL) {
        table->dict = dict;
        xmlDictReference(dict);
    }
    return(table);
}

/**
 * xmlHashGrow:
 * @table: the hash table
 * @size: the new size of the hash table
 *
 * resize the hash table
 *
 * Returns 0 in case of success, -1 in case of failure
 */
static int
xmlHashGrow(xmlHashTablePtr table, int size) {
    unsigned long key;
    int oldsize, i;
    xmlHashEntryPtr iter, next;
    struct _xmlHashEntry *oldtable;
#ifdef DEBUG_GROW
    unsigned long nbElem = 0;
#endif

    if (table == NULL)
        return(-1);
    if (size < 8)
        return(-1);
    if (size > 8 * 2048)
        return(-1);

    oldsize = table->size;
    oldtable = table->table;
    if (oldtable == NULL)
        return(-1);

    table->table = xmlMalloc(size * sizeof(xmlHashEntry));
    if (table->table == NULL) {
        table->table = oldtable;
        return(-1);
    }
    memset(table->table, 0, size * sizeof(xmlHashEntry));
    table->size = size;

    /*  If the two loops are merged, there would be situations where
        a new entry needs to allocated and data copied into it from
        the main table. So instead, we run through the array twice, first
        copying all the elements in the main array (where we can't get
        conflicts) and then the rest, so we only free (and don't allocate)
    */
    for (i = 0; i < oldsize; i++) {
        if (oldtable[i].valid == 0)
            continue;
        key = xmlHashComputeKey(table, oldtable[i].name, oldtable[i].name2,
                                oldtable[i].name3);
        memcpy(&(table->table[key]), &(oldtable[i]), sizeof(xmlHashEntry));
        table->table[key].next = NULL;
    }

    for (i = 0; i < oldsize; i++) {
        iter = oldtable[i].next;
        while (iter) {
            next = iter->next;

            /*
             * put back the entry in the new table
             */

            key = xmlHashComputeKey(table, iter->name, iter->name2,
                                    iter->name3);
            if (table->table[key].valid == 0) {
                memcpy(&(table->table[key]), iter, sizeof(xmlHashEntry));
                table->table[key].next = NULL;
                xmlFree(iter);
            } else {
                iter->next = table->table[key].next;
                table->table[key].next = iter;
            }

#ifdef DEBUG_GROW
            nbElem++;
```

```
#endif

            iter = next;
        }
    }

    xmlFree(oldtable);

#ifdef DEBUG_GROW
    xmlGenericError(xmlGenericErrorContext,
            "xmlHashGrow : from %d to %d, %d elems\n", oldsize, size, nbElem);
#endif

    return(0);
}

/**
 * xmlHashFree:
 * @table: the hash table
 * @f:  the deallocator function for items in the hash
 *
 * Free the hash @table and its contents. The userdata is
 * deallocated with @f if provided.
 */
void
xmlHashFree(xmlHashTablePtr table, xmlHashDeallocator f) {
    int i;
    xmlHashEntryPtr iter;
    xmlHashEntryPtr next;
    int inside_table = 0;
    int nbElems;

    if (table == NULL)
        return;
    if (table->table) {
        nbElems = table->nbElems;
        for(i = 0; (i < table->size) && (nbElems > 0); i++) {
            iter = &(table->table[i]);
            if (iter->valid == 0)
                continue;
            inside_table = 1;
            while (iter) {
                next = iter->next;
                if ((f != NULL) && (iter->payload != NULL))
                    f(iter->payload, iter->name);
                if (table->dict == NULL) {
                    if (iter->name)
                        xmlFree(iter->name);
                    if (iter->name2)
                        xmlFree(iter->name2);
                    if (iter->name3)
                        xmlFree(iter->name3);
                }
                iter->payload = NULL;
                if (!inside_table)
                    xmlFree(iter);
                nbElems--;
                inside_table = 0;
                iter = next;
            }
        }
        xmlFree(table->table);
    }
    if (table->dict)
        xmlDictFree(table->dict);
    xmlFree(table);
}

/**
 * xmlHashAddEntry:
 * @table: the hash table
 * @name: the name of the userdata
 * @userdata: a pointer to the userdata
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the @name. Duplicate names generate errors.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashAddEntry(xmlHashTablePtr table, const xmlChar *name, void *userdata) {
    return(xmlHashAddEntry3(table, name, NULL, NULL, userdata));
}

/**
 * xmlHashAddEntry2:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @userdata: a pointer to the userdata
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the (@name, @name2) tuple. Duplicate tuples generate errors.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashAddEntry2(xmlHashTablePtr table, const xmlChar *name,
```

```
                const xmlChar *name2, void *userdata) {
    return(xmlHashAddEntry3(table, name, name2, NULL, userdata));
}

/**
 * xmlHashUpdateEntry:
 * @table: the hash table
 * @name: the name of the userdata
 * @userdata: a pointer to the userdata
 * @f: the deallocator function for replaced item (if any)
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the @name. Existing entry for this @name will be removed
 * and freed with @f if found.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashUpdateEntry(xmlHashTablePtr table, const xmlChar *name,
                   void *userdata, xmlHashDeallocator f) {
    return(xmlHashUpdateEntry3(table, name, NULL, NULL, userdata, f));
}

/**
 * xmlHashUpdateEntry2:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @userdata: a pointer to the userdata
 * @f: the deallocator function for replaced item (if any)
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the (@name, @name2) tuple. Existing entry for this tuple will
 * be removed and freed with @f if found.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashUpdateEntry2(xmlHashTablePtr table, const xmlChar *name,
                    const xmlChar *name2, void *userdata,
                    xmlHashDeallocator f) {
    return(xmlHashUpdateEntry3(table, name, name2, NULL, userdata, f));
}

/**
 * xmlHashLookup:
 * @table: the hash table
 * @name: the name of the userdata
 *
 * Find the userdata specified by the @name.
 *
 * Returns the pointer to the userdata
 */
void *
xmlHashLookup(xmlHashTablePtr table, const xmlChar *name) {
    return(xmlHashLookup3(table, name, NULL, NULL));
}

/**
 * xmlHashLookup2:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 *
 * Find the userdata specified by the (@name, @name2) tuple.
 *
 * Returns the pointer to the userdata
 */
void *
xmlHashLookup2(xmlHashTablePtr table, const xmlChar *name,
               const xmlChar *name2) {
    return(xmlHashLookup3(table, name, name2, NULL));
}

/**
 * xmlHashQLookup:
 * @table: the hash table
 * @prefix: the prefix of the userdata
 * @name: the name of the userdata
 *
 * Find the userdata specified by the QName @prefix:@name/@name.
 *
 * Returns the pointer to the userdata
 */
void *
xmlHashQLookup(xmlHashTablePtr table, const xmlChar *prefix,
               const xmlChar *name) {
    return(xmlHashQLookup3(table, prefix, name, NULL, NULL, NULL, NULL));
}

/**
 * xmlHashQLookup2:
 * @table: the hash table
 * @prefix: the prefix of the userdata
 * @name: the name of the userdata
 * @prefix2: the second prefix of the userdata
 * @name2: a second name of the userdata
 *
```

```
 * Find the userdata specified by the QNames tuple
 *
 * Returns the pointer to the userdata
 */
void *
xmlHashQLookup2(xmlHashTablePtr table, const xmlChar *prefix,
                const xmlChar *name, const xmlChar *prefix2,
                const xmlChar *name2) {
    return(xmlHashQLookup3(table, prefix, name, prefix2, name2, NULL, NULL));
}

/**
 * xmlHashAddEntry3:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @name3: a third name of the userdata
 * @userdata: a pointer to the userdata
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the tuple (@name, @name2, @name3). Duplicate entries generate
 * errors.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashAddEntry3(xmlHashTablePtr table, const xmlChar *name,
                 const xmlChar *name2, const xmlChar *name3,
                 void *userdata) {
    unsigned long key, len = 0;
    xmlHashEntryPtr entry;
    xmlHashEntryPtr insert;

    if ((table == NULL) || (name == NULL))
        return(-1);

    /*
     * If using a dict internalize if needed
     */
    if (table->dict) {
        if (!xmlDictOwns(table->dict, name)) {
            name = xmlDictLookup(table->dict, name, -1);
            if (name == NULL)
                return(-1);
        }
        if ((name2 != NULL) && (!xmlDictOwns(table->dict, name2))) {
            name2 = xmlDictLookup(table->dict, name2, -1);
            if (name2 == NULL)
                return(-1);
        }
        if ((name3 != NULL) && (!xmlDictOwns(table->dict, name3))) {
            name3 = xmlDictLookup(table->dict, name3, -1);
            if (name3 == NULL)
                return(-1);
        }
    }

    /*
     * Check for duplicate and insertion location.
     */
    key = xmlHashComputeKey(table, name, name2, name3);
    if (table->table[key].valid == 0) {
        insert = NULL;
    } else {
        if (table->dict) {
            for (insert = &(table->table[key]); insert->next != NULL;
                 insert = insert->next) {
                if ((insert->name == name) &&
                    (insert->name2 == name2) &&
                    (insert->name3 == name3))
                    return(-1);
                len++;
            }
            if ((insert->name == name) &&
                (insert->name2 == name2) &&
                (insert->name3 == name3))
                return(-1);
        } else {
            for (insert = &(table->table[key]); insert->next != NULL;
                 insert = insert->next) {
                if ((xmlStrEqual(insert->name, name)) &&
                    (xmlStrEqual(insert->name2, name2)) &&
                    (xmlStrEqual(insert->name3, name3)))
                    return(-1);
                len++;
            }
            if ((xmlStrEqual(insert->name, name)) &&
                (xmlStrEqual(insert->name2, name2)) &&
                (xmlStrEqual(insert->name3, name3)))
                return(-1);
        }
    }

    if (insert == NULL) {
        entry = &(table->table[key]);
    } else {
        entry = xmlMalloc(sizeof(xmlHashEntry));
        if (entry == NULL)
```

```
            return(-1);
    }

    if (table->dict != NULL) {
        entry->name = (xmlChar *) name;
        entry->name2 = (xmlChar *) name2;
        entry->name3 = (xmlChar *) name3;
    } else {
        entry->name = xmlStrdup(name);
        entry->name2 = xmlStrdup(name2);
        entry->name3 = xmlStrdup(name3);
    }
    entry->payload = userdata;
    entry->next = NULL;
    entry->valid = 1;


    if (insert != NULL)
        insert->next = entry;

    table->nbElems++;

    if (len > MAX_HASH_LEN)
        xmlHashGrow(table, MAX_HASH_LEN * table->size);

    return(0);
}

/**
 * xmlHashUpdateEntry3:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @name3: a third name of the userdata
 * @userdata: a pointer to the userdata
 * @f: the deallocator function for replaced item (if any)
 *
 * Add the @userdata to the hash @table. This can later be retrieved
 * by using the tuple (@name, @name2, @name3). Existing entry for this tuple
 * will be removed and freed with @f if found.
 *
 * Returns 0 the addition succeeded and -1 in case of error.
 */
int
xmlHashUpdateEntry3(xmlHashTablePtr table, const xmlChar *name,
                    const xmlChar *name2, const xmlChar *name3,
                    void *userdata, xmlHashDeallocator f) {
    unsigned long key;
    xmlHashEntryPtr entry;
    xmlHashEntryPtr insert;

    if ((table == NULL) || name == NULL)
        return(-1);

    /*
     * If using a dict internalize if needed
     */
    if (table->dict) {
        if (!xmlDictOwns(table->dict, name)) {
            name = xmlDictLookup(table->dict, name, -1);
            if (name == NULL)
                return(-1);
        }
        if ((name2 != NULL) && (!xmlDictOwns(table->dict, name2))) {
            name2 = xmlDictLookup(table->dict, name2, -1);
            if (name2 == NULL)
                return(-1);
        }
        if ((name3 != NULL) && (!xmlDictOwns(table->dict, name3))) {
            name3 = xmlDictLookup(table->dict, name3, -1);
            if (name3 == NULL)
                return(-1);
        }
    }

    /*
     * Check for duplicate and insertion location.
     */
    key = xmlHashComputeKey(table, name, name2, name3);
    if (table->table[key].valid == 0) {
        insert = NULL;
    } else {
        if (table ->dict) {
            for (insert = &(table->table[key]); insert->next != NULL;
                 insert = insert->next) {
                if ((insert->name == name) &&
                    (insert->name2 == name2) &&
                    (insert->name3 == name3)) {
                    if (f)
                        f(insert->payload, insert->name);
                    insert->payload = userdata;
                    return(0);
                }
            }
            if ((insert->name == name) &&
                (insert->name2 == name2) &&
                (insert->name3 == name3)) {
                if (f)
```

```
                    f(insert->payload, insert->name);
                insert->payload = userdata;
                return(0);
            }
        } else {
            for (insert = &(table->table[key]); insert->next != NULL;
                 insert = insert->next) {
                if ((xmlStrEqual(insert->name, name)) &&
                    (xmlStrEqual(insert->name2, name2)) &&
                    (xmlStrEqual(insert->name3, name3))) {
                    if (f)
                        f(insert->payload, insert->name);
                    insert->payload = userdata;
                    return(0);
                }
            }
            if ((xmlStrEqual(insert->name, name)) &&
                (xmlStrEqual(insert->name2, name2)) &&
                (xmlStrEqual(insert->name3, name3))) {
                if (f)
                    f(insert->payload, insert->name);
                insert->payload = userdata;
                return(0);
            }
        }
    }

    if (insert == NULL) {
        entry =  &(table->table[key]);
    } else {
        entry = xmlMalloc(sizeof(xmlHashEntry));
        if (entry == NULL)
             return(-1);
    }

    if (table->dict != NULL) {
        entry->name = (xmlChar *) name;
        entry->name2 = (xmlChar *) name2;
        entry->name3 = (xmlChar *) name3;
    } else {
        entry->name = xmlStrdup(name);
        entry->name2 = xmlStrdup(name2);
        entry->name3 = xmlStrdup(name3);
    }
    entry->payload = userdata;
    entry->next = NULL;
    entry->valid = 1;
    table->nbElems++;


    if (insert != NULL) {
        insert->next = entry;
    }
    return(0);
}

/**
 * xmlHashLookup3:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @name3: a third name of the userdata
 *
 * Find the userdata specified by the (@name, @name2, @name3) tuple.
 *
 * Returns the a pointer to the userdata
 */
void *
xmlHashLookup3(xmlHashTablePtr table, const xmlChar *name,
               const xmlChar *name2, const xmlChar *name3) {
    unsigned long key;
    xmlHashEntryPtr entry;

    if (table == NULL)
        return(NULL);
    if (name == NULL)
        return(NULL);
    key = xmlHashComputeKey(table, name, name2, name3);
    if (table->table[key].valid == 0)
        return(NULL);
    if (table->dict) {
        for (entry = &(table->table[key]); entry != NULL; entry = entry->next) {
            if ((entry->name == name) &&
                (entry->name2 == name2) &&
                (entry->name3 == name3))
                return(entry->payload);
        }
    }
    for (entry = &(table->table[key]); entry != NULL; entry = entry->next) {
        if ((xmlStrEqual(entry->name, name)) &&
            (xmlStrEqual(entry->name2, name2)) &&
            (xmlStrEqual(entry->name3, name3)))
            return(entry->payload);
    }
    return(NULL);
}

/**
```

```
 * xmlHashQLookup3:
 * @table: the hash table
 * @prefix: the prefix of the userdata
 * @name: the name of the userdata
 * @prefix2: the second prefix of the userdata
 * @name2: a second name of the userdata
 * @prefix3: the third prefix of the userdata
 * @name3: a third name of the userdata
 *
 * Find the userdata specified by the (@name, @name2, @name3) tuple.
 *
 * Returns the a pointer to the userdata
 */
void *
xmlHashQLookup3(xmlHashTablePtr table,
                const xmlChar *prefix, const xmlChar *name,
                const xmlChar *prefix2, const xmlChar *name2,
                const xmlChar *prefix3, const xmlChar *name3) {
    unsigned long key;
    xmlHashEntryPtr entry;

    if (table == NULL)
        return(NULL);
    if (name == NULL)
        return(NULL);
    key = xmlHashComputeQKey(table, prefix, name, prefix2,
                             name2, prefix3, name3);
    if (table->table[key].valid == 0)
        return(NULL);
    for (entry = &(table->table[key]); entry != NULL; entry = entry->next) {
        if ((xmlStrQEqual(prefix, name, entry->name)) &&
            (xmlStrQEqual(prefix2, name2, entry->name2)) &&
            (xmlStrQEqual(prefix3, name3, entry->name3)))
            return(entry->payload);
    }
    return(NULL);
}

typedef struct {
    xmlHashScanner hashscanner;
    void *data;
} stubData;

static void
stubHashScannerFull (void *payload, void *data, const xmlChar *name,
                     const xmlChar *name2 ATTRIBUTE_UNUSED,
                     const xmlChar *name3 ATTRIBUTE_UNUSED) {
    stubData *stubdata = (stubData *) data;
    stubdata->hashscanner (payload, stubdata->data, (xmlChar *) name);
}

/**
 * xmlHashScan:
 * @table: the hash table
 * @f:  the scanner function for items in the hash
 * @data:  extra data passed to f
 *
 * Scan the hash @table and applied @f to each value.
 */
void
xmlHashScan(xmlHashTablePtr table, xmlHashScanner f, void *data) {
    stubData stubdata;
    stubdata.data = data;
    stubdata.hashscanner = f;
    xmlHashScanFull (table, stubHashScannerFull, &stubdata);
}

/**
 * xmlHashScanFull:
 * @table: the hash table
 * @f:  the scanner function for items in the hash
 * @data:  extra data passed to f
 *
 * Scan the hash @table and applied @f to each value.
 */
void
xmlHashScanFull(xmlHashTablePtr table, xmlHashScannerFull f, void *data) {
    int i, nb;
    xmlHashEntryPtr iter;
    xmlHashEntryPtr next;

    if (table == NULL)
        return;
    if (f == NULL)
        return;

    if (table->table) {
        for(i = 0; i < table->size; i++) {
            if (table->table[i].valid == 0)
                continue;
            iter = &(table->table[i]);
            while (iter) {
                next = iter->next;
                nb = table->nbElems;
                if ((f != NULL) && (iter->payload != NULL))
                    f(iter->payload, data, iter->name,
                      iter->name2, iter->name3);
                if (nb != table->nbElems) {
```

```
                    /* table was modified by the callback, be careful */
                    if (iter == &(table->table[i])) {
                        if (table->table[i].valid == 0)
                            iter = NULL;
                        if (table->table[i].next != next)
                            iter = &(table->table[i]);
                    } else
                        iter = next;
                } else
                    iter = next;
            }
        }
    }
}

/**
 * xmlHashScan3:
 * @table: the hash table
 * @name: the name of the userdata or NULL
 * @name2: a second name of the userdata or NULL
 * @name3: a third name of the userdata or NULL
 * @f:  the scanner function for items in the hash
 * @data:  extra data passed to f
 *
 * Scan the hash @table and applied @f to each value matching
 * (@name, @name2, @name3) tuple. If one of the names is null,
 * the comparison is considered to match.
 */
void
xmlHashScan3(xmlHashTablePtr table, const xmlChar *name,
             const xmlChar *name2, const xmlChar *name3,
             xmlHashScanner f, void *data) {
    xmlHashScanFull3 (table, name, name2, name3,
                      (xmlHashScannerFull) f, data);
}

/**
 * xmlHashScanFull3:
 * @table: the hash table
 * @name: the name of the userdata or NULL
 * @name2: a second name of the userdata or NULL
 * @name3: a third name of the userdata or NULL
 * @f:  the scanner function for items in the hash
 * @data:  extra data passed to f
 *
 * Scan the hash @table and applied @f to each value matching
 * (@name, @name2, @name3) tuple. If one of the names is null,
 * the comparison is considered to match.
 */
void
xmlHashScanFull3(xmlHashTablePtr table, const xmlChar *name,
                 const xmlChar *name2, const xmlChar *name3,
                 xmlHashScannerFull f, void *data) {
    int i;
    xmlHashEntryPtr iter;
    xmlHashEntryPtr next;

    if (table == NULL)
        return;
    if (f == NULL)
        return;

    if (table->table) {
        for(i = 0; i < table->size; i++) {
            if (table->table[i].valid == 0)
                continue;
            iter = &(table->table[i]);
            while (iter) {
                next = iter->next;
                if (((name == NULL) || (xmlStrEqual(name, iter->name))) &&
                    ((name2 == NULL) || (xmlStrEqual(name2, iter->name2))) &&
                    ((name3 == NULL) || (xmlStrEqual(name3, iter->name3))) &&
                    (iter->payload != NULL)) {
                    f(iter->payload, data, iter->name,
                      iter->name2, iter->name3);
                }
                iter = next;
            }
        }
    }
}

/**
 * xmlHashCopy:
 * @table: the hash table
 * @f:  the copier function for items in the hash
 *
 * Scan the hash @table and applied @f to each value.
 *
 * Returns the new table or NULL in case of error.
 */
xmlHashTablePtr
xmlHashCopy(xmlHashTablePtr table, xmlHashCopier f) {
    int i;
    xmlHashEntryPtr iter;
    xmlHashEntryPtr next;
    xmlHashTablePtr ret;
```

```
    if (table == NULL)
        return(NULL);
    if (f == NULL)
        return(NULL);

    ret = xmlHashCreate(table->size);
    if (ret == NULL)
        return(NULL);

    if (table->table) {
        for(i = 0; i < table->size; i++) {
            if (table->table[i].valid == 0)
                continue;
            iter = &(table->table[i]);
            while (iter) {
                next = iter->next;
                xmlHashAddEntry3(ret, iter->name, iter->name2,
                                 iter->name3, f(iter->payload, iter->name));
                iter = next;
            }
        }
    }
    ret->nbElems = table->nbElems;
    return(ret);
}

/**
 * xmlHashSize:
 * @table: the hash table
 *
 * Query the number of elements installed in the hash @table.
 *
 * Returns the number of elements in the hash table or
 * -1 in case of error
 */
int
xmlHashSize(xmlHashTablePtr table) {
    if (table == NULL)
        return(-1);
    return(table->nbElems);
}

/**
 * xmlHashRemoveEntry:
 * @table: the hash table
 * @name: the name of the userdata
 * @f: the deallocator function for removed item (if any)
 *
 * Find the userdata specified by the @name and remove
 * it from the hash @table. Existing userdata for this tuple will be removed
 * and freed with @f.
 *
 * Returns 0 if the removal succeeded and -1 in case of error or not found.
 */
int xmlHashRemoveEntry(xmlHashTablePtr table, const xmlChar *name,
                       xmlHashDeallocator f) {
    return(xmlHashRemoveEntry3(table, name, NULL, NULL, f));
}

/**
 * xmlHashRemoveEntry2:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @f: the deallocator function for removed item (if any)
 *
 * Find the userdata specified by the (@name, @name2) tuple and remove
 * it from the hash @table. Existing userdata for this tuple will be removed
 * and freed with @f.
 *
 * Returns 0 if the removal succeeded and -1 in case of error or not found.
 */
int
xmlHashRemoveEntry2(xmlHashTablePtr table, const xmlChar *name,
                    const xmlChar *name2, xmlHashDeallocator f) {
    return(xmlHashRemoveEntry3(table, name, name2, NULL, f));
}

/**
 * xmlHashRemoveEntry3:
 * @table: the hash table
 * @name: the name of the userdata
 * @name2: a second name of the userdata
 * @name3: a third name of the userdata
 * @f: the deallocator function for removed item (if any)
 *
 * Find the userdata specified by the (@name, @name2, @name3) tuple and remove
 * it from the hash @table. Existing userdata for this tuple will be removed
 * and freed with @f.
 *
 * Returns 0 if the removal succeeded and -1 in case of error or not found.
 */
int
xmlHashRemoveEntry3(xmlHashTablePtr table, const xmlChar *name,
    const xmlChar *name2, const xmlChar *name3, xmlHashDeallocator f) {
    unsigned long key;
    xmlHashEntryPtr entry;
    xmlHashEntryPtr prev = NULL;
```

```
    if (table == NULL || name == NULL)
        return(-1);

    key = xmlHashComputeKey(table, name, name2, name3);
    if (table->table[key].valid == 0) {
        return(-1);
    } else {
        for (entry = &(table->table[key]); entry != NULL; entry = entry->next) {
            if (xmlStrEqual(entry->name, name) &&
                    xmlStrEqual(entry->name2, name2) &&
                    xmlStrEqual(entry->name3, name3)) {
                if ((f != NULL) && (entry->payload != NULL))
                    f(entry->payload, entry->name);
                entry->payload = NULL;
                if (table->dict == NULL) {
                    if(entry->name)
                        xmlFree(entry->name);
                    if(entry->name2)
                        xmlFree(entry->name2);
                    if(entry->name3)
                        xmlFree(entry->name3);
                }
                if(prev) {
                    prev->next = entry->next;
                    xmlFree(entry);
                } else {
                    if (entry->next == NULL) {
                        entry->valid = 0;
                    } else {
                        entry = entry->next;
                        memcpy(&(table->table[key]), entry, sizeof(xmlHashEntry));
                        xmlFree(entry);
                    }
                }
                table->nbElems--;
                return(0);
            }
            prev = entry;
        }
        return(-1);
    }
}

#define bottom_hash
#include "elfgcchack.h"
```

Notice for package(s)

libxml2

```
/*
 * list.c: lists handling implementation
 *
 * Copyright (C) 2000 Gary Pennington and Daniel Veillard.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
 * MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND
 * CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.
 *
 * Author: Gary.Pennington@uk.sun.com
 */

#define IN_LIBXML
#include "libxml.h"

#include <stdlib.h>
#include <string.h>
#include <libxml/xmlmemory.h>
#include <libxml/list.h>
#include <libxml/globals.h>

/*
 * Type definition are kept internal
 */

struct _xmlLink
{
    struct _xmlLink *next;
    struct _xmlLink *prev;
    void *data;
};

struct _xmlList
{
    xmlLinkPtr sentinel;
```

```
    void (*linkDeallocator)(xmlLinkPtr );
    int (*linkCompare)(const void *, const void*);
};

/************************************************************************
 *                                  *
 *                Interfaces                *
 *                                  *
 ************************************************************************/

/**
 * xmlLinkDeallocator:
 * @l:  a list
 * @lk:  a link
 *
 * Unlink and deallocate @lk from list @l
 */
static void
xmlLinkDeallocator(xmlListPtr l, xmlLinkPtr lk)
{
    (lk->prev)->next = lk->next;
    (lk->next)->prev = lk->prev;
    if(l->linkDeallocator)
        l->linkDeallocator(lk);
    xmlFree(lk);
}

/**
 * xmlLinkCompare:
 * @data0:  first data
 * @data1:  second data
 *
 * Compares two arbitrary data
 *
 * Returns -1, 0 or 1 depending on whether data1 is greater equal or smaller
 *          than data0
 */
static int
xmlLinkCompare(const void *data0, const void *data1)
{
    if (data0 < data1)
        return (-1);
    else if (data0 == data1)
        return (0);
    return (1);
}

/**
 * xmlListLowerSearch:
 * @l:  a list
 * @data:  a data
 *
 * Search data in the ordered list walking from the beginning
 *
 * Returns the link containing the data or NULL
 */
static xmlLinkPtr
xmlListLowerSearch(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;

    if (l == NULL)
        return(NULL);
    for(lk = l->sentinel->next;lk != l->sentinel && l->linkCompare(lk->data, data) <0 ;lk = lk->next);
    return lk;
}

/**
 * xmlListHigherSearch:
 * @l:  a list
 * @data:  a data
 *
 * Search data in the ordered list walking backward from the end
 *
 * Returns the link containing the data or NULL
 */
static xmlLinkPtr
xmlListHigherSearch(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;

    if (l == NULL)
        return(NULL);
    for(lk = l->sentinel->prev;lk != l->sentinel && l->linkCompare(lk->data, data) >0 ;lk = lk->prev);
    return lk;
}

/**
 * xmlListSearch:
 * @l:  a list
 * @data:  a data
 *
 * Search data in the list
 *
 * Returns the link containing the data or NULL
 */
static xmlLinkPtr
xmlListLinkSearch(xmlListPtr l, void *data)
```

```
{
    xmlLinkPtr lk;
    if (l == NULL)
        return(NULL);
    lk = xmlListLowerSearch(l, data);
    if (lk == l->sentinel)
        return NULL;
    else {
        if (l->linkCompare(lk->data, data) ==0)
            return lk;
        return NULL;
    }
}

/**
 * xmlListLinkReverseSearch:
 * @l:  a list
 * @data:  a data
 *
 * Search data in the list processing backward
 *
 * Returns the link containing the data or NULL
 */
static xmlLinkPtr
xmlListLinkReverseSearch(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;
    if (l == NULL)
        return(NULL);
    lk = xmlListHigherSearch(l, data);
    if (lk == l->sentinel)
        return NULL;
    else {
        if (l->linkCompare(lk->data, data) ==0)
            return lk;
        return NULL;
    }
}

/**
 * xmlListCreate:
 * @deallocator:  an optional deallocator function
 * @compare:  an optional comparison function
 *
 * Create a new list
 *
 * Returns the new list or NULL in case of error
 */
xmlListPtr
xmlListCreate(xmlListDeallocator deallocator, xmlListDataCompare compare)
{
    xmlListPtr l;
    if (NULL == (l = (xmlListPtr )xmlMalloc( sizeof(xmlList)))) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for list");
        return (NULL);
    }
    /* Initialize the list to NULL */
    memset(l, 0, sizeof(xmlList));

    /* Add the sentinel */
    if (NULL ==(l->sentinel = (xmlLinkPtr )xmlMalloc(sizeof(xmlLink)))) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for sentinel");
        xmlFree(l);
        return (NULL);
    }
    l->sentinel->next = l->sentinel;
    l->sentinel->prev = l->sentinel;
    l->sentinel->data = NULL;

    /* If there is a link deallocator, use it */
    if (deallocator != NULL)
        l->linkDeallocator = deallocator;
    /* If there is a link comparator, use it */
    if (compare != NULL)
        l->linkCompare = compare;
    else /* Use our own */
        l->linkCompare = xmlLinkCompare;
    return l;
}

/**
 * xmlListSearch:
 * @l:  a list
 * @data:  a search value
 *
 * Search the list for an existing value of @data
 *
 * Returns the value associated to @data or NULL in case of error
 */
void *
xmlListSearch(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;
    if (l == NULL)
        return(NULL);
    lk = xmlListLinkSearch(l, data);
```

```
    if (lk)
        return (lk->data);
    return NULL;
}

/**
 * xmlListReverseSearch:
 * @l:  a list
 * @data:  a search value
 *
 * Search the list in reverse order for an existing value of @data
 *
 * Returns the value associated to @data or NULL in case of error
 */
void *
xmlListReverseSearch(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;
    if (l == NULL)
        return(NULL);
    lk = xmlListLinkReverseSearch(l, data);
    if (lk)
        return (lk->data);
    return NULL;
}

/**
 * xmlListInsert:
 * @l:  a list
 * @data:  the data
 *
 * Insert data in the ordered list at the beginning for this value
 *
 * Returns 0 in case of success, 1 in case of failure
 */
int
xmlListInsert(xmlListPtr l, void *data)
{
    xmlLinkPtr lkPlace, lkNew;

    if (l == NULL)
        return(1);
    lkPlace = xmlListLowerSearch(l, data);
    /* Add the new link */
    lkNew = (xmlLinkPtr) xmlMalloc(sizeof(xmlLink));
    if (lkNew == NULL) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for new link");
        return (1);
    }
    lkNew->data = data;
    lkPlace = lkPlace->prev;
    lkNew->next = lkPlace->next;
    (lkPlace->next)->prev = lkNew;
    lkPlace->next = lkNew;
    lkNew->prev = lkPlace;
    return 0;
}

/**
 * xmlListAppend:
 * @l:  a list
 * @data:  the data
 *
 * Insert data in the ordered list at the end for this value
 *
 * Returns 0 in case of success, 1 in case of failure
 */
int xmlListAppend(xmlListPtr l, void *data)
{
    xmlLinkPtr lkPlace, lkNew;

    if (l == NULL)
        return(1);
    lkPlace = xmlListHigherSearch(l, data);
    /* Add the new link */
    lkNew = (xmlLinkPtr) xmlMalloc(sizeof(xmlLink));
    if (lkNew == NULL) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for new link");
        return (1);
    }
    lkNew->data = data;
    lkNew->next = lkPlace->next;
    (lkPlace->next)->prev = lkNew;
    lkPlace->next = lkNew;
    lkNew->prev = lkPlace;
    return 0;
}

/**
 * xmlListDelete:
 * @l:  a list
 *
 * Deletes the list and its associated data
 */
void xmlListDelete(xmlListPtr l)
{
```

```
    if (l == NULL)
        return;

    xmlListClear(l);
    xmlFree(l->sentinel);
    xmlFree(l);
}

/**
 * xmlListRemoveFirst:
 * @l:  a list
 * @data:  list data
 *
 * Remove the first instance associated to data in the list
 *
 * Returns 1 if a deallocation occured, or 0 if not found
 */
int
xmlListRemoveFirst(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;

    if (l == NULL)
        return(0);
    /*Find the first instance of this data */
    lk = xmlListLinkSearch(l, data);
    if (lk != NULL) {
        xmlLinkDeallocator(l, lk);
        return 1;
    }
    return 0;
}

/**
 * xmlListRemoveLast:
 * @l:  a list
 * @data:  list data
 *
 * Remove the last instance associated to data in the list
 *
 * Returns 1 if a deallocation occured, or 0 if not found
 */
int
xmlListRemoveLast(xmlListPtr l, void *data)
{
    xmlLinkPtr lk;

    if (l == NULL)
        return(0);
    /*Find the last instance of this data */
    lk = xmlListLinkReverseSearch(l, data);
    if (lk != NULL) {
        xmlLinkDeallocator(l, lk);
        return 1;
    }
    return 0;
}

/**
 * xmlListRemoveAll:
 * @l:  a list
 * @data:  list data
 *
 * Remove the all instance associated to data in the list
 *
 * Returns the number of deallocation, or 0 if not found
 */
int
xmlListRemoveAll(xmlListPtr l, void *data)
{
    int count=0;

    if (l == NULL)
        return(0);

    while(xmlListRemoveFirst(l, data))
        count++;
    return count;
}

/**
 * xmlListClear:
 * @l:  a list
 *
 * Remove the all data in the list
 */
void
xmlListClear(xmlListPtr l)
{
    xmlLinkPtr  lk;

    if (l == NULL)
        return;
    lk = l->sentinel->next;
    while(lk != l->sentinel) {
        xmlLinkPtr next = lk->next;

        xmlLinkDeallocator(l, lk);
```

```
        lk = next;
    }
}

/**
 * xmlListEmpty:
 * @l:  a list
 *
 * Is the list empty ?
 *
 * Returns 1 if the list is empty, 0 if not empty and -1 in case of error
 */
int
xmlListEmpty(xmlListPtr l)
{
    if (l == NULL)
        return(-1);
    return (l->sentinel->next == l->sentinel);
}

/**
 * xmlListFront:
 * @l:  a list
 *
 * Get the first element in the list
 *
 * Returns the first element in the list, or NULL
 */
xmlLinkPtr
xmlListFront(xmlListPtr l)
{
    if (l == NULL)
        return(NULL);
    return (l->sentinel->next);
}

/**
 * xmlListEnd:
 * @l:  a list
 *
 * Get the last element in the list
 *
 * Returns the last element in the list, or NULL
 */
xmlLinkPtr
xmlListEnd(xmlListPtr l)
{
    if (l == NULL)
        return(NULL);
    return (l->sentinel->prev);
}

/**
 * xmlListSize:
 * @l:  a list
 *
 * Get the number of elements in the list
 *
 * Returns the number of elements in the list or -1 in case of error
 */
int
xmlListSize(xmlListPtr l)
{
    xmlLinkPtr lk;
    int count=0;

    if (l == NULL)
        return(-1);
    /* TODO: keep a counter in xmlList instead */
    for(lk = l->sentinel->next; lk != l->sentinel; lk = lk->next, count++);
    return count;
}

/**
 * xmlListPopFront:
 * @l:  a list
 *
 * Removes the first element in the list
 */
void
xmlListPopFront(xmlListPtr l)
{
    if(!xmlListEmpty(l))
        xmlLinkDeallocator(l, l->sentinel->next);
}

/**
 * xmlListPopBack:
 * @l:  a list
 *
 * Removes the last element in the list
 */
void
xmlListPopBack(xmlListPtr l)
{
    if(!xmlListEmpty(l))
        xmlLinkDeallocator(l, l->sentinel->prev);
}
```

```
/**
 * xmlListPushFront:
 * @l:  a list
 * @data:  new data
 *
 * add the new data at the beginning of the list
 *
 * Returns 1 if successful, 0 otherwise
 */
int
xmlListPushFront(xmlListPtr l, void *data)
{
    xmlLinkPtr lkPlace, lkNew;

    if (l == NULL)
        return(0);
    lkPlace = l->sentinel;
    /* Add the new link */
    lkNew = (xmlLinkPtr) xmlMalloc(sizeof(xmlLink));
    if (lkNew == NULL) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for new link");
        return (0);
    }
    lkNew->data = data;
    lkNew->next = lkPlace->next;
    (lkPlace->next)->prev = lkNew;
    lkPlace->next = lkNew;
    lkNew->prev = lkPlace;
    return 1;
}

/**
 * xmlListPushBack:
 * @l:  a list
 * @data:  new data
 *
 * add the new data at the end of the list
 *
 * Returns 1 if successful, 0 otherwise
 */
int
xmlListPushBack(xmlListPtr l, void *data)
{
    xmlLinkPtr lkPlace, lkNew;

    if (l == NULL)
        return(0);
    lkPlace = l->sentinel->prev;
    /* Add the new link */
    if (NULL ==(lkNew = (xmlLinkPtr )xmlMalloc(sizeof(xmlLink)))) {
        xmlGenericError(xmlGenericErrorContext,
                        "Cannot initialize memory for new link");
        return (0);
    }
    lkNew->data = data;
    lkNew->next = lkPlace->next;
    (lkPlace->next)->prev = lkNew;
    lkPlace->next = lkNew;
    lkNew->prev = lkPlace;
    return 1;
}

/**
 * xmlLinkGetData:
 * @lk:  a link
 *
 * See Returns.
 *
 * Returns a pointer to the data referenced from this link
 */
void *
xmlLinkGetData(xmlLinkPtr lk)
{
    if (lk == NULL)
        return(NULL);
    return lk->data;
}

/**
 * xmlListReverse:
 * @l:  a list
 *
 * Reverse the order of the elements in the list
 */
void
xmlListReverse(xmlListPtr l)
{
    xmlLinkPtr lk;
    xmlLinkPtr lkPrev;

    if (l == NULL)
        return;
    lkPrev = l->sentinel;
    for (lk = l->sentinel->next; lk != l->sentinel; lk = lk->next) {
        lkPrev->next = lkPrev->prev;
        lkPrev->prev = lk;
```

```
        lkPrev = lk;
    }
    /* Fix up the last node */
    lkPrev->next = lkPrev->prev;
    lkPrev->prev = lk;
}

/**
 * xmlListSort:
 * @l:  a list
 *
 * Sort all the elements in the list
 */
void
xmlListSort(xmlListPtr l)
{
    xmlListPtr lTemp;

    if (l == NULL)
        return;
    if(xmlListEmpty(l))
        return;

    /* I think that the real answer is to implement quicksort, the
     * alternative is to implement some list copying procedure which
     * would be based on a list copy followed by a clear followed by
     * an insert. This is slow...
     */

    if (NULL ==(lTemp = xmlListDup(l)))
        return;
    xmlListClear(l);
    xmlListMerge(l, lTemp);
    xmlListDelete(lTemp);
    return;
}

/**
 * xmlListWalk:
 * @l:  a list
 * @walker:  a processing function
 * @user:  a user parameter passed to the walker function
 *
 * Walk all the element of the first from first to last and
 * apply the walker function to it
 */
void
xmlListWalk(xmlListPtr l, xmlListWalker walker, const void *user) {
    xmlLinkPtr lk;

    if ((l == NULL) || (walker == NULL))
        return;
    for(lk = l->sentinel->next; lk != l->sentinel; lk = lk->next) {
        if((walker(lk->data, user)) == 0)
                break;
    }
}

/**
 * xmlListReverseWalk:
 * @l:  a list
 * @walker:  a processing function
 * @user:  a user parameter passed to the walker function
 *
 * Walk all the element of the list in reverse order and
 * apply the walker function to it
 */
void
xmlListReverseWalk(xmlListPtr l, xmlListWalker walker, const void *user) {
    xmlLinkPtr lk;

    if ((l == NULL) || (walker == NULL))
        return;
    for(lk = l->sentinel->prev; lk != l->sentinel; lk = lk->prev) {
        if((walker(lk->data, user)) == 0)
                break;
    }
}

/**
 * xmlListMerge:
 * @l1:  the original list
 * @l2:  the new list
 *
 * include all the elements of the second list in the first one and
 * clear the second list
 */
void
xmlListMerge(xmlListPtr l1, xmlListPtr l2)
{
    xmlListCopy(l1, l2);
    xmlListClear(l2);
}

/**
 * xmlListDup:
 * @old:  the list
 *
```

```
 * Duplicate the list
 *
 * Returns a new copy of the list or NULL in case of error
 */
xmlListPtr
xmlListDup(const xmlListPtr old)
{
    xmlListPtr cur;

    if (old == NULL)
        return(NULL);
    /* Hmmm, how to best deal with allocation issues when copying
     * lists. If there is a de-allocator, should responsibility lie with
     * the new list or the old list. Surely not both. I'll arbitrarily
     * set it to be the old list for the time being whilst I work out
     * the answer
     */
    if (NULL ==(cur = xmlListCreate(NULL, old->linkCompare)))
        return (NULL);
    if (0 != xmlListCopy(cur, old))
        return NULL;
    return cur;
}

/**
 * xmlListCopy:
 * @cur:  the new list
 * @old:  the old list
 *
 * Move all the element from the old list in the new list
 *
 * Returns 0 in case of success 1 in case of error
 */
int
xmlListCopy(xmlListPtr cur, const xmlListPtr old)
{
    /* Walk the old tree and insert the data into the new one */
    xmlLinkPtr lk;

    if ((old == NULL) || (cur == NULL))
        return(1);
    for(lk = old->sentinel->next; lk != old->sentinel; lk = lk->next) {
        if (0 !=xmlListInsert(cur, lk->data)) {
            xmlListDelete(cur);
            return (1);
        }
    }
    return (0);
}
/* xmlListUnique() */
/* xmlListSwap */
#define bottom_list
#include "elfgcchack.h"
```

Notice for package(s)

libxml2

```
/*************************************************************************
 *
 * $Id$
 *
 * Copyright (C) 1998 Bjorn Reese and Daniel Stenberg.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
 * MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND
 * CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.
 *
 ************************************************************************
 *
 * A note to trio contributors:
 *
 * Avoid heap allocation at all costs to ensure that the trio functions
 * are async-safe. The exceptions are the printf/fprintf functions, which
 * uses fputc, and the asprintf functions and the <alloc> modifier, which
 * by design are required to allocate form the heap.
 *
 ************************************************************************/

/*
 * TODO:
 *  - Scan is probably too permissive about its modifiers.
 *  - C escapes in %#[] ?
 *  - Multibyte characters (done for format parsing, except scan groups)
 *  - Complex numbers? (C99 _Complex)
 *  - Boolean values? (C99 _Bool)
```

```
 *  - C99 NaN(n-char-sequence) missing. The n-char-sequence can be used
 *    to print the mantissa, e.g. NaN(0xc000000000000000)
 *  - Should we support the GNU %a alloc modifier? GNU has an ugly hack
 *    for %a, because C99 used %a for other purposes. If specified as
 *    %as or %a[ it is interpreted as the alloc modifier, otherwise as
 *    the C99 hex-float. This means that you cannot scan %as as a hex-float
 *    immediately followed by an 's'.
 *  - Scanning of collating symbols.
 */

/*************************************************************************
 * Trio include files
 */
#include "triodef.h"
#include "trio.h"
#include "triop.h"
#include "trionan.h"
#if !defined(TRIO_MINIMAL)
# include "triostr.h"
#endif

/*************************************************************************
 *
 * Definitions
 *
 ************************************************************************/

#include <math.h>
#include <limits.h>
#include <float.h>

#if (defined(__STDC_ISO_10646__) || defined(MB_LEN_MAX) \
     || defined(USE_MULTIBYTE) || TRIO_WIDECHAR) \
    && !defined(_WIN32_WCE)
# define TRIO_COMPILER_SUPPORTS_MULTIBYTE
# if !defined(MB_LEN_MAX)
#  define MB_LEN_MAX 6
# endif
#endif

#if (defined(TRIO_COMPILER_MSVC) && (_MSC_VER >= 1100)) || defined(TRIO_COMPILER_BCB)
# define TRIO_COMPILER_SUPPORTS_MSVC_INT
#endif

#if defined(_WIN32_WCE)
#include <wincecompat.h>
#endif

/*************************************************************************
 * Generic definitions
 */

#if !(defined(DEBUG) || defined(NDEBUG))
# define NDEBUG
#endif

#include <assert.h>
#include <ctype.h>
#if !defined(TRIO_COMPILER_SUPPORTS_C99)
# define isblank(x) (((x)==32) || ((x)==9))
#endif
#if defined(TRIO_COMPILER_ANCIENT)
# include <varargs.h>
#else
# include <stdarg.h>
#endif
#include <stddef.h>

#if defined( HAVE_ERRNO_H ) || defined( __VMS )
#include <errno.h>
#endif

#ifndef NULL
# define NULL 0
#endif
#define NIL ((char)0)
#ifndef FALSE
# define FALSE (1 == 0)
# define TRUE (! FALSE)
#endif
#define BOOLEAN_T int

/* mincore() can be used for debugging purposes */
#define VALID(x) (NULL != (x))

#if TRIO_ERRORS
  /*
   * Encode the error code and the position. This is decoded
   * with TRIO_ERROR_CODE and TRIO_ERROR_POSITION.
   */
# define TRIO_ERROR_RETURN(x,y) (- ((x) + ((y) << 8)))
#else
# define TRIO_ERROR_RETURN(x,y) (-1)
#endif

#ifndef VA_LIST_IS_ARRAY
#define TRIO_VA_LIST_PTR        va_list *
#define TRIO_VA_LIST_ADDR(l)    (&(l))
```

```
#define TRIO_VA_LIST_DEREF(l)   (*(l))
#else
#define TRIO_VA_LIST_PTR        va_list
#define TRIO_VA_LIST_ADDR(l)    (l)
#define TRIO_VA_LIST_DEREF(l)   (l)
#endif

typedef unsigned long trio_flags_t;


/*************************************************************************
 * Platform specific definitions
 */
#if defined(TRIO_PLATFORM_UNIX) || defined(TRIO_PLATFORM_OS400)
# include <unistd.h>
# include <signal.h>
# include <locale.h>
# define USE_LOCALE
#endif /* TRIO_PLATFORM_UNIX */
#if defined(TRIO_PLATFORM_VMS)
# include <unistd.h>
#endif
#if defined(TRIO_PLATFORM_WIN32)
# if defined(_WIN32_WCE)
#  include <wincecompat.h>
# else
#  include <io.h>
#  define read _read
#  define write _write
# endif
#endif /* TRIO_PLATFORM_WIN32 */

#if TRIO_WIDECHAR
# if defined(TRIO_COMPILER_SUPPORTS_ISO94)
#  include <wchar.h>
#  include <wctype.h>
typedef wchar_t trio_wchar_t;
typedef wint_t trio_wint_t;
# else
typedef char trio_wchar_t;
typedef int trio_wint_t;
#  define WCONST(x) L ## x
#  define WEOF EOF
#  define iswalnum(x) isalnum(x)
#  define iswalpha(x) isalpha(x)
#  define iswblank(x) isblank(x)
#  define iswcntrl(x) iscntrl(x)
#  define iswdigit(x) isdigit(x)
#  define iswgraph(x) isgraph(x)
#  define iswlower(x) islower(x)
#  define iswprint(x) isprint(x)
#  define iswpunct(x) ispunct(x)
#  define iswspace(x) isspace(x)
#  define iswupper(x) isupper(x)
#  define iswxdigit(x) isxdigit(x)
# endif
#endif


/*************************************************************************
 * Compiler dependent definitions
 */

/* Support for long long */
#ifndef __cplusplus
# if !defined(USE_LONGLONG)
#  if defined(TRIO_COMPILER_GCC) && !defined(__STRICT_ANSI__)
#   define USE_LONGLONG
#  elif defined(TRIO_COMPILER_SUNPRO)
#   define USE_LONGLONG
#  elif defined(_LONG_LONG) || defined(_LONGLONG)
#   define USE_LONGLONG
#  endif
# endif
#endif

/* The extra long numbers */
#if defined(USE_LONGLONG)
typedef signed long long int trio_longlong_t;
typedef unsigned long long int trio_ulonglong_t;
#elif defined(TRIO_COMPILER_SUPPORTS_MSVC_INT)
typedef signed __int64 trio_longlong_t;
typedef unsigned __int64 trio_ulonglong_t;
#else
typedef TRIO_SIGNED long int trio_longlong_t;
typedef unsigned long int trio_ulonglong_t;
#endif

/* Maximal and fixed integer types */
#if defined(TRIO_COMPILER_SUPPORTS_C99) && !defined( __VMS )
# include <stdint.h>
typedef intmax_t trio_intmax_t;
typedef uintmax_t trio_uintmax_t;
typedef int8_t trio_int8_t;
typedef int16_t trio_int16_t;
typedef int32_t trio_int32_t;
typedef int64_t trio_int64_t;
#elif defined(TRIO_COMPILER_SUPPORTS_UNIX98) || defined( __VMS )
```

```
# include <inttypes.h>
#ifdef __VMS
typedef long long int          intmax_t;
typedef unsigned long long int uintmax_t;
#endif
typedef intmax_t trio_intmax_t;
typedef uintmax_t trio_uintmax_t;
typedef int8_t trio_int8_t;
typedef int16_t trio_int16_t;
typedef int32_t trio_int32_t;
typedef int64_t trio_int64_t;
#elif defined(TRIO_COMPILER_SUPPORTS_MSVC_INT)
typedef trio_longlong_t trio_intmax_t;
typedef trio_ulonglong_t trio_uintmax_t;
typedef __int8 trio_int8_t;
typedef __int16 trio_int16_t;
typedef __int32 trio_int32_t;
typedef __int64 trio_int64_t;
#else
typedef trio_longlong_t trio_intmax_t;
typedef trio_ulonglong_t trio_uintmax_t;
# if defined(TRIO_INT8_T)
typedef TRIO_INT8_T trio_int8_t;
# else
typedef TRIO_SIGNED char trio_int8_t;
# endif
# if defined(TRIO_INT16_T)
typedef TRIO_INT16_T trio_int16_t;
# else
typedef TRIO_SIGNED short trio_int16_t;
# endif
# if defined(TRIO_INT32_T)
typedef TRIO_INT32_T trio_int32_t;
# else
typedef TRIO_SIGNED int trio_int32_t;
# endif
# if defined(TRIO_INT64_T)
typedef TRIO_INT64_T trio_int64_t;
# else
typedef trio_longlong_t trio_int64_t;
# endif
#endif

#if (!(defined(TRIO_COMPILER_SUPPORTS_C99) \
 || defined(TRIO_COMPILER_SUPPORTS_UNIX01))) \
 && !defined(_WIN32_WCE)
# define floorl(x) floor((double)(x))
# define fmodl(x,y) fmod((double)(x),(double)(y))
# define powl(x,y) pow((double)(x),(double)(y))
#endif

#define TRIO_FABS(x) (((x) < 0.0) ? -(x) : (x))

/*************************************************************************
 * Internal Definitions
 */

#ifndef DECIMAL_DIG
# define DECIMAL_DIG DBL_DIG
#endif

/* Long double sizes */
#ifdef LDBL_DIG
# define MAX_MANTISSA_DIGITS LDBL_DIG
# define MAX_EXPONENT_DIGITS 4
# define MAX_DOUBLE_DIGITS LDBL_MAX_10_EXP
#else
# define MAX_MANTISSA_DIGITS DECIMAL_DIG
# define MAX_EXPONENT_DIGITS 3
# define MAX_DOUBLE_DIGITS DBL_MAX_10_EXP
#endif

#if defined(TRIO_COMPILER_ANCIENT) || !defined(LDBL_DIG)
# undef LDBL_DIG
# undef LDBL_MANT_DIG
# undef LDBL_EPSILON
# define LDBL_DIG DBL_DIG
# define LDBL_MANT_DIG DBL_MANT_DIG
# define LDBL_EPSILON DBL_EPSILON
#endif

/* The maximal number of digits is for base 2 */
#define MAX_CHARS_IN(x) (sizeof(x) * CHAR_BIT)
/* The width of a pointer. The number of bits in a hex digit is 4 */
#define POINTER_WIDTH ((sizeof("0x") - 1) + sizeof(trio_pointer_t) * CHAR_BIT / 4)

/* Infinite and Not-A-Number for floating-point */
#define INFINITE_LOWER "inf"
#define INFINITE_UPPER "INF"
#define LONG_INFINITE_LOWER "infinite"
#define LONG_INFINITE_UPPER "INFINITE"
#define NAN_LOWER "nan"
#define NAN_UPPER "NAN"

#if !defined(HAVE_ISASCII) && !defined(isascii)
#ifndef __VMS
# define isascii(x) ((unsigned int)(x) < 128)
#endif
```

```
#endif

/* Various constants */
enum {
  TYPE_PRINT = 1,
  TYPE_SCAN  = 2,

  /* Flags. FLAGS_LAST must be less than ULONG_MAX */
  FLAGS_NEW                 = 0,
  FLAGS_STICKY              = 1,
  FLAGS_SPACE               = 2 * FLAGS_STICKY,
  FLAGS_SHOWSIGN            = 2 * FLAGS_SPACE,
  FLAGS_LEFTADJUST          = 2 * FLAGS_SHOWSIGN,
  FLAGS_ALTERNATIVE         = 2 * FLAGS_LEFTADJUST,
  FLAGS_SHORT               = 2 * FLAGS_ALTERNATIVE,
  FLAGS_SHORTSHORT          = 2 * FLAGS_SHORT,
  FLAGS_LONG                = 2 * FLAGS_SHORTSHORT,
  FLAGS_QUAD                = 2 * FLAGS_LONG,
  FLAGS_LONGDOUBLE          = 2 * FLAGS_QUAD,
  FLAGS_SIZE_T              = 2 * FLAGS_LONGDOUBLE,
  FLAGS_PTRDIFF_T           = 2 * FLAGS_SIZE_T,
  FLAGS_INTMAX_T            = 2 * FLAGS_PTRDIFF_T,
  FLAGS_NILPADDING          = 2 * FLAGS_INTMAX_T,
  FLAGS_UNSIGNED            = 2 * FLAGS_NILPADDING,
  FLAGS_UPPER               = 2 * FLAGS_UNSIGNED,
  FLAGS_WIDTH               = 2 * FLAGS_UPPER,
  FLAGS_WIDTH_PARAMETER     = 2 * FLAGS_WIDTH,
  FLAGS_PRECISION           = 2 * FLAGS_WIDTH_PARAMETER,
  FLAGS_PRECISION_PARAMETER = 2 * FLAGS_PRECISION,
  FLAGS_BASE                = 2 * FLAGS_PRECISION_PARAMETER,
  FLAGS_BASE_PARAMETER      = 2 * FLAGS_BASE,
  FLAGS_FLOAT_E             = 2 * FLAGS_BASE_PARAMETER,
  FLAGS_FLOAT_G             = 2 * FLAGS_FLOAT_E,
  FLAGS_QUOTE               = 2 * FLAGS_FLOAT_G,
  FLAGS_WIDECHAR            = 2 * FLAGS_QUOTE,
  FLAGS_ALLOC               = 2 * FLAGS_WIDECHAR,
  FLAGS_IGNORE              = 2 * FLAGS_ALLOC,
  FLAGS_IGNORE_PARAMETER    = 2 * FLAGS_IGNORE,
  FLAGS_VARSIZE_PARAMETER   = 2 * FLAGS_IGNORE_PARAMETER,
  FLAGS_FIXED_SIZE          = 2 * FLAGS_VARSIZE_PARAMETER,
  FLAGS_LAST                = FLAGS_FIXED_SIZE,
  /* Reused flags */
  FLAGS_EXCLUDE             = FLAGS_SHORT,
  FLAGS_USER_DEFINED        = FLAGS_IGNORE,
  FLAGS_ROUNDING            = FLAGS_INTMAX_T,
  /* Compounded flags */
  FLAGS_ALL_VARSIZES        = FLAGS_LONG | FLAGS_QUAD | FLAGS_INTMAX_T | FLAGS_PTRDIFF_T | FLAGS_SIZE_T,
  FLAGS_ALL_SIZES           = FLAGS_ALL_VARSIZES | FLAGS_SHORTSHORT | FLAGS_SHORT,

  NO_POSITION  = -1,
  NO_WIDTH     =  0,
  NO_PRECISION = -1,
  NO_SIZE      = -1,

  /* Do not change these */
  NO_BASE      = -1,
  MIN_BASE     =  2,
  MAX_BASE     = 36,
  BASE_BINARY  =  2,
  BASE_OCTAL   =  8,
  BASE_DECIMAL = 10,
  BASE_HEX     = 16,

  /* Maximal number of allowed parameters */
  MAX_PARAMETERS = 64,
  /* Maximal number of characters in class */
  MAX_CHARACTER_CLASS = UCHAR_MAX + 1,

  /* Maximal string lengths for user-defined specifiers */
  MAX_USER_NAME = 64,
  MAX_USER_DATA = 256,

  /* Maximal length of locale separator strings */
  MAX_LOCALE_SEPARATOR_LENGTH = MB_LEN_MAX,
  /* Maximal number of integers in grouping */
  MAX_LOCALE_GROUPS = 64,

  /* Initial size of asprintf buffer */
  DYNAMIC_START_SIZE = 32
};

#define NO_GROUPING ((int)CHAR_MAX)

/* Fundamental formatting parameter types */
#define FORMAT_UNKNOWN   0
#define FORMAT_INT       1
#define FORMAT_DOUBLE    2
#define FORMAT_CHAR      3
#define FORMAT_STRING    4
#define FORMAT_POINTER   5
#define FORMAT_COUNT     6
#define FORMAT_PARAMETER 7
#define FORMAT_GROUP     8
#if TRIO_GNU
# define FORMAT_ERRNO    9
#endif
#if TRIO_EXTENSION
# define FORMAT_USER_DEFINED 10
```

```
#endif

/* Character constants */
#define CHAR_IDENTIFIER '%'
#define CHAR_BACKSLASH '\\'
#define CHAR_QUOTE '\"'
#define CHAR_ADJUST ' '

/* Character class expressions */
#define CLASS_ALNUM "[:alnum:]"
#define CLASS_ALPHA "[:alpha:]"
#define CLASS_BLANK "[:blank:]"
#define CLASS_CNTRL "[:cntrl:]"
#define CLASS_DIGIT "[:digit:]"
#define CLASS_GRAPH "[:graph:]"
#define CLASS_LOWER "[:lower:]"
#define CLASS_PRINT "[:print:]"
#define CLASS_PUNCT "[:punct:]"
#define CLASS_SPACE "[:space:]"
#define CLASS_UPPER "[:upper:]"
#define CLASS_XDIGIT "[:xdigit:]"

/*
 * SPECIFIERS:
 *
 *
 * a  Hex-float
 * A  Hex-float
 * c  Character
 * C  Widechar character (wint_t)
 * d  Decimal
 * e  Float
 * E  Float
 * F  Float
 * F  Float
 * g  Float
 * G  Float
 * i  Integer
 * m  Error message
 * n  Count
 * o  Octal
 * p  Pointer
 * s  String
 * S  Widechar string (wchar_t *)
 * u  Unsigned
 * x  Hex
 * X  Hex
 * [] Group
 * <> User-defined
 *
 * Reserved:
 *
 * D  Binary Coded Decimal %D(length,precision) (OS/390)
 */
#define SPECIFIER_CHAR 'c'
#define SPECIFIER_STRING 's'
#define SPECIFIER_DECIMAL 'd'
#define SPECIFIER_INTEGER 'i'
#define SPECIFIER_UNSIGNED 'u'
#define SPECIFIER_OCTAL 'o'
#define SPECIFIER_HEX 'x'
#define SPECIFIER_HEX_UPPER 'X'
#define SPECIFIER_FLOAT_E 'e'
#define SPECIFIER_FLOAT_E_UPPER 'E'
#define SPECIFIER_FLOAT_F 'f'
#define SPECIFIER_FLOAT_F_UPPER 'F'
#define SPECIFIER_FLOAT_G 'g'
#define SPECIFIER_FLOAT_G_UPPER 'G'
#define SPECIFIER_POINTER 'p'
#define SPECIFIER_GROUP '['
#define SPECIFIER_UNGROUP ']'
#define SPECIFIER_COUNT 'n'
#if TRIO_UNIX98
# define SPECIFIER_CHAR_UPPER 'C'
# define SPECIFIER_STRING_UPPER 'S'
#endif
#if TRIO_C99
# define SPECIFIER_HEXFLOAT 'a'
# define SPECIFIER_HEXFLOAT_UPPER 'A'
#endif
#if TRIO_GNU
# define SPECIFIER_ERRNO 'm'
#endif
#if TRIO_EXTENSION
# define SPECIFIER_BINARY 'b'
# define SPECIFIER_BINARY_UPPER 'B'
# define SPECIFIER_USER_DEFINED_BEGIN '<'
# define SPECIFIER_USER_DEFINED_END '>'
# define SPECIFIER_USER_DEFINED_SEPARATOR ':'
#endif

/*
 * QUALIFIERS:
 *
 *
 * Numbers = d,i,o,u,x,X
 * Float = a,A,e,E,f,F,g,G
 * String = s
```

```
 * Char = c
 *
 *
 * 9$ Position
 *      Use the 9th parameter. 9 can be any number between 1 and
 *      the maximal argument
 *
 * 9 Width
 *      Set width to 9. 9 can be any number, but must not be postfixed
 *      by '$'
 *
 * h  Short
 *    Numbers:
 *      (unsigned) short int
 *
 * hh Short short
 *    Numbers:
 *      (unsigned) char
 *
 * l  Long
 *    Numbers:
 *      (unsigned) long int
 *    String:
 *      as the S specifier
 *    Char:
 *      as the C specifier
 *
 * ll Long Long
 *    Numbers:
 *      (unsigned) long long int
 *
 * L  Long Double
 *    Float
 *      long double
 *
 * #  Alternative
 *    Float:
 *      Decimal-point is always present
 *    String:
 *      non-printable characters are handled as \number
 *
 *    Spacing
 *
 * +  Sign
 *
 * -  Alignment
 *
 * .  Precision
 *
 * *  Parameter
 *    print: use parameter
 *    scan: no parameter (ignore)
 *
 * q  Quad
 *
 * Z  size_t
 *
 * w  Widechar
 *
 * '  Thousands/quote
 *    Numbers:
 *      Integer part grouped in thousands
 *    Binary numbers:
 *      Number grouped in nibbles (4 bits)
 *    String:
 *      Quoted string
 *
 * j  intmax_t
 * t  prtdiff_t
 * z  size_t
 *
 * !  Sticky
 * @  Parameter (for both print and scan)
 *
 * I  n-bit Integer
 *    Numbers:
 *      The following options exists
 *        I8  = 8-bit integer
 *        I16 = 16-bit integer
 *        I32 = 32-bit integer
 *        I64 = 64-bit integer
 */
#define QUALIFIER_POSITION '$'
#define QUALIFIER_SHORT 'h'
#define QUALIFIER_LONG 'l'
#define QUALIFIER_LONG_UPPER 'L'
#define QUALIFIER_ALTERNATIVE '#'
#define QUALIFIER_SPACE ' '
#define QUALIFIER_PLUS '+'
#define QUALIFIER_MINUS '-'
#define QUALIFIER_DOT '.'
#define QUALIFIER_STAR '*'
#define QUALIFIER_CIRCUMFLEX '^' /* For scanlists */
#if TRIO_C99
# define QUALIFIER_SIZE_T 'z'
# define QUALIFIER_PTRDIFF_T 't'
# define QUALIFIER_INTMAX_T 'j'
#endif
```

```
#if TRIO_BSD || TRIO_GNU
# define QUALIFIER_QUAD 'q'
#endif
#if TRIO_GNU
# define QUALIFIER_SIZE_T_UPPER 'Z'
#endif
#if TRIO_MISC
# define QUALIFIER_WIDECHAR 'w'
#endif
#if TRIO_MICROSOFT
# define QUALIFIER_FIXED_SIZE 'I'
#endif
#if TRIO_EXTENSION
# define QUALIFIER_QUOTE '\''
# define QUALIFIER_STICKY '!'
# define QUALIFIER_VARSIZE '&' /* This should remain undocumented */
# define QUALIFIER_PARAM '@' /* Experimental */
# define QUALIFIER_COLON ':' /* For scanlists */
# define QUALIFIER_EQUAL '=' /* For scanlists */
# define QUALIFIER_ROUNDING_UPPER 'R'
#endif


/*************************************************************************
 *
 * Internal Structures
 *
 ************************************************************************/

/* Parameters */
typedef struct {
  /* An indication of which entry in the data union is used */
  int type;
  /* The flags */
  trio_flags_t flags;
  /* The width qualifier */
  int width;
  /* The precision qualifier */
  int precision;
  /* The base qualifier */
  int base;
  /* The size for the variable size qualifier */
  int varsize;
  /* The marker of the end of the specifier */
  int indexAfterSpecifier;
  /* The data from the argument list */
  union {
    char *string;
#if TRIO_WIDECHAR
    trio_wchar_t *wstring;
#endif
    trio_pointer_t pointer;
    union {
      trio_intmax_t as_signed;
      trio_uintmax_t as_unsigned;
    } number;
    double doubleNumber;
    double *doublePointer;
    trio_long_double_t longdoubleNumber;
    trio_long_double_t *longdoublePointer;
    int errorNumber;
  } data;
  /* For the user-defined specifier */
  char user_name[MAX_USER_NAME];
  char user_data[MAX_USER_DATA];
} trio_parameter_t;

/* Container for customized functions */
typedef struct {
  union {
    trio_outstream_t out;
    trio_instream_t in;
  } stream;
  trio_pointer_t closure;
} trio_custom_t;

/* General trio "class" */
typedef struct _trio_class_t {
  /*
   * The function to write characters to a stream.
   */
  void (*OutStream) TRIO_PROTO((struct _trio_class_t *, int));
  /*
   * The function to read characters from a stream.
   */
  void (*InStream) TRIO_PROTO((struct _trio_class_t *, int *));
  /*
   * The current location in the stream.
   */
  trio_pointer_t location;
  /*
   * The character currently being processed.
   */
  int current;
  /*
   * The number of characters that would have been written/read
   * if there had been sufficient space.
   */
```

```
  int processed;
  /*
   * The number of characters that are actually written/read.
   * Processed and committed will only differ for the *nprintf
   * and *nscanf functions.
   */
  int committed;
  /*
   * The upper limit of characters that may be written/read.
   */
  int max;
  /*
   * The last output error that was detected.
   */
  int error;
} trio_class_t;

/* References (for user-defined callbacks) */
typedef struct _trio_reference_t {
  trio_class_t *data;
  trio_parameter_t *parameter;
} trio_reference_t;

/* Registered entries (for user-defined callbacks) */
typedef struct _trio_userdef_t {
  struct _trio_userdef_t *next;
  trio_callback_t callback;
  char *name;
} trio_userdef_t;

/*************************************************************************
 *
 * Internal Variables
 *
 ************************************************************************/

static TRIO_CONST char rcsid[] = "@(#)$Id$";

/*
 * Need this to workaround a parser bug in HP C/iX compiler that fails
 * to resolves macro definitions that includes type 'long double',
 * e.g: va_arg(arg_ptr, long double)
 */
#if defined(TRIO_PLATFORM_MPEIX)
static TRIO_CONST trio_long_double_t ___dummy_long_double = 0;
#endif

static TRIO_CONST char internalNullString[] = "(nil)";

#if defined(USE_LOCALE)
static struct lconv *internalLocaleValues = NULL;
#endif

/*
 * UNIX98 says "in a locale where the radix character is not defined,
 * the radix character defaults to a period (.)"
 */
static int internalDecimalPointLength = 1;
static int internalThousandSeparatorLength = 1;
static char internalDecimalPoint = '.';
static char internalDecimalPointString[MAX_LOCALE_SEPARATOR_LENGTH + 1] = ".";
static char internalThousandSeparator[MAX_LOCALE_SEPARATOR_LENGTH + 1] = ",";
static char internalGrouping[MAX_LOCALE_GROUPS] = { (char)NO_GROUPING };

static TRIO_CONST char internalDigitsLower[] = "0123456789abcdefghijklmnopqrstuvwxyz";
static TRIO_CONST char internalDigitsUpper[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
static BOOLEAN_T internalDigitsUnconverted = TRUE;
static int internalDigitArray[128];
#if TRIO_EXTENSION
static BOOLEAN_T internalCollationUnconverted = TRUE;
static char internalCollationArray[MAX_CHARACTER_CLASS][MAX_CHARACTER_CLASS];
#endif

#if TRIO_EXTENSION
static TRIO_VOLATILE trio_callback_t internalEnterCriticalRegion = NULL;
static TRIO_VOLATILE trio_callback_t internalLeaveCriticalRegion = NULL;
static trio_userdef_t *internalUserDef = NULL;
#endif


/*************************************************************************
 *
 * Internal Functions
 *
 ************************************************************************/

#if defined(TRIO_MINIMAL)
# define TRIO_STRING_PUBLIC static
# include "triostr.c"
#endif /* defined(TRIO_MINIMAL) */

/*************************************************************************
 * TrioIsQualifier
 *
 * Description:
 *  Remember to add all new qualifiers to this function.
 *  QUALIFIER_POSITION must not be added.
 */
```

```
TRIO_PRIVATE BOOLEAN_T
TrioIsQualifier
TRIO_ARGS1((character),
           TRIO_CONST char character)
{
  /* QUALIFIER_POSITION is not included */
  switch (character)
    {
    case '0': case '1': case '2': case '3': case '4':
    case '5': case '6': case '7': case '8': case '9':
    case QUALIFIER_PLUS:
    case QUALIFIER_MINUS:
    case QUALIFIER_SPACE:
    case QUALIFIER_DOT:
    case QUALIFIER_STAR:
    case QUALIFIER_ALTERNATIVE:
    case QUALIFIER_SHORT:
    case QUALIFIER_LONG:
    case QUALIFIER_LONG_UPPER:
    case QUALIFIER_CIRCUMFLEX:
#if defined(QUALIFIER_SIZE_T)
    case QUALIFIER_SIZE_T:
#endif
#if defined(QUALIFIER_PTRDIFF_T)
    case QUALIFIER_PTRDIFF_T:
#endif
#if defined(QUALIFIER_INTMAX_T)
    case QUALIFIER_INTMAX_T:
#endif
#if defined(QUALIFIER_QUAD)
    case QUALIFIER_QUAD:
#endif
#if defined(QUALIFIER_SIZE_T_UPPER)
    case QUALIFIER_SIZE_T_UPPER:
#endif
#if defined(QUALIFIER_WIDECHAR)
    case QUALIFIER_WIDECHAR:
#endif
#if defined(QUALIFIER_QUOTE)
    case QUALIFIER_QUOTE:
#endif
#if defined(QUALIFIER_STICKY)
    case QUALIFIER_STICKY:
#endif
#if defined(QUALIFIER_VARSIZE)
    case QUALIFIER_VARSIZE:
#endif
#if defined(QUALIFIER_PARAM)
    case QUALIFIER_PARAM:
#endif
#if defined(QUALIFIER_FIXED_SIZE)
    case QUALIFIER_FIXED_SIZE:
#endif
#if defined(QUALIFIER_ROUNDING_UPPER)
    case QUALIFIER_ROUNDING_UPPER:
#endif
      return TRUE;
    default:
      return FALSE;
    }
}

/*************************************************************************
 * TrioSetLocale
 */
#if defined(USE_LOCALE)
TRIO_PRIVATE void
TrioSetLocale(TRIO_NOARGS)
{
  internalLocaleValues = (struct lconv *)localeconv();
  if (internalLocaleValues)
    {
      if ((internalLocaleValues->decimal_point) &&
          (internalLocaleValues->decimal_point[0] != NIL))
        {
          internalDecimalPointLength = trio_length(internalLocaleValues->decimal_point);
          if (internalDecimalPointLength == 1)
            {
              internalDecimalPoint = internalLocaleValues->decimal_point[0];
            }
          else
            {
              internalDecimalPoint = NIL;
              trio_copy_max(internalDecimalPointString,
                            sizeof(internalDecimalPointString),
                            internalLocaleValues->decimal_point);
            }
        }
      if ((internalLocaleValues->thousands_sep) &&
          (internalLocaleValues->thousands_sep[0] != NIL))
        {
          trio_copy_max(internalThousandSeparator,
                        sizeof(internalThousandSeparator),
                        internalLocaleValues->thousands_sep);
          internalThousandSeparatorLength = trio_length(internalThousandSeparator);
        }
      if ((internalLocaleValues->grouping) &&
          (internalLocaleValues->grouping[0] != NIL))
```

```
        {
          trio_copy_max(internalGrouping,
                        sizeof(internalGrouping),
                        internalLocaleValues->grouping);
        }
    }
}
#endif /* defined(USE_LOCALE) */

TRIO_PRIVATE int
TrioCalcThousandSeparatorLength
TRIO_ARGS1((digits),
           int digits)
{
#if TRIO_EXTENSION
  int count = 0;
  int step = NO_GROUPING;
  char *groupingPointer = internalGrouping;

  while (digits > 0)
    {
      if (*groupingPointer == CHAR_MAX)
        {
          /* Disable grouping */
          break; /* while */
        }
      else if (*groupingPointer == 0)
        {
          /* Repeat last group */
          if (step == NO_GROUPING)
            {
              /* Error in locale */
              break; /* while */
            }
        }
      else
        {
          step = *groupingPointer++;
        }
      if (digits > step)
        count += internalThousandSeparatorLength;
      digits -= step;
    }
  return count;
#else
  return 0;
#endif
}

TRIO_PRIVATE BOOLEAN_T
TrioFollowedBySeparator
TRIO_ARGS1((position),
           int position)
{
#if TRIO_EXTENSION
  int step = 0;
  char *groupingPointer = internalGrouping;

  position--;
  if (position == 0)
    return FALSE;
  while (position > 0)
    {
      if (*groupingPointer == CHAR_MAX)
        {
          /* Disable grouping */
          break; /* while */
        }
      else if (*groupingPointer != 0)
        {
          step = *groupingPointer++;
        }
      if (step == 0)
        break;
      position -= step;
    }
  return (position == 0);
#else
  return FALSE;
#endif
}

/*************************************************************************
 * TrioGetPosition
 *
 * Get the %n$ position.
 */
TRIO_PRIVATE int
TrioGetPosition
TRIO_ARGS2((format, indexPointer),
           TRIO_CONST char *format,
           int *indexPointer)
{
#if TRIO_UNIX98
  char *tmpformat;
  int number = 0;
  int index = *indexPointer;
```

```
  number = (int)trio_to_long(&format[index], &tmpformat, BASE_DECIMAL);
  index = (int)(tmpformat - format);
  if ((number != 0) && (QUALIFIER_POSITION == format[index++]))
    {
      *indexPointer = index;
      /*
       * number is decreased by 1, because n$ starts from 1, whereas
       * the array it is indexing starts from 0.
       */
      return number - 1;
    }
#endif
  return NO_POSITION;
}

#if TRIO_EXTENSION
/*************************************************************************
 * TrioFindNamespace
 *
 * Find registered user-defined specifier.
 * The prev argument is used for optimization only.
 */
TRIO_PRIVATE trio_userdef_t *
TrioFindNamespace
TRIO_ARGS2((name, prev),
           TRIO_CONST char *name,
           trio_userdef_t **prev)
{
  trio_userdef_t *def;

  if (internalEnterCriticalRegion)
    (void)internalEnterCriticalRegion(NULL);

  for (def = internalUserDef; def; def = def->next)
    {
      /* Case-sensitive string comparison */
      if (trio_equal_case(def->name, name))
        break;

      if (prev)
        *prev = def;
    }

  if (internalLeaveCriticalRegion)
    (void)internalLeaveCriticalRegion(NULL);

  return def;
}
#endif

/*************************************************************************
 * TrioPower
 *
 * Description:
 *  Calculate pow(base, exponent), where number and exponent are integers.
 */
TRIO_PRIVATE trio_long_double_t
TrioPower
TRIO_ARGS2((number, exponent),
           int number,
           int exponent)
{
  trio_long_double_t result;

  if (number == 10)
    {
      switch (exponent)
        {
          /* Speed up calculation of common cases */
        case 0:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E-1);
          break;
        case 1:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+0);
          break;
        case 2:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+1);
          break;
        case 3:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+2);
          break;
        case 4:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+3);
          break;
        case 5:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+4);
          break;
        case 6:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+5);
          break;
        case 7:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+6);
          break;
        case 8:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+7);
          break;
        case 9:
          result = (trio_long_double_t)number * TRIO_SUFFIX_LONG(1E+8);
```

```
          break;
        default:
          result = powl((trio_long_double_t)number,
                        (trio_long_double_t)exponent);
          break;
        }
    }
  else
    {
      return powl((trio_long_double_t)number, (trio_long_double_t)exponent);
    }
  return result;
}

/*************************************************************************
 * TrioLogarithm
 */
TRIO_PRIVATE double
TrioLogarithm
TRIO_ARGS2((number, base),
           double number,
           int base)
{
  double result;

  if (number <= 0.0)
    {
      /* xlC crashes on log(0) */
      result = (number == 0.0) ? trio_ninf() : trio_nan();
    }
  else
    {
      if (base == 10)
        {
          result = log10(number);
        }
      else
        {
          result = log10(number) / log10((double)base);
        }
    }
  return result;
}

/*************************************************************************
 * TrioLogarithmBase
 */
TRIO_PRIVATE double
TrioLogarithmBase
TRIO_ARGS1((base),
           int base)
{
  switch (base)
    {
    case BASE_BINARY : return 1.0;
    case BASE_OCTAL  : return 3.0;
    case BASE_DECIMAL: return 3.321928094887362345;
    case BASE_HEX    : return 4.0;
    default          : return TrioLogarithm((double)base, 2);
    }
}

/*************************************************************************
 * TrioParse
 *
 * Description:
 *  Parse the format string
 */
TRIO_PRIVATE int
TrioParse
TRIO_ARGS5((type, format, parameters, arglist, argarray),
           int type,
           TRIO_CONST char *format,
           trio_parameter_t *parameters,
           TRIO_VA_LIST_PTR arglist,
           trio_pointer_t *argarray)
{
  /* Count the number of times a parameter is referenced */
  unsigned short usedEntries[MAX_PARAMETERS];
  /* Parameter counters */
  int parameterPosition;
  int currentParam;
  int maxParam = -1;
  /* Utility variables */
  trio_flags_t flags;
  int width;
  int precision;
  int varsize;
  int base;
  int index;  /* Index into formatting string */
  int dots;  /* Count number of dots in modifier part */
  BOOLEAN_T positional;  /* Does the specifier have a positional? */
  BOOLEAN_T gotSticky = FALSE;  /* Are there any sticky modifiers at all? */
  /*
   * indices specifies the order in which the parameters must be
   * read from the va_args (this is necessary to handle positionals)
   */
  int indices[MAX_PARAMETERS];
```

```
  int pos = 0;
  /* Various variables */
  char ch;
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  int charlen;
#endif
  int save_errno;
  int i = -1;
  int num;
  char *tmpformat;

  /* One and only one of arglist and argarray must be used */
  assert((arglist != NULL) ^ (argarray != NULL));

  /*
   * The 'parameters' array is not initialized, but we need to
   * know which entries we have used.
   */
  memset(usedEntries, 0, sizeof(usedEntries));

  save_errno = errno;
  index = 0;
  parameterPosition = 0;
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  (void)mblen(NULL, 0);
#endif

  while (format[index])
    {
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
      if (! isascii(format[index]))
        {
          /*
           * Multibyte characters cannot be legal specifiers or
           * modifiers, so we skip over them.
           */
          charlen = mblen(&format[index], MB_LEN_MAX);
          index += (charlen > 0) ? charlen : 1;
          continue; /* while */
        }
#endif /* TRIO_COMPILER_SUPPORTS_MULTIBYTE */
      if (CHAR_IDENTIFIER == format[index++])
        {
          if (CHAR_IDENTIFIER == format[index])
            {
              index++;
              continue; /* while */
            }

          flags = FLAGS_NEW;
          dots = 0;
          currentParam = TrioGetPosition(format, &index);
          positional = (NO_POSITION != currentParam);
          if (!positional)
            {
              /* We have no positional, get the next counter */
              currentParam = parameterPosition;
            }
          if(currentParam >= MAX_PARAMETERS)
            {
              /* Bail out completely to make the error more obvious */
              return TRIO_ERROR_RETURN(TRIO_ETOOMANY, index);
            }

          if (currentParam > maxParam)
            maxParam = currentParam;

          /* Default values */
          width = NO_WIDTH;
          precision = NO_PRECISION;
          base = NO_BASE;
          varsize = NO_SIZE;

          while (TrioIsQualifier(format[index]))
            {
              ch = format[index++];

              switch (ch)
                {
                case QUALIFIER_SPACE:
                  flags |= FLAGS_SPACE;
                  break;

                case QUALIFIER_PLUS:
                  flags |= FLAGS_SHOWSIGN;
                  break;

                case QUALIFIER_MINUS:
                  flags |= FLAGS_LEFTADJUST;
                  flags &= ~FLAGS_NILPADDING;
                  break;

                case QUALIFIER_ALTERNATIVE:
                  flags |= FLAGS_ALTERNATIVE;
                  break;

                case QUALIFIER_DOT:
                  if (dots == 0) /* Precision */
```

```
                  {
                    dots++;

                    /* Skip if no precision */
                    if (QUALIFIER_DOT == format[index])
                      break;

                    /* After the first dot we have the precision */
                    flags |= FLAGS_PRECISION;
                    if ((QUALIFIER_STAR == format[index])
#if defined(QUALIFIER_PARAM)
                        || (QUALIFIER_PARAM == format[index])
#endif
                        )
                      {
                        index++;
                        flags |= FLAGS_PRECISION_PARAMETER;

                        precision = TrioGetPosition(format, &index);
                        if (precision == NO_POSITION)
                          {
                            parameterPosition++;
                            if (positional)
                              precision = parameterPosition;
                            else
                              {
                                precision = currentParam;
                                currentParam = precision + 1;
                              }
                          }
                        else
                          {
                            if (! positional)
                              currentParam = precision + 1;
                            if (width > maxParam)
                              maxParam = precision;
                          }
                        if (currentParam > maxParam)
                          maxParam = currentParam;
                      }
                    else
                      {
                        precision = trio_to_long(&format[index],
                                                 &tmpformat,
                                                 BASE_DECIMAL);
                        index = (int)(tmpformat - format);
                      }
                  }
                else if (dots == 1) /* Base */
                  {
                    dots++;

                    /* After the second dot we have the base */
                    flags |= FLAGS_BASE;
                    if ((QUALIFIER_STAR == format[index])
#if defined(QUALIFIER_PARAM)
                        || (QUALIFIER_PARAM == format[index])
#endif
                        )
                      {
                        index++;
                        flags |= FLAGS_BASE_PARAMETER;
                        base = TrioGetPosition(format, &index);
                        if (base == NO_POSITION)
                          {
                            parameterPosition++;
                            if (positional)
                              base = parameterPosition;
                            else
                              {
                                base = currentParam;
                                currentParam = base + 1;
                              }
                          }
                        else
                          {
                            if (! positional)
                              currentParam = base + 1;
                            if (base > maxParam)
                              maxParam = base;
                          }
                        if (currentParam > maxParam)
                          maxParam = currentParam;
                      }
                    else
                      {
                        base = trio_to_long(&format[index],
                                            &tmpformat,
                                            BASE_DECIMAL);
                        if (base > MAX_BASE)
                          return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                        index = (int)(tmpformat - format);
                      }
                  }
                else
                  {
                    return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                  }
```

```
                  break; /* QUALIFIER_DOT */

#if defined(QUALIFIER_PARAM)
                case QUALIFIER_PARAM:
                  type = TYPE_PRINT;
                  /* FALLTHROUGH */
#endif
                case QUALIFIER_STAR:
                  /* This has different meanings for print and scan */
                  if (TYPE_PRINT == type)
                    {
                      /* Read with from parameter */
                      flags |= (FLAGS_WIDTH | FLAGS_WIDTH_PARAMETER);
                      width = TrioGetPosition(format, &index);
                      if (width == NO_POSITION)
                        {
                          parameterPosition++;
                          if (positional)
                            width = parameterPosition;
                          else
                            {
                              width = currentParam;
                              currentParam = width + 1;
                            }
                        }
                      else
                        {
                          if (! positional)
                            currentParam = width + 1;
                          if (width > maxParam)
                            maxParam = width;
                        }
                      if (currentParam > maxParam)
                        maxParam = currentParam;
                    }
                  else
                    {
                      /* Scan, but do not store result */
                      flags |= FLAGS_IGNORE;
                    }

                  break; /* QUALIFIER_STAR */

                case '0':
                  if (! (flags & FLAGS_LEFTADJUST))
                    flags |= FLAGS_NILPADDING;
                  /* FALLTHROUGH */
                case '1': case '2': case '3': case '4':
                case '5': case '6': case '7': case '8': case '9':
                  flags |= FLAGS_WIDTH;
                  /* &format[index - 1] is used to "rewind" the read
                   * character from format
                   */
                  width = trio_to_long(&format[index - 1],
                                       &tmpformat,
                                       BASE_DECIMAL);
                  index = (int)(tmpformat - format);
                  break;

                case QUALIFIER_SHORT:
                  if (flags & FLAGS_SHORTSHORT)
                    return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                  else if (flags & FLAGS_SHORT)
                    flags |= FLAGS_SHORTSHORT;
                  else
                    flags |= FLAGS_SHORT;
                  break;

                case QUALIFIER_LONG:
                  if (flags & FLAGS_QUAD)
                    return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                  else if (flags & FLAGS_LONG)
                    flags |= FLAGS_QUAD;
                  else
                    flags |= FLAGS_LONG;
                  break;

                case QUALIFIER_LONG_UPPER:
                  flags |= FLAGS_LONGDOUBLE;
                  break;

#if defined(QUALIFIER_SIZE_T)
                case QUALIFIER_SIZE_T:
                  flags |= FLAGS_SIZE_T;
                  /* Modify flags for later truncation of number */
                  if (sizeof(size_t) == sizeof(trio_ulonglong_t))
                    flags |= FLAGS_QUAD;
                  else if (sizeof(size_t) == sizeof(long))
                    flags |= FLAGS_LONG;
                  break;
#endif

#if defined(QUALIFIER_PTRDIFF_T)
                case QUALIFIER_PTRDIFF_T:
                  flags |= FLAGS_PTRDIFF_T;
                  if (sizeof(ptrdiff_t) == sizeof(trio_ulonglong_t))
                    flags |= FLAGS_QUAD;
                  else if (sizeof(ptrdiff_t) == sizeof(long))
```

```
                  flags |= FLAGS_LONG;
                break;
#endif

#if defined(QUALIFIER_INTMAX_T)
              case QUALIFIER_INTMAX_T:
                flags |= FLAGS_INTMAX_T;
                if (sizeof(trio_intmax_t) == sizeof(trio_ulonglong_t))
                  flags |= FLAGS_QUAD;
                else if (sizeof(trio_intmax_t) == sizeof(long))
                  flags |= FLAGS_LONG;
                break;
#endif

#if defined(QUALIFIER_QUAD)
              case QUALIFIER_QUAD:
                flags |= FLAGS_QUAD;
                break;
#endif

#if defined(QUALIFIER_FIXED_SIZE)
              case QUALIFIER_FIXED_SIZE:
                if (flags & FLAGS_FIXED_SIZE)
                  return TRIO_ERROR_RETURN(TRIO_EINVAL, index);

                if (flags & (FLAGS_ALL_SIZES | FLAGS_LONGDOUBLE |
                             FLAGS_WIDECHAR | FLAGS_VARSIZE_PARAMETER))
                  return TRIO_ERROR_RETURN(TRIO_EINVAL, index);

                if ((format[index] == '6') &&
                    (format[index + 1] == '4'))
                  {
                    varsize = sizeof(trio_int64_t);
                    index += 2;
                  }
                else if ((format[index] == '3') &&
                         (format[index + 1] == '2'))
                  {
                    varsize = sizeof(trio_int32_t);
                    index += 2;
                  }
                else if ((format[index] == '1') &&
                         (format[index + 1] == '6'))
                  {
                    varsize = sizeof(trio_int16_t);
                    index += 2;
                  }
                else if (format[index] == '8')
                  {
                    varsize = sizeof(trio_int8_t);
                    index++;
                  }
                else
                  return TRIO_ERROR_RETURN(TRIO_EINVAL, index);

                flags |= FLAGS_FIXED_SIZE;
                break;
#endif

#if defined(QUALIFIER_WIDECHAR)
              case QUALIFIER_WIDECHAR:
                flags |= FLAGS_WIDECHAR;
                break;
#endif

#if defined(QUALIFIER_SIZE_T_UPPER)
              case QUALIFIER_SIZE_T_UPPER:
                break;
#endif

#if defined(QUALIFIER_QUOTE)
              case QUALIFIER_QUOTE:
                flags |= FLAGS_QUOTE;
                break;
#endif

#if defined(QUALIFIER_STICKY)
              case QUALIFIER_STICKY:
                flags |= FLAGS_STICKY;
                gotSticky = TRUE;
                break;
#endif

#if defined(QUALIFIER_VARSIZE)
              case QUALIFIER_VARSIZE:
                flags |= FLAGS_VARSIZE_PARAMETER;
                parameterPosition++;
                if (positional)
                  varsize = parameterPosition;
                else
                  {
                    varsize = currentParam;
                    currentParam = varsize + 1;
                  }
                if (currentParam > maxParam)
                  maxParam = currentParam;
                break;
#endif
```

```
#if defined(QUALIFIER_ROUNDING_UPPER)
                case QUALIFIER_ROUNDING_UPPER:
                  flags |= FLAGS_ROUNDING;
                  break;
#endif

                default:
                  /* Bail out completely to make the error more obvious */
                  return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                }
            } /* while qualifier */

          /*
           * Parameters only need the type and value. The value is
           * read later.
           */
          if (flags & FLAGS_WIDTH_PARAMETER)
            {
              usedEntries[width] += 1;
              parameters[pos].type = FORMAT_PARAMETER;
              parameters[pos].flags = 0;
              indices[width] = pos;
              width = pos++;
            }
          if (flags & FLAGS_PRECISION_PARAMETER)
            {
              usedEntries[precision] += 1;
              parameters[pos].type = FORMAT_PARAMETER;
              parameters[pos].flags = 0;
              indices[precision] = pos;
              precision = pos++;
            }
          if (flags & FLAGS_BASE_PARAMETER)
            {
              usedEntries[base] += 1;
              parameters[pos].type = FORMAT_PARAMETER;
              parameters[pos].flags = 0;
              indices[base] = pos;
              base = pos++;
            }
          if (flags & FLAGS_VARSIZE_PARAMETER)
            {
              usedEntries[varsize] += 1;
              parameters[pos].type = FORMAT_PARAMETER;
              parameters[pos].flags = 0;
              indices[varsize] = pos;
              varsize = pos++;
            }

          indices[currentParam] = pos;

          switch (format[index++])
            {
#if defined(SPECIFIER_CHAR_UPPER)
            case SPECIFIER_CHAR_UPPER:
              flags |= FLAGS_WIDECHAR;
              /* FALLTHROUGH */
#endif
            case SPECIFIER_CHAR:
              if (flags & FLAGS_LONG)
                flags |= FLAGS_WIDECHAR;
              else if (flags & FLAGS_SHORT)
                flags &= ~FLAGS_WIDECHAR;
              parameters[pos].type = FORMAT_CHAR;
              break;

#if defined(SPECIFIER_STRING_UPPER)
            case SPECIFIER_STRING_UPPER:
              flags |= FLAGS_WIDECHAR;
              /* FALLTHROUGH */
#endif
            case SPECIFIER_STRING:
              if (flags & FLAGS_LONG)
                flags |= FLAGS_WIDECHAR;
              else if (flags & FLAGS_SHORT)
                flags &= ~FLAGS_WIDECHAR;
              parameters[pos].type = FORMAT_STRING;
              break;

            case SPECIFIER_GROUP:
              if (TYPE_SCAN == type)
                {
                  int depth = 1;
                  parameters[pos].type = FORMAT_GROUP;
                  if (format[index] == QUALIFIER_CIRCUMFLEX)
                    index++;
                  if (format[index] == SPECIFIER_UNGROUP)
                    index++;
                  if (format[index] == QUALIFIER_MINUS)
                    index++;
                  /* Skip nested brackets */
                  while (format[index] != NIL)
                    {
                      if (format[index] == SPECIFIER_GROUP)
                        {
                          depth++;
                        }
```

```
                    else if (format[index] == SPECIFIER_UNGROUP)
                      {
                        if (--depth <= 0)
                          {
                            index++;
                            break;
                          }
                      }
                    index++;
                  }
              }
            break;

          case SPECIFIER_INTEGER:
            parameters[pos].type = FORMAT_INT;
            break;

          case SPECIFIER_UNSIGNED:
            flags |= FLAGS_UNSIGNED;
            parameters[pos].type = FORMAT_INT;
            break;

          case SPECIFIER_DECIMAL:
            /* Disable base modifier */
            flags &= ~FLAGS_BASE_PARAMETER;
            base = BASE_DECIMAL;
            parameters[pos].type = FORMAT_INT;
            break;

          case SPECIFIER_OCTAL:
            flags |= FLAGS_UNSIGNED;
            flags &= ~FLAGS_BASE_PARAMETER;
            base = BASE_OCTAL;
            parameters[pos].type = FORMAT_INT;
            break;

#if defined(SPECIFIER_BINARY)
          case SPECIFIER_BINARY_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
          case SPECIFIER_BINARY:
            flags |= FLAGS_NILPADDING;
            flags &= ~FLAGS_BASE_PARAMETER;
            base = BASE_BINARY;
            parameters[pos].type = FORMAT_INT;
            break;
#endif

          case SPECIFIER_HEX_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
          case SPECIFIER_HEX:
            flags |= FLAGS_UNSIGNED;
            flags &= ~FLAGS_BASE_PARAMETER;
            base = BASE_HEX;
            parameters[pos].type = FORMAT_INT;
            break;

          case SPECIFIER_FLOAT_E_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
          case SPECIFIER_FLOAT_E:
            flags |= FLAGS_FLOAT_E;
            parameters[pos].type = FORMAT_DOUBLE;
            break;

          case SPECIFIER_FLOAT_G_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
          case SPECIFIER_FLOAT_G:
            flags |= FLAGS_FLOAT_G;
            parameters[pos].type = FORMAT_DOUBLE;
            break;

          case SPECIFIER_FLOAT_F_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
          case SPECIFIER_FLOAT_F:
            parameters[pos].type = FORMAT_DOUBLE;
            break;

          case SPECIFIER_POINTER:
            if (sizeof(trio_pointer_t) == sizeof(trio_ulonglong_t))
              flags |= FLAGS_QUAD;
            else if (sizeof(trio_pointer_t) == sizeof(long))
              flags |= FLAGS_LONG;
            parameters[pos].type = FORMAT_POINTER;
            break;

          case SPECIFIER_COUNT:
            parameters[pos].type = FORMAT_COUNT;
            break;

#if defined(SPECIFIER_HEXFLOAT)
# if defined(SPECIFIER_HEXFLOAT_UPPER)
          case SPECIFIER_HEXFLOAT_UPPER:
            flags |= FLAGS_UPPER;
            /* FALLTHROUGH */
```

```
# endif
            case SPECIFIER_HEXFLOAT:
              base = BASE_HEX;
              parameters[pos].type = FORMAT_DOUBLE;
              break;
#endif

#if defined(FORMAT_ERRNO)
            case SPECIFIER_ERRNO:
              parameters[pos].type = FORMAT_ERRNO;
              break;
#endif

#if defined(SPECIFIER_USER_DEFINED_BEGIN)
            case SPECIFIER_USER_DEFINED_BEGIN:
              {
                unsigned int max;
                int without_namespace = TRUE;

                parameters[pos].type = FORMAT_USER_DEFINED;
                parameters[pos].user_name[0] = NIL;
                tmpformat = (char *)&format[index];

                while ((ch = format[index]))
                  {
                    index++;
                    if (ch == SPECIFIER_USER_DEFINED_END)
                      {
                        if (without_namespace)
                          {
                            /* We must get the handle first */
                            parameters[pos].type = FORMAT_PARAMETER;
                            parameters[pos].indexAfterSpecifier = index;
                            parameters[pos].flags = FLAGS_USER_DEFINED;
                            /* Adjust parameters for insertion of new one */
                            pos++;
                            usedEntries[currentParam] += 1;
                            parameters[pos].type = FORMAT_USER_DEFINED;
                            currentParam++;
                            indices[currentParam] = pos;
                            if (currentParam > maxParam)
                              maxParam = currentParam;
                          }
                        /* Copy the user data */
                        max = (unsigned int)(&format[index] - tmpformat);
                        if (max > MAX_USER_DATA)
                          max = MAX_USER_DATA;
                        trio_copy_max(parameters[pos].user_data,
                                      max,
                                      tmpformat);
                        break; /* while */
                      }
                    if (ch == SPECIFIER_USER_DEFINED_SEPARATOR)
                      {
                        without_namespace = FALSE;
                        /* Copy the namespace for later looking-up */
                        max = (int)(&format[index] - tmpformat);
                        if (max > MAX_USER_NAME)
                          max = MAX_USER_NAME;
                        trio_copy_max(parameters[pos].user_name,
                                      max,
                                      tmpformat);
                        tmpformat = (char *)&format[index];
                      }
                  }
                if (ch != SPECIFIER_USER_DEFINED_END)
                  return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
              }
              break;
#endif /* defined(SPECIFIER_USER_DEFINED_BEGIN) */

            default:
              /* Bail out completely to make the error more obvious */
              return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
            }

          /*  Count the number of times this entry has been used */
          usedEntries[currentParam] += 1;

          /* Find last sticky parameters */
          if (gotSticky && !(flags & FLAGS_STICKY))
            {
              for (i = pos - 1; i >= 0; i--)
                {
                  if (parameters[i].type == FORMAT_PARAMETER)
                    continue;
                  if ((parameters[i].flags & FLAGS_STICKY) &&
                      (parameters[i].type == parameters[pos].type))
                    {
                      /* Do not overwrite current qualifiers */
                      flags |= (parameters[i].flags & (unsigned long)~FLAGS_STICKY);
                      if (width == NO_WIDTH)
                        width = parameters[i].width;
                      if (precision == NO_PRECISION)
                        precision = parameters[i].precision;
                      if (base == NO_BASE)
                        base = parameters[i].base;
                      break;
```

```
                    }
                }
              }

            parameters[pos].indexAfterSpecifier = index;
            parameters[pos].flags = flags;
            parameters[pos].width = width;
            parameters[pos].precision = precision;
            parameters[pos].base = (base == NO_BASE) ? BASE_DECIMAL : base;
            parameters[pos].varsize = varsize;
            pos++;

            if (! positional)
              parameterPosition++;

          } /* if identifier */

      } /* while format characters left */

    for (num = 0; num <= maxParam; num++)
      {
        if (usedEntries[num] != 1)
          {
            if (usedEntries[num] == 0) /* gap detected */
              return TRIO_ERROR_RETURN(TRIO_EGAP, num);
            else /* double references detected */
              return TRIO_ERROR_RETURN(TRIO_EDBLREF, num);
          }

        i = indices[num];

        /*
         * FORMAT_PARAMETERS are only present if they must be read,
         * so it makes no sense to check the ignore flag (besides,
         * the flags variable is not set for that particular type)
         */
        if ((parameters[i].type != FORMAT_PARAMETER) &&
            (parameters[i].flags & FLAGS_IGNORE))
          continue; /* for all arguments */

        /*
         * The stack arguments are read according to ANSI C89
         * default argument promotions:
         *
         *  char           = int
         *  short          = int
         *  unsigned char  = unsigned int
         *  unsigned short = unsigned int
         *  float          = double
         *
         * In addition to the ANSI C89 these types are read (the
         * default argument promotions of C99 has not been
         * considered yet)
         *
         *  long long
         *  long double
         *  size_t
         *  ptrdiff_t
         *  intmax_t
         */
        switch (parameters[i].type)
          {
          case FORMAT_GROUP:
          case FORMAT_STRING:
#if TRIO_WIDECHAR
            if (flags & FLAGS_WIDECHAR)
              {
                parameters[i].data.wstring = (argarray == NULL)
                  ? va_arg(TRIO_VA_LIST_DEREF(arglist), trio_wchar_t *)
                  : (trio_wchar_t *)(argarray[num]);
              }
            else
#endif
              {
                parameters[i].data.string = (argarray == NULL)
                  ? va_arg(TRIO_VA_LIST_DEREF(arglist), char *)
                  : (char *)(argarray[num]);
              }
            break;

#if defined(FORMAT_USER_DEFINED)
          case FORMAT_USER_DEFINED:
#endif
          case FORMAT_POINTER:
          case FORMAT_COUNT:
          case FORMAT_UNKNOWN:
            parameters[i].data.pointer = (argarray == NULL)
              ? va_arg(TRIO_VA_LIST_DEREF(arglist), trio_pointer_t )
              : argarray[num];
            break;

          case FORMAT_CHAR:
          case FORMAT_INT:
            if (TYPE_SCAN == type)
              {
                if (argarray == NULL)
                  parameters[i].data.pointer =
                    (trio_pointer_t)va_arg(TRIO_VA_LIST_DEREF(arglist), trio_pointer_t);
```

```
              else
                {
                  if (parameters[i].type == FORMAT_CHAR)
                    parameters[i].data.pointer =
                      (trio_pointer_t)((char *)argarray[num]);
                  else if (parameters[i].flags & FLAGS_SHORT)
                    parameters[i].data.pointer =
                      (trio_pointer_t)((short *)argarray[num]);
                  else
                    parameters[i].data.pointer =
                      (trio_pointer_t)((int *)argarray[num]);
                }
            }
          else
            {
#if defined(QUALIFIER_VARSIZE) || defined(QUALIFIER_FIXED_SIZE)
              if (parameters[i].flags
                  & (FLAGS_VARSIZE_PARAMETER | FLAGS_FIXED_SIZE))
                {
                  if (parameters[i].flags & FLAGS_VARSIZE_PARAMETER)
                    {
                      /*
                       * Variable sizes are mapped onto the fixed sizes, in
                       * accordance with integer promotion.
                       *
                       * Please note that this may not be portable, as we
                       * only guess the size, not the layout of the numbers.
                       * For example, if int is little-endian, and long is
                       * big-endian, then this will fail.
                       */
                      varsize = (int)parameters[parameters[i].varsize].data.number.as_unsigned;
                    }
                  else
                    {
                      /* Used for the I<bits> modifiers */
                      varsize = parameters[i].varsize;
                    }
                  parameters[i].flags &= ~FLAGS_ALL_VARSIZES;

                  if (varsize <= (int)sizeof(int))
                    ;
                  else if (varsize <= (int)sizeof(long))
                    parameters[i].flags |= FLAGS_LONG;
#if defined(QUALIFIER_INTMAX_T)
                  else if (varsize <= (int)sizeof(trio_longlong_t))
                    parameters[i].flags |= FLAGS_QUAD;
                  else
                    parameters[i].flags |= FLAGS_INTMAX_T;
#else
                  else
                    parameters[i].flags |= FLAGS_QUAD;
#endif
                }
#endif /* defined(QUALIFIER_VARSIZE) */
#if defined(QUALIFIER_SIZE_T) || defined(QUALIFIER_SIZE_T_UPPER)
              if (parameters[i].flags & FLAGS_SIZE_T)
                parameters[i].data.number.as_unsigned = (argarray == NULL)
                  ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), size_t)
                  : (trio_uintmax_t)(*((size_t *)argarray[num]));
              else
#endif
#if defined(QUALIFIER_PTRDIFF_T)
              if (parameters[i].flags & FLAGS_PTRDIFF_T)
                parameters[i].data.number.as_unsigned = (argarray == NULL)
                  ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), ptrdiff_t)
                  : (trio_uintmax_t)(*((ptrdiff_t *)argarray[num]));
              else
#endif
#if defined(QUALIFIER_INTMAX_T)
              if (parameters[i].flags & FLAGS_INTMAX_T)
                parameters[i].data.number.as_unsigned = (argarray == NULL)
                  ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), trio_intmax_t)
                  : (trio_uintmax_t)(*((trio_intmax_t *)argarray[num]));
              else
#endif
              if (parameters[i].flags & FLAGS_QUAD)
                parameters[i].data.number.as_unsigned = (argarray == NULL)
                  ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), trio_ulonglong_t)
                  : (trio_uintmax_t)(*((trio_ulonglong_t *)argarray[num]));
              else if (parameters[i].flags & FLAGS_LONG)
                parameters[i].data.number.as_unsigned = (argarray == NULL)
                  ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), long)
                  : (trio_uintmax_t)(*((long *)argarray[num]));
              else
                {
                  if (argarray == NULL)
                    parameters[i].data.number.as_unsigned = (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), int);
                  else
                    {
                      if (parameters[i].type == FORMAT_CHAR)
                        parameters[i].data.number.as_unsigned = (trio_uintmax_t)(*((char *)argarray[num]));
                      else if (parameters[i].flags & FLAGS_SHORT)
                        parameters[i].data.number.as_unsigned = (trio_uintmax_t)(*((short *)argarray[num]));
                      else
                        parameters[i].data.number.as_unsigned = (trio_uintmax_t)(*((int *)argarray[num]));
                    }
                }
            }
```

```
          break;

        case FORMAT_PARAMETER:
          /*
           * The parameter for the user-defined specifier is a pointer,
           * whereas the rest (width, precision, base) uses an integer.
           */
          if (parameters[i].flags & FLAGS_USER_DEFINED)
            parameters[i].data.pointer = (argarray == NULL)
              ? va_arg(TRIO_VA_LIST_DEREF(arglist), trio_pointer_t )
              : argarray[num];
          else
            parameters[i].data.number.as_unsigned = (argarray == NULL)
              ? (trio_uintmax_t)va_arg(TRIO_VA_LIST_DEREF(arglist), int)
              : (trio_uintmax_t)(*((int *)argarray[num]));
          break;

        case FORMAT_DOUBLE:
          if (TYPE_SCAN == type)
            {
              if (parameters[i].flags & FLAGS_LONGDOUBLE)
                parameters[i].data.longdoublePointer = (argarray == NULL)
                  ? va_arg(TRIO_VA_LIST_DEREF(arglist), trio_long_double_t *)
                  : (trio_long_double_t *)argarray[num];
              else
                {
                  if (parameters[i].flags & FLAGS_LONG)
                    parameters[i].data.doublePointer = (argarray == NULL)
                      ? va_arg(TRIO_VA_LIST_DEREF(arglist), double *)
                      : (double *)argarray[num];
                  else
                    parameters[i].data.doublePointer = (argarray == NULL)
                      ? (double *)va_arg(TRIO_VA_LIST_DEREF(arglist), float *)
                      : (double *)((float *)argarray[num]);
                }
            }
          else
            {
              if (parameters[i].flags & FLAGS_LONGDOUBLE)
                parameters[i].data.longdoubleNumber = (argarray == NULL)
                  ? va_arg(TRIO_VA_LIST_DEREF(arglist), trio_long_double_t)
                  : (trio_long_double_t)(*((trio_long_double_t *)argarray[num]));
              else
                {
                  if (argarray == NULL)
                    parameters[i].data.longdoubleNumber =
                      (trio_long_double_t)va_arg(TRIO_VA_LIST_DEREF(arglist), double);
                  else
                    {
                      if (parameters[i].flags & FLAGS_SHORT)
                        parameters[i].data.longdoubleNumber =
                          (trio_long_double_t)(*((float *)argarray[num]));
                      else
                        parameters[i].data.longdoubleNumber =
                          (trio_long_double_t)(*((double *)argarray[num]));
                    }
                }
            }
          break;

#if defined(FORMAT_ERRNO)
        case FORMAT_ERRNO:
          parameters[i].data.errorNumber = save_errno;
          break;
#endif

        default:
          break;
        }
    } /* for all specifiers */
  return num;
}


/*************************************************************************
 *
 * FORMATTING
 *
 ************************************************************************/


/*************************************************************************
 * TrioWriteNumber
 *
 * Description:
 *  Output a number.
 *  The complexity of this function is a result of the complexity
 *  of the dependencies of the flags.
 */
TRIO_PRIVATE void
TrioWriteNumber
TRIO_ARGS6((self, number, flags, width, precision, base),
           trio_class_t *self,
           trio_uintmax_t number,
           trio_flags_t flags,
           int width,
           int precision,
           int base)
```

```
{
  BOOLEAN_T isNegative;
  BOOLEAN_T isNumberZero;
  BOOLEAN_T isPrecisionZero;
  BOOLEAN_T ignoreNumber;
  char buffer[MAX_CHARS_IN(trio_uintmax_t) * (1 + MAX_LOCALE_SEPARATOR_LENGTH) + 1];
  char *bufferend;
  char *pointer;
  TRIO_CONST char *digits;
  int i;
  int length;
  char *p;
  int count;

  assert(VALID(self));
  assert(VALID(self->OutStream));
  assert(((base >= MIN_BASE) && (base <= MAX_BASE)) || (base == NO_BASE));

  digits = (flags & FLAGS_UPPER) ? internalDigitsUpper : internalDigitsLower;
  if (base == NO_BASE)
    base = BASE_DECIMAL;

  isNumberZero = (number == 0);
  isPrecisionZero = (precision == 0);
  ignoreNumber = (isNumberZero
                  && isPrecisionZero
                  && !((flags & FLAGS_ALTERNATIVE) && (base == BASE_OCTAL)));

  if (flags & FLAGS_UNSIGNED)
    {
      isNegative = FALSE;
      flags &= ~FLAGS_SHOWSIGN;
    }
  else
    {
      isNegative = ((trio_intmax_t)number < 0);
      if (isNegative)
        number = -((trio_intmax_t)number);
    }

  if (flags & FLAGS_QUAD)
    number &= (trio_ulonglong_t)-1;
  else if (flags & FLAGS_LONG)
    number &= (unsigned long)-1;
  else
    number &= (unsigned int)-1;

  /* Build number */
  pointer = bufferend = &buffer[sizeof(buffer) - 1];
  *pointer-- = NIL;
  for (i = 1; i < (int)sizeof(buffer); i++)
    {
      *pointer-- = digits[number % base];
      number /= base;
      if (number == 0)
        break;

      if ((flags & FLAGS_QUOTE) && TrioFollowedBySeparator(i + 1))
        {
          /*
           * We are building the number from the least significant
           * to the most significant digit, so we have to copy the
           * thousand separator backwards
           */
          length = internalThousandSeparatorLength;
          if (((int)(pointer - buffer) - length) > 0)
            {
              p = &internalThousandSeparator[length - 1];
              while (length-- > 0)
                *pointer-- = *p--;
            }
        }
    }

  if (! ignoreNumber)
    {
      /* Adjust width */
      width -= (bufferend - pointer) - 1;
    }

  /* Adjust precision */
  if (NO_PRECISION != precision)
    {
      precision -= (bufferend - pointer) - 1;
      if (precision < 0)
        precision = 0;
      flags |= FLAGS_NILPADDING;
    }

  /* Calculate padding */
  count = (! ((flags & FLAGS_LEFTADJUST) || (precision == NO_PRECISION)))
    ? precision
    : 0;

  /* Adjust width further */
  if (isNegative || (flags & FLAGS_SHOWSIGN) || (flags & FLAGS_SPACE))
    width--;
  if ((flags & FLAGS_ALTERNATIVE) && !isNumberZero)
```

```
    {
      switch (base)
        {
        case BASE_BINARY:
        case BASE_HEX:
          width -= 2;
          break;
        case BASE_OCTAL:
          if (!(flags & FLAGS_NILPADDING) || (count == 0))
            width--;
          break;
        default:
          break;
        }
    }

  /* Output prefixes spaces if needed */
  if (! ((flags & FLAGS_LEFTADJUST) ||
         ((flags & FLAGS_NILPADDING) && (precision == NO_PRECISION))))
    {
      while (width-- > count)
        self->OutStream(self, CHAR_ADJUST);
    }

  /* width has been adjusted for signs and alternatives */
  if (isNegative)
    self->OutStream(self, '-');
  else if (flags & FLAGS_SHOWSIGN)
    self->OutStream(self, '+');
  else if (flags & FLAGS_SPACE)
    self->OutStream(self, ' ');

  /* Prefix is not written when the value is zero */
  if ((flags & FLAGS_ALTERNATIVE) && !isNumberZero)
    {
      switch (base)
        {
        case BASE_BINARY:
          self->OutStream(self, '0');
          self->OutStream(self, (flags & FLAGS_UPPER) ? 'B' : 'b');
          break;

        case BASE_OCTAL:
          if (!(flags & FLAGS_NILPADDING) || (count == 0))
            self->OutStream(self, '0');
          break;

        case BASE_HEX:
          self->OutStream(self, '0');
          self->OutStream(self, (flags & FLAGS_UPPER) ? 'X' : 'x');
          break;

        default:
          break;
        } /* switch base */
    }

  /* Output prefixed zero padding if needed */
  if (flags & FLAGS_NILPADDING)
    {
      if (precision == NO_PRECISION)
        precision = width;
      while (precision-- > 0)
        {
          self->OutStream(self, '0');
          width--;
        }
    }

  if (! ignoreNumber)
    {
      /* Output the number itself */
      while (*(++pointer))
        {
          self->OutStream(self, *pointer);
        }
    }

  /* Output trailing spaces if needed */
  if (flags & FLAGS_LEFTADJUST)
    {
      while (width-- > 0)
        self->OutStream(self, CHAR_ADJUST);
    }
}

/*************************************************************************
 * TrioWriteStringCharacter
 *
 * Description:
 *  Output a single character of a string
 */
TRIO_PRIVATE void
TrioWriteStringCharacter
TRIO_ARGS3((self, ch, flags),
           trio_class_t *self,
           int ch,
           trio_flags_t flags)
```

```
{
  if (flags & FLAGS_ALTERNATIVE)
    {
      if (! isprint(ch))
        {
          /*
           * Non-printable characters are converted to C escapes or
           * \number, if no C escape exists.
           */
          self->OutStream(self, CHAR_BACKSLASH);
          switch (ch)
            {
            case '\007': self->OutStream(self, 'a'); break;
            case '\b': self->OutStream(self, 'b'); break;
            case '\f': self->OutStream(self, 'f'); break;
            case '\n': self->OutStream(self, 'n'); break;
            case '\r': self->OutStream(self, 'r'); break;
            case '\t': self->OutStream(self, 't'); break;
            case '\v': self->OutStream(self, 'v'); break;
            case '\\': self->OutStream(self, '\\'); break;
            default:
              self->OutStream(self, 'x');
              TrioWriteNumber(self, (trio_uintmax_t)ch,
                              FLAGS_UNSIGNED | FLAGS_NILPADDING,
                              2, 2, BASE_HEX);
              break;
            }
        }
      else if (ch == CHAR_BACKSLASH)
        {
          self->OutStream(self, CHAR_BACKSLASH);
          self->OutStream(self, CHAR_BACKSLASH);
        }
      else
        {
          self->OutStream(self, ch);
        }
    }
  else
    {
      self->OutStream(self, ch);
    }
}

/*************************************************************************
 * TrioWriteString
 *
 * Description:
 *  Output a string
 */
TRIO_PRIVATE void
TrioWriteString
TRIO_ARGS5((self, string, flags, width, precision),
           trio_class_t *self,
           TRIO_CONST char *string,
           trio_flags_t flags,
           int width,
           int precision)
{
  int length;
  int ch;

  assert(VALID(self));
  assert(VALID(self->OutStream));

  if (string == NULL)
    {
      string = internalNullString;
      length = sizeof(internalNullString) - 1;
      /* Disable quoting for the null pointer */
      flags &= (~FLAGS_QUOTE);
      width = 0;
    }
  else
    {
      length = trio_length(string);
    }
  if ((NO_PRECISION != precision) &&
      (precision < length))
    {
      length = precision;
    }
  width -= length;

  if (flags & FLAGS_QUOTE)
    self->OutStream(self, CHAR_QUOTE);

  if (! (flags & FLAGS_LEFTADJUST))
    {
      while (width-- > 0)
        self->OutStream(self, CHAR_ADJUST);
    }

  while (length-- > 0)
    {
      /* The ctype parameters must be an unsigned char (or EOF) */
      ch = (int)((unsigned char)(*string++));
      TrioWriteStringCharacter(self, ch, flags);
```

```
    }

  if (flags & FLAGS_LEFTADJUST)
    {
      while (width-- > 0)
        self->OutStream(self, CHAR_ADJUST);
    }
  if (flags & FLAGS_QUOTE)
    self->OutStream(self, CHAR_QUOTE);
}

/*************************************************************************
 * TrioWriteWideStringCharacter
 *
 * Description:
 *  Output a wide string as a multi-byte sequence
 */
#if TRIO_WIDECHAR
TRIO_PRIVATE int
TrioWriteWideStringCharacter
TRIO_ARGS4((self, wch, flags, width),
           trio_class_t *self,
           trio_wchar_t wch,
           trio_flags_t flags,
           int width)
{
  int size;
  int i;
  int ch;
  char *string;
  char buffer[MB_LEN_MAX + 1];

  if (width == NO_WIDTH)
    width = sizeof(buffer);

  size = wctomb(buffer, wch);
  if ((size <= 0) || (size > width) || (buffer[0] == NIL))
    return 0;

  string = buffer;
  i = size;
  while ((width >= i) && (width-- > 0) && (i-- > 0))
    {
      /* The ctype parameters must be an unsigned char (or EOF) */
      ch = (int)((unsigned char)(*string++));
      TrioWriteStringCharacter(self, ch, flags);
    }
  return size;
}
#endif /* TRIO_WIDECHAR */

/*************************************************************************
 * TrioWriteWideString
 *
 * Description:
 *  Output a wide character string as a multi-byte string
 */
#if TRIO_WIDECHAR
TRIO_PRIVATE void
TrioWriteWideString
TRIO_ARGS5((self, wstring, flags, width, precision),
           trio_class_t *self,
           TRIO_CONST trio_wchar_t *wstring,
           trio_flags_t flags,
           int width,
           int precision)
{
  int length;
  int size;

  assert(VALID(self));
  assert(VALID(self->OutStream));

#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  (void)mblen(NULL, 0);
#endif

  if (wstring == NULL)
    {
      TrioWriteString(self, NULL, flags, width, precision);
      return;
    }

  if (NO_PRECISION == precision)
    {
      length = INT_MAX;
    }
  else
    {
      length = precision;
      width -= length;
    }

  if (flags & FLAGS_QUOTE)
    self->OutStream(self, CHAR_QUOTE);

  if (! (flags & FLAGS_LEFTADJUST))
    {
```

```
      while (width-- > 0)
        self->OutStream(self, CHAR_ADJUST);
    }

  while (length > 0)
    {
      size = TrioWriteWideStringCharacter(self, *wstring++, flags, length);
      if (size == 0)
        break; /* while */
      length -= size;
    }

  if (flags & FLAGS_LEFTADJUST)
    {
      while (width-- > 0)
        self->OutStream(self, CHAR_ADJUST);
    }
  if (flags & FLAGS_QUOTE)
    self->OutStream(self, CHAR_QUOTE);
}
#endif /* TRIO_WIDECHAR */

/*************************************************************************
 * TrioWriteDouble
 *
 * http://wwwold.dkuug.dk/JTC1/SC22/WG14/www/docs/dr_211.htm
 *
 * "5.2.4.2.2 paragraph #4
 *
 *  The accuracy [...] is implementation defined, as is the accuracy
 *  of the conversion between floating-point internal representations
 *  and string representations performed by the libray routine in
 *  <stdio.h>"
 */
/* FIXME: handle all instances of constant long-double number (L)
 *   and *l() math functions.
 */
TRIO_PRIVATE void
TrioWriteDouble
TRIO_ARGS6((self, number, flags, width, precision, base),
           trio_class_t *self,
           trio_long_double_t number,
           trio_flags_t flags,
           int width,
           int precision,
           int base)
{
  trio_long_double_t integerNumber;
  trio_long_double_t fractionNumber;
  trio_long_double_t workNumber;
  int integerDigits;
  int fractionDigits;
  int exponentDigits;
  int baseDigits;
  int integerThreshold;
  int fractionThreshold;
  int expectedWidth;
  int exponent = 0;
  unsigned int uExponent = 0;
  int exponentBase;
  trio_long_double_t dblBase;
  trio_long_double_t dblIntegerBase;
  trio_long_double_t dblFractionBase;
  trio_long_double_t integerAdjust;
  trio_long_double_t fractionAdjust;
  BOOLEAN_T isNegative;
  BOOLEAN_T isExponentNegative = FALSE;
  BOOLEAN_T requireTwoDigitExponent;
  BOOLEAN_T isHex;
  TRIO_CONST char *digits;
  char *groupingPointer;
  int i;
  int index;
  BOOLEAN_T hasOnlyZeroes;
  int zeroes = 0;
  register int trailingZeroes;
  BOOLEAN_T keepTrailingZeroes;
  BOOLEAN_T keepDecimalPoint;
  trio_long_double_t epsilon;

  assert(VALID(self));
  assert(VALID(self->OutStream));
  assert(((base >= MIN_BASE) && (base <= MAX_BASE)) || (base == NO_BASE));

  /* Determine sign and look for special quantities */
  switch (trio_fpclassify_and_signbit(number, &isNegative))
    {
    case TRIO_FP_NAN:
      TrioWriteString(self,
                      (flags & FLAGS_UPPER)
                      ? NAN_UPPER
                      : NAN_LOWER,
                      flags, width, precision);
      return;

    case TRIO_FP_INFINITE:
      if (isNegative)
        {
```

```
        /* Negative infinity */
        TrioWriteString(self,
                        (flags & FLAGS_UPPER)
                        ? "-" INFINITE_UPPER
                        : "-" INFINITE_LOWER,
                        flags, width, precision);
        return;
      }
    else
      {
        /* Positive infinity */
        TrioWriteString(self,
                        (flags & FLAGS_UPPER)
                        ? INFINITE_UPPER
                        : INFINITE_LOWER,
                        flags, width, precision);
        return;
      }

  default:
    /* Finitude */
    break;
  }

/* Normal numbers */
if (flags & FLAGS_LONGDOUBLE)
  {
    baseDigits = (base == 10)
      ? LDBL_DIG
      : (int)floor(LDBL_MANT_DIG / TrioLogarithmBase(base));
    epsilon = LDBL_EPSILON;
  }
else if (flags & FLAGS_SHORT)
  {
    baseDigits = (base == BASE_DECIMAL)
      ? FLT_DIG
      : (int)floor(FLT_MANT_DIG / TrioLogarithmBase(base));
    epsilon = FLT_EPSILON;
  }
else
  {
    baseDigits = (base == BASE_DECIMAL)
      ? DBL_DIG
      : (int)floor(DBL_MANT_DIG / TrioLogarithmBase(base));
    epsilon = DBL_EPSILON;
  }

digits = (flags & FLAGS_UPPER) ? internalDigitsUpper : internalDigitsLower;
isHex = (base == BASE_HEX);
if (base == NO_BASE)
  base = BASE_DECIMAL;
dblBase = (trio_long_double_t)base;
keepTrailingZeroes = !( (flags & FLAGS_ROUNDING) ||
                        ( (flags & FLAGS_FLOAT_G) &&
                          !(flags & FLAGS_ALTERNATIVE) ) );

if (flags & FLAGS_ROUNDING)
  precision = baseDigits;

if (precision == NO_PRECISION)
  {
    if (isHex)
      {
        keepTrailingZeroes = FALSE;
        precision = FLT_MANT_DIG;
      }
    else
      {
        precision = FLT_DIG;
      }
  }

if (isNegative)
  number = -number;

if (isHex)
  flags |= FLAGS_FLOAT_E;

if (flags & FLAGS_FLOAT_G)
  {
    if (precision == 0)
      precision = 1;

    if ((number < 1.0E-4) || (number > powl(base,
                                            (trio_long_double_t)precision)))
      {
        /* Use scientific notation */
        flags |= FLAGS_FLOAT_E;
      }
    else if (number < 1.0)
      {
        /*
         * Use normal notation. If the integer part of the number is
         * zero, then adjust the precision to include leading fractional
         * zeros.
         */
        workNumber = TrioLogarithm(number, base);
        workNumber = TRIO_FABS(workNumber);
```

```
        if (workNumber - floorl(workNumber) < 0.001)
          workNumber--;
        zeroes = (int)floorl(workNumber);
      }
  }

if (flags & FLAGS_FLOAT_E)
  {
    /* Scale the number */
    workNumber = TrioLogarithm(number, base);
    if (trio_isinf(workNumber) == -1)
      {
        exponent = 0;
        /* Undo setting */
        if (flags & FLAGS_FLOAT_G)
          flags &= ~FLAGS_FLOAT_E;
      }
    else
      {
        exponent = (int)floorl(workNumber);
        number /= powl(dblBase, (trio_long_double_t)exponent);
        isExponentNegative = (exponent < 0);
        uExponent = (isExponentNegative) ? -exponent : exponent;
        if (isHex)
          uExponent *= 4; /* log16(2) */
        /* No thousand separators */
        flags &= ~FLAGS_QUOTE;
      }
  }

integerNumber = floorl(number);
fractionNumber = number - integerNumber;

/*
 * Truncated number.
 *
 * Precision is number of significant digits for FLOAT_G
 * and number of fractional digits for others.
 */
integerDigits = (integerNumber > epsilon)
  ? 1 + (int)TrioLogarithm(integerNumber, base)
  : 1;
fractionDigits = ((flags & FLAGS_FLOAT_G) && (zeroes == 0))
  ? precision - integerDigits
  : zeroes + precision;

dblFractionBase = TrioPower(base, fractionDigits);

workNumber = number + 0.5 / dblFractionBase;
if (floorl(number) != floorl(workNumber))
  {
    if (flags & FLAGS_FLOAT_E)
      {
        /* Adjust if number was rounded up one digit (ie. 0.99 to 1.00) */
        exponent++;
        isExponentNegative = (exponent < 0);
        uExponent = (isExponentNegative) ? -exponent : exponent;
        if (isHex)
          uExponent *= 4; /* log16(2) */
        workNumber = (number + 0.5 / dblFractionBase) / dblBase;
        integerNumber = floorl(workNumber);
        fractionNumber = workNumber - integerNumber;
      }
    else
      {
        /* Adjust if number was rounded up one digit (ie. 99 to 100) */
        integerNumber = floorl(number + 0.5);
        fractionNumber = 0.0;
        integerDigits = (integerNumber > epsilon)
          ? 1 + (int)TrioLogarithm(integerNumber, base)
          : 1;
      }
  }

/* Estimate accuracy */
integerAdjust = fractionAdjust = 0.5;
if (flags & FLAGS_ROUNDING)
  {
    if (integerDigits > baseDigits)
      {
        integerThreshold = baseDigits;
        fractionDigits = 0;
        dblFractionBase = 1.0;
        fractionThreshold = 0;
        precision = 0; /* Disable decimal-point */
        integerAdjust = TrioPower(base, integerDigits - integerThreshold - 1);
        fractionAdjust = 0.0;
      }
    else
      {
        integerThreshold = integerDigits;
        fractionThreshold = fractionDigits - integerThreshold;
        fractionAdjust = 1.0;
      }
  }
else
  {
    integerThreshold = INT_MAX;
```

```
    fractionThreshold = INT_MAX;
  }

/*
 * Calculate expected width.
 *  sign + integer part + thousands separators + decimal point
 *  + fraction + exponent
 */
fractionAdjust /= dblFractionBase;
hasOnlyZeroes = (floorl((fractionNumber + fractionAdjust) * dblFractionBase) < epsilon);
keepDecimalPoint = ( (flags & FLAGS_ALTERNATIVE) ||
                     !((precision == 0) ||
                       (!keepTrailingZeroes && hasOnlyZeroes)) );
if (flags & FLAGS_FLOAT_E)
  {
    exponentDigits = (uExponent == 0)
      ? 1
      : (int)ceil(TrioLogarithm((double)(uExponent + 1),
                                (isHex) ? 10.0 : base));
  }
else
  exponentDigits = 0;
requireTwoDigitExponent = ((base == BASE_DECIMAL) && (exponentDigits == 1));

expectedWidth = integerDigits + fractionDigits
  + (keepDecimalPoint
     ? internalDecimalPointLength
     : 0)
  + ((flags & FLAGS_QUOTE)
     ? TrioCalcThousandSeparatorLength(integerDigits)
     : 0);
if (isNegative || (flags & FLAGS_SHOWSIGN) || (flags & FLAGS_SPACE))
  expectedWidth += sizeof("-") - 1;
if (exponentDigits > 0)
  expectedWidth += exponentDigits +
    ((requireTwoDigitExponent ? sizeof("E+0") : sizeof("E+")) - 1);
if (isHex)
  expectedWidth += sizeof("0X") - 1;

/* Output prefixing */
if (flags & FLAGS_NILPADDING)
  {
    /* Leading zeros must be after sign */
    if (isNegative)
      self->OutStream(self, '-');
    else if (flags & FLAGS_SHOWSIGN)
      self->OutStream(self, '+');
    else if (flags & FLAGS_SPACE)
      self->OutStream(self, ' ');
    if (isHex)
      {
        self->OutStream(self, '0');
        self->OutStream(self, (flags & FLAGS_UPPER) ? 'X' : 'x');
      }
    if (!(flags & FLAGS_LEFTADJUST))
      {
        for (i = expectedWidth; i < width; i++)
          {
            self->OutStream(self, '0');
          }
      }
  }
else
  {
    /* Leading spaces must be before sign */
    if (!(flags & FLAGS_LEFTADJUST))
      {
        for (i = expectedWidth; i < width; i++)
          {
            self->OutStream(self, CHAR_ADJUST);
          }
      }
    if (isNegative)
      self->OutStream(self, '-');
    else if (flags & FLAGS_SHOWSIGN)
      self->OutStream(self, '+');
    else if (flags & FLAGS_SPACE)
      self->OutStream(self, ' ');
    if (isHex)
      {
        self->OutStream(self, '0');
        self->OutStream(self, (flags & FLAGS_UPPER) ? 'X' : 'x');
      }
  }

/* Output the integer part and thousand separators */
dblIntegerBase = 1.0 / TrioPower(base, integerDigits - 1);
for (i = 0; i < integerDigits; i++)
  {
    workNumber = floorl(((integerNumber + integerAdjust) * dblIntegerBase));
    if (i > integerThreshold)
      {
        /* Beyond accuracy */
        self->OutStream(self, digits[0]);
      }
    else
      {
        self->OutStream(self, digits[(int)fmodl(workNumber, dblBase)]);
```

```
      }
    dblIntegerBase *= dblBase;

    if (((flags & (FLAGS_FLOAT_E | FLAGS_QUOTE)) == FLAGS_QUOTE)
        && TrioFollowedBySeparator(integerDigits - i))
      {
        for (groupingPointer = internalThousandSeparator;
             *groupingPointer != NIL;
             groupingPointer++)
          {
            self->OutStream(self, *groupingPointer);
          }
      }
  }

/* Insert decimal point and build the fraction part */
trailingZeroes = 0;

if (keepDecimalPoint)
  {
    if (internalDecimalPoint)
      {
        self->OutStream(self, internalDecimalPoint);
      }
    else
      {
        for (i = 0; i < internalDecimalPointLength; i++)
          {
            self->OutStream(self, internalDecimalPointString[i]);
          }
      }
  }

for (i = 0; i < fractionDigits; i++)
  {
    if ((integerDigits > integerThreshold) || (i > fractionThreshold))
      {
        /* Beyond accuracy */
        trailingZeroes++;
      }
    else
      {
        fractionNumber *= dblBase;
        fractionAdjust *= dblBase;
        workNumber = floorl(fractionNumber + fractionAdjust);
        fractionNumber -= workNumber;
        index = (int)fmodl(workNumber, dblBase);
        if (index == 0)
          {
            trailingZeroes++;
          }
        else
          {
            while (trailingZeroes > 0)
              {
                /* Not trailing zeroes after all */
                self->OutStream(self, digits[0]);
                trailingZeroes--;
              }
            self->OutStream(self, digits[index]);
          }
      }
  }

if (keepTrailingZeroes)
  {
    while (trailingZeroes > 0)
      {
        self->OutStream(self, digits[0]);
        trailingZeroes--;
      }
  }

/* Output exponent */
if (exponentDigits > 0)
  {
    self->OutStream(self,
                    isHex
                    ? ((flags & FLAGS_UPPER) ? 'P' : 'p')
                    : ((flags & FLAGS_UPPER) ? 'E' : 'e'));
    self->OutStream(self, (isExponentNegative) ? '-' : '+');

    /* The exponent must contain at least two digits */
    if (requireTwoDigitExponent)
      self->OutStream(self, '0');

    if (isHex)
      base = 10.0;
    exponentBase = (int)TrioPower(base, exponentDigits - 1);
    for (i = 0; i < exponentDigits; i++)
      {
        self->OutStream(self, digits[(uExponent / exponentBase) % base]);
        exponentBase /= base;
      }
  }
/* Output trailing spaces */
if (flags & FLAGS_LEFTADJUST)
  {
```

```
      for (i = expectedWidth; i < width; i++)
        {
          self->OutStream(self, CHAR_ADJUST);
        }
    }
}

/*************************************************************************
 * TrioFormatProcess
 *
 * Description:
 *  This is the main engine for formatting output
 */
TRIO_PRIVATE int
TrioFormatProcess
TRIO_ARGS3((data, format, parameters),
           trio_class_t *data,
           TRIO_CONST char *format,
           trio_parameter_t *parameters)
{
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  int charlen;
#endif
  int i;
  TRIO_CONST char *string;
  trio_pointer_t pointer;
  trio_flags_t flags;
  int width;
  int precision;
  int base;
  int index;

  index = 0;
  i = 0;
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  (void)mblen(NULL, 0);
#endif

  while (format[index])
    {
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
      if (! isascii(format[index]))
        {
          charlen = mblen(&format[index], MB_LEN_MAX);
          /*
           * Only valid multibyte characters are handled here. Invalid
           * multibyte characters (charlen == -1) are handled as normal
           * characters.
           */
          if (charlen != -1)
            {
              while (charlen-- > 0)
                {
                  data->OutStream(data, format[index++]);
                }
              continue; /* while characters left in formatting string */
            }
        }
#endif /* TRIO_COMPILER_SUPPORTS_MULTIBYTE */
      if (CHAR_IDENTIFIER == format[index])
        {
          if (CHAR_IDENTIFIER == format[index + 1])
            {
              data->OutStream(data, CHAR_IDENTIFIER);
              index += 2;
            }
          else
            {
              /* Skip the parameter entries */
              while (parameters[i].type == FORMAT_PARAMETER)
                i++;

              flags = parameters[i].flags;

              /* Find width */
              width = parameters[i].width;
              if (flags & FLAGS_WIDTH_PARAMETER)
                {
                  /* Get width from parameter list */
                  width = (int)parameters[width].data.number.as_signed;
                  if (width < 0)
                    {
                      /*
                       * A negative width is the same as the - flag and
                       * a positive width.
                       */
                      flags |= FLAGS_LEFTADJUST;
                      flags &= ~FLAGS_NILPADDING;
                      width = -width;
                    }
                }

              /* Find precision */
              if (flags & FLAGS_PRECISION)
                {
                  precision = parameters[i].precision;
                  if (flags & FLAGS_PRECISION_PARAMETER)
                    {
```

```
                    /* Get precision from parameter list */
                    precision = (int)parameters[precision].data.number.as_signed;
                    if (precision < 0)
                      {
                        /*
                         * A negative precision is the same as no
                         * precision
                         */
                        precision = NO_PRECISION;
                      }
                  }
              }
            else
              {
                precision = NO_PRECISION;
              }

            /* Find base */
            base = parameters[i].base;
            if (flags & FLAGS_BASE_PARAMETER)
              {
                /* Get base from parameter list */
                base = (int)parameters[base].data.number.as_signed;
              }

            switch (parameters[i].type)
              {
              case FORMAT_CHAR:
                if (flags & FLAGS_QUOTE)
                  data->OutStream(data, CHAR_QUOTE);
                if (! (flags & FLAGS_LEFTADJUST))
                  {
                    while (--width > 0)
                      data->OutStream(data, CHAR_ADJUST);
                  }
#if TRIO_WIDECHAR
                if (flags & FLAGS_WIDECHAR)
                  {
                    TrioWriteWideStringCharacter(data,
                                                 (trio_wchar_t)parameters[i].data.number.as_signed,
                                                 flags,
                                                 NO_WIDTH);
                  }
                else
#endif
                  {
                    TrioWriteStringCharacter(data,
                                             (int)parameters[i].data.number.as_signed,
                                             flags);
                  }

                if (flags & FLAGS_LEFTADJUST)
                  {
                    while(--width > 0)
                      data->OutStream(data, CHAR_ADJUST);
                  }
                if (flags & FLAGS_QUOTE)
                  data->OutStream(data, CHAR_QUOTE);

                break; /* FORMAT_CHAR */

              case FORMAT_INT:
                TrioWriteNumber(data,
                                parameters[i].data.number.as_unsigned,
                                flags,
                                width,
                                precision,
                                base);

                break; /* FORMAT_INT */

              case FORMAT_DOUBLE:
                TrioWriteDouble(data,
                                parameters[i].data.longdoubleNumber,
                                flags,
                                width,
                                precision,
                                base);
                break; /* FORMAT_DOUBLE */

              case FORMAT_STRING:
#if TRIO_WIDECHAR
                if (flags & FLAGS_WIDECHAR)
                  {
                    TrioWriteWideString(data,
                                        parameters[i].data.wstring,
                                        flags,
                                        width,
                                        precision);
                  }
                else
#endif
                  {
                    TrioWriteString(data,
                                    parameters[i].data.string,
                                    flags,
                                    width,
                                    precision);
```

```
                  }
                break; /* FORMAT_STRING */

              case FORMAT_POINTER:
                {
                  trio_reference_t reference;

                  reference.data = data;
                  reference.parameter = ¶meters[i];
                  trio_print_pointer(&reference, parameters[i].data.pointer);
                }
                break; /* FORMAT_POINTER */

              case FORMAT_COUNT:
                pointer = parameters[i].data.pointer;
                if (NULL != pointer)
                  {
                    /*
                     * C99 paragraph 7.19.6.1.8 says "the number of
                     * characters written to the output stream so far by
                     * this call", which is data->committed
                     */
#if defined(QUALIFIER_SIZE_T) || defined(QUALIFIER_SIZE_T_UPPER)
                    if (flags & FLAGS_SIZE_T)
                      *(size_t *)pointer = (size_t)data->committed;
                    else
#endif
#if defined(QUALIFIER_PTRDIFF_T)
                    if (flags & FLAGS_PTRDIFF_T)
                      *(ptrdiff_t *)pointer = (ptrdiff_t)data->committed;
                    else
#endif
#if defined(QUALIFIER_INTMAX_T)
                    if (flags & FLAGS_INTMAX_T)
                      *(trio_intmax_t *)pointer = (trio_intmax_t)data->committed;
                    else
#endif
                    if (flags & FLAGS_QUAD)
                      {
                        *(trio_ulonglong_t *)pointer = (trio_ulonglong_t)data->committed;
                      }
                    else if (flags & FLAGS_LONG)
                      {
                        *(long int *)pointer = (long int)data->committed;
                      }
                    else if (flags & FLAGS_SHORT)
                      {
                        *(short int *)pointer = (short int)data->committed;
                      }
                    else
                      {
                        *(int *)pointer = (int)data->committed;
                      }
                  }
                break; /* FORMAT_COUNT */

              case FORMAT_PARAMETER:
                break; /* FORMAT_PARAMETER */

#if defined(FORMAT_ERRNO)
              case FORMAT_ERRNO:
                string = trio_error(parameters[i].data.errorNumber);
                if (string)
                  {
                    TrioWriteString(data,
                                    string,
                                    flags,
                                    width,
                                    precision);
                  }
                else
                  {
                    data->OutStream(data, '#');
                    TrioWriteNumber(data,
                                    (trio_uintmax_t)parameters[i].data.errorNumber,
                                    flags,
                                    width,
                                    precision,
                                    BASE_DECIMAL);
                  }
                break; /* FORMAT_ERRNO */
#endif /* defined(FORMAT_ERRNO) */

#if defined(FORMAT_USER_DEFINED)
              case FORMAT_USER_DEFINED:
                {
                  trio_reference_t reference;
                  trio_userdef_t *def = NULL;

                  if (parameters[i].user_name[0] == NIL)
                    {
                      /* Use handle */
                      if ((i > 0) ||
                          (parameters[i - 1].type == FORMAT_PARAMETER))
                        def = (trio_userdef_t *)parameters[i - 1].data.pointer;
                    }
                  else
                    {
```

```
                      /* Look up namespace */
                      def = TrioFindNamespace(parameters[i].user_name, NULL);
                    }
                  if (def) {
                    reference.data = data;
                    reference.parameter = ¶meters[i];
                    def->callback(&reference);
                  }
                }
                break;
#endif /* defined(FORMAT_USER_DEFINED) */

              default:
                break;
              } /* switch parameter type */

            /* Prepare for next */
            index = parameters[i].indexAfterSpecifier;
            i++;
          }
      }
    else /* not identifier */
      {
        data->OutStream(data, format[index++]);
      }
  }
  return data->processed;
}

/*************************************************************************
 * TrioFormatRef
 */
TRIO_PRIVATE int
TrioFormatRef
TRIO_ARGS4((reference, format, arglist, argarray),
           trio_reference_t *reference,
           TRIO_CONST char *format,
           TRIO_VA_LIST_PTR arglist,
           trio_pointer_t *argarray)
{
  int status;
  trio_parameter_t parameters[MAX_PARAMETERS];

  status = TrioParse(TYPE_PRINT, format, parameters, arglist, argarray);
  if (status < 0)
    return status;

  status = TrioFormatProcess(reference->data, format, parameters);
  if (reference->data->error != 0)
    {
      status = reference->data->error;
    }
  return status;
}

/*************************************************************************
 * TrioFormat
 */
TRIO_PRIVATE int
TrioFormat
TRIO_ARGS6((destination, destinationSize, OutStream, format, arglist, argarray),
           trio_pointer_t destination,
           size_t destinationSize,
           void (*OutStream) TRIO_PROTO((trio_class_t *, int)),
           TRIO_CONST char *format,
           TRIO_VA_LIST_PTR arglist,
           trio_pointer_t *argarray)
{
  int status;
  trio_class_t data;
  trio_parameter_t parameters[MAX_PARAMETERS];

  assert(VALID(OutStream));
  assert(VALID(format));

  memset(&data, 0, sizeof(data));
  data.OutStream = OutStream;
  data.location = destination;
  data.max = destinationSize;
  data.error = 0;

#if defined(USE_LOCALE)
  if (NULL == internalLocaleValues)
    {
      TrioSetLocale();
    }
#endif

  status = TrioParse(TYPE_PRINT, format, parameters, arglist, argarray);
  if (status < 0)
    return status;

  status = TrioFormatProcess(&data, format, parameters);
  if (data.error != 0)
    {
      status = data.error;
    }
  return status;
```

```
}

/*************************************************************************
 * TrioOutStreamFile
 */
TRIO_PRIVATE void
TrioOutStreamFile
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
  FILE *file;

  assert(VALID(self));
  assert(VALID(self->location));

  file = (FILE *)self->location;
  self->processed++;
  if (fputc(output, file) == EOF)
    {
      self->error = TRIO_ERROR_RETURN(TRIO_EOF, 0);
    }
  else
    {
      self->committed++;
    }
}

/*************************************************************************
 * TrioOutStreamFileDescriptor
 */
TRIO_PRIVATE void
TrioOutStreamFileDescriptor
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
  int fd;
  char ch;

  assert(VALID(self));

  fd = *((int *)self->location);
  ch = (char)output;
  self->processed++;
  if (write(fd, &ch, sizeof(char)) == -1)
    {
      self->error = TRIO_ERROR_RETURN(TRIO_ERRNO, 0);
    }
  else
    {
      self->committed++;
    }
}

/*************************************************************************
 * TrioOutStreamCustom
 */
TRIO_PRIVATE void
TrioOutStreamCustom
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
  int status;
  trio_custom_t *data;

  assert(VALID(self));
  assert(VALID(self->location));

  data = (trio_custom_t *)self->location;
  if (data->stream.out)
    {
      status = (data->stream.out)(data->closure, output);
      if (status >= 0)
        {
          self->committed++;
        }
      else
        {
          if (self->error == 0)
            {
              self->error = TRIO_ERROR_RETURN(TRIO_ECUSTOM, -status);
            }
        }
    }
  self->processed++;
}

/*************************************************************************
 * TrioOutStreamString
 */
TRIO_PRIVATE void
TrioOutStreamString
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
```

```
  char **buffer;

  assert(VALID(self));
  assert(VALID(self->location));

  buffer = (char **)self->location;
  **buffer = (char)output;
  (*buffer)++;
  self->processed++;
  self->committed++;
}

/*************************************************************************
 * TrioOutStreamStringMax
 */
TRIO_PRIVATE void
TrioOutStreamStringMax
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
  char **buffer;

  assert(VALID(self));
  assert(VALID(self->location));

  buffer = (char **)self->location;

  if (self->processed < self->max)
    {
      **buffer = (char)output;
      (*buffer)++;
      self->committed++;
    }
  self->processed++;
}

/*************************************************************************
 * TrioOutStreamStringDynamic
 */
TRIO_PRIVATE void
TrioOutStreamStringDynamic
TRIO_ARGS2((self, output),
           trio_class_t *self,
           int output)
{
  assert(VALID(self));
  assert(VALID(self->location));

  if (self->error == 0)
    {
      trio_xstring_append_char((trio_string_t *)self->location,
                               (char)output);
      self->committed++;
    }
  /* The processed variable must always be increased */
  self->processed++;
}

/*************************************************************************
 *
 * Formatted printing functions
 *
 *************************************************************************/

#if defined(TRIO_DOCUMENTATION)
# include "doc/doc_printf.h"
#endif
/** @addtogroup Printf
    @{
*/

/*************************************************************************
 * printf
 */

/**
   Print to standard output stream.

   @param format Formatting string.
   @param ... Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_printf
TRIO_VARGS2((format, va_alist),
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioFormat(stdout, 0, TrioOutStreamFile, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
```

```
}

/**
   Print to standard output stream.

   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_vprintf
TRIO_ARGS2((format, args),
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(format));

  return TrioFormat(stdout, 0, TrioOutStreamFile, format, TRIO_VA_LIST_ADDR(args), NULL);
}

/**
   Print to standard output stream.

   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_printfv
TRIO_ARGS2((format, args),
           TRIO_CONST char *format,
           trio_pointer_t * args)
{
  assert(VALID(format));

  return TrioFormat(stdout, 0, TrioOutStreamFile, format, NULL, args);
}

/*************************************************************************
 * fprintf
 */

/**
   Print to file.

   @param file File pointer.
   @param format Formatting string.
   @param ... Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_fprintf
TRIO_VARGS3((file, format, va_alist),
            FILE *file,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(file));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioFormat(file, 0, TrioOutStreamFile, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

/**
   Print to file.

   @param file File pointer.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_vfprintf
TRIO_ARGS3((file, format, args),
           FILE *file,
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(file));
  assert(VALID(format));

  return TrioFormat(file, 0, TrioOutStreamFile, format, TRIO_VA_LIST_ADDR(args), NULL);
}

/**
   Print to file.

   @param file File pointer.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
```

```
TRIO_PUBLIC int
trio_fprintfv
TRIO_ARGS3((file, format, args),
           FILE *file,
           TRIO_CONST char *format,
           trio_pointer_t * args)
{
  assert(VALID(file));
  assert(VALID(format));

  return TrioFormat(file, 0, TrioOutStreamFile, format, NULL, args);
}

/*************************************************************************
 * dprintf
 */

/**
   Print to file descriptor.

   @param fd File descriptor.
   @param format Formatting string.
   @param ... Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_dprintf
TRIO_VARGS3((fd, format, va_alist),
            int fd,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioFormat(&fd, 0, TrioOutStreamFileDescriptor, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

/**
   Print to file descriptor.

   @param fd File descriptor.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_vdprintf
TRIO_ARGS3((fd, format, args),
           int fd,
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(format));

  return TrioFormat(&fd, 0, TrioOutStreamFileDescriptor, format, TRIO_VA_LIST_ADDR(args), NULL);
}

/**
   Print to file descriptor.

   @param fd File descriptor.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_dprintfv
TRIO_ARGS3((fd, format, args),
           int fd,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  assert(VALID(format));

  return TrioFormat(&fd, 0, TrioOutStreamFileDescriptor, format, NULL, args);
}

/*************************************************************************
 * cprintf
 */
TRIO_PUBLIC int
trio_cprintf
TRIO_VARGS4((stream, closure, format, va_alist),
            trio_outstream_t stream,
            trio_pointer_t closure,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;
  trio_custom_t data;
```

```
  assert(VALID(stream));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  data.stream.out = stream;
  data.closure = closure;
  status = TrioFormat(&data, 0, TrioOutStreamCustom, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vcprintf
TRIO_ARGS4((stream, closure, format, args),
           trio_outstream_t stream,
           trio_pointer_t closure,
           TRIO_CONST char *format,
           va_list args)
{
  trio_custom_t data;

  assert(VALID(stream));
  assert(VALID(format));

  data.stream.out = stream;
  data.closure = closure;
  return TrioFormat(&data, 0, TrioOutStreamCustom, format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_cprintfv
TRIO_ARGS4((stream, closure, format, args),
           trio_outstream_t stream,
           trio_pointer_t closure,
           TRIO_CONST char *format,
           void **args)
{
  trio_custom_t data;

  assert(VALID(stream));
  assert(VALID(format));

  data.stream.out = stream;
  data.closure = closure;
  return TrioFormat(&data, 0, TrioOutStreamCustom, format, NULL, args);
}

/*************************************************************************
 * sprintf
 */

/**
   Print to string.

   @param buffer Output string.
   @param format Formatting string.
   @param ... Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_sprintf
TRIO_VARGS3((buffer, format, va_alist),
            char *buffer,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(buffer));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioFormat(&buffer, 0, TrioOutStreamString, format, TRIO_VA_LIST_ADDR(args), NULL);
  *buffer = NIL; /* Terminate with NIL character */
  TRIO_VA_END(args);
  return status;
}

/**
   Print to string.

   @param buffer Output string.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_vsprintf
TRIO_ARGS3((buffer, format, args),
           char *buffer,
           TRIO_CONST char *format,
           va_list args)
{
  int status;

  assert(VALID(buffer));
  assert(VALID(format));
```

```
  status = TrioFormat(&buffer, 0, TrioOutStreamString, format, TRIO_VA_LIST_ADDR(args), NULL);
  *buffer = NIL;
  return status;
}

/**
   Print to string.

   @param buffer Output string.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_sprintfv
TRIO_ARGS3((buffer, format, args),
           char *buffer,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  int status;

  assert(VALID(buffer));
  assert(VALID(format));

  status = TrioFormat(&buffer, 0, TrioOutStreamString, format, NULL, args);
  *buffer = NIL;
  return status;
}

/*************************************************************************
 * snprintf
 */

/**
   Print at most @p max characters to string.

   @param buffer Output string.
   @param max Maximum number of characters to print.
   @param format Formatting string.
   @param ... Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_snprintf
TRIO_VARGS4((buffer, max, format, va_alist),
            char *buffer,
            size_t max,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(buffer));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioFormat(&buffer, max > 0 ? max - 1 : 0,
                      TrioOutStreamStringMax, format, TRIO_VA_LIST_ADDR(args), NULL);
  if (max > 0)
    *buffer = NIL;
  TRIO_VA_END(args);
  return status;
}

/**
   Print at most @p max characters to string.

   @param buffer Output string.
   @param max Maximum number of characters to print.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_vsnprintf
TRIO_ARGS4((buffer, max, format, args),
           char *buffer,
           size_t max,
           TRIO_CONST char *format,
           va_list args)
{
  int status;

  assert(VALID(buffer));
  assert(VALID(format));

  status = TrioFormat(&buffer, max > 0 ? max - 1 : 0,
                      TrioOutStreamStringMax, format, TRIO_VA_LIST_ADDR(args), NULL);
  if (max > 0)
    *buffer = NIL;
  return status;
}

/**
   Print at most @p max characters to string.
```

```
   @param buffer Output string.
   @param max Maximum number of characters to print.
   @param format Formatting string.
   @param args Arguments.
   @return Number of printed characters.
 */
TRIO_PUBLIC int
trio_snprintfv
TRIO_ARGS4((buffer, max, format, args),
           char *buffer,
           size_t max,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  int status;

  assert(VALID(buffer));
  assert(VALID(format));

  status = TrioFormat(&buffer, max > 0 ? max - 1 : 0,
                      TrioOutStreamStringMax, format, NULL, args);
  if (max > 0)
    *buffer = NIL;
  return status;
}

/*************************************************************************
 * snprintfcat
 * Appends the new string to the buffer string overwriting the '\0'
 * character at the end of buffer.
 */
TRIO_PUBLIC int
trio_snprintfcat
TRIO_VARGS4((buffer, max, format, va_alist),
            char *buffer,
            size_t max,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;
  size_t buf_len;

  TRIO_VA_START(args, format);

  assert(VALID(buffer));
  assert(VALID(format));

  buf_len = trio_length(buffer);
  buffer = &buffer[buf_len];

  status = TrioFormat(&buffer, max - 1 - buf_len,
                      TrioOutStreamStringMax, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  *buffer = NIL;
  return status;
}

TRIO_PUBLIC int
trio_vsnprintfcat
TRIO_ARGS4((buffer, max, format, args),
           char *buffer,
           size_t max,
           TRIO_CONST char *format,
           va_list args)
{
  int status;
  size_t buf_len;

  assert(VALID(buffer));
  assert(VALID(format));

  buf_len = trio_length(buffer);
  buffer = &buffer[buf_len];
  status = TrioFormat(&buffer, max - 1 - buf_len,
                      TrioOutStreamStringMax, format, TRIO_VA_LIST_ADDR(args), NULL);
  *buffer = NIL;
  return status;
}

/*************************************************************************
 * trio_aprintf
 */

/* Deprecated */
TRIO_PUBLIC char *
trio_aprintf
TRIO_VARGS2((format, va_alist),
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  va_list args;
  trio_string_t *info;
  char *result = NULL;

  assert(VALID(format));
```

```
  info = trio_xstring_duplicate("");
  if (info)
    {
      TRIO_VA_START(args, format);
      (void)TrioFormat(info, 0, TrioOutStreamStringDynamic,
                       format, TRIO_VA_LIST_ADDR(args), NULL);
      TRIO_VA_END(args);

      trio_string_terminate(info);
      result = trio_string_extract(info);
      trio_string_destroy(info);
    }
  return result;
}

/* Deprecated */
TRIO_PUBLIC char *
trio_vaprintf
TRIO_ARGS2((format, args),
           TRIO_CONST char *format,
           va_list args)
{
  trio_string_t *info;
  char *result = NULL;

  assert(VALID(format));

  info = trio_xstring_duplicate("");
  if (info)
    {
      (void)TrioFormat(info, 0, TrioOutStreamStringDynamic,
                       format, TRIO_VA_LIST_ADDR(args), NULL);
      trio_string_terminate(info);
      result = trio_string_extract(info);
      trio_string_destroy(info);
    }
  return result;
}

TRIO_PUBLIC int
trio_asprintf
TRIO_VARGS3((result, format, va_alist),
            char **result,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  va_list args;
  int status;
  trio_string_t *info;

  assert(VALID(format));

  *result = NULL;

  info = trio_xstring_duplicate("");
  if (info == NULL)
    {
      status = TRIO_ERROR_RETURN(TRIO_ENOMEM, 0);
    }
  else
    {
      TRIO_VA_START(args, format);
      status = TrioFormat(info, 0, TrioOutStreamStringDynamic,
                          format, TRIO_VA_LIST_ADDR(args), NULL);
      TRIO_VA_END(args);
      if (status >= 0)
        {
          trio_string_terminate(info);
          *result = trio_string_extract(info);
        }
      trio_string_destroy(info);
    }
  return status;
}

TRIO_PUBLIC int
trio_vasprintf
TRIO_ARGS3((result, format, args),
           char **result,
           TRIO_CONST char *format,
           va_list args)
{
  int status;
  trio_string_t *info;

  assert(VALID(format));

  *result = NULL;

  info = trio_xstring_duplicate("");
  if (info == NULL)
    {
      status = TRIO_ERROR_RETURN(TRIO_ENOMEM, 0);
    }
  else
    {
      status = TrioFormat(info, 0, TrioOutStreamStringDynamic,
                          format, TRIO_VA_LIST_ADDR(args), NULL);
```

```
      if (status >= 0)
        {
          trio_string_terminate(info);
          *result = trio_string_extract(info);
        }
      trio_string_destroy(info);
    }
  return status;
}

/** @} End of Printf documentation module */

/*************************************************************************
 *
 * CALLBACK
 *
 ************************************************************************/

#if defined(TRIO_DOCUMENTATION)
# include "doc/doc_register.h"
#endif
/**
   @addtogroup UserDefined
   @{
*/

#if TRIO_EXTENSION

/*************************************************************************
 * trio_register
 */

/**
   Register new user-defined specifier.

   @param callback
   @param name
   @return Handle.
 */
TRIO_PUBLIC trio_pointer_t
trio_register
TRIO_ARGS2((callback, name),
           trio_callback_t callback,
           TRIO_CONST char *name)
{
  trio_userdef_t *def;
  trio_userdef_t *prev = NULL;

  if (callback == NULL)
    return NULL;

  if (name)
    {
      /* Handle built-in namespaces */
      if (name[0] == ':')
        {
          if (trio_equal(name, ":enter"))
            {
              internalEnterCriticalRegion = callback;
            }
          else if (trio_equal(name, ":leave"))
            {
              internalLeaveCriticalRegion = callback;
            }
          return NULL;
        }

      /* Bail out if namespace is too long */
      if (trio_length(name) >= MAX_USER_NAME)
        return NULL;

      /* Bail out if namespace already is registered */
      def = TrioFindNamespace(name, &prev);
      if (def)
        return NULL;
    }

  def = (trio_userdef_t *)TRIO_MALLOC(sizeof(trio_userdef_t));
  if (def)
    {
      if (internalEnterCriticalRegion)
        (void)internalEnterCriticalRegion(NULL);

      if (name)
        {
          /* Link into internal list */
          if (prev == NULL)
            internalUserDef = def;
          else
            prev->next = def;
        }
      /* Initialize */
      def->callback = callback;
      def->name = (name == NULL)
        ? NULL
        : trio_duplicate(name);
      def->next = NULL;
```

```
      if (internalLeaveCriticalRegion)
        (void)internalLeaveCriticalRegion(NULL);
    }
  return (trio_pointer_t)def;
}

/**
   Unregister an existing user-defined specifier.

   @param handle
 */
void
trio_unregister
TRIO_ARGS1((handle),
           trio_pointer_t handle)
{
  trio_userdef_t *self = (trio_userdef_t *)handle;
  trio_userdef_t *def;
  trio_userdef_t *prev = NULL;

  assert(VALID(self));

  if (self->name)
    {
      def = TrioFindNamespace(self->name, &prev);
      if (def)
        {
          if (internalEnterCriticalRegion)
            (void)internalEnterCriticalRegion(NULL);

          if (prev == NULL)
            internalUserDef = NULL;
          else
            prev->next = def->next;

          if (internalLeaveCriticalRegion)
            (void)internalLeaveCriticalRegion(NULL);
        }
      trio_destroy(self->name);
    }
  TRIO_FREE(self);
}

/*************************************************************************
 * trio_get_format [public]
 */
TRIO_CONST char *
trio_get_format
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
#if defined(FORMAT_USER_DEFINED)
  assert(((trio_reference_t *)ref)->parameter->type == FORMAT_USER_DEFINED);
#endif

  return (((trio_reference_t *)ref)->parameter->user_data);
}

/*************************************************************************
 * trio_get_argument [public]
 */
trio_pointer_t
trio_get_argument
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
#if defined(FORMAT_USER_DEFINED)
  assert(((trio_reference_t *)ref)->parameter->type == FORMAT_USER_DEFINED);
#endif

  return ((trio_reference_t *)ref)->parameter->data.pointer;
}

/*************************************************************************
 * trio_get_width / trio_set_width [public]
 */
int
trio_get_width
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return ((trio_reference_t *)ref)->parameter->width;
}

void
trio_set_width
TRIO_ARGS2((ref, width),
           trio_pointer_t ref,
           int width)
{
  ((trio_reference_t *)ref)->parameter->width = width;
}

/*************************************************************************
 * trio_get_precision / trio_set_precision [public]
 */
int
trio_get_precision
```

```
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->precision);
}

void
trio_set_precision
TRIO_ARGS2((ref, precision),
           trio_pointer_t ref,
           int precision)
{
  ((trio_reference_t *)ref)->parameter->precision = precision;
}

/*************************************************************************
 * trio_get_base / trio_set_base [public]
 */
int
trio_get_base
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->base);
}

void
trio_set_base
TRIO_ARGS2((ref, base),
           trio_pointer_t ref,
           int base)
{
  ((trio_reference_t *)ref)->parameter->base = base;
}

/*************************************************************************
 * trio_get_long / trio_set_long [public]
 */
int
trio_get_long
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_LONG)
    ? TRUE
    : FALSE;
}

void
trio_set_long
TRIO_ARGS2((ref, is_long),
           trio_pointer_t ref,
           int is_long)
{
  if (is_long)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_LONG;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_LONG;
}

/*************************************************************************
 * trio_get_longlong / trio_set_longlong [public]
 */
int
trio_get_longlong
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_QUAD)
    ? TRUE
    : FALSE;
}

void
trio_set_longlong
TRIO_ARGS2((ref, is_longlong),
           trio_pointer_t ref,
           int is_longlong)
{
  if (is_longlong)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_QUAD;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_QUAD;
}

/*************************************************************************
 * trio_get_longdouble / trio_set_longdouble [public]
 */
int
trio_get_longdouble
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_LONGDOUBLE)
    ? TRUE
    : FALSE;
}
```

```
void
trio_set_longdouble
TRIO_ARGS2((ref, is_longdouble),
           trio_pointer_t ref,
           int is_longdouble)
{
  if (is_longdouble)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_LONGDOUBLE;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_LONGDOUBLE;
}

/*************************************************************************
 * trio_get_short / trio_set_short [public]
 */
int
trio_get_short
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_SHORT)
    ? TRUE
    : FALSE;
}

void
trio_set_short
TRIO_ARGS2((ref, is_short),
           trio_pointer_t ref,
           int is_short)
{
  if (is_short)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_SHORT;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_SHORT;
}

/*************************************************************************
 * trio_get_shortshort / trio_set_shortshort [public]
 */
int
trio_get_shortshort
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_SHORTSHORT)
    ? TRUE
    : FALSE;
}

void
trio_set_shortshort
TRIO_ARGS2((ref, is_shortshort),
           trio_pointer_t ref,
           int is_shortshort)
{
  if (is_shortshort)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_SHORTSHORT;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_SHORTSHORT;
}

/*************************************************************************
 * trio_get_alternative / trio_set_alternative [public]
 */
int
trio_get_alternative
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_ALTERNATIVE)
    ? TRUE
    : FALSE;
}

void
trio_set_alternative
TRIO_ARGS2((ref, is_alternative),
           trio_pointer_t ref,
           int is_alternative)
{
  if (is_alternative)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_ALTERNATIVE;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_ALTERNATIVE;
}

/*************************************************************************
 * trio_get_alignment / trio_set_alignment [public]
 */
int
trio_get_alignment
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_LEFTADJUST)
    ? TRUE
    : FALSE;
```

```
}

void
trio_set_alignment
TRIO_ARGS2((ref, is_leftaligned),
           trio_pointer_t ref,
           int is_leftaligned)
{
  if (is_leftaligned)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_LEFTADJUST;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_LEFTADJUST;
}

/*************************************************************************
 * trio_get_spacing /trio_set_spacing [public]
 */
int
trio_get_spacing
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_SPACE)
    ? TRUE
    : FALSE;
}

void
trio_set_spacing
TRIO_ARGS2((ref, is_space),
           trio_pointer_t ref,
           int is_space)
{
  if (is_space)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_SPACE;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_SPACE;
}

/*************************************************************************
 * trio_get_sign / trio_set_sign [public]
 */
int
trio_get_sign
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_SHOWSIGN)
    ? TRUE
    : FALSE;
}

void
trio_set_sign
TRIO_ARGS2((ref, is_sign),
           trio_pointer_t ref,
           int is_sign)
{
  if (is_sign)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_SHOWSIGN;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_SHOWSIGN;
}

/*************************************************************************
 * trio_get_padding / trio_set_padding [public]
 */
int
trio_get_padding
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_NILPADDING)
    ? TRUE
    : FALSE;
}

void
trio_set_padding
TRIO_ARGS2((ref, is_padding),
           trio_pointer_t ref,
           int is_padding)
{
  if (is_padding)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_NILPADDING;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_NILPADDING;
}

/*************************************************************************
 * trio_get_quote / trio_set_quote [public]
 */
int
trio_get_quote
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_QUOTE)
```

```
    ? TRUE
    : FALSE;
}

void
trio_set_quote
TRIO_ARGS2((ref, is_quote),
           trio_pointer_t ref,
           int is_quote)
{
  if (is_quote)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_QUOTE;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_QUOTE;
}

/*************************************************************************
 * trio_get_upper / trio_set_upper [public]
 */
int
trio_get_upper
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_UPPER)
    ? TRUE
    : FALSE;
}

void
trio_set_upper
TRIO_ARGS2((ref, is_upper),
           trio_pointer_t ref,
           int is_upper)
{
  if (is_upper)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_UPPER;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_UPPER;
}

/*************************************************************************
 * trio_get_largest / trio_set_largest [public]
 */
#if TRIO_C99
int
trio_get_largest
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_INTMAX_T)
    ? TRUE
    : FALSE;
}

void
trio_set_largest
TRIO_ARGS2((ref, is_largest),
           trio_pointer_t ref,
           int is_largest)
{
  if (is_largest)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_INTMAX_T;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_INTMAX_T;
}
#endif

/*************************************************************************
 * trio_get_ptrdiff / trio_set_ptrdiff [public]
 */
int
trio_get_ptrdiff
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_PTRDIFF_T)
    ? TRUE
    : FALSE;
}

void
trio_set_ptrdiff
TRIO_ARGS2((ref, is_ptrdiff),
           trio_pointer_t ref,
           int is_ptrdiff)
{
  if (is_ptrdiff)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_PTRDIFF_T;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_PTRDIFF_T;
}

/*************************************************************************
 * trio_get_size / trio_set_size [public]
 */
#if TRIO_C99
int
```

```
trio_get_size
TRIO_ARGS1((ref),
           trio_pointer_t ref)
{
  return (((trio_reference_t *)ref)->parameter->flags & FLAGS_SIZE_T)
    ? TRUE
    : FALSE;
}

void
trio_set_size
TRIO_ARGS2((ref, is_size),
           trio_pointer_t ref,
           int is_size)
{
  if (is_size)
    ((trio_reference_t *)ref)->parameter->flags |= FLAGS_SIZE_T;
  else
    ((trio_reference_t *)ref)->parameter->flags &= ~FLAGS_SIZE_T;
}
#endif

/*************************************************************************
 * trio_print_int [public]
 */
void
trio_print_int
TRIO_ARGS2((ref, number),
           trio_pointer_t ref,
           int number)
{
  trio_reference_t *self = (trio_reference_t *)ref;

  TrioWriteNumber(self->data,
                  (trio_uintmax_t)number,
                  self->parameter->flags,
                  self->parameter->width,
                  self->parameter->precision,
                  self->parameter->base);
}

/*************************************************************************
 * trio_print_uint [public]
 */
void
trio_print_uint
TRIO_ARGS2((ref, number),
           trio_pointer_t ref,
           unsigned int number)
{
  trio_reference_t *self = (trio_reference_t *)ref;

  TrioWriteNumber(self->data,
                  (trio_uintmax_t)number,
                  self->parameter->flags | FLAGS_UNSIGNED,
                  self->parameter->width,
                  self->parameter->precision,
                  self->parameter->base);
}

/*************************************************************************
 * trio_print_double [public]
 */
void
trio_print_double
TRIO_ARGS2((ref, number),
           trio_pointer_t ref,
           double number)
{
  trio_reference_t *self = (trio_reference_t *)ref;

  TrioWriteDouble(self->data,
                  number,
                  self->parameter->flags,
                  self->parameter->width,
                  self->parameter->precision,
                  self->parameter->base);
}

/*************************************************************************
 * trio_print_string [public]
 */
void
trio_print_string
TRIO_ARGS2((ref, string),
           trio_pointer_t ref,
           char *string)
{
  trio_reference_t *self = (trio_reference_t *)ref;

  TrioWriteString(self->data,
                  string,
                  self->parameter->flags,
                  self->parameter->width,
                  self->parameter->precision);
}

/*************************************************************************
```

```
 * trio_print_ref [public]
 */
int
trio_print_ref
TRIO_VARGS3((ref, format, va_alist),
            trio_pointer_t ref,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list arglist;

  assert(VALID(format));

  TRIO_VA_START(arglist, format);
  status = TrioFormatRef((trio_reference_t *)ref, format, TRIO_VA_LIST_ADDR(arglist), NULL);
  TRIO_VA_END(arglist);
  return status;
}

/*************************************************************************
 * trio_vprint_ref [public]
 */
int
trio_vprint_ref
TRIO_ARGS3((ref, format, arglist),
           trio_pointer_t ref,
           TRIO_CONST char *format,
           va_list arglist)
{
  assert(VALID(format));

  return TrioFormatRef((trio_reference_t *)ref, format, TRIO_VA_LIST_ADDR(arglist), NULL);
}

/*************************************************************************
 * trio_printv_ref [public]
 */
int
trio_printv_ref
TRIO_ARGS3((ref, format, argarray),
           trio_pointer_t ref,
           TRIO_CONST char *format,
           trio_pointer_t *argarray)
{
  assert(VALID(format));

  return TrioFormatRef((trio_reference_t *)ref, format, NULL, argarray);
}

#endif /* TRIO_EXTENSION */

/*************************************************************************
 * trio_print_pointer [public]
 */
void
trio_print_pointer
TRIO_ARGS2((ref, pointer),
           trio_pointer_t ref,
           trio_pointer_t pointer)
{
  trio_reference_t *self = (trio_reference_t *)ref;
  trio_flags_t flags;
  trio_uintmax_t number;

  if (NULL == pointer)
    {
      TRIO_CONST char *string = internalNullString;
      while (*string)
        self->data->OutStream(self->data, *string++);
    }
  else
    {
      /*
       * The subtraction of the null pointer is a workaround
       * to avoid a compiler warning. The performance overhead
       * is negligible (and likely to be removed by an
       * optimizing compiler). The (char *) casting is done
       * to please ANSI C++.
       */
      number = (trio_uintmax_t)((char *)pointer - (char *)0);
      /* Shrink to size of pointer */
      number &= (trio_uintmax_t)-1;
      flags = self->parameter->flags;
      flags |= (FLAGS_UNSIGNED | FLAGS_ALTERNATIVE |
                FLAGS_NILPADDING);
      TrioWriteNumber(self->data,
                      number,
                      flags,
                      POINTER_WIDTH,
                      NO_PRECISION,
                      BASE_HEX);
    }
}

/** @} End of UserDefined documentation module */

/*************************************************************************
```

```
 *
 * LOCALES
 *
 ************************************************************************/

/*************************************************************************
 * trio_locale_set_decimal_point
 *
 * Decimal point can only be one character. The input argument is a
 * string to enable multibyte characters. At most MB_LEN_MAX characters
 * will be used.
 */
TRIO_PUBLIC void
trio_locale_set_decimal_point
TRIO_ARGS1((decimalPoint),
           char *decimalPoint)
{
#if defined(USE_LOCALE)
  if (NULL == internalLocaleValues)
    {
      TrioSetLocale();
    }
#endif
  internalDecimalPointLength = trio_length(decimalPoint);
  if (internalDecimalPointLength == 1)
    {
      internalDecimalPoint = *decimalPoint;
    }
  else
    {
      internalDecimalPoint = NIL;
      trio_copy_max(internalDecimalPointString,
                    sizeof(internalDecimalPointString),
                    decimalPoint);
    }
}

/*************************************************************************
 * trio_locale_set_thousand_separator
 *
 * See trio_locale_set_decimal_point
 */
TRIO_PUBLIC void
trio_locale_set_thousand_separator
TRIO_ARGS1((thousandSeparator),
           char *thousandSeparator)
{
#if defined(USE_LOCALE)
  if (NULL == internalLocaleValues)
    {
      TrioSetLocale();
    }
#endif
  trio_copy_max(internalThousandSeparator,
                sizeof(internalThousandSeparator),
                thousandSeparator);
  internalThousandSeparatorLength = trio_length(internalThousandSeparator);
}

/*************************************************************************
 * trio_locale_set_grouping
 *
 * Array of bytes. Reversed order.
 *
 *  CHAR_MAX : No further grouping
 *  0        : Repeat last group for the remaining digits (not necessary
 *             as C strings are zero-terminated)
 *  n        : Set current group to n
 *
 * Same order as the grouping attribute in LC_NUMERIC.
 */
TRIO_PUBLIC void
trio_locale_set_grouping
TRIO_ARGS1((grouping),
           char *grouping)
{
#if defined(USE_LOCALE)
  if (NULL == internalLocaleValues)
    {
      TrioSetLocale();
    }
#endif
  trio_copy_max(internalGrouping,
                sizeof(internalGrouping),
                grouping);
}


/*************************************************************************
 *
 * SCANNING
 *
 ************************************************************************/

/*************************************************************************
 * TrioSkipWhitespaces
 */
TRIO_PRIVATE int
```

```
TrioSkipWhitespaces
TRIO_ARGS1((self),
           trio_class_t *self)
{
  int ch;

  ch = self->current;
  while (isspace(ch))
    {
      self->InStream(self, &ch);
    }
  return ch;
}

/*************************************************************************
 * TrioGetCollation
 */
#if TRIO_EXTENSION
TRIO_PRIVATE void
TrioGetCollation(TRIO_NOARGS)
{
  int i;
  int j;
  int k;
  char first[2];
  char second[2];

  /* This is computationally expensive */
  first[1] = NIL;
  second[1] = NIL;
  for (i = 0; i < MAX_CHARACTER_CLASS; i++)
    {
      k = 0;
      first[0] = (char)i;
      for (j = 0; j < MAX_CHARACTER_CLASS; j++)
        {
          second[0] = (char)j;
          if (trio_equal_locale(first, second))
            internalCollationArray[i][k++] = (char)j;
        }
      internalCollationArray[i][k] = NIL;
    }
}
#endif

/*************************************************************************
 * TrioGetCharacterClass
 *
 * FIXME:
 *  multibyte
 */
TRIO_PRIVATE int
TrioGetCharacterClass
TRIO_ARGS4((format, indexPointer, flagsPointer, characterclass),
           TRIO_CONST char *format,
           int *indexPointer,
           trio_flags_t *flagsPointer,
           int *characterclass)
{
  int index = *indexPointer;
  int i;
  char ch;
  char range_begin;
  char range_end;

  *flagsPointer &= ~FLAGS_EXCLUDE;

  if (format[index] == QUALIFIER_CIRCUMFLEX)
    {
      *flagsPointer |= FLAGS_EXCLUDE;
      index++;
    }
  /*
   * If the ungroup character is at the beginning of the scanlist,
   * it will be part of the class, and a second ungroup character
   * must follow to end the group.
   */
  if (format[index] == SPECIFIER_UNGROUP)
    {
      characterclass[(int)SPECIFIER_UNGROUP]++;
      index++;
    }
  /*
   * Minus is used to specify ranges. To include minus in the class,
   * it must be at the beginning of the list
   */
  if (format[index] == QUALIFIER_MINUS)
    {
      characterclass[(int)QUALIFIER_MINUS]++;
      index++;
    }
  /* Collect characters */
  for (ch = format[index];
       (ch != SPECIFIER_UNGROUP) && (ch != NIL);
       ch = format[++index])
    {
      switch (ch)
        {
```

```
        case QUALIFIER_MINUS: /* Scanlist ranges */

          /*
           * Both C99 and UNIX98 describes ranges as implementation-
           * defined.
           *
           * We support the following behaviour (although this may
           * change as we become wiser)
           * - only increasing ranges, ie. [a-b] but not [b-a]
           * - transitive ranges, ie. [a-b-c] == [a-c]
           * - trailing minus, ie. [a-] is interpreted as an 'a'
           *   and a '-'
           * - duplicates (although we can easily convert these
           *   into errors)
           */
          range_begin = format[index - 1];
          range_end = format[++index];
          if (range_end == SPECIFIER_UNGROUP)
            {
              /* Trailing minus is included */
              characterclass[(int)ch]++;
              ch = range_end;
              break; /* for */
            }
          if (range_end == NIL)
            return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
          if (range_begin > range_end)
            return TRIO_ERROR_RETURN(TRIO_ERANGE, index);

          for (i = (int)range_begin; i <= (int)range_end; i++)
            characterclass[i]++;

          ch = range_end;
          break;

#if TRIO_EXTENSION

        case SPECIFIER_GROUP:

          switch (format[index + 1])
            {
            case QUALIFIER_DOT: /* Collating symbol */
              /*
               * FIXME: This will be easier to implement when multibyte
               * characters have been implemented. Until now, we ignore
               * this feature.
               */
              for (i = index + 2; ; i++)
                {
                  if (format[i] == NIL)
                    /* Error in syntax */
                    return -1;
                  else if (format[i] == QUALIFIER_DOT)
                    break; /* for */
                }
              if (format[++i] != SPECIFIER_UNGROUP)
                return -1;

              index = i;
              break;

            case QUALIFIER_EQUAL: /* Equivalence class expressions */
              {
                unsigned int j;
                unsigned int k;

                if (internalCollationUnconverted)
                  {
                    /* Lazy evaluation of collation array */
                    TrioGetCollation();
                    internalCollationUnconverted = FALSE;
                  }
                for (i = index + 2; ; i++)
                  {
                    if (format[i] == NIL)
                      /* Error in syntax */
                      return -1;
                    else if (format[i] == QUALIFIER_EQUAL)
                      break; /* for */
                    else
                      {
                        /* Mark any equivalent character */
                        k = (unsigned int)format[i];
                        for (j = 0; internalCollationArray[k][j] != NIL; j++)
                          characterclass[(int)internalCollationArray[k][j]]++;
                      }
                  }
                if (format[++i] != SPECIFIER_UNGROUP)
                  return -1;

                index = i;
              }
              break;

            case QUALIFIER_COLON: /* Character class expressions */

              if (trio_equal_max(CLASS_ALNUM, sizeof(CLASS_ALNUM) - 1,
                                 &format[index]))
```

```
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isalnum(i))
            characterclass[i]++;
        index += sizeof(CLASS_ALNUM) - 1;
      }
    else if (trio_equal_max(CLASS_ALPHA, sizeof(CLASS_ALPHA) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isalpha(i))
            characterclass[i]++;
        index += sizeof(CLASS_ALPHA) - 1;
      }
    else if (trio_equal_max(CLASS_CNTRL, sizeof(CLASS_CNTRL) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (iscntrl(i))
            characterclass[i]++;
        index += sizeof(CLASS_CNTRL) - 1;
      }
    else if (trio_equal_max(CLASS_DIGIT, sizeof(CLASS_DIGIT) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isdigit(i))
            characterclass[i]++;
        index += sizeof(CLASS_DIGIT) - 1;
      }
    else if (trio_equal_max(CLASS_GRAPH, sizeof(CLASS_GRAPH) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isgraph(i))
            characterclass[i]++;
        index += sizeof(CLASS_GRAPH) - 1;
      }
    else if (trio_equal_max(CLASS_LOWER, sizeof(CLASS_LOWER) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (islower(i))
            characterclass[i]++;
        index += sizeof(CLASS_LOWER) - 1;
      }
    else if (trio_equal_max(CLASS_PRINT, sizeof(CLASS_PRINT) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isprint(i))
            characterclass[i]++;
        index += sizeof(CLASS_PRINT) - 1;
      }
    else if (trio_equal_max(CLASS_PUNCT, sizeof(CLASS_PUNCT) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (ispunct(i))
            characterclass[i]++;
        index += sizeof(CLASS_PUNCT) - 1;
      }
    else if (trio_equal_max(CLASS_SPACE, sizeof(CLASS_SPACE) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isspace(i))
            characterclass[i]++;
        index += sizeof(CLASS_SPACE) - 1;
      }
    else if (trio_equal_max(CLASS_UPPER, sizeof(CLASS_UPPER) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isupper(i))
            characterclass[i]++;
        index += sizeof(CLASS_UPPER) - 1;
      }
    else if (trio_equal_max(CLASS_XDIGIT, sizeof(CLASS_XDIGIT) - 1,
                            &format[index]))
      {
        for (i = 0; i < MAX_CHARACTER_CLASS; i++)
          if (isxdigit(i))
            characterclass[i]++;
        index += sizeof(CLASS_XDIGIT) - 1;
      }
    else
      {
        characterclass[(int)ch]++;
      }
    break;

  default:
    characterclass[(int)ch]++;
    break;
  }
break;
```

```
#endif /* TRIO_EXTENSION */

        default:
          characterclass[(int)ch]++;
          break;
        }
    }
  return 0;
}

/*************************************************************************
 * TrioReadNumber
 *
 * We implement our own number conversion in preference of strtol and
 * strtoul, because we must handle 'long long' and thousand separators.
 */
TRIO_PRIVATE BOOLEAN_T
TrioReadNumber
TRIO_ARGS5((self, target, flags, width, base),
           trio_class_t *self,
           trio_uintmax_t *target,
           trio_flags_t flags,
           int width,
           int base)
{
  trio_uintmax_t number = 0;
  int digit;
  int count;
  BOOLEAN_T isNegative = FALSE;
  BOOLEAN_T gotNumber = FALSE;
  int j;

  assert(VALID(self));
  assert(VALID(self->InStream));
  assert((base >= MIN_BASE && base <= MAX_BASE) || (base == NO_BASE));

  if (internalDigitsUnconverted)
    {
      /* Lazy evaluation of digits array */
      memset(internalDigitArray, -1, sizeof(internalDigitArray));
      for (j = 0; j < (int)sizeof(internalDigitsLower) - 1; j++)
        {
          internalDigitArray[(int)internalDigitsLower[j]] = j;
          internalDigitArray[(int)internalDigitsUpper[j]] = j;
        }
      internalDigitsUnconverted = FALSE;
    }

  TrioSkipWhitespaces(self);

  if (!(flags & FLAGS_UNSIGNED))
    {
      /* Leading sign */
      if (self->current == '+')
        {
          self->InStream(self, NULL);
        }
      else if (self->current == '-')
        {
          self->InStream(self, NULL);
          isNegative = TRUE;
        }
    }

  count = self->processed;

  if (flags & FLAGS_ALTERNATIVE)
    {
      switch (base)
        {
        case NO_BASE:
        case BASE_OCTAL:
        case BASE_HEX:
        case BASE_BINARY:
          if (self->current == '0')
            {
              self->InStream(self, NULL);
              if (self->current)
                {
                  if ((base == BASE_HEX) &&
                      (trio_to_upper(self->current) == 'X'))
                    {
                      self->InStream(self, NULL);
                    }
                  else if ((base == BASE_BINARY) &&
                           (trio_to_upper(self->current) == 'B'))
                    {
                      self->InStream(self, NULL);
                    }
                }
            }
          else
            return FALSE;
          break;
        default:
          break;
        }
    }
```

```
  while (((width == NO_WIDTH) || (self->processed - count < width)) &&
         (! ((self->current == EOF) || isspace(self->current))))
    {
      if (isascii(self->current))
        {
          digit = internalDigitArray[self->current];
          /* Abort if digit is not allowed in the specified base */
          if ((digit == -1) || (digit >= base))
            break;
        }
      else if (flags & FLAGS_QUOTE)
        {
          /* Compare with thousands separator */
          for (j = 0; internalThousandSeparator[j] && self->current; j++)
            {
              if (internalThousandSeparator[j] != self->current)
                break;

              self->InStream(self, NULL);
            }
          if (internalThousandSeparator[j])
            break; /* Mismatch */
          else
            continue; /* Match */
        }
      else
        break;

      number *= base;
      number += digit;
      gotNumber = TRUE; /* we need at least one digit */

      self->InStream(self, NULL);
    }

  /* Was anything read at all? */
  if (!gotNumber)
    return FALSE;

  if (target)
    *target = (isNegative) ? -((trio_intmax_t)number) : number;
  return TRUE;
}

/*************************************************************************
 * TrioReadChar
 */
TRIO_PRIVATE int
TrioReadChar
TRIO_ARGS4((self, target, flags, width),
           trio_class_t *self,
           char *target,
           trio_flags_t flags,
           int width)
{
  int i;
  char ch;
  trio_uintmax_t number;

  assert(VALID(self));
  assert(VALID(self->InStream));

  for (i = 0;
       (self->current != EOF) && (i < width);
       i++)
    {
      ch = (char)self->current;
      self->InStream(self, NULL);
      if ((flags & FLAGS_ALTERNATIVE) && (ch == CHAR_BACKSLASH))
        {
          switch (self->current)
            {
            case '\\': ch = '\\'; break;
            case 'a': ch = '\007'; break;
            case 'b': ch = '\b'; break;
            case 'f': ch = '\f'; break;
            case 'n': ch = '\n'; break;
            case 'r': ch = '\r'; break;
            case 't': ch = '\t'; break;
            case 'v': ch = '\v'; break;
            default:
              if (isdigit(self->current))
                {
                  /* Read octal number */
                  if (!TrioReadNumber(self, &number, 0, 3, BASE_OCTAL))
                    return 0;
                  ch = (char)number;
                }
              else if (trio_to_upper(self->current) == 'X')
                {
                  /* Read hexadecimal number */
                  self->InStream(self, NULL);
                  if (!TrioReadNumber(self, &number, 0, 2, BASE_HEX))
                    return 0;
                  ch = (char)number;
                }
              else
```

```
                {
                  ch = (char)self->current;
                }
              break;
            }
        }

      if (target)
        target[i] = ch;
    }
  return i + 1;
}

/*************************************************************************
 * TrioReadString
 */
TRIO_PRIVATE BOOLEAN_T
TrioReadString
TRIO_ARGS4((self, target, flags, width),
           trio_class_t *self,
           char *target,
           trio_flags_t flags,
           int width)
{
  int i;

  assert(VALID(self));
  assert(VALID(self->InStream));

  TrioSkipWhitespaces(self);

  /*
   * Continue until end of string is reached, a whitespace is encountered,
   * or width is exceeded
   */
  for (i = 0;
       ((width == NO_WIDTH) || (i < width)) &&
       (! ((self->current == EOF) || isspace(self->current)));
       i++)
    {
      if (TrioReadChar(self, (target ? &target[i] : 0), flags, 1) == 0)
        break; /* for */
    }
  if (target)
    target[i] = NIL;
  return TRUE;
}

/*************************************************************************
 * TrioReadWideChar
 */
#if TRIO_WIDECHAR
TRIO_PRIVATE int
TrioReadWideChar
TRIO_ARGS4((self, target, flags, width),
           trio_class_t *self,
           trio_wchar_t *target,
           trio_flags_t flags,
           int width)
{
  int i;
  int j;
  int size;
  int amount = 0;
  trio_wchar_t wch;
  char buffer[MB_LEN_MAX + 1];

  assert(VALID(self));
  assert(VALID(self->InStream));

  for (i = 0;
       (self->current != EOF) && (i < width);
       i++)
    {
      if (isascii(self->current))
        {
          if (TrioReadChar(self, buffer, flags, 1) == 0)
            return 0;
          buffer[1] = NIL;
        }
      else
        {
          /*
           * Collect a multibyte character, by enlarging buffer until
           * it contains a fully legal multibyte character, or the
           * buffer is full.
           */
          j = 0;
          do
            {
              buffer[j++] = (char)self->current;
              buffer[j] = NIL;
              self->InStream(self, NULL);
            }
          while ((j < (int)sizeof(buffer)) && (mblen(buffer, (size_t)j) != j));
        }
      if (target)
        {
```

```
          size = mbtowc(&wch, buffer, sizeof(buffer));
          if (size > 0)
            target[i] = wch;
        }
      amount += size;
      self->InStream(self, NULL);
    }
  return amount;
}
#endif /* TRIO_WIDECHAR */

/*************************************************************************
 * TrioReadWideString
 */
#if TRIO_WIDECHAR
TRIO_PRIVATE BOOLEAN_T
TrioReadWideString
TRIO_ARGS4((self, target, flags, width),
           trio_class_t *self,
           trio_wchar_t *target,
           trio_flags_t flags,
           int width)
{
  int i;
  int size;

  assert(VALID(self));
  assert(VALID(self->InStream));

  TrioSkipWhitespaces(self);

#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  (void)mblen(NULL, 0);
#endif

  /*
   * Continue until end of string is reached, a whitespace is encountered,
   * or width is exceeded
   */
  for (i = 0;
       ((width == NO_WIDTH) || (i < width)) &&
       (! ((self->current == EOF) || isspace(self->current)));
       )
    {
      size = TrioReadWideChar(self, &target[i], flags, 1);
      if (size == 0)
        break; /* for */

      i += size;
    }
  if (target)
    target[i] = WCONST('\0');
  return TRUE;
}
#endif /* TRIO_WIDECHAR */

/*************************************************************************
 * TrioReadGroup
 *
 * FIXME: characterclass does not work with multibyte characters
 */
TRIO_PRIVATE BOOLEAN_T
TrioReadGroup
TRIO_ARGS5((self, target, characterclass, flags, width),
           trio_class_t *self,
           char *target,
           int *characterclass,
           trio_flags_t flags,
           int width)
{
  int ch;
  int i;

  assert(VALID(self));
  assert(VALID(self->InStream));

  ch = self->current;
  for (i = 0;
       ((width == NO_WIDTH) || (i < width)) &&
       (! ((ch == EOF) ||
           (((flags & FLAGS_EXCLUDE) != 0) ^ (characterclass[ch] == 0))));
       i++)
    {
      if (target)
        target[i] = (char)ch;
      self->InStream(self, &ch);
    }

  if (target)
    target[i] = NIL;
  return TRUE;
}

/*************************************************************************
 * TrioReadDouble
 *
 * FIXME:
 *  add long double
```

```
 *  handle base
 */
TRIO_PRIVATE BOOLEAN_T
TrioReadDouble
TRIO_ARGS4((self, target, flags, width),
           trio_class_t *self,
           trio_pointer_t target,
           trio_flags_t flags,
           int width)
{
  int ch;
  char doubleString[512];
  int index = 0;
  int start;
  int j;
  BOOLEAN_T isHex = FALSE;

  doubleString[0] = 0;

  if ((width == NO_WIDTH) || (width > (int)sizeof(doubleString) - 1))
    width = sizeof(doubleString) - 1;

  TrioSkipWhitespaces(self);

  /*
   * Read entire double number from stream. trio_to_double requires
   * a string as input, but InStream can be anything, so we have to
   * collect all characters.
   */
  ch = self->current;
  if ((ch == '+') || (ch == '-'))
    {
      doubleString[index++] = (char)ch;
      self->InStream(self, &ch);
      width--;
    }

  start = index;
  switch (ch)
    {
    case 'n':
    case 'N':
      /* Not-a-number */
      if (index != 0)
        break;
      /* FALLTHROUGH */
    case 'i':
    case 'I':
      /* Infinity */
      while (isalpha(ch) && (index - start < width))
        {
          doubleString[index++] = (char)ch;
          self->InStream(self, &ch);
        }
      doubleString[index] = NIL;

      /* Case insensitive string comparison */
      if (trio_equal(&doubleString[start], INFINITE_UPPER) ||
          trio_equal(&doubleString[start], LONG_INFINITE_UPPER))
        {
          if (flags & FLAGS_LONGDOUBLE)
            {
              if ((start == 1) && (doubleString[0] == '-'))
                {
                  *((trio_long_double_t *)target) = trio_ninf();
                }
              else
                {
                  *((trio_long_double_t *)target) = trio_pinf();
                }
            }
          else
            {
              if ((start == 1) && (doubleString[0] == '-'))
                {
                  *((double *)target) = trio_ninf();
                }
              else
                {
                  *((double *)target) = trio_pinf();
                }
            }
          return TRUE;
        }
      if (trio_equal(doubleString, NAN_UPPER))
        {
          /* NaN must not have a preceeding + nor - */
          if (flags & FLAGS_LONGDOUBLE)
            {
              *((trio_long_double_t *)target) = trio_nan();
            }
          else
            {
              *((double *)target) = trio_nan();
            }
          return TRUE;
        }
      return FALSE;
```

```
    case '0':
      doubleString[index++] = (char)ch;
      self->InStream(self, &ch);
      if (trio_to_upper(ch) == 'X')
        {
          isHex = TRUE;
          doubleString[index++] = (char)ch;
          self->InStream(self, &ch);
        }
      break;

    default:
      break;
    }

  while ((ch != EOF) && (index - start < width))
    {
      /* Integer part */
      if (isHex ? isxdigit(ch) : isdigit(ch))
        {
          doubleString[index++] = (char)ch;
          self->InStream(self, &ch);
        }
      else if (flags & FLAGS_QUOTE)
        {
          /* Compare with thousands separator */
          for (j = 0; internalThousandSeparator[j] && self->current; j++)
            {
              if (internalThousandSeparator[j] != self->current)
                break;

              self->InStream(self, &ch);
            }
          if (internalThousandSeparator[j])
            break; /* Mismatch */
          else
            continue; /* Match */
        }
      else
        break; /* while */
    }
  if (ch == '.')
    {
      /* Decimal part */
      doubleString[index++] = (char)ch;
      self->InStream(self, &ch);
      while ((isHex ? isxdigit(ch) : isdigit(ch)) &&
             (index - start < width))
        {
          doubleString[index++] = (char)ch;
          self->InStream(self, &ch);
        }
      if (isHex ? (trio_to_upper(ch) == 'P') : (trio_to_upper(ch) == 'E'))
        {
          /* Exponent */
          doubleString[index++] = (char)ch;
          self->InStream(self, &ch);
          if ((ch == '+') || (ch == '-'))
            {
              doubleString[index++] = (char)ch;
              self->InStream(self, &ch);
            }
          while (isdigit(ch) && (index - start < width))
            {
              doubleString[index++] = (char)ch;
              self->InStream(self, &ch);
            }
        }
    }

  if ((index == start) || (*doubleString == NIL))
    return FALSE;

  doubleString[index] = 0;

  if (flags & FLAGS_LONGDOUBLE)
    {
      *((trio_long_double_t *)target) = trio_to_long_double(doubleString, NULL);
    }
  else
    {
      *((double *)target) = trio_to_double(doubleString, NULL);
    }
  return TRUE;
}

/*************************************************************************
 * TrioReadPointer
 */
TRIO_PRIVATE BOOLEAN_T
TrioReadPointer
TRIO_ARGS3((self, target, flags),
           trio_class_t *self,
           trio_pointer_t *target,
           trio_flags_t flags)
{
  trio_uintmax_t number;
```

```
  char buffer[sizeof(internalNullString)];

  flags |= (FLAGS_UNSIGNED | FLAGS_ALTERNATIVE | FLAGS_NILPADDING);

  if (TrioReadNumber(self,
                     &number,
                     flags,
                     POINTER_WIDTH,
                     BASE_HEX))
    {
      /*
       * The strange assignment of number is a workaround for a compiler
       * warning
       */
      if (target)
        *target = (char *)0 + number;
      return TRUE;
    }
  else if (TrioReadString(self,
                          (flags & FLAGS_IGNORE)
                          ? NULL
                          : buffer,
                          0,
                          sizeof(internalNullString) - 1))
    {
      if (trio_equal_case(buffer, internalNullString))
        {
          if (target)
            *target = NULL;
          return TRUE;
        }
    }
  return FALSE;
}

/*************************************************************************
 * TrioScanProcess
 */
TRIO_PRIVATE int
TrioScanProcess
TRIO_ARGS3((data, format, parameters),
           trio_class_t *data,
           TRIO_CONST char *format,
           trio_parameter_t *parameters)
{
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  int charlen;
  int cnt;
#endif
  int assignment;
  int ch;
  int index; /* Index of format string */
  int i; /* Index of current parameter */
  trio_flags_t flags;
  int width;
  int base;
  trio_pointer_t pointer;

  assignment = 0;
  i = 0;
  index = 0;
  data->InStream(data, &ch);

#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
  (void)mblen(NULL, 0);
#endif

  while (format[index])
    {
#if defined(TRIO_COMPILER_SUPPORTS_MULTIBYTE)
      if (! isascii(format[index]))
        {
          charlen = mblen(&format[index], MB_LEN_MAX);
          if (charlen != -1)
            {
              /* Compare multibyte characters in format string */
              for (cnt = 0; cnt < charlen - 1; cnt++)
                {
                  if (ch != format[index + cnt])
                    {
                      return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
                    }
                  data->InStream(data, &ch);
                }
              continue; /* while characters left in formatting string */
            }
        }
#endif /* TRIO_COMPILER_SUPPORTS_MULTIBYTE */

      if ((EOF == ch) && (parameters[i].type != FORMAT_COUNT))
        {
          return (assignment > 0) ? assignment : EOF;
        }

      if (CHAR_IDENTIFIER == format[index])
        {
          if (CHAR_IDENTIFIER == format[index + 1])
            {
```

```
              /* Two % in format matches one % in input stream */
              if (CHAR_IDENTIFIER == ch)
                {
                  data->InStream(data, &ch);
                  index += 2;
                  continue; /* while format chars left */
                }
              else
                return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
            }

          /* Skip the parameter entries */
          while (parameters[i].type == FORMAT_PARAMETER)
            i++;

          flags = parameters[i].flags;
          /* Find width */
          width = parameters[i].width;
          if (flags & FLAGS_WIDTH_PARAMETER)
            {
              /* Get width from parameter list */
              width = (int)parameters[width].data.number.as_signed;
            }
          /* Find base */
          base = parameters[i].base;
          if (flags & FLAGS_BASE_PARAMETER)
            {
              /* Get base from parameter list */
              base = (int)parameters[base].data.number.as_signed;
            }

          switch (parameters[i].type)
            {
            case FORMAT_INT:
              {
                trio_uintmax_t number;

                if (0 == base)
                  base = BASE_DECIMAL;

                if (!TrioReadNumber(data,
                                    &number,
                                    flags,
                                    width,
                                    base))
                  return assignment;

                if (!(flags & FLAGS_IGNORE))
                  {
                    assignment++;

                    pointer = parameters[i].data.pointer;
#if defined(QUALIFIER_SIZE_T) || defined(QUALIFIER_SIZE_T_UPPER)
                    if (flags & FLAGS_SIZE_T)
                      *(size_t *)pointer = (size_t)number;
                    else
#endif
#if defined(QUALIFIER_PTRDIFF_T)
                    if (flags & FLAGS_PTRDIFF_T)
                      *(ptrdiff_t *)pointer = (ptrdiff_t)number;
                    else
#endif
#if defined(QUALIFIER_INTMAX_T)
                    if (flags & FLAGS_INTMAX_T)
                      *(trio_intmax_t *)pointer = (trio_intmax_t)number;
                    else
#endif
                    if (flags & FLAGS_QUAD)
                      *(trio_ulonglong_t *)pointer = (trio_ulonglong_t)number;
                    else if (flags & FLAGS_LONG)
                      *(long int *)pointer = (long int)number;
                    else if (flags & FLAGS_SHORT)
                      *(short int *)pointer = (short int)number;
                    else
                      *(int *)pointer = (int)number;
                  }
              }
              break; /* FORMAT_INT */

            case FORMAT_STRING:
#if TRIO_WIDECHAR
              if (flags & FLAGS_WIDECHAR)
                {
                  if (!TrioReadWideString(data,
                                          (flags & FLAGS_IGNORE)
                                          ? NULL
                                          : parameters[i].data.wstring,
                                          flags,
                                          width))
                    return assignment;
                }
              else
#endif
                {
                  if (!TrioReadString(data,
                                      (flags & FLAGS_IGNORE)
                                      ? NULL
                                      : parameters[i].data.string,
```

```
                                flags,
                                width))
                    return assignment;
                }
              if (!(flags & FLAGS_IGNORE))
                assignment++;
              break; /* FORMAT_STRING */

            case FORMAT_DOUBLE:
              {
                trio_pointer_t pointer;

                if (flags & FLAGS_IGNORE)
                  {
                    pointer = NULL;
                  }
                else
                  {
                    pointer = (flags & FLAGS_LONGDOUBLE)
                      ? (trio_pointer_t)parameters[i].data.longdoublePointer
                      : (trio_pointer_t)parameters[i].data.doublePointer;
                  }
                if (!TrioReadDouble(data, pointer, flags, width))
                  {
                    return assignment;
                  }
                if (!(flags & FLAGS_IGNORE))
                  {
                    assignment++;
                  }
                break; /* FORMAT_DOUBLE */
              }
            case FORMAT_GROUP:
              {
                int characterclass[MAX_CHARACTER_CLASS + 1];
                int rc;

                /* Skip over modifiers */
                while (format[index] != SPECIFIER_GROUP)
                  {
                    index++;
                  }
                /* Skip over group specifier */
                index++;

                memset(characterclass, 0, sizeof(characterclass));
                rc = TrioGetCharacterClass(format,
                                           &index,
                                           &flags,
                                           characterclass);
                if (rc < 0)
                  return rc;

                if (!TrioReadGroup(data,
                                   (flags & FLAGS_IGNORE)
                                   ? NULL
                                   : parameters[i].data.string,
                                   characterclass,
                                   flags,
                                   parameters[i].width))
                  return assignment;
                if (!(flags & FLAGS_IGNORE))
                  assignment++;
              }
              break; /* FORMAT_GROUP */

            case FORMAT_COUNT:
              pointer = parameters[i].data.pointer;
              if (NULL != pointer)
                {
                  int count = data->committed;
                  if (ch != EOF)
                    count--; /* a character is read, but is not consumed yet */
#if defined(QUALIFIER_SIZE_T) || defined(QUALIFIER_SIZE_T_UPPER)
                  if (flags & FLAGS_SIZE_T)
                    *(size_t *)pointer = (size_t)count;
                  else
#endif
#if defined(QUALIFIER_PTRDIFF_T)
                  if (flags & FLAGS_PTRDIFF_T)
                    *(ptrdiff_t *)pointer = (ptrdiff_t)count;
                  else
#endif
#if defined(QUALIFIER_INTMAX_T)
                  if (flags & FLAGS_INTMAX_T)
                    *(trio_intmax_t *)pointer = (trio_intmax_t)count;
                  else
#endif
                  if (flags & FLAGS_QUAD)
                    {
                      *(trio_ulonglong_t *)pointer = (trio_ulonglong_t)count;
                    }
                  else if (flags & FLAGS_LONG)
                    {
                      *(long int *)pointer = (long int)count;
                    }
                  else if (flags & FLAGS_SHORT)
                    {
```

```
                    *(short int *)pointer = (short int)count;
                  }
                else
                  {
                    *(int *)pointer = (int)count;
                  }
              }
            break; /* FORMAT_COUNT */

          case FORMAT_CHAR:
#if TRIO_WIDECHAR
            if (flags & FLAGS_WIDECHAR)
              {
                if (TrioReadWideChar(data,
                                     (flags & FLAGS_IGNORE)
                                     ? NULL
                                     : parameters[i].data.wstring,
                                     flags,
                                     (width == NO_WIDTH) ? 1 : width) == 0)
                  return assignment;
              }
            else
#endif
              {
                if (TrioReadChar(data,
                                 (flags & FLAGS_IGNORE)
                                 ? NULL
                                 : parameters[i].data.string,
                                 flags,
                                 (width == NO_WIDTH) ? 1 : width) == 0)
                  return assignment;
              }
            if (!(flags & FLAGS_IGNORE))
              assignment++;
            break; /* FORMAT_CHAR */

          case FORMAT_POINTER:
            if (!TrioReadPointer(data,
                                 (flags & FLAGS_IGNORE)
                                 ? NULL
                                 : (trio_pointer_t *)parameters[i].data.pointer,
                                 flags))
              return assignment;
            if (!(flags & FLAGS_IGNORE))
              assignment++;
            break; /* FORMAT_POINTER */

          case FORMAT_PARAMETER:
            break; /* FORMAT_PARAMETER */

          default:
            return TRIO_ERROR_RETURN(TRIO_EINVAL, index);
          }
        ch = data->current;
        index = parameters[i].indexAfterSpecifier;
        i++;
      }
    else /* Not an % identifier */
      {
        if (isspace((int)format[index]))
          {
            /* Whitespaces may match any amount of whitespaces */
            ch = TrioSkipWhitespaces(data);
          }
        else if (ch == format[index])
          {
            data->InStream(data, &ch);
          }
        else
          return assignment;

        index++;
      }
  }
  return assignment;
}

/*************************************************************************
 * TrioScan
 */
TRIO_PRIVATE int
TrioScan
TRIO_ARGS6((source, sourceSize, InStream, format, arglist, argarray),
           trio_pointer_t source,
           size_t sourceSize,
           void (*InStream) TRIO_PROTO((trio_class_t *, int *)),
           TRIO_CONST char *format,
           TRIO_VA_LIST_PTR arglist,
           trio_pointer_t *argarray)
{
  int status;
  trio_parameter_t parameters[MAX_PARAMETERS];
  trio_class_t data;

  assert(VALID(InStream));
  assert(VALID(format));

  memset(&data, 0, sizeof(data));
```

```
  data.InStream = InStream;
  data.location = (trio_pointer_t)source;
  data.max = sourceSize;
  data.error = 0;

#if defined(USE_LOCALE)
  if (NULL == internalLocaleValues)
    {
      TrioSetLocale();
    }
#endif

  status = TrioParse(TYPE_SCAN, format, parameters, arglist, argarray);
  if (status < 0)
    return status;

  status = TrioScanProcess(&data, format, parameters);
  if (data.error != 0)
    {
      status = data.error;
    }
  return status;
}

/*************************************************************************
 * TrioInStreamFile
 */
TRIO_PRIVATE void
TrioInStreamFile
TRIO_ARGS2((self, intPointer),
           trio_class_t *self,
           int *intPointer)
{
  FILE *file;

  assert(VALID(self));
  assert(VALID(self->location));
  assert(VALID(file));

  file = (FILE *)self->location;

  self->current = fgetc(file);
  if (self->current == EOF)
    {
      self->error = (ferror(file))
        ? TRIO_ERROR_RETURN(TRIO_ERRNO, 0)
        : TRIO_ERROR_RETURN(TRIO_EOF, 0);
    }
  else
    {
      self->processed++;
      self->committed++;
    }

  if (VALID(intPointer))
    {
      *intPointer = self->current;
    }
}

/*************************************************************************
 * TrioInStreamFileDescriptor
 */
TRIO_PRIVATE void
TrioInStreamFileDescriptor
TRIO_ARGS2((self, intPointer),
           trio_class_t *self,
           int *intPointer)
{
  int fd;
  int size;
  unsigned char input;

  assert(VALID(self));
  assert(VALID(self->location));

  fd = *((int *)self->location);

  size = read(fd, &input, sizeof(char));
  if (size == -1)
    {
      self->error = TRIO_ERROR_RETURN(TRIO_ERRNO, 0);
      self->current = EOF;
    }
  else
    {
      self->current = (size == 0) ? EOF : input;
    }
  if (self->current != EOF)
    {
      self->committed++;
      self->processed++;
    }

  if (VALID(intPointer))
    {
      *intPointer = self->current;
    }
```

```
}

/*************************************************************************
 * TrioInStreamCustom
 */
TRIO_PRIVATE void
TrioInStreamCustom
TRIO_ARGS2((self, intPointer),
           trio_class_t *self,
           int *intPointer)
{
  trio_custom_t *data;

  assert(VALID(self));
  assert(VALID(self->location));

  data = (trio_custom_t *)self->location;

  self->current = (data->stream.in == NULL)
    ? NIL
    : (data->stream.in)(data->closure);

  if (self->current == NIL)
    {
      self->current = EOF;
    }
  else
    {
      self->processed++;
      self->committed++;
    }

  if (VALID(intPointer))
    {
      *intPointer = self->current;
    }
}

/*************************************************************************
 * TrioInStreamString
 */
TRIO_PRIVATE void
TrioInStreamString
TRIO_ARGS2((self, intPointer),
           trio_class_t *self,
           int *intPointer)
{
  unsigned char **buffer;

  assert(VALID(self));
  assert(VALID(self->location));

  buffer = (unsigned char **)self->location;
  self->current = (*buffer)[0];
  if (self->current == NIL)
    {
      self->current = EOF;
    }
  else
    {
      (*buffer)++;
      self->processed++;
      self->committed++;
    }

  if (VALID(intPointer))
    {
      *intPointer = self->current;
    }
}

/*************************************************************************
 *
 * Formatted scanning functions
 *
 *************************************************************************/

#if defined(TRIO_DOCUMENTATION)
# include "doc/doc_scanf.h"
#endif
/** @addtogroup Scanf
    @{
*/

/*************************************************************************
 * scanf
 */

/**
   Scan characters from standard input stream.

   @param format Formatting string.
   @param ... Arguments.
   @return Number of scanned characters.
 */
TRIO_PUBLIC int
trio_scanf
TRIO_VARGS2((format, va_alist),
```

```
           TRIO_CONST char *format,
           TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioScan((trio_pointer_t)stdin, 0,
                    TrioInStreamFile,
                    format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vscanf
TRIO_ARGS2((format, args),
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(format));

  return TrioScan((trio_pointer_t)stdin, 0,
                  TrioInStreamFile,
                  format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_scanfv
TRIO_ARGS2((format, args),
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  assert(VALID(format));

  return TrioScan((trio_pointer_t)stdin, 0,
                  TrioInStreamFile,
                  format, NULL, args);
}

/*************************************************************************
 * fscanf
 */
TRIO_PUBLIC int
trio_fscanf
TRIO_VARGS3((file, format, va_alist),
            FILE *file,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(file));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioScan((trio_pointer_t)file, 0,
                    TrioInStreamFile,
                    format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vfscanf
TRIO_ARGS3((file, format, args),
           FILE *file,
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(file));
  assert(VALID(format));

  return TrioScan((trio_pointer_t)file, 0,
                  TrioInStreamFile,
                  format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_fscanfv
TRIO_ARGS3((file, format, args),
           FILE *file,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  assert(VALID(file));
  assert(VALID(format));

  return TrioScan((trio_pointer_t)file, 0,
                  TrioInStreamFile,
                  format, NULL, args);
}

/*************************************************************************
 * dscanf
```

```
 */
TRIO_PUBLIC int
trio_dscanf
TRIO_VARGS3((fd, format, va_alist),
            int fd,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioScan((trio_pointer_t)&fd, 0,
                    TrioInStreamFileDescriptor,
                    format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vdscanf
TRIO_ARGS3((fd, format, args),
           int fd,
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(format));

  return TrioScan((trio_pointer_t)&fd, 0,
                  TrioInStreamFileDescriptor,
                  format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_dscanfv
TRIO_ARGS3((fd, format, args),
           int fd,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  assert(VALID(format));

  return TrioScan((trio_pointer_t)&fd, 0,
                  TrioInStreamFileDescriptor,
                  format, NULL, args);
}

/*************************************************************************
 * cscanf
 */
TRIO_PUBLIC int
trio_cscanf
TRIO_VARGS4((stream, closure, format, va_alist),
            trio_instream_t stream,
            trio_pointer_t closure,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;
  trio_custom_t data;

  assert(VALID(stream));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  data.stream.in = stream;
  data.closure = closure;
  status = TrioScan(&data, 0, TrioInStreamCustom, format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vcscanf
TRIO_ARGS4((stream, closure, format, args),
           trio_instream_t stream,
           trio_pointer_t closure,
           TRIO_CONST char *format,
           va_list args)
{
  trio_custom_t data;

  assert(VALID(stream));
  assert(VALID(format));

  data.stream.in = stream;
  data.closure = closure;
  return TrioScan(&data, 0, TrioInStreamCustom, format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_cscanfv
TRIO_ARGS4((stream, closure, format, args),
           trio_instream_t stream,
           trio_pointer_t closure,
```

```
          TRIO_CONST char *format,
          trio_pointer_t *args)
{
  trio_custom_t data;

  assert(VALID(stream));
  assert(VALID(format));

  data.stream.in = stream;
  data.closure = closure;
  return TrioScan(&data, 0, TrioInStreamCustom, format, NULL, args);
}

/*************************************************************************
 * sscanf
 */
TRIO_PUBLIC int
trio_sscanf
TRIO_VARGS3((buffer, format, va_alist),
            TRIO_CONST char *buffer,
            TRIO_CONST char *format,
            TRIO_VA_DECL)
{
  int status;
  va_list args;

  assert(VALID(buffer));
  assert(VALID(format));

  TRIO_VA_START(args, format);
  status = TrioScan((trio_pointer_t)&buffer, 0,
                    TrioInStreamString,
                    format, TRIO_VA_LIST_ADDR(args), NULL);
  TRIO_VA_END(args);
  return status;
}

TRIO_PUBLIC int
trio_vsscanf
TRIO_ARGS3((buffer, format, args),
           TRIO_CONST char *buffer,
           TRIO_CONST char *format,
           va_list args)
{
  assert(VALID(buffer));
  assert(VALID(format));

  return TrioScan((trio_pointer_t)&buffer, 0,
                  TrioInStreamString,
                  format, TRIO_VA_LIST_ADDR(args), NULL);
}

TRIO_PUBLIC int
trio_sscanfv
TRIO_ARGS3((buffer, format, args),
           TRIO_CONST char *buffer,
           TRIO_CONST char *format,
           trio_pointer_t *args)
{
  assert(VALID(buffer));
  assert(VALID(format));

  return TrioScan((trio_pointer_t)&buffer, 0,
                  TrioInStreamString,
                  format, NULL, args);
}

/** @} End of Scanf documentation module */

/*************************************************************************
 * trio_strerror
 */
TRIO_PUBLIC TRIO_CONST char *
trio_strerror
TRIO_ARGS1((errorcode),
           int errorcode)
{
  /* Textual versions of the error codes */
  switch (TRIO_ERROR_CODE(errorcode))
    {
    case TRIO_EOF:
      return "End of file";
    case TRIO_EINVAL:
      return "Invalid argument";
    case TRIO_ETOOMANY:
      return "Too many arguments";
    case TRIO_EDBLREF:
      return "Double reference";
    case TRIO_EGAP:
      return "Reference gap";
    case TRIO_ENOMEM:
      return "Out of memory";
    case TRIO_ERANGE:
      return "Invalid range";
    case TRIO_ECUSTOM:
      return "Custom error";
    default:
      return "Unknown";
```

```
    }
}
```

Notice for package(s)

popt

```
Copyright (c) 1998  Red Hat Software

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from the X Consortium.
```

Notice for package(s)

liburcu

```
Userspace RCU library licensing
Mathieu Desnoyers
September 3, 2012


* LGPLv2.1

The library part is distributed under LGPLv2.1 or later. See lgpl-2.1.txt for
license details. Refer to the individual file headers for details.

LGPL-compatible source code can statically use the library header using :

#define _LGPL_SOURCE
#include <urcu.h>

Dynamic-only linking with the LGPL library is used if _LGPL_SOURCE is not
defined. It permits relinking with newer versions of the library, which is
required by the LGPL license.

See lgpl-relicensing.txt for details.


* MIT-style license :

xchg() primitive has been rewritten from scratch starting from atomic_ops 1.2
which has a MIT-style license that is intended to allow use in both free and
proprietary software:
        http://www.hpl.hp.com/research/linux/atomic_ops/LICENSING.txt
        http://www.hpl.hp.com/personal/Hans_Boehm/gc/gc_source/

This MIT-style license (BSD like) apply to:

uatomic/gcc.h
uatomic/unknown.h
uatomic/generic.h
uatomic/sparc64.h
uatomic/arm.h
uatomic/ppc.h
uatomic/x86.h
uatomic.h

MIT/X11 (BSD like) license apply to:

compiler.h
arch/s390.h
uatomic/alpha.h
uatomic/mips.h
uatomic/s390.h
system.h
```

```
* GPLv2

Library test code is distributed under the GPLv2 license. See gpl-2.0.txt for
license details. See headers of individual files under tests/ for details.


* GPLv3 (or later)

The following build-related macro is under GPLv3 (or later):

m4/ax_tls.m4
```

Notice for package(s)

liburcu

```
#ifndef _URCU_H
#define _URCU_H

/*
 * urcu.h
 *
 * Userspace RCU header
 *
 * Copyright (c) 2009 Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
 * Copyright (c) 2009 Paul E. McKenney, IBM Corporation.
 *
 * LGPL-compatible code should include this header with :
 *
 * #define _LGPL_SOURCE
 * #include <urcu.h>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 *
 * IBM's contributions to this file may be relicensed under LGPLv2 or later.
 */

#include <stdlib.h>
#include <pthread.h>

/*
 * See urcu-pointer.h and urcu/static/urcu-pointer.h for pointer
 * publication headers.
 */
#include <urcu-pointer.h>

#ifdef __cplusplus
extern "C" {
#endif

#include <urcu/map/urcu.h>

/*
 * Important !
 *
 * Each thread containing read-side critical sections must be registered
 * with rcu_register_thread_mb() before calling rcu_read_lock_mb().
 * rcu_unregister_thread_mb() should be called before the thread exits.
 */

#ifdef _LGPL_SOURCE

#include <urcu/static/urcu.h>

/*
 * Mappings for static use of the userspace RCU library.
 * Should only be used in LGPL-compatible code.
 */

/*
 * rcu_read_lock()
 * rcu_read_unlock()
 *
 * Mark the beginning and end of a read-side critical section.
 * DON'T FORGET TO USE RCU_REGISTER/UNREGISTER_THREAD() FOR EACH THREAD WITH
 * READ-SIDE CRITICAL SECTION.
 */
#ifdef RCU_MEMBARRIER
#define rcu_read_lock_memb              _rcu_read_lock
```

```
#define rcu_read_unlock_memb            _rcu_read_unlock
#define rcu_read_ongoing_memb           _rcu_read_ongoing
#elif defined(RCU_SIGNAL)
#define rcu_read_lock_sig               _rcu_read_lock
#define rcu_read_unlock_sig             _rcu_read_unlock
#define rcu_read_ongoing_sig            _rcu_read_ongoing
#elif defined(RCU_MB)
#define rcu_read_lock_mb                _rcu_read_lock
#define rcu_read_unlock_mb              _rcu_read_unlock
#define rcu_read_ongoing_mb             _rcu_read_ongoing
#endif

#else /* !_LGPL_SOURCE */

/*
 * library wrappers to be used by non-LGPL compatible source code.
 * See LGPL-only urcu/static/urcu-pointer.h for documentation.
 */

extern void rcu_read_lock(void);
extern void rcu_read_unlock(void);
extern int rcu_read_ongoing(void);

#endif /* !_LGPL_SOURCE */

extern void synchronize_rcu(void);

/*
 * Reader thread registration.
 */
extern void rcu_register_thread(void);
extern void rcu_unregister_thread(void);

/*
 * Explicit rcu initialization, for "early" use within library constructors.
 */
extern void rcu_init(void);

/*
 * Q.S. reporting are no-ops for these URCU flavors.
 */
static inline void rcu_quiescent_state(void)
{
}

static inline void rcu_thread_offline(void)
{
}

static inline void rcu_thread_online(void)
{
}

#ifdef __cplusplus
}
#endif

#include <urcu-call-rcu.h>
#include <urcu-defer.h>
#include <urcu-flavor.h>

#endif /* _URCU_H */
```

Notice for package(s)

liburcu

```
#ifndef _URCU_ARCH_UATOMIC_X86_H
#define _URCU_ARCH_UATOMIC_X86_H

/*
 * Copyright (c) 1991-1994 by Xerox Corporation.  All rights reserved.
 * Copyright (c) 1996-1999 by Silicon Graphics.  All rights reserved.
 * Copyright (c) 1999-2004 Hewlett-Packard Development Company, L.P.
 * Copyright (c) 2009      Mathieu Desnoyers
 *
 * THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED
 * OR IMPLIED.  ANY USE IS AT YOUR OWN RISK.
 *
 * Permission is hereby granted to use or copy this program
 * for any purpose,  provided the above notices are retained on all copies.
 * Permission to modify the code and to distribute modified code is granted,
 * provided the above notices are retained, and a notice that the code was
 * modified is included with the above copyright notice.
 *
 * Code inspired from libuatomic_ops-1.2, inherited in part from the
 * Boehm-Demers-Weiser conservative garbage collector.
 */

#include <urcu/compiler.h>
#include <urcu/system.h>
```

```
#define UATOMIC_HAS_ATOMIC_BYTE
#define UATOMIC_HAS_ATOMIC_SHORT

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Derived from AO_compare_and_swap() and AO_test_and_set_full().
 */

struct __uatomic_dummy {
        unsigned long v[10];
};
#define __hp(x) ((struct __uatomic_dummy *)(x))

#define _uatomic_set(addr, v)   ((void) CMM_STORE_SHARED(*(addr), (v)))

/* cmpxchg */

static inline __attribute__((always_inline))
unsigned long __uatomic_cmpxchg(void *addr, unsigned long old,
                                unsigned long _new, int len)
{
        switch (len) {
        case 1:
        {
                unsigned char result = old;

                __asm__ __volatile__(
                "lock; cmpxchgb %2, %1"
                        : "+a"(result), "+m"(*__hp(addr))
                        : "q"((unsigned char)_new)
                        : "memory");
                return result;
        }
        case 2:
        {
                unsigned short result = old;

                __asm__ __volatile__(
                "lock; cmpxchgw %2, %1"
                        : "+a"(result), "+m"(*__hp(addr))
                        : "r"((unsigned short)_new)
                        : "memory");
                return result;
        }
        case 4:
        {
                unsigned int result = old;

                __asm__ __volatile__(
                "lock; cmpxchgl %2, %1"
                        : "+a"(result), "+m"(*__hp(addr))
                        : "r"((unsigned int)_new)
                        : "memory");
                return result;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                unsigned long result = old;

                __asm__ __volatile__(
                "lock; cmpxchgq %2, %1"
                        : "+a"(result), "+m"(*__hp(addr))
                        : "r"((unsigned long)_new)
                        : "memory");
                return result;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return 0;
}

#define _uatomic_cmpxchg(addr, old, _new)                                   \
        ((__typeof__(*(addr))) __uatomic_cmpxchg((addr),                    \
                                                 caa_cast_long_keep_sign(old), \
                                                 caa_cast_long_keep_sign(_new),\
                                                 sizeof(*(addr))))

/* xchg */

static inline __attribute__((always_inline))
unsigned long __uatomic_exchange(void *addr, unsigned long val, int len)
{
        /* Note: the "xchg" instruction does not need a "lock" prefix. */
        switch (len) {
        case 1:
        {
                unsigned char result;
                __asm__ __volatile__(
```

```
                "xchgb %0, %1"
                        : "=q"(result), "+m"(*__hp(addr))
                        : "0" ((unsigned char)val)
                        : "memory");
                return result;
        }
        case 2:
        {
                unsigned short result;
                __asm__ __volatile__(
                "xchgw %0, %1"
                        : "=r"(result), "+m"(*__hp(addr))
                        : "0" ((unsigned short)val)
                        : "memory");
                return result;
        }
        case 4:
        {
                unsigned int result;
                __asm__ __volatile__(
                "xchgl %0, %1"
                        : "=r"(result), "+m"(*__hp(addr))
                        : "0" ((unsigned int)val)
                        : "memory");
                return result;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                unsigned long result;
                __asm__ __volatile__(
                "xchgq %0, %1"
                        : "=r"(result), "+m"(*__hp(addr))
                        : "0" ((unsigned long)val)
                        : "memory");
                return result;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return 0;
}

#define _uatomic_xchg(addr, v)                                                \
        ((__typeof__(*(addr))) __uatomic_exchange((addr),                     \
                                                  caa_cast_long_keep_sign(v),   \
                                                  sizeof(*(addr))))

/* uatomic_add_return */

static inline __attribute__((always_inline))
unsigned long __uatomic_add_return(void *addr, unsigned long val,
                                   int len)
{
        switch (len) {
        case 1:
        {
                unsigned char result = val;

                __asm__ __volatile__(
                "lock; xaddb %1, %0"
                        : "+m"(*__hp(addr)), "+q" (result)
                        :
                        : "memory");
                return result + (unsigned char)val;
        }
        case 2:
        {
                unsigned short result = val;

                __asm__ __volatile__(
                "lock; xaddw %1, %0"
                        : "+m"(*__hp(addr)), "+r" (result)
                        :
                        : "memory");
                return result + (unsigned short)val;
        }
        case 4:
        {
                unsigned int result = val;

                __asm__ __volatile__(
                "lock; xaddl %1, %0"
                        : "+m"(*__hp(addr)), "+r" (result)
                        :
                        : "memory");
                return result + (unsigned int)val;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                unsigned long result = val;

                __asm__ __volatile__(
```

```
                "lock; xaddq %1, %0"
                        : "+m"(*__hp(addr)), "+r" (result)
                        :
                        : "memory");
                return result + (unsigned long)val;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return 0;
}

#define _uatomic_add_return(addr, v)                                            \
        ((__typeof__(*(addr))) __uatomic_add_return((addr),                     \
                                                    caa_cast_long_keep_sign(v), \
                                                    sizeof(*(addr))))

/* uatomic_and */

static inline __attribute__((always_inline))
void __uatomic_and(void *addr, unsigned long val, int len)
{
        switch (len) {
        case 1:
        {
                __asm__ __volatile__(
                "lock; andb %1, %0"
                        : "=m"(*__hp(addr))
                        : "iq" ((unsigned char)val)
                        : "memory");
                return;
        }
        case 2:
        {
                __asm__ __volatile__(
                "lock; andw %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned short)val)
                        : "memory");
                return;
        }
        case 4:
        {
                __asm__ __volatile__(
                "lock; andl %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned int)val)
                        : "memory");
                return;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                __asm__ __volatile__(
                "lock; andq %1, %0"
                        : "=m"(*__hp(addr))
                        : "er" ((unsigned long)val)
                        : "memory");
                return;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return;
}

#define _uatomic_and(addr, v)                                                   \
        (__uatomic_and((addr), caa_cast_long_keep_sign(v), sizeof(*(addr))))

/* uatomic_or */

static inline __attribute__((always_inline))
void __uatomic_or(void *addr, unsigned long val, int len)
{
        switch (len) {
        case 1:
        {
                __asm__ __volatile__(
                "lock; orb %1, %0"
                        : "=m"(*__hp(addr))
                        : "iq" ((unsigned char)val)
                        : "memory");
                return;
        }
        case 2:
        {
                __asm__ __volatile__(
                "lock; orw %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned short)val)
```

```
                        : "memory");
                return;
        }
        case 4:
        {
                __asm__ __volatile__(
                "lock; orl %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned int)val)
                        : "memory");
                return;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                __asm__ __volatile__(
                "lock; orq %1, %0"
                        : "=m"(*__hp(addr))
                        : "er" ((unsigned long)val)
                        : "memory");
                return;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return;
}

#define _uatomic_or(addr, v)                                                \
        (__uatomic_or((addr), caa_cast_long_keep_sign(v), sizeof(*(addr))))

/* uatomic_add */

static inline __attribute__((always_inline))
void __uatomic_add(void *addr, unsigned long val, int len)
{
        switch (len) {
        case 1:
        {
                __asm__ __volatile__(
                "lock; addb %1, %0"
                        : "=m"(*__hp(addr))
                        : "iq" ((unsigned char)val)
                        : "memory");
                return;
        }
        case 2:
        {
                __asm__ __volatile__(
                "lock; addw %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned short)val)
                        : "memory");
                return;
        }
        case 4:
        {
                __asm__ __volatile__(
                "lock; addl %1, %0"
                        : "=m"(*__hp(addr))
                        : "ir" ((unsigned int)val)
                        : "memory");
                return;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                __asm__ __volatile__(
                "lock; addq %1, %0"
                        : "=m"(*__hp(addr))
                        : "er" ((unsigned long)val)
                        : "memory");
                return;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return;
}

#define _uatomic_add(addr, v)                                               \
        (__uatomic_add((addr), caa_cast_long_keep_sign(v), sizeof(*(addr))))


/* uatomic_inc */

static inline __attribute__((always_inline))
void __uatomic_inc(void *addr, int len)
{
        switch (len) {
```

```
        case 1:
        {
                __asm__ __volatile__(
                "lock; incb %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
        case 2:
        {
                __asm__ __volatile__(
                "lock; incw %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
        case 4:
        {
                __asm__ __volatile__(
                "lock; incl %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                __asm__ __volatile__(
                "lock; incq %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
#endif
        }
        /* generate an illegal instruction. Cannot catch this with linker tricks
         * when optimizations are disabled. */
        __asm__ __volatile__("ud2");
        return;
}

#define _uatomic_inc(addr)      (__uatomic_inc((addr), sizeof(*(addr))))

/* uatomic_dec */

static inline __attribute__((always_inline))
void __uatomic_dec(void *addr, int len)
{
        switch (len) {
        case 1:
        {
                __asm__ __volatile__(
                "lock; decb %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
        case 2:
        {
                __asm__ __volatile__(
                "lock; decw %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
        case 4:
        {
                __asm__ __volatile__(
                "lock; decl %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
#if (CAA_BITS_PER_LONG == 64)
        case 8:
        {
                __asm__ __volatile__(
                "lock; decq %0"
                        : "=m"(*__hp(addr))
                        :
                        : "memory");
                return;
        }
#endif
        }
        /*
         * generate an illegal instruction. Cannot catch this with
         * linker tricks when optimizations are disabled.
         */
        __asm__ __volatile__("ud2");
        return;
```

```
}

#define _uatomic_dec(addr)	(__uatomic_dec((addr), sizeof(*(addr))))

#if ((CAA_BITS_PER_LONG != 64) && defined(CONFIG_RCU_COMPAT_ARCH))
extern int __rcu_cas_avail;
extern int __rcu_cas_init(void);

#define UATOMIC_COMPAT(insn)                                                  \
	((caa_likely(__rcu_cas_avail > 0))                                            \
	? (_uatomic_##insn)                                                   \
		: ((caa_unlikely(__rcu_cas_avail < 0)                         \
			? ((__rcu_cas_init() > 0)                             \
				? (_uatomic_##insn)                           \
				: (compat_uatomic_##insn))                    \
			: (compat_uatomic_##insn))))

/*
 * We leave the return value so we don't break the ABI, but remove the
 * return value from the API.
 */
extern unsigned long _compat_uatomic_set(void *addr,
                                         unsigned long _new, int len);
#define compat_uatomic_set(addr, _new)                                        \
	((void) _compat_uatomic_set((addr),                                   \
				    caa_cast_long_keep_sign(_new),            \
				    sizeof(*(addr))))


extern unsigned long _compat_uatomic_xchg(void *addr,
                                          unsigned long _new, int len);
#define compat_uatomic_xchg(addr, _new)                                       \
	((__typeof__(*(addr))) _compat_uatomic_xchg((addr),                   \
						    caa_cast_long_keep_sign(_new), \
						    sizeof(*(addr))))

extern unsigned long _compat_uatomic_cmpxchg(void *addr, unsigned long old,
                                             unsigned long _new, int len);
#define compat_uatomic_cmpxchg(addr, old, _new)                               \
	((__typeof__(*(addr))) _compat_uatomic_cmpxchg((addr),                \
						       caa_cast_long_keep_sign(old),  \
						       caa_cast_long_keep_sign(_new), \
						       sizeof(*(addr))))

extern void _compat_uatomic_and(void *addr, unsigned long _new, int len);
#define compat_uatomic_and(addr, v)                                           \
	(_compat_uatomic_and((addr),                                          \
			     caa_cast_long_keep_sign(v),                      \
			     sizeof(*(addr))))

extern void _compat_uatomic_or(void *addr, unsigned long _new, int len);
#define compat_uatomic_or(addr, v)                                            \
	(_compat_uatomic_or((addr),                                           \
			    caa_cast_long_keep_sign(v),                       \
			    sizeof(*(addr))))

extern unsigned long _compat_uatomic_add_return(void *addr,
                                                unsigned long _new, int len);
#define compat_uatomic_add_return(addr, v)                                    \
	((__typeof__(*(addr))) _compat_uatomic_add_return((addr),             \
						    caa_cast_long_keep_sign(v), \
						    sizeof(*(addr))))

#define compat_uatomic_add(addr, v)                                           \
		((void)compat_uatomic_add_return((addr), (v)))
#define compat_uatomic_inc(addr)                                              \
		(compat_uatomic_add((addr), 1))
#define compat_uatomic_dec(addr)                                              \
		(compat_uatomic_add((addr), -1))

#else
#define UATOMIC_COMPAT(insn)	(_uatomic_##insn)
#endif

/* Read is atomic even in compat mode */
#define uatomic_set(addr, v)			\
		UATOMIC_COMPAT(set(addr, v))

#define uatomic_cmpxchg(addr, old, _new)	\
		UATOMIC_COMPAT(cmpxchg(addr, old, _new))
#define uatomic_xchg(addr, v)			\
		UATOMIC_COMPAT(xchg(addr, v))

#define uatomic_and(addr, v)		\
		UATOMIC_COMPAT(and(addr, v))
#define cmm_smp_mb__before_uatomic_and()	cmm_barrier()
#define cmm_smp_mb__after_uatomic_and()		cmm_barrier()

#define uatomic_or(addr, v)		\
		UATOMIC_COMPAT(or(addr, v))
#define cmm_smp_mb__before_uatomic_or()		cmm_barrier()
#define cmm_smp_mb__after_uatomic_or()		cmm_barrier()

#define uatomic_add_return(addr, v)		\
		UATOMIC_COMPAT(add_return(addr, v))

#define uatomic_add(addr, v)	UATOMIC_COMPAT(add(addr, v))
#define cmm_smp_mb__before_uatomic_add()	cmm_barrier()
```

```
#define cmm_smp_mb__after_uatomic_add()         cmm_barrier()

#define uatomic_inc(addr)       UATOMIC_COMPAT(inc(addr))
#define cmm_smp_mb__before_uatomic_inc()        cmm_barrier()
#define cmm_smp_mb__after_uatomic_inc()         cmm_barrier()

#define uatomic_dec(addr)       UATOMIC_COMPAT(dec(addr))
#define cmm_smp_mb__before_uatomic_dec()        cmm_barrier()
#define cmm_smp_mb__after_uatomic_dec()         cmm_barrier()

#ifdef __cplusplus
}
#endif

#include <urcu/uatomic/generic.h>

#endif /* _URCU_ARCH_UATOMIC_X86_H */
```

Notice for package(s)

lttng-ust

```
LTTng UST - Userspace Tracer

Copyright (C) 2009-2012 Mathieu Desnoyers <mathieu.desnoyers@efficios.com>
Copyright (C) 2010 Pierre-Marc Fournier, Nils Carlson, David Goulet and others.


* This library is licensed under the LGPL v2.1 license, except when
  specified otherwise.

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; only
version 2.1 of the License.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301 USA


* System headers are MIT-licensed (BSD style):

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.


* liblttng-ust-ctl/ustctl.c is GPL v2. It is only used by the
  lttng-sessiond program.

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; version 2 of the License only.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Notice for package(s)

lttng-ust

```
/*      $OpenBSD: snprintf.c,v 1.16 2009/10/22 01:23:16 guenther Exp $ */
/*-
 * Copyright (c) 1990, 1993
 *      The Regents of the University of California.  All rights reserved.
 *
 * This code is derived from software contributed to Berkeley by
```

```
 * Chris Torek.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

#include <limits.h>
#include <stdio.h>
#include <string.h>
#include <stdarg.h>
#include "local.h"
#include "ust_snprintf.h"

#define DUMMY_LEN       1

int ust_safe_vsnprintf(char *str, size_t n, const char *fmt, va_list ap)
{
        int ret;
        char dummy[DUMMY_LEN];
        LTTNG_UST_LFILE f;
        struct __lttng_ust_sfileext fext;

        /* While snprintf(3) specifies size_t stdio uses an int internally */
        if (n > INT_MAX)
                n = INT_MAX;
        /* Stdio internals do not deal correctly with zero length buffer */
        if (n == 0) {
                str = dummy;
                n = DUMMY_LEN;
        }
        _FILEEXT_SETUP(&f, &fext);
        f._file = -1;
        f._flags = __SWR | __SSTR;
        f._bf._base = f._p = (unsigned char *)str;
        f._bf._size = f._w = n - 1;
        ret = ust_safe_vfprintf(&f, fmt, ap);
        *f._p = '\0';
        return (ret);
}

int ust_safe_snprintf(char *str, size_t n, const char *fmt, ...)
{
        va_list ap;
        int ret;

        va_start(ap, fmt);
        ret = ust_safe_vsnprintf(str, n, fmt, ap);
        va_end(ap);

        return ret;
}
```

Notice for package(s)

lttng-ust

```
/*-
 * Copyright (c) 1990 The Regents of the University of California.
 * All rights reserved.
 *
 * This code is derived from software contributed to Berkeley by
 * Chris Torek.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
```

```
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

#ifndef UST_SNPRINTF_VARIOUS_H
#define UST_SNPRINTF_VARIOUS_H

#include <stdarg.h>

struct __lttng_ust_sbuf {
        unsigned char *_base;
        int     _size;
};

/*
 * stdio state variables.
 *
 * The following always hold:
 *
 *      if (_flags&(__SLBF|__SWR)) == (__SLBF|__SWR),
 *              _lbfsize is -_bf._size, else _lbfsize is 0
 *      if _flags&__SRD, _w is 0
 *      if _flags&__SWR, _r is 0
 *
 * This ensures that the getc and putc macros (or inline functions) never
 * try to write or read from a file that is in `read' or `write' mode.
 * (Moreover, they can, and do, automatically switch from read mode to
 * write mode, and back, on "r+" and "w+" files.)
 *
 * _lbfsize is used only to make the inline line-buffered output stream
 * code as compact as possible.
 *
 * _ub, _up, and _ur are used when ungetc() pushes back more characters
 * than fit in the current _bf, or when ungetc() pushes back a character
 * that does not match the previous one in _bf.  When this happens,
 * _ub._base becomes non-nil (i.e., a stream has ungetc() data iff
 * _ub._base!=NULL) and _up and _ur save the current values of _p and _r.
 */
typedef struct __lttng_ust_sFILE {
        unsigned char *_p;      /* current position in (some) buffer */
        int     _r;             /* read space left for getc() */
        int     _w;             /* write space left for putc() */
        short   _flags;         /* flags, below; this FILE is free if 0 */
        short   _file;          /* fileno, if Unix descriptor, else -1 */
        struct  __lttng_ust_sbuf _bf;     /* the buffer (at least 1 byte, if !NULL) */
        int     _lbfsize;       /* 0 or -_bf._size, for inline putc */

        /* operations */
        void    *_cookie;       /* cookie passed to io functions */
        int     (*_close)(void *);
        int     (*_read)(void *, char *, int);
        fpos_t  (*_seek)(void *, fpos_t, int);
        int     (*_write)(void *, const char *, int);

        /* extension data, to avoid further ABI breakage */
        struct  __lttng_ust_sbuf _ext;
        /* data for long sequences of ungetc() */
        unsigned char *_up;     /* saved _p when _p is doing ungetc data */
        int     _ur;            /* saved _r when _r is counting ungetc data */

        /* tricks to meet minimum requirements even when malloc() fails */
        unsigned char _ubuf[3]; /* guarantee an ungetc() buffer */
        unsigned char _nbuf[1]; /* guarantee a getc() buffer */

        /* separate buffer for fgetln() when line crosses buffer boundary */
        struct  __lttng_ust_sbuf _lb;     /* buffer for fgetln() */

        /* Unix stdio files get aligned to block boundaries on fseek() */
        int     _blksize;       /* stat.st_blksize (may be != _bf._size) */
        fpos_t  _offset;        /* current lseek offset */
} LTTNG_UST_LFILE;

#define __SLBF  0x0001          /* line buffered */
#define __SNBF  0x0002          /* unbuffered */
#define __SRD   0x0004          /* OK to read */
#define __SWR   0x0008          /* OK to write */
        /* RD and WR are never simultaneously asserted */
#define __SRW   0x0010          /* open for reading & writing */
#define __SEOF  0x0020          /* found EOF */
#define __SERR  0x0040          /* found error */
#define __SMBF  0x0080          /* _buf is from malloc */
#define __SAPP  0x0100          /* fdopen()ed in append mode */
#define __SSTR  0x0200          /* this is an sprintf/snprintf string */
#define __SOPT  0x0400          /* do fseek() optimisation */
```

```
#define __SNPT  0x0800          /* do not do fseek() optimisation */
#define __SOFF  0x1000          /* set iff _offset is in fact correct */
#define __SMOD  0x2000          /* true => fgetln modified _p text */
#define __SALC  0x4000          /* allocate string space dynamically */

#define __sferror(p)    (((p)->_flags & __SERR) != 0)

extern int ust_safe_fflush(LTTNG_UST_LFILE *fp);
extern int ust_safe_vfprintf(LTTNG_UST_LFILE *fp, const char *fmt0, va_list ap);

extern size_t ust_safe_mbrtowc(wchar_t *pwc, const char *s, size_t n, mbstate_t *ps);

#endif /* UST_SNPRINTF_VARIOUS_H */
```

Notice for package(s)

lttng-tools

```
LTTng Tools licensing
David Goulet <david.goulet@polymtl.ca>
July 18, 2011

* LGPLv2.1

The library part is distributed under LGPLv2.1. See lgpl-2.1.txt for details.
This applies to:

-) include/lttng/*
-) src/lib/lttng-ctl/*

We have some tests LGPL but should not impact anything even if we change them
in the future to GPL.

LGPL-compatible source code can statically use the library header using:

#define _LGPL_SOURCE
#include <lttng/lttng.h>

Dynamic-only linking with the LGPL library is used if _LGPL_SOURCE is not
defined. It permits relinking with newer versions of the library, which is
required by the LGPL license.

* GPLv2

All remaining source code is distributed under the GPLv2 license. See
gpl-2.0.txt for details.
```

Notice for package(s)

cryptodev-linux
ethtool
gmp
iptables
libtool
lttng-tools
lzo
nettle
procps
util-linux
xz

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
```

have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

```
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```

Notice for package(s)

lttng-tools

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

  Finally, software patents pose a constant threat to the existence of
any free program.  We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder.  Therefore, we insist that
any patent license obtained for a version of the library must be
```

consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data

structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not

signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH

DAMAGES.

                    END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!

Notice for package(s)

libffi

```
libffi - Copyright (c) 1996-2014  Anthony Green, Red Hat, Inc and others.
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
``Software''), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Notice for package(s)

glib-2.0

```
/* GLIB - Library of useful routines for C programming
 * Copyright (C) 1995-1997  Peter Mattis, Spencer Kimball and Josh MacDonald
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2 of the License, or (at your option) any later version.
```

```
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, see <http://www.gnu.org/licenses/>.
 */

/*
 * Modified by the GLib Team and others 1997-2000.  See the AUTHORS
 * file for a list of people on the GLib Team.  See the ChangeLog
 * files for a list of changes.  These files are distributed with
 * GLib at ftp://ftp.gtk.org/pub/gtk/.
 */

#ifndef __G_LIB_H__
#define __G_LIB_H__

#define __GLIB_H_INSIDE__

#include <glib/galloca.h>
#include <glib/garray.h>
#include <glib/gasyncqueue.h>
#include <glib/gatomic.h>
#include <glib/gbacktrace.h>
#include <glib/gbase64.h>
#include <glib/gbitlock.h>
#include <glib/gbookmarkfile.h>
#include <glib/gbytes.h>
#include <glib/gcharset.h>
#include <glib/gchecksum.h>
#include <glib/gconvert.h>
#include <glib/gdataset.h>
#include <glib/gdate.h>
#include <glib/gdatetime.h>
#include <glib/gdir.h>
#include <glib/genviron.h>
#include <glib/gerror.h>
#include <glib/gfileutils.h>
#include <glib/ggettext.h>
#include <glib/ghash.h>
#include <glib/ghmac.h>
#include <glib/ghook.h>
#include <glib/ghostutils.h>
#include <glib/giochannel.h>
#include <glib/gkeyfile.h>
#include <glib/glist.h>
#include <glib/gmacros.h>
#include <glib/gmain.h>
#include <glib/gmappedfile.h>
#include <glib/gmarkup.h>
#include <glib/gmem.h>
#include <glib/gmessages.h>
#include <glib/gnode.h>
#include <glib/goption.h>
#include <glib/gpattern.h>
#include <glib/gpoll.h>
#include <glib/gprimes.h>
#include <glib/gqsort.h>
#include <glib/gquark.h>
#include <glib/gqueue.h>
#include <glib/grand.h>
#include <glib/gregex.h>
#include <glib/gscanner.h>
#include <glib/gsequence.h>
#include <glib/gshell.h>
#include <glib/gslice.h>
#include <glib/gslist.h>
#include <glib/gspawn.h>
#include <glib/gstrfuncs.h>
#include <glib/gstring.h>
#include <glib/gstringchunk.h>
#include <glib/gtestutils.h>
#include <glib/gthread.h>
#include <glib/gthreadpool.h>
#include <glib/gtimer.h>
#include <glib/gtimezone.h>
#include <glib/gtrashstack.h>
#include <glib/gtree.h>
#include <glib/gtypes.h>
#include <glib/gunicode.h>
#include <glib/gurifuncs.h>
#include <glib/gutils.h>
#include <glib/gvarianttype.h>
#include <glib/gvariant.h>
#include <glib/gversion.h>
#include <glib/gversionmacros.h>
#ifdef G_PLATFORM_WIN32
#include <glib/gwin32.h>
#endif

#ifndef G_DISABLE_DEPRECATED
#include <glib/deprecated/gallocator.h>
#include <glib/deprecated/gcache.h>
#include <glib/deprecated/gcompletion.h>
#include <glib/deprecated/gmain.h>
```

```
#include <glib/deprecated/grel.h>
#include <glib/deprecated/gthread.h>
#endif /* G_DISABLE_DEPRECATED */

#include <glib/glib-autocleanups.h>

#undef __GLIB_H_INSIDE__

#endif /* __G_LIB_H__ */
```

Notice for package(s)

glib-2.0

```
/* GMODULE - GLIB wrapper code for dynamic module loading
 * Copyright (C) 1998 Tim Janik
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, see <http://www.gnu.org/licenses/>.
 */

/*
 * Modified by the GLib Team and others 1997-2000.  See the AUTHORS
 * file for a list of people on the GLib Team.  See the ChangeLog
 * files for a list of changes.  These files are distributed with
 * GLib at ftp://ftp.gtk.org/pub/gtk/.
 */

#ifndef __GMODULE_H__
#define __GMODULE_H__

#include <glib.h>

G_BEGIN_DECLS

/* exporting and importing functions, this is special cased
 * to feature Windows dll stubs.
 */
#define G_MODULE_IMPORT         extern
#ifdef G_PLATFORM_WIN32
#  define       G_MODULE_EXPORT         __declspec(dllexport)
#else /* !G_PLATFORM_WIN32 */
#  define       G_MODULE_EXPORT
#endif /* !G_PLATFORM_WIN32 */

typedef enum
{
  G_MODULE_BIND_LAZY    = 1 << 0,
  G_MODULE_BIND_LOCAL   = 1 << 1,
  G_MODULE_BIND_MASK    = 0x03
} GModuleFlags;

typedef struct _GModule                 GModule;
typedef const gchar* (*GModuleCheckInit) (GModule       *module);
typedef void         (*GModuleUnload)    (GModule       *module);

/* return TRUE if dynamic module loading is supported */
GLIB_AVAILABLE_IN_ALL
gboolean        g_module_supported         (void) G_GNUC_CONST;

/* open a module 'file_name' and return handle, which is NULL on error */
GLIB_AVAILABLE_IN_ALL
GModule*              g_module_open          (const gchar  *file_name,
                                              GModuleFlags  flags);

/* close a previously opened module, returns TRUE on success */
GLIB_AVAILABLE_IN_ALL
gboolean              g_module_close         (GModule      *module);

/* make a module resident so g_module_close on it will be ignored */
GLIB_AVAILABLE_IN_ALL
void                  g_module_make_resident (GModule      *module);

/* query the last module error as a string */
GLIB_AVAILABLE_IN_ALL
const gchar *         g_module_error         (void);

/* retrieve a symbol pointer from 'module', returns TRUE on success */
GLIB_AVAILABLE_IN_ALL
gboolean              g_module_symbol        (GModule      *module,
                                              const gchar  *symbol_name,
```

```
                                          gpointer     *symbol);

/* retrieve the file name from an existing module */
GLIB_AVAILABLE_IN_ALL
const gchar *         g_module_name          (GModule      *module);

/* Build the actual file name containing a module. 'directory' is the
 * directory where the module file is supposed to be, or NULL or empty
 * in which case it should either be in the current directory or, on
 * some operating systems, in some standard place, for instance on the
 * PATH. Hence, to be absoultely sure to get the correct module,
 * always pass in a directory. The file name consists of the directory,
 * if supplied, and 'module_name' suitably decorated according to
 * the operating system's conventions (for instance lib*.so or *.dll).
 *
 * No checks are made that the file exists, or is of correct type.
 */
GLIB_AVAILABLE_IN_ALL
gchar*                g_module_build_path    (const gchar  *directory,
                                              const gchar  *module_name);


#ifndef __GTK_DOC_IGNORE__
#ifdef G_OS_WIN32
#define g_module_open g_module_open_utf8
#define g_module_name g_module_name_utf8

GLIB_AVAILABLE_IN_ALL
GModule *    g_module_open_utf8 (const gchar  *file_name,
                                 GModuleFlags  flags);
GLIB_AVAILABLE_IN_ALL
const gchar *g_module_name_utf8 (GModule      *module);
#endif
#endif

G_END_DECLS

#endif /* __GMODULE_H__ */
```

Notice for package(s)

glib-2.0

```
/*************************************************
*       Perl-Compatible Regular Expressions      *
*************************************************/

/* This is the public header file for the PCRE library, to be #included by
applications that call the PCRE functions.

           Copyright (c) 1997-2012 University of Cambridge

------------------------------------------------------------------------------
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.

    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.

    * Neither the name of the University of Cambridge nor the names of its
      contributors may be used to endorse or promote products derived from
      this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
------------------------------------------------------------------------------
*/

#ifndef _PCRE_H
#define _PCRE_H

/* The current PCRE version information. */

#define PCRE_MAJOR          8
#define PCRE_MINOR          31
#define PCRE_PRERELEASE
#define PCRE_DATE           2012-07-06
```

```
/* When an application links to a PCRE DLL in Windows, the symbols that are
imported have to be identified as such. When building PCRE, the appropriate
export setting is defined in pcre_internal.h, which includes this file. So we
don't change existing definitions of PCRE_EXP_DECL and PCRECPP_EXP_DECL. */

#if defined(_WIN32) && !defined(PCRE_STATIC)
#  ifndef PCRE_EXP_DECL
#    define PCRE_EXP_DECL  extern __declspec(dllimport)
#  endif
#  ifdef __cplusplus
#    ifndef PCRECPP_EXP_DECL
#      define PCRECPP_EXP_DECL  extern __declspec(dllimport)
#    endif
#    ifndef PCRECPP_EXP_DEFN
#      define PCRECPP_EXP_DEFN  __declspec(dllimport)
#    endif
#  endif
#endif

/* By default, we use the standard "extern" declarations. */

#ifndef PCRE_EXP_DECL
#  ifdef __cplusplus
#    define PCRE_EXP_DECL  extern "C"
#  else
#    define PCRE_EXP_DECL  extern
#  endif
#endif

#ifdef __cplusplus
#  ifndef PCRECPP_EXP_DECL
#    define PCRECPP_EXP_DECL  extern
#  endif
#  ifndef PCRECPP_EXP_DEFN
#    define PCRECPP_EXP_DEFN
#  endif
#endif

/* Have to include stdlib.h in order to ensure that size_t is defined;
it is needed here for malloc. */

#include <stdlib.h>

/* Allow for C++ users */

#ifdef __cplusplus
extern "C" {
#endif

/* Options. Some are compile-time only, some are run-time only, and some are
both, so we keep them all distinct. However, almost all the bits in the options
word are now used. In the long run, we may have to re-use some of the
compile-time only bits for runtime options, or vice versa. In the comments
below, "compile", "exec", and "DFA exec" mean that the option is permitted to
be set for those functions; "used in" means that an option may be set only for
compile, but is subsequently referenced in exec and/or DFA exec. Any of the
compile-time options may be inspected during studying (and therefore JIT
compiling). */

#define PCRE_CASELESS           0x00000001  /* Compile */
#define PCRE_MULTILINE          0x00000002  /* Compile */
#define PCRE_DOTALL             0x00000004  /* Compile */
#define PCRE_EXTENDED           0x00000008  /* Compile */
#define PCRE_ANCHORED           0x00000010  /* Compile, exec, DFA exec */
#define PCRE_DOLLAR_ENDONLY     0x00000020  /* Compile, used in exec, DFA exec */
#define PCRE_EXTRA              0x00000040  /* Compile */
#define PCRE_NOTBOL             0x00000080  /* Exec, DFA exec */
#define PCRE_NOTEOL             0x00000100  /* Exec, DFA exec */
#define PCRE_UNGREEDY           0x00000200  /* Compile */
#define PCRE_NOTEMPTY           0x00000400  /* Exec, DFA exec */
/* The next two are also used in exec and DFA exec */
#define PCRE_UTF8               0x00000800  /* Compile (same as PCRE_UTF16) */
#define PCRE_UTF16              0x00000800  /* Compile (same as PCRE_UTF8) */
#define PCRE_NO_AUTO_CAPTURE    0x00001000  /* Compile */
/* The next two are also used in exec and DFA exec */
#define PCRE_NO_UTF8_CHECK      0x00002000  /* Compile (same as PCRE_NO_UTF16_CHECK) */
#define PCRE_NO_UTF16_CHECK     0x00002000  /* Compile (same as PCRE_NO_UTF8_CHECK) */
#define PCRE_AUTO_CALLOUT       0x00004000  /* Compile */
#define PCRE_PARTIAL_SOFT       0x00008000  /* Exec, DFA exec */
#define PCRE_PARTIAL            0x00008000  /* Backwards compatible synonym */
#define PCRE_DFA_SHORTEST       0x00010000  /* DFA exec */
#define PCRE_DFA_RESTART        0x00020000  /* DFA exec */
#define PCRE_FIRSTLINE          0x00040000  /* Compile, used in exec, DFA exec */
#define PCRE_DUPNAMES           0x00080000  /* Compile */
#define PCRE_NEWLINE_CR         0x00100000  /* Compile, exec, DFA exec */
#define PCRE_NEWLINE_LF         0x00200000  /* Compile, exec, DFA exec */
#define PCRE_NEWLINE_CRLF       0x00300000  /* Compile, exec, DFA exec */
#define PCRE_NEWLINE_ANY        0x00400000  /* Compile, exec, DFA exec */
#define PCRE_NEWLINE_ANYCRLF    0x00500000  /* Compile, exec, DFA exec */
#define PCRE_BSR_ANYCRLF        0x00800000  /* Compile, exec, DFA exec */
#define PCRE_BSR_UNICODE        0x01000000  /* Compile, exec, DFA exec */
#define PCRE_JAVASCRIPT_COMPAT  0x02000000  /* Compile, used in exec */
#define PCRE_NO_START_OPTIMIZE  0x04000000  /* Compile, exec, DFA exec */
#define PCRE_NO_START_OPTIMISE  0x04000000  /* Synonym */
#define PCRE_PARTIAL_HARD       0x08000000  /* Exec, DFA exec */
#define PCRE_NOTEMPTY_ATSTART   0x10000000  /* Exec, DFA exec */
#define PCRE_UCP                0x20000000  /* Compile, used in exec, DFA exec */
```

```
/* Exec-time and get/set-time error codes */

#define PCRE_ERROR_NOMATCH          (-1)
#define PCRE_ERROR_NULL             (-2)
#define PCRE_ERROR_BADOPTION        (-3)
#define PCRE_ERROR_BADMAGIC         (-4)
#define PCRE_ERROR_UNKNOWN_OPCODE   (-5)
#define PCRE_ERROR_UNKNOWN_NODE     (-5)  /* For backward compatibility */
#define PCRE_ERROR_NOMEMORY         (-6)
#define PCRE_ERROR_NOSUBSTRING      (-7)
#define PCRE_ERROR_MATCHLIMIT       (-8)
#define PCRE_ERROR_CALLOUT          (-9)  /* Never used by PCRE itself */
#define PCRE_ERROR_BADUTF8         (-10)  /* Same for 8/16 */
#define PCRE_ERROR_BADUTF16        (-10)  /* Same for 8/16 */
#define PCRE_ERROR_BADUTF8_OFFSET  (-11)  /* Same for 8/16 */
#define PCRE_ERROR_BADUTF16_OFFSET (-11)  /* Same for 8/16 */
#define PCRE_ERROR_PARTIAL         (-12)
#define PCRE_ERROR_BADPARTIAL      (-13)
#define PCRE_ERROR_INTERNAL        (-14)
#define PCRE_ERROR_BADCOUNT        (-15)
#define PCRE_ERROR_DFA_UITEM       (-16)
#define PCRE_ERROR_DFA_UCOND       (-17)
#define PCRE_ERROR_DFA_UMLIMIT     (-18)
#define PCRE_ERROR_DFA_WSSIZE      (-19)
#define PCRE_ERROR_DFA_RECURSE     (-20)
#define PCRE_ERROR_RECURSIONLIMIT  (-21)
#define PCRE_ERROR_NULLWSLIMIT     (-22)  /* No longer actually used */
#define PCRE_ERROR_BADNEWLINE      (-23)
#define PCRE_ERROR_BADOFFSET       (-24)
#define PCRE_ERROR_SHORTUTF8       (-25)
#define PCRE_ERROR_SHORTUTF16      (-25)  /* Same for 8/16 */
#define PCRE_ERROR_RECURSELOOP     (-26)
#define PCRE_ERROR_JIT_STACKLIMIT  (-27)
#define PCRE_ERROR_BADMODE         (-28)
#define PCRE_ERROR_BADENDIANNESS   (-29)
#define PCRE_ERROR_DFA_BADRESTART  (-30)

/* Specific error codes for UTF-8 validity checks */

#define PCRE_UTF8_ERR0               0
#define PCRE_UTF8_ERR1               1
#define PCRE_UTF8_ERR2               2
#define PCRE_UTF8_ERR3               3
#define PCRE_UTF8_ERR4               4
#define PCRE_UTF8_ERR5               5
#define PCRE_UTF8_ERR6               6
#define PCRE_UTF8_ERR7               7
#define PCRE_UTF8_ERR8               8
#define PCRE_UTF8_ERR9               9
#define PCRE_UTF8_ERR10             10
#define PCRE_UTF8_ERR11             11
#define PCRE_UTF8_ERR12             12
#define PCRE_UTF8_ERR13             13
#define PCRE_UTF8_ERR14             14
#define PCRE_UTF8_ERR15             15
#define PCRE_UTF8_ERR16             16
#define PCRE_UTF8_ERR17             17
#define PCRE_UTF8_ERR18             18
#define PCRE_UTF8_ERR19             19
#define PCRE_UTF8_ERR20             20
#define PCRE_UTF8_ERR21             21

/* Specific error codes for UTF-16 validity checks */

#define PCRE_UTF16_ERR0              0
#define PCRE_UTF16_ERR1              1
#define PCRE_UTF16_ERR2              2
#define PCRE_UTF16_ERR3              3
#define PCRE_UTF16_ERR4              4

/* Request types for pcre_fullinfo() */

#define PCRE_INFO_OPTIONS            0
#define PCRE_INFO_SIZE               1
#define PCRE_INFO_CAPTURECOUNT       2
#define PCRE_INFO_BACKREFMAX         3
#define PCRE_INFO_FIRSTBYTE          4
#define PCRE_INFO_FIRSTCHAR          4  /* For backwards compatibility */
#define PCRE_INFO_FIRSTTABLE         5
#define PCRE_INFO_LASTLITERAL        6
#define PCRE_INFO_NAMEENTRYSIZE      7
#define PCRE_INFO_NAMECOUNT          8
#define PCRE_INFO_NAMETABLE          9
#define PCRE_INFO_STUDYSIZE         10
#define PCRE_INFO_DEFAULT_TABLES    11
#define PCRE_INFO_OKPARTIAL         12
#define PCRE_INFO_JCHANGED          13
#define PCRE_INFO_HASCRORLF         14
#define PCRE_INFO_MINLENGTH         15
#define PCRE_INFO_JIT               16
#define PCRE_INFO_JITSIZE           17
#define PCRE_INFO_MAXLOOKBEHIND     18

/* Request types for pcre_config(). Do not re-arrange, in order to remain
compatible. */

#define PCRE_CONFIG_UTF8                    0
#define PCRE_CONFIG_NEWLINE                 1
```

```
#define PCRE_CONFIG_LINK_SIZE               2
#define PCRE_CONFIG_POSIX_MALLOC_THRESHOLD  3
#define PCRE_CONFIG_MATCH_LIMIT             4
#define PCRE_CONFIG_STACKRECURSE            5
#define PCRE_CONFIG_UNICODE_PROPERTIES      6
#define PCRE_CONFIG_MATCH_LIMIT_RECURSION   7
#define PCRE_CONFIG_BSR                     8
#define PCRE_CONFIG_JIT                     9
#define PCRE_CONFIG_UTF16                  10
#define PCRE_CONFIG_JITTARGET              11

/* Request types for pcre_study(). Do not re-arrange, in order to remain
compatible. */

#define PCRE_STUDY_JIT_COMPILE                0x0001
#define PCRE_STUDY_JIT_PARTIAL_SOFT_COMPILE   0x0002
#define PCRE_STUDY_JIT_PARTIAL_HARD_COMPILE   0x0004

/* Bit flags for the pcre[16]_extra structure. Do not re-arrange or redefine
these bits, just add new ones on the end, in order to remain compatible. */

#define PCRE_EXTRA_STUDY_DATA             0x0001
#define PCRE_EXTRA_MATCH_LIMIT            0x0002
#define PCRE_EXTRA_CALLOUT_DATA           0x0004
#define PCRE_EXTRA_TABLES                 0x0008
#define PCRE_EXTRA_MATCH_LIMIT_RECURSION  0x0010
#define PCRE_EXTRA_MARK                   0x0020
#define PCRE_EXTRA_EXECUTABLE_JIT         0x0040

/* Types */

struct real_pcre;                 /* declaration; the definition is private  */
typedef struct real_pcre pcre;

struct real_pcre16;               /* declaration; the definition is private  */
typedef struct real_pcre16 pcre16;

struct real_pcre_jit_stack;       /* declaration; the definition is private  */
typedef struct real_pcre_jit_stack pcre_jit_stack;

struct real_pcre16_jit_stack;     /* declaration; the definition is private  */
typedef struct real_pcre16_jit_stack pcre16_jit_stack;

/* If PCRE is compiled with 16 bit character support, PCRE_UCHAR16 must contain
a 16 bit wide signed data type. Otherwise it can be a dummy data type since
pcre16 functions are not implemented. There is a check for this in pcre_internal.h. */
#ifndef PCRE_UCHAR16
#define PCRE_UCHAR16 unsigned short
#endif

#ifndef PCRE_SPTR16
#define PCRE_SPTR16 const PCRE_UCHAR16 *
#endif

/* When PCRE is compiled as a C++ library, the subject pointer type can be
replaced with a custom type. For conventional use, the public interface is a
const char *. */

#ifndef PCRE_SPTR
#define PCRE_SPTR const char *
#endif

/* The structure for passing additional data to pcre_exec(). This is defined in
such as way as to be extensible. Always add new fields at the end, in order to
remain compatible. */

typedef struct pcre_extra {
  unsigned long int flags;        /* Bits for which fields are set */
  void *study_data;               /* Opaque data from pcre_study() */
  unsigned long int match_limit;  /* Maximum number of calls to match() */
  void *callout_data;             /* Data passed back in callouts */
  const unsigned char *tables;    /* Pointer to character tables */
  unsigned long int match_limit_recursion; /* Max recursive calls to match() */
  unsigned char **mark;           /* For passing back a mark pointer */
  void *executable_jit;           /* Contains a pointer to a compiled jit code */
} pcre_extra;

/* Same structure as above, but with 16 bit char pointers. */

typedef struct pcre16_extra {
  unsigned long int flags;        /* Bits for which fields are set */
  void *study_data;               /* Opaque data from pcre_study() */
  unsigned long int match_limit;  /* Maximum number of calls to match() */
  void *callout_data;             /* Data passed back in callouts */
  const unsigned char *tables;    /* Pointer to character tables */
  unsigned long int match_limit_recursion; /* Max recursive calls to match() */
  PCRE_UCHAR16 **mark;            /* For passing back a mark pointer */
  void *executable_jit;           /* Contains a pointer to a compiled jit code */
} pcre16_extra;

/* The structure for passing out data via the pcre_callout_function. We use a
structure so that new fields can be added on the end in future versions,
without changing the API of the function, thereby allowing old clients to work
without modification. */

typedef struct pcre_callout_block {
  int          version;           /* Identifies version of block */
  /* ------------------------ Version 0 ------------------------------- */
```

```
  int          callout_number;    /* Number compiled into pattern */
  int         *offset_vector;     /* The offset vector */
  PCRE_SPTR    subject;           /* The subject being matched */
  int          subject_length;    /* The length of the subject */
  int          start_match;       /* Offset to start of this match attempt */
  int          current_position;  /* Where we currently are in the subject */
  int          capture_top;       /* Max current capture */
  int          capture_last;      /* Most recently closed capture */
  void        *callout_data;      /* Data passed in with the call */
  /* ------------------- Added for Version 1 -------------------------- */
  int          pattern_position;  /* Offset to next item in the pattern */
  int          next_item_length;  /* Length of next item in the pattern */
  /* ------------------- Added for Version 2 -------------------------- */
  const unsigned char *mark;      /* Pointer to current mark or NULL    */
  /* ------------------------------------------------------------------ */
} pcre_callout_block;

/* Same structure as above, but with 16 bit char pointers. */

typedef struct pcre16_callout_block {
  int          version;           /* Identifies version of block */
  /* ------------------------ Version 0 ------------------------------- */
  int          callout_number;    /* Number compiled into pattern */
  int         *offset_vector;     /* The offset vector */
  PCRE_SPTR16  subject;           /* The subject being matched */
  int          subject_length;    /* The length of the subject */
  int          start_match;       /* Offset to start of this match attempt */
  int          current_position;  /* Where we currently are in the subject */
  int          capture_top;       /* Max current capture */
  int          capture_last;      /* Most recently closed capture */
  void        *callout_data;      /* Data passed in with the call */
  /* ------------------- Added for Version 1 -------------------------- */
  int          pattern_position;  /* Offset to next item in the pattern */
  int          next_item_length;  /* Length of next item in the pattern */
  /* ------------------- Added for Version 2 -------------------------- */
  const PCRE_UCHAR16 *mark;       /* Pointer to current mark or NULL    */
  /* ------------------------------------------------------------------ */
} pcre16_callout_block;

/* Indirection for store get and free functions. These can be set to
alternative malloc/free functions if required. Special ones are used in the
non-recursive case for "frames". There is also an optional callout function
that is triggered by the (?) regex item. For Virtual Pascal, these definitions
have to take another form. */

#ifndef VPCOMPAT
PCRE_EXP_DECL void *(*pcre_malloc)(size_t);
PCRE_EXP_DECL void  (*pcre_free)(void *);
PCRE_EXP_DECL void *(*pcre_stack_malloc)(size_t);
PCRE_EXP_DECL void  (*pcre_stack_free)(void *);
PCRE_EXP_DECL int   (*pcre_callout)(pcre_callout_block *);

PCRE_EXP_DECL void *(*pcre16_malloc)(size_t);
PCRE_EXP_DECL void  (*pcre16_free)(void *);
PCRE_EXP_DECL void *(*pcre16_stack_malloc)(size_t);
PCRE_EXP_DECL void  (*pcre16_stack_free)(void *);
PCRE_EXP_DECL int   (*pcre16_callout)(pcre16_callout_block *);
#else   /* VPCOMPAT */
PCRE_EXP_DECL void *pcre_malloc(size_t);
PCRE_EXP_DECL void  pcre_free(void *);
PCRE_EXP_DECL void *pcre_stack_malloc(size_t);
PCRE_EXP_DECL void  pcre_stack_free(void *);
PCRE_EXP_DECL int   pcre_callout(pcre_callout_block *);

PCRE_EXP_DECL void *pcre16_malloc(size_t);
PCRE_EXP_DECL void  pcre16_free(void *);
PCRE_EXP_DECL void *pcre16_stack_malloc(size_t);
PCRE_EXP_DECL void  pcre16_stack_free(void *);
PCRE_EXP_DECL int   pcre16_callout(pcre16_callout_block *);
#endif  /* VPCOMPAT */

/* User defined callback which provides a stack just before the match starts. */

typedef pcre_jit_stack *(*pcre_jit_callback)(void *);
typedef pcre16_jit_stack *(*pcre16_jit_callback)(void *);

/* Exported PCRE functions */

PCRE_EXP_DECL pcre *pcre_compile(const char *, int, const char **, int *,
                  const unsigned char *);
PCRE_EXP_DECL pcre16 *pcre16_compile(PCRE_SPTR16, int, const char **, int *,
                  const unsigned char *);
PCRE_EXP_DECL pcre *pcre_compile2(const char *, int, int *, const char **,
                  int *, const unsigned char *);
PCRE_EXP_DECL pcre16 *pcre16_compile2(PCRE_SPTR16, int, int *, const char **,
                  int *, const unsigned char *);
PCRE_EXP_DECL int  pcre_config(int, void *);
PCRE_EXP_DECL int  pcre16_config(int, void *);
PCRE_EXP_DECL int  pcre_copy_named_substring(const pcre *, const char *,
                  int *, int, const char *, char *, int);
PCRE_EXP_DECL int  pcre16_copy_named_substring(const pcre16 *, PCRE_SPTR16,
                  int *, int, PCRE_SPTR16, PCRE_UCHAR16 *, int);
PCRE_EXP_DECL int  pcre_copy_substring(const char *, int *, int, int,
                  char *, int);
PCRE_EXP_DECL int  pcre16_copy_substring(PCRE_SPTR16, int *, int, int,
                  PCRE_UCHAR16 *, int);
PCRE_EXP_DECL int  pcre_dfa_exec(const pcre *, const pcre_extra *,
                  const char *, int, int, int, int *, int , int *, int);
```

```
PCRE_EXP_DECL int  pcre16_dfa_exec(const pcre16 *, const pcre16_extra *,
                  PCRE_SPTR16, int, int, int, int *, int , int *, int);
PCRE_EXP_DECL int  pcre_exec(const pcre *, const pcre_extra *, PCRE_SPTR,
                   int, int, int, int *, int);
PCRE_EXP_DECL int  pcre16_exec(const pcre16 *, const pcre16_extra *,
                   PCRE_SPTR16, int, int, int, int *, int);
PCRE_EXP_DECL void pcre_free_substring(const char *);
PCRE_EXP_DECL void pcre16_free_substring(PCRE_SPTR16);
PCRE_EXP_DECL void pcre_free_substring_list(const char **);
PCRE_EXP_DECL void pcre16_free_substring_list(PCRE_SPTR16 *);
PCRE_EXP_DECL int  pcre_fullinfo(const pcre *, const pcre_extra *, int,
                  void *);
PCRE_EXP_DECL int  pcre16_fullinfo(const pcre16 *, const pcre16_extra *, int,
                  void *);
PCRE_EXP_DECL int  pcre_get_named_substring(const pcre *, const char *,
                  int *, int, const char *, const char **);
PCRE_EXP_DECL int  pcre16_get_named_substring(const pcre16 *, PCRE_SPTR16,
                  int *, int, PCRE_SPTR16, PCRE_SPTR16 *);
PCRE_EXP_DECL int  pcre_get_stringnumber(const pcre *, const char *);
PCRE_EXP_DECL int  pcre16_get_stringnumber(const pcre16 *, PCRE_SPTR16);
PCRE_EXP_DECL int  pcre_get_stringtable_entries(const pcre *, const char *,
                  char **, char **);
PCRE_EXP_DECL int  pcre16_get_stringtable_entries(const pcre16 *, PCRE_SPTR16,
                  PCRE_UCHAR16 **, PCRE_UCHAR16 **);
PCRE_EXP_DECL int  pcre_get_substring(const char *, int *, int, int,
                  const char **);
PCRE_EXP_DECL int  pcre16_get_substring(PCRE_SPTR16, int *, int, int,
                  PCRE_SPTR16 *);
PCRE_EXP_DECL int  pcre_get_substring_list(const char *, int *, int,
                  const char ***);
PCRE_EXP_DECL int  pcre16_get_substring_list(PCRE_SPTR16, int *, int,
                  PCRE_SPTR16 **);
PCRE_EXP_DECL const unsigned char *pcre_maketables(void);
PCRE_EXP_DECL const unsigned char *pcre16_maketables(void);
PCRE_EXP_DECL int  pcre_refcount(pcre *, int);
PCRE_EXP_DECL int  pcre16_refcount(pcre16 *, int);
PCRE_EXP_DECL pcre_extra *pcre_study(const pcre *, int, const char **);
PCRE_EXP_DECL pcre16_extra *pcre16_study(const pcre16 *, int, const char **);
PCRE_EXP_DECL void pcre_free_study(pcre_extra *);
PCRE_EXP_DECL void pcre16_free_study(pcre16_extra *);
PCRE_EXP_DECL const char *pcre_version(void);
PCRE_EXP_DECL const char *pcre16_version(void);

/* Utility functions for byte order swaps. */
PCRE_EXP_DECL int  pcre_pattern_to_host_byte_order(pcre *, pcre_extra *,
                  const unsigned char *);
PCRE_EXP_DECL int  pcre16_pattern_to_host_byte_order(pcre16 *, pcre16_extra *,
                  const unsigned char *);
PCRE_EXP_DECL int  pcre16_utf16_to_host_byte_order(PCRE_UCHAR16 *,
                  PCRE_SPTR16, int, int *, int);

/* JIT compiler related functions. */

PCRE_EXP_DECL pcre_jit_stack *pcre_jit_stack_alloc(int, int);
PCRE_EXP_DECL pcre16_jit_stack *pcre16_jit_stack_alloc(int, int);
PCRE_EXP_DECL void pcre_jit_stack_free(pcre_jit_stack *);
PCRE_EXP_DECL void pcre16_jit_stack_free(pcre16_jit_stack *);
PCRE_EXP_DECL void pcre_assign_jit_stack(pcre_extra *,
                  pcre_jit_callback, void *);
PCRE_EXP_DECL void pcre16_assign_jit_stack(pcre16_extra *,
                  pcre16_jit_callback, void *);

#ifdef __cplusplus
}  /* extern "C" */
#endif

#endif /* End of pcre.h */
```

Notice for package(s)

glib-2.0

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. SHOULD THE WORK PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE WORK AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
WORK, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

Notice for package(s)

babeltrace

Common Trace Format - Licensing
Mathieu Desnoyers
September 26, 2010

* MIT license :

This library is distributed under the MIT license. It is intended to allow use
in both free and proprietary software. See mit-license.txt for details.

* GPLv2

Library test code is distributed under the GPLv2 license, as specified in the
per-file license. See gpl-2.0.txt for details.

* LGPLv2.1

The file include/babeltrace/list.h is licensed under LGPLv2.1. It only
contains trivial static inline functions and macros, and, therefore,
including it does not make babeltrace a derivative work on this header.
Please refer to the LGPLv2.1 license for details.

Notice for package(s)

openssl

  LICENSE ISSUES
  ==============

  The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
  the OpenSSL License and the original SSLeay license apply to the toolkit.
  See below for the actual license texts. Actually both licenses are BSD-style
  Open Source licenses. In case of any license issues related to OpenSSL
  please contact openssl-core@openssl.org.

  OpenSSL License

```
/* ====================================================================
 * Copyright (c) 1998-2016 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
```

```
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

Notice for package(s)

openssl

OpenSSL License

 Original SSLeay License

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS`` AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTI
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN A

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot
[including the GNU Public Licence.]

Notice for package(s)

gdbm

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007, 2011 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of

works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under

the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

  When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

    c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

  A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

    a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

    c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or
authors of the material; or

e) Declining to grant rights under trademark law for use of some
trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that
material by anyone who conveys the material (or modified versions of
it) with contractual assumptions of liability to the recipient, for
any liability that these contractual assumptions directly impose on
those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

  Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

  Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

    <program> Copyright (C) <year> <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

```
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

Notice for package(s)

python-requests

```
Copyright 2015 Kenneth Reitz

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

Notice for package(s)

openvswitch
protobuf-c
python-requests

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial revisions, annotations, elaborations, or other modifications
      represent, as a whole, an original work of authorship. For the purposes
      of this License, Derivative Works shall not include works that remain
      separable from, or merely link (or bind by name) to the interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or additions
```

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don`t include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Notice for package(s)

protobuf

```
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
    * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it.  This code is not
```

```
standalone and requires a support library to be linked with it.  This
support library is itself covered by the above license.
```

Notice for package(s)

protobuf
tipcutils

```
Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the document
Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
```

Notice for package(s)

protobuf-c

```
/* --- protobuf-c.c: public protobuf c runtime implementation --- */

/*
 * Copyright 2008, Dave Benson.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with
 * the License. You may obtain a copy of the License
 * at http://www.apache.org/licenses/LICENSE-2.0 Unless
 * required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

/* TODO items:

     * 64-BIT OPTIMIZATION: certain implementations use 32-bit math even on 64-bit platforms
        (uint64_size, uint64_pack, parse_uint64)

     * get_packed_size and pack seem to use type-prefixed names,
       whereas parse uses type-suffixed names.  pick one and stick with it.
       Decision:  go with type-suffixed, since the type (or its instance)
       is typically the object of the verb.
       NOTE: perhaps the "parse" methods should be reanemd to "unpack"
       at the same time. (this only affects internal (static) functions)

     * use TRUE and FALSE instead of 1 and 0 as appropriate

     * use size_t consistently
 */

#if HAVE_PROTOBUF_C_CONFIG_H
#include "protobuf-c-config.h"
#endif
#include <stdio.h>                      /* for occasional printf()s */
#include <stdlib.h>                     /* for abort(), malloc() etc */
#include <string.h>                     /* for strlen(), memcpy(), memmove() */
#if HAVE_ALLOCA_H
#include <alloca.h>
#elif HAVE_MALLOC_H
#include <malloc.h>
#endif

#ifndef PRINT_UNPACK_ERRORS
#define PRINT_UNPACK_ERRORS              1
#endif

#include "protobuf-c.h"

#define MAX_UINT64_ENCODED_SIZE 10

/* convenience macros */
#define TMPALLOC(allocator, size) ((allocator)->tmp_alloc ((allocator)->allocator_data, (size)))
#define FREE(allocator, ptr)   \
   do { if ((ptr) != NULL) ((allocator)->free ((allocator)->allocator_data, (ptr))); } while(0)
#define UNALIGNED_ALLOC(allocator, size) ALLOC (allocator, size) /* placeholder */
#define STRUCT_MEMBER_P(struct_p, struct_offset)   \
    ((void *) ((uint8_t*) (struct_p) + (struct_offset)))
```

```
#define STRUCT_MEMBER(member_type, struct_p, struct_offset)   \
    (*(member_type*) STRUCT_MEMBER_P ((struct_p), (struct_offset)))
#define STRUCT_MEMBER_PTR(member_type, struct_p, struct_offset)   \
    ((member_type*) STRUCT_MEMBER_P ((struct_p), (struct_offset)))
#define TRUE 1
#define FALSE 0

static void
alloc_failed_warning (unsigned size, const char *filename, unsigned line)
{
  fprintf (stderr,
           "WARNING: out-of-memory allocating a block of size %u (%s:%u)\n",
           size, filename, line);
}

/* Try to allocate memory, running some special code if it fails. */
#define DO_ALLOC(dst, allocator, size, fail_code)                          \
{ size_t da__allocation_size = (size);                                     \
  if (da__allocation_size == 0)                                            \
    dst = NULL;                                                            \
  else if ((dst=((allocator)->alloc ((allocator)->allocator_data,          \
                                     da__allocation_size))) == NULL)       \
    {                                                                      \
      alloc_failed_warning (da__allocation_size, __FILE__, __LINE__);      \
      fail_code;                                                           \
    }                                                                      \
}
#define DO_UNALIGNED_ALLOC  DO_ALLOC            /* placeholder */



#define ASSERT_IS_ENUM_DESCRIPTOR(desc) \
  assert((desc)->magic == PROTOBUF_C_ENUM_DESCRIPTOR_MAGIC)
#define ASSERT_IS_MESSAGE_DESCRIPTOR(desc) \
  assert((desc)->magic == PROTOBUF_C_MESSAGE_DESCRIPTOR_MAGIC)
#define ASSERT_IS_MESSAGE(message) \
  ASSERT_IS_MESSAGE_DESCRIPTOR((message)->descriptor)
#define ASSERT_IS_SERVICE_DESCRIPTOR(desc) \
  assert((desc)->magic == PROTOBUF_C_SERVICE_DESCRIPTOR_MAGIC)

/* --- allocator --- */

static void protobuf_c_out_of_memory_default (void)
{
  fprintf (stderr, "Out Of Memory!!!\n");
  abort ();
}
void (*protobuf_c_out_of_memory) (void) = protobuf_c_out_of_memory_default;

static void *system_alloc(void *allocator_data, size_t size)
{
  void *rv;
  (void) allocator_data;
  if (size == 0)
    return NULL;
  rv = malloc (size);
  if (rv == NULL)
    protobuf_c_out_of_memory ();
  return rv;
}

static void system_free (void *allocator_data, void *data)
{
  (void) allocator_data;
  if (data)
    free (data);
}

/* Some users may configure the default allocator;
   providing your own allocator to unpack() is prefered.
   this allocator is still used for packing nested messages. */
ProtobufCAllocator protobuf_c_default_allocator =
{
  system_alloc,
  system_free,
  NULL,
  8192,
  NULL
};

/* Users should NOT modify this structure,
   but it's difficult to prevent.

   please modify protobuf_c_default_allocator instead. */
ProtobufCAllocator protobuf_c_system_allocator =
{
  system_alloc,
  system_free,
  NULL,
  8192,
  NULL
};

/* === buffer-simple === */
void
protobuf_c_buffer_simple_append (ProtobufCBuffer *buffer,
                                 size_t           len,
                                 const uint8_t   *data)
```

```
{
  ProtobufCBufferSimple *simp = (ProtobufCBufferSimple *) buffer;
  size_t new_len = simp->len + len;
  if (new_len > simp->alloced)
    {
      size_t new_alloced = simp->alloced * 2;
      uint8_t *new_data;
      while (new_alloced < new_len)
        new_alloced += new_alloced;
      DO_ALLOC (new_data, &protobuf_c_default_allocator, new_alloced, return);
      memcpy (new_data, simp->data, simp->len);
      if (simp->must_free_data)
        FREE (&protobuf_c_default_allocator, simp->data);
      else
        simp->must_free_data = 1;
      simp->data = new_data;
      simp->alloced = new_alloced;
    }
  memcpy (simp->data + simp->len, data, len);
  simp->len = new_len;
}

/* === get_packed_size() === */

/* Return the number of bytes required to store the
   tag for the field (which includes 3 bits for
   the wire-type, and a single bit that denotes the end-of-tag. */
static inline size_t
get_tag_size (unsigned number)
{
  if (number < (1<<4))
    return 1;
  else if (number < (1<<11))
    return 2;
  else if (number < (1<<18))
    return 3;
  else if (number < (1<<25))
    return 4;
  else
    return 5;
}

/* Return the number of bytes required to store
   a variable-length unsigned integer that fits in 32-bit uint
   in base-128 encoding. */
static inline size_t
uint32_size (uint32_t v)
{
  if (v < (1<<7))
    return 1;
  else if (v < (1<<14))
    return 2;
  else if (v < (1<<21))
    return 3;
  else if (v < (1<<28))
    return 4;
  else
    return 5;
}
/* Return the number of bytes required to store
   a variable-length signed integer that fits in 32-bit int
   in base-128 encoding. */
static inline size_t
int32_size (int32_t v)
{
  if (v < 0)
    return 10;
  else if (v < (1<<7))
    return 1;
  else if (v < (1<<14))
    return 2;
  else if (v < (1<<21))
    return 3;
  else if (v < (1<<28))
    return 4;
  else
    return 5;
}
/* return the zigzag-encoded 32-bit unsigned int from a 32-bit signed int */
static inline uint32_t
zigzag32 (int32_t v)
{
  if (v < 0)
    return ((uint32_t)(-v)) * 2 - 1;
  else
    return v * 2;
}
/* Return the number of bytes required to store
   a variable-length signed integer that fits in 32-bit int,
   converted to unsigned via the zig-zag algorithm,
   then packed using base-128 encoding. */
static inline size_t
sint32_size (int32_t v)
{
  return uint32_size(zigzag32(v));
}

/* Return the number of bytes required to store
```

```
   a variable-length unsigned integer that fits in 64-bit uint
   in base-128 encoding. */
static inline size_t
uint64_size (uint64_t v)
{
  uint32_t upper_v = (uint32_t )(v>>32);
  if (upper_v == 0)
    return uint32_size ((uint32_t)v);
  else if (upper_v < (1<<3))
    return 5;
  else if (upper_v < (1<<10))
    return 6;
  else if (upper_v < (1<<17))
    return 7;
  else if (upper_v < (1<<24))
    return 8;
  else if (upper_v < (1U<<31))
    return 9;
  else
    return 10;
}

/* return the zigzag-encoded 64-bit unsigned int from a 64-bit signed int */
static inline uint64_t
zigzag64 (int64_t v)
{
  if (v < 0)
    return ((uint64_t)(-v)) * 2 - 1;
  else
    return v * 2;
}

/* Return the number of bytes required to store
   a variable-length signed integer that fits in 64-bit int,
   converted to unsigned via the zig-zag algorithm,
   then packed using base-128 encoding. */
static inline size_t
sint64_size (int64_t v)
{
  return uint64_size(zigzag64(v));
}

/* Get serialized size of a single field in the message,
   including the space needed by the identifying tag. */
static size_t
required_field_get_packed_size (const ProtobufCFieldDescriptor *field,
                                const void *member)
{
  size_t rv = get_tag_size (field->id);
  switch (field->type)
    {
    case PROTOBUF_C_TYPE_SINT32:
      return rv + sint32_size (*(const int32_t *) member);
    case PROTOBUF_C_TYPE_INT32:
      return rv + int32_size (*(const uint32_t *) member);
    case PROTOBUF_C_TYPE_UINT32:
      return rv + uint32_size (*(const uint32_t *) member);
    case PROTOBUF_C_TYPE_SINT64:
      return rv + sint64_size (*(const int64_t *) member);
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      return rv + uint64_size (*(const uint64_t *) member);
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
      return rv + 4;
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
      return rv + 8;
    case PROTOBUF_C_TYPE_BOOL:
      return rv + 1;
    case PROTOBUF_C_TYPE_FLOAT:
      return rv + 4;
    case PROTOBUF_C_TYPE_DOUBLE:
      return rv + 8;
    case PROTOBUF_C_TYPE_ENUM:
      // TODO: is this correct for negative-valued enums?
      return rv + uint32_size (*(const uint32_t *) member);
    case PROTOBUF_C_TYPE_STRING:
      {
        const char *str = *(char * const *) member;
        size_t len = str ? strlen (str) : 0;
        return rv + uint32_size (len) + len;
      }
    case PROTOBUF_C_TYPE_BYTES:
      {
        size_t len = ((const ProtobufCBinaryData*) member)->len;
        return rv + uint32_size (len) + len;
      }
    //case PROTOBUF_C_TYPE_GROUP:
    case PROTOBUF_C_TYPE_MESSAGE:
      {
        const ProtobufCMessage *msg = * (ProtobufCMessage * const *) member;
        size_t subrv = msg ? protobuf_c_message_get_packed_size (msg) : 0;
        return rv + uint32_size (subrv) + subrv;
      }
    }
  PROTOBUF_C_ASSERT_NOT_REACHED ();
  return 0;
```

```
}

/* Get serialized size of a single optional field in the message,
   including the space needed by the identifying tag.
   Returns 0 if the optional field isn't set. */
static size_t
optional_field_get_packed_size (const ProtobufCFieldDescriptor *field,
                                const protobuf_c_boolean *has,
                                const void *member)
{
  if (field->type == PROTOBUF_C_TYPE_MESSAGE
   || field->type == PROTOBUF_C_TYPE_STRING)
    {
      const void *ptr = * (const void * const *) member;
      if (ptr == NULL
       || ptr == field->default_value)
        return 0;
    }
  else
    {
      if (!*has)
        return 0;
    }
  return required_field_get_packed_size (field, member);
}

/* Get serialized size of a repeated field in the message,
   which may consist of any number of values (including 0).
   Includes the space needed by the identifying tags (as needed). */
static size_t
repeated_field_get_packed_size (const ProtobufCFieldDescriptor *field,
                                size_t count,
                                const void *member)
{
  size_t header_size;
  size_t rv = 0;
  unsigned i;
  void *array = * (void * const *) member;
  if (count == 0)
    return 0;
  header_size = get_tag_size (field->id);
  if (!field->packed)
    header_size *= count;
  switch (field->type)
    {
    case PROTOBUF_C_TYPE_SINT32:
      for (i = 0; i < count; i++)
        rv += sint32_size (((int32_t*)array)[i]);
      break;
    case PROTOBUF_C_TYPE_INT32:
      for (i = 0; i < count; i++)
        rv += int32_size (((uint32_t*)array)[i]);
      break;
    case PROTOBUF_C_TYPE_UINT32:
    case PROTOBUF_C_TYPE_ENUM:
      for (i = 0; i < count; i++)
        rv += uint32_size (((uint32_t*)array)[i]);
      break;
    case PROTOBUF_C_TYPE_SINT64:
      for (i = 0; i < count; i++)
        rv += sint64_size (((int64_t*)array)[i]);
      break;
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      for (i = 0; i < count; i++)
        rv += uint64_size (((uint64_t*)array)[i]);
      break;
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      rv += 4 * count;
      break;
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      rv += 8 * count;
      break;
    case PROTOBUF_C_TYPE_BOOL:
      rv += count;
      break;
    case PROTOBUF_C_TYPE_STRING:
      for (i = 0; i < count; i++)
        {
          size_t len = strlen (((char**) array)[i]);
          rv += uint32_size (len) + len;
        }
      break;

    case PROTOBUF_C_TYPE_BYTES:
      for (i = 0; i < count; i++)
        {
          size_t len = ((ProtobufCBinaryData*) array)[i].len;
          rv += uint32_size (len) + len;
        }
      break;
    case PROTOBUF_C_TYPE_MESSAGE:
      for (i = 0; i < count; i++)
        {
```

```
          size_t len = protobuf_c_message_get_packed_size (((ProtobufCMessage **) array)[i]);
          rv += uint32_size (len) + len;
        }
      break;
    //case PROTOBUF_C_TYPE_GROUP:          // NOT SUPPORTED
    }
  if (field->packed)
    header_size += uint32_size (rv);
  return header_size + rv;
}

/* Get the packed size of a unknown field (meaning one that
   is passed through mostly uninterpreted... this is done
   for forward compatibilty with the addition of new fields). */
static inline size_t
unknown_field_get_packed_size (const ProtobufCMessageUnknownField *field)
{
  return get_tag_size (field->tag) + field->len;
}

/* Get the number of bytes that the message will occupy once serialized. */
size_t
protobuf_c_message_get_packed_size(const ProtobufCMessage *message)
{
  unsigned i;
  size_t rv = 0;
  ASSERT_IS_MESSAGE (message);
  for (i = 0; i < message->descriptor->n_fields; i++)
    {
      const ProtobufCFieldDescriptor *field = message->descriptor->fields + i;
      const void *member = ((const char *) message) + field->offset;
      const void *qmember = ((const char *) message) + field->quantifier_offset;

      if (field->label == PROTOBUF_C_LABEL_REQUIRED)
        rv += required_field_get_packed_size (field, member);
      else if (field->label == PROTOBUF_C_LABEL_OPTIONAL)
        rv += optional_field_get_packed_size (field, qmember, member);
      else
        rv += repeated_field_get_packed_size (field, * (const size_t *) qmember, member);
    }
  for (i = 0; i < message->n_unknown_fields; i++)
    rv += unknown_field_get_packed_size (&message->unknown_fields[i]);
  return rv;
}
/* === pack() === */
/* Pack an unsigned 32-bit integer in base-128 encoding, and return the number of bytes needed:
   this will be 5 or less. */
static inline size_t
uint32_pack (uint32_t value, uint8_t *out)
{
  unsigned rv = 0;
  if (value >= 0x80)
    {
      out[rv++] = value | 0x80;
      value >>= 7;
      if (value >= 0x80)
        {
          out[rv++] = value | 0x80;
          value >>= 7;
          if (value >= 0x80)
            {
              out[rv++] = value | 0x80;
              value >>= 7;
              if (value >= 0x80)
                {
                  out[rv++] = value | 0x80;
                  value >>= 7;
                }
            }
        }
    }
  /* assert: value<128 */
  out[rv++] = value;
  return rv;
}

/* Pack a 32-bit signed integer, returning the number of bytes needed.
   Negative numbers are packed as twos-complement 64-bit integers. */
static inline size_t
int32_pack (int32_t value, uint8_t *out)
{
  if (value < 0)
    {
      out[0] = value | 0x80;
      out[1] = (value>>7) | 0x80;
      out[2] = (value>>14) | 0x80;
      out[3] = (value>>21) | 0x80;
      out[4] = (value>>28) | 0x80;
      out[5] = out[6] = out[7] = out[8] = 0xff;
      out[9] = 0x01;
      return 10;
    }
  else
    return uint32_pack (value, out);
}

/* Pack a 32-bit integer in zigwag encoding. */
static inline size_t
```

```
sint32_pack (int32_t value, uint8_t *out)
{
  return uint32_pack (zigzag32 (value), out);
}

/* Pack a 64-bit unsigned integer that fits in a 64-bit uint,
   using base-128 encoding. */
static size_t
uint64_pack (uint64_t value, uint8_t *out)
{
  uint32_t hi = (uint32_t )(value>>32);
  uint32_t lo = (uint32_t )value;
  unsigned rv;
  if (hi == 0)
    return uint32_pack ((uint32_t)lo, out);
  out[0] = (lo) | 0x80;
  out[1] = (lo>>7) | 0x80;
  out[2] = (lo>>14) | 0x80;
  out[3] = (lo>>21) | 0x80;
  if (hi < 8)
    {
      out[4] = (hi<<4) | (lo>>28);
      return 5;
    }
  else
    {
      out[4] = ((hi&7)<<4) | (lo>>28) | 0x80;
      hi >>= 3;
    }
  rv = 5;
  while (hi >= 128)
    {
      out[rv++] = hi | 0x80;
      hi >>= 7;
    }
  out[rv++] = hi;
  return rv;
}

/* Pack a 64-bit signed integer in zigzan encoding,
   return the size of the packed output.
   (Max returned value is 10) */
static inline size_t
sint64_pack (int64_t value, uint8_t *out)
{
  return uint64_pack (zigzag64 (value), out);
}

/* Pack a 32-bit value, little-endian.
   Used for fixed32, sfixed32, float) */
static inline size_t
fixed32_pack (uint32_t value, void *out)
{
#if IS_LITTLE_ENDIAN
  memcpy (out, &value, 4);
#else
  uint8_t *buf = out;
  buf[0] = value;
  buf[1] = value>>8;
  buf[2] = value>>16;
  buf[3] = value>>24;
#endif
  return 4;
}

/* Pack a 64-bit fixed-length value.
   (Used for fixed64, sfixed64, double) */
/* XXX: the big-endian impl is really only good for 32-bit machines,
   a 64-bit version would be appreciated, plus a way
   to decide to use 64-bit math where convenient. */
static inline size_t
fixed64_pack (uint64_t value, void *out)
{
#if IS_LITTLE_ENDIAN
  memcpy (out, &value, 8);
#else
  fixed32_pack (value, out);
  fixed32_pack (value>>32, out+4);
#endif
  return 8;
}


/* Pack a boolean as 0 or 1, even though the protobuf_c_boolean
   can really assume any integer value. */
/* XXX: perhaps on some platforms "*out = !!value" would be
   a better impl, b/c that is idiotmatic c++ in some stl impls. */
static inline size_t
boolean_pack (protobuf_c_boolean value, uint8_t *out)
{
  *out = value ? 1 : 0;
  return 1;
}

/* Pack a length-prefixed string.
   The input string is NUL-terminated.

   The NULL pointer is treated as an empty string.
```

```
   This isn't really necessary, but it allows people
   to leave required strings blank.
   (See Issue 13 in the bug tracker for a
   little more explanation).
 */
static inline size_t
string_pack (const char * str, uint8_t *out)
{
  if (str == NULL)
    {
      out[0] = 0;
      return 1;
    }
  else
    {
      size_t len = strlen (str);
      size_t rv = uint32_pack (len, out);
      memcpy (out + rv, str, len);
      return rv + len;
    }
}

static inline size_t
binary_data_pack (const ProtobufCBinaryData *bd, uint8_t *out)
{
  size_t len = bd->len;
  size_t rv = uint32_pack (len, out);
  memcpy (out + rv, bd->data, len);
  return rv + len;
}

static inline size_t
prefixed_message_pack (const ProtobufCMessage *message, uint8_t *out)
{
  if (message == NULL)
    {
      out[0] = 0;
      return 1;
    }
  else
    {
      size_t rv = protobuf_c_message_pack (message, out + 1);
      uint32_t rv_packed_size = uint32_size (rv);
      if (rv_packed_size != 1)
        memmove (out + rv_packed_size, out + 1, rv);
      return uint32_pack (rv, out) + rv;
    }
}

/* wire-type will be added in required_field_pack() */
/* XXX: just call uint64_pack on 64-bit platforms. */
static size_t
tag_pack (uint32_t id, uint8_t *out)
{
  if (id < (1<<(32-3)))
    return uint32_pack (id<<3, out);
  else
    return uint64_pack (((uint64_t)id) << 3, out);
}

static size_t
required_field_pack (const ProtobufCFieldDescriptor *field,
                     const void *member,
                     uint8_t *out)
{
  size_t rv = tag_pack (field->id, out);
  switch (field->type)
    {
    case PROTOBUF_C_TYPE_SINT32:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + sint32_pack (*(const int32_t *) member, out + rv);
    case PROTOBUF_C_TYPE_INT32:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + int32_pack (*(const uint32_t *) member, out + rv);
    case PROTOBUF_C_TYPE_UINT32:
    case PROTOBUF_C_TYPE_ENUM:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + uint32_pack (*(const uint32_t *) member, out + rv);
    case PROTOBUF_C_TYPE_SINT64:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + sint64_pack (*(const int64_t *) member, out + rv);
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + uint64_pack (*(const uint64_t *) member, out + rv);
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      out[0] |= PROTOBUF_C_WIRE_TYPE_32BIT;
      return rv + fixed32_pack (*(const uint32_t *) member, out + rv);
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      out[0] |= PROTOBUF_C_WIRE_TYPE_64BIT;
      return rv + fixed64_pack (*(const uint64_t *) member, out + rv);
    case PROTOBUF_C_TYPE_BOOL:
      out[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      return rv + boolean_pack (*(const protobuf_c_boolean *) member, out + rv);
```

```
    case PROTOBUF_C_TYPE_STRING:
      {
        out[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        return rv + string_pack (*(char * const *) member, out + rv);
      }

    case PROTOBUF_C_TYPE_BYTES:
      {
        const ProtobufCBinaryData * bd = ((const ProtobufCBinaryData*) member);
        out[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        return rv + binary_data_pack (bd, out + rv);
      }
    //case PROTOBUF_C_TYPE_GROUP:          // NOT SUPPORTED
    case PROTOBUF_C_TYPE_MESSAGE:
      {
        out[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        return rv + prefixed_message_pack (*(ProtobufCMessage * const *) member,
                                           out + rv);
      }
    }
  PROTOBUF_C_ASSERT_NOT_REACHED ();
  return 0;
}
static size_t
optional_field_pack (const ProtobufCFieldDescriptor *field,
                     const protobuf_c_boolean *has,
                     const void *member,
                     uint8_t *out)
{
  if (field->type == PROTOBUF_C_TYPE_MESSAGE
   || field->type == PROTOBUF_C_TYPE_STRING)
    {
      const void *ptr = * (const void * const *) member;
      if (ptr == NULL
       || ptr == field->default_value)
        return 0;
    }
  else
    {
      if (!*has)
        return 0;
    }
  return required_field_pack (field, member, out);
}

/* TODO: implement as a table lookup */
static inline size_t
sizeof_elt_in_repeated_array (ProtobufCType type)
{
  switch (type)
    {
    case PROTOBUF_C_TYPE_SINT32:
    case PROTOBUF_C_TYPE_INT32:
    case PROTOBUF_C_TYPE_UINT32:
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
    case PROTOBUF_C_TYPE_ENUM:
      return 4;
    case PROTOBUF_C_TYPE_SINT64:
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      return 8;
    case PROTOBUF_C_TYPE_BOOL:
      return sizeof (protobuf_c_boolean);
    case PROTOBUF_C_TYPE_STRING:
    case PROTOBUF_C_TYPE_MESSAGE:
      return sizeof (void *);
    case PROTOBUF_C_TYPE_BYTES:
      return sizeof (ProtobufCBinaryData);
    }
  PROTOBUF_C_ASSERT_NOT_REACHED ();
  return 0;
}

static void
copy_to_little_endian_32 (void *out, const void *in, unsigned N)
{
#if IS_LITTLE_ENDIAN
  memcpy (out, in, N * 4);
#else
  unsigned i;
  const uint32_t *ini = in;
  for (i = 0; i < N; i++)
    fixed32_pack (ini[i], (uint32_t*)out + i);
#endif
}
static void
copy_to_little_endian_64 (void *out, const void *in, unsigned N)
{
#if IS_LITTLE_ENDIAN
  memcpy (out, in, N * 8);
#else
  unsigned i;
  const uint64_t *ini = in;
```

```
  for (i = 0; i < N; i++)
    fixed64_pack (ini[i], (uint64_t*)out + i);
#endif
}

static unsigned
get_type_min_size (ProtobufCType type)
{
  if (type == PROTOBUF_C_TYPE_SFIXED32
   || type == PROTOBUF_C_TYPE_FIXED32
   || type == PROTOBUF_C_TYPE_FLOAT)
    return 4;
  if (type == PROTOBUF_C_TYPE_SFIXED64
   || type == PROTOBUF_C_TYPE_FIXED64
   || type == PROTOBUF_C_TYPE_DOUBLE)
    return 8;
  return 1;
}

static size_t
repeated_field_pack (const ProtobufCFieldDescriptor *field,
                     size_t count,
                     const void *member,
                     uint8_t *out)
{
  char *array = * (char * const *) member;
  unsigned i;
  if (field->packed)
    {
      unsigned header_len;
      unsigned len_start;
      unsigned min_length;
      unsigned payload_len;
      unsigned length_size_min;
      unsigned actual_length_size;
      uint8_t *payload_at;
      if (count == 0)
        return 0;
      header_len = tag_pack (field->id, out);
      out[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
      len_start = header_len;
      min_length = get_type_min_size (field->type) * count;
      length_size_min = uint32_size (min_length);
      header_len += length_size_min;
      payload_at = out + header_len;
      switch (field->type)
        {
        case PROTOBUF_C_TYPE_SFIXED32:
        case PROTOBUF_C_TYPE_FIXED32:
        case PROTOBUF_C_TYPE_FLOAT:
          copy_to_little_endian_32 (payload_at, array, count);
          payload_at += count * 4;
          break;

        case PROTOBUF_C_TYPE_SFIXED64:
        case PROTOBUF_C_TYPE_FIXED64:
        case PROTOBUF_C_TYPE_DOUBLE:
          copy_to_little_endian_64 (payload_at, array, count);
          payload_at += count * 8;
          break;

        case PROTOBUF_C_TYPE_INT32:
          {
            const int32_t *arr = (const int32_t *) array;
            for (i = 0; i < count; i++)
              payload_at += int32_pack (arr[i], payload_at);
          }
          break;

        case PROTOBUF_C_TYPE_SINT32:
          {
            const int32_t *arr = (const int32_t *) array;
            for (i = 0; i < count; i++)
              payload_at += sint32_pack (arr[i], payload_at);
          }
          break;

        case PROTOBUF_C_TYPE_SINT64:
          {
            const int64_t *arr = (const int64_t *) array;
            for (i = 0; i < count; i++)
              payload_at += sint64_pack (arr[i], payload_at);
          }
          break;
        case PROTOBUF_C_TYPE_ENUM:
        case PROTOBUF_C_TYPE_UINT32:
          {
            const uint32_t *arr = (const uint32_t *) array;
            for (i = 0; i < count; i++)
              payload_at += uint32_pack (arr[i], payload_at);
          }
          break;
        case PROTOBUF_C_TYPE_INT64:
        case PROTOBUF_C_TYPE_UINT64:
          {
            const uint64_t *arr = (const uint64_t *) array;
            for (i = 0; i < count; i++)
              payload_at += uint64_pack (arr[i], payload_at);
```

```
          }
          break;
        case PROTOBUF_C_TYPE_BOOL:
          {
            const protobuf_c_boolean *arr = (const protobuf_c_boolean *) array;
            for (i = 0; i < count; i++)
              payload_at += boolean_pack (arr[i], payload_at);
          }
          break;

        default:
          assert (0);
        }
      payload_len = payload_at - (out + header_len);
      actual_length_size = uint32_size (payload_len);
      if (length_size_min != actual_length_size)
        {
          assert (actual_length_size == length_size_min + 1);
          memmove (out + header_len + 1, out + header_len, payload_len);
          header_len++;
        }
      uint32_pack (payload_len, out + len_start);
      return header_len + payload_len;
    }
  else
    {
      /* CONSIDER: optimize this case a bit (by putting the loop inside the switch) */
      size_t rv = 0;
      unsigned siz = sizeof_elt_in_repeated_array (field->type);
      for (i = 0; i < count; i++)
        {
          rv += required_field_pack (field, array, out + rv);
          array += siz;
        }
      return rv;
    }
}
static size_t
unknown_field_pack (const ProtobufCMessageUnknownField *field,
                    uint8_t *out)
{
  size_t rv = tag_pack (field->tag, out);
  out[0] |= field->wire_type;
  memcpy (out + rv, field->data, field->len);
  return rv + field->len;
}

size_t
protobuf_c_message_pack           (const ProtobufCMessage *message,
                                   uint8_t                *out)
{
  unsigned i;
  size_t rv = 0;
  ASSERT_IS_MESSAGE (message);
  for (i = 0; i < message->descriptor->n_fields; i++)
    {
      const ProtobufCFieldDescriptor *field = message->descriptor->fields + i;
      const void *member = ((const char *) message) + field->offset;

      /* it doesn't hurt to compute qmember (a pointer to the quantifier
         field of the structure), but the pointer is only valid if
         the field is one of:
           - a repeated field
           - an optional field that isn't a pointer type
             (meaning: not a message or a string) */
      const void *qmember = ((const char *) message) + field->quantifier_offset;

      if (field->label == PROTOBUF_C_LABEL_REQUIRED)
        rv += required_field_pack (field, member, out + rv);
      else if (field->label == PROTOBUF_C_LABEL_OPTIONAL)
        /* note that qmember is bogus for strings and messages,
           but it isn't used */
        rv += optional_field_pack (field, qmember, member, out + rv);
      else
        rv += repeated_field_pack (field, * (const size_t *) qmember, member, out + rv);
    }
  for (i = 0; i < message->n_unknown_fields; i++)
    rv += unknown_field_pack (&message->unknown_fields[i], out + rv);
  return rv;
}

/* === pack_to_buffer() === */
static size_t
required_field_pack_to_buffer (const ProtobufCFieldDescriptor *field,
                               const void *member,
                               ProtobufCBuffer *buffer)
{
  size_t rv;
  uint8_t scratch[MAX_UINT64_ENCODED_SIZE * 2];
  rv = tag_pack (field->id, scratch);
  switch (field->type)
    {
    case PROTOBUF_C_TYPE_SINT32:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += sint32_pack (*(const int32_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_INT32:
```

```
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += int32_pack (*(const uint32_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_UINT32:
    case PROTOBUF_C_TYPE_ENUM:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += uint32_pack (*(const uint32_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_SINT64:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += sint64_pack (*(const int64_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += uint64_pack (*(const uint64_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_32BIT;
      rv += fixed32_pack (*(const uint32_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_64BIT;
      rv += fixed64_pack (*(const uint64_t *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_BOOL:
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_VARINT;
      rv += boolean_pack (*(const protobuf_c_boolean *) member, scratch + rv);
      buffer->append (buffer, rv, scratch);
      break;
    case PROTOBUF_C_TYPE_STRING:
      {
        const char *str = *(char * const *) member;
        size_t sublen = str ? strlen (str) : 0;
        scratch[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        rv += uint32_pack (sublen, scratch + rv);
        buffer->append (buffer, rv, scratch);
        buffer->append (buffer, sublen, (const uint8_t *) str);
        rv += sublen;
        break;
      }

    case PROTOBUF_C_TYPE_BYTES:
      {
        const ProtobufCBinaryData * bd = ((const ProtobufCBinaryData*) member);
        size_t sublen = bd->len;
        scratch[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        rv += uint32_pack (sublen, scratch + rv);
        buffer->append (buffer, rv, scratch);
        buffer->append (buffer, sublen, bd->data);
        rv += sublen;
        break;
      }
    //PROTOBUF_C_TYPE_GROUP,          // NOT SUPPORTED
    case PROTOBUF_C_TYPE_MESSAGE:
      {
        uint8_t simple_buffer_scratch[256];
        size_t sublen;
        ProtobufCBufferSimple simple_buffer
          = PROTOBUF_C_BUFFER_SIMPLE_INIT (simple_buffer_scratch);
        const ProtobufCMessage *msg = *(ProtobufCMessage * const *) member;
        scratch[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
        if (msg == NULL)
          sublen = 0;
        else
          sublen = protobuf_c_message_pack_to_buffer (msg, &simple_buffer.base);
        rv += uint32_pack (sublen, scratch + rv);
        buffer->append (buffer, rv, scratch);
        buffer->append (buffer, sublen, simple_buffer.data);
        rv += sublen;
        PROTOBUF_C_BUFFER_SIMPLE_CLEAR (&simple_buffer);
        break;
      }
    default:
      PROTOBUF_C_ASSERT_NOT_REACHED ();
    }
  return rv;
}
static size_t
optional_field_pack_to_buffer (const ProtobufCFieldDescriptor *field,
                               const protobuf_c_boolean *has,
                               const void *member,
                               ProtobufCBuffer *buffer)
{
  if (field->type == PROTOBUF_C_TYPE_MESSAGE
   || field->type == PROTOBUF_C_TYPE_STRING)
    {
      const void *ptr = * (const void * const *) member;
```

```
      if (ptr == NULL
       || ptr == field->default_value)
        return 0;
    }
  else
    {
      if (!*has)
        return 0;
    }
  return required_field_pack_to_buffer (field, member, buffer);
}

static size_t
get_packed_payload_length (const ProtobufCFieldDescriptor *field,
                           unsigned count,
                           const void *array)
{
  unsigned rv = 0;
  unsigned i;
  switch (field->type)
    {
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      return count * 4;

    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      return count * 8;

    case PROTOBUF_C_TYPE_INT32:
      {
        const int32_t *arr = (const int32_t *) array;
        for (i = 0; i < count; i++)
          rv += int32_size (arr[i]);
      }
      break;

    case PROTOBUF_C_TYPE_SINT32:
      {
        const int32_t *arr = (const int32_t *) array;
        for (i = 0; i < count; i++)
          rv += sint32_size (arr[i]);
      }
      break;
    case PROTOBUF_C_TYPE_ENUM:
    case PROTOBUF_C_TYPE_UINT32:
      {
        const uint32_t *arr = (const uint32_t *) array;
        for (i = 0; i < count; i++)
          rv += uint32_size (arr[i]);
      }
      break;

    case PROTOBUF_C_TYPE_SINT64:
      {
        const int64_t *arr = (const int64_t *) array;
        for (i = 0; i < count; i++)
          rv += sint64_size (arr[i]);
      }
      break;
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      {
        const uint64_t *arr = (const uint64_t *) array;
        for (i = 0; i < count; i++)
          rv += uint64_size (arr[i]);
      }
      break;
    case PROTOBUF_C_TYPE_BOOL:
      return count;
    default:
      assert (0);
    }
  return rv;
}
static size_t
pack_buffer_packed_payload (const ProtobufCFieldDescriptor *field,
                            unsigned count,
                            const void *array,
                            ProtobufCBuffer *buffer)
{
  uint8_t scratch[16];
  size_t rv = 0;
  unsigned i;
  switch (field->type)
    {
      case PROTOBUF_C_TYPE_SFIXED32:
      case PROTOBUF_C_TYPE_FIXED32:
      case PROTOBUF_C_TYPE_FLOAT:
#if IS_LITTLE_ENDIAN
        rv = count * 4;
        goto no_packing_needed;
#else
        for (i = 0; i < count; i++)
          {
            unsigned len = fixed32_pack (((uint32_t*)array)[i], scratch);
```

```
            buffer->append (buffer, len, scratch);
            rv += len;
          }
#endif
        break;
      case PROTOBUF_C_TYPE_SFIXED64:
      case PROTOBUF_C_TYPE_FIXED64:
      case PROTOBUF_C_TYPE_DOUBLE:
#if IS_LITTLE_ENDIAN
        rv = count * 8;
        goto no_packing_needed;
#else
        for (i = 0; i < count; i++)
          {
            unsigned len = fixed64_pack (((uint64_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;
#endif
      case PROTOBUF_C_TYPE_INT32:
        for (i = 0; i < count; i++)
          {
            unsigned len = int32_pack (((int32_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;

      case PROTOBUF_C_TYPE_SINT32:
        for (i = 0; i < count; i++)
          {
            unsigned len = sint32_pack (((int32_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;
      case PROTOBUF_C_TYPE_ENUM:
      case PROTOBUF_C_TYPE_UINT32:
        for (i = 0; i < count; i++)
          {
            unsigned len = uint32_pack (((uint32_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;

      case PROTOBUF_C_TYPE_SINT64:
        for (i = 0; i < count; i++)
          {
            unsigned len = sint64_pack (((int64_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;
      case PROTOBUF_C_TYPE_INT64:
      case PROTOBUF_C_TYPE_UINT64:
        for (i = 0; i < count; i++)
          {
            unsigned len = uint64_pack (((uint64_t*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        break;
      case PROTOBUF_C_TYPE_BOOL:
        for (i = 0; i < count; i++)
          {
            unsigned len = boolean_pack (((protobuf_c_boolean*)array)[i], scratch);
            buffer->append (buffer, len, scratch);
            rv += len;
          }
        return count;
      default:
        assert(0);
    }
  return rv;

no_packing_needed:
  buffer->append (buffer, rv, array);
  return rv;
}

static size_t
repeated_field_pack_to_buffer (const ProtobufCFieldDescriptor *field,
                               unsigned count,
                               const void *member,
                               ProtobufCBuffer *buffer)
{
  char *array = * (char * const *) member;
  if (count == 0)
    return 0;
  if (field->packed)
    {
      uint8_t scratch[MAX_UINT64_ENCODED_SIZE * 2];
      size_t rv = tag_pack (field->id, scratch);
      size_t payload_len = get_packed_payload_length (field, count, array);
      size_t tmp;
      scratch[0] |= PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED;
```

```
      rv += uint32_pack (payload_len, scratch + rv);
      buffer->append (buffer, rv, scratch);
      tmp = pack_buffer_packed_payload (field, count, array, buffer);
      assert (tmp == payload_len);
      return rv + payload_len;
    }
  else
    {
      size_t siz;
      unsigned i;
      /* CONSIDER: optimize this case a bit (by putting the loop inside the switch) */
      unsigned rv = 0;
      siz = sizeof_elt_in_repeated_array (field->type);
      for (i = 0; i < count; i++)
        {
          rv += required_field_pack_to_buffer (field, array, buffer);
          array += siz;
        }
      return rv;
    }
}

static size_t
unknown_field_pack_to_buffer (const ProtobufCMessageUnknownField *field,
                              ProtobufCBuffer *buffer)
{
  uint8_t header[MAX_UINT64_ENCODED_SIZE];
  size_t rv = tag_pack (field->tag, header);
  header[0] |= field->wire_type;
  buffer->append (buffer, rv, header);
  buffer->append (buffer, field->len, field->data);
  return rv + field->len;
}

size_t
protobuf_c_message_pack_to_buffer (const ProtobufCMessage *message,
                                   ProtobufCBuffer  *buffer)
{
  unsigned i;
  size_t rv = 0;
  ASSERT_IS_MESSAGE (message);
  for (i = 0; i < message->descriptor->n_fields; i++)
    {
      const ProtobufCFieldDescriptor *field = message->descriptor->fields + i;
      const void *member = ((const char *) message) + field->offset;
      const void *qmember = ((const char *) message) + field->quantifier_offset;

      if (field->label == PROTOBUF_C_LABEL_REQUIRED)
        rv += required_field_pack_to_buffer (field, member, buffer);
      else if (field->label == PROTOBUF_C_LABEL_OPTIONAL)
        rv += optional_field_pack_to_buffer (field, qmember, member, buffer);
      else
        rv += repeated_field_pack_to_buffer (field, * (const size_t *) qmember, member, buffer);
    }
  for (i = 0; i < message->n_unknown_fields; i++)
    rv += unknown_field_pack_to_buffer (&message->unknown_fields[i], buffer);

  return rv;
}

/* === unpacking === */
#if PRINT_UNPACK_ERRORS
# define UNPACK_ERROR(args)  do { printf args;printf("\n"); }while(0)
#else
# define UNPACK_ERROR(args)  do { } while (0)
#endif

static inline int
int_range_lookup (unsigned n_ranges,
                  const ProtobufCIntRange *ranges,
                  int value)
{
  unsigned start, n;
  if (n_ranges == 0)
    return -1;
  start = 0;
  n = n_ranges;
  while (n > 1)
    {
      unsigned mid = start + n / 2;
      if (value < ranges[mid].start_value)
        {
          n = mid - start;
        }
      else if (value >= ranges[mid].start_value + (int)(ranges[mid+1].orig_index-ranges[mid].orig_index))
        {
          unsigned new_start = mid + 1;
          n = start + n - new_start;
          start = new_start;
        }
      else
        return (value - ranges[mid].start_value) + ranges[mid].orig_index;
    }
  if (n > 0)
    {
      unsigned start_orig_index = ranges[start].orig_index;
      unsigned range_size = ranges[start+1].orig_index - start_orig_index;
```

```
      if (ranges[start].start_value <= value
       && value < (int)(ranges[start].start_value + range_size))
        return (value - ranges[start].start_value) + start_orig_index;
    }
  return -1;
}

static size_t
parse_tag_and_wiretype (size_t len,
                        const uint8_t *data,
                        uint32_t *tag_out,
                        ProtobufCWireType *wiretype_out)
{
  unsigned max_rv = len > 5 ? 5 : len;
  uint32_t tag = (data[0]&0x7f) >> 3;
  unsigned shift = 4;
  unsigned rv;
  *wiretype_out = data[0] & 7;
  if ((data[0] & 0x80) == 0)
    {
      *tag_out = tag;
      return 1;
    }
  for (rv = 1; rv < max_rv; rv++)
    if (data[rv] & 0x80)
      {
        tag |= (data[rv] & 0x7f) << shift;
        shift += 7;
      }
    else
      {
        tag |= data[rv] << shift;
        *tag_out = tag;
        return rv + 1;
      }
  return 0;                     /* error: bad header */
}

/* sizeof(ScannedMember) must be <= (1<<BOUND_SIZEOF_SCANNED_MEMBER_LOG2) */
#define BOUND_SIZEOF_SCANNED_MEMBER_LOG2  5
typedef struct _ScannedMember ScannedMember;
struct _ScannedMember
{
  uint32_t tag;
  uint8_t wire_type;
  uint8_t length_prefix_len;
  const ProtobufCFieldDescriptor *field;
  size_t len;
  const uint8_t *data;
};

static inline uint32_t
scan_length_prefixed_data (size_t len, const uint8_t *data, size_t *prefix_len_out)
{
  unsigned hdr_max = len < 5 ? len : 5;
  unsigned hdr_len;
  uint32_t val = 0;
  unsigned i;
  unsigned shift = 0;
  for (i = 0; i < hdr_max; i++)
    {
      val |= (data[i] & 0x7f) << shift;
      shift += 7;
      if ((data[i] & 0x80) == 0)
        break;
    }
  if (i == hdr_max)
    {
      UNPACK_ERROR (("error parsing length for length-prefixed data"));
      return 0;
    }
  hdr_len = i + 1;
  *prefix_len_out = hdr_len;
  if (hdr_len + val > len)
    {
      UNPACK_ERROR (("data too short after length-prefix of %u",
                     val));
      return 0;
    }
  return hdr_len + val;
}

static size_t
max_b128_numbers (size_t len, const uint8_t *data)
{
  size_t rv = 0;
  while (len--)
    if ((*data++ & 0x80) == 0)
      ++rv;
  return rv;
}


/* Given a raw slab of packed-repeated values,
   determine the number of elements.
   This function detects certain kinds of errors
   but not others; the remaining error checking is done by
   parse_packed_repeated_member() */
```

```
static protobuf_c_boolean
count_packed_elements (ProtobufCType type,
                       size_t len,
                       const uint8_t *data,
                       size_t *count_out)
{
  switch (type)
    {
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      if (len % 4 != 0)
        {
          UNPACK_ERROR (("length must be a multiple of 4 for fixed-length 32-bit types"));
          return FALSE;
        }
      *count_out = len / 4;
      return TRUE;

    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      if (len % 8 != 0)
        {
          UNPACK_ERROR (("length must be a multiple of 8 for fixed-length 64-bit types"));
          return FALSE;
        }
      *count_out = len / 8;
      return TRUE;

    case PROTOBUF_C_TYPE_INT32:
    case PROTOBUF_C_TYPE_SINT32:
    case PROTOBUF_C_TYPE_ENUM:
    case PROTOBUF_C_TYPE_UINT32:
    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_SINT64:
    case PROTOBUF_C_TYPE_UINT64:
      *count_out = max_b128_numbers (len, data);
      return TRUE;
    case PROTOBUF_C_TYPE_BOOL:
      *count_out = len;
      return TRUE;

    case PROTOBUF_C_TYPE_STRING:
    case PROTOBUF_C_TYPE_BYTES:
    case PROTOBUF_C_TYPE_MESSAGE:
    default:
      UNPACK_ERROR (("bad protobuf-c type %u for packed-repeated", type));
      return FALSE;
    }
}

static inline uint32_t
parse_uint32 (unsigned len, const uint8_t *data)
{
  unsigned rv = data[0] & 0x7f;
  if (len > 1)
    {
      rv |= ((data[1] & 0x7f) << 7);
      if (len > 2)
        {
          rv |= ((data[2] & 0x7f) << 14);
          if (len > 3)
            {
              rv |= ((data[3] & 0x7f) << 21);
              if (len > 4)
                rv |= (data[4] << 28);
            }
        }
    }
  return rv;
}
static inline uint32_t
parse_int32 (unsigned len, const uint8_t *data)
{
  return parse_uint32 (len, data);
}
static inline int32_t
unzigzag32 (uint32_t v)
{
  if (v&1)
    return -(v>>1) - 1;
  else
    return v>>1;
}
static inline uint32_t
parse_fixed_uint32 (const uint8_t *data)
{
#if IS_LITTLE_ENDIAN
  uint32_t t;
  memcpy (&t, data, 4);
  return t;
#else
  return data[0] | (data[1] << 8) | (data[2] << 16) | (data[3] << 24);
#endif
}
static uint64_t
parse_uint64 (unsigned len, const uint8_t *data)
```

```
{
  unsigned shift, i;
  uint64_t rv;
  if (len < 5)
    return parse_uint32 (len, data);
  rv = ((data[0] & 0x7f))
             | ((data[1] & 0x7f)<<7)
             | ((data[2] & 0x7f)<<14)
             | ((data[3] & 0x7f)<<21);
  shift = 28;
  for (i = 4; i < len; i++)
    {
      rv |= (((uint64_t)(data[i]&0x7f)) << shift);
      shift += 7;
    }
  return rv;
}
static inline int64_t
unzigzag64 (uint64_t v)
{
  if (v&1)
    return -(v>>1) - 1;
  else
    return v>>1;
}
static inline uint64_t
parse_fixed_uint64 (const uint8_t *data)
{
#if IS_LITTLE_ENDIAN
  uint64_t t;
  memcpy (&t, data, 8);
  return t;
#else
  return (uint64_t)parse_fixed_uint32 (data)
      | (((uint64_t)parse_fixed_uint32(data+4)) << 32);
#endif
}
static protobuf_c_boolean
parse_boolean (unsigned len, const uint8_t *data)
{
  unsigned i;
  for (i = 0; i < len; i++)
    if (data[i] & 0x7f)
      return 1;
  return 0;
}
static protobuf_c_boolean
parse_required_member (ScannedMember *scanned_member,
                       void *member,
                       ProtobufCAllocator *allocator,
                       protobuf_c_boolean maybe_clear)
{
  unsigned len = scanned_member->len;
  const uint8_t *data = scanned_member->data;
  ProtobufCWireType wire_type = scanned_member->wire_type;
  switch (scanned_member->field->type)
    {
    case PROTOBUF_C_TYPE_INT32:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(uint32_t*)member = parse_int32 (len, data);
      return 1;
    case PROTOBUF_C_TYPE_UINT32:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(uint32_t*)member = parse_uint32 (len, data);
      return 1;
    case PROTOBUF_C_TYPE_SINT32:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(int32_t*)member = unzigzag32 (parse_uint32 (len, data));
      return 1;
    case PROTOBUF_C_TYPE_SFIXED32:
    case PROTOBUF_C_TYPE_FIXED32:
    case PROTOBUF_C_TYPE_FLOAT:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_32BIT)
        return 0;
      *(uint32_t*)member = parse_fixed_uint32 (data);
      return 1;

    case PROTOBUF_C_TYPE_INT64:
    case PROTOBUF_C_TYPE_UINT64:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(uint64_t*)member = parse_uint64 (len, data);
      return 1;
    case PROTOBUF_C_TYPE_SINT64:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(int64_t*)member = unzigzag64 (parse_uint64 (len, data));
      return 1;
    case PROTOBUF_C_TYPE_SFIXED64:
    case PROTOBUF_C_TYPE_FIXED64:
    case PROTOBUF_C_TYPE_DOUBLE:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_64BIT)
        return 0;
      *(uint64_t*)member = parse_fixed_uint64 (data);
      return 1;
```

```
    case PROTOBUF_C_TYPE_BOOL:
      *(protobuf_c_boolean*)member = parse_boolean (len, data);
      return 1;

    case PROTOBUF_C_TYPE_ENUM:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_VARINT)
        return 0;
      *(uint32_t*)member = parse_uint32 (len, data);
      return 1;

    case PROTOBUF_C_TYPE_STRING:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED)
        return 0;
      {
        char **pstr = member;
        unsigned pref_len = scanned_member->length_prefix_len;
        if (maybe_clear && *pstr != NULL)
          {
            const char *def = scanned_member->field->default_value;
            if (*pstr != NULL && *pstr != def)
              FREE (allocator, *pstr);
          }
        DO_ALLOC (*pstr, allocator, len - pref_len + 1, return 0);
        memcpy (*pstr, data + pref_len, len - pref_len);
        (*pstr)[len-pref_len] = 0;
        return 1;
      }
    case PROTOBUF_C_TYPE_BYTES:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED)
        return 0;
      {
        ProtobufCBinaryData *bd = member;
        const ProtobufCBinaryData *def_bd;
        unsigned pref_len = scanned_member->length_prefix_len;
        def_bd = scanned_member->field->default_value;
        if (maybe_clear
         && bd->data != NULL
         && (def_bd == NULL || bd->data != def_bd->data))
          FREE (allocator, bd->data);
        DO_ALLOC (bd->data, allocator, len - pref_len, return 0);
        memcpy (bd->data, data + pref_len, len - pref_len);
        bd->len = len - pref_len;
        return 1;
      }
    //case PROTOBUF_C_TYPE_GROUP,          // NOT SUPPORTED
    case PROTOBUF_C_TYPE_MESSAGE:
      if (wire_type != PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED)
        return 0;
      {
        ProtobufCMessage **pmessage = member;
        ProtobufCMessage *subm;
        const ProtobufCMessage *def_mess;
        unsigned pref_len = scanned_member->length_prefix_len;
        def_mess = scanned_member->field->default_value;
        if (maybe_clear && *pmessage != NULL && *pmessage != def_mess)
          protobuf_c_message_free_unpacked (*pmessage, allocator);
        subm = protobuf_c_message_unpack (scanned_member->field->descriptor,
                                          allocator,
                                          len - pref_len, data + pref_len);
        *pmessage = subm;       /* since we freed the message we must clear the field, even if NULL */
        if (subm == NULL)
          return 0;
        return 1;
      }
    }
  return 0;
}

static protobuf_c_boolean
parse_optional_member (ScannedMember *scanned_member,
                       void *member,
                       ProtobufCMessage *message,
                       ProtobufCAllocator *allocator)
{
  if (!parse_required_member (scanned_member, member, allocator, TRUE))
    return 0;
  if (scanned_member->field->quantifier_offset != 0)
    STRUCT_MEMBER (protobuf_c_boolean,
                   message,
                   scanned_member->field->quantifier_offset) = 1;
  return 1;
}

static protobuf_c_boolean
parse_repeated_member (ScannedMember *scanned_member,
                       void *member,
                       ProtobufCMessage *message,
                       ProtobufCAllocator *allocator)
{
  const ProtobufCFieldDescriptor *field = scanned_member->field;
  size_t *p_n = STRUCT_MEMBER_PTR(size_t, message, field->quantifier_offset);
  size_t siz = sizeof_elt_in_repeated_array (field->type);
  char *array = *(char**)member;
  if (!parse_required_member (scanned_member,
                              array + siz * (*p_n),
                              allocator,
                              FALSE))
```

```
    return 0;
  *p_n += 1;
  return 1;
}

static unsigned scan_varint (unsigned len, const uint8_t *data)
{
  unsigned i;
  if (len > 10)
    len = 10;
  for (i = 0; i < len; i++)
    if ((data[i] & 0x80) == 0)
      break;
  if (i == len)
    return 0;
  return i + 1;
}

static protobuf_c_boolean
parse_packed_repeated_member (ScannedMember *scanned_member,
                              void *member,
                              ProtobufCMessage *message)
{
  const ProtobufCFieldDescriptor *field = scanned_member->field;
  size_t *p_n = STRUCT_MEMBER_PTR(size_t, message, field->quantifier_offset);
  size_t siz = sizeof_elt_in_repeated_array (field->type);
  char *array = *(char**)member + siz * (*p_n);
  const uint8_t *at = scanned_member->data + scanned_member->length_prefix_len;
  size_t rem = scanned_member->len - scanned_member->length_prefix_len;
  size_t count = 0;
  unsigned i;
  switch (field->type)
    {
      case PROTOBUF_C_TYPE_SFIXED32:
      case PROTOBUF_C_TYPE_FIXED32:
      case PROTOBUF_C_TYPE_FLOAT:
        count = (scanned_member->len - scanned_member->length_prefix_len) / 4;
#if IS_LITTLE_ENDIAN
        goto no_unpacking_needed;
#else
        for (i = 0; i < count; i++)
          {
            ((uint32_t*)array)[i] = parse_fixed_uint32 (at);
            at += 4;
          }
#endif
        break;
      case PROTOBUF_C_TYPE_SFIXED64:
      case PROTOBUF_C_TYPE_FIXED64:
      case PROTOBUF_C_TYPE_DOUBLE:
        count = (scanned_member->len - scanned_member->length_prefix_len) / 8;
#if IS_LITTLE_ENDIAN
        goto no_unpacking_needed;
#else
        for (i = 0; i < count; i++)
          {
            ((uint64_t*)array)[i] = parse_fixed_uint64 (at);
            at += 8;
          }
        break;
#endif
      case PROTOBUF_C_TYPE_INT32:
        while (rem > 0)
          {
            unsigned s = scan_varint (rem, at);
            if (s == 0)
              {
                UNPACK_ERROR (("bad packed-repeated int32 value"));
                return FALSE;
              }
            ((int32_t*)array)[count++] = parse_int32 (s, at);
            at += s;
            rem -= s;
          }
        break;

      case PROTOBUF_C_TYPE_SINT32:
        while (rem > 0)
          {
            unsigned s = scan_varint (rem, at);
            if (s == 0)
              {
                UNPACK_ERROR (("bad packed-repeated sint32 value"));
                return FALSE;
              }
            ((int32_t*)array)[count++] = unzigzag32 (parse_uint32 (s, at));
            at += s;
            rem -= s;
          }
        break;
      case PROTOBUF_C_TYPE_ENUM:
      case PROTOBUF_C_TYPE_UINT32:
        while (rem > 0)
          {
            unsigned s = scan_varint (rem, at);
            if (s == 0)
              {
                UNPACK_ERROR (("bad packed-repeated enum or uint32 value"));
```

```
                return FALSE;
              }
            ((uint32_t*)array)[count++] = parse_uint32 (s, at);
            at += s;
            rem -= s;
          }
        break;

      case PROTOBUF_C_TYPE_SINT64:
        while (rem > 0)
          {
            unsigned s = scan_varint (rem, at);
            if (s == 0)
              {
                UNPACK_ERROR (("bad packed-repeated sint64 value"));
                return FALSE;
              }
            ((int64_t*)array)[count++] = unzigzag64 (parse_uint64 (s, at));
            at += s;
            rem -= s;
          }
        break;
      case PROTOBUF_C_TYPE_INT64:
      case PROTOBUF_C_TYPE_UINT64:
        while (rem > 0)
          {
            unsigned s = scan_varint (rem, at);
            if (s == 0)
              {
                UNPACK_ERROR (("bad packed-repeated int64/uint64 value"));
                return FALSE;
              }
            ((int64_t*)array)[count++] = parse_uint64 (s, at);
            at += s;
            rem -= s;
          }
        break;
      case PROTOBUF_C_TYPE_BOOL:
        count = rem;
        for (i = 0; i < count; i++)
          {
            if (at[i] > 1)
              {
                UNPACK_ERROR (("bad packed-repeated boolean value"));
                return FALSE;
              }
            ((protobuf_c_boolean*)array)[i] = at[i];
          }
        break;
      default:
        assert(0);
    }
  *p_n += count;
  return TRUE;

no_unpacking_needed:
  memcpy (array, at, count * siz);
  *p_n += count;
  return TRUE;
}

static protobuf_c_boolean
parse_member (ScannedMember *scanned_member,
              ProtobufCMessage *message,
              ProtobufCAllocator *allocator)
{
  const ProtobufCFieldDescriptor *field = scanned_member->field;
  void *member;
  if (field == NULL)
    {
      ProtobufCMessageUnknownField *ufield = message->unknown_fields + (message->n_unknown_fields++);
      ufield->tag = scanned_member->tag;
      ufield->wire_type = scanned_member->wire_type;
      ufield->len = scanned_member->len;
      DO_UNALIGNED_ALLOC (ufield->data, allocator, scanned_member->len, return 0);
      memcpy (ufield->data, scanned_member->data, ufield->len);
      return 1;
    }
  member = (char*)message + field->offset;
  switch (field->label)
    {
    case PROTOBUF_C_LABEL_REQUIRED:
      return parse_required_member (scanned_member, member, allocator, TRUE);
    case PROTOBUF_C_LABEL_OPTIONAL:
      return parse_optional_member (scanned_member, member, message, allocator);
    case PROTOBUF_C_LABEL_REPEATED:
      if (field->packed
       && scanned_member->wire_type == PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED)
        return parse_packed_repeated_member (scanned_member, member, message);
      else
        return parse_repeated_member (scanned_member, member, message, allocator);
    }
  PROTOBUF_C_ASSERT_NOT_REACHED ();
  return 0;
}


/* TODO: expose/use this function if desc->message_init==NULL
```

```
   (which occurs for old code, and may be useful for certain
   programatic techniques for generating descriptors). */
void
protobuf_c_message_init_generic (const ProtobufCMessageDescriptor *desc,
                                 ProtobufCMessage *message)
{
  unsigned i;
  memset (message, 0, desc->sizeof_message);
  message->descriptor = desc;
  for (i = 0; i < desc->n_fields; i++)
    if (desc->fields[i].default_value != NULL
     && desc->fields[i].label != PROTOBUF_C_LABEL_REPEATED)
      {
        void *field = STRUCT_MEMBER_P (message, desc->fields[i].offset);
        const void *dv = desc->fields[i].default_value;
        switch (desc->fields[i].type)
        {
        case PROTOBUF_C_TYPE_INT32:
        case PROTOBUF_C_TYPE_SINT32:
        case PROTOBUF_C_TYPE_SFIXED32:
        case PROTOBUF_C_TYPE_UINT32:
        case PROTOBUF_C_TYPE_FIXED32:
        case PROTOBUF_C_TYPE_FLOAT:
        case PROTOBUF_C_TYPE_ENUM:
          memcpy (field, dv, 4);
          break;

        case PROTOBUF_C_TYPE_INT64:
        case PROTOBUF_C_TYPE_SINT64:
        case PROTOBUF_C_TYPE_SFIXED64:
        case PROTOBUF_C_TYPE_UINT64:
        case PROTOBUF_C_TYPE_FIXED64:
        case PROTOBUF_C_TYPE_DOUBLE:
          memcpy (field, dv, 8);
          break;

        case PROTOBUF_C_TYPE_BOOL:
          memcpy (field, dv, sizeof (protobuf_c_boolean));
          break;

        case PROTOBUF_C_TYPE_BYTES:
          memcpy (field, dv, sizeof (ProtobufCBinaryData));
          break;

        case PROTOBUF_C_TYPE_STRING:
        case PROTOBUF_C_TYPE_MESSAGE:
          /* the next line essentially implements a cast from const,
             which is totally unavoidable. */
          *(const void**)field = dv;
          break;
        }
      }
}

/* ScannedMember slabs (an unpacking implementation detail).
   Before doing real unpacking, we first scan through the
   elements to see how many there are (for repeated fields),
   and which field to use (for non-repeated fields given twice).

 * In order to avoid allocations for small messages,
   we keep a stack-allocated slab of ScannedMembers of
   size FIRST_SCANNED_MEMBER_SLAB_SIZE (16).
   After we fill that up, we allocate each slab twice
   as large as the previous one. */
#define FIRST_SCANNED_MEMBER_SLAB_SIZE_LOG2             4

/* The number of slabs, including the stack-allocated ones;
   choose the number so that we would overflow if we needed
   a slab larger than provided. */
#define MAX_SCANNED_MEMBER_SLAB                          \
  (sizeof(void*)*8 - 1                                   \
   - BOUND_SIZEOF_SCANNED_MEMBER_LOG2                    \
   - FIRST_SCANNED_MEMBER_SLAB_SIZE_LOG2)

ProtobufCMessage *
protobuf_c_message_unpack         (const ProtobufCMessageDescriptor *desc,
                                   ProtobufCAllocator  *allocator,
                                   size_t               len,
                                   const uint8_t       *data)
{
  ProtobufCMessage *rv;
  size_t rem = len;
  const uint8_t *at = data;
  const ProtobufCFieldDescriptor *last_field = desc->fields + 0;
  ScannedMember first_member_slab[1<<FIRST_SCANNED_MEMBER_SLAB_SIZE_LOG2];

  /* scanned_member_slabs[i] is an array of arrays of ScannedMember.
     The first slab (scanned_member_slabs[0] is just a pointer to
     first_member_slab), above.  All subsequent slabs will be allocated
     using the allocator. */
  ScannedMember *scanned_member_slabs[MAX_SCANNED_MEMBER_SLAB+1];
  unsigned which_slab = 0;       /* the slab we are currently populating */
  unsigned in_slab_index = 0;    /* number of members in the slab */
  size_t n_unknown = 0;
  unsigned f;
  unsigned i_slab;
  unsigned last_field_index = 0;
  unsigned long *required_fields_bitmap;
```

```
unsigned required_fields_bitmap_len;
static const unsigned word_bits = sizeof(long) * 8;

ASSERT_IS_MESSAGE_DESCRIPTOR (desc);

if (allocator == NULL)
  allocator = &protobuf_c_default_allocator;

required_fields_bitmap_len = (desc->n_fields + word_bits - 1) / word_bits;
required_fields_bitmap = alloca(required_fields_bitmap_len * sizeof(long));
memset(required_fields_bitmap, 0, required_fields_bitmap_len * sizeof(long));

DO_ALLOC (rv, allocator, desc->sizeof_message, return NULL);
scanned_member_slabs[0] = first_member_slab;

/* Generated code always defines "message_init".
   However, we provide a fallback for (1) users of old protobuf-c
   generated-code that do not provide the function,
   and (2) descriptors constructed from some other source
   (most likely, direct construction from the .proto file) */
if (desc->message_init != NULL)
  protobuf_c_message_init (desc, rv);
else
  protobuf_c_message_init_generic (desc, rv);

while (rem > 0)
  {
    uint32_t tag;
    ProtobufCWireType wire_type;
    size_t used = parse_tag_and_wiretype (rem, at, &tag, &wire_type);
    const ProtobufCFieldDescriptor *field;
    ScannedMember tmp;
    if (used == 0)
      {
        UNPACK_ERROR (("error parsing tag/wiretype at offset %u",
                       (unsigned)(at-data)));
        goto error_cleanup_during_scan;
      }
    /* XXX: consider optimizing for field[1].id == tag, if field[1] exists! */
    if (last_field == NULL || last_field->id != tag)
      {
        /* lookup field */
        int field_index = int_range_lookup (desc->n_field_ranges,
                                            desc->field_ranges,
                                            tag);
        if (field_index < 0)
          {
            field = NULL;
            n_unknown++;
          }
        else
          {
            field = desc->fields + field_index;
            last_field = field;
            last_field_index = field_index;
          }
      }
    else
      field = last_field;

    if (field != NULL && field->label == PROTOBUF_C_LABEL_REQUIRED)
      required_fields_bitmap[last_field_index / word_bits] |= (1UL << (last_field_index % word_bits));

    at += used;
    rem -= used;
    tmp.tag = tag;
    tmp.wire_type = wire_type;
    tmp.field = field;
    tmp.data = at;
    switch (wire_type)
      {
      case PROTOBUF_C_WIRE_TYPE_VARINT:
        {
          unsigned max_len = rem < 10 ? rem : 10;
          unsigned i;
          for (i = 0; i < max_len; i++)
            if ((at[i] & 0x80) == 0)
              break;
          if (i == max_len)
            {
              UNPACK_ERROR (("unterminated varint at offset %u",
                             (unsigned)(at-data)));
              goto error_cleanup_during_scan;
            }
          tmp.len = i + 1;
        }
        break;
      case PROTOBUF_C_WIRE_TYPE_64BIT:
        if (rem < 8)
          {
            UNPACK_ERROR (("too short after 64bit wiretype at offset %u",
                           (unsigned)(at-data)));
            goto error_cleanup_during_scan;
          }
        tmp.len = 8;
        break;
      case PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED:
        {
```

```
            size_t pref_len;
            tmp.len = scan_length_prefixed_data (rem, at, &pref_len);
            if (tmp.len == 0)
              {
                /* NOTE: scan_length_prefixed_data calls UNPACK_ERROR */
                goto error_cleanup_during_scan;
              }
            tmp.length_prefix_len = pref_len;
            break;
          }
        case PROTOBUF_C_WIRE_TYPE_32BIT:
          if (rem < 4)
            {
              UNPACK_ERROR (("too short after 32bit wiretype at offset %u",
                             (unsigned)(at-data)));
              goto error_cleanup_during_scan;
            }
          tmp.len = 4;
          break;
        default:
          UNPACK_ERROR (("unsupported tag %u at offset %u",
                         wire_type, (unsigned)(at-data)));
          goto error_cleanup_during_scan;
        }
      if (in_slab_index == (1U<<(which_slab+FIRST_SCANNED_MEMBER_SLAB_SIZE_LOG2)))
        {
          size_t size;
          in_slab_index = 0;
          if (which_slab == MAX_SCANNED_MEMBER_SLAB)
            {
              UNPACK_ERROR (("too many fields"));
              goto error_cleanup_during_scan;
            }
          which_slab++;
          size = sizeof(ScannedMember) << (which_slab+FIRST_SCANNED_MEMBER_SLAB_SIZE_LOG2);
          /* TODO: consider using alloca() ! */
          if (allocator->tmp_alloc != NULL)
            scanned_member_slabs[which_slab] = TMPALLOC(allocator, size);
          else
            DO_ALLOC (scanned_member_slabs[which_slab], allocator, size, goto error_cleanup_during_scan);
        }
      scanned_member_slabs[which_slab][in_slab_index++] = tmp;

      if (field != NULL && field->label == PROTOBUF_C_LABEL_REPEATED)
        {
          size_t *n = STRUCT_MEMBER_PTR (size_t, rv, field->quantifier_offset);
          if (field->packed
           && wire_type == PROTOBUF_C_WIRE_TYPE_LENGTH_PREFIXED)
            {
              size_t count;
              if (!count_packed_elements (field->type,
                                          tmp.len - tmp.length_prefix_len,
                                          tmp.data + tmp.length_prefix_len,
                                          &count))
                {
                  UNPACK_ERROR (("counting packed elements"));
                  goto error_cleanup_during_scan;
                }
              *n += count;
            }
          else
           *n += 1;
        }

      at += tmp.len;
      rem -= tmp.len;
    }

  /* allocate space for repeated fields, also check that all required fields have been set */
  for (f = 0; f < desc->n_fields; f++)
  {
    const ProtobufCFieldDescriptor *field = desc->fields + f;
    if (field->label == PROTOBUF_C_LABEL_REPEATED)
    {
        size_t siz = sizeof_elt_in_repeated_array (field->type);
        size_t *n_ptr = STRUCT_MEMBER_PTR (size_t, rv, field->quantifier_offset);
        if (*n_ptr != 0)
          {
            unsigned n = *n_ptr;
            *n_ptr = 0;
            assert(rv->descriptor != NULL);
#define CLEAR_REMAINING_N_PTRS()                                          \
            for(f++;f < desc->n_fields; f++)                              \
              {                                                           \
                field = desc->fields + f;                                 \
                if (field->label == PROTOBUF_C_LABEL_REPEATED)            \
                  STRUCT_MEMBER (size_t, rv, field->quantifier_offset) = 0; \
              }
            DO_ALLOC (STRUCT_MEMBER (void *, rv, field->offset),
                      allocator, siz * n,
                      CLEAR_REMAINING_N_PTRS (); goto error_cleanup);
#undef CLEAR_REMAINING_N_PTRS
          }
    }
    else if (field->label == PROTOBUF_C_LABEL_REQUIRED)
    {
      if (field->default_value == NULL && 0 == (required_fields_bitmap[f / word_bits] & (1UL << (f % word_bits))))
      {
```

```
        UNPACK_ERROR (("message '%s': missing required field '%s'", desc->name, field->name));
        goto error_cleanup;
      }
    }
  }

  /* allocate space for unknown fields */
  if (n_unknown)
    {
      DO_ALLOC (rv->unknown_fields,
                allocator, n_unknown * sizeof (ProtobufCMessageUnknownField),
                goto error_cleanup);
    }

  /* do real parsing */
  for (i_slab = 0; i_slab <= which_slab; i_slab++)
    {
      unsigned max = (i_slab == which_slab) ? in_slab_index : (1U<<(i_slab+4));
      ScannedMember *slab = scanned_member_slabs[i_slab];
      unsigned j;
      for (j = 0; j < max; j++)
        {
          if (!parse_member (slab + j, rv, allocator))
            {
              UNPACK_ERROR (("error parsing member %s of %s",
                             slab->field ? slab->field->name : "*unknown-field*", desc->name));
              goto error_cleanup;
            }
        }
    }

  /* cleanup */
  if (allocator->tmp_alloc == NULL)
    {
      unsigned j;
      for (j = 1; j <= which_slab; j++)
        FREE (allocator, scanned_member_slabs[j]);
    }

  return rv;

error_cleanup:
  protobuf_c_message_free_unpacked (rv, allocator);
  if (allocator->tmp_alloc == NULL)
    {
      unsigned j;
      for (j = 1; j <= which_slab; j++)
        FREE (allocator, scanned_member_slabs[j]);
    }
  return NULL;

error_cleanup_during_scan:
  FREE (allocator, rv);
  if (allocator->tmp_alloc == NULL)
    {
      unsigned j;
      for (j = 1; j <= which_slab; j++)
        FREE (allocator, scanned_member_slabs[j]);
    }
  return NULL;
}

/* === free_unpacked === */
void
protobuf_c_message_free_unpacked  (ProtobufCMessage    *message,
                                   ProtobufCAllocator  *allocator)
{
  const ProtobufCMessageDescriptor *desc = message->descriptor;
  unsigned f;
  ASSERT_IS_MESSAGE (message);
  if (allocator == NULL)
    allocator = &protobuf_c_default_allocator;
  message->descriptor = NULL;
  for (f = 0; f < desc->n_fields; f++)
    {
      if (desc->fields[f].label == PROTOBUF_C_LABEL_REPEATED)
        {
          size_t n = STRUCT_MEMBER (size_t, message, desc->fields[f].quantifier_offset);
          void * arr = STRUCT_MEMBER (void *, message, desc->fields[f].offset);
          if (desc->fields[f].type == PROTOBUF_C_TYPE_STRING)
            {
              unsigned i;
              for (i = 0; i < n; i++)
                FREE (allocator, ((char**)arr)[i]);
            }
          else if (desc->fields[f].type == PROTOBUF_C_TYPE_BYTES)
            {
              unsigned i;
              for (i = 0; i < n; i++)
                FREE (allocator, ((ProtobufCBinaryData*)arr)[i].data);
            }
          else if (desc->fields[f].type == PROTOBUF_C_TYPE_MESSAGE)
            {
              unsigned i;
              for (i = 0; i < n; i++)
                protobuf_c_message_free_unpacked (((ProtobufCMessage**)arr)[i], allocator);
            }
          if (arr != NULL)
```

```
            FREE (allocator, arr);
        }
      else if (desc->fields[f].type == PROTOBUF_C_TYPE_STRING)
        {
          char *str = STRUCT_MEMBER (char *, message, desc->fields[f].offset);
          if (str && str != desc->fields[f].default_value)
            FREE (allocator, str);
        }
      else if (desc->fields[f].type == PROTOBUF_C_TYPE_BYTES)
        {
          void *data = STRUCT_MEMBER (ProtobufCBinaryData, message, desc->fields[f].offset).data;
          const ProtobufCBinaryData *default_bd;
          default_bd = desc->fields[f].default_value;
          if (data != NULL
           && (default_bd == NULL || default_bd->data != data))
            FREE (allocator, data);
        }
      else if (desc->fields[f].type == PROTOBUF_C_TYPE_MESSAGE)
        {
          ProtobufCMessage *sm;
          sm = STRUCT_MEMBER (ProtobufCMessage *, message,desc->fields[f].offset);
          if (sm && sm != desc->fields[f].default_value)
            protobuf_c_message_free_unpacked (sm, allocator);
        }
    }

  for (f = 0; f < message->n_unknown_fields; f++)
    FREE (allocator, message->unknown_fields[f].data);
  if (message->unknown_fields != NULL)
    FREE (allocator, message->unknown_fields);

  FREE (allocator, message);
}

void
protobuf_c_message_init (const ProtobufCMessageDescriptor *descriptor,
                         void                    *message)
{
  descriptor->message_init((ProtobufCMessage*) (message));
}

/* === services === */
typedef void (*GenericHandler)(void *service,
                               const ProtobufCMessage *input,
                               ProtobufCClosure  closure,
                               void             *closure_data);
void
protobuf_c_service_invoke_internal(ProtobufCService *service,
                                   unsigned          method_index,
                                   const ProtobufCMessage *input,
                                   ProtobufCClosure  closure,
                                   void             *closure_data)
{
  GenericHandler *handlers;
  GenericHandler handler;

  /* Verify that method_index is within range.
     If this fails, you are likely invoking a newly added
     method on an old service.  (Although other memory corruption
     bugs can cause this assertion too) */
  PROTOBUF_C_ASSERT (method_index < service->descriptor->n_methods);

  /* Get the array of virtual methods (which are enumerated by
     the generated code) */
  handlers = (GenericHandler *) (service + 1);

  /* get our method and invoke it */
  /* TODO: seems like handler==NULL is a situation that
     needs handling */
  handler = handlers[method_index];
  (*handler) (service, input, closure, closure_data);
}

void
protobuf_c_service_generated_init (ProtobufCService *service,
                                   const ProtobufCServiceDescriptor *descriptor,
                                   ProtobufCServiceDestroy destroy)
{
  ASSERT_IS_SERVICE_DESCRIPTOR(descriptor);
  service->descriptor = descriptor;
  service->destroy = destroy;
  service->invoke = protobuf_c_service_invoke_internal;
  memset (service + 1, 0, descriptor->n_methods * sizeof (GenericHandler));
}

void protobuf_c_service_destroy (ProtobufCService *service)
{
  service->destroy (service);
}

/* --- querying the descriptors --- */
const ProtobufCEnumValue *
protobuf_c_enum_descriptor_get_value_by_name
                        (const ProtobufCEnumDescriptor    *desc,
                         const char                       *name)
{
  unsigned start = 0, count = desc->n_value_names;
  while (count > 1)
```

```
    {
      unsigned mid = start + count / 2;
      int rv = strcmp (desc->values_by_name[mid].name, name);
      if (rv == 0)
        return desc->values + desc->values_by_name[mid].index;
      else if (rv < 0)
        {
          count = start + count - (mid + 1);
          start = mid + 1;
        }
      else
        count = mid - start;
    }
  if (count == 0)
    return NULL;
  if (strcmp (desc->values_by_name[start].name, name) == 0)
    return desc->values + desc->values_by_name[start].index;
  return NULL;
}
const ProtobufCEnumValue *
protobuf_c_enum_descriptor_get_value
                        (const ProtobufCEnumDescriptor    *desc,
                         int                               value)
{
  int rv = int_range_lookup (desc->n_value_ranges, desc->value_ranges, value);
  if (rv < 0)
    return NULL;
  return desc->values + rv;
}

const ProtobufCFieldDescriptor *
protobuf_c_message_descriptor_get_field_by_name
                        (const ProtobufCMessageDescriptor *desc,
                         const char                       *name)
{
  unsigned start = 0, count = desc->n_fields;
  const ProtobufCFieldDescriptor *field;
  while (count > 1)
    {
      unsigned mid = start + count / 2;
      int rv;
      field = desc->fields + desc->fields_sorted_by_name[mid];
      rv = strcmp (field->name, name);
      if (rv == 0)
        return field;
      else if (rv < 0)
        {
          count = start + count - (mid + 1);
          start = mid + 1;
        }
      else
        count = mid - start;
    }
  if (count == 0)
    return NULL;
  field = desc->fields + desc->fields_sorted_by_name[start];
  if (strcmp (field->name, name) == 0)
    return field;
  return NULL;
}

const ProtobufCFieldDescriptor *
protobuf_c_message_descriptor_get_field
                        (const ProtobufCMessageDescriptor *desc,
                         unsigned                          value)
{
  int rv = int_range_lookup (desc->n_field_ranges,
                             desc->field_ranges,
                             value);
  if (rv < 0)
    return NULL;
  return desc->fields + rv;
}

const ProtobufCMethodDescriptor *
protobuf_c_service_descriptor_get_method_by_name
                        (const ProtobufCServiceDescriptor *desc,
                         const char                       *name)
{
  unsigned start = 0, count = desc->n_methods;
  while (count > 1)
    {
      unsigned mid = start + count / 2;
      unsigned mid_index = desc->method_indices_by_name[mid];
      const char *mid_name = desc->methods[mid_index].name;
      int rv = strcmp (mid_name, name);
      if (rv == 0)
        return desc->methods + desc->method_indices_by_name[mid];
      if (rv < 0)
        {
          count = start + count - (mid + 1);
          start = mid + 1;
        }
      else
        {
          count = mid - start;
        }
    }
```

```
  if (count == 0)
    return NULL;
  if (strcmp (desc->methods[desc->method_indices_by_name[start]].name, name) == 0)
    return desc->methods + desc->method_indices_by_name[start];
  return NULL;
}
```

Notice for package(s)

ethtool

```
/*
 * ethtool.c: Linux ethernet device configuration tool.
 *
 * Copyright (C) 1998 David S. Miller (davem@dm.cobaltmicro.com)
 * Portions Copyright 2001 Sun Microsystems
 * Kernel 2.4 update Copyright 2001 Jeff Garzik <jgarzik@mandrakesoft.com>
 * Wake-on-LAN,natsemi,misc support by Tim Hockin <thockin@sun.com>
 * Portions Copyright 2002 Intel
 * Portions Copyright (C) Sun Microsystems 2008
 * do_test support by Eli Kupermann <eli.kupermann@intel.com>
 * ETHTOOL_PHYS_ID support by Chris Leech <christopher.leech@intel.com>
 * e1000 support by Scott Feldman <scott.feldman@intel.com>
 * e100 support by Wen Tao <wen-hwa.tao@intel.com>
 * ixgb support by Nicholas Nunley <Nicholas.d.nunley@intel.com>
 * amd8111e support by Reeja John <reeja.john@amd.com>
 * long arguments by Andi Kleen.
 * SMSC LAN911x support by Steve Glendinning <steve.glendinning@smsc.com>
 * Rx Network Flow Control configuration support <santwona.behera@sun.com>
 * Various features by Ben Hutchings <bhutchings@solarflare.com>;
 *      Copyright 2009, 2010 Solarflare Communications
 * MDI-X set support by Jesse Brandeburg <jesse.brandeburg@intel.com>
 *      Copyright 2012 Intel Corporation
 *
 * TODO:
 *   * show settings for all devices
 */

#include "internal.h"
#include <string.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stddef.h>
#include <errno.h>
#include <sys/utsname.h>
#include <limits.h>
#include <ctype.h>

#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#include <linux/sockios.h>

#ifndef MAX_ADDR_LEN
#define MAX_ADDR_LEN    32
#endif

#define ALL_ADVERTISED_MODES                    \
        (ADVERTISED_10baseT_Half |              \
         ADVERTISED_10baseT_Full |              \
         ADVERTISED_100baseT_Half |             \
         ADVERTISED_100baseT_Full |             \
         ADVERTISED_1000baseT_Half |            \
         ADVERTISED_1000baseT_Full |            \
         ADVERTISED_1000baseKX_Full|            \
         ADVERTISED_2500baseX_Full |            \
         ADVERTISED_10000baseT_Full |           \
         ADVERTISED_10000baseKX4_Full |         \
         ADVERTISED_10000baseKR_Full |          \
         ADVERTISED_10000baseR_FEC |            \
         ADVERTISED_20000baseMLD2_Full |        \
         ADVERTISED_20000baseKR2_Full |         \
         ADVERTISED_40000baseKR4_Full |         \
         ADVERTISED_40000baseCR4_Full |         \
         ADVERTISED_40000baseSR4_Full |         \
         ADVERTISED_40000baseLR4_Full |         \
         ADVERTISED_56000baseKR4_Full |         \
         ADVERTISED_56000baseCR4_Full |         \
         ADVERTISED_56000baseSR4_Full |         \
         ADVERTISED_56000baseLR4_Full)

#define ALL_ADVERTISED_FLAGS                    \
        (ADVERTISED_Autoneg |                   \
         ADVERTISED_TP |                        \
         ADVERTISED_AUI |                       \
         ADVERTISED_MII |                       \
         ADVERTISED_FIBRE |                     \
         ADVERTISED_BNC |                       \
         ADVERTISED_Pause |                     \
```

```
         ADVERTISED_Asym_Pause |                \
         ADVERTISED_Backplane |                 \
         ALL_ADVERTISED_MODES)

#ifndef HAVE_NETIF_MSG
enum {
        NETIF_MSG_DRV           = 0x0001,
        NETIF_MSG_PROBE         = 0x0002,
        NETIF_MSG_LINK          = 0x0004,
        NETIF_MSG_TIMER         = 0x0008,
        NETIF_MSG_IFDOWN        = 0x0010,
        NETIF_MSG_IFUP          = 0x0020,
        NETIF_MSG_RX_ERR        = 0x0040,
        NETIF_MSG_TX_ERR        = 0x0080,
        NETIF_MSG_TX_QUEUED     = 0x0100,
        NETIF_MSG_INTR          = 0x0200,
        NETIF_MSG_TX_DONE       = 0x0400,
        NETIF_MSG_RX_STATUS     = 0x0800,
        NETIF_MSG_PKTDATA       = 0x1000,
        NETIF_MSG_HW            = 0x2000,
        NETIF_MSG_WOL           = 0x4000,
};
#endif

#define KERNEL_VERSION(a,b,c) (((a) << 16) + ((b) << 8) + (c))

static void exit_bad_args(void) __attribute__((noreturn));

static void exit_bad_args(void)
{
        fprintf(stderr,
                "ethtool: bad command line argument(s)\n"
                "For more information run ethtool -h\n");
        exit(1);
}

typedef enum {
        CMDL_NONE,
        CMDL_BOOL,
        CMDL_S32,
        CMDL_U8,
        CMDL_U16,
        CMDL_U32,
        CMDL_U64,
        CMDL_BE16,
        CMDL_IP4,
        CMDL_STR,
        CMDL_FLAG,
        CMDL_MAC,
} cmdline_type_t;

struct cmdline_info {
        const char *name;
        cmdline_type_t type;
        /* Points to int (BOOL), s32, u16, u32 (U32/FLAG/IP4), u64,
         * char * (STR) or u8[6] (MAC).  For FLAG, the value accumulates
         * all flags to be set. */
        void *wanted_val;
        void *ioctl_val;
        /* For FLAG, the flag value to be set/cleared */
        u32 flag_val;
        /* For FLAG, points to u32 and accumulates all flags seen.
         * For anything else, points to int and is set if the option is
         * seen. */
        void *seen_val;
};

struct flag_info {
        const char *name;
        u32 value;
};

static const struct flag_info flags_msglvl[] = {
        { "drv",        NETIF_MSG_DRV },
        { "probe",      NETIF_MSG_PROBE },
        { "link",       NETIF_MSG_LINK },
        { "timer",      NETIF_MSG_TIMER },
        { "ifdown",     NETIF_MSG_IFDOWN },
        { "ifup",       NETIF_MSG_IFUP },
        { "rx_err",     NETIF_MSG_RX_ERR },
        { "tx_err",     NETIF_MSG_TX_ERR },
        { "tx_queued",  NETIF_MSG_TX_QUEUED },
        { "intr",       NETIF_MSG_INTR },
        { "tx_done",    NETIF_MSG_TX_DONE },
        { "rx_status",  NETIF_MSG_RX_STATUS },
        { "pktdata",    NETIF_MSG_PKTDATA },
        { "hw",         NETIF_MSG_HW },
        { "wol",        NETIF_MSG_WOL },
};

struct off_flag_def {
        const char *short_name;
        const char *long_name;
        const char *kernel_name;
        u32 get_cmd, set_cmd;
        u32 value;
        /* For features exposed through ETHTOOL_GFLAGS, the oldest
         * kernel version for which we can trust the result.  Where
```

```
         * the flag was added at the same time the kernel started
         * supporting the feature, this is 0 (to allow for backports).
         * Where the feature was supported before the flag was added,
         * it is the version that introduced the flag.
         */
        u32 min_kernel_ver;
};
static const struct off_flag_def off_flag_def[] = {
        { "rx",     "rx-checksumming",              "rx-checksum",
          ETHTOOL_GRXCSUM, ETHTOOL_SRXCSUM, ETH_FLAG_RXCSUM,    0 },
        { "tx",     "tx-checksumming",              "tx-checksum-*",
          ETHTOOL_GTXCSUM, ETHTOOL_STXCSUM, ETH_FLAG_TXCSUM,    0 },
        { "sg",     "scatter-gather",               "tx-scatter-gather*",
          ETHTOOL_GSG,     ETHTOOL_SSG,     ETH_FLAG_SG,        0 },
        { "tso",    "tcp-segmentation-offload",     "tx-tcp*-segmentation",
          ETHTOOL_GTSO,    ETHTOOL_STSO,    ETH_FLAG_TSO,       0 },
        { "ufo",    "udp-fragmentation-offload",    "tx-udp-fragmentation",
          ETHTOOL_GUFO,    ETHTOOL_SUFO,    ETH_FLAG_UFO,       0 },
        { "gso",    "generic-segmentation-offload", "tx-generic-segmentation",
          ETHTOOL_GGSO,    ETHTOOL_SGSO,    ETH_FLAG_GSO,       0 },
        { "gro",    "generic-receive-offload",      "rx-gro",
          ETHTOOL_GGRO,    ETHTOOL_SGRO,    ETH_FLAG_GRO,       0 },
        { "lro",    "large-receive-offload",        "rx-lro",
          0,               0,               ETH_FLAG_LRO,
          KERNEL_VERSION(2,6,24) },
        { "rxvlan", "rx-vlan-offload",              "rx-vlan-hw-parse",
          0,               0,               ETH_FLAG_RXVLAN,
          KERNEL_VERSION(2,6,37) },
        { "txvlan", "tx-vlan-offload",              "tx-vlan-hw-insert",
          0,               0,               ETH_FLAG_TXVLAN,
          KERNEL_VERSION(2,6,37) },
        { "ntuple", "ntuple-filters",               "rx-ntuple-filter",
          0,               0,               ETH_FLAG_NTUPLE,    0 },
        { "rxhash", "receive-hashing",              "rx-hashing",
          0,               0,               ETH_FLAG_RXHASH,    0 },
};

struct feature_def {
        char name[ETH_GSTRING_LEN];
        int off_flag_index; /* index in off_flag_def; negative if none match */
};

struct feature_defs {
        size_t n_features;
        /* Number of features each offload flag is associated with */
        unsigned int off_flag_matched[ARRAY_SIZE(off_flag_def)];
        /* Name and offload flag index for each feature */
        struct feature_def def[0];
};

#define FEATURE_BITS_TO_BLOCKS(n_bits)          DIV_ROUND_UP(n_bits, 32U)
#define FEATURE_WORD(blocks, index, field)      ((blocks)[(index) / 32U].field)
#define FEATURE_FIELD_FLAG(index)               (1U << (index) % 32U)
#define FEATURE_BIT_SET(blocks, index, field)                   \
        (FEATURE_WORD(blocks, index, field) |= FEATURE_FIELD_FLAG(index))
#define FEATURE_BIT_CLEAR(blocks, index, field)                 \
        (FEATURE_WORD(blocks, index, filed) &= ~FEATURE_FIELD_FLAG(index))
#define FEATURE_BIT_IS_SET(blocks, index, field)                \
        (FEATURE_WORD(blocks, index, field) & FEATURE_FIELD_FLAG(index))

static long long
get_int_range(char *str, int base, long long min, long long max)
{
        long long v;
        char *endp;

        if (!str)
                exit_bad_args();
        errno = 0;
        v = strtoll(str, &endp, base);
        if (errno || *endp || v < min || v > max)
                exit_bad_args();
        return v;
}

static unsigned long long
get_uint_range(char *str, int base, unsigned long long max)
{
        unsigned long long v;
        char *endp;

        if (!str)
                exit_bad_args();
        errno = 0;
        v = strtoull(str, &endp, base);
        if ( errno || *endp || v > max)
                exit_bad_args();
        return v;
}

static int get_int(char *str, int base)
{
        return get_int_range(str, base, INT_MIN, INT_MAX);
}

static u32 get_u32(char *str, int base)
{
        return get_uint_range(str, base, 0xffffffff);
```

```
}

static void get_mac_addr(char *src, unsigned char *dest)
{
        int count;
        int i;
        int buf[ETH_ALEN];

        count = sscanf(src, "%2x:%2x:%2x:%2x:%2x:%2x",
                &buf[0], &buf[1], &buf[2], &buf[3], &buf[4], &buf[5]);
        if (count != ETH_ALEN)
                exit_bad_args();

        for (i = 0; i < count; i++) {
                dest[i] = buf[i];
        }
}

static void parse_generic_cmdline(struct cmd_context *ctx,
                                  int *changed,
                                  struct cmdline_info *info,
                                  unsigned int n_info)
{
        int argc = ctx->argc;
        char **argp = ctx->argp;
        int i, idx;
        int found;

        for (i = 0; i < argc; i++) {
                found = 0;
                for (idx = 0; idx < n_info; idx++) {
                        if (!strcmp(info[idx].name, argp[i])) {
                                found = 1;
                                *changed = 1;
                                if (info[idx].type != CMDL_FLAG &&
                                    info[idx].seen_val)
                                        *(int *)info[idx].seen_val = 1;
                                i += 1;
                                if (i >= argc)
                                        exit_bad_args();
                                switch (info[idx].type) {
                                case CMDL_BOOL: {
                                        int *p = info[idx].wanted_val;
                                        if (!strcmp(argp[i], "on"))
                                                *p = 1;
                                        else if (!strcmp(argp[i], "off"))
                                                *p = 0;
                                        else
                                                exit_bad_args();
                                        break;
                                }
                                case CMDL_S32: {
                                        s32 *p = info[idx].wanted_val;
                                        *p = get_int_range(argp[i], 0,
                                                           -0x80000000LL,
                                                           0x7fffffff);
                                        break;
                                }
                                case CMDL_U8: {
                                        u8 *p = info[idx].wanted_val;
                                        *p = get_uint_range(argp[i], 0, 0xff);
                                        break;
                                }
                                case CMDL_U16: {
                                        u16 *p = info[idx].wanted_val;
                                        *p = get_uint_range(argp[i], 0, 0xffff);
                                        break;
                                }
                                case CMDL_U32: {
                                        u32 *p = info[idx].wanted_val;
                                        *p = get_uint_range(argp[i], 0,
                                                            0xffffffff);
                                        break;
                                }
                                case CMDL_U64: {
                                        u64 *p = info[idx].wanted_val;
                                        *p = get_uint_range(
                                                argp[i], 0,
                                                0xffffffffffffffffLL);
                                        break;
                                }
                                case CMDL_BE16: {
                                        u16 *p = info[idx].wanted_val;
                                        *p = cpu_to_be16(
                                                get_uint_range(argp[i], 0,
                                                               0xffff));
                                        break;
                                }
                                case CMDL_IP4: {
                                        u32 *p = info[idx].wanted_val;
                                        struct in_addr in;
                                        if (!inet_aton(argp[i], &in))
                                                exit_bad_args();
                                        *p = in.s_addr;
                                        break;
                                }
                                case CMDL_MAC:
                                        get_mac_addr(argp[i],
```

```
                                                info[idx].wanted_val);
                                        break;
                                case CMDL_FLAG: {
                                        u32 *p;
                                        p = info[idx].seen_val;
                                        *p |= info[idx].flag_val;
                                        if (!strcmp(argp[i], "on")) {
                                                p = info[idx].wanted_val;
                                                *p |= info[idx].flag_val;
                                        } else if (strcmp(argp[i], "off")) {
                                                exit_bad_args();
                                        }
                                        break;
                                }
                                case CMDL_STR: {
                                        char **s = info[idx].wanted_val;
                                        *s = strdup(argp[i]);
                                        break;
                                }
                                default:
                                        exit_bad_args();
                                }
                                break;
                        }
                }
                if( !found)
                        exit_bad_args();
        }
}

static void flag_to_cmdline_info(const char *name, u32 value,
                                 u32 *wanted, u32 *mask,
                                 struct cmdline_info *cli)
{
        memset(cli, 0, sizeof(*cli));
        cli->name = name;
        cli->type = CMDL_FLAG;
        cli->flag_val = value;
        cli->wanted_val = wanted;
        cli->seen_val = mask;
}

static void
print_flags(const struct flag_info *info, unsigned int n_info, u32 value)
{
        const char *sep = "";

        while (n_info) {
                if (value & info->value) {
                        printf("%s%s", sep, info->name);
                        sep = " ";
                        value &= ~info->value;
                }
                ++info;
                --n_info;
        }

        /* Print any unrecognised flags in hex */
        if (value)
                printf("%s%#x", sep, value);
}

static int rxflow_str_to_type(const char *str)
{
        int flow_type = 0;

        if (!strcmp(str, "tcp4"))
                flow_type = TCP_V4_FLOW;
        else if (!strcmp(str, "udp4"))
                flow_type = UDP_V4_FLOW;
        else if (!strcmp(str, "ah4") || !strcmp(str, "esp4"))
                flow_type = AH_ESP_V4_FLOW;
        else if (!strcmp(str, "sctp4"))
                flow_type = SCTP_V4_FLOW;
        else if (!strcmp(str, "tcp6"))
                flow_type = TCP_V6_FLOW;
        else if (!strcmp(str, "udp6"))
                flow_type = UDP_V6_FLOW;
        else if (!strcmp(str, "ah6") || !strcmp(str, "esp6"))
                flow_type = AH_ESP_V6_FLOW;
        else if (!strcmp(str, "sctp6"))
                flow_type = SCTP_V6_FLOW;
        else if (!strcmp(str, "ether"))
                flow_type = ETHER_FLOW;

        return flow_type;
}

static int do_version(struct cmd_context *ctx)
{
        fprintf(stdout,
                PACKAGE " version " VERSION
#ifndef ETHTOOL_ENABLE_PRETTY_DUMP
                " (pretty dumps disabled)"
#endif
                "\n");
        return 0;
}
```

```
static void dump_link_caps(const char *prefix, const char *an_prefix, u32 mask,
                           int link_mode_only);

static void dump_supported(struct ethtool_cmd *ep)
{
        u32 mask = ep->supported;

        fprintf(stdout, "       Supported ports: [ ");
        if (mask & SUPPORTED_TP)
                fprintf(stdout, "TP ");
        if (mask & SUPPORTED_AUI)
                fprintf(stdout, "AUI ");
        if (mask & SUPPORTED_BNC)
                fprintf(stdout, "BNC ");
        if (mask & SUPPORTED_MII)
                fprintf(stdout, "MII ");
        if (mask & SUPPORTED_FIBRE)
                fprintf(stdout, "FIBRE ");
        if (mask & SUPPORTED_Backplane)
                fprintf(stdout, "Backplane ");
        fprintf(stdout, "]\n");

        dump_link_caps("Supported", "Supports", mask, 0);
}

/* Print link capability flags (supported, advertised or lp_advertised).
 * Assumes that the corresponding SUPPORTED and ADVERTISED flags are equal.
 */
static void
dump_link_caps(const char *prefix, const char *an_prefix, u32 mask,
               int link_mode_only)
{
        static const struct {
                int same_line; /* print on same line as previous */
                u32 value;
                const char *name;
        } mode_defs[] = {
                { 0, ADVERTISED_10baseT_Half,       "10baseT/Half" },
                { 1, ADVERTISED_10baseT_Full,       "10baseT/Full" },
                { 0, ADVERTISED_100baseT_Half,      "100baseT/Half" },
                { 1, ADVERTISED_100baseT_Full,      "100baseT/Full" },
                { 0, ADVERTISED_1000baseT_Half,     "1000baseT/Half" },
                { 1, ADVERTISED_1000baseT_Full,     "1000baseT/Full" },
                { 0, ADVERTISED_1000baseKX_Full,    "1000baseKX/Full" },
                { 0, ADVERTISED_2500baseX_Full,     "2500baseX/Full" },
                { 0, ADVERTISED_10000baseT_Full,    "10000baseT/Full" },
                { 0, ADVERTISED_10000baseKX4_Full,  "10000baseKX4/Full" },
                { 0, ADVERTISED_10000baseKR_Full,   "10000baseKR/Full" },
                { 0, ADVERTISED_20000baseMLD2_Full, "20000baseMLD2/Full" },
                { 0, ADVERTISED_20000baseKR2_Full,  "20000baseKR2/Full" },
                { 0, ADVERTISED_40000baseKR4_Full,  "40000baseKR4/Full" },
                { 0, ADVERTISED_40000baseCR4_Full,  "40000baseCR4/Full" },
                { 0, ADVERTISED_40000baseSR4_Full,  "40000baseSR4/Full" },
                { 0, ADVERTISED_40000baseLR4_Full,  "40000baseLR4/Full" },
                { 0, ADVERTISED_56000baseKR4_Full,  "56000baseKR4/Full" },
                { 0, ADVERTISED_56000baseCR4_Full,  "56000baseCR4/Full" },
                { 0, ADVERTISED_56000baseSR4_Full,  "56000baseSR4/Full" },
                { 0, ADVERTISED_56000baseLR4_Full,  "56000baseLR4/Full" },
        };
        int indent;
        int did1, new_line_pend, i;

        /* Indent just like the separate functions used to */
        indent = strlen(prefix) + 14;
        if (indent < 24)
                indent = 24;

        fprintf(stdout, "       %s link modes:%*s", prefix,
                indent - (int)strlen(prefix) - 12, "");
        did1 = 0;
        new_line_pend = 0;
        for (i = 0; i < ARRAY_SIZE(mode_defs); i++) {
                if (did1 && !mode_defs[i].same_line)
                        new_line_pend = 1;
                if (mask & mode_defs[i].value) {
                        if (new_line_pend) {
                                fprintf(stdout, "\n");
                                fprintf(stdout, "       %*s", indent, "");
                                new_line_pend = 0;
                        }
                        did1++;
                        fprintf(stdout, "%s ", mode_defs[i].name);
                }
        }
        if (did1 == 0)
                 fprintf(stdout, "Not reported");
        fprintf(stdout, "\n");

        if (!link_mode_only) {
                fprintf(stdout, "       %s pause frame use: ", prefix);
                if (mask & ADVERTISED_Pause) {
                        fprintf(stdout, "Symmetric");
                        if (mask & ADVERTISED_Asym_Pause)
                                fprintf(stdout, " Receive-only");
                        fprintf(stdout, "\n");
                } else {
                        if (mask & ADVERTISED_Asym_Pause)
```

```
                                fprintf(stdout, "Transmit-only\n");
                        else
                                fprintf(stdout, "No\n");
                }

                fprintf(stdout, "       %s auto-negotiation: ", an_prefix);
                if (mask & ADVERTISED_Autoneg)
                        fprintf(stdout, "Yes\n");
                else
                        fprintf(stdout, "No\n");
        }
}

static int dump_ecmd(struct ethtool_cmd *ep)
{
        u32 speed;

        dump_supported(ep);
        dump_link_caps("Advertised", "Advertised", ep->advertising, 0);
        if (ep->lp_advertising)
                dump_link_caps("Link partner advertised",
                               "Link partner advertised", ep->lp_advertising,
                               0);

        fprintf(stdout, "       Speed: ");
        speed = ethtool_cmd_speed(ep);
        if (speed == 0 || speed == (u16)(-1) || speed == (u32)(-1))
                fprintf(stdout, "Unknown!\n");
        else
                fprintf(stdout, "%uMb/s\n", speed);

        fprintf(stdout, "       Duplex: ");
        switch (ep->duplex) {
        case DUPLEX_HALF:
                fprintf(stdout, "Half\n");
                break;
        case DUPLEX_FULL:
                fprintf(stdout, "Full\n");
                break;
        default:
                fprintf(stdout, "Unknown! (%i)\n", ep->duplex);
                break;
        };

        fprintf(stdout, "       Port: ");
        switch (ep->port) {
        case PORT_TP:
                fprintf(stdout, "Twisted Pair\n");
                break;
        case PORT_AUI:
                fprintf(stdout, "AUI\n");
                break;
        case PORT_BNC:
                fprintf(stdout, "BNC\n");
                break;
        case PORT_MII:
                fprintf(stdout, "MII\n");
                break;
        case PORT_FIBRE:
                fprintf(stdout, "FIBRE\n");
                break;
        case PORT_DA:
                fprintf(stdout, "Direct Attach Copper\n");
                break;
        case PORT_NONE:
                fprintf(stdout, "None\n");
                break;
        case PORT_OTHER:
                fprintf(stdout, "Other\n");
                break;
        default:
                fprintf(stdout, "Unknown! (%i)\n", ep->port);
                break;
        };

        fprintf(stdout, "       PHYAD: %d\n", ep->phy_address);
        fprintf(stdout, "       Transceiver: ");
        switch (ep->transceiver) {
        case XCVR_INTERNAL:
                fprintf(stdout, "internal\n");
                break;
        case XCVR_EXTERNAL:
                fprintf(stdout, "external\n");
                break;
        default:
                fprintf(stdout, "Unknown!\n");
                break;
        };

        fprintf(stdout, "       Auto-negotiation: %s\n",
                (ep->autoneg == AUTONEG_DISABLE) ?
                "off" : "on");

        if (ep->port == PORT_TP) {
                fprintf(stdout, "       MDI-X: ");
                if (ep->eth_tp_mdix_ctrl == ETH_TP_MDI) {
                        fprintf(stdout, "off (forced)\n");
                } else if (ep->eth_tp_mdix_ctrl == ETH_TP_MDI_X) {
```

```
                        fprintf(stdout, "on (forced)\n");
                } else {
                        switch (ep->eth_tp_mdix) {
                        case ETH_TP_MDI:
                                fprintf(stdout, "off");
                                break;
                        case ETH_TP_MDI_X:
                                fprintf(stdout, "on");
                                break;
                        default:
                                fprintf(stdout, "Unknown");
                                break;
                        }
                        if (ep->eth_tp_mdix_ctrl == ETH_TP_MDI_AUTO)
                                fprintf(stdout, " (auto)");
                        fprintf(stdout, "\n");
                }
        }

        return 0;
}

static int dump_drvinfo(struct ethtool_drvinfo *info)
{
        fprintf(stdout,
                "driver: %.*s\n"
                "version: %.*s\n"
                "firmware-version: %.*s\n"
                "expansion-rom-version: %.*s\n"
                "bus-info: %.*s\n"
                "supports-statistics: %s\n"
                "supports-test: %s\n"
                "supports-eeprom-access: %s\n"
                "supports-register-dump: %s\n"
                "supports-priv-flags: %s\n",
                (int)sizeof(info->driver), info->driver,
                (int)sizeof(info->version), info->version,
                (int)sizeof(info->fw_version), info->fw_version,
                (int)sizeof(info->erom_version), info->erom_version,
                (int)sizeof(info->bus_info), info->bus_info,
                info->n_stats ? "yes" : "no",
                info->testinfo_len ? "yes" : "no",
                info->eedump_len ? "yes" : "no",
                info->regdump_len ? "yes" : "no",
                info->n_priv_flags ? "yes" : "no");

        return 0;
}

static int parse_wolopts(char *optstr, u32 *data)
{
        *data = 0;
        while (*optstr) {
                switch (*optstr) {
                        case 'p':
                                *data |= WAKE_PHY;
                                break;
                        case 'u':
                                *data |= WAKE_UCAST;
                                break;
                        case 'm':
                                *data |= WAKE_MCAST;
                                break;
                        case 'b':
                                *data |= WAKE_BCAST;
                                break;
                        case 'a':
                                *data |= WAKE_ARP;
                                break;
                        case 'g':
                                *data |= WAKE_MAGIC;
                                break;
                        case 's':
                                *data |= WAKE_MAGICSECURE;
                                break;
                        case 'd':
                                *data = 0;
                                break;
                        default:
                                return -1;
                }
                optstr++;
        }
        return 0;
}

static char *unparse_wolopts(int wolopts)
{
        static char buf[16];
        char *p = buf;

        memset(buf, 0, sizeof(buf));

        if (wolopts) {
                if (wolopts & WAKE_PHY)
                        *p++ = 'p';
                if (wolopts & WAKE_UCAST)
                        *p++ = 'u';
```

```
                if (wolopts & WAKE_MCAST)
                        *p++ = 'm';
                if (wolopts & WAKE_BCAST)
                        *p++ = 'b';
                if (wolopts & WAKE_ARP)
                        *p++ = 'a';
                if (wolopts & WAKE_MAGIC)
                        *p++ = 'g';
                if (wolopts & WAKE_MAGICSECURE)
                        *p++ = 's';
        } else {
                *p = 'd';
        }

        return buf;
}

static int dump_wol(struct ethtool_wolinfo *wol)
{
        fprintf(stdout, "       Supports Wake-on: %s\n",
                unparse_wolopts(wol->supported));
        fprintf(stdout, "       Wake-on: %s\n",
                unparse_wolopts(wol->wolopts));
        if (wol->supported & WAKE_MAGICSECURE) {
                int i;
                int delim = 0;
                fprintf(stdout, "        SecureOn password: ");
                for (i = 0; i < SOPASS_MAX; i++) {
                        fprintf(stdout, "%s%02x", delim?":"":"", wol->sopass[i]);
                        delim=1;
                }
                fprintf(stdout, "\n");
        }

        return 0;
}

static int parse_rxfhashopts(char *optstr, u32 *data)
{
        *data = 0;
        while (*optstr) {
                switch (*optstr) {
                        case 'm':
                                *data |= RXH_L2DA;
                                break;
                        case 'v':
                                *data |= RXH_VLAN;
                                break;
                        case 't':
                                *data |= RXH_L3_PROTO;
                                break;
                        case 's':
                                *data |= RXH_IP_SRC;
                                break;
                        case 'd':
                                *data |= RXH_IP_DST;
                                break;
                        case 'f':
                                *data |= RXH_L4_B_0_1;
                                break;
                        case 'n':
                                *data |= RXH_L4_B_2_3;
                                break;
                        case 'r':
                                *data |= RXH_DISCARD;
                                break;
                        default:
                                return -1;
                }
                optstr++;
        }
        return 0;
}

static char *unparse_rxfhashopts(u64 opts)
{
        static char buf[300];

        memset(buf, 0, sizeof(buf));

        if (opts) {
                if (opts & RXH_L2DA) {
                        strcat(buf, "L2DA\n");
                }
                if (opts & RXH_VLAN) {
                        strcat(buf, "VLAN tag\n");
                }
                if (opts & RXH_L3_PROTO) {
                        strcat(buf, "L3 proto\n");
                }
                if (opts & RXH_IP_SRC) {
                        strcat(buf, "IP SA\n");
                }
                if (opts & RXH_IP_DST) {
                        strcat(buf, "IP DA\n");
                }
                if (opts & RXH_L4_B_0_1) {
                        strcat(buf, "L4 bytes 0 & 1 [TCP/UDP src port]\n");
```

```
                }
                if (opts & RXH_L4_B_2_3) {
                        strcat(buf, "L4 bytes 2 & 3 [TCP/UDP dst port]\n");
                }
        } else {
                sprintf(buf, "None");
        }

        return buf;
}

static int convert_string_to_hashkey(char *rss_hkey, u32 key_size,
                                     const char *rss_hkey_string)
{
        u32 i = 0;
        int hex_byte, len;

        do {
                if (i > (key_size - 1)) {
                        fprintf(stderr,
                                "Key is too long for device (%u > %u)\n",
                                i + 1, key_size);
                        goto err;
                }

                if (sscanf(rss_hkey_string, "%2x%n", &hex_byte, &len) < 1 ||
                    len != 2) {
                        fprintf(stderr, "Invalid RSS hash key format\n");
                        goto err;
                }

                rss_hkey[i++] = hex_byte;
                rss_hkey_string += 2;

                if (*rss_hkey_string == ':') {
                        rss_hkey_string++;
                } else if (*rss_hkey_string != '\0') {
                        fprintf(stderr, "Invalid RSS hash key format\n");
                        goto err;
                }

        } while (*rss_hkey_string);

        if (i != key_size) {
                fprintf(stderr, "Key is too short for device (%u < %u)\n",
                        i, key_size);
                goto err;
        }

        return 0;
err:
        return 2;
}

static int parse_hkey(char **rss_hkey, u32 key_size,
                      const char *rss_hkey_string)
{
        if (!key_size) {
                fprintf(stderr,
                        "Cannot set RX flow hash configuration:\n"
                        " Hash key setting not supported\n");
                return 1;
        }

        *rss_hkey = malloc(key_size);
        if (!(*rss_hkey)) {
                perror("Cannot allocate memory for RSS hash key");
                return 1;
        }

        if (convert_string_to_hashkey(*rss_hkey, key_size,
                                      rss_hkey_string)) {
                free(*rss_hkey);
                *rss_hkey = NULL;
                return 2;
        }
        return 0;
}

static const struct {
        const char *name;
        int (*func)(struct ethtool_drvinfo *info, struct ethtool_regs *regs);

} driver_list[] = {
#ifdef ETHTOOL_ENABLE_PRETTY_DUMP
        { "8139cp", realtek_dump_regs },
        { "8139too", realtek_dump_regs },
        { "r8169", realtek_dump_regs },
        { "de2104x", de2104x_dump_regs },
        { "e1000", e1000_dump_regs },
        { "e1000e", e1000_dump_regs },
        { "igb", igb_dump_regs },
        { "ixgb", ixgb_dump_regs },
        { "ixgbe", ixgbe_dump_regs },
        { "ixgbevf", ixgbevf_dump_regs },
        { "natsemi", natsemi_dump_regs },
        { "e100", e100_dump_regs },
        { "amd8111e", amd8111e_dump_regs },
```

```
        { "pcnet32", pcnet32_dump_regs },
        { "fec_8xx", fec_8xx_dump_regs },
        { "ibm_emac", ibm_emac_dump_regs },
        { "tg3", tg3_dump_regs },
        { "skge", skge_dump_regs },
        { "sky2", sky2_dump_regs },
        { "vioc", vioc_dump_regs },
        { "smsc911x", smsc911x_dump_regs },
        { "at76c50x-usb", at76c50x_usb_dump_regs },
        { "sfc", sfc_dump_regs },
        { "st_mac100", st_mac100_dump_regs },
        { "st_gmac", st_gmac_dump_regs },
        { "et131x", et131x_dump_regs },
        { "altera_tse", altera_tse_dump_regs },
#endif
};

void dump_hex(FILE *file, const u8 *data, int len, int offset)
{
        int i;

        fprintf(file, "Offset\t\tValues\n");
        fprintf(file, "------\t\t------");
        for (i = 0; i < len; i++) {
                if (i % 16 == 0)
                        fprintf(file, "\n0x%04x:\t\t", i + offset);
                fprintf(file, "%02x ", data[i]);
        }
        fprintf(file, "\n");
}

static int dump_regs(int gregs_dump_raw, int gregs_dump_hex,
                     const char *gregs_dump_file,
                     struct ethtool_drvinfo *info, struct ethtool_regs *regs)
{
        int i;

        if (gregs_dump_raw) {
                fwrite(regs->data, regs->len, 1, stdout);
                return 0;
        }

        if (gregs_dump_file) {
                FILE *f = fopen(gregs_dump_file, "r");
                struct stat st;

                if (!f || fstat(fileno(f), &st) < 0) {
                        fprintf(stderr, "Can't open '%s': %s\n",
                                gregs_dump_file, strerror(errno));
                        return -1;
                }

                regs = realloc(regs, sizeof(*regs) + st.st_size);
                regs->len = st.st_size;
                fread(regs->data, regs->len, 1, f);
                fclose(f);
        }

        if (!gregs_dump_hex)
                for (i = 0; i < ARRAY_SIZE(driver_list); i++)
                        if (!strncmp(driver_list[i].name, info->driver,
                                     ETHTOOL_BUSINFO_LEN)) {
                                if (driver_list[i].func(info, regs) == 0)
                                        return 0;
                                /* This version (or some other
                                 * variation in the dump format) is
                                 * not handled; fall back to hex
                                 */
                                break;
                        }

        dump_hex(stdout, regs->data, regs->len, 0);

        return 0;
}

static int dump_eeprom(int geeprom_dump_raw, struct ethtool_drvinfo *info,
                       struct ethtool_eeprom *ee)
{
        if (geeprom_dump_raw) {
                fwrite(ee->data, 1, ee->len, stdout);
                return 0;
        }
#ifdef ETHTOOL_ENABLE_PRETTY_DUMP
        if (!strncmp("natsemi", info->driver, ETHTOOL_BUSINFO_LEN)) {
                return natsemi_dump_eeprom(info, ee);
        } else if (!strncmp("tg3", info->driver, ETHTOOL_BUSINFO_LEN)) {
                return tg3_dump_eeprom(info, ee);
        }
#endif
        dump_hex(stdout, ee->data, ee->len, ee->offset);

        return 0;
}

static int dump_test(struct ethtool_test *test,
                     struct ethtool_gstrings *strings)
{
```

```
        int i, rc;

        rc = test->flags & ETH_TEST_FL_FAILED;
        fprintf(stdout, "The test result is %s\n", rc ? "FAIL" : "PASS");

        if (test->flags & ETH_TEST_FL_EXTERNAL_LB)
                fprintf(stdout, "External loopback test was %sexecuted\n",
                        (test->flags & ETH_TEST_FL_EXTERNAL_LB_DONE) ?
                        "" : "not ");

        if (strings->len)
                fprintf(stdout, "The test extra info:\n");

        for (i = 0; i < strings->len; i++) {
                fprintf(stdout, "%s\t %d\n",
                        (char *)(strings->data + i * ETH_GSTRING_LEN),
                        (u32) test->data[i]);
        }

        fprintf(stdout, "\n");
        return rc;
}

static int dump_pause(const struct ethtool_pauseparam *epause,
                      u32 advertising, u32 lp_advertising)
{
        fprintf(stdout,
                "Autonegotiate: %s\n"
                "RX:            %s\n"
                "TX:            %s\n",
                epause->autoneg ? "on" : "off",
                epause->rx_pause ? "on" : "off",
                epause->tx_pause ? "on" : "off");

        if (lp_advertising) {
                int an_rx = 0, an_tx = 0;

                /* Work out negotiated pause frame usage per
                 * IEEE 802.3-2005 table 28B-3.
                 */
                if (advertising & lp_advertising & ADVERTISED_Pause) {
                        an_tx = 1;
                        an_rx = 1;
                } else if (advertising & lp_advertising &
                           ADVERTISED_Asym_Pause) {
                        if (advertising & ADVERTISED_Pause)
                                an_rx = 1;
                        else if (lp_advertising & ADVERTISED_Pause)
                                an_tx = 1;
                }

                fprintf(stdout,
                        "RX negotiated: %s\n"
                        "TX negotiated: %s\n",
                        an_rx ? "on" : "off",
                        an_tx ? "on" : "off");
        }

        fprintf(stdout, "\n");
        return 0;
}

static int dump_ring(const struct ethtool_ringparam *ering)
{
        fprintf(stdout,
                "Pre-set maximums:\n"
                "RX:            %u\n"
                "RX Mini:       %u\n"
                "RX Jumbo:      %u\n"
                "TX:            %u\n",
                ering->rx_max_pending,
                ering->rx_mini_max_pending,
                ering->rx_jumbo_max_pending,
                ering->tx_max_pending);

        fprintf(stdout,
                "Current hardware settings:\n"
                "RX:            %u\n"
                "RX Mini:       %u\n"
                "RX Jumbo:      %u\n"
                "TX:            %u\n",
                ering->rx_pending,
                ering->rx_mini_pending,
                ering->rx_jumbo_pending,
                ering->tx_pending);

        fprintf(stdout, "\n");
        return 0;
}

static int dump_channels(const struct ethtool_channels *echannels)
{
        fprintf(stdout,
                "Pre-set maximums:\n"
                "RX:            %u\n"
                "TX:            %u\n"
                "Other:         %u\n"
                "Combined:      %u\n",
```

```
                echannels->max_rx, echannels->max_tx,
                echannels->max_other,
                echannels->max_combined);

        fprintf(stdout,
                "Current hardware settings:\n"
                "RX:             %u\n"
                "TX:             %u\n"
                "Other:          %u\n"
                "Combined:       %u\n",
                echannels->rx_count, echannels->tx_count,
                echannels->other_count,
                echannels->combined_count);

        fprintf(stdout, "\n");
        return 0;
}

static int dump_coalesce(const struct ethtool_coalesce *ecoal)
{
        fprintf(stdout, "Adaptive RX: %s  TX: %s\n",
                ecoal->use_adaptive_rx_coalesce ? "on" : "off",
                ecoal->use_adaptive_tx_coalesce ? "on" : "off");

        fprintf(stdout,
                "stats-block-usecs: %u\n"
                "sample-interval: %u\n"
                "pkt-rate-low: %u\n"
                "pkt-rate-high: %u\n"
                "\n"
                "rx-usecs: %u\n"
                "rx-frames: %u\n"
                "rx-usecs-irq: %u\n"
                "rx-frames-irq: %u\n"
                "\n"
                "tx-usecs: %u\n"
                "tx-frames: %u\n"
                "tx-usecs-irq: %u\n"
                "tx-frames-irq: %u\n"
                "\n"
                "rx-usecs-low: %u\n"
                "rx-frame-low: %u\n"
                "tx-usecs-low: %u\n"
                "tx-frame-low: %u\n"
                "\n"
                "rx-usecs-high: %u\n"
                "rx-frame-high: %u\n"
                "tx-usecs-high: %u\n"
                "tx-frame-high: %u\n"
                "\n",
                ecoal->stats_block_coalesce_usecs,
                ecoal->rate_sample_interval,
                ecoal->pkt_rate_low,
                ecoal->pkt_rate_high,

                ecoal->rx_coalesce_usecs,
                ecoal->rx_max_coalesced_frames,
                ecoal->rx_coalesce_usecs_irq,
                ecoal->rx_max_coalesced_frames_irq,

                ecoal->tx_coalesce_usecs,
                ecoal->tx_max_coalesced_frames,
                ecoal->tx_coalesce_usecs_irq,
                ecoal->tx_max_coalesced_frames_irq,

                ecoal->rx_coalesce_usecs_low,
                ecoal->rx_max_coalesced_frames_low,
                ecoal->tx_coalesce_usecs_low,
                ecoal->tx_max_coalesced_frames_low,

                ecoal->rx_coalesce_usecs_high,
                ecoal->rx_max_coalesced_frames_high,
                ecoal->tx_coalesce_usecs_high,
                ecoal->tx_max_coalesced_frames_high);

        return 0;
}

struct feature_state {
        u32 off_flags;
        struct ethtool_gfeatures features;
};

static void dump_one_feature(const char *indent, const char *name,
                             const struct feature_state *state,
                             const struct feature_state *ref_state,
                             u32 index)
{
        if (ref_state &&
            !(FEATURE_BIT_IS_SET(state->features.features, index, active) ^
              FEATURE_BIT_IS_SET(ref_state->features.features, index, active)))
                return;

        printf("%s%s: %s%s\n",
               indent, name,
               FEATURE_BIT_IS_SET(state->features.features, index, active) ?
               "on" : "off",
               (!FEATURE_BIT_IS_SET(state->features.features, index, available)
```

```
                || FEATURE_BIT_IS_SET(state->features.features, index,
                                      never_changed))
              ? " [fixed]"
              : (FEATURE_BIT_IS_SET(state->features.features, index, requested)
                 ^ FEATURE_BIT_IS_SET(state->features.features, index, active))
              ? (FEATURE_BIT_IS_SET(state->features.features, index, requested)
                 ? " [requested on]" : " [requested off]")
              : "");
}

static int linux_version_code(void)
{
        struct utsname utsname;
        unsigned version, patchlevel, sublevel = 0;

        if (uname(&utsname))
                return -1;
        if (sscanf(utsname.release, "%u.%u.%u", &version, &patchlevel, &sublevel) < 2)
                return -1;
        return KERNEL_VERSION(version, patchlevel, sublevel);
}

static void dump_features(const struct feature_defs *defs,
                          const struct feature_state *state,
                          const struct feature_state *ref_state)
{
        int kernel_ver = linux_version_code();
        u32 value;
        int indent;
        int i, j;

        for (i = 0; i < ARRAY_SIZE(off_flag_def); i++) {
                /* Don't show features whose state is unknown on this
                 * kernel version
                 */
                if (defs->off_flag_matched[i] == 0 &&
                    off_flag_def[i].get_cmd == 0 &&
                    kernel_ver < off_flag_def[i].min_kernel_ver)
                        continue;

                value = off_flag_def[i].value;

                /* If this offload flag matches exactly one generic
                 * feature then it's redundant to show the flag and
                 * feature states separately.  Otherwise, show the
                 * flag state first.
                 */
                if (defs->off_flag_matched[i] != 1 &&
                    (!ref_state ||
                     (state->off_flags ^ ref_state->off_flags) & value)) {
                        printf("%s: %s\n",
                               off_flag_def[i].long_name,
                               (state->off_flags & value) ? "on" : "off");
                        indent = 1;
                } else {
                        indent = 0;
                }

                /* Show matching features */
                for (j = 0; j < defs->n_features; j++) {
                        if (defs->def[j].off_flag_index != i)
                                continue;
                        if (defs->off_flag_matched[i] != 1)
                                /* Show all matching feature states */
                                dump_one_feature(indent ? "\t" : "",
                                                 defs->def[j].name,
                                                 state, ref_state, j);
                        else
                                /* Show full state with the old flag name */
                                dump_one_feature("", off_flag_def[i].long_name,
                                                 state, ref_state, j);
                }
        }

        /* Show all unmatched features that have non-null names */
        for (j = 0; j < defs->n_features; j++)
                if (defs->def[j].off_flag_index < 0 && defs->def[j].name[0])
                        dump_one_feature("", defs->def[j].name,
                                         state, ref_state, j);
}

static int dump_rxfhash(int fhash, u64 val)
{
        switch (fhash) {
        case TCP_V4_FLOW:
                fprintf(stdout, "TCP over IPV4 flows");
                break;
        case UDP_V4_FLOW:
                fprintf(stdout, "UDP over IPV4 flows");
                break;
        case SCTP_V4_FLOW:
                fprintf(stdout, "SCTP over IPV4 flows");
                break;
        case AH_ESP_V4_FLOW:
        case AH_V4_FLOW:
        case ESP_V4_FLOW:
                fprintf(stdout, "IPSEC AH/ESP over IPV4 flows");
                break;
```

```
        case TCP_V6_FLOW:
                fprintf(stdout, "TCP over IPV6 flows");
                break;
        case UDP_V6_FLOW:
                fprintf(stdout, "UDP over IPV6 flows");
                break;
        case SCTP_V6_FLOW:
                fprintf(stdout, "SCTP over IPV6 flows");
                break;
        case AH_ESP_V6_FLOW:
        case AH_V6_FLOW:
        case ESP_V6_FLOW:
                fprintf(stdout, "IPSEC AH/ESP over IPV6 flows");
                break;
        default:
                break;
        }

        if (val & RXH_DISCARD) {
                fprintf(stdout, " - All matching flows discarded on RX\n");
                return 0;
        }
        fprintf(stdout, " use these fields for computing Hash flow key:\n");

        fprintf(stdout, "%s\n", unparse_rxfhashopts(val));

        return 0;
}

static void dump_eeecmd(struct ethtool_eee *ep)
{

        fprintf(stdout, "       EEE status: ");
        if (!ep->supported) {
                fprintf(stdout, "not supported\n");
                return;
        } else if (!ep->eee_enabled) {
                fprintf(stdout, "disabled\n");
        } else {
                fprintf(stdout, "enabled - ");
                if (ep->eee_active)
                        fprintf(stdout, "active\n");
                else
                        fprintf(stdout, "inactive\n");
        }

        fprintf(stdout, "       Tx LPI:");
        if (ep->tx_lpi_enabled)
                fprintf(stdout, " %d (us)\n", ep->tx_lpi_timer);
        else
                fprintf(stdout, " disabled\n");

        dump_link_caps("Supported EEE", "", ep->supported, 1);
        dump_link_caps("Advertised EEE", "", ep->advertised, 1);
        dump_link_caps("Link partner advertised EEE", "", ep->lp_advertised, 1);
}

#define N_SOTS 7

static char *so_timestamping_labels[N_SOTS] = {
        "hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)",
        "software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)",
        "hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)",
        "software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)",
        "software-system-clock (SOF_TIMESTAMPING_SOFTWARE)",
        "hardware-legacy-clock (SOF_TIMESTAMPING_SYS_HARDWARE)",
        "hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)",
};

#define N_TX_TYPES (HWTSTAMP_TX_ONESTEP_SYNC + 1)

static char *tx_type_labels[N_TX_TYPES] = {
        "off                   (HWTSTAMP_TX_OFF)",
        "on                    (HWTSTAMP_TX_ON)",
        "one-step-sync         (HWTSTAMP_TX_ONESTEP_SYNC)",
};

#define N_RX_FILTERS (HWTSTAMP_FILTER_PTP_V2_DELAY_REQ + 1)

static char *rx_filter_labels[N_RX_FILTERS] = {
        "none                  (HWTSTAMP_FILTER_NONE)",
        "all                   (HWTSTAMP_FILTER_ALL)",
        "some                  (HWTSTAMP_FILTER_SOME)",
        "ptpv1-l4-event        (HWTSTAMP_FILTER_PTP_V1_L4_EVENT)",
        "ptpv1-l4-sync         (HWTSTAMP_FILTER_PTP_V1_L4_SYNC)",
        "ptpv1-l4-delay-req    (HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ)",
        "ptpv2-l4-event        (HWTSTAMP_FILTER_PTP_V2_L4_EVENT)",
        "ptpv2-l4-sync         (HWTSTAMP_FILTER_PTP_V2_L4_SYNC)",
        "ptpv2-l4-delay-req    (HWTSTAMP_FILTER_PTP_V2_L4_DELAY_REQ)",
        "ptpv2-l2-event        (HWTSTAMP_FILTER_PTP_V2_L2_EVENT)",
        "ptpv2-l2-sync         (HWTSTAMP_FILTER_PTP_V2_L2_SYNC)",
        "ptpv2-l2-delay-req    (HWTSTAMP_FILTER_PTP_V2_L2_DELAY_REQ)",
        "ptpv2-event           (HWTSTAMP_FILTER_PTP_V2_EVENT)",
        "ptpv2-sync            (HWTSTAMP_FILTER_PTP_V2_SYNC)",
        "ptpv2-delay-req       (HWTSTAMP_FILTER_PTP_V2_DELAY_REQ)",
};

static int dump_tsinfo(const struct ethtool_ts_info *info)
```

```
{
        int i;

        fprintf(stdout, "Capabilities:\n");

        for (i = 0; i < N_SOTS; i++) {
                if (info->so_timestamping & (1 << i))
                        fprintf(stdout, "\t%s\n", so_timestamping_labels[i]);
        }

        fprintf(stdout, "PTP Hardware Clock: ");

        if (info->phc_index < 0)
                fprintf(stdout, "none\n");
        else
                fprintf(stdout, "%d\n", info->phc_index);

        fprintf(stdout, "Hardware Transmit Timestamp Modes:");

        if (!info->tx_types)
                fprintf(stdout, " none\n");
        else
                fprintf(stdout, "\n");

        for (i = 0; i < N_TX_TYPES; i++) {
                if (info->tx_types & (1 << i))
                        fprintf(stdout, "\t%s\n", tx_type_labels[i]);
        }

        fprintf(stdout, "Hardware Receive Filter Modes:");

        if (!info->rx_filters)
                fprintf(stdout, " none\n");
        else
                fprintf(stdout, "\n");

        for (i = 0; i < N_RX_FILTERS; i++) {
                if (info->rx_filters & (1 << i))
                        fprintf(stdout, "\t%s\n", rx_filter_labels[i]);
        }

        return 0;
}

static struct ethtool_gstrings *
get_stringset(struct cmd_context *ctx, enum ethtool_stringset set_id,
              ptrdiff_t drvinfo_offset, int null_terminate)
{
        struct {
                struct ethtool_sset_info hdr;
                u32 buf[1];
        } sset_info;
        struct ethtool_drvinfo drvinfo;
        u32 len, i;
        struct ethtool_gstrings *strings;

        sset_info.hdr.cmd = ETHTOOL_GSSET_INFO;
        sset_info.hdr.reserved = 0;
        sset_info.hdr.sset_mask = 1ULL << set_id;
        if (send_ioctl(ctx, &sset_info) == 0) {
                len = sset_info.hdr.sset_mask ? sset_info.hdr.data[0] : 0;
        } else if (errno == EOPNOTSUPP && drvinfo_offset != 0) {
                /* Fallback for old kernel versions */
                drvinfo.cmd = ETHTOOL_GDRVINFO;
                if (send_ioctl(ctx, &drvinfo))
                        return NULL;
                len = *(u32 *)((char *)&drvinfo + drvinfo_offset);
        } else {
                return NULL;
        }

        strings = calloc(1, sizeof(*strings) + len * ETH_GSTRING_LEN);
        if (!strings)
                return NULL;

        strings->cmd = ETHTOOL_GSTRINGS;
        strings->string_set = set_id;
        strings->len = len;
        if (len != 0 && send_ioctl(ctx, strings)) {
                free(strings);
                return NULL;
        }

        if (null_terminate)
                for (i = 0; i < len; i++)
                        strings->data[(i + 1) * ETH_GSTRING_LEN - 1] = 0;

        return strings;
}

static struct feature_defs *get_feature_defs(struct cmd_context *ctx)
{
        struct ethtool_gstrings *names;
        struct feature_defs *defs;
        u32 n_features;
        int i, j;

        names = get_stringset(ctx, ETH_SS_FEATURES, 0, 1);
```

```
	if (names) {
		n_features = names->len;
	} else if (errno == EOPNOTSUPP || errno == EINVAL) {
		/* Kernel doesn't support named features; not an error */
		n_features = 0;
	} else if (errno == EPERM) {
		/* Kernel bug: ETHTOOL_GSSET_INFO was privileged.
		 * Work around it. */
		n_features = 0;
	} else {
		return NULL;
	}

	defs = malloc(sizeof(*defs) + sizeof(defs->def[0]) * n_features);
	if (!defs)
		return NULL;

	defs->n_features = n_features;
	memset(defs->off_flag_matched, 0, sizeof(defs->off_flag_matched));

	/* Copy out feature names and find those associated with legacy flags */
	for (i = 0; i < defs->n_features; i++) {
		memcpy(defs->def[i].name, names->data + i * ETH_GSTRING_LEN,
		       ETH_GSTRING_LEN);
		defs->def[i].off_flag_index = -1;

		for (j = 0;
		     j < ARRAY_SIZE(off_flag_def) &&
			     defs->def[i].off_flag_index < 0;
		     j++) {
			const char *pattern =
				off_flag_def[j].kernel_name;
			const char *name = defs->def[i].name;
			for (;;) {
				if (*pattern == '*') {
					/* There is only one wildcard; so
					 * switch to a suffix comparison */
					size_t pattern_len =
						strlen(pattern + 1);
					size_t name_len = strlen(name);
					if (name_len < pattern_len)
						break; /* name is too short */
					name += name_len - pattern_len;
					++pattern;
				} else if (*pattern != *name) {
					break; /* mismatch */
				} else if (*pattern == 0) {
					defs->def[i].off_flag_index = j;
					defs->off_flag_matched[j]++;
					break;
				} else {
					++name;
					++pattern;
				}
			}
		}
	}

	free(names);
	return defs;
}

static int do_gdrv(struct cmd_context *ctx)
{
	int err;
	struct ethtool_drvinfo drvinfo;

	if (ctx->argc != 0)
		exit_bad_args();

	drvinfo.cmd = ETHTOOL_GDRVINFO;
	err = send_ioctl(ctx, &drvinfo);
	if (err < 0) {
		perror("Cannot get driver information");
		return 71;
	}
	return dump_drvinfo(&drvinfo);
}

static int do_gpause(struct cmd_context *ctx)
{
	struct ethtool_pauseparam epause;
	struct ethtool_cmd ecmd;
	int err;

	if (ctx->argc != 0)
		exit_bad_args();

	fprintf(stdout, "Pause parameters for %s:\n", ctx->devname);

	epause.cmd = ETHTOOL_GPAUSEPARAM;
	err = send_ioctl(ctx, &epause);
	if (err) {
		perror("Cannot get device pause settings");
		return 76;
	}

	if (epause.autoneg) {
```

```
                ecmd.cmd = ETHTOOL_GSET;
                err = send_ioctl(ctx, &ecmd);
                if (err) {
                        perror("Cannot get device settings");
                        return 1;
                }
                dump_pause(&epause, ecmd.advertising, ecmd.lp_advertising);
        } else {
                dump_pause(&epause, 0, 0);
        }

        return 0;
}

static void do_generic_set1(struct cmdline_info *info, int *changed_out)
{
        int wanted, *v1, *v2;

        v1 = info->wanted_val;
        wanted = *v1;

        if (wanted < 0)
                return;

        v2 = info->ioctl_val;
        if (wanted == *v2) {
                fprintf(stderr, "%s unmodified, ignoring\n", info->name);
        } else {
                *v2 = wanted;
                *changed_out = 1;
        }
}

static void do_generic_set(struct cmdline_info *info,
                           unsigned int n_info,
                           int *changed_out)
{
        unsigned int i;

        for (i = 0; i < n_info; i++)
                do_generic_set1(&info[i], changed_out);
}

static int do_spause(struct cmd_context *ctx)
{
        struct ethtool_pauseparam epause;
        int gpause_changed = 0;
        int pause_autoneg_wanted = -1;
        int pause_rx_wanted = -1;
        int pause_tx_wanted = -1;
        struct cmdline_info cmdline_pause[] = {
                { "autoneg", CMDL_BOOL, &pause_autoneg_wanted,
                  &epause.autoneg },
                { "rx", CMDL_BOOL, &pause_rx_wanted, &epause.rx_pause },
                { "tx", CMDL_BOOL, &pause_tx_wanted, &epause.tx_pause },
        };
        int err, changed = 0;

        parse_generic_cmdline(ctx, &gpause_changed,
                              cmdline_pause, ARRAY_SIZE(cmdline_pause));

        epause.cmd = ETHTOOL_GPAUSEPARAM;
        err = send_ioctl(ctx, &epause);
        if (err) {
                perror("Cannot get device pause settings");
                return 77;
        }

        do_generic_set(cmdline_pause, ARRAY_SIZE(cmdline_pause), &changed);

        if (!changed) {
                fprintf(stderr, "no pause parameters changed, aborting\n");
                return 78;
        }

        epause.cmd = ETHTOOL_SPAUSEPARAM;
        err = send_ioctl(ctx, &epause);
        if (err) {
                perror("Cannot set device pause parameters");
                return 79;
        }

        return 0;
}

static int do_sring(struct cmd_context *ctx)
{
        struct ethtool_ringparam ering;
        int gring_changed = 0;
        s32 ring_rx_wanted = -1;
        s32 ring_rx_mini_wanted = -1;
        s32 ring_rx_jumbo_wanted = -1;
        s32 ring_tx_wanted = -1;
        struct cmdline_info cmdline_ring[] = {
                { "rx", CMDL_S32, &ring_rx_wanted, &ering.rx_pending },
                { "rx-mini", CMDL_S32, &ring_rx_mini_wanted,
                  &ering.rx_mini_pending },
                { "rx-jumbo", CMDL_S32, &ring_rx_jumbo_wanted,
```

```
                  &ering.rx_jumbo_pending },
                { "tx", CMDL_S32, &ring_tx_wanted, &ering.tx_pending },
        };
        int err, changed = 0;

        parse_generic_cmdline(ctx, &gring_changed,
                              cmdline_ring, ARRAY_SIZE(cmdline_ring));

        ering.cmd = ETHTOOL_GRINGPARAM;
        err = send_ioctl(ctx, &ering);
        if (err) {
                perror("Cannot get device ring settings");
                return 76;
        }

        do_generic_set(cmdline_ring, ARRAY_SIZE(cmdline_ring), &changed);

        if (!changed) {
                fprintf(stderr, "no ring parameters changed, aborting\n");
                return 80;
        }

        ering.cmd = ETHTOOL_SRINGPARAM;
        err = send_ioctl(ctx, &ering);
        if (err) {
                perror("Cannot set device ring parameters");
                return 81;
        }

        return 0;
}

static int do_gring(struct cmd_context *ctx)
{
        struct ethtool_ringparam ering;
        int err;

        if (ctx->argc != 0)
                exit_bad_args();

        fprintf(stdout, "Ring parameters for %s:\n", ctx->devname);

        ering.cmd = ETHTOOL_GRINGPARAM;
        err = send_ioctl(ctx, &ering);
        if (err == 0) {
                err = dump_ring(&ering);
                if (err)
                        return err;
        } else {
                perror("Cannot get device ring settings");
                return 76;
        }

        return 0;
}

static int do_schannels(struct cmd_context *ctx)
{
        struct ethtool_channels echannels;
        int gchannels_changed;
        s32 channels_rx_wanted = -1;
        s32 channels_tx_wanted = -1;
        s32 channels_other_wanted = -1;
        s32 channels_combined_wanted = -1;
        struct cmdline_info cmdline_channels[] = {
                { "rx", CMDL_S32, &channels_rx_wanted, &echannels.rx_count },
                { "tx", CMDL_S32, &channels_tx_wanted, &echannels.tx_count },
                { "other", CMDL_S32, &channels_other_wanted,
                  &echannels.other_count },
                { "combined", CMDL_S32, &channels_combined_wanted,
                  &echannels.combined_count },
        };
        int err, changed = 0;

        parse_generic_cmdline(ctx, &gchannels_changed,
                              cmdline_channels, ARRAY_SIZE(cmdline_channels));

        echannels.cmd = ETHTOOL_GCHANNELS;
        err = send_ioctl(ctx, &echannels);
        if (err) {
                perror("Cannot get device channel parameters");
                return 1;
        }

        do_generic_set(cmdline_channels, ARRAY_SIZE(cmdline_channels),
                       &changed);

        if (!changed) {
                fprintf(stderr, "no channel parameters changed, aborting\n");
                fprintf(stderr, "current values: tx %u rx %u other %u"
                        " combined %u\n", echannels.rx_count,
                        echannels.tx_count, echannels.other_count,
                        echannels.combined_count);
                return 1;
        }

        echannels.cmd = ETHTOOL_SCHANNELS;
        err = send_ioctl(ctx, &echannels);
```

```
        if (err) {
                perror("Cannot set device channel parameters");
                return 1;
        }

        return 0;
}

static int do_gchannels(struct cmd_context *ctx)
{
        struct ethtool_channels echannels;
        int err;

        if (ctx->argc != 0)
                exit_bad_args();

        fprintf(stdout, "Channel parameters for %s:\n", ctx->devname);

        echannels.cmd = ETHTOOL_GCHANNELS;
        err = send_ioctl(ctx, &echannels);
        if (err == 0) {
                err = dump_channels(&echannels);
                if (err)
                        return err;
        } else {
                perror("Cannot get device channel parameters\n");
                return 1;
        }
        return 0;

}

static int do_gcoalesce(struct cmd_context *ctx)
{
        struct ethtool_coalesce ecoal;
        int err;

        if (ctx->argc != 0)
                exit_bad_args();

        fprintf(stdout, "Coalesce parameters for %s:\n", ctx->devname);

        ecoal.cmd = ETHTOOL_GCOALESCE;
        err = send_ioctl(ctx, &ecoal);
        if (err == 0) {
                err = dump_coalesce(&ecoal);
                if (err)
                        return err;
        } else {
                perror("Cannot get device coalesce settings");
                return 82;
        }

        return 0;
}

static int do_scoalesce(struct cmd_context *ctx)
{
        struct ethtool_coalesce ecoal;
        int gcoalesce_changed = 0;
        s32 coal_stats_wanted = -1;
        int coal_adaptive_rx_wanted = -1;
        int coal_adaptive_tx_wanted = -1;
        s32 coal_sample_rate_wanted = -1;
        s32 coal_pkt_rate_low_wanted = -1;
        s32 coal_pkt_rate_high_wanted = -1;
        s32 coal_rx_usec_wanted = -1;
        s32 coal_rx_frames_wanted = -1;
        s32 coal_rx_usec_irq_wanted = -1;
        s32 coal_rx_frames_irq_wanted = -1;
        s32 coal_tx_usec_wanted = -1;
        s32 coal_tx_frames_wanted = -1;
        s32 coal_tx_usec_irq_wanted = -1;
        s32 coal_tx_frames_irq_wanted = -1;
        s32 coal_rx_usec_low_wanted = -1;
        s32 coal_rx_frames_low_wanted = -1;
        s32 coal_tx_usec_low_wanted = -1;
        s32 coal_tx_frames_low_wanted = -1;
        s32 coal_rx_usec_high_wanted = -1;
        s32 coal_rx_frames_high_wanted = -1;
        s32 coal_tx_usec_high_wanted = -1;
        s32 coal_tx_frames_high_wanted = -1;
        struct cmdline_info cmdline_coalesce[] = {
                { "adaptive-rx", CMDL_BOOL, &coal_adaptive_rx_wanted,
                  &ecoal.use_adaptive_rx_coalesce },
                { "adaptive-tx", CMDL_BOOL, &coal_adaptive_tx_wanted,
                  &ecoal.use_adaptive_tx_coalesce },
                { "sample-interval", CMDL_S32, &coal_sample_rate_wanted,
                  &ecoal.rate_sample_interval },
                { "stats-block-usecs", CMDL_S32, &coal_stats_wanted,
                  &ecoal.stats_block_coalesce_usecs },
                { "pkt-rate-low", CMDL_S32, &coal_pkt_rate_low_wanted,
                  &ecoal.pkt_rate_low },
                { "pkt-rate-high", CMDL_S32, &coal_pkt_rate_high_wanted,
                  &ecoal.pkt_rate_high },
                { "rx-usecs", CMDL_S32, &coal_rx_usec_wanted,
                  &ecoal.rx_coalesce_usecs },
                { "rx-frames", CMDL_S32, &coal_rx_frames_wanted,
```

```
                  &ecoal.rx_max_coalesced_frames },
                { "rx-usecs-irq", CMDL_S32, &coal_rx_usec_irq_wanted,
                  &ecoal.rx_coalesce_usecs_irq },
                { "rx-frames-irq", CMDL_S32, &coal_rx_frames_irq_wanted,
                  &ecoal.rx_max_coalesced_frames_irq },
                { "tx-usecs", CMDL_S32, &coal_tx_usec_wanted,
                  &ecoal.tx_coalesce_usecs },
                { "tx-frames", CMDL_S32, &coal_tx_frames_wanted,
                  &ecoal.tx_max_coalesced_frames },
                { "tx-usecs-irq", CMDL_S32, &coal_tx_usec_irq_wanted,
                  &ecoal.tx_coalesce_usecs_irq },
                { "tx-frames-irq", CMDL_S32, &coal_tx_frames_irq_wanted,
                  &ecoal.tx_max_coalesced_frames_irq },
                { "rx-usecs-low", CMDL_S32, &coal_rx_usec_low_wanted,
                  &ecoal.rx_coalesce_usecs_low },
                { "rx-frames-low", CMDL_S32, &coal_rx_frames_low_wanted,
                  &ecoal.rx_max_coalesced_frames_low },
                { "tx-usecs-low", CMDL_S32, &coal_tx_usec_low_wanted,
                  &ecoal.tx_coalesce_usecs_low },
                { "tx-frames-low", CMDL_S32, &coal_tx_frames_low_wanted,
                  &ecoal.tx_max_coalesced_frames_low },
                { "rx-usecs-high", CMDL_S32, &coal_rx_usec_high_wanted,
                  &ecoal.rx_coalesce_usecs_high },
                { "rx-frames-high", CMDL_S32, &coal_rx_frames_high_wanted,
                  &ecoal.rx_max_coalesced_frames_high },
                { "tx-usecs-high", CMDL_S32, &coal_tx_usec_high_wanted,
                  &ecoal.tx_coalesce_usecs_high },
                { "tx-frames-high", CMDL_S32, &coal_tx_frames_high_wanted,
                  &ecoal.tx_max_coalesced_frames_high },
        };
        int err, changed = 0;

        parse_generic_cmdline(ctx, &gcoalesce_changed,
                              cmdline_coalesce, ARRAY_SIZE(cmdline_coalesce));

        ecoal.cmd = ETHTOOL_GCOALESCE;
        err = send_ioctl(ctx, &ecoal);
        if (err) {
                perror("Cannot get device coalesce settings");
                return 76;
        }

        do_generic_set(cmdline_coalesce, ARRAY_SIZE(cmdline_coalesce),
                       &changed);

        if (!changed) {
                fprintf(stderr, "no coalesce parameters changed, aborting\n");
                return 80;
        }

        ecoal.cmd = ETHTOOL_SCOALESCE;
        err = send_ioctl(ctx, &ecoal);
        if (err) {
                perror("Cannot set device coalesce parameters");
                return 81;
        }

        return 0;
}

static struct feature_state *
get_features(struct cmd_context *ctx, const struct feature_defs *defs)
{
        struct feature_state *state;
        struct ethtool_value eval;
        int err, allfail = 1;
        u32 value;
        int i;

        state = malloc(sizeof(*state) +
                       FEATURE_BITS_TO_BLOCKS(defs->n_features) *
                       sizeof(state->features.features[0]));
        if (!state)
                return NULL;

        state->off_flags = 0;

        for (i = 0; i < ARRAY_SIZE(off_flag_def); i++) {
                value = off_flag_def[i].value;
                if (!off_flag_def[i].get_cmd)
                        continue;
                eval.cmd = off_flag_def[i].get_cmd;
                err = send_ioctl(ctx, &eval);
                if (err) {
                        fprintf(stderr,
                                "Cannot get device %s settings: %m\n",
                                off_flag_def[i].long_name);
                } else {
                        if (eval.data)
                                state->off_flags |= value;
                        allfail = 0;
                }
        }

        eval.cmd = ETHTOOL_GFLAGS;
        err = send_ioctl(ctx, &eval);
        if (err) {
                perror("Cannot get device flags");
```

```
        } else {
                state->off_flags |= eval.data & ETH_FLAG_EXT_MASK;
                allfail = 0;
        }

        if (defs->n_features) {
                state->features.cmd = ETHTOOL_GFEATURES;
                state->features.size = FEATURE_BITS_TO_BLOCKS(defs->n_features);
                err = send_ioctl(ctx, &state->features);
                if (err)
                        perror("Cannot get device generic features");
                else
                        allfail = 0;
        }

        if (allfail) {
                free(state);
                return NULL;
        }

        return state;
}

static int do_gfeatures(struct cmd_context *ctx)
{
        struct feature_defs *defs;
        struct feature_state *features;

        if (ctx->argc != 0)
                exit_bad_args();

        defs = get_feature_defs(ctx);
        if (!defs) {
                perror("Cannot get device feature names");
                return 1;
        }

        fprintf(stdout, "Features for %s:\n", ctx->devname);

        features = get_features(ctx, defs);
        if (!features) {
                fprintf(stdout, "no feature info available\n");
                return 1;
        }

        dump_features(defs, features, NULL);
        return 0;
}

static int do_sfeatures(struct cmd_context *ctx)
{
        struct feature_defs *defs;
        int any_changed = 0, any_mismatch = 0;
        u32 off_flags_wanted = 0;
        u32 off_flags_mask = 0;
        struct ethtool_sfeatures *efeatures;
        struct cmdline_info *cmdline_features;
        struct feature_state *old_state, *new_state;
        struct ethtool_value eval;
        int err;
        int i, j;

        defs = get_feature_defs(ctx);
        if (!defs) {
                perror("Cannot get device feature names");
                return 1;
        }
        if (defs->n_features) {
                efeatures = malloc(sizeof(*efeatures) +
                                   FEATURE_BITS_TO_BLOCKS(defs->n_features) *
                                   sizeof(efeatures->features[0]));
                if (!efeatures) {
                        perror("Cannot parse arguments");
                        return 1;
                }
                efeatures->cmd = ETHTOOL_SFEATURES;
                efeatures->size = FEATURE_BITS_TO_BLOCKS(defs->n_features);
                memset(efeatures->features, 0,
                       FEATURE_BITS_TO_BLOCKS(defs->n_features) *
                       sizeof(efeatures->features[0]));
        } else {
                efeatures = NULL;
        }

        /* Generate cmdline_info for legacy flags and kernel-named
         * features, and parse our arguments.
         */
        cmdline_features = calloc(ARRAY_SIZE(off_flag_def) + defs->n_features,
                                  sizeof(cmdline_features[0]));
        if (!cmdline_features) {
                perror("Cannot parse arguments");
                return 1;
        }
        for (i = 0; i < ARRAY_SIZE(off_flag_def); i++)
                flag_to_cmdline_info(off_flag_def[i].short_name,
                                     off_flag_def[i].value,
                                     &off_flags_wanted, &off_flags_mask,
                                     &cmdline_features[i]);
```

```
for (i = 0; i < defs->n_features; i++)
        flag_to_cmdline_info(
                defs->def[i].name, FEATURE_FIELD_FLAG(i),
                &FEATURE_WORD(efeatures->features, i, requested),
                &FEATURE_WORD(efeatures->features, i, valid),
                &cmdline_features[ARRAY_SIZE(off_flag_def) + i]);
parse_generic_cmdline(ctx, &any_changed, cmdline_features,
                      ARRAY_SIZE(off_flag_def) + defs->n_features);
free(cmdline_features);

if (!any_changed) {
        fprintf(stdout, "no features changed\n");
        return 0;
}

old_state = get_features(ctx, defs);
if (!old_state)
        return 1;

if (efeatures) {
        /* For each offload that the user specified, update any
         * related features that the user did not specify and that
         * are not fixed.  Warn if all related features are fixed.
         */
        for (i = 0; i < ARRAY_SIZE(off_flag_def); i++) {
                int fixed = 1;

                if (!(off_flags_mask & off_flag_def[i].value))
                        continue;

                for (j = 0; j < defs->n_features; j++) {
                        if (defs->def[j].off_flag_index != i ||
                            !FEATURE_BIT_IS_SET(
                                    old_state->features.features,
                                    j, available) ||
                            FEATURE_BIT_IS_SET(
                                    old_state->features.features,
                                    j, never_changed))
                                continue;

                        fixed = 0;
                        if (!FEATURE_BIT_IS_SET(efeatures->features,
                                                j, valid)) {
                                FEATURE_BIT_SET(efeatures->features,
                                                j, valid);
                                if (off_flags_wanted &
                                    off_flag_def[i].value)
                                        FEATURE_BIT_SET(
                                                efeatures->features,
                                                j, requested);
                        }
                }

                if (fixed)
                        fprintf(stderr, "Cannot change %s\n",
                                off_flag_def[i].long_name);
        }

        err = send_ioctl(ctx, efeatures);
        if (err < 0) {
                perror("Cannot set device feature settings");
                return 1;
        }
} else {
        for (i = 0; i < ARRAY_SIZE(off_flag_def); i++) {
                if (!off_flag_def[i].set_cmd)
                        continue;
                if (off_flags_mask & off_flag_def[i].value) {
                        eval.cmd = off_flag_def[i].set_cmd;
                        eval.data = !!(off_flags_wanted &
                                       off_flag_def[i].value);
                        err = send_ioctl(ctx, &eval);
                        if (err) {
                                fprintf(stderr,
                                        "Cannot set device %s settings: %m\n",
                                        off_flag_def[i].long_name);
                                return 1;
                        }
                }
        }

        if (off_flags_mask & ETH_FLAG_EXT_MASK) {
                eval.cmd = ETHTOOL_SFLAGS;
                eval.data = (old_state->off_flags & ~off_flags_mask &
                             ETH_FLAG_EXT_MASK);
                eval.data |= off_flags_wanted & ETH_FLAG_EXT_MASK;

                err = send_ioctl(ctx, &eval);
                if (err) {
                        perror("Cannot set device flag settings");
                        return 92;
                }
        }
}

/* Compare new state with requested state */
new_state = get_features(ctx, defs);
if (!new_state)
```

```
                return 1;
        any_changed = new_state->off_flags != old_state->off_flags;
        any_mismatch = (new_state->off_flags !=
                        ((old_state->off_flags & ~off_flags_mask) |
                         off_flags_wanted));
        for (i = 0; i < FEATURE_BITS_TO_BLOCKS(defs->n_features); i++) {
                if (new_state->features.features[i].active !=
                    old_state->features.features[i].active)
                        any_changed = 1;
                if (new_state->features.features[i].active !=
                    ((old_state->features.features[i].active &
                      ~efeatures->features[i].valid) |
                     efeatures->features[i].requested))
                        any_mismatch = 1;
        }
        if (any_mismatch) {
                if (!any_changed) {
                        fprintf(stderr,
                                "Could not change any device features\n");
                        return 1;
                }
                printf("Actual changes:\n");
                dump_features(defs, new_state, old_state);
        }

        return 0;
}

static int do_gset(struct cmd_context *ctx)
{
        int err;
        struct ethtool_cmd ecmd;
        struct ethtool_wolinfo wolinfo;
        struct ethtool_value edata;
        int allfail = 1;

        if (ctx->argc != 0)
                exit_bad_args();

        fprintf(stdout, "Settings for %s:\n", ctx->devname);

        ecmd.cmd = ETHTOOL_GSET;
        err = send_ioctl(ctx, &ecmd);
        if (err == 0) {
                err = dump_ecmd(&ecmd);
                if (err)
                        return err;
                allfail = 0;
        } else if (errno != EOPNOTSUPP) {
                perror("Cannot get device settings");
        }

        wolinfo.cmd = ETHTOOL_GWOL;
        err = send_ioctl(ctx, &wolinfo);
        if (err == 0) {
                err = dump_wol(&wolinfo);
                if (err)
                        return err;
                allfail = 0;
        } else if (errno != EOPNOTSUPP) {
                perror("Cannot get wake-on-lan settings");
        }

        edata.cmd = ETHTOOL_GMSGLVL;
        err = send_ioctl(ctx, &edata);
        if (err == 0) {
                fprintf(stdout, "       Current message level: 0x%08x (%d)\n"
                                "                              ",
                        edata.data, edata.data);
                print_flags(flags_msglvl, ARRAY_SIZE(flags_msglvl),
                            edata.data);
                fprintf(stdout, "\n");
                allfail = 0;
        } else if (errno != EOPNOTSUPP) {
                perror("Cannot get message level");
        }

        edata.cmd = ETHTOOL_GLINK;
        err = send_ioctl(ctx, &edata);
        if (err == 0) {
                fprintf(stdout, "       Link detected: %s\n",
                        edata.data ? "yes":"no");
                allfail = 0;
        } else if (errno != EOPNOTSUPP) {
                perror("Cannot get link status");
        }

        if (allfail) {
                fprintf(stdout, "No data available\n");
                return 75;
        }
        return 0;
}

static int do_sset(struct cmd_context *ctx)
{
        int speed_wanted = -1;
        int duplex_wanted = -1;
```

```
int port_wanted = -1;
int mdix_wanted = -1;
int autoneg_wanted = -1;
int phyad_wanted = -1;
int xcvr_wanted = -1;
int full_advertising_wanted = -1;
int advertising_wanted = -1;
int gset_changed = 0; /* did anything in GSET change? */
u32 wol_wanted = 0;
int wol_change = 0;
u8 sopass_wanted[SOPASS_MAX];
int sopass_change = 0;
int gwol_changed = 0; /* did anything in GWOL change? */
int msglvl_changed = 0;
u32 msglvl_wanted = 0;
u32 msglvl_mask = 0;
struct cmdline_info cmdline_msglvl[ARRAY_SIZE(flags_msglvl)];
int argc = ctx->argc;
char **argp = ctx->argp;
int i;
int err;

for (i = 0; i < ARRAY_SIZE(flags_msglvl); i++)
        flag_to_cmdline_info(flags_msglvl[i].name,
                             flags_msglvl[i].value,
                             &msglvl_wanted, &msglvl_mask,
                             &cmdline_msglvl[i]);

for (i = 0; i < argc; i++) {
        if (!strcmp(argp[i], "speed")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                speed_wanted = get_int(argp[i],10);
        } else if (!strcmp(argp[i], "duplex")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                if (!strcmp(argp[i], "half"))
                        duplex_wanted = DUPLEX_HALF;
                else if (!strcmp(argp[i], "full"))
                        duplex_wanted = DUPLEX_FULL;
                else
                        exit_bad_args();
        } else if (!strcmp(argp[i], "port")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                if (!strcmp(argp[i], "tp"))
                        port_wanted = PORT_TP;
                else if (!strcmp(argp[i], "aui"))
                        port_wanted = PORT_AUI;
                else if (!strcmp(argp[i], "bnc"))
                        port_wanted = PORT_BNC;
                else if (!strcmp(argp[i], "mii"))
                        port_wanted = PORT_MII;
                else if (!strcmp(argp[i], "fibre"))
                        port_wanted = PORT_FIBRE;
                else
                        exit_bad_args();
        } else if (!strcmp(argp[i], "mdix")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                if (!strcmp(argp[i], "auto"))
                        mdix_wanted = ETH_TP_MDI_AUTO;
                else if (!strcmp(argp[i], "on"))
                        mdix_wanted = ETH_TP_MDI_X;
                else if (!strcmp(argp[i], "off"))
                        mdix_wanted = ETH_TP_MDI;
                else
                        exit_bad_args();
        } else if (!strcmp(argp[i], "autoneg")) {
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                if (!strcmp(argp[i], "on")) {
                        gset_changed = 1;
                        autoneg_wanted = AUTONEG_ENABLE;
                } else if (!strcmp(argp[i], "off")) {
                        gset_changed = 1;
                        autoneg_wanted = AUTONEG_DISABLE;
                } else {
                        exit_bad_args();
                }
        } else if (!strcmp(argp[i], "advertise")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                full_advertising_wanted = get_int(argp[i], 16);
        } else if (!strcmp(argp[i], "phyad")) {
                gset_changed = 1;
                i += 1;
```

```
                if (i >= argc)
                        exit_bad_args();
                phyad_wanted = get_int(argp[i], 0);
        } else if (!strcmp(argp[i], "xcvr")) {
                gset_changed = 1;
                i += 1;
                if (i >= argc)
                        exit_bad_args();
                if (!strcmp(argp[i], "internal"))
                        xcvr_wanted = XCVR_INTERNAL;
                else if (!strcmp(argp[i], "external"))
                        xcvr_wanted = XCVR_EXTERNAL;
                else
                        exit_bad_args();
        } else if (!strcmp(argp[i], "wol")) {
                gwol_changed = 1;
                i++;
                if (i >= argc)
                        exit_bad_args();
                if (parse_wolopts(argp[i], &wol_wanted) < 0)
                        exit_bad_args();
                wol_change = 1;
        } else if (!strcmp(argp[i], "sopass")) {
                gwol_changed = 1;
                i++;
                if (i >= argc)
                        exit_bad_args();
                get_mac_addr(argp[i], sopass_wanted);
                sopass_change = 1;
        } else if (!strcmp(argp[i], "msglvl")) {
                i++;
                if (i >= argc)
                        exit_bad_args();
                if (isdigit((unsigned char)argp[i][0])) {
                        msglvl_changed = 1;
                        msglvl_mask = ~0;
                        msglvl_wanted =
                                get_uint_range(argp[i], 0,
                                               0xffffffff);
                } else {
                        ctx->argc -= i;
                        ctx->argp += i;
                        parse_generic_cmdline(
                                ctx, &msglvl_changed,
                                cmdline_msglvl,
                                ARRAY_SIZE(cmdline_msglvl));
                        break;
                }
        } else {
                exit_bad_args();
        }
}

if (full_advertising_wanted < 0) {
        /* User didn't supply a full advertisement bitfield:
         * construct one from the specified speed and duplex.
         */
        if (speed_wanted == SPEED_10 && duplex_wanted == DUPLEX_HALF)
                advertising_wanted = ADVERTISED_10baseT_Half;
        else if (speed_wanted == SPEED_10 &&
                 duplex_wanted == DUPLEX_FULL)
                advertising_wanted = ADVERTISED_10baseT_Full;
        else if (speed_wanted == SPEED_100 &&
                 duplex_wanted == DUPLEX_HALF)
                advertising_wanted = ADVERTISED_100baseT_Half;
        else if (speed_wanted == SPEED_100 &&
                 duplex_wanted == DUPLEX_FULL)
                advertising_wanted = ADVERTISED_100baseT_Full;
        else if (speed_wanted == SPEED_1000 &&
                 duplex_wanted == DUPLEX_HALF)
                advertising_wanted = ADVERTISED_1000baseT_Half;
        else if (speed_wanted == SPEED_1000 &&
                 duplex_wanted == DUPLEX_FULL)
                advertising_wanted = ADVERTISED_1000baseT_Full;
        else if (speed_wanted == SPEED_2500 &&
                 duplex_wanted == DUPLEX_FULL)
                advertising_wanted = ADVERTISED_2500baseX_Full;
        else if (speed_wanted == SPEED_10000 &&
                 duplex_wanted == DUPLEX_FULL)
                advertising_wanted = ADVERTISED_10000baseT_Full;
        else
                /* auto negotiate without forcing,
                 * all supported speed will be assigned below
                 */
                advertising_wanted = 0;
}

if (gset_changed) {
        struct ethtool_cmd ecmd;

        ecmd.cmd = ETHTOOL_GSET;
        err = send_ioctl(ctx, &ecmd);
        if (err < 0) {
                perror("Cannot get current device settings");
        } else {
                /* Change everything the user specified. */
                if (speed_wanted != -1)
                        ethtool_cmd_speed_set(&ecmd, speed_wanted);
```

```
                if (duplex_wanted != -1)
                        ecmd.duplex = duplex_wanted;
                if (port_wanted != -1)
                        ecmd.port = port_wanted;
                if (mdix_wanted != -1) {
                        /* check driver supports MDI-X */
                        if (ecmd.eth_tp_mdix_ctrl != ETH_TP_MDI_INVALID)
                                ecmd.eth_tp_mdix_ctrl = mdix_wanted;
                        else
                                fprintf(stderr, "setting MDI not supported\n");
                }
                if (autoneg_wanted != -1)
                        ecmd.autoneg = autoneg_wanted;
                if (phyad_wanted != -1)
                        ecmd.phy_address = phyad_wanted;
                if (xcvr_wanted != -1)
                        ecmd.transceiver = xcvr_wanted;
                /* XXX If the user specified speed or duplex
                 * then we should mask the advertised modes
                 * accordingly.  For now, warn that we aren't
                 * doing that.
                 */
                if ((speed_wanted != -1 || duplex_wanted != -1) &&
                    ecmd.autoneg && advertising_wanted == 0) {
                        fprintf(stderr, "Cannot advertise");
                        if (speed_wanted >= 0)
                                fprintf(stderr, " speed %d",
                                        speed_wanted);
                        if (duplex_wanted >= 0)
                                fprintf(stderr, " duplex %s",
                                        duplex_wanted ?
                                        "full" : "half");
                        fprintf(stderr, "\n");
                }
                if (autoneg_wanted == AUTONEG_ENABLE &&
                    advertising_wanted == 0) {
                        /* Auto negotiation enabled, but with
                         * unspecified speed and duplex: enable all
                         * supported speeds and duplexes.
                         */
                        ecmd.advertising =
                                (ecmd.advertising &
                                 ~ALL_ADVERTISED_MODES) |
                                (ALL_ADVERTISED_MODES & ecmd.supported);

                        /* If driver supports unknown flags, we cannot
                         * be sure that we enable all link modes.
                         */
                        if ((ecmd.supported & ALL_ADVERTISED_FLAGS) !=
                            ecmd.supported) {
                                fprintf(stderr, "Driver supports one "
                                        "or more unknown flags\n");
                        }
                } else if (advertising_wanted > 0) {
                        /* Enable all requested modes */
                        ecmd.advertising =
                                (ecmd.advertising &
                                 ~ALL_ADVERTISED_MODES) |
                                advertising_wanted;
                } else if (full_advertising_wanted > 0) {
                        ecmd.advertising = full_advertising_wanted;
                }

                /* Try to perform the update. */
                ecmd.cmd = ETHTOOL_SSET;
                err = send_ioctl(ctx, &ecmd);
                if (err < 0)
                        perror("Cannot set new settings");
        }
        if (err < 0) {
                if (speed_wanted != -1)
                        fprintf(stderr, "  not setting speed\n");
                if (duplex_wanted != -1)
                        fprintf(stderr, "  not setting duplex\n");
                if (port_wanted != -1)
                        fprintf(stderr, "  not setting port\n");
                if (autoneg_wanted != -1)
                        fprintf(stderr, "  not setting autoneg\n");
                if (phyad_wanted != -1)
                        fprintf(stderr, "  not setting phy_address\n");
                if (xcvr_wanted != -1)
                        fprintf(stderr, "  not setting transceiver\n");
                if (mdix_wanted != -1)
                        fprintf(stderr, "  not setting mdix\n");
        }
}

if (gwol_changed) {
        struct ethtool_wolinfo wol;

        wol.cmd = ETHTOOL_GWOL;
        err = send_ioctl(ctx, &wol);
        if (err < 0) {
                perror("Cannot get current wake-on-lan settings");
        } else {
                /* Change everything the user specified. */
                if (wol_change) {
                        wol.wolopts = wol_wanted;
```

```
                        }
                        if (sopass_change) {
                                int i;
                                for (i = 0; i < SOPASS_MAX; i++) {
                                        wol.sopass[i] = sopass_wanted[i];
                                }
                        }

                        /* Try to perform the update. */
                        wol.cmd = ETHTOOL_SWOL;
                        err = send_ioctl(ctx, &wol);
                        if (err < 0)
                                perror("Cannot set new wake-on-lan settings");
                }
                if (err < 0) {
                        if (wol_change)
                                fprintf(stderr, "  not setting wol\n");
                        if (sopass_change)
                                fprintf(stderr, "  not setting sopass\n");
                }
        }

        if (msglvl_changed) {
                struct ethtool_value edata;

                edata.cmd = ETHTOOL_GMSGLVL;
                err = send_ioctl(ctx, &edata);
                if (err < 0) {
                        perror("Cannot get msglvl");
                } else {
                        edata.cmd = ETHTOOL_SMSGLVL;
                        edata.data = ((edata.data & ~msglvl_mask) |
                                      msglvl_wanted);
                        err = send_ioctl(ctx, &edata);
                        if (err < 0)
                                perror("Cannot set new msglvl");
                }
        }

        return 0;
}

static int do_gregs(struct cmd_context *ctx)
{
        int gregs_changed = 0;
        int gregs_dump_raw = 0;
        int gregs_dump_hex = 0;
        char *gregs_dump_file = NULL;
        struct cmdline_info cmdline_gregs[] = {
                { "raw", CMDL_BOOL, &gregs_dump_raw, NULL },
                { "hex", CMDL_BOOL, &gregs_dump_hex, NULL },
                { "file", CMDL_STR, &gregs_dump_file, NULL },
        };
        int err;
        struct ethtool_drvinfo drvinfo;
        struct ethtool_regs *regs;

        parse_generic_cmdline(ctx, &gregs_changed,
                              cmdline_gregs, ARRAY_SIZE(cmdline_gregs));

        drvinfo.cmd = ETHTOOL_GDRVINFO;
        err = send_ioctl(ctx, &drvinfo);
        if (err < 0) {
                perror("Cannot get driver information");
                return 72;
        }

        regs = calloc(1, sizeof(*regs)+drvinfo.regdump_len);
        if (!regs) {
                perror("Cannot allocate memory for register dump");
                return 73;
        }
        regs->cmd = ETHTOOL_GREGS;
        regs->len = drvinfo.regdump_len;
        err = send_ioctl(ctx, regs);
        if (err < 0) {
                perror("Cannot get register dump");
                free(regs);
                return 74;
        }
        if (dump_regs(gregs_dump_raw, gregs_dump_hex, gregs_dump_file,
                      &drvinfo, regs) < 0) {
                fprintf(stderr, "Cannot dump registers\n");
                free(regs);
                return 75;
        }
        free(regs);

        return 0;
}

static int do_nway_rst(struct cmd_context *ctx)
{
        struct ethtool_value edata;
        int err;

        if (ctx->argc != 0)
                exit_bad_args();
```

```
	edata.cmd = ETHTOOL_NWAY_RST;
	err = send_ioctl(ctx, &edata);
	if (err < 0)
		perror("Cannot restart autonegotiation");

	return err;
}

static int do_geeprom(struct cmd_context *ctx)
{
	int geeprom_changed = 0;
	int geeprom_dump_raw = 0;
	u32 geeprom_offset = 0;
	u32 geeprom_length = -1;
	struct cmdline_info cmdline_geeprom[] = {
		{ "offset", CMDL_U32, &geeprom_offset, NULL },
		{ "length", CMDL_U32, &geeprom_length, NULL },
		{ "raw", CMDL_BOOL, &geeprom_dump_raw, NULL },
	};
	int err;
	struct ethtool_drvinfo drvinfo;
	struct ethtool_eeprom *eeprom;

	parse_generic_cmdline(ctx, &geeprom_changed,
			      cmdline_geeprom, ARRAY_SIZE(cmdline_geeprom));

	drvinfo.cmd = ETHTOOL_GDRVINFO;
	err = send_ioctl(ctx, &drvinfo);
	if (err < 0) {
		perror("Cannot get driver information");
		return 74;
	}

	if (geeprom_length == -1)
		geeprom_length = drvinfo.eedump_len;

	if (drvinfo.eedump_len < geeprom_offset + geeprom_length)
		geeprom_length = drvinfo.eedump_len - geeprom_offset;

	eeprom = calloc(1, sizeof(*eeprom)+geeprom_length);
	if (!eeprom) {
		perror("Cannot allocate memory for EEPROM data");
		return 75;
	}
	eeprom->cmd = ETHTOOL_GEEPROM;
	eeprom->len = geeprom_length;
	eeprom->offset = geeprom_offset;
	err = send_ioctl(ctx, eeprom);
	if (err < 0) {
		perror("Cannot get EEPROM data");
		free(eeprom);
		return 74;
	}
	err = dump_eeprom(geeprom_dump_raw, &drvinfo, eeprom);
	free(eeprom);

	return err;
}

static int do_seeprom(struct cmd_context *ctx)
{
	int seeprom_changed = 0;
	u32 seeprom_magic = 0;
	u32 seeprom_length = -1;
	u32 seeprom_offset = 0;
	u8 seeprom_value = 0;
	int seeprom_value_seen = 0;
	struct cmdline_info cmdline_seeprom[] = {
		{ "magic", CMDL_U32, &seeprom_magic, NULL },
		{ "offset", CMDL_U32, &seeprom_offset, NULL },
		{ "length", CMDL_U32, &seeprom_length, NULL },
		{ "value", CMDL_U8, &seeprom_value, NULL,
		  0, &seeprom_value_seen },
	};
	int err;
	struct ethtool_drvinfo drvinfo;
	struct ethtool_eeprom *eeprom;

	parse_generic_cmdline(ctx, &seeprom_changed,
			      cmdline_seeprom, ARRAY_SIZE(cmdline_seeprom));

	drvinfo.cmd = ETHTOOL_GDRVINFO;
	err = send_ioctl(ctx, &drvinfo);
	if (err < 0) {
		perror("Cannot get driver information");
		return 74;
	}

	if (seeprom_value_seen)
		seeprom_length = 1;

	if (seeprom_length == -1)
		seeprom_length = drvinfo.eedump_len;

	if (drvinfo.eedump_len < seeprom_offset + seeprom_length)
		seeprom_length = drvinfo.eedump_len - seeprom_offset;
```

```
        eeprom = calloc(1, sizeof(*eeprom)+seeprom_length);
        if (!eeprom) {
                perror("Cannot allocate memory for EEPROM data");
                return 75;
        }

        eeprom->cmd = ETHTOOL_SEEPROM;
        eeprom->len = seeprom_length;
        eeprom->offset = seeprom_offset;
        eeprom->magic = seeprom_magic;
        eeprom->data[0] = seeprom_value;

        /* Multi-byte write: read input from stdin */
        if (!seeprom_value_seen)
                eeprom->len = fread(eeprom->data, 1, eeprom->len, stdin);

        err = send_ioctl(ctx, eeprom);
        if (err < 0) {
                perror("Cannot set EEPROM data");
                err = 87;
        }
        free(eeprom);

        return err;
}

static int do_test(struct cmd_context *ctx)
{
        enum {
                ONLINE=0,
                OFFLINE,
                EXTERNAL_LB,
        } test_type;
        int err;
        struct ethtool_test *test;
        struct ethtool_gstrings *strings;

        if (ctx->argc > 1)
                exit_bad_args();
        if (ctx->argc == 1) {
                if (!strcmp(ctx->argp[0], "online")) {
                        test_type = ONLINE;
                } else if (!strcmp(*ctx->argp, "offline")) {
                        test_type = OFFLINE;
                } else if (!strcmp(*ctx->argp, "external_lb")) {
                        test_type = EXTERNAL_LB;
                } else {
                        exit_bad_args();
                }
        } else {
                test_type = OFFLINE;
        }

        strings = get_stringset(ctx, ETH_SS_TEST,
                                offsetof(struct ethtool_drvinfo, testinfo_len),
                                1);
        if (!strings) {
                perror("Cannot get strings");
                return 74;
        }

        test = calloc(1, sizeof(*test) + strings->len * sizeof(u64));
        if (!test) {
                perror("Cannot allocate memory for test info");
                free(strings);
                return 73;
        }
        memset(test->data, 0, strings->len * sizeof(u64));
        test->cmd = ETHTOOL_TEST;
        test->len = strings->len;
        if (test_type == EXTERNAL_LB)
                test->flags = (ETH_TEST_FL_OFFLINE | ETH_TEST_FL_EXTERNAL_LB);
        else if (test_type == OFFLINE)
                test->flags = ETH_TEST_FL_OFFLINE;
        else
                test->flags = 0;
        err = send_ioctl(ctx, test);
        if (err < 0) {
                perror("Cannot test");
                free (test);
                free(strings);
                return 74;
        }

        err = dump_test(test, strings);
        free(test);
        free(strings);

        return err;
}

static int do_phys_id(struct cmd_context *ctx)
{
        int err;
        struct ethtool_value edata;
        int phys_id_time;

        if (ctx->argc > 1)
```

```
                exit_bad_args();
        if (ctx->argc == 1)
                phys_id_time = get_int(*ctx->argp, 0);
        else
                phys_id_time = 0;

        edata.cmd = ETHTOOL_PHYS_ID;
        edata.data = phys_id_time;
        err = send_ioctl(ctx, &edata);
        if (err < 0)
                perror("Cannot identify NIC");

        return err;
}

static int do_gstats(struct cmd_context *ctx)
{
        struct ethtool_gstrings *strings;
        struct ethtool_stats *stats;
        unsigned int n_stats, sz_stats, i;
        int err;

        if (ctx->argc != 0)
                exit_bad_args();

        strings = get_stringset(ctx, ETH_SS_STATS,
                                offsetof(struct ethtool_drvinfo, n_stats),
                                0);
        if (!strings) {
                perror("Cannot get stats strings information");
                return 96;
        }

        n_stats = strings->len;
        if (n_stats < 1) {
                fprintf(stderr, "no stats available\n");
                free(strings);
                return 94;
        }

        sz_stats = n_stats * sizeof(u64);

        stats = calloc(1, sz_stats + sizeof(struct ethtool_stats));
        if (!stats) {
                fprintf(stderr, "no memory available\n");
                free(strings);
                return 95;
        }

        stats->cmd = ETHTOOL_GSTATS;
        stats->n_stats = n_stats;
        err = send_ioctl(ctx, stats);
        if (err < 0) {
                perror("Cannot get stats information");
                free(strings);
                free(stats);
                return 97;
        }

        /* todo - pretty-print the strings per-driver */
        fprintf(stdout, "NIC statistics:\n");
        for (i = 0; i < n_stats; i++) {
                fprintf(stdout, "     %.*s: %llu\n",
                        ETH_GSTRING_LEN,
                        &strings->data[i * ETH_GSTRING_LEN],
                        stats->data[i]);
        }
        free(strings);
        free(stats);

        return 0;
}

static int do_srxntuple(struct cmd_context *ctx,
                        struct ethtool_rx_flow_spec *rx_rule_fs);

static int do_srxclass(struct cmd_context *ctx)
{
        int err;

        if (ctx->argc < 2)
                exit_bad_args();

        if (ctx->argc == 3 && !strcmp(ctx->argp[0], "rx-flow-hash")) {
                int rx_fhash_set;
                u32 rx_fhash_val;
                struct ethtool_rxnfc nfccmd;

                rx_fhash_set = rxflow_str_to_type(ctx->argp[1]);
                if (!rx_fhash_set)
                        exit_bad_args();
                if (parse_rxfhashopts(ctx->argp[2], &rx_fhash_val) < 0)
                        exit_bad_args();

                nfccmd.cmd = ETHTOOL_SRXFH;
                nfccmd.flow_type = rx_fhash_set;
                nfccmd.data = rx_fhash_val;
```

```
                err = send_ioctl(ctx, &nfccmd);
                if (err < 0)
                        perror("Cannot change RX network flow hashing options");
        } else if (!strcmp(ctx->argp[0], "flow-type")) {
                struct ethtool_rx_flow_spec rx_rule_fs;

                ctx->argc--;
                ctx->argp++;
                if (rxclass_parse_ruleopts(ctx, &rx_rule_fs) < 0)
                        exit_bad_args();

                /* attempt to add rule via N-tuple specifier */
                err = do_srxntuple(ctx, &rx_rule_fs);
                if (!err)
                        return 0;

                /* attempt to add rule via network flow classifier */
                err = rxclass_rule_ins(ctx, &rx_rule_fs);
                if (err < 0) {
                        fprintf(stderr, "Cannot insert"
                                " classification rule\n");
                        return 1;
                }
        } else if (!strcmp(ctx->argp[0], "delete")) {
                int rx_class_rule_del =
                        get_uint_range(ctx->argp[1], 0, INT_MAX);

                err = rxclass_rule_del(ctx, rx_class_rule_del);

                if (err < 0) {
                        fprintf(stderr, "Cannot delete"
                                " classification rule\n");
                        return 1;
                }
        } else {
                exit_bad_args();
        }

        return 0;
}

static int do_grxclass(struct cmd_context *ctx)
{
        struct ethtool_rxnfc nfccmd;
        int err;

        if (ctx->argc == 2 && !strcmp(ctx->argp[0], "rx-flow-hash")) {
                int rx_fhash_get;

                rx_fhash_get = rxflow_str_to_type(ctx->argp[1]);
                if (!rx_fhash_get)
                        exit_bad_args();

                nfccmd.cmd = ETHTOOL_GRXFH;
                nfccmd.flow_type = rx_fhash_get;
                err = send_ioctl(ctx, &nfccmd);
                if (err < 0)
                        perror("Cannot get RX network flow hashing options");
                else
                        dump_rxfhash(rx_fhash_get, nfccmd.data);
        } else if (ctx->argc == 2 && !strcmp(ctx->argp[0], "rule")) {
                int rx_class_rule_get =
                        get_uint_range(ctx->argp[1], 0, INT_MAX);

                err = rxclass_rule_get(ctx, rx_class_rule_get);
                if (err < 0)
                        fprintf(stderr, "Cannot get RX classification rule\n");
        } else if (ctx->argc == 0) {
                nfccmd.cmd = ETHTOOL_GRXRINGS;
                err = send_ioctl(ctx, &nfccmd);
                if (err < 0)
                        perror("Cannot get RX rings");
                else
                        fprintf(stdout, "%d RX rings available\n",
                                (int)nfccmd.data);

                err = rxclass_rule_getall(ctx);
                if (err < 0)
                        fprintf(stderr, "RX classification rule retrieval failed\n");

        } else {
                exit_bad_args();
        }

        return err ? 1 : 0;
}

static void print_indir_table(struct cmd_context *ctx,
                              struct ethtool_rxnfc *ring_count,
                              u32 indir_size, u32 *indir)
{
        u32 i;

        printf("RX flow hash indirection table for %s with %llu RX ring(s):\n",
               ctx->devname, ring_count->data);

        if (!indir_size)
                printf("Operation not supported\n");
```

```
        for (i = 0; i < indir_size; i++) {
                if (i % 8 == 0)
                        printf("%5u: ", i);
                printf(" %5u", indir[i]);
                if (i % 8 == 7 || i == indir_size - 1)
                        fputc('\n', stdout);
        }
}

static int do_grxfhindir(struct cmd_context *ctx,
                         struct ethtool_rxnfc *ring_count)
{
        struct ethtool_rxfh_indir indir_head;
        struct ethtool_rxfh_indir *indir;
        int err;

        indir_head.cmd = ETHTOOL_GRXFHINDIR;
        indir_head.size = 0;
        err = send_ioctl(ctx, &indir_head);
        if (err < 0) {
                perror("Cannot get RX flow hash indirection table size");
                return 1;
        }

        indir = malloc(sizeof(*indir) +
                       indir_head.size * sizeof(*indir->ring_index));
        if (!indir) {
                perror("Cannot allocate memory for indirection table");
                return 1;
        }

        indir->cmd = ETHTOOL_GRXFHINDIR;
        indir->size = indir_head.size;
        err = send_ioctl(ctx, indir);
        if (err < 0) {
                perror("Cannot get RX flow hash indirection table");
                free(indir);
                return 1;
        }

        print_indir_table(ctx, ring_count, indir->size, indir->ring_index);

        free(indir);
        return 0;
}

static int do_grxfh(struct cmd_context *ctx)
{
        struct ethtool_rxfh rss_head = {0};
        struct ethtool_rxnfc ring_count;
        struct ethtool_rxfh *rss;
        u32 i, indir_bytes;
        char *hkey;
        int err;

        ring_count.cmd = ETHTOOL_GRXRINGS;
        err = send_ioctl(ctx, &ring_count);
        if (err < 0) {
                perror("Cannot get RX ring count");
                return 1;
        }

        rss_head.cmd = ETHTOOL_GRSSH;
        err = send_ioctl(ctx, &rss_head);
        if (err < 0 && errno == EOPNOTSUPP) {
                return do_grxfhindir(ctx, &ring_count);
        } else if (err < 0) {
                perror("Cannot get RX flow hash indir size and/or key size");
                return 1;
        }

        rss = calloc(1, sizeof(*rss) +
                        rss_head.indir_size * sizeof(rss_head.rss_config[0]) +
                        rss_head.key_size);
        if (!rss) {
                perror("Cannot allocate memory for RX flow hash config");
                return 1;
        }

        rss->cmd = ETHTOOL_GRSSH;
        rss->indir_size = rss_head.indir_size;
        rss->key_size = rss_head.key_size;
        err = send_ioctl(ctx, rss);
        if (err < 0) {
                perror("Cannot get RX flow hash configuration");
                free(rss);
                return 1;
        }

        print_indir_table(ctx, &ring_count, rss->indir_size, rss->rss_config);

        indir_bytes = rss->indir_size * sizeof(rss->rss_config[0]);
        hkey = ((char *)rss->rss_config + indir_bytes);

        printf("RSS hash key:\n");
        if (!rss->key_size)
                printf("Operation not supported\n");
```

```
        for (i = 0; i < rss->key_size; i++) {
                if (i == (rss->key_size - 1))
                        printf("%02x\n", (u8) hkey[i]);
                else
                        printf("%02x:", (u8) hkey[i]);
        }

        free(rss);
        return 0;
}

static int fill_indir_table(u32 *indir_size, u32 *indir, int rxfhindir_equal,
                            char **rxfhindir_weight, u32 num_weights)
{
        u32 i;
        /*
         * "*indir_size == 0" ==> reset indir to default
         */
        if (rxfhindir_equal) {
                for (i = 0; i < *indir_size; i++)
                        indir[i] = i % rxfhindir_equal;
        } else if (rxfhindir_weight) {
                u32 j, weight, sum = 0, partial = 0;

                for (j = 0; j < num_weights; j++) {
                        weight = get_u32(rxfhindir_weight[j], 0);
                        sum += weight;
                }

                if (sum == 0) {
                        fprintf(stderr,
                                "At least one weight must be non-zero\n");
                        return 2;
                }

                if (sum > *indir_size) {
                        fprintf(stderr,
                                "Total weight exceeds the size of the "
                                "indirection table\n");
                        return 2;
                }

                j = -1;
                for (i = 0; i < *indir_size; i++) {
                        while (i >= (*indir_size) * partial / sum) {
                                j += 1;
                                weight = get_u32(rxfhindir_weight[j], 0);
                                partial += weight;
                        }
                        indir[i] = j;
                }
        } else {
                *indir_size = ETH_RXFH_INDIR_NO_CHANGE;
        }

        return 0;
}

static int do_srxfhindir(struct cmd_context *ctx, int rxfhindir_equal,
                         char **rxfhindir_weight, u32 num_weights)
{
        struct ethtool_rxfh_indir indir_head;
        struct ethtool_rxfh_indir *indir;
        int err;

        indir_head.cmd = ETHTOOL_GRXFHINDIR;
        indir_head.size = 0;
        err = send_ioctl(ctx, &indir_head);
        if (err < 0) {
                perror("Cannot get RX flow hash indirection table size");
                return 1;
        }

        indir = malloc(sizeof(*indir) +
                       indir_head.size * sizeof(*indir->ring_index));

        if (!indir) {
                perror("Cannot allocate memory for indirection table");
                return 1;
        }

        indir->cmd = ETHTOOL_SRXFHINDIR;
        indir->size = indir_head.size;

        if (fill_indir_table(&indir->size, indir->ring_index, rxfhindir_equal,
                             rxfhindir_weight, num_weights)) {
                free(indir);
                return 1;
        }

        err = send_ioctl(ctx, indir);
        if (err < 0) {
                perror("Cannot set RX flow hash indirection table");
                free(indir);
                return 1;
        }
```

```
        free(indir);
        return 0;
}

static int do_srxfh(struct cmd_context *ctx)
{
        struct ethtool_rxfh rss_head = {0};
        struct ethtool_rxfh *rss;
        struct ethtool_rxnfc ring_count;
        int rxfhindir_equal = 0;
        char **rxfhindir_weight = NULL;
        char *rxfhindir_key = NULL;
        char *hkey = NULL;
        int err = 0;
        u32 arg_num = 0, indir_bytes = 0;
        u32 entry_size = sizeof(rss_head.rss_config[0]);
        u32 num_weights = 0;

        if (ctx->argc < 2)
                exit_bad_args();

        while (arg_num < ctx->argc) {
                if (!strcmp(ctx->argp[arg_num], "equal")) {
                        ++arg_num;
                        rxfhindir_equal = get_int_range(ctx->argp[arg_num],
                                                        0, 1, INT_MAX);
                        ++arg_num;
                } else if (!strcmp(ctx->argp[arg_num], "weight")) {
                        ++arg_num;
                        rxfhindir_weight = ctx->argp + arg_num;
                        while (arg_num < ctx->argc &&
                               isdigit((unsigned char)ctx->argp[arg_num][0])) {
                                ++arg_num;
                                ++num_weights;
                        }
                        if (!num_weights)
                                exit_bad_args();
                } else if (!strcmp(ctx->argp[arg_num], "hkey")) {
                        ++arg_num;
                        rxfhindir_key = ctx->argp[arg_num];
                        if (!rxfhindir_key)
                                exit_bad_args();
                        ++arg_num;
                } else {
                        exit_bad_args();
                }
        }

        if (rxfhindir_equal && rxfhindir_weight) {
                fprintf(stderr,
                        "Equal and weight options are mutually exclusive\n");
                return 1;
        }

        ring_count.cmd = ETHTOOL_GRXRINGS;
        err = send_ioctl(ctx, &ring_count);
        if (err < 0) {
                perror("Cannot get RX ring count");
                return 1;
        }

        rss_head.cmd = ETHTOOL_GRSSH;
        err = send_ioctl(ctx, &rss_head);
        if (err < 0 && errno == EOPNOTSUPP && !rxfhindir_key) {
                return do_srxfhindir(ctx, rxfhindir_equal, rxfhindir_weight,
                                     num_weights);
        } else if (err < 0) {
                perror("Cannot get RX flow hash indir size and key size");
                return 1;
        }

        if (rxfhindir_key) {
                err = parse_hkey(&hkey, rss_head.key_size,
                                 rxfhindir_key);
                if (err)
                        return err;
        }

        if (rxfhindir_equal || rxfhindir_weight)
                indir_bytes = rss_head.indir_size * entry_size;

        rss = calloc(1, sizeof(*rss) + indir_bytes + rss_head.key_size);
        if (!rss) {
                perror("Cannot allocate memory for RX flow hash config");
                return 1;
        }
        rss->cmd = ETHTOOL_SRSSH;
        rss->indir_size = rss_head.indir_size;
        rss->key_size = rss_head.key_size;

        if (fill_indir_table(&rss->indir_size, rss->rss_config, rxfhindir_equal,
                             rxfhindir_weight, num_weights)) {
                err = 1;
                goto free;
        }

        if (hkey)
                memcpy((char *)rss->rss_config + indir_bytes,
```

```
                        hkey, rss->key_size);
        else
                rss->key_size = 0;

        err = send_ioctl(ctx, rss);
        if (err < 0) {
                perror("Cannot set RX flow hash configuration");
                err = 1;
        }

free:
        if (hkey)
                free(hkey);

        free(rss);
        return err;
}

static int do_flash(struct cmd_context *ctx)
{
        char *flash_file;
        int flash_region;
        struct ethtool_flash efl;
        int err;

        if (ctx->argc < 1 || ctx->argc > 2)
                exit_bad_args();
        flash_file = ctx->argp[0];
        if (ctx->argc == 2) {
                flash_region = strtol(ctx->argp[1], NULL, 0);
                if (flash_region < 0)
                        exit_bad_args();
        } else {
                flash_region = -1;
        }

        if (strlen(flash_file) > ETHTOOL_FLASH_MAX_FILENAME - 1) {
                fprintf(stdout, "Filename too long\n");
                return 99;
        }

        efl.cmd = ETHTOOL_FLASHDEV;
        strcpy(efl.data, flash_file);

        if (flash_region < 0)
                efl.region = ETHTOOL_FLASH_ALL_REGIONS;
        else
                efl.region = flash_region;

        err = send_ioctl(ctx, &efl);
        if (err < 0)
                perror("Flashing failed");

        return err;
}

static int do_permaddr(struct cmd_context *ctx)
{
        int i, err;
        struct ethtool_perm_addr *epaddr;

        epaddr = malloc(sizeof(struct ethtool_perm_addr) + MAX_ADDR_LEN);
        epaddr->cmd = ETHTOOL_GPERMADDR;
        epaddr->size = MAX_ADDR_LEN;

        err = send_ioctl(ctx, epaddr);
        if (err < 0)
                perror("Cannot read permanent address");
        else {
                printf("Permanent address:");
                for (i = 0; i < epaddr->size; i++)
                        printf("%c%02x", (i == 0) ? ' ' : ':',
                                epaddr->data[i]);
                printf("\n");
        }
        free(epaddr);

        return err;
}

static int flow_spec_to_ntuple(struct ethtool_rx_flow_spec *fsp,
                               struct ethtool_rx_ntuple_flow_spec *ntuple)
{
        size_t i;

        /* verify location is not specified */
        if (fsp->location != RX_CLS_LOC_ANY)
                return -1;

        /* destination MAC address in L3/L4 rules is not supported by ntuple */
        if (fsp->flow_type & FLOW_MAC_EXT)
                return -1;

        /* verify ring cookie can transfer to action */
        if (fsp->ring_cookie > INT_MAX && fsp->ring_cookie < (u64)(-2))
                return -1;

        /* verify only one field is setting data field */
```

```
        if ((fsp->flow_type & FLOW_EXT) &&
            (fsp->m_ext.data[0] || fsp->m_ext.data[1]) &&
            fsp->m_ext.vlan_etype)
                return -1;

        /* Set entire ntuple to ~0 to guarantee all masks are set */
        memset(ntuple, ~0, sizeof(*ntuple));

        /* set non-filter values */
        ntuple->flow_type = fsp->flow_type;
        ntuple->action = fsp->ring_cookie;

        /*
         * Copy over header union, they are identical in layout however
         * the ntuple union contains additional padding on the end
         */
        memcpy(&ntuple->h_u, &fsp->h_u, sizeof(fsp->h_u));

        /*
         * The same rule mentioned above applies to the mask union.  However,
         * in addition we need to invert the mask bits to match the ntuple
         * mask which is 1 for masked, versus 0 for masked as seen in nfc.
         */
        memcpy(&ntuple->m_u, &fsp->m_u, sizeof(fsp->m_u));
        for (i = 0; i < sizeof(fsp->m_u); i++)
                ntuple->m_u.hdata[i] ^= 0xFF;

        /* copy extended fields */
        if (fsp->flow_type & FLOW_EXT) {
                ntuple->vlan_tag =
                        ntohs(fsp->h_ext.vlan_tci);
                ntuple->vlan_tag_mask =
                        ~ntohs(fsp->m_ext.vlan_tci);
                if (fsp->m_ext.vlan_etype) {
                        /*
                         * vlan_etype and user data are mutually exclusive
                         * in ntuple configuration as they occupy the same
                         * space.
                         */
                        if (fsp->m_ext.data[0] || fsp->m_ext.data[1])
                                return -1;
                        ntuple->data =
                                ntohl(fsp->h_ext.vlan_etype);
                        ntuple->data_mask =
                                ~(u64)ntohl(fsp->m_ext.vlan_etype);
                } else {
                        ntuple->data =
                                (u64)ntohl(fsp->h_ext.data[0]) << 32;
                        ntuple->data |=
                                (u64)ntohl(fsp->h_ext.data[1]);
                        ntuple->data_mask =
                                (u64)ntohl(~fsp->m_ext.data[0]) << 32;
                        ntuple->data_mask |=
                                (u64)ntohl(~fsp->m_ext.data[1]);
                }
        }

        /* Mask out the extended bit, because ntuple does not know it! */
        ntuple->flow_type &= ~FLOW_EXT;

        return 0;
}

static int do_srxntuple(struct cmd_context *ctx,
                        struct ethtool_rx_flow_spec *rx_rule_fs)
{
        struct ethtool_rx_ntuple ntuplecmd;
        struct ethtool_value eval;
        int err;

        /* attempt to convert the flow classifier to an ntuple classifier */
        err = flow_spec_to_ntuple(rx_rule_fs, &ntuplecmd.fs);
        if (err)
                return -1;

        /*
         * Check to see if the flag is set for N-tuple, this allows
         * us to avoid the possible EINVAL response for the N-tuple
         * flag not being set on the device
         */
        eval.cmd = ETHTOOL_GFLAGS;
        err = send_ioctl(ctx, &eval);
        if (err || !(eval.data & ETH_FLAG_NTUPLE))
                return -1;

        /* send rule via N-tuple */
        ntuplecmd.cmd = ETHTOOL_SRXNTUPLE;
        err = send_ioctl(ctx, &ntuplecmd);

        /*
         * Display error only if response is something other than op not
         * supported.  It is possible that the interface uses the network
         * flow classifier interface instead of N-tuple.
         */
        if (err < 0) {
                if (errno != EOPNOTSUPP)
                        perror("Cannot add new rule via N-tuple");
                return -1;
```

```
        }

        return 0;
}

static int do_writefwdump(struct ethtool_dump *dump, const char *dump_file)
{
        int err = 0;
        FILE *f;
        size_t bytes;

        f = fopen(dump_file, "wb+");

        if (!f) {
                fprintf(stderr, "Can't open file %s: %s\n",
                        dump_file, strerror(errno));
                return 1;
        }
        bytes = fwrite(dump->data, 1, dump->len, f);
        if (bytes != dump->len) {
                fprintf(stderr, "Can not write all of dump data\n");
                err = 1;
        }
        if (fclose(f)) {
                fprintf(stderr, "Can't close file %s: %s\n",
                        dump_file, strerror(errno));
                err = 1;
        }
        return err;
}

static int do_getfwdump(struct cmd_context *ctx)
{
        u32 dump_flag;
        char *dump_file;
        int err;
        struct ethtool_dump edata;
        struct ethtool_dump *data;

        if (ctx->argc == 2 && !strcmp(ctx->argp[0], "data")) {
                dump_flag = ETHTOOL_GET_DUMP_DATA;
                dump_file = ctx->argp[1];
        } else if (ctx->argc == 0) {
                dump_flag = 0;
                dump_file = NULL;
        } else {
                exit_bad_args();
        }

        edata.cmd = ETHTOOL_GET_DUMP_FLAG;

        err = send_ioctl(ctx, &edata);
        if (err < 0) {
                perror("Can not get dump level\n");
                return 1;
        }
        if (dump_flag != ETHTOOL_GET_DUMP_DATA) {
                fprintf(stdout, "flag: %u, version: %u, length: %u\n",
                        edata.flag, edata.version, edata.len);
                return 0;
        }
        data = calloc(1, offsetof(struct ethtool_dump, data) + edata.len);
        if (!data) {
                perror("Can not allocate enough memory\n");
                return 1;
        }
        data->cmd = ETHTOOL_GET_DUMP_DATA;
        data->len = edata.len;
        err = send_ioctl(ctx, data);
        if (err < 0) {
                perror("Can not get dump data\n");
                err = 1;
                goto free;
        }
        err = do_writefwdump(data, dump_file);
free:
        free(data);
        return err;
}

static int do_setfwdump(struct cmd_context *ctx)
{
        u32 dump_flag;
        int err;
        struct ethtool_dump dump;

        if (ctx->argc != 1)
                exit_bad_args();
        dump_flag = get_u32(ctx->argp[0], 0);

        dump.cmd = ETHTOOL_SET_DUMP;
        dump.flag = dump_flag;
        err = send_ioctl(ctx, &dump);
        if (err < 0) {
                perror("Can not set dump level\n");
                return 1;
        }
        return 0;
```

```
}

static int do_gprivflags(struct cmd_context *ctx)
{
        struct ethtool_gstrings *strings;
        struct ethtool_value flags;
        unsigned int i;
        int max_len = 0, cur_len;

        if (ctx->argc != 0)
                exit_bad_args();

        strings = get_stringset(ctx, ETH_SS_PRIV_FLAGS,
                                offsetof(struct ethtool_drvinfo, n_priv_flags),
                                1);
        if (!strings) {
                perror("Cannot get private flag names");
                return 1;
        }
        if (strings->len == 0) {
                fprintf(stderr, "No private flags defined\n");
                return 1;
        }
        if (strings->len > 32) {
                /* ETHTOOL_GPFLAGS can only cover 32 flags */
                fprintf(stderr, "Only showing first 32 private flags\n");
                strings->len = 32;
        }

        flags.cmd = ETHTOOL_GPFLAGS;
        if (send_ioctl(ctx, &flags)) {
                perror("Cannot get private flags");
                return 1;
        }

        /* Find longest string and align all strings accordingly */
        for (i = 0; i < strings->len; i++) {
                cur_len = strlen((const char*)strings->data +
                                 i * ETH_GSTRING_LEN);
                if (cur_len > max_len)
                        max_len = cur_len;
        }

        printf("Private flags for %s:\n", ctx->devname);
        for (i = 0; i < strings->len; i++)
                printf("%-*s: %s\n",
                       max_len,
                       (const char *)strings->data + i * ETH_GSTRING_LEN,
                       (flags.data & (1U << i)) ? "on" : "off");

        return 0;
}

static int do_sprivflags(struct cmd_context *ctx)
{
        struct ethtool_gstrings *strings;
        struct cmdline_info *cmdline;
        struct ethtool_value flags;
        u32 wanted_flags = 0, seen_flags = 0;
        int any_changed;
        unsigned int i;

        strings = get_stringset(ctx, ETH_SS_PRIV_FLAGS,
                                offsetof(struct ethtool_drvinfo, n_priv_flags),
                                1);
        if (!strings) {
                perror("Cannot get private flag names");
                return 1;
        }
        if (strings->len == 0) {
                fprintf(stderr, "No private flags defined\n");
                return 1;
        }
        if (strings->len > 32) {
                /* ETHTOOL_{G,S}PFLAGS can only cover 32 flags */
                fprintf(stderr, "Only setting first 32 private flags\n");
                strings->len = 32;
        }

        cmdline = calloc(strings->len, sizeof(*cmdline));
        if (!cmdline) {
                perror("Cannot parse arguments");
                return 1;
        }
        for (i = 0; i < strings->len; i++) {
                cmdline[i].name = ((const char *)strings->data +
                                   i * ETH_GSTRING_LEN);
                cmdline[i].type = CMDL_FLAG;
                cmdline[i].wanted_val = &wanted_flags;
                cmdline[i].flag_val = 1U << i;
                cmdline[i].seen_val = &seen_flags;
        }
        parse_generic_cmdline(ctx, &any_changed, cmdline, strings->len);
        free(cmdline);

        flags.cmd = ETHTOOL_GPFLAGS;
        if (send_ioctl(ctx, &flags)) {
                perror("Cannot get private flags");
```

```
                return 1;
        }

        flags.cmd = ETHTOOL_SPFLAGS;
        flags.data = (flags.data & ~seen_flags) | wanted_flags;
        if (send_ioctl(ctx, &flags)) {
                perror("Cannot set private flags");
                return 1;
        }

        return 0;
}

static int do_tsinfo(struct cmd_context *ctx)
{
        struct ethtool_ts_info info;

        if (ctx->argc != 0)
                exit_bad_args();

        fprintf(stdout, "Time stamping parameters for %s:\n", ctx->devname);
        info.cmd = ETHTOOL_GET_TS_INFO;
        if (send_ioctl(ctx, &info)) {
                perror("Cannot get device time stamping settings");
                return -1;
        }
        dump_tsinfo(&info);
        return 0;
}

static int do_getmodule(struct cmd_context *ctx)
{
        struct ethtool_modinfo modinfo;
        struct ethtool_eeprom *eeprom;
        u32 geeprom_offset = 0;
        u32 geeprom_length = -1;
        int geeprom_changed = 0;
        int geeprom_dump_raw = 0;
        int geeprom_dump_hex = 0;
        int err;

        struct cmdline_info cmdline_geeprom[] = {
                { "offset", CMDL_U32, &geeprom_offset, NULL },
                { "length", CMDL_U32, &geeprom_length, NULL },
                { "raw", CMDL_BOOL, &geeprom_dump_raw, NULL },
                { "hex", CMDL_BOOL, &geeprom_dump_hex, NULL },
        };

        parse_generic_cmdline(ctx, &geeprom_changed,
                              cmdline_geeprom, ARRAY_SIZE(cmdline_geeprom));

        if (geeprom_dump_raw && geeprom_dump_hex) {
                printf("Hex and raw dump cannot be specified together\n");
                return 1;
        }

        modinfo.cmd = ETHTOOL_GMODULEINFO;
        err = send_ioctl(ctx, &modinfo);
        if (err < 0) {
                perror("Cannot get module EEPROM information");
                return 1;
        }

        if (geeprom_length == -1)
                geeprom_length = modinfo.eeprom_len;

        if (modinfo.eeprom_len < geeprom_offset + geeprom_length)
                geeprom_length = modinfo.eeprom_len - geeprom_offset;

        eeprom = calloc(1, sizeof(*eeprom)+geeprom_length);
        if (!eeprom) {
                perror("Cannot allocate memory for Module EEPROM data");
                return 1;
        }

        eeprom->cmd = ETHTOOL_GMODULEEEPROM;
        eeprom->len = geeprom_length;
        eeprom->offset = geeprom_offset;
        err = send_ioctl(ctx, eeprom);
        if (err < 0) {
                perror("Cannot get Module EEPROM data");
                free(eeprom);
                return 1;
        }

        /*
         * SFF-8079 EEPROM layout contains the memory available at A0 address on
         * the PHY EEPROM.
         * SFF-8472 defines a virtual extension of the EEPROM, where the
         * microcontroller on the SFP/SFP+ generates a page at the A2 address,
         * which contains data relative to optical diagnostics.
         * The current kernel implementation returns a blob, which contains:
         *  - ETH_MODULE_SFF_8079 => The A0 page only.
         *  - ETH_MODULE_SFF_8472 => The A0 and A2 page concatenated.
         */
        if (geeprom_dump_raw) {
                fwrite(eeprom->data, 1, eeprom->len, stdout);
        } else {
```

```
                if (eeprom->offset != 0  ||
                    (eeprom->len != modinfo.eeprom_len)) {
                        geeprom_dump_hex = 1;
                } else if (!geeprom_dump_hex) {
                        switch (modinfo.type) {
#ifdef ETHTOOL_ENABLE_PRETTY_DUMP
                        case ETH_MODULE_SFF_8079:
                                sff8079_show_all(eeprom->data);
                                break;
                        case ETH_MODULE_SFF_8472:
                                sff8079_show_all(eeprom->data);
                                sff8472_show_all(eeprom->data);
                                break;
#endif
                        default:
                                geeprom_dump_hex = 1;
                                break;
                        }
                }
                if (geeprom_dump_hex)
                        dump_hex(stdout, eeprom->data,
                                 eeprom->len, eeprom->offset);
        }

        free(eeprom);

        return 0;
}

static int do_geee(struct cmd_context *ctx)
{
        struct ethtool_eee eeecmd;

        if (ctx->argc != 0)
                exit_bad_args();

        eeecmd.cmd = ETHTOOL_GEEE;
        if (send_ioctl(ctx, &eeecmd)) {
                perror("Cannot get EEE settings");
                return 1;
        }

        fprintf(stdout, "EEE Settings for %s:\n", ctx->devname);
        dump_eeecmd(&eeecmd);

        return 0;
}

static int do_seee(struct cmd_context *ctx)
{
        int adv_c = -1, lpi_c = -1, lpi_time_c = -1, eee_c = -1;
        int change = -1, change2 = 0;
        struct ethtool_eee eeecmd;
        struct cmdline_info cmdline_eee[] = {
                { "advertise",    CMDL_U32,  &adv_c,       &eeecmd.advertised },
                { "tx-lpi",       CMDL_BOOL, &lpi_c,   &eeecmd.tx_lpi_enabled },
                { "tx-timer",     CMDL_U32,  &lpi_time_c, &eeecmd.tx_lpi_timer},
                { "eee",          CMDL_BOOL, &eee_c,       &eeecmd.eee_enabled},
        };

        if (ctx->argc == 0)
                exit_bad_args();

        parse_generic_cmdline(ctx, &change, cmdline_eee,
                              ARRAY_SIZE(cmdline_eee));

        eeecmd.cmd = ETHTOOL_GEEE;
        if (send_ioctl(ctx, &eeecmd)) {
                perror("Cannot get EEE settings");
                return 1;
        }

        do_generic_set(cmdline_eee, ARRAY_SIZE(cmdline_eee), &change2);

        if (change2) {

                eeecmd.cmd = ETHTOOL_SEEE;
                if (send_ioctl(ctx, &eeecmd)) {
                        perror("Cannot set EEE settings");
                        return 1;
                }
        }

        return 0;
}

#ifndef TEST_ETHTOOL
int send_ioctl(struct cmd_context *ctx, void *cmd)
{
        ctx->ifr.ifr_data = cmd;
        return ioctl(ctx->fd, SIOCETHTOOL, &ctx->ifr);
}
#endif

static int show_usage(struct cmd_context *ctx);

static const struct option {
        const char *opts;
```

```
        int want_device;
        int (*func)(struct cmd_context *);
        char *help;
        char *opthelp;
} args[] = {
        { "-s|--change", 1, do_sset, "Change generic options",
          "             [ speed %d ]\n"
          "             [ duplex half|full ]\n"
          "             [ port tp|aui|bnc|mii|fibre ]\n"
          "             [ mdix auto|on|off ]\n"
          "             [ autoneg on|off ]\n"
          "             [ advertise %x ]\n"
          "             [ phyad %d ]\n"
          "             [ xcvr internal|external ]\n"
          "             [ wol p|u|m|b|a|g|s|d... ]\n"
          "             [ sopass %x:%x:%x:%x:%x:%x ]\n"
          "             [ msglvl %d | msglvl type on|off ... ]\n" },
        { "-a|--show-pause", 1, do_gpause, "Show pause options" },
        { "-A|--pause", 1, do_spause, "Set pause options",
          "             [ autoneg on|off ]\n"
          "             [ rx on|off ]\n"
          "             [ tx on|off ]\n" },
        { "-c|--show-coalesce", 1, do_gcoalesce, "Show coalesce options" },
        { "-C|--coalesce", 1, do_scoalesce, "Set coalesce options",
          "             [adaptive-rx on|off]\n"
          "             [adaptive-tx on|off]\n"
          "             [rx-usecs N]\n"
          "             [rx-frames N]\n"
          "             [rx-usecs-irq N]\n"
          "             [rx-frames-irq N]\n"
          "             [tx-usecs N]\n"
          "             [tx-frames N]\n"
          "             [tx-usecs-irq N]\n"
          "             [tx-frames-irq N]\n"
          "             [stats-block-usecs N]\n"
          "             [pkt-rate-low N]\n"
          "             [rx-usecs-low N]\n"
          "             [rx-frames-low N]\n"
          "             [tx-usecs-low N]\n"
          "             [tx-frames-low N]\n"
          "             [pkt-rate-high N]\n"
          "             [rx-usecs-high N]\n"
          "             [rx-frames-high N]\n"
          "             [tx-usecs-high N]\n"
          "             [tx-frames-high N]\n"
          "             [sample-interval N]\n" },
        { "-g|--show-ring", 1, do_gring, "Query RX/TX ring parameters" },
        { "-G|--set-ring", 1, do_sring, "Set RX/TX ring parameters",
          "             [ rx N ]\n"
          "             [ rx-mini N ]\n"
          "             [ rx-jumbo N ]\n"
          "             [ tx N ]\n" },
        { "-k|--show-features|--show-offload", 1, do_gfeatures,
          "Get state of protocol offload and other features" },
        { "-K|--features|--offload", 1, do_sfeatures,
          "Set protocol offload and other features",
          "             FEATURE on|off ...\n" },
        { "-i|--driver", 1, do_gdrv, "Show driver information" },
        { "-d|--register-dump", 1, do_gregs, "Do a register dump",
          "             [ raw on|off ]\n"
          "             [ file FILENAME ]\n" },
        { "-e|--eeprom-dump", 1, do_geeprom, "Do a EEPROM dump",
          "             [ raw on|off ]\n"
          "             [ offset N ]\n"
          "             [ length N ]\n" },
        { "-E|--change-eeprom", 1, do_seeprom,
          "Change bytes in device EEPROM",
          "             [ magic N ]\n"
          "             [ offset N ]\n"
          "             [ length N ]\n"
          "             [ value N ]\n" },
        { "-r|--negotiate", 1, do_nway_rst, "Restart N-WAY negotiation" },
        { "-p|--identify", 1, do_phys_id,
          "Show visible port identification (e.g. blinking)",
          "               [ TIME-IN-SECONDS ]\n" },
        { "-t|--test", 1, do_test, "Execute adapter self test",
          "               [ online | offline | external_lb ]\n" },
        { "-S|--statistics", 1, do_gstats, "Show adapter statistics" },
        { "-n|-u|--show-nfc|--show-ntuple", 1, do_grxclass,
          "Show Rx network flow classification options or rules",
          "             [ rx-flow-hash tcp4|udp4|ah4|esp4|sctp4|"
          "tcp6|udp6|ah6|esp6|sctp6 |\n"
          "               rule %d ]\n" },
        { "-N|-U|--config-nfc|--config-ntuple", 1, do_srxclass,
          "Configure Rx network flow classification options or rules",
          "             rx-flow-hash tcp4|udp4|ah4|esp4|sctp4|"
          "tcp6|udp6|ah6|esp6|sctp6 m|v|t|s|d|f|n|r... |\n"
          "             flow-type ether|ip4|tcp4|udp4|sctp4|ah4|esp4\n"
          "                     [ src %x:%x:%x:%x:%x:%x [m %x:%x:%x:%x:%x:%x] ]\n"
          "                     [ dst %x:%x:%x:%x:%x:%x [m %x:%x:%x:%x:%x:%x] ]\n"
          "                     [ proto %d [m %x] ]\n"
          "                     [ src-ip %d.%d.%d.%d [m %d.%d.%d.%d] ]\n"
          "                     [ dst-ip %d.%d.%d.%d [m %d.%d.%d.%d] ]\n"
          "                     [ tos %d [m %x] ]\n"
          "                     [ l4proto %d [m %x] ]\n"
          "                     [ src-port %d [m %x] ]\n"
          "                     [ dst-port %d [m %x] ]\n"
          "                     [ spi %d [m %x] ]\n"
```

```
          "                      [ vlan-etype %x [m %x] ]\n"
          "                      [ vlan %x [m %x] ]\n"
          "                      [ user-def %x [m %x] ]\n"
          "                      [ dst-mac %x:%x:%x:%x:%x:%x [m %x:%x:%x:%x:%x:%x] ]\n"
          "                      [ action %d ]\n"
          "                      [ loc %d]] |\n"
          "             delete %d\n" },
        { "-T|--show-time-stamping", 1, do_tsinfo,
          "Show time stamping capabilities" },
        { "-x|--show-rxfh-indir|--show-rxfh", 1, do_grxfh,
          "Show Rx flow hash indirection and/or hash key" },
        { "-X|--set-rxfh-indir|--rxfh", 1, do_srxfh,
          "Set Rx flow hash indirection and/or hash key",
          "             [ equal N | weight W0 W1 ... ]\n"
          "             [ hkey %x:%x:%x:%x:%x:.... ]\n" },
        { "-f|--flash", 1, do_flash,
          "Flash firmware image from the specified file to a region on the device",
          "               FILENAME [ REGION-NUMBER-TO-FLASH ]\n" },
        { "-P|--show-permaddr", 1, do_permaddr,
          "Show permanent hardware address" },
        { "-w|--get-dump", 1, do_getfwdump,
          "Get dump flag, data",
          "             [ data FILENAME ]\n" },
        { "-W|--set-dump", 1, do_setfwdump,
          "Set dump flag of the device",
          "             N\n"},
        { "-l|--show-channels", 1, do_gchannels, "Query Channels" },
        { "-L|--set-channels", 1, do_schannels, "Set Channels",
          "               [ rx N ]\n"
          "               [ tx N ]\n"
          "               [ other N ]\n"
          "               [ combined N ]\n" },
        { "--show-priv-flags" , 1, do_gprivflags, "Query private flags" },
        { "--set-priv-flags", 1, do_sprivflags, "Set private flags",
          "             FLAG on|off ...\n" },
        { "-m|--dump-module-eeprom|--module-info", 1, do_getmodule,
          "Query/Decode Module EEPROM information and optical diagnostics if available",
          "             [ raw on|off ]\n"
          "             [ hex on|off ]\n"
          "             [ offset N ]\n"
          "             [ length N ]\n" },
        { "--show-eee", 1, do_geee, "Show EEE settings"},
        { "--set-eee", 1, do_seee, "Set EEE settings",
          "             [ eee on|off ]\n"
          "             [ advertise %x ]\n"
          "             [ tx-lpi on|off ]\n"
          "             [ tx-timer %d ]\n"},
        { "-h|--help", 0, show_usage, "Show this help" },
        { "--version", 0, do_version, "Show version number" },
        {}
};

static int show_usage(struct cmd_context *ctx)
{
        int i;

        /* ethtool -h */
        fprintf(stdout, PACKAGE " version " VERSION "\n");
        fprintf(stdout,
                "Usage:\n"
                "        ethtool DEVNAME\t"
                "Display standard information about device\n");
        for (i = 0; args[i].opts; i++) {
                fputs("        ethtool ", stdout);
                fprintf(stdout, "%s %s\t%s\n",
                        args[i].opts,
                        args[i].want_device ? "DEVNAME" : "\t",
                        args[i].help);
                if (args[i].opthelp)
                        fputs(args[i].opthelp, stdout);
        }

        return 0;
}

int main(int argc, char **argp)
{
        int (*func)(struct cmd_context *);
        int want_device;
        struct cmd_context ctx;
        int k;

        /* Skip command name */
        argp++;
        argc--;

        /* First argument must be either a valid option or a device
         * name to get settings for (which we don't expect to begin
         * with '-').
         */
        if (argc == 0)
                exit_bad_args();
        for (k = 0; args[k].opts; k++) {
                const char *opt;
                size_t len;
                opt = args[k].opts;
                for (;;) {
                        len = strcspn(opt, "|");
```

```
                        if (strncmp(*argp, opt, len) == 0 &&
                            (*argp)[len] == 0) {
                                argp++;
                                argc--;
                                func = args[k].func;
                                want_device = args[k].want_device;
                                goto opt_found;
                        }
                        if (opt[len] == 0)
                                break;
                        opt += len + 1;
                }
        }
        if ((*argp)[0] == '-')
                exit_bad_args();
        func = do_gset;
        want_device = 1;

opt_found:
        if (want_device) {
                ctx.devname = *argp++;
                argc--;

                if (ctx.devname == NULL)
                        exit_bad_args();
                if (strlen(ctx.devname) >= IFNAMSIZ)
                        exit_bad_args();

                /* Setup our control structures. */
                memset(&ctx.ifr, 0, sizeof(ctx.ifr));
                strcpy(ctx.ifr.ifr_name, ctx.devname);

                /* Open control socket. */
                ctx.fd = socket(AF_INET, SOCK_DGRAM, 0);
                if (ctx.fd < 0) {
                        perror("Cannot get control socket");
                        return 70;
                }
        } else {
                ctx.fd = -1;
        }

        ctx.argc = argc;
        ctx.argp = argp;

        return func(&ctx);
}
```

Notice for package(s)

libcap

```
Unless otherwise *explicitly* stated, the following text describes the
licensed conditions under which the contents of this libcap release
may be used and distributed:

-----------------------------------------------------------------------
Redistribution and use in source and binary forms of libcap, with
or without modification, are permitted provided that the following
conditions are met:

1. Redistributions of source code must retain any existing copyright
   notice, and this entire permission notice in its entirety,
   including the disclaimer of warranties.

2. Redistributions in binary form must reproduce all prior and current
   copyright notices, this list of conditions, and the following
   disclaimer in the documentation and/or other materials provided
   with the distribution.

3. The name of any author may not be used to endorse or promote
   products derived from this software without their specific prior
   written permission.

ALTERNATIVELY, this product may be distributed under the terms of the
GNU General Public License (v2.0 - see below), in which case the
provisions of the GNU GPL are required INSTEAD OF the above
restrictions.  (This clause is necessary due to a potential conflict
between the GNU GPL and the restrictions contained in a BSD-style
copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
```

DAMAGE.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code. (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

  5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the

Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

```
  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```

Notice for package(s)

ntp

```
This file is automatically generated from html/copyright.html

  Copyright Notice

   jpg "Clone me," says Dolly sheepishly.

   Last update: 2-Jan-2017 11:58 UTC
     ____________________________________________________________________

   The following copyright notice applies to all files collectively
   called the Network Time Protocol Version 4 Distribution. Unless
   specifically declared otherwise in an individual file, this entire
   notice applies as if the text was explicitly included in the file.
********************************************************************
*                                                                  *
* Copyright (c) University of Delaware 1992-2015                   *
*                                                                  *
* Permission to use, copy, modify, and distribute this software and   *
* its documentation for any purpose with or without fee is hereby     *
* granted, provided that the above copyright notice appears in all    *
* copies and that both the copyright notice and this permission       *
* notice appear in supporting documentation, and that the name        *
* University of Delaware not be used in advertising or publicity      *
* pertaining to distribution of the software without specific,        *
* written prior permission. The University of Delaware makes no       *
* representations about the suitability this software for any         *
* purpose. It is provided "as is" without express or implied          *
* warranty.                                                        *
*                                                                  *
********************************************************************

   Content starting in 2011 from Harlan Stenn, Danny Mayer, and Martin
   Burnicki is:
********************************************************************
```

The following individuals contributed in part to the Network Time Protocol Distribution Version 4 and are acknowledged as authors of this work.

1. [1]Takao Abe <takao_abe@xurb.jp> Clock driver for JJY receivers
2. [2]Mark Andrews <mark_andrews@isc.org> Leitch atomic clock controller
3. [3]Bernd Altmeier <altmeier@atlsoft.de> hopf Elektronik serial line and PCI-bus devices
4. [4]Viraj Bais <vbais@mailman1.intel.com> and [5]Clayton Kirkwood <kirkwood@striderfm.intel.com> port to WindowsNT 3.5
5. [6]Michael Barone <michael,barone@lmco.com> GPSVME fixes
6. [7]Karl Berry <karl@owl.HQ.ileaf.com> syslog to file option
7. [8]Greg Brackley <greg.brackley@bigfoot.com> Major rework of WINNT port. Clean up recvbuf and iosignal code into separate modules.
8. [9]Marc Brett <Marc.Brett@westgeo.com> Magnavox GPS clock driver
9. [10]Piete Brooks <Piete.Brooks@cl.cam.ac.uk> MSF clock driver, Trimble PARSE support
10. [11]Nelson B Bolyard <nelson@bolyard.me> update and complete broadcast and crypto features in sntp
11. [12]Jean-Francois Boudreault <Jean-Francois.Boudreault@viagenie.qc.ca> IPv6 support
12. [13]Reg Clemens <reg@dwf.com> Oncore driver (Current maintainer)
13. [14]Steve Clift <clift@ml.csiro.au> OMEGA clock driver
14. [15]Casey Crellin <casey@csc.co.za> vxWorks (Tornado) port and help with target configuration
15. [16]Sven Dietrich <sven_dietrich@trimble.com> Palisade reference clock driver, NT adj. residuals, integrated Greg's Winnt port.
16. [17]John A. Dundas III <dundas@salt.jpl.nasa.gov> Apple A/UX port
17. [18]Torsten Duwe <duwe@immd4.informatik.uni-erlangen.de> Linux port
18. [19]Dennis Ferguson <dennis@mrbill.canet.ca> foundation code for NTP Version 2 as specified in RFC-1119
19. [20]John Hay <jhay@icomtek.csir.co.za> IPv6 support and testing
20. [21]Dave Hart <davehart@davehart.com> General maintenance, Windows port interpolation rewrite
21. [22]Claas Hilbrecht <neoclock4x@linum.com> NeoClock4X clock driver
22. [23]Glenn Hollinger <glenn@herald.usask.ca> GOES clock driver
23. [24]Mike Iglesias <iglesias@uci.edu> DEC Alpha port
24. [25]Jim Jagielski <jim@jagubox.gsfc.nasa.gov> A/UX port
25. [26]Jeff Johnson <jbj@chatham.usdesign.com> massive prototyping overhaul
26. [27]Hans Lambermont <Hans.Lambermont@nl.origin-it.com> or [28]<H.Lambermont@chello.nl> ntpsweep
27. [29]Poul-Henning Kamp <phk@FreeBSD.ORG> Oncore driver (Original author)
28. [30]Frank Kardel [31]<kardel (at) ntp (dot) org> PARSE <GENERIC> (driver 14 reference clocks), STREAMS modules for PARSE, support scripts, syslog cleanup, dynamic interface handling
29. [32]Johannes Maximilian Kuehn <kuehn@ntp.org> Rewrote sntp to comply with NTPv4 specification, ntpq saveconfig
30. [33]William L. Jones <jones@hermes.chpc.utexas.edu> RS/6000 AIX modifications, HPUX modifications
31. [34]Dave Katz <dkatz@cisco.com> RS/6000 AIX port
32. [35]Craig Leres <leres@ee.lbl.gov> 4.4BSD port, ppsclock, Magnavox GPS clock driver
33. [36]George Lindholm <lindholm@ucs.ubc.ca> SunOS 5.1 port
34. [37]Louis A. Mamakos <louie@ni.umd.edu> MD5-based authentication
35. [38]Lars H. Mathiesen <thorinn@diku.dk> adaptation of foundation code for Version 3 as specified in RFC-1305
36. [39]Danny Mayer <mayer@ntp.org>Network I/O, Windows Port, Code Maintenance
37. [40]David L. Mills <mills@udel.edu> Version 4 foundation, precision kernel; clock drivers: 1, 3, 4, 6, 7, 11, 13, 18, 19, 22, 36
38. [41]Wolfgang Moeller <moeller@gwdgv1.dnet.gwdg.de> VMS port
39. [42]Jeffrey Mogul <mogul@pa.dec.com> ntptrace utility
40. [43]Tom Moore <tmoore@fievel.daytonoh.ncr.com> i386 svr4 port
41. [44]Kamal A Mostafa <kamal@whence.com> SCO OpenServer port
42. [45]Derek Mulcahy <derek@toybox.demon.co.uk> and [46]Damon

Hart-Davis <d@hd.org> ARCRON MSF clock driver
43. [47]Rob Neal <neal@ntp.org> Bancomm refclock and config/parse code maintenance
44. [48]Rainer Pruy <Rainer.Pruy@informatik.uni-erlangen.de> monitoring/trap scripts, statistics file handling
45. [49]Dirce Richards <dirce@zk3.dec.com> Digital UNIX V4.0 port
46. [50]Wilfredo S&#65533;nchez <wsanchez@apple.com> added support for NetInfo
47. [51]Nick Sayer <mrapple@quack.kfu.com> SunOS streams modules
48. [52]Jack Sasportas <jack@innovativeinternet.com> Saved a Lot of space on the stuff in the html/pic/ subdirectory
49. [53]Ray Schnitzler <schnitz@unipress.com> Unixware1 port
50. [54]Michael Shields <shields@tembel.org> USNO clock driver
51. [55]Jeff Steinman <jss@pebbles.jpl.nasa.gov> Datum PTS clock driver
52. [56]Harlan Stenn <harlan@pfcs.com> GNU automake/autoconfigure makeover, various other bits (see the ChangeLog)
53. [57]Kenneth Stone <ken@sdd.hp.com> HP-UX port
54. [58]Ajit Thyagarajan <ajit@ee.udel.edu>IP multicast/anycast support
55. [59]Tomoaki TSURUOKA <tsuruoka@nc.fukuoka-u.ac.jp>TRAK clock driver
56. [60]Brian Utterback <brian.utterback@oracle.com> General codebase, Solaris issues
57. [61]Loganaden Velvindron <loganaden@gmail.com> Sandboxing (libseccomp) support
58. [62]Paul A Vixie <vixie@vix.com> TrueTime GPS driver, generic TrueTime clock driver
59. [63]Ulrich Windl <Ulrich.Windl@rz.uni-regensburg.de> corrected and validated HTML documents according to the HTML DTD

References

1. mailto:%20takao_abe@xurb.jp
2. mailto:%20mark_andrews@isc.org
3. mailto:%20altmeier@atlsoft.de
4. mailto:%20vbais@mailman1.intel.co
5. mailto:%20kirkwood@striderfm.intel.com
6. mailto:%20michael.barone@lmco.com
7. mailto:%20karl@owl.HQ.ileaf.com
8. mailto:%20greg.brackley@bigfoot.com
9. mailto:%20Marc.Brett@westgeo.com
10. mailto:%20Piete.Brooks@cl.cam.ac.uk
11. mailto:%20nelson@bolyard.me
12. mailto:%20Jean-Francois.Boudreault@viagenie.qc.ca
13. mailto:%20reg@dwf.com
14. mailto:%20clift@ml.csiro.au
15. mailto:%20casey@csc.co.za
16. mailto:%20Sven_Dietrich@trimble.COM
17. mailto:%20dundas@salt.jpl.nasa.gov
18. mailto:%20duwe@immd4.informatik.uni-erlangen.de
19. mailto:%20dennis@mrbill.canet.ca
20. mailto:%20jhay@icomtek.csir.co.za
21. mailto:%20davehart@davehart.com
22. mailto:%20neoclock4x@linum.com
23. mailto:%20glenn@herald.usask.ca
24. mailto:%20iglesias@uci.edu
25. mailto:%20jagubox.gsfc.nasa.gov
26. mailto:%20jbj@chatham.usdesign.com
27. mailto:%20Hans.Lambermont@nl.origin-it.com
28. mailto:H.Lambermont@chello.nl
29. mailto:%20phk@FreeBSD.ORG
30. http://www4.informatik.uni-erlangen.de/%7ekardel
31. mailto:%20kardel%20%28at%29%20ntp%20%28dot%29%20org
32. mailto:kuehn@ntp.org
33. mailto:%20jones@hermes.chpc.utexas.edu
34. mailto:%20dkatz@cisco.com
35. mailto:%20leres@ee.lbl.gov
36. mailto:%20lindholm@ucs.ubc.ca
37. mailto:%20louie@ni.umd.edu
38. mailto:%20thorinn@diku.dk
39. mailto:%20mayer@ntp.org
40. mailto:%20mills@udel.edu
41. mailto:%20moeller@gwdgv1.dnet.gwdg.de
42. mailto:%20mogul@pa.dec.com
43. mailto:%20tmoore@fievel.daytonoh.ncr.com
44. mailto:%20kamal@whence.com
45. mailto:%20derek@toybox.demon.co.uk
46. mailto:%20d@hd.org
47. mailto:%20neal@ntp.org
48. mailto:%20Rainer.Pruy@informatik.uni-erlangen.de
49. mailto:%20dirce@zk3.dec.com
50. mailto:%20wsanchez@apple.com
51. mailto:%20mrapple@quack.kfu.com
52. mailto:%20jack@innovativeinternet.com
53. mailto:%20schnitz@unipress.com
54. mailto:%20shields@tembel.org
55. mailto:%20pebbles.jpl.nasa.gov
56. mailto:%20harlan@pfcs.com
57. mailto:%20ken@sdd.hp.com
58. mailto:%20ajit@ee.udel.edu
59. mailto:%20tsuruoka@nc.fukuoka-u.ac.jp
60. mailto:%20brian.utterback@oracle.com
61. mailto:%20loganaden@gmail.com
62. mailto:%20vixie@vix.com
63. mailto:%20Ulrich.Windl@rz.uni-regensburg.de

Notice for package(s)

ntp

```
NTP License (NTP)

Copyright (c) (CopyrightHoldersName) (From 4-digit-year)-(To 4-digit-year)

Permission to use, copy, modify, and distribute this software and its documentation for any purpose with or without fee is hereby granted, prc
```

Notice for package(s)

perl

```
                    GNU GENERAL PUBLIC LICENSE
                     Version 1, February 1989

 Copyright (C) 1989 Free Software Foundation, Inc.
                    51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The license agreements of most software companies try to keep users
at the mercy of those companies.  By contrast, our General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  The
General Public License applies to the Free Software Foundation's
software and to any other program whose authors commit to using it.
You can use it for your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Specifically, the General Public License is designed to make
sure that you have the freedom to give away or sell copies of free
software, that you receive source code or can get it if you want it,
that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of a such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must tell them their rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any program or other work which
contains a notice placed by the copyright holder saying it may be
distributed under the terms of this General Public License.  The
"Program", below, refers to any such program or work, and a "work based
on the Program" means either the Program or any work containing the
Program or a portion of it, either verbatim or with modifications.  Each
licensee is addressed as "you".

  1. You may copy and distribute verbatim copies of the Program's source
code as you receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this
General Public License and to the absence of any warranty; and give any
other recipients of the Program a copy of this General Public License
along with the Program.  You may charge a fee for the physical act of
```

transferring a copy.

  2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

    a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

    b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

    d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

  3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

    a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

    b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

    c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

  4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

  5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

  6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

  7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

  8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and

of promoting the sharing and reuse of software generally.

NO WARRANTY

  9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to humanity, the best way to achieve this is to make it
free software which everyone can redistribute and change under these
terms.

  To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 1, or (at your option)
    any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19xx name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the
commands you use may be called something other than `show w' and `show
c'; they could even be mouse-clicks or menu items--whatever suits your
program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  program `Gnomovision' (a program to direct compilers to make passes
  at assemblers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

That's all there is to it!

Notice for package(s)

perl

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

> "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
>
> "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.
>
> "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
>
> "You" is you, if you're thinking about copying or distributing this Package.
>
> "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
>
> "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

    a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

    b) use the modified Package only within your corporation or organization.

    c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

    d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

    a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

    b) accompany the distribution with the machine-readable source of the Package with your modifications.

    c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

    d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated

them, and may be sold commercially, and may be aggregated with this
Package. If such scripts or library files are aggregated with this
Package via the so-called "undump" or "unexec" methods of producing a
binary executable image, then distribution of such an image shall
neither be construed as a distribution of this Package nor shall it
fall under the restrictions of Paragraphs 3 and 4, provided that you do
not represent such an executable image as a Standard Version of this
Package.

7. C subroutines (or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to
emulate subroutines and variables of the language defined by this
Package shall not be considered part of this Package, but are the
equivalent of input as in Paragraph 6, provided these subroutines do
not change the language in any way that would cause it to fail the
regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always
permitted provided that the use of this Package is embedded; that is,
when no overt attempt is made to make this Package's interfaces visible
to the end user of the commercial distribution. Such use shall not be
construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

                                The End

Notice for package(s)

perl

GNU General Public License, version 1

                    GNU GENERAL PUBLIC LICENSE
                     Version 1, February 1989

 Copyright (C) 1989 Free Software Foundation, Inc.
                    51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The license agreements of most software companies try to keep users
at the mercy of those companies. By contrast, our General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. The
General Public License applies to the Free Software Foundation`s
software and to any other program whose authors commit to using it.
You can use it for your programs, too.

  When we speak of free software, we are referring to freedom, not
price. Specifically, the General Public License is designed to make
sure that you have the freedom to give away or sell copies of free
software, that you receive source code or can get it if you want it,
that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of a such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must tell them their rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author`s protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors` reputations.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program`s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients` exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program`s name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 1, or (at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w`.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c` for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision` (a program to direct compilers to make passes
at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That`s all there is to it!

Notice for package(s)

libevent

```
Libevent is available for use under the following license, commonly known
as the 3-clause (or "modified") BSD license:

==============================
Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
Copyright (c) 2007-2012 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
==============================

Portions of Libevent are based on works by others, also made available by
them under the three-clause BSD license above.  The copyright notices are
available in the corresponding source files; the license is as above.  Here's
a list:

log.c:
   Copyright (c) 2000 Dug Song <dugsong@monkey.org>
   Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:
   Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32select.c:
   Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:
   Copyright (c) 2007 Sun Microsystems

ht-internal.h:
   Copyright (c) 2002 Christopher Clark

minheap-internal.h:
   Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

==============================

The arc4module is available under the following, sometimes called the
"OpenBSD" license:

   Copyright (c) 1996, David Mazieres <dm@uun.org>
   Copyright (c) 2008, Damien Miller <djm@openbsd.org>

   Permission to use, copy, modify, and distribute this software for any
   purpose with or without fee is hereby granted, provided that the above
   copyright notice and this permission notice appear in all copies.

   THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
   WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
   MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
   ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
   WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
   ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
   OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

Notice for package(s)

iproute2

```
/*
 * ip.c         "ip" utility frontend.
 *
 *              This program is free software; you can redistribute it and/or
 *              modify it under the terms of the GNU General Public License
 *              as published by the Free Software Foundation; either version
 *              2 of the License, or (at your option) any later version.
 *
 * Authors:     Alexey Kuznetsov, <kuznet@ms2.inr.ac.ru>
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <syslog.h>
#include <fcntl.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <errno.h>

#include "SNAPSHOT.h"
#include "utils.h"
#include "ip_common.h"
#include "namespace.h"
#include "color.h"

int preferred_family = AF_UNSPEC;
int human_readable = 0;
int use_iec = 0;
int show_stats = 0;
int show_details = 0;
int resolve_hosts = 0;
int oneline = 0;
int timestamp = 0;
char * _SL_ = NULL;
int force = 0;
int max_flush_loops = 10;
int batch_mode = 0;
bool do_all = false;

struct rtnl_handle rth = { .fd = -1 };

static void usage(void) __attribute__((noreturn));

static void usage(void)
{
        fprintf(stderr,
"Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }\n"
"       ip [ -force ] -batch filename\n"
"where  OBJECT := { link | addr | addrlabel | route | rule | neigh | ntable |\n"
"                   tunnel | tuntap | maddr | mroute | mrule | monitor | xfrm |\n"
"                   netns | l2tp | fou | tcp_metrics | token | netconf }\n"
"       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |\n"
"                    -h[uman-readable] | -iec |\n"
"                    -f[amily] { inet | inet6 | ipx | dnet | mpls | bridge | link } |\n"
"                    -4 | -6 | -I | -D | -B | -0 |\n"
"                    -l[oops] { maximum-addr-flush-attempts } |\n"
"                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |\n"
"                    -rc[vbuf] [size] | -n[etns] name | -a[ll] | -c[olor]}\n");
        exit(-1);
}

static int do_help(int argc, char **argv)
{
        usage();
        return 0;
}

static const struct cmd {
        const char *cmd;
        int (*func)(int argc, char **argv);
} cmds[] = {
        { "address",    do_ipaddr },
        { "addrlabel",  do_ipaddrlabel },
        { "maddress",   do_multiaddr },
        { "route",      do_iproute },
        { "rule",       do_iprule },
        { "neighbor",   do_ipneigh },
        { "neighbour",  do_ipneigh },
        { "ntable",     do_ipntable },
        { "ntbl",       do_ipntable },
        { "link",       do_iplink },
        { "l2tp",       do_ipl2tp },
        { "fou",        do_ipfou },
        { "tunnel",     do_iptunnel },
        { "tunl",       do_iptunnel },
        { "tuntap",     do_iptuntap },
        { "tap",        do_iptuntap },
        { "token",      do_iptoken },
        { "tcpmetrics", do_tcp_metrics },
        { "tcp_metrics",do_tcp_metrics },
        { "monitor",    do_ipmonitor },
        { "xfrm",       do_xfrm },
        { "mroute",     do_multiroute },
        { "mrule",      do_multirule },
        { "netns",      do_netns },
        { "netconf",    do_ipnetconf },
```

```
        { "help",       do_help },
        { 0 }
};

static int do_cmd(const char *argv0, int argc, char **argv)
{
        const struct cmd *c;

        for (c = cmds; c->cmd; ++c) {
                if (matches(argv0, c->cmd) == 0) {
                        return -(c->func(argc-1, argv+1));
                }
        }

        fprintf(stderr, "Object \"%s\" is unknown, try \"ip help\".\n", argv0);
        return EXIT_FAILURE;
}

static int batch(const char *name)
{
        char *line = NULL;
        size_t len = 0;
        int ret = EXIT_SUCCESS;

        batch_mode = 1;

        if (name && strcmp(name, "-") != 0) {
                if (freopen(name, "r", stdin) == NULL) {
                        fprintf(stderr, "Cannot open file \"%s\" for reading: %s\n",
                                name, strerror(errno));
                        return EXIT_FAILURE;
                }
        }

        if (rtnl_open(&rth, 0) < 0) {
                fprintf(stderr, "Cannot open rtnetlink\n");
                return EXIT_FAILURE;
        }

        cmdlineno = 0;
        while (getcmdline(&line, &len, stdin) != -1) {
                char *largv[100];
                int largc;

                largc = makeargs(line, largv, 100);
                if (largc == 0)
                        continue;       /* blank line */

                if (do_cmd(largv[0], largc, largv)) {
                        fprintf(stderr, "Command failed %s:%d\n", name, cmdlineno);
                        ret = EXIT_FAILURE;
                        if (!force)
                                break;
                }
        }
        if (line)
                free(line);

        rtnl_close(&rth);
        return ret;
}


int main(int argc, char **argv)
{
        char *basename;
        char *batch_file = NULL;

        basename = strrchr(argv[0], '/');
        if (basename == NULL)
                basename = argv[0];
        else
                basename++;

        while (argc > 1) {
                char *opt = argv[1];
                if (strcmp(opt,"--") == 0) {
                        argc--; argv++;
                        break;
                }
                if (opt[0] != '-')
                        break;
                if (opt[1] == '-')
                        opt++;
                if (matches(opt, "-loops") == 0) {
                        argc--;
                        argv++;
                        if (argc <= 1)
                                usage();
                        max_flush_loops = atoi(argv[1]);
                } else if (matches(opt, "-family") == 0) {
                        argc--;
                        argv++;
                        if (argc <= 1)
                                usage();
                        if (strcmp(argv[1], "help") == 0)
                                usage();
                        else
```

```
                        preferred_family = read_family(argv[1]);
                if (preferred_family == AF_UNSPEC)
                        invarg("invalid protocol family", argv[1]);
        } else if (strcmp(opt, "-4") == 0) {
                preferred_family = AF_INET;
        } else if (strcmp(opt, "-6") == 0) {
                preferred_family = AF_INET6;
        } else if (strcmp(opt, "-0") == 0) {
                preferred_family = AF_PACKET;
        } else if (strcmp(opt, "-I") == 0) {
                preferred_family = AF_IPX;
        } else if (strcmp(opt, "-D") == 0) {
                preferred_family = AF_DECnet;
        } else if (strcmp(opt, "-M") == 0) {
                preferred_family = AF_MPLS;
        } else if (strcmp(opt, "-B") == 0) {
                preferred_family = AF_BRIDGE;
        } else if (matches(opt, "-human") == 0 ||
                   matches(opt, "-human-readable") == 0) {
                ++human_readable;
        } else if (matches(opt, "-iec") == 0) {
                ++use_iec;
        } else if (matches(opt, "-stats") == 0 ||
                   matches(opt, "-statistics") == 0) {
                ++show_stats;
        } else if (matches(opt, "-details") == 0) {
                ++show_details;
        } else if (matches(opt, "-resolve") == 0) {
                ++resolve_hosts;
        } else if (matches(opt, "-oneline") == 0) {
                ++oneline;
        } else if (matches(opt, "-timestamp") == 0) {
                ++timestamp;
        } else if (matches(opt, "-tshort") == 0) {
                ++timestamp;
                ++timestamp_short;
#if 0
        } else if (matches(opt, "-numeric") == 0) {
                rtnl_names_numeric++;
#endif
        } else if (matches(opt, "-Version") == 0) {
                printf("ip utility, iproute2-ss%s\n", SNAPSHOT);
                exit(0);
        } else if (matches(opt, "-force") == 0) {
                ++force;
        } else if (matches(opt, "-batch") == 0) {
                argc--;
                argv++;
                if (argc <= 1)
                        usage();
                batch_file = argv[1];
        } else if (matches(opt, "-rcvbuf") == 0) {
                unsigned int size;

                argc--;
                argv++;
                if (argc <= 1)
                        usage();
                if (get_unsigned(&size, argv[1], 0)) {
                        fprintf(stderr, "Invalid rcvbuf size '%s'\n",
                                argv[1]);
                        exit(-1);
                }
                rcvbuf = size;
        } else if (matches(opt, "-color") == 0) {
                enable_color();
        } else if (matches(opt, "-help") == 0) {
                usage();
        } else if (matches(opt, "-netns") == 0) {
                NEXT_ARG();
                if (netns_switch(argv[1]))
                        exit(-1);
        } else if (matches(opt, "-all") == 0) {
                do_all = true;
        } else {
                fprintf(stderr, "Option \"%s\" is unknown, try \"ip -help\".\n", opt);
                exit(-1);
        }
        argc--; argv++;
    }

    _SL_ = oneline ? "\\" : "\n" ;

    if (batch_file)
            return batch(batch_file);

    if (rtnl_open(&rth, 0) < 0)
            exit(1);

    if (strlen(basename) > 2)
            return do_cmd(basename+2, argc, argv);

    if (argc > 1)
            return do_cmd(argv[1], argc-1, argv+1);

    rtnl_close(&rth);
    usage();
}
```

Notice for package(s)

libmnl
libtool
util-linux
xz

---

are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application. Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

  It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

  7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

    b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

  11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or

```
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

gmp
nettle

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                      Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


  This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

  0. Additional Definitions.

  As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

  "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

  An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

  A "Combined Work" is a work produced by combining or linking an
Application with the Library.  The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

  The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

  The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

  1. Exception to Section 3 of the GNU GPL.

  You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

  2. Conveying Modified Versions.

  If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
   whatever part of its purpose remains meaningful, or
```

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser

General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

  If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

Notice for package(s)

gmp
nettle

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, su
0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for por

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and
1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.
2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that

    * a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the functi
    * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object cod

    * a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered b
    * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of th

    * a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered
    * b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
    * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notice
    * d) Do one of the following:
        o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suita
        o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a c
    * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities

    * a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conve
    * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the acco

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new vers

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall app

---

Notice for package(s)

libnftnl

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

```
    Copyright (C) 19yy  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA


Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

nftables

```
nftables is distributed under the terms of the GPL version 2. Note that
*only* version 2 of the GPL applies, not "any later version".

Patrick McHardy <kaber@trash.net>

------------------------------------------------------------------------------

                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
               51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.
```

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

```
You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Library General
Public License instead of this License.
```

Notice for package(s)

nettle

```
/* serpent-decrypt.c

   The serpent block cipher.

   For more details on this algorithm, see the Serpent website at
   http://www.cl.cam.ac.uk/~rja14/serpent.html

   Copyright (C) 2011  Niels Möller
   Copyright (C) 2010, 2011  Simon Josefsson
   Copyright (C) 2003, 2004, 2005 Free Software Foundation, Inc.

   This file is part of GNU Nettle.

   GNU Nettle is free software: you can redistribute it and/or
   modify it under the terms of either:

     * the GNU Lesser General Public License as published by the Free
       Software Foundation; either version 3 of the License, or (at your
       option) any later version.

   or

     * the GNU General Public License as published by the Free
       Software Foundation; either version 2 of the License, or (at your
       option) any later version.

   or both in parallel, as here.

   GNU Nettle is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   General Public License for more details.

   You should have received copies of the GNU General Public License and
   the GNU Lesser General Public License along with this program.  If
   not, see http://www.gnu.org/licenses/.
*/

/* This file is derived from cipher/serpent.c in Libgcrypt v1.4.6.
   The adaption to Nettle was made by Simon Josefsson on 2010-12-07
   with final touches on 2011-05-30.  Changes include replacing
   libgcrypt with nettle in the license template, renaming
   serpent_context to serpent_ctx, renaming u32 to uint32_t, removing
   libgcrypt stubs and selftests, modifying entry function prototypes,
   using FOR_BLOCKS to iterate through data in encrypt/decrypt, using
   LE_READ_UINT32 and LE_WRITE_UINT32 to access data in
   encrypt/decrypt, and running indent on the code. */

#if HAVE_CONFIG_H
#include "config.h"
#endif

#include <assert.h>
#include <limits.h>

#include "serpent.h"

#include "macros.h"
#include "serpent-internal.h"

/* These are the S-Boxes of Serpent.  They are copied from Serpents
   reference implementation (the optimized one, contained in
   `floppy2') and are therefore:

     Copyright (C) 1998 Ross Anderson, Eli Biham, Lars Knudsen.

  To quote the Serpent homepage
  (http://www.cl.cam.ac.uk/~rja14/serpent.html):

  "Serpent is now completely in the public domain, and we impose no
```

```
   restrictions on its use.  This was announced on the 21st August at
   the First AES Candidate Conference. The optimised implementations
   in the submission package are now under the GNU PUBLIC LICENSE
   (GPL), although some comments in the code still say otherwise. You
   are welcome to use Serpent for any application."  */

/* S0 inverse:  13  3 11  0 10  6  5 12  1 14  4  7 15  9  8  2 */
/* Original single-assignment form:

     t01 = x2  ^ x3;
     t02 = x0  | x1;
     t03 = x1  | x2;
     t04 = x2  & t01;
     t05 = t02 ^ t01;
     t06 = x0  | t04;
     y2  =     ~ t05;
     t08 = x1  ^ x3;
     t09 = t03 & t08;
     t10 = x3  | y2;
     y1  = t09 ^ t06;
     t12 = x0  | t05;
     t13 = y1  ^ t12;
     t14 = t03 ^ t10;
     t15 = x0  ^ x2;
     y3  = t14 ^ t13;
     t17 = t05 & t13;
     t18 = t14 | t17;
     y0  = t15 ^ t18;
*/
#define SBOX0_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)           \
  do {                                                          \
    y0  = x0 ^ x2;                                              \
    y2  = x0 | x1;                                              \
    y1  = x2 ^ x3;                                              \
    y2 ^= y1;                                                   \
    y1 &= x2;                                                   \
    x2 |= x1;                                                   \
    x1 ^= x3;                                                   \
    y1 |= x0;                                                   \
    x1 &= x2;                                                   \
    y1 ^= x1;                                                   \
    x0 |= y2;                                                   \
    x0 ^= y1;                                                   \
    x1  = y2 & x0;                                              \
    y2  = ~ y2;                                                 \
    x3 |= y2;                                                   \
    x3 ^= x2;                                                   \
    y3  = x3 ^ x0;                                              \
    x1 |= x3;                                                   \
    y0 ^= x1;                                                   \
  } while (0)

/* S1 inverse:   5  8  2 14 15  6 12  3 11  4  7  9  1 13 10  0 */
/* Original single-assignment form:
     t01 = x0  ^ x1;
     t02 = x1  | x3;
     t03 = x0  & x2;
     t04 = x2  ^ t02;
     t05 = x0  | t04;
     t06 = t01 & t05;
     t07 = x3  | t03;
     t08 = x1  ^ t06;
     t09 = t07 ^ t06;
     t10 = t04 | t03;
     t11 = x3  & t08;
     y2  =     ~ t09;
     y1  = t10 ^ t11;
     t14 = x0  | y2;
     t15 = t06 ^ y1;
     y3  = t01 ^ t04;
     t17 = x2  ^ t15;
     y0  = t14 ^ t17;
*/
#define SBOX1_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)           \
  do {                                                          \
    y1  = x1 | x3;                                              \
    y1 ^= x2;                                                   \
    y3  = x0 ^ x1;                                              \
    y0  = x0 | y1;                                              \
    y0 &= y3;                                                   \
    x1 ^= y0;                                                   \
    y3 ^= y1;                                                   \
    x1 &= x3;                                                   \
    y2  = x0 & x2;                                              \
    y1 |= y2;                                                   \
    y2 |= x3;                                                   \
    y2 ^= y0;                                                   \
    y2  = ~ y2;                                                 \
    y1 ^= x1;                                                   \
    y0 ^= y1;                                                   \
    y0 ^= x2;                                                   \
    x0 |= y2;                                                   \
    y0 ^= x0;                                                   \
  } while (0)

/* S2 inverse:  12  9 15  4 11 14  1  2  0  3  6 13  5  8 10  7 */
/* Original single-assignment form:
     t01 = x0  ^ x3;
```

```
     t02 = x2  ^ x3;
     t03 = x0  & x2;
     t04 = x1  | t02;
     y0  = t01 ^ t04;
     t06 = x0  | x2;
     t07 = x3  | y0;
     t08 =     ~ x3;
     t09 = x1  & t06;
     t10 = t08 | t03;
     t11 = x1  & t07;
     t12 = t06 & t02;
     y3  = t09 ^ t10;
     y1  = t12 ^ t11;
     t15 = x2  & y3;
     t16 = y0  ^ y1;
     t17 = t10 ^ t15;
     y2  = t16 ^ t17;
*/
#define SBOX2_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)           \
  do {                                                          \
    y0  = x0 ^ x3;                                              \
    y2  = x2 ^ x3;                                              \
    y1  = x1 | y2;                                              \
    y0 ^= y1;                                                   \
    y1  = x3 | y0;                                              \
    y1 &= x1;                                                   \
    x3  = ~ x3;                                                 \
    y3  = x0 | x2;                                              \
    y2 &= y3;                                                   \
    y1 ^= y2;                                                   \
    y3 &= x1;                                                   \
    x0 &= x2;                                                   \
    x0 |= x3;                                                   \
    y3 ^= x0;                                                   \
    x2 &= y3;                                                   \
    x2 ^= x0;                                                   \
    y2  = y0 ^ y1;                                              \
    y2 ^= x2;                                                   \
  } while (0)

/* S3 inverse:   0  9 10  7 11 14  6 13  3  5 12  2  4  8 15  1 */
/* Original single-assignment form:
     t01 = x2  | x3;
     t02 = x0  | x3;
     t03 = x2  ^ t02;
     t04 = x1  ^ t02;
     t05 = x0  ^ x3;
     t06 = t04 & t03;
     t07 = x1  & t01;
     y2  = t05 ^ t06;
     t09 = x0  ^ t03;
     y0  = t07 ^ t03;
     t11 = y0  | t05;
     t12 = t09 & t11;
     t13 = x0  & y2;
     t14 = t01 ^ t05;
     y1  = x1  ^ t12;
     t16 = x1  | t13;
     y3  = t14 ^ t16;
*/
#define SBOX3_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)           \
  do {                                                          \
    y3  = x2 | x3;                                              \
    y0  = x1 & y3;                                              \
    y2  = x0 | x3;                                              \
    y1  = x2 ^ y2;                                              \
    y0 ^= y1;                                                   \
    x3 ^= x0;                                                   \
    y3 ^= x3;                                                   \
    y2 ^= x1;                                                   \
    y2 &= y1;                                                   \
    y2 ^= x3;                                                   \
    y1 ^= x0;                                                   \
    x3 |= y0;                                                   \
    y1 &= x3;                                                   \
    y1 ^= x1;                                                   \
    x0 &= y2;                                                   \
    x0 |= x1;                                                   \
    y3 ^= x0;                                                   \
  } while (0)

/* S4 inverse:   5  0  8  3 10  9  7 14  2 12 11  6  4 15 13  1 */
/* Original single-assignment form:
     t01 = x1  | x3;
     t02 = x2  | x3;
     t03 = x0  & t01;
     t04 = x1  ^ t02;
     t05 = x2  ^ x3;
     t06 =     ~ t03;
     t07 = x0  & t04;
     y1  = t05 ^ t07;
     t09 = y1  | t06;
     t10 = x0  ^ t07;
     t11 = t01 ^ t09;
     t12 = x3  ^ t04;
     t13 = x2  | t10;
     y3  = t03 ^ t12;
     t15 = x0  ^ t04;
```

```
     y2  = t11 ^ t13;
     y0  = t15 ^ t09;
*/
#define SBOX4_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)       \
  do {                                                      \
    y1  = x2 ^ x3;                                          \
    y2  = x2 | x3;                                          \
    y2 ^= x1;                                               \
    x1 |= x3;                                               \
    y0  = x0 ^ y2;                                          \
    x3 ^= y2;                                               \
    y2 &= x0;                                               \
    y1 ^= y2;                                               \
    y2 ^= x0;                                               \
    y2 |= x2;                                               \
    x0 &=  x1;                                              \
    y3  = x0 ^ x3;                                          \
    x0  = ~ x0;                                             \
    x0 |= y1;                                               \
    y0 ^= x0;                                               \
    x0 ^= x1;                                               \
    y2 ^= x0;                                               \
  } while (0)

/* S5 inverse:   8 15  2  9  4  1 13 14 11  6  5  3  7 12 10  0 */
/* Original single-assignment form:
     t01 = x0  & x3;
     t02 = x2  ^ t01;
     t03 = x0  ^ x3;
     t04 = x1  & t02;
     t05 = x0  & x2;
     y0  = t03 ^ t04;
     t07 = x0  & y0;
     t08 = t01 ^ y0;
     t09 = x1  | t05;
     t10 =     ~ x1;
     y1  = t08 ^ t09;
     t12 = t10 | t07;
     t13 = y0  | y1;
     y3  = t02 ^ t12;
     t15 = t02 ^ t13;
     t16 = x1  ^ x3;
     y2  = t16 ^ t15;
*/
#define SBOX5_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)       \
  do {                                                      \
    y1  = x0 & x3;                                          \
    y3  = x2 ^ y1;                                          \
    y0  = x1 & y3;                                          \
    y2  = x0 ^ x3;                                          \
    x3 ^= x1;                                               \
    y0 ^= y2;                                               \
    x2 &= x0;                                               \
    x0 &= y0;                                               \
    x2 |= x1;                                               \
    y1 ^= y0;                                               \
    y1 ^= x2;                                               \
    y2  = y0 | y1;                                          \
    y2 ^= y3;                                               \
    y2 ^= x3;                                               \
    x1  = ~ x1;                                             \
    x1 |= x0;                                               \
    y3 ^= x1;                                               \
  } while (0)

/* S6 inverse:  15 10  1 13  5  3  6  0  4  9 14  7  2 12  8 11 */
/* Original single-assignment form:
     t01 = x0  ^ x2;
     t02 =     ~ x2;
     t03 = x1  & t01;
     t04 = x1  | t02;
     t05 = x3  | t03;
     t06 = x1  ^ x3;
     t07 = x0  & t04;
     t08 = x0  | t02;
     t09 = t07 ^ t05;
     y1  = t06 ^ t08;
     y0  =     ~ t09;
     t12 = x1  & y0;
     t13 = t01 & t05;
     t14 = t01 ^ t12;
     t15 = t07 ^ t13;
     t16 = x3  | t02;
     t17 = x0  ^ y1;
     y3  = t17 ^ t15;
     y2  = t16 ^ t14;
 */
#define SBOX6_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)       \
  do {                                                      \
    y2  = x0 ^ x2;                                          \
    x2  = ~ x2;                                             \
    y0  = x1 ^ x3;                                          \
    y1  = x0 | x2;                                          \
    y1 ^= y0;                                               \
    y3  = x1 & y2;                                          \
    y3 |= x3;                                               \
    x3 |= x2;                                               \
    x2 |= x1;                                               \
```

```
    x2 &= x0;                                                   \
    y0  = x2 ^ y3;                                              \
    y0  = ~ y0;                                                 \
    y3 &= y2;                                                   \
    y3 ^= x2;                                                   \
    x0 ^= y1;                                                   \
    y3 ^= x0;                                                   \
    x1 &= y0;                                                   \
    y2 ^= x1;                                                   \
    y2 ^= x3;                                                   \
  } while (0)

/* S7 inverse:   3  0  6 13  9 14 15  8  5 12 11  7 10  1  4  2 */
/* Original single-assignment form:
     t01 = x0  & x1;
     t02 = x0  | x1;
     t03 = x2  | t01;
     t04 = x3  & t02;
     y3  = t03 ^ t04;
     t06 = x1  ^ t04;
     t07 = x3  ^ y3;
     t08 =     ~ t07;
     t09 = t06 | t08;
     t10 = x1  ^ x3;
     t11 = x0  | x3;
     y1  = x0  ^ t09;
     t13 = x2  ^ t06;
     t14 = x2  & t11;
     t15 = x3  | y1;
     t16 = t01 | t10;
     y0  = t13 ^ t15;
     y2  = t14 ^ t16;
*/
#define SBOX7_INVERSE(x0, x1, x2, x3, y0, y1, y2, y3)       \
  do {                                                      \
    y3  = x0 & x1;                                          \
    y2  = x1 ^ x3;                                          \
    y2 |= y3;                                               \
    y1  = x0 | x3;                                          \
    y1 &= x2;                                               \
    y2 ^= y1;                                               \
    y3 |= x2;                                               \
    y0  = x0 | x1;                                          \
    y0 &= x3;                                               \
    y3 ^= y0;                                               \
    y0 ^= x1;                                               \
    y1  = x3 ^ y3;                                          \
    y1  = ~ y1;                                             \
    y1 |= y0;                                               \
    y0 ^= x2;                                               \
    y1 ^= x0;                                               \
    x3 |= y1;                                               \
    y0 ^= x3;                                               \
  } while (0)

/* In-place inverse linear transformation.  */
#define LINEAR_TRANSFORMATION_INVERSE(x0,x1,x2,x3)       \
  do {                                                   \
    x2 = ROTL32 (10, x2);                    \
    x0 = ROTL32 (27, x0);                    \
    x2 = x2 ^ x3 ^ (x1 << 7); \
    x0 = x0 ^ x1 ^ x3;        \
    x3 = ROTL32 (25, x3);                     \
    x1 = ROTL32 (31, x1);                     \
    x3 = x3 ^ x2 ^ (x0 << 3); \
    x1 = x1 ^ x0 ^ x2;        \
    x2 = ROTL32 (29, x2);                     \
    x0 = ROTL32 (19, x0);                    \
  } while (0)

/* Round inputs are x0,x1,x2,x3 (destroyed), and round outputs are
   y0,y1,y2,y3. */
#define ROUND_INVERSE(which, subkey, x0,x1,x2,x3, y0,y1,y2,y3) \
  do {                                                         \
    LINEAR_TRANSFORMATION_INVERSE (x0,x1,x2,x3);               \
    SBOX##which##_INVERSE(x0,x1,x2,x3, y0,y1,y2,y3);           \
    KEYXOR(y0,y1,y2,y3, subkey);                               \
  } while (0)

#if HAVE_NATIVE_64_BIT

/* In-place inverse linear transformation.  */
#define LINEAR_TRANSFORMATION64_INVERSE(x0,x1,x2,x3)     \
  do {                                                   \
    x2 = DROTL32 (10, x2);                    \
    x0 = DROTL32 (27, x0);                    \
    x2 = x2 ^ x3 ^ DRSHIFT32(7, x1); \
    x0 = x0 ^ x1 ^ x3;        \
    x3 = DROTL32 (25, x3);                     \
    x1 = DROTL32 (31, x1);                     \
    x3 = x3 ^ x2 ^ DRSHIFT32(3, x0); \
    x1 = x1 ^ x0 ^ x2;        \
    x2 = DROTL32 (29, x2);                     \
    x0 = DROTL32 (19, x0);                    \
  } while (0)

#define ROUND64_INVERSE(which, subkey, x0,x1,x2,x3, y0,y1,y2,y3) \
  do {                                                           \
```

```
    LINEAR_TRANSFORMATION64_INVERSE (x0,x1,x2,x3);             \
    SBOX##which##_INVERSE(x0,x1,x2,x3, y0,y1,y2,y3);           \
    KEYXOR64(y0,y1,y2,y3, subkey);                             \
  } while (0)

#endif /* HAVE_NATIVE_64_BIT */

void
serpent_decrypt (const struct serpent_ctx *ctx,
                 size_t length, uint8_t * dst, const uint8_t * src)
{
  assert( !(length % SERPENT_BLOCK_SIZE));

#if HAVE_NATIVE_64_BIT
  if (length & SERPENT_BLOCK_SIZE)
#else
  while (length >= SERPENT_BLOCK_SIZE)
#endif
    {
      uint32_t x0,x1,x2,x3, y0,y1,y2,y3;
      unsigned k;

      x0 = LE_READ_UINT32 (src);
      x1 = LE_READ_UINT32 (src + 4);
      x2 = LE_READ_UINT32 (src + 8);
      x3 = LE_READ_UINT32 (src + 12);

      /* Inverse of special round */
      KEYXOR (x0,x1,x2,x3, ctx->keys[32]);
      SBOX7_INVERSE (x0,x1,x2,x3, y0,y1,y2,y3);
      KEYXOR (y0,y1,y2,y3, ctx->keys[31]);

      k = 24;
      goto start32;
      while (k > 0)
        {
          k -= 8;
          ROUND_INVERSE (7, ctx->keys[k+7], x0,x1,x2,x3, y0,y1,y2,y3);
        start32:
          ROUND_INVERSE (6, ctx->keys[k+6], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND_INVERSE (5, ctx->keys[k+5], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND_INVERSE (4, ctx->keys[k+4], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND_INVERSE (3, ctx->keys[k+3], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND_INVERSE (2, ctx->keys[k+2], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND_INVERSE (1, ctx->keys[k+1], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND_INVERSE (0, ctx->keys[k], y0,y1,y2,y3, x0,x1,x2,x3);
        }

      LE_WRITE_UINT32 (dst, x0);
      LE_WRITE_UINT32 (dst + 4, x1);
      LE_WRITE_UINT32 (dst + 8, x2);
      LE_WRITE_UINT32 (dst + 12, x3);

      src += SERPENT_BLOCK_SIZE;
      dst += SERPENT_BLOCK_SIZE;
      length -= SERPENT_BLOCK_SIZE;
    }
#if HAVE_NATIVE_64_BIT
  FOR_BLOCKS(length, dst, src, 2*SERPENT_BLOCK_SIZE)
    {
      uint64_t x0,x1,x2,x3, y0,y1,y2,y3;
      unsigned k;

      x0 = LE_READ_UINT32 (src);
      x1 = LE_READ_UINT32 (src + 4);
      x2 = LE_READ_UINT32 (src + 8);
      x3 = LE_READ_UINT32 (src + 12);

      x0 <<= 32; x0 |= LE_READ_UINT32 (src + 16);
      x1 <<= 32; x1 |= LE_READ_UINT32 (src + 20);
      x2 <<= 32; x2 |= LE_READ_UINT32 (src + 24);
      x3 <<= 32; x3 |= LE_READ_UINT32 (src + 28);

      /* Inverse of special round */
      KEYXOR64 (x0,x1,x2,x3, ctx->keys[32]);
      SBOX7_INVERSE (x0,x1,x2,x3, y0,y1,y2,y3);
      KEYXOR64 (y0,y1,y2,y3, ctx->keys[31]);

      k = 24;
      goto start64;
      while (k > 0)
        {
          k -= 8;
          ROUND64_INVERSE (7, ctx->keys[k+7], x0,x1,x2,x3, y0,y1,y2,y3);
        start64:
          ROUND64_INVERSE (6, ctx->keys[k+6], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND64_INVERSE (5, ctx->keys[k+5], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND64_INVERSE (4, ctx->keys[k+4], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND64_INVERSE (3, ctx->keys[k+3], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND64_INVERSE (2, ctx->keys[k+2], y0,y1,y2,y3, x0,x1,x2,x3);
          ROUND64_INVERSE (1, ctx->keys[k+1], x0,x1,x2,x3, y0,y1,y2,y3);
          ROUND64_INVERSE (0, ctx->keys[k], y0,y1,y2,y3, x0,x1,x2,x3);
        }

      LE_WRITE_UINT32 (dst + 16, x0);
      LE_WRITE_UINT32 (dst + 20, x1);
      LE_WRITE_UINT32 (dst + 24, x2);
      LE_WRITE_UINT32 (dst + 28, x3);
```

```
      x0 >>= 32; LE_WRITE_UINT32 (dst, x0);
      x1 >>= 32; LE_WRITE_UINT32 (dst + 4, x1);
      x2 >>= 32; LE_WRITE_UINT32 (dst + 8, x2);
      x3 >>= 32; LE_WRITE_UINT32 (dst + 12, x3);
    }
#endif /* HAVE_NATIVE_64_BIT */
}
```

Notice for package(s)

nettle

```
/* serpent-set-key.c

   The serpent block cipher.

   For more details on this algorithm, see the Serpent website at
   http://www.cl.cam.ac.uk/~rja14/serpent.html

   Copyright (C) 2011, 2014  Niels Möller
   Copyright (C) 2010, 2011  Simon Josefsson
   Copyright (C) 2003, 2004, 2005 Free Software Foundation, Inc.

   This file is part of GNU Nettle.

   GNU Nettle is free software: you can redistribute it and/or
   modify it under the terms of either:

     * the GNU Lesser General Public License as published by the Free
       Software Foundation; either version 3 of the License, or (at your
       option) any later version.

   or

     * the GNU General Public License as published by the Free
       Software Foundation; either version 2 of the License, or (at your
       option) any later version.

   or both in parallel, as here.

   GNU Nettle is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   General Public License for more details.

   You should have received copies of the GNU General Public License and
   the GNU Lesser General Public License along with this program.  If
   not, see http://www.gnu.org/licenses/.
*/

/* This file is derived from cipher/serpent.c in Libgcrypt v1.4.6.
   The adaption to Nettle was made by Simon Josefsson on 2010-12-07
   with final touches on 2011-05-30.  Changes include replacing
   libgcrypt with nettle in the license template, renaming
   serpent_context to serpent_ctx, renaming u32 to uint32_t, removing
   libgcrypt stubs and selftests, modifying entry function prototypes,
   using FOR_BLOCKS to iterate through data in encrypt/decrypt, using
   LE_READ_UINT32 and LE_WRITE_UINT32 to access data in
   encrypt/decrypt, and running indent on the code. */

#if HAVE_CONFIG_H
#include "config.h"
#endif

#include <assert.h>
#include <limits.h>

#include "serpent.h"

#include "macros.h"
#include "serpent-internal.h"

/* Magic number, used during generating of the subkeys.  */
#define PHI 0x9E3779B9

/* These are the S-Boxes of Serpent.  They are copied from Serpents
   reference implementation (the optimized one, contained in
   `floppy2') and are therefore:

     Copyright (C) 1998 Ross Anderson, Eli Biham, Lars Knudsen.

  To quote the Serpent homepage
  (http://www.cl.cam.ac.uk/~rja14/serpent.html):

  "Serpent is now completely in the public domain, and we impose no
   restrictions on its use.  This was announced on the 21st August at
   the First AES Candidate Conference. The optimised implementations
   in the submission package are now under the GNU PUBLIC LICENSE
   (GPL), although some comments in the code still say otherwise. You
   are welcome to use Serpent for any application."  */
```

```
/* FIXME: Except when used within the key schedule, the inputs are not
   used after the substitution, and hence we could allow them to be
   destroyed. Can this freedom be used to optimize the sboxes? */
#define SBOX0(type, a, b, c, d, w, x, y, z)     \
  do { \
    type t02, t03, t05, t06, t07, t08, t09; \
    type t11, t12, t13, t14, t15, t17, t01; \
    t01 = b   ^ c  ; \
    t02 = a   | d  ; \
    t03 = a   ^ b  ; \
    z   = t02 ^ t01; \
    t05 = c   | z  ; \
    t06 = a   ^ d  ; \
    t07 = b   | c  ; \
    t08 = d   & t05; \
    t09 = t03 & t07; \
    y   = t09 ^ t08; \
    t11 = t09 & y  ; \
    t12 = c   ^ d  ; \
    t13 = t07 ^ t11; \
    t14 = b   & t06; \
    t15 = t06 ^ t13; \
    w   =     ~ t15; \
    t17 = w   ^ t14; \
    x   = t12 ^ t17; \
  } while (0)

#define SBOX1(type, a, b, c, d, w, x, y, z)     \
  do { \
    type t02, t03, t04, t05, t06, t07, t08; \
    type t10, t11, t12, t13, t16, t17, t01; \
    t01 = a   | d  ; \
    t02 = c   ^ d  ; \
    t03 =     ~ b  ; \
    t04 = a   ^ c  ; \
    t05 = a   | t03; \
    t06 = d   & t04; \
    t07 = t01 & t02; \
    t08 = b   | t06; \
    y   = t02 ^ t05; \
    t10 = t07 ^ t08; \
    t11 = t01 ^ t10; \
    t12 = y   ^ t11; \
    t13 = b   & d  ; \
    z   =     ~ t10; \
    x   = t13 ^ t12; \
    t16 = t10 | x  ; \
    t17 = t05 & t16; \
    w   = c   ^ t17; \
  } while (0)

#define SBOX2(type, a, b, c, d, w, x, y, z) \
  do {                                      \
    type t02, t03, t05, t06, t07, t08; \
    type t09, t10, t12, t13, t14, t01; \
    t01 = a   | c  ; \
    t02 = a   ^ b  ; \
    t03 = d   ^ t01; \
    w   = t02 ^ t03; \
    t05 = c   ^ w  ; \
    t06 = b   ^ t05; \
    t07 = b   | t05; \
    t08 = t01 & t06; \
    t09 = t03 ^ t07; \
    t10 = t02 | t09; \
    x   = t10 ^ t08; \
    t12 = a   | d  ; \
    t13 = t09 ^ x  ; \
    t14 = b   ^ t13; \
    z   =     ~ t09; \
    y   = t12 ^ t14; \
  } while (0)

#define SBOX3(type, a, b, c, d, w, x, y, z) \
  do {                                      \
    type t02, t03, t04, t05, t06, t07, t08; \
    type t09, t10, t11, t13, t14, t15, t01; \
    t01 = a   ^ c  ; \
    t02 = a   | d  ; \
    t03 = a   & d  ; \
    t04 = t01 & t02; \
    t05 = b   | t03; \
    t06 = a   & b  ; \
    t07 = d   ^ t04; \
    t08 = c   | t06; \
    t09 = b   ^ t07; \
    t10 = d   & t05; \
    t11 = t02 ^ t10; \
    z   = t08 ^ t09; \
    t13 = d   | z  ; \
    t14 = a   | t07; \
    t15 = b   & t13; \
    y   = t08 ^ t11; \
    w   = t14 ^ t15; \
    x   = t05 ^ t04; \
  } while (0)

#define SBOX4(type, a, b, c, d, w, x, y, z) \
```

```
  do { \
    type t02, t03, t04, t05, t06, t08, t09; \
    type t10, t11, t12, t13, t14, t15, t16, t01; \
    t01 = a   | b  ; \
    t02 = b   | c  ; \
    t03 = a   ^ t02; \
    t04 = b   ^ d  ; \
    t05 = d   | t03; \
    t06 = d   & t01; \
    z   = t03 ^ t06; \
    t08 = z   & t04; \
    t09 = t04 & t05; \
    t10 = c   ^ t06; \
    t11 = b   & c  ; \
    t12 = t04 ^ t08; \
    t13 = t11 | t03; \
    t14 = t10 ^ t09; \
    t15 = a   & t05; \
    t16 = t11 | t12; \
    y   = t13 ^ t08; \
    x   = t15 ^ t16; \
    w   =     ~ t14; \
  } while (0)

#define SBOX5(type, a, b, c, d, w, x, y, z) \
  do { \
    type t02, t03, t04, t05, t07, t08, t09; \
    type t10, t11, t12, t13, t14, t01; \
    t01 = b   ^ d  ; \
    t02 = b   | d  ; \
    t03 = a   & t01; \
    t04 = c   ^ t02; \
    t05 = t03 ^ t04; \
    w   =     ~ t05; \
    t07 = a   ^ t01; \
    t08 = d   | w  ; \
    t09 = b   | t05; \
    t10 = d   ^ t08; \
    t11 = b   | t07; \
    t12 = t03 | w  ; \
    t13 = t07 | t10; \
    t14 = t01 ^ t11; \
    y   = t09 ^ t13; \
    x   = t07 ^ t08; \
    z   = t12 ^ t14; \
  } while (0)

#define SBOX6(type, a, b, c, d, w, x, y, z) \
  do { \
    type t02, t03, t04, t05, t07, t08, t09, t10;        \
    type t11, t12, t13, t15, t17, t18, t01; \
    t01 = a   & d  ; \
    t02 = b   ^ c  ; \
    t03 = a   ^ d  ; \
    t04 = t01 ^ t02; \
    t05 = b   | c  ; \
    x   =     ~ t04; \
    t07 = t03 & t05; \
    t08 = b   & x  ; \
    t09 = a   | c  ; \
    t10 = t07 ^ t08; \
    t11 = b   | d  ; \
    t12 = c   ^ t11; \
    t13 = t09 ^ t10; \
    y   =     ~ t13; \
    t15 = x   & t03; \
    z   = t12 ^ t07; \
    t17 = a   ^ b  ; \
    t18 = y   ^ t15; \
    w   = t17 ^ t18; \
  } while (0)

#define SBOX7(type, a, b, c, d, w, x, y, z) \
  do { \
    type t02, t03, t04, t05, t06, t08, t09, t10;        \
    type t11, t13, t14, t15, t16, t17, t01; \
    t01 = a   & c  ; \
    t02 =     ~ d  ; \
    t03 = a   & t02; \
    t04 = b   | t01; \
    t05 = a   & b  ; \
    t06 = c   ^ t04; \
    z   = t03 ^ t06; \
    t08 = c   | z  ; \
    t09 = d   | t05; \
    t10 = a   ^ t08; \
    t11 = t04 & z  ; \
    x   = t09 ^ t10; \
    t13 = b   ^ x  ; \
    t14 = t01 ^ x  ; \
    t15 = c   ^ t05; \
    t16 = t11 | t13; \
    t17 = t02 | t14; \
    w   = t15 ^ t17; \
    y   = a   ^ t16; \
  } while (0)

/* Key schedule */
```

```
/* Note: Increments k */
#define KS_RECURRENCE(w, i, k)                                          \
  do {                                                                  \
    uint32_t _wn = (w)[(i)] ^ (w)[((i)+3)&7] ^ w[((i)+5)&7]             \
      ^ w[((i)+7)&7] ^ PHI ^ (k)++;                                     \
    ((w)[(i)] = ROTL32(11, _wn));                                       \
  } while (0)

/* Note: Increments k four times and keys once */
#define KS(keys, s, w, i, k)                                    \
  do {                                                          \
    KS_RECURRENCE(w, (i), (k));                                 \
    KS_RECURRENCE(w, (i)+1, (k));                               \
    KS_RECURRENCE(w, (i)+2, (k));                               \
    KS_RECURRENCE(w, (i)+3, (k));                               \
    SBOX##s(uint32_t, w[(i)],w[(i)+1],w[(i)+2],w[(i)+3],                \
            (*keys)[0],(*keys)[1],(*keys)[2],(*keys)[3]);       \
    (keys)++;                                                   \
  } while (0)

/* Pad user key and convert to an array of 8 uint32_t. */
static void
serpent_key_pad (const uint8_t *key, unsigned int key_length,
                 uint32_t *w)
{
  unsigned int i;

  assert (key_length <= SERPENT_MAX_KEY_SIZE);

  for (i = 0; key_length >= 4; key_length -=4, key += 4)
    w[i++] = LE_READ_UINT32(key);

  if (i < 8)
    {
      /* Key must be padded according to the Serpent specification.
         "aabbcc" -> "aabbcc0100...00" -> 0x01ccbbaa. */
      uint32_t pad = 0x01;

      while (key_length > 0)
        pad = pad << 8 | key[--key_length];

      w[i++] = pad;

      while (i < 8)
        w[i++] = 0;
    }
}

/* Initialize CONTEXT with the key KEY of LENGTH bytes.  */
void
serpent_set_key (struct serpent_ctx *ctx,
                 size_t length, const uint8_t * key)
{
  uint32_t w[8];
  uint32_t (*keys)[4];
  unsigned k;

  serpent_key_pad (key, length, w);

  /* Derive the 33 subkeys from KEY and store them in SUBKEYS. We do
     the recurrence in the key schedule using W as a circular buffer
     of just 8 uint32_t. */

  /* FIXME: Would be better to invoke SBOX with scalar variables as
     arguments, no arrays. To do that, unpack w into separate
     variables, use temporary variables as the SBOX destination. */

  keys = ctx->keys;
  k = 0;
  for (;;)
    {
      KS(keys, 3, w, 0, k);
      if (k == 132)
        break;
      KS(keys, 2, w, 4, k);
      KS(keys, 1, w, 0, k);
      KS(keys, 0, w, 4, k);
      KS(keys, 7, w, 0, k);
      KS(keys, 6, w, 4, k);
      KS(keys, 5, w, 0, k);
      KS(keys, 4, w, 4, k);
    }
  assert (keys == ctx->keys + 33);
}

void
serpent128_set_key (struct serpent_ctx *ctx, const uint8_t *key)
{
  serpent_set_key (ctx, SERPENT128_KEY_SIZE, key);
}

void
serpent192_set_key (struct serpent_ctx *ctx, const uint8_t *key)
{
  serpent_set_key (ctx, SERPENT192_KEY_SIZE, key);
}

void
```

```
serpent256_set_key (struct serpent_ctx *ctx, const uint8_t *key)
{
  serpent_set_key (ctx, SERPENT256_KEY_SIZE, key);
}
```

Notice for package(s)

iputils

```
/*
 * Copyright (c) 1989 The Regents of the University of California.
 * All rights reserved.
 *
 * This code is derived from software contributed to Berkeley by
 * Mike Muuss.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the University of
 *      California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

#ifndef lint
char copyright[] =
"@(#) Copyright (c) 1989 The Regents of the University of California.\n\
 All rights reserved.\n";
#endif /* not lint */

/*
 *                      P I N G . C
 *
 * Using the InterNet Control Message Protocol (ICMP) "ECHO" facility,
 * measure round-trip-delays and packet loss across network paths.
 *
 * Author -
 *      Mike Muuss
 *      U. S. Army Ballistic Research Laboratory
 *      December, 1983
 *
 * Status -
 *      Public Domain.  Distribution Unlimited.
 * Bugs -
 *      More statistics could always be gathered.
 *      This program has to run SUID to ROOT to access the ICMP socket.
 */

#include "ping_common.h"

#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#ifndef WITHOUT_IFADDRS
#include <ifaddrs.h>
#endif

#ifndef ICMP_FILTER
#define ICMP_FILTER     1
struct icmp_filter {
        __u32   data;
};
#endif


#define MAXIPLEN        60
#define MAXICMPLEN      76
#define NROUTES         9               /* number of record route slots */
#define TOS_MAX         255             /* 8-bit TOS field */
#define MAX_HOSTNAMELEN NI_MAXHOST
```

```
static int ts_type;
static int nroute = 0;
static __u32 route[10];



struct sockaddr_in whereto;     /* who to ping */
int optlen = 0;
int settos = 0;                 /* Set TOS, Precendence or other QOS options */
int icmp_sock;                  /* socket file descriptor */
u_char outpack[0x10000];
int maxpacket = sizeof(outpack);

static int broadcast_pings = 0;

static char *pr_addr(__u32);
static void pr_options(unsigned char * cp, int hlen);
static void pr_iph(struct iphdr *ip);
static void usage(void) __attribute__((noreturn));
static u_short in_cksum(const u_short *addr, int len, u_short salt);
static void pr_icmph(__u8 type, __u8 code, __u32 info, struct icmphdr *icp);
static int parsetos(char *str);

static struct {
        struct cmsghdr cm;
        struct in_pktinfo ipi;
} cmsg = { {sizeof(struct cmsghdr) + sizeof(struct in_pktinfo), SOL_IP, IP_PKTINFO},
           {0, }};
int cmsg_len;

struct sockaddr_in source;
char *device;
int pmtudisc = -1;


int
main(int argc, char **argv)
{
        struct hostent *hp;
        int ch, hold, packlen;
        int socket_errno;
        u_char *packet;
        char *target;
#ifdef USE_IDN
        char *hnamebuf = NULL;
#else
        char hnamebuf[MAX_HOSTNAMELEN];
#endif
        char rspace[3 + 4 * NROUTES + 1];       /* record route space */

        limit_capabilities();

#ifdef USE_IDN
        setlocale(LC_ALL, "");
#endif

        enable_capability_raw();

        icmp_sock = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);
        socket_errno = errno;

        disable_capability_raw();

        source.sin_family = AF_INET;

        preload = 1;
        while ((ch = getopt(argc, argv, COMMON_OPTSTR "bRT:")) != EOF) {
                switch(ch) {
                case 'b':
                        broadcast_pings = 1;
                        break;
                case 'Q':
                        settos = parsetos(optarg);
                        if (settos &&
                            (setsockopt(icmp_sock, IPPROTO_IP, IP_TOS,
                                        (char *)&settos, sizeof(int)) < 0)) {
                                perror("ping: error setting QOS sockopts");
                                exit(2);
                        }
                        break;
                case 'R':
                        if (options & F_TIMESTAMP) {
                                fprintf(stderr, "Only one of -T or -R may be used\n");
                                exit(2);
                        }
                        options |= F_RROUTE;
                        break;
                case 'T':
                        if (options & F_RROUTE) {
                                fprintf(stderr, "Only one of -T or -R may be used\n");
                                exit(2);
                        }
                        options |= F_TIMESTAMP;
                        if (strcmp(optarg, "tsonly") == 0)
                                ts_type = IPOPT_TS_TSONLY;
                        else if (strcmp(optarg, "tsandaddr") == 0)
                                ts_type = IPOPT_TS_TSANDADDR;
```

```
                        else if (strcmp(optarg, "tsprespec") == 0)
                                ts_type = IPOPT_TS_PRESPEC;
                        else {
                                fprintf(stderr, "Invalid timestamp type\n");
                                exit(2);
                        }
                        break;
                case 'I':
                {
#if 0
                        char dummy;
                        int i1, i2, i3, i4;

                        if (sscanf(optarg, "%u.%u.%u.%u%c",
                                   &i1, &i2, &i3, &i4, &dummy) == 4) {
                                __u8 *ptr;
                                ptr = (__u8*)&source.sin_addr;
                                ptr[0] = i1;
                                ptr[1] = i2;
                                ptr[2] = i3;
                                ptr[3] = i4;
                                options |= F_STRICTSOURCE;
                        } else {
                                device = optarg;
                        }
#else
                        if (inet_pton(AF_INET, optarg, &source.sin_addr) > 0)
                                options |= F_STRICTSOURCE;
                        else
                                device = optarg;
#endif
                        break;
                }
                case 'M':
                        if (strcmp(optarg, "do") == 0)
                                pmtudisc = IP_PMTUDISC_DO;
                        else if (strcmp(optarg, "dont") == 0)
                                pmtudisc = IP_PMTUDISC_DONT;
                        else if (strcmp(optarg, "want") == 0)
                                pmtudisc = IP_PMTUDISC_WANT;
                        else {
                                fprintf(stderr, "ping: wrong value for -M: do, dont, want are valid ones.\n");
                                exit(2);
                        }
                        break;
                case 'V':
                        printf("ping utility, iputils-%s\n", SNAPSHOT);
                        exit(0);
                COMMON_OPTIONS
                        common_options(ch);
                        break;
                default:
                        usage();
                }
        }
        argc -= optind;
        argv += optind;

        if (argc == 0)
                usage();
        if (argc > 1) {
                if (options & F_RROUTE)
                        usage();
                else if (options & F_TIMESTAMP) {
                        if (ts_type != IPOPT_TS_PRESPEC)
                                usage();
                        if (argc > 5)
                                usage();
                } else {
                        if (argc > 10)
                                usage();
                        options |= F_SOURCEROUTE;
                }
        }
        while (argc > 0) {
                target = *argv;

                memset((char *)&whereto, 0, sizeof(whereto));
                whereto.sin_family = AF_INET;
                if (inet_aton(target, &whereto.sin_addr) == 1) {
                        hostname = target;
                        if (argc == 1)
                                options |= F_NUMERIC;
                } else {
                        char *idn;
#ifdef USE_IDN
                        int rc;

                        if (hnamebuf) {
                                free(hnamebuf);
                                hnamebuf = NULL;
                        }

                        rc = idna_to_ascii_lz(target, &idn, 0);
                        if (rc != IDNA_SUCCESS) {
                                fprintf(stderr, "ping: IDN encoding failed: %s\n", idna_strerror(rc));
                                exit(2);
                        }
```

```
#else
                        idn = target;
#endif
                        hp = gethostbyname2(idn, AF_INET);
                        if (!hp) {
                                fprintf(stderr, "ping: unknown host %s\n", target);
                                exit(2);
                        }
#ifdef USE_IDN
                        free(idn);
#endif
                        memcpy(&whereto.sin_addr, hp->h_addr, 4);
#ifdef USE_IDN
                        if (idna_to_unicode_lzlz(hp->h_name, &hnamebuf, 0) != IDNA_SUCCESS) {
                                hnamebuf = strdup(hp->h_name);
                                if (!hnamebuf) {
                                        perror("ping: strdup");
                                        exit(-1);
                                }
                        }
#else
                        strncpy(hnamebuf, hp->h_name, sizeof(hnamebuf) - 1);
                        hnamebuf[sizeof(hnamebuf) - 1] = 0;
#endif
                        hostname = hnamebuf;
                }
                if (argc > 1)
                        route[nroute++] = whereto.sin_addr.s_addr;
                argc--;
                argv++;
        }

        if (source.sin_addr.s_addr == 0) {
                socklen_t alen;
                struct sockaddr_in dst = whereto;
                int probe_fd = socket(AF_INET, SOCK_DGRAM, 0);

                if (probe_fd < 0) {
                        perror("socket");
                        exit(2);
                }
                if (device) {
                        struct ifreq ifr;
                        int rc;

                        memset(&ifr, 0, sizeof(ifr));
                        strncpy(ifr.ifr_name, device, IFNAMSIZ-1);

                        enable_capability_raw();
                        rc = setsockopt(probe_fd, SOL_SOCKET, SO_BINDTODEVICE, device, strlen(device)+1);
                        disable_capability_raw();

                        if (rc == -1) {
                                if (IN_MULTICAST(ntohl(dst.sin_addr.s_addr))) {
                                        struct ip_mreqn imr;
                                        if (ioctl(probe_fd, SIOCGIFINDEX, &ifr) < 0) {
                                                fprintf(stderr, "ping: unknown iface %s\n", device);
                                                exit(2);
                                        }
                                        memset(&imr, 0, sizeof(imr));
                                        imr.imr_ifindex = ifr.ifr_ifindex;
                                        if (setsockopt(probe_fd, SOL_IP, IP_MULTICAST_IF, &imr, sizeof(imr)) == -1) {
                                                perror("ping: IP_MULTICAST_IF");
                                                exit(2);
                                        }
                                } else {
                                        perror("ping: SO_BINDTODEVICE");
                                        exit(2);
                                }
                        }
                }

                if (settos &&
                    setsockopt(probe_fd, IPPROTO_IP, IP_TOS, (char *)&settos, sizeof(int)) < 0)
                        perror("Warning: error setting QOS sockopts");

                dst.sin_port = htons(1025);
                if (nroute)
                        dst.sin_addr.s_addr = route[0];
                if (connect(probe_fd, (struct sockaddr*)&dst, sizeof(dst)) == -1) {
                        if (errno == EACCES) {
                                if (broadcast_pings == 0) {
                                        fprintf(stderr, "Do you want to ping broadcast? Then -b\n");
                                        exit(2);
                                }
                                fprintf(stderr, "WARNING: pinging broadcast address\n");
                                if (setsockopt(probe_fd, SOL_SOCKET, SO_BROADCAST,
                                               &broadcast_pings, sizeof(broadcast_pings)) < 0) {
                                        perror ("can't set broadcasting");
                                        exit(2);
                                }
                                if (connect(probe_fd, (struct sockaddr*)&dst, sizeof(dst)) == -1) {
                                        perror("connect");
                                        exit(2);
                                }
                        } else {
                                perror("connect");
                                exit(2);
```

```
                        }
                }
                alen = sizeof(source);
                if (getsockname(probe_fd, (struct sockaddr*)&source, &alen) == -1) {
                        perror("getsockname");
                        exit(2);
                }
                source.sin_port = 0;

#ifndef WITHOUT_IFADDRS
                if (device) {
                        struct ifaddrs *ifa0, *ifa;
                        int ret;

                        ret = getifaddrs(&ifa0);
                        if (ret) {
                                fprintf(stderr, "gatifaddrs() failed.\n");
                                exit(2);
                        }
                        for (ifa = ifa0; ifa; ifa = ifa->ifa_next) {
                                if (!ifa->ifa_addr || ifa->ifa_addr->sa_family != AF_INET)
                                        continue;
                                if (!strncmp(ifa->ifa_name, device, sizeof(device) - 1) &&
                                    !memcmp(&((struct sockaddr_in *)ifa->ifa_addr)->sin_addr,
                                            &source.sin_addr, sizeof(source.sin_addr)))
                                        break;
                        }
                        freeifaddrs(ifa0);
                        if (!ifa)
                                fprintf(stderr, "ping: Warning: source address might be selected on device other than %s.\n", device);
                }
#endif
                close(probe_fd);
        } while (0);

        if (whereto.sin_addr.s_addr == 0)
                whereto.sin_addr.s_addr = source.sin_addr.s_addr;

        if (icmp_sock < 0) {
                errno = socket_errno;
                perror("ping: icmp open socket");
                exit(2);
        }

        if (device) {
                struct ifreq ifr;

                memset(&ifr, 0, sizeof(ifr));
                strncpy(ifr.ifr_name, device, IFNAMSIZ-1);
                if (ioctl(icmp_sock, SIOCGIFINDEX, &ifr) < 0) {
                        fprintf(stderr, "ping: unknown iface %s\n", device);
                        exit(2);
                }
                cmsg.ipi.ipi_ifindex = ifr.ifr_ifindex;
                cmsg_len = sizeof(cmsg);
        }

        if (broadcast_pings || IN_MULTICAST(ntohl(whereto.sin_addr.s_addr))) {
                if (uid) {
                        if (interval < 1000) {
                                fprintf(stderr, "ping: broadcast ping with too short interval.\n");
                                exit(2);
                        }
                        if (pmtudisc >= 0 && pmtudisc != IP_PMTUDISC_DO) {
                                fprintf(stderr, "ping: broadcast ping does not fragment.\n");
                                exit(2);
                        }
                }
                if (pmtudisc < 0)
                        pmtudisc = IP_PMTUDISC_DO;
        }

        if (pmtudisc >= 0) {
                if (setsockopt(icmp_sock, SOL_IP, IP_MTU_DISCOVER, &pmtudisc, sizeof(pmtudisc)) == -1) {
                        perror("ping: IP_MTU_DISCOVER");
                        exit(2);
                }
        }

        if ((options&F_STRICTSOURCE) &&
            bind(icmp_sock, (struct sockaddr*)&source, sizeof(source)) == -1) {
                perror("bind");
                exit(2);
        }

        if (1) {
                struct icmp_filter filt;
                filt.data = ~((1<<ICMP_SOURCE_QUENCH)|
                              (1<<ICMP_DEST_UNREACH)|
                              (1<<ICMP_TIME_EXCEEDED)|
                              (1<<ICMP_PARAMETERPROB)|
                              (1<<ICMP_REDIRECT)|
                              (1<<ICMP_ECHOREPLY));
                if (setsockopt(icmp_sock, SOL_RAW, ICMP_FILTER, (char*)&filt, sizeof(filt)) == -1)
                        perror("WARNING: setsockopt(ICMP_FILTER)");
        }

        hold = 1;
```

```
if (setsockopt(icmp_sock, SOL_IP, IP_RECVERR, (char *)&hold, sizeof(hold)))
        fprintf(stderr, "WARNING: your kernel is veeery old. No problems.\n");

/* record route option */
if (options & F_RROUTE) {
        memset(rspace, 0, sizeof(rspace));
        rspace[0] = IPOPT_NOP;
        rspace[1+IPOPT_OPTVAL] = IPOPT_RR;
        rspace[1+IPOPT_OLEN] = sizeof(rspace)-1;
        rspace[1+IPOPT_OFFSET] = IPOPT_MINOFF;
        optlen = 40;
        if (setsockopt(icmp_sock, IPPROTO_IP, IP_OPTIONS, rspace, sizeof(rspace)) < 0) {
                perror("ping: record route");
                exit(2);
        }
}
if (options & F_TIMESTAMP) {
        memset(rspace, 0, sizeof(rspace));
        rspace[0] = IPOPT_TIMESTAMP;
        rspace[1] = (ts_type==IPOPT_TS_TSONLY ? 40 : 36);
        rspace[2] = 5;
        rspace[3] = ts_type;
        if (ts_type == IPOPT_TS_PRESPEC) {
                int i;
                rspace[1] = 4+nroute*8;
                for (i=0; i<nroute; i++)
                        *(__u32*)&rspace[4+i*8] = route[i];
        }
        if (setsockopt(icmp_sock, IPPROTO_IP, IP_OPTIONS, rspace, rspace[1]) < 0) {
                rspace[3] = 2;
                if (setsockopt(icmp_sock, IPPROTO_IP, IP_OPTIONS, rspace, rspace[1]) < 0) {
                        perror("ping: ts option");
                        exit(2);
                }
        }
        optlen = 40;
}
if (options & F_SOURCEROUTE) {
        int i;
        memset(rspace, 0, sizeof(rspace));
        rspace[0] = IPOPT_NOOP;
        rspace[1+IPOPT_OPTVAL] = (options & F_SO_DONTROUTE) ? IPOPT_SSRR
                : IPOPT_LSRR;
        rspace[1+IPOPT_OLEN] = 3 + nroute*4;
        rspace[1+IPOPT_OFFSET] = IPOPT_MINOFF;
        for (i=0; i<nroute; i++)
                *(__u32*)&rspace[4+i*4] = route[i];

        if (setsockopt(icmp_sock, IPPROTO_IP, IP_OPTIONS, rspace, 4 + nroute*4) < 0) {
                perror("ping: record route");
                exit(2);
        }
        optlen = 40;
}

/* Estimate memory eaten by single packet. It is rough estimate.
 * Actually, for small datalen's it depends on kernel side a lot. */
hold = datalen + 8;
hold += ((hold+511)/512)*(optlen + 20 + 16 + 64 + 160);
sock_setbufs(icmp_sock, hold);

if (broadcast_pings) {
        if (setsockopt(icmp_sock, SOL_SOCKET, SO_BROADCAST,
                       &broadcast_pings, sizeof(broadcast_pings)) < 0) {
                perror ("ping: can't set broadcasting");
                exit(2);
        }
}

if (options & F_NOLOOP) {
        int loop = 0;
        if (setsockopt(icmp_sock, IPPROTO_IP, IP_MULTICAST_LOOP,
                                              &loop, 1) == -1) {
                perror ("ping: can't disable multicast loopback");
                exit(2);
        }
}
if (options & F_TTL) {
        int ittl = ttl;
        if (setsockopt(icmp_sock, IPPROTO_IP, IP_MULTICAST_TTL,
                                              &ttl, 1) == -1) {
                perror ("ping: can't set multicast time-to-live");
                exit(2);
        }
        if (setsockopt(icmp_sock, IPPROTO_IP, IP_TTL,
                                              &ittl, sizeof(ittl)) == -1) {
                perror ("ping: can't set unicast time-to-live");
                exit(2);
        }
}

if (datalen > 0xFFFF - 8 - optlen - 20) {
        if (uid || datalen > sizeof(outpack)-8) {
                fprintf(stderr, "Error: packet size %d is too large. Maximum is %d\n", datalen, 0xFFFF-8-20-optlen);
                exit(2);
        }
        /* Allow small oversize to root yet. It will cause EMSGSIZE. */
        fprintf(stderr, "WARNING: packet size %d is too large. Maximum is %d\n", datalen, 0xFFFF-8-20-optlen);
```

```
        }

        if (datalen >= sizeof(struct timeval))  /* can we time transfer */
                timing = 1;
        packlen = datalen + MAXIPLEN + MAXICMPLEN;
        if (!(packet = (u_char *)malloc((u_int)packlen))) {
                fprintf(stderr, "ping: out of memory.\n");
                exit(2);
        }

        printf("PING %s (%s) ", hostname, inet_ntoa(whereto.sin_addr));
        if (device || (options&F_STRICTSOURCE))
                printf("from %s %s: ", inet_ntoa(source.sin_addr), device ?: "");
        printf("%d(%d) bytes of data.\n", datalen, datalen+8+optlen+20);

        setup(icmp_sock);

        main_loop(icmp_sock, packet, packlen);
}


int receive_error_msg()
{
        int res;
        char cbuf[512];
        struct iovec  iov;
        struct msghdr msg;
        struct cmsghdr *cmsg;
        struct sock_extended_err *e;
        struct icmphdr icmph;
        struct sockaddr_in target;
        int net_errors = 0;
        int local_errors = 0;
        int saved_errno = errno;

        iov.iov_base = &icmph;
        iov.iov_len = sizeof(icmph);
        msg.msg_name = (void*)⏚
        msg.msg_namelen = sizeof(target);
        msg.msg_iov = &iov;
        msg.msg_iovlen = 1;
        msg.msg_flags = 0;
        msg.msg_control = cbuf;
        msg.msg_controllen = sizeof(cbuf);

        res = recvmsg(icmp_sock, &msg, MSG_ERRQUEUE|MSG_DONTWAIT);
        if (res < 0)
                goto out;

        e = NULL;
        for (cmsg = CMSG_FIRSTHDR(&msg); cmsg; cmsg = CMSG_NXTHDR(&msg, cmsg)) {
                if (cmsg->cmsg_level == SOL_IP) {
                        if (cmsg->cmsg_type == IP_RECVERR)
                                e = (struct sock_extended_err *)CMSG_DATA(cmsg);
                }
        }
        if (e == NULL)
                abort();

        if (e->ee_origin == SO_EE_ORIGIN_LOCAL) {
                local_errors++;
                if (options & F_QUIET)
                        goto out;
                if (options & F_FLOOD)
                        write_stdout("E", 1);
                else if (e->ee_errno != EMSGSIZE)
                        fprintf(stderr, "ping: local error: %s\n", strerror(e->ee_errno));
                else
                        fprintf(stderr, "ping: local error: Message too long, mtu=%u\n", e->ee_info);
                nerrors++;
        } else if (e->ee_origin == SO_EE_ORIGIN_ICMP) {
                struct sockaddr_in *sin = (struct sockaddr_in*)(e+1);

                if (res < sizeof(icmph) ||
                    target.sin_addr.s_addr != whereto.sin_addr.s_addr ||
                    icmph.type != ICMP_ECHO ||
                    icmph.un.echo.id != ident) {
                        /* Not our error, not an error at all. Clear. */
                        saved_errno = 0;
                        goto out;
                }

                acknowledge(ntohs(icmph.un.echo.sequence));

                if (!working_recverr) {
                        struct icmp_filter filt;
                        working_recverr = 1;
                        /* OK, it works. Add stronger filter. */
                        filt.data = ~((1<<ICMP_SOURCE_QUENCH)|
                                      (1<<ICMP_REDIRECT)|
                                      (1<<ICMP_ECHOREPLY));
                        if (setsockopt(icmp_sock, SOL_RAW, ICMP_FILTER, (char*)&filt, sizeof(filt)) == -1)
                                perror("\rWARNING: setsockopt(ICMP_FILTER)");
                }

                net_errors++;
                nerrors++;
```

```
                if (options & F_QUIET)
                        goto out;
                if (options & F_FLOOD) {
                        write_stdout("\bE", 2);
                } else {
                        print_timestamp();
                        printf("From %s icmp_seq=%u ", pr_addr(sin->sin_addr.s_addr), ntohs(icmph.un.echo.sequence));
                        pr_icmph(e->ee_type, e->ee_code, e->ee_info, NULL);
                        fflush(stdout);
                }
        }

out:
        errno = saved_errno;
        return net_errors ? : -local_errors;
}

/*
 * pinger --
 *      Compose and transmit an ICMP ECHO REQUEST packet.  The IP packet
 * will be added on by the kernel.  The ID field is our UNIX process ID,
 * and the sequence number is an ascending integer.  The first 8 bytes
 * of the data portion are used to hold a UNIX "timeval" struct in VAX
 * byte-order, to compute the round-trip time.
 */
int send_probe()
{
        struct icmphdr *icp;
        int cc;
        int i;

        icp = (struct icmphdr *)outpack;
        icp->type = ICMP_ECHO;
        icp->code = 0;
        icp->checksum = 0;
        icp->un.echo.sequence = htons(ntransmitted+1);
        icp->un.echo.id = ident;                        /* ID */

        rcvd_clear(ntransmitted+1);

        if (timing) {
                if (options&F_LATENCY) {
                        struct timeval tmp_tv;
                        gettimeofday(&tmp_tv, NULL);
                        memcpy(icp+1, &tmp_tv, sizeof(tmp_tv));
                } else {
                        memset(icp+1, 0, sizeof(struct timeval));
                }
        }

        cc = datalen + 8;                       /* skips ICMP portion */

        /* compute ICMP checksum here */
        icp->checksum = in_cksum((u_short *)icp, cc, 0);

        if (timing && !(options&F_LATENCY)) {
                struct timeval tmp_tv;
                gettimeofday(&tmp_tv, NULL);
                memcpy(icp+1, &tmp_tv, sizeof(tmp_tv));
                icp->checksum = in_cksum((u_short *)&tmp_tv, sizeof(tmp_tv), ~icp->checksum);
        }

        do {
                static struct iovec iov = {outpack, 0};
                static struct msghdr m = { &whereto, sizeof(whereto),
                                                   &iov, 1, &cmsg, 0, 0 };
                m.msg_controllen = cmsg_len;
                iov.iov_len = cc;

                i = sendmsg(icmp_sock, &m, confirm);
                confirm = 0;
        } while (0);

        return (cc == i ? 0 : i);
}

/*
 * parse_reply --
 *      Print out the packet, if it came from us.  This logic is necessary
 * because ALL readers of the ICMP socket get a copy of ALL ICMP packets
 * which arrive ('tis only fair).  This permits multiple copies of this
 * program to be run without having intermingled output (or statistics!).
 */
void pr_echo_reply(__u8 *_icp, int len)
{
        struct icmphdr *icp = (struct icmphdr *)_icp;
        printf(" icmp_seq=%u", ntohs(icp->un.echo.sequence));
}

int
parse_reply(struct msghdr *msg, int cc, void *addr, struct timeval *tv)
{
        struct sockaddr_in *from = addr;
        __u8 *buf = msg->msg_iov->iov_base;
        struct icmphdr *icp;
        struct iphdr *ip;
        int hlen;
        int csfailed;
```

```
/* Check the IP header */
ip = (struct iphdr *)buf;
hlen = ip->ihl*4;
if (cc < hlen + 8 || ip->ihl < 5) {
        if (options & F_VERBOSE)
                fprintf(stderr, "ping: packet too short (%d bytes) from %s\n", cc,
                        pr_addr(from->sin_addr.s_addr));
        return 1;
}

/* Now the ICMP part */
cc -= hlen;
icp = (struct icmphdr *)(buf + hlen);
csfailed = in_cksum((u_short *)icp, cc, 0);

if (icp->type == ICMP_ECHOREPLY) {
        if (icp->un.echo.id != ident)
                return 1;                       /* 'Twas not our ECHO */
        if (gather_statistics((__u8*)icp, sizeof(*icp), cc,
                              ntohs(icp->un.echo.sequence),
                              ip->ttl, 0, tv, pr_addr(from->sin_addr.s_addr),
                              pr_echo_reply))
                return 0;
} else {
        /* We fall here when a redirect or source quench arrived.
         * Also this branch processes icmp errors, when IP_RECVERR
         * is broken. */

        switch (icp->type) {
        case ICMP_ECHO:
                /* MUST NOT */
                return 1;
        case ICMP_SOURCE_QUENCH:
        case ICMP_REDIRECT:
        case ICMP_DEST_UNREACH:
        case ICMP_TIME_EXCEEDED:
        case ICMP_PARAMETERPROB:
                {
                        struct iphdr * iph = (struct  iphdr *)(&icp[1]);
                        struct icmphdr *icp1 = (struct icmphdr*)((unsigned char *)iph + iph->ihl*4);
                        int error_pkt;
                        if (cc < 8+sizeof(struct iphdr)+8 ||
                            cc < 8+iph->ihl*4+8)
                                return 1;
                        if (icp1->type != ICMP_ECHO ||
                            iph->daddr != whereto.sin_addr.s_addr ||
                            icp1->un.echo.id != ident)
                                return 1;
                        error_pkt = (icp->type != ICMP_REDIRECT &&
                                     icp->type != ICMP_SOURCE_QUENCH);
                        if (error_pkt) {
                                acknowledge(ntohs(icp1->un.echo.sequence));
                                if (working_recverr) {
                                        return 0;
                                } else {
                                        static int once;
                                        /* Sigh, IP_RECVERR for raw socket
                                         * was broken until 2.4.9. So, we ignore
                                         * the first error and warn on the second.
                                         */
                                        if (once++ == 1)
                                                fprintf(stderr, "\rWARNING: kernel is not very fresh, upgrade is recommended.\n");
                                        if (once == 1)
                                                return 0;
                                }
                        }
                        nerrors+=error_pkt;
                        if (options&F_QUIET)
                                return !error_pkt;
                        if (options & F_FLOOD) {
                                if (error_pkt)
                                        write_stdout("\bE", 2);
                                return !error_pkt;
                        }
                        print_timestamp();
                        printf("From %s: icmp_seq=%u ",
                               pr_addr(from->sin_addr.s_addr),
                               ntohs(icp1->un.echo.sequence));
                        if (csfailed)
                                printf("(BAD CHECKSUM)");
                        pr_icmph(icp->type, icp->code, ntohl(icp->un.gateway), icp);
                        return !error_pkt;
                }
        default:
                /* MUST NOT */
                break;
        }
        if ((options & F_FLOOD) && !(options & (F_VERBOSE|F_QUIET))) {
                if (!csfailed)
                        write_stdout("!E", 2);
                else
                        write_stdout("!EC", 3);
                return 0;
        }
        if (!(options & F_VERBOSE) || uid)
                return 0;
        if (options & F_PTIMEOFDAY) {
```

```
                        struct timeval recv_time;
                        gettimeofday(&recv_time, NULL);
                        printf("%lu.%06lu ", (unsigned long)recv_time.tv_sec, (unsigned long)recv_time.tv_usec);
                }
                printf("From %s: ", pr_addr(from->sin_addr.s_addr));
                if (csfailed) {
                        printf("(BAD CHECKSUM)\n");
                        return 0;
                }
                pr_icmph(icp->type, icp->code, ntohl(icp->un.gateway), icp);
                return 0;
        }

        if (!(options & F_FLOOD)) {
                pr_options(buf + sizeof(struct iphdr), hlen);

                if (options & F_AUDIBLE)
                        putchar('\a');
                putchar('\n');
                fflush(stdout);
        } else {
                putchar('\a');
                fflush(stdout);
        }
        return 0;
}


#if BYTE_ORDER == LITTLE_ENDIAN
# define ODDBYTE(v)     (v)
#elif BYTE_ORDER == BIG_ENDIAN
# define ODDBYTE(v)     ((u_short)(v) << 8)
#else
# define ODDBYTE(v)     htons((u_short)(v) << 8)
#endif

u_short
in_cksum(const u_short *addr, register int len, u_short csum)
{
        register int nleft = len;
        const u_short *w = addr;
        register u_short answer;
        register int sum = csum;

        /*
         *  Our algorithm is simple, using a 32 bit accumulator (sum),
         *  we add sequential 16 bit words to it, and at the end, fold
         *  back all the carry bits from the top 16 bits into the lower
         *  16 bits.
         */
        while (nleft > 1)  {
                sum += *w++;
                nleft -= 2;
        }

        /* mop up an odd byte, if necessary */
        if (nleft == 1)
                sum += ODDBYTE(*(u_char *)w); /* le16toh() may be unavailable on old systems */

        /*
         * add back carry outs from top 16 bits to low 16 bits
         */
        sum = (sum >> 16) + (sum & 0xffff);     /* add hi 16 to low 16 */
        sum += (sum >> 16);                     /* add carry */
        answer = ~sum;                          /* truncate to 16 bits */
        return (answer);
}

/*
 * pr_icmph --
 *      Print a descriptive string about an ICMP header.
 */
void pr_icmph(__u8 type, __u8 code, __u32 info, struct icmphdr *icp)
{
        switch(type) {
        case ICMP_ECHOREPLY:
                printf("Echo Reply\n");
                /* XXX ID + Seq + Data */
                break;
        case ICMP_DEST_UNREACH:
                switch(code) {
                case ICMP_NET_UNREACH:
                        printf("Destination Net Unreachable\n");
                        break;
                case ICMP_HOST_UNREACH:
                        printf("Destination Host Unreachable\n");
                        break;
                case ICMP_PROT_UNREACH:
                        printf("Destination Protocol Unreachable\n");
                        break;
                case ICMP_PORT_UNREACH:
                        printf("Destination Port Unreachable\n");
                        break;
                case ICMP_FRAG_NEEDED:
                        printf("Frag needed and DF set (mtu = %u)\n", info);
                        break;
                case ICMP_SR_FAILED:
                        printf("Source Route Failed\n");
```

```
                break;
        case ICMP_NET_UNKNOWN:
                printf("Destination Net Unknown\n");
                break;
        case ICMP_HOST_UNKNOWN:
                printf("Destination Host Unknown\n");
                break;
        case ICMP_HOST_ISOLATED:
                printf("Source Host Isolated\n");
                break;
        case ICMP_NET_ANO:
                printf("Destination Net Prohibited\n");
                break;
        case ICMP_HOST_ANO:
                printf("Destination Host Prohibited\n");
                break;
        case ICMP_NET_UNR_TOS:
                printf("Destination Net Unreachable for Type of Service\n");
                break;
        case ICMP_HOST_UNR_TOS:
                printf("Destination Host Unreachable for Type of Service\n");
                break;
        case ICMP_PKT_FILTERED:
                printf("Packet filtered\n");
                break;
        case ICMP_PREC_VIOLATION:
                printf("Precedence Violation\n");
                break;
        case ICMP_PREC_CUTOFF:
                printf("Precedence Cutoff\n");
                break;
        default:
                printf("Dest Unreachable, Bad Code: %d\n", code);
                break;
        }
        if (icp && (options & F_VERBOSE))
                pr_iph((struct iphdr*)(icp + 1));
        break;
case ICMP_SOURCE_QUENCH:
        printf("Source Quench\n");
        if (icp && (options & F_VERBOSE))
                pr_iph((struct iphdr*)(icp + 1));
        break;
case ICMP_REDIRECT:
        switch(code) {
        case ICMP_REDIR_NET:
                printf("Redirect Network");
                break;
        case ICMP_REDIR_HOST:
                printf("Redirect Host");
                break;
        case ICMP_REDIR_NETTOS:
                printf("Redirect Type of Service and Network");
                break;
        case ICMP_REDIR_HOSTTOS:
                printf("Redirect Type of Service and Host");
                break;
        default:
                printf("Redirect, Bad Code: %d", code);
                break;
        }
        if (icp)
                printf("(New nexthop: %s)\n", pr_addr(icp->un.gateway));
        if (icp && (options & F_VERBOSE))
                pr_iph((struct iphdr*)(icp + 1));
        break;
case ICMP_ECHO:
        printf("Echo Request\n");
        /* XXX ID + Seq + Data */
        break;
case ICMP_TIME_EXCEEDED:
        switch(code) {
        case ICMP_EXC_TTL:
                printf("Time to live exceeded\n");
                break;
        case ICMP_EXC_FRAGTIME:
                printf("Frag reassembly time exceeded\n");
                break;
        default:
                printf("Time exceeded, Bad Code: %d\n", code);
                break;
        }
        if (icp && (options & F_VERBOSE))
                pr_iph((struct iphdr*)(icp + 1));
        break;
case ICMP_PARAMETERPROB:
        printf("Parameter problem: pointer = %u\n", icp ? (ntohl(icp->un.gateway)>>24) : info);
        if (icp && (options & F_VERBOSE))
                pr_iph((struct iphdr*)(icp + 1));
        break;
case ICMP_TIMESTAMP:
        printf("Timestamp\n");
        /* XXX ID + Seq + 3 timestamps */
        break;
case ICMP_TIMESTAMPREPLY:
        printf("Timestamp Reply\n");
        /* XXX ID + Seq + 3 timestamps */
        break;
```

```
        case ICMP_INFO_REQUEST:
                printf("Information Request\n");
                /* XXX ID + Seq */
                break;
        case ICMP_INFO_REPLY:
                printf("Information Reply\n");
                /* XXX ID + Seq */
                break;
#ifdef ICMP_MASKREQ
        case ICMP_MASKREQ:
                printf("Address Mask Request\n");
                break;
#endif
#ifdef ICMP_MASKREPLY
        case ICMP_MASKREPLY:
                printf("Address Mask Reply\n");
                break;
#endif
        default:
                printf("Bad ICMP type: %d\n", type);
        }
}

void pr_options(unsigned char * cp, int hlen)
{
        int i, j;
        int optlen, totlen;
        unsigned char * optptr;
        static int old_rrlen;
        static char old_rr[MAX_IPOPTLEN];

        totlen = hlen-sizeof(struct iphdr);
        optptr = cp;

        while (totlen > 0) {
                if (*optptr == IPOPT_EOL)
                        break;
                if (*optptr == IPOPT_NOP) {
                        totlen--;
                        optptr++;
                        printf("\nNOP");
                        continue;
                }
                cp = optptr;
                optlen = optptr[1];
                if (optlen < 2 || optlen > totlen)
                        break;

                switch (*cp) {
                case IPOPT_SSRR:
                case IPOPT_LSRR:
                        printf("\n%cSRR: ", *cp==IPOPT_SSRR ? 'S' : 'L');
                        j = *++cp;
                        i = *++cp;
                        i -= 4;
                        cp++;
                        if (j > IPOPT_MINOFF) {
                                for (;;) {
                                        __u32 address;
                                        memcpy(&address, cp, 4);
                                        cp += 4;
                                        if (address == 0)
                                                printf("\t0.0.0.0");
                                        else
                                                printf("\t%s", pr_addr(address));
                                        j -= 4;
                                        putchar('\n');
                                        if (j <= IPOPT_MINOFF)
                                                break;
                                }
                        }
                        break;
                case IPOPT_RR:
                        j = *++cp;              /* get length */
                        i = *++cp;              /* and pointer */
                        if (i > j)
                                i = j;
                        i -= IPOPT_MINOFF;
                        if (i <= 0)
                                break;
                        if (i == old_rrlen
                            && !memcmp(cp, old_rr, i)
                            && !(options & F_FLOOD)) {
                                printf("\t(same route)");
                                i = ((i + 3) / 4) * 4;
                                cp += i;
                                break;
                        }
                        old_rrlen = i;
                        memcpy(old_rr, (char *)cp, i);
                        printf("\nRR: ");
                        cp++;
                        for (;;) {
                                __u32 address;
                                memcpy(&address, cp, 4);
                                cp += 4;
                                if (address == 0)
                                        printf("\t0.0.0.0");
```

```
                                else
                                        printf("\t%s", pr_addr(address));
                                i -= 4;
                                putchar('\n');
                                if (i <= 0)
                                        break;
                        }
                        break;
                case IPOPT_TS:
                {
                        int stdtime = 0, nonstdtime = 0;
                        __u8 flags;
                        j = *++cp;              /* get length */
                        i = *++cp;              /* and pointer */
                        if (i > j)
                                i = j;
                        i -= 5;
                        if (i <= 0)
                                break;
                        flags = *++cp;
                        printf("\nTS: ");
                        cp++;
                        for (;;) {
                                long l;

                                if ((flags&0xF) != IPOPT_TS_TSONLY) {
                                        __u32 address;
                                        memcpy(&address, cp, 4);
                                        cp += 4;
                                        if (address == 0)
                                                printf("\t0.0.0.0");
                                        else
                                                printf("\t%s", pr_addr(address));
                                        i -= 4;
                                        if (i <= 0)
                                                break;
                                }
                                l = *cp++;
                                l = (l<<8) + *cp++;
                                l = (l<<8) + *cp++;
                                l = (l<<8) + *cp++;

                                if  (l & 0x80000000) {
                                        if (nonstdtime==0)
                                                printf("\t%ld absolute not-standard", l&0x7fffffff);
                                        else
                                                printf("\t%ld not-standard", (l&0x7fffffff) - nonstdtime);
                                        nonstdtime = l&0x7fffffff;
                                } else {
                                        if (stdtime==0)
                                                printf("\t%ld absolute", l);
                                        else
                                                printf("\t%ld", l - stdtime);
                                        stdtime = l;
                                }
                                i -= 4;
                                putchar('\n');
                                if (i <= 0)
                                        break;
                        }
                        if (flags>>4)
                                printf("Unrecorded hops: %d\n", flags>>4);
                        break;
                }
                default:
                        printf("\nunknown option %x", *cp);
                        break;
                }
                totlen -= optlen;
                optptr += optlen;
        }
}


/*
 * pr_iph --
 *      Print an IP header with options.
 */
void pr_iph(struct iphdr *ip)
{
        int hlen;
        u_char *cp;

        hlen = ip->ihl << 2;
        cp = (u_char *)ip + 20;         /* point to options */

        printf("Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst Data\n");
        printf(" %1x  %1x  %02x %04x %04x",
               ip->version, ip->ihl, ip->tos, ip->tot_len, ip->id);
        printf("   %1x %04x", ((ip->frag_off) & 0xe000) >> 13,
               (ip->frag_off) & 0x1fff);
        printf("  %02x  %02x %04x", ip->ttl, ip->protocol, ip->check);
        printf(" %s ", inet_ntoa(*(struct in_addr *)&ip->saddr));
        printf(" %s ", inet_ntoa(*(struct in_addr *)&ip->daddr));
        printf("\n");
        pr_options(cp, hlen);
}
```

```
/*
 * pr_addr --
 *      Return an ascii host address as a dotted quad and optionally with
 * a hostname.
 */
char *
pr_addr(__u32 addr)
{
        struct hostent *hp;
        static char buf[4096];

        in_pr_addr = !setjmp(pr_addr_jmp);

        if (exiting || (options & F_NUMERIC) ||
            !(hp = gethostbyaddr((char *)&addr, 4, AF_INET)))
                sprintf(buf, "%s", inet_ntoa(*(struct in_addr *)&addr));
        else {
                char *s;
#if USE_IDN
                if (idna_to_unicode_lzlz(hp->h_name, &s, 0) != IDNA_SUCCESS)
                        s = NULL;
#else
                s = NULL;
#endif
                snprintf(buf, sizeof(buf), "%s (%s)", s ? s : hp->h_name,
                         inet_ntoa(*(struct in_addr *)&addr));
#if USE_IDN
                free(s);
#endif
        }

        in_pr_addr = 0;

        return(buf);
}


/* Set Type of Service (TOS) and other Quality of Service relating bits */
int parsetos(char *str)
{
        const char *cp;
        int tos;
        char *ep;

        /* handle both hex and decimal values */
        if (str[0] == '0' && (str[1] == 'x' || str[1] == 'X')) {
                cp = str + 2;
                tos = (int)strtol(cp, &ep, 16);
        } else
                tos = (int)strtol(str, &ep, 10);

        /* doesn't look like decimal or hex, eh? */
        if (*ep != '\0') {
                fprintf(stderr, "ping: \"%s\" bad value for TOS\n", str);
                exit(2);
        }

        if (tos > TOS_MAX) {
                fprintf(stderr, "ping: the decimal value of TOS bits must be 0-254 (or zero)\n");
                exit(2);
        }
        return(tos);
}

#include <linux/filter.h>

void install_filter(void)
{
        static int once;
        static struct sock_filter insns[] = {
                BPF_STMT(BPF_LDX|BPF_B|BPF_MSH, 0), /* Skip IP header. F..g BSD... Look into ping6. */
                BPF_STMT(BPF_LD|BPF_H|BPF_IND, 4), /* Load icmp echo ident */
                BPF_JUMP(BPF_JMP|BPF_JEQ|BPF_K, 0xAAAA, 0, 1), /* Ours? */
                BPF_STMT(BPF_RET|BPF_K, ~0U), /* Yes, it passes. */
                BPF_STMT(BPF_LD|BPF_B|BPF_IND, 0), /* Load icmp type */
                BPF_JUMP(BPF_JMP|BPF_JEQ|BPF_K, ICMP_ECHOREPLY, 1, 0), /* Echo? */
                BPF_STMT(BPF_RET|BPF_K, 0xFFFFFFF), /* No. It passes. */
                BPF_STMT(BPF_RET|BPF_K, 0) /* Echo with wrong ident. Reject. */
        };
        static struct sock_fprog filter = {
                sizeof insns / sizeof(insns[0]),
                insns
        };

        if (once)
                return;
        once = 1;

        /* Patch bpflet for current identifier. */
        insns[2] = (struct sock_filter)BPF_JUMP(BPF_JMP|BPF_JEQ|BPF_K, htons(ident), 0, 1);

        if (setsockopt(icmp_sock, SOL_SOCKET, SO_ATTACH_FILTER, &filter, sizeof(filter)))
                perror("WARNING: failed to install socket filter\n");
}

#define USAGE_NEWLINE   "\n           "

void usage(void)
```

```
{
        fprintf(stderr,
                "Usage: ping"
                " [-"
                        "aAbBdDfhLnOqrRUvV"
                "]"
                " [-c count]"
                " [-i interval]"
                " [-I interface]"
                USAGE_NEWLINE
                " [-m mark]"
                " [-M pmtudisc_option]"
                " [-l preload]"
                " [-p pattern]"
                " [-Q tos]"
                USAGE_NEWLINE
                " [-s packetsize]"
                " [-S sndbuf]"
                " [-t ttl]"
                " [-T timestamp_option]"
                USAGE_NEWLINE
                " [-w deadline]"
                " [-W timeout]"
                " [hop1 ...] destination"
                "\n"
        );
        exit(2);
}
```

Notice for package(s)

iputils

```
/*
 * tracepath.c
 *
 *              This program is free software; you can redistribute it and/or
 *              modify it under the terms of the GNU General Public License
 *              as published by the Free Software Foundation; either version
 *              2 of the License, or (at your option) any later version.
 *
 * Authors:     Alexey Kuznetsov, <kuznet@ms2.inr.ac.ru>
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <linux/types.h>
#include <linux/errqueue.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <resolv.h>
#include <sys/time.h>
#include <sys/uio.h>
#include <arpa/inet.h>
#ifdef USE_IDN
#include <idna.h>
#include <locale.h>
#endif

#ifndef IP_PMTUDISC_PROBE
#define IP_PMTUDISC_PROBE       3
#endif

struct hhistory
{
        int     hops;
        struct timeval sendtime;
};

struct hhistory his[64];
int hisptr;

struct sockaddr_in target;
__u16 base_port;

const int overhead = 28;
int mtu = 65535;
void *pktbuf;
int hops_to = -1;
int hops_from = -1;
int no_resolve = 0;
int show_both = 0;

#define HOST_COLUMN_SIZE        52

struct probehdr
{
```

```
        __u32 ttl;
        struct timeval tv;
};

void data_wait(int fd)
{
        fd_set fds;
        struct timeval tv;
        FD_ZERO(&fds);
        FD_SET(fd, &fds);
        tv.tv_sec = 1;
        tv.tv_usec = 0;
        select(fd+1, &fds, NULL, NULL, &tv);
}

void print_host(const char *a, const char *b, int both)
{
        int plen = 0;
        printf("%s", a);
        plen = strlen(a);
        if (both) {
                printf(" (%s)", b);
                plen += strlen(b) + 3;
        }
        if (plen >= HOST_COLUMN_SIZE)
                plen = HOST_COLUMN_SIZE - 1;
        printf("%*s", HOST_COLUMN_SIZE - plen, "");
}

int recverr(int fd, int ttl)
{
        int res;
        struct probehdr rcvbuf;
        char cbuf[512];
        struct iovec  iov;
        struct msghdr msg;
        struct cmsghdr *cmsg;
        struct sock_extended_err *e;
        struct sockaddr_in addr;
        struct timeval tv;
        struct timeval *rettv;
        int slot;
        int rethops;
        int sndhops;
        int progress = -1;
        int broken_router;

restart:
        memset(&rcvbuf, -1, sizeof(rcvbuf));
        iov.iov_base = &rcvbuf;
        iov.iov_len = sizeof(rcvbuf);
        msg.msg_name = (__u8*)&addr;
        msg.msg_namelen = sizeof(addr);
        msg.msg_iov = &iov;
        msg.msg_iovlen = 1;
        msg.msg_flags = 0;
        msg.msg_control = cbuf;
        msg.msg_controllen = sizeof(cbuf);

        gettimeofday(&tv, NULL);
        res = recvmsg(fd, &msg, MSG_ERRQUEUE);
        if (res < 0) {
                if (errno == EAGAIN)
                        return progress;
                goto restart;
        }

        progress = mtu;

        rethops = -1;
        sndhops = -1;
        e = NULL;
        rettv = NULL;
        slot = ntohs(addr.sin_port) - base_port;
        if (slot>=0 && slot < 63 && his[slot].hops) {
                sndhops = his[slot].hops;
                rettv = &his[slot].sendtime;
                his[slot].hops = 0;
        }
        broken_router = 0;
        if (res == sizeof(rcvbuf)) {
                if (rcvbuf.ttl == 0 || rcvbuf.tv.tv_sec == 0) {
                        broken_router = 1;
                } else {
                        sndhops = rcvbuf.ttl;
                        rettv = &rcvbuf.tv;
                }
        }

        for (cmsg = CMSG_FIRSTHDR(&msg); cmsg; cmsg = CMSG_NXTHDR(&msg, cmsg)) {
                if (cmsg->cmsg_level == SOL_IP) {
                        if (cmsg->cmsg_type == IP_RECVERR) {
                                e = (struct sock_extended_err *) CMSG_DATA(cmsg);
                        } else if (cmsg->cmsg_type == IP_TTL) {
                                memcpy(&rethops, CMSG_DATA(cmsg), sizeof(rethops));
                        } else {
                                printf("cmsg:%d\n ", cmsg->cmsg_type);
                        }
```

```
                }
        }
        if (e == NULL) {
                printf("no info\n");
                return 0;
        }
        if (e->ee_origin == SO_EE_ORIGIN_LOCAL) {
                printf("%2d?: %*s ", ttl, -(HOST_COLUMN_SIZE - 1), "[LOCALHOST]");
        } else if (e->ee_origin == SO_EE_ORIGIN_ICMP) {
                char abuf[128];
                struct sockaddr_in *sin = (struct sockaddr_in*)(e+1);
                struct hostent *h = NULL;
                char *idn = NULL;

                inet_ntop(AF_INET, &sin->sin_addr, abuf, sizeof(abuf));

                if (sndhops>0)
                        printf("%2d:  ", sndhops);
                else
                        printf("%2d?: ", ttl);

                if (!no_resolve || show_both) {
                        fflush(stdout);
                        h = gethostbyaddr((char *) &sin->sin_addr, sizeof(sin->sin_addr), AF_INET);
                }

#ifdef USE_IDN
                if (h && idna_to_unicode_lzlz(h->h_name, &idn, 0) != IDNA_SUCCESS)
                        idn = NULL;
#endif
                if (no_resolve)
                        print_host(abuf, h ? (idn ? idn : h->h_name) : abuf, show_both);
                else
                        print_host(h ? (idn ? idn : h->h_name) : abuf, abuf, show_both);

#ifdef USE_IDN
                free(idn);
#endif
        }

        if (rettv) {
                int diff = (tv.tv_sec-rettv->tv_sec)*1000000+(tv.tv_usec-rettv->tv_usec);
                printf("%3d.%03dms ", diff/1000, diff%1000);
                if (broken_router)
                        printf("(This broken router returned corrupted payload) ");
        }

        switch (e->ee_errno) {
        case ETIMEDOUT:
                printf("\n");
                break;
        case EMSGSIZE:
                printf("pmtu %d\n", e->ee_info);
                mtu = e->ee_info;
                progress = mtu;
                break;
        case ECONNREFUSED:
                printf("reached\n");
                hops_to = sndhops<0 ? ttl : sndhops;
                hops_from = rethops;
                return 0;
        case EPROTO:
                printf("!P\n");
                return 0;
        case EHOSTUNREACH:
                if (e->ee_origin == SO_EE_ORIGIN_ICMP &&
                    e->ee_type == 11 &&
                    e->ee_code == 0) {
                        if (rethops>=0) {
                                if (rethops<=64)
                                        rethops = 65-rethops;
                                else if (rethops<=128)
                                        rethops = 129-rethops;
                                else
                                        rethops = 256-rethops;
                                if (sndhops>=0 && rethops != sndhops)
                                        printf("asymm %2d ", rethops);
                                else if (sndhops<0 && rethops != ttl)
                                        printf("asymm %2d ", rethops);
                        }
                        printf("\n");
                        break;
                }
                printf("!H\n");
                return 0;
        case ENETUNREACH:
                printf("!N\n");
                return 0;
        case EACCES:
                printf("!A\n");
                return 0;
        default:
                printf("\n");
                errno = e->ee_errno;
                perror("NET ERROR");
                return 0;
        }
        goto restart;
```

```
}

int probe_ttl(int fd, int ttl)
{
        int i;
        struct probehdr *hdr = pktbuf;

        memset(pktbuf, 0, mtu);
restart:
        for (i=0; i<10; i++) {
                int res;

                hdr->ttl = ttl;
                target.sin_port = htons(base_port + hisptr);
                gettimeofday(&hdr->tv, NULL);
                his[hisptr].hops = ttl;
                his[hisptr].sendtime = hdr->tv;
                if (sendto(fd, pktbuf, mtu-overhead, 0, (struct sockaddr*)&target, sizeof(target)) > 0)
                        break;
                res = recverr(fd, ttl);
                his[hisptr].hops = 0;
                if (res==0)
                        return 0;
                if (res > 0)
                        goto restart;
        }
        hisptr = (hisptr + 1)&63;

        if (i<10) {
                data_wait(fd);
                if (recv(fd, pktbuf, mtu, MSG_DONTWAIT) > 0) {
                        printf("%2d?: reply received 8)\n", ttl);
                        return 0;
                }
                return recverr(fd, ttl);
        }

        printf("%2d:  send failed\n", ttl);
        return 0;
}

static void usage(void) __attribute((noreturn));

static void usage(void)
{
        fprintf(stderr, "Usage: tracepath [-n] [-b] [-l <len>] [-p port] <destination>\n");
        exit(-1);
}

int
main(int argc, char **argv)
{
        struct hostent *he;
        int fd;
        int on;
        int ttl;
        char *p;
        int ch;
#ifdef USE_IDN
        int rc;
        setlocale(LC_ALL, "");
#endif

        while ((ch = getopt(argc, argv, "nbh?l:p:")) != EOF) {
                switch(ch) {
                case 'n':
                        no_resolve = 1;
                        break;
                case 'b':
                        show_both = 1;
                        break;
                case 'l':
                        if ((mtu = atoi(optarg)) <= overhead) {
                                fprintf(stderr, "Error: pktlen must be > %d and <= %d.\n",
                                        overhead, INT_MAX);
                                exit(1);
                        }
                        break;
                case 'p':
                        base_port = atoi(optarg);
                        break;
                default:
                        usage();
                }
        }

        argc -= optind;
        argv += optind;

        if (argc != 1)
                usage();

        fd = socket(AF_INET, SOCK_DGRAM, 0);
        if (fd < 0) {
                perror("socket");
                exit(1);
        }
        target.sin_family = AF_INET;
```

```
        /* Backward compatiblity */
        if (!base_port) {
                p = strchr(argv[0], '/');
                if (p) {
                        *p = 0;
                        base_port = atoi(p+1);
                } else
                        base_port = 44444;
        }

        p = argv[0];
#ifdef USE_IDN
        rc = idna_to_ascii_lz(argv[0], &p, 0);
        if (rc != IDNA_SUCCESS) {
                fprintf(stderr, "IDNA encoding failed: %s\n", idna_strerror(rc));
                exit(2);
        }
#endif

        he = gethostbyname2(argv[0], AF_INET);
        if (he == NULL) {
                herror("gethostbyname2");
                exit(1);
        }

#ifdef USE_IDN
        free(p);
#endif

        memcpy(&target.sin_addr, he->h_addr, 4);

        on = IP_PMTUDISC_PROBE;
        if (setsockopt(fd, SOL_IP, IP_MTU_DISCOVER, &on, sizeof(on)) &&
            (on = IP_PMTUDISC_DO,
             setsockopt(fd, SOL_IP, IP_MTU_DISCOVER, &on, sizeof(on)))) {
                perror("IP_MTU_DISCOVER");
                exit(1);
        }
        on = 1;
        if (setsockopt(fd, SOL_IP, IP_RECVERR, &on, sizeof(on))) {
                perror("IP_RECVERR");
                exit(1);
        }
        if (setsockopt(fd, SOL_IP, IP_RECVTTL, &on, sizeof(on))) {
                perror("IP_RECVTTL");
                exit(1);
        }

        pktbuf = malloc(mtu);
        if (!pktbuf) {
                perror("malloc");
                exit(1);
        }

        for (ttl=1; ttl<32; ttl++) {
                int res;
                int i;

                on = ttl;
                if (setsockopt(fd, SOL_IP, IP_TTL, &on, sizeof(on))) {
                        perror("IP_TTL");
                        exit(1);
                }

restart:
                for (i=0; i<3; i++) {
                        int old_mtu;

                        old_mtu = mtu;
                        res = probe_ttl(fd, ttl);
                        if (mtu != old_mtu)
                                goto restart;
                        if (res == 0)
                                goto done;
                        if (res > 0)
                                break;
                }

                if (res < 0)
                        printf("%2d:  no reply\n", ttl);
        }
        printf("     Too many hops: pmtu %d\n", mtu);
done:
        printf("     Resume: pmtu %d ", mtu);
        if (hops_to>=0)
                printf("hops %d ", hops_to);
        if (hops_from>=0)
                printf("back %d ", hops_from);
        printf("\n");
        exit(0);
}
```

Notice for package(s)

```
/*
 * arping.c
 *
 *              This program is free software; you can redistribute it and/or
 *              modify it under the terms of the GNU General Public License
 *              as published by the Free Software Foundation; either version
 *              2 of the License, or (at your option) any later version.
 *
 * Authors:     Alexey Kuznetsov, <kuznet@ms2.inr.ac.ru>
 *              YOSHIFUJI Hideaki <yoshfuji@linux-ipv6.org>
 */

#include <stdlib.h>
#include <sys/param.h>
#include <sys/socket.h>
#include <linux/sockios.h>
#include <sys/file.h>
#include <sys/time.h>
#include <sys/signal.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <linux/if_packet.h>
#include <linux/if_ether.h>
#include <net/if_arp.h>
#include <sys/uio.h>
#ifdef CAPABILITIES
#include <sys/prctl.h>
#include <sys/capability.h>
#endif

#include <netdb.h>
#include <unistd.h>
#include <stdio.h>
#include <ctype.h>
#include <errno.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#ifdef USE_SYSFS
#include <sysfs/libsysfs.h>
struct sysfs_devattr_values;
#endif

#ifndef WITHOUT_IFADDRS
#include <ifaddrs.h>
#endif

#ifdef USE_IDN
#include <idna.h>
#include <locale.h>
#endif

#include "SNAPSHOT.h"

static void usage(void) __attribute__((noreturn));

#ifdef DEFAULT_DEVICE
# define DEFAULT_DEVICE_STR     DEFAULT_DEVICE
#else
# define DEFAULT_DEVICE         NULL
#endif

struct device {
        char *name;
        int ifindex;
#ifndef WITHOUT_IFADDRS
        struct ifaddrs *ifa;
#endif
#ifdef USE_SYSFS
        struct sysfs_devattr_values *sysfs;
#endif
};

int quit_on_reply=0;
struct device device = {
        .name = DEFAULT_DEVICE,
};
char *source;
struct in_addr src, dst;
char *target;
int dad, unsolicited, advert;
int quiet;
int count=-1;
int timeout;
int unicasting;
int s;
int broadcast_only;

struct sockaddr_storage me;
struct sockaddr_storage he;

struct timeval start, last;
```

```
int sent, brd_sent;
int received, brd_recv, req_recv;

#ifndef CAPABILITIES
static uid_t euid;
#endif

#define MS_TDIFF(tv1,tv2) ( ((tv1).tv_sec-(tv2).tv_sec)*1000 + \
                            ((tv1).tv_usec-(tv2).tv_usec)/1000 )

#define OFFSET_OF(name,ele)     ((size_t)(((name *)0)->ele))

static inline socklen_t sll_len(size_t halen)
{
        socklen_t len = OFFSET_OF(struct sockaddr_ll, sll_addr) + halen;
        if (len < sizeof(struct sockaddr_ll))
                len = sizeof(struct sockaddr_ll);
        return len;
}

#define SLL_LEN(hln)            sll_len(hln)

void usage(void)
{
        fprintf(stderr,
                "Usage: arping [-fqbDUAV] [-c count] [-w timeout] [-I device] [-s source] destination\n"
                "  -f : quit on first reply\n"
                "  -q : be quiet\n"
                "  -b : keep broadcasting, don't go unicast\n"
                "  -D : duplicate address detection mode\n"
                "  -U : Unsolicited ARP mode, update your neighbours\n"
                "  -A : ARP answer mode, update your neighbours\n"
                "  -V : print version and exit\n"
                "  -c count : how many packets to send\n"
                "  -w timeout : how long to wait for a reply\n"
                "  -I device : which ethernet device to use"
#ifdef DEFAULT_DEVICE_STR
                        " (" DEFAULT_DEVICE_STR ")"
#endif
                        "\n"
                "  -s source : source ip address\n"
                "  destination : ask for what ip address\n"
                );
        exit(2);
}

void set_signal(int signo, void (*handler)(void))
{
        struct sigaction sa;

        memset(&sa, 0, sizeof(sa));
        sa.sa_handler = (void (*)(int))handler;
        sa.sa_flags = SA_RESTART;
        sigaction(signo, &sa, NULL);
}

#ifdef CAPABILITIES
static const cap_value_t caps[] = { CAP_NET_RAW, };
static cap_flag_value_t cap_raw = CAP_CLEAR;
#endif

void limit_capabilities(void)
{
#ifdef CAPABILITIES
        cap_t cap_p;

        cap_p = cap_get_proc();
        if (!cap_p) {
                perror("arping: cap_get_proc");
                exit(-1);
        }

        cap_get_flag(cap_p, CAP_NET_RAW, CAP_PERMITTED, &cap_raw);

        if (cap_raw != CAP_CLEAR) {
                if (cap_clear(cap_p) < 0) {
                        perror("arping: cap_clear");
                        exit(-1);
                }

                cap_set_flag(cap_p, CAP_PERMITTED, 1, caps, CAP_SET);

                if (cap_set_proc(cap_p) < 0) {
                        perror("arping: cap_set_proc");
                        if (errno != EPERM)
                                exit(-1);
                }
        }

        if (prctl(PR_SET_KEEPCAPS, 1) < 0) {
                perror("arping: prctl");
                exit(-1);
        }

        if (setuid(getuid()) < 0) {
                perror("arping: setuid");
                exit(-1);
        }
```

```
        if (prctl(PR_SET_KEEPCAPS, 0) < 0) {
                perror("arping: prctl");
                exit(-1);
        }

        cap_free(cap_p);
#else
        euid = geteuid();
#endif
}

int modify_capability_raw(int on)
{
#ifdef CAPABILITIES
        cap_t cap_p;

        if (cap_raw != CAP_SET)
                return on ? -1 : 0;

        cap_p = cap_get_proc();
        if (!cap_p) {
                perror("arping: cap_get_proc");
                return -1;
        }

        cap_set_flag(cap_p, CAP_EFFECTIVE, 1, caps, on ? CAP_SET : CAP_CLEAR);

        if (cap_set_proc(cap_p) < 0) {
                perror("arping: cap_set_proc");
                return -1;
        }

        cap_free(cap_p);
#else
        if (setuid(on ? euid : getuid())) {
                perror("arping: setuid");
                return -1;
        }
#endif
        return 0;
}

static inline int enable_capability_raw(void)
{
        return modify_capability_raw(1);
}

static inline int disable_capability_raw(void)
{
        return modify_capability_raw(0);
}

void drop_capabilities(void)
{
#ifdef CAPABILITIES
        cap_t cap_p = cap_init();

        if (!cap_p) {
                perror("arping: cap_init");
                exit(-1);
        }

        if (cap_set_proc(cap_p) < 0) {
                perror("arping: cap_set_proc");
                exit(-1);
        }

        cap_free(cap_p);
#else
        if (setuid(getuid()) < 0) {
                perror("arping: setuid");
                exit(-1);
        }
#endif
}

int send_pack(int s, struct in_addr src, struct in_addr dst,
              struct sockaddr_ll *ME, struct sockaddr_ll *HE)
{
        int err;
        struct timeval now;
        unsigned char buf[256];
        struct arphdr *ah = (struct arphdr*)buf;
        unsigned char *p = (unsigned char *)(ah+1);

        ah->ar_hrd = htons(ME->sll_hatype);
        if (ah->ar_hrd == htons(ARPHRD_FDDI))
                ah->ar_hrd = htons(ARPHRD_ETHER);
        ah->ar_pro = htons(ETH_P_IP);
        ah->ar_hln = ME->sll_halen;
        ah->ar_pln = 4;
        ah->ar_op  = advert ? htons(ARPOP_REPLY) : htons(ARPOP_REQUEST);

        memcpy(p, &ME->sll_addr, ah->ar_hln);
        p+=ME->sll_halen;

        memcpy(p, &src, 4);
```

```
        p+=4;

        if (advert)
                memcpy(p, &ME->sll_addr, ah->ar_hln);
        else
                memcpy(p, &HE->sll_addr, ah->ar_hln);
        p+=ah->ar_hln;

        memcpy(p, &dst, 4);
        p+=4;

        gettimeofday(&now, NULL);
        err = sendto(s, buf, p-buf, 0, (struct sockaddr*)HE, SLL_LEN(ah->ar_hln));
        if (err == p-buf) {
                last = now;
                sent++;
                if (!unicasting)
                        brd_sent++;
        }
        return err;
}

void finish(void)
{
        if (!quiet) {
                printf("Sent %d probes (%d broadcast(s))\n", sent, brd_sent);
                printf("Received %d response(s)", received);
                if (brd_recv || req_recv) {
                        printf(" (");
                        if (req_recv)
                                printf("%d request(s)", req_recv);
                        if (brd_recv)
                                printf("%s%d broadcast(s)",
                                       req_recv ? ", " : "",
                                       brd_recv);
                        printf(")");
                }
                printf("\n");
                fflush(stdout);
        }
        if (dad)
                exit(!!received);
        if (unsolicited)
                exit(0);
        exit(!received);
}

void catcher(void)
{
        struct timeval tv, tv_s, tv_o;

        gettimeofday(&tv, NULL);

        if (start.tv_sec==0)
                start = tv;

        timersub(&tv, &start, &tv_s);
        tv_o.tv_sec = timeout;
        tv_o.tv_usec = 500 * 1000;

        if (count-- == 0 || (timeout && timercmp(&tv_s, &tv_o, >)))
                finish();

        timersub(&tv, &last, &tv_s);
        tv_o.tv_sec = 0;

        if (last.tv_sec==0 || timercmp(&tv_s, &tv_o, >)) {
                send_pack(s, src, dst,
                          (struct sockaddr_ll *)&me, (struct sockaddr_ll *)&he);
                if (count == 0 && unsolicited)
                        finish();
        }
        alarm(1);
}

void print_hex(unsigned char *p, int len)
{
        int i;
        for (i=0; i<len; i++) {
                printf("%02X", p[i]);
                if (i != len-1)
                        printf(":");
        }
}

int recv_pack(unsigned char *buf, int len, struct sockaddr_ll *FROM)
{
        struct timeval tv;
        struct arphdr *ah = (struct arphdr*)buf;
        unsigned char *p = (unsigned char *)(ah+1);
        struct in_addr src_ip, dst_ip;

        gettimeofday(&tv, NULL);

        /* Filter out wild packets */
        if (FROM->sll_pkttype != PACKET_HOST &&
            FROM->sll_pkttype != PACKET_BROADCAST &&
            FROM->sll_pkttype != PACKET_MULTICAST)
```

```
                return 0;

        /* Only these types are recognised */
        if (ah->ar_op != htons(ARPOP_REQUEST) &&
            ah->ar_op != htons(ARPOP_REPLY))
                return 0;

        /* ARPHRD check and this darned FDDI hack here :-( */
        if (ah->ar_hrd != htons(FROM->sll_hatype) &&
            (FROM->sll_hatype != ARPHRD_FDDI || ah->ar_hrd != htons(ARPHRD_ETHER)))
                return 0;

        /* Protocol must be IP. */
        if (ah->ar_pro != htons(ETH_P_IP))
                return 0;
        if (ah->ar_pln != 4)
                return 0;
        if (ah->ar_hln != ((struct sockaddr_ll *)&me)->sll_halen)
                return 0;
        if (len < sizeof(*ah) + 2*(4 + ah->ar_hln))
                return 0;
        memcpy(&src_ip, p+ah->ar_hln, 4);
        memcpy(&dst_ip, p+ah->ar_hln+4+ah->ar_hln, 4);
        if (!dad) {
                if (src_ip.s_addr != dst.s_addr)
                        return 0;
                if (src.s_addr != dst_ip.s_addr)
                        return 0;
                if (memcmp(p+ah->ar_hln+4, ((struct sockaddr_ll *)&me)->sll_addr, ah->ar_hln))
                        return 0;
        } else {
                /* DAD packet was:
                   src_ip = 0 (or some src)
                   src_hw = ME
                   dst_ip = tested address
                   dst_hw = <unspec>

                   We fail, if receive request/reply with:
                   src_ip = tested_address
                   src_hw != ME
                   if src_ip in request was not zero, check
                   also that it matches to dst_ip, otherwise
                   dst_ip/dst_hw do not matter.
                 */
                if (src_ip.s_addr != dst.s_addr)
                        return 0;
                if (memcmp(p, ((struct sockaddr_ll *)&me)->sll_addr, ((struct sockaddr_ll *)&me)->sll_halen) == 0)
                        return 0;
                if (src.s_addr && src.s_addr != dst_ip.s_addr)
                        return 0;
        }
        if (!quiet) {
                int s_printed = 0;
                printf("%s ", FROM->sll_pkttype==PACKET_HOST ? "Unicast" : "Broadcast");
                printf("%s from ", ah->ar_op == htons(ARPOP_REPLY) ? "reply" : "request");
                printf("%s [", inet_ntoa(src_ip));
                print_hex(p, ah->ar_hln);
                printf("] ");
                if (dst_ip.s_addr != src.s_addr) {
                        printf("for %s ", inet_ntoa(dst_ip));
                        s_printed = 1;
                }
                if (memcmp(p+ah->ar_hln+4, ((struct sockaddr_ll *)&me)->sll_addr, ah->ar_hln)) {
                        if (!s_printed)
                                printf("for ");
                        printf("[");
                        print_hex(p+ah->ar_hln+4, ah->ar_hln);
                        printf("]");
                }
                if (last.tv_sec) {
                        long usecs = (tv.tv_sec-last.tv_sec) * 1000000 +
                                tv.tv_usec-last.tv_usec;
                        long msecs = (usecs+500)/1000;
                        usecs -= msecs*1000 - 500;
                        printf(" %ld.%03ldms\n", msecs, usecs);
                } else {
                        printf(" UNSOLICITED?\n");
                }
                fflush(stdout);
        }
        received++;
        if (FROM->sll_pkttype != PACKET_HOST)
                brd_recv++;
        if (ah->ar_op == htons(ARPOP_REQUEST))
                req_recv++;
        if (quit_on_reply)
                finish();
        if(!broadcast_only) {
                memcpy(((struct sockaddr_ll *)&he)->sll_addr, p, ((struct sockaddr_ll *)&me)->sll_halen);
                unicasting=1;
        }
        return 1;
}

#ifdef USE_SYSFS
union sysfs_devattr_value {
        unsigned long   ulong;
        void            *ptr;
```

```
};

enum {
        SYSFS_DEVATTR_IFINDEX,
        SYSFS_DEVATTR_FLAGS,
        SYSFS_DEVATTR_ADDR_LEN,
#if 0
        SYSFS_DEVATTR_TYPE,
        SYSFS_DEVATTR_ADDRESS,
#endif
        SYSFS_DEVATTR_BROADCAST,
        SYSFS_DEVATTR_NUM
};

struct sysfs_devattr_values
{
        char *ifname;
        union sysfs_devattr_value       value[SYSFS_DEVATTR_NUM];
};

static int sysfs_devattr_ulong_dec(char *ptr, struct sysfs_devattr_values *v, unsigned idx);
static int sysfs_devattr_ulong_hex(char *ptr, struct sysfs_devattr_values *v, unsigned idx);
static int sysfs_devattr_macaddr(char *ptr, struct sysfs_devattr_values *v, unsigned idx);

struct sysfs_devattrs {
        const char *name;
        int (*handler)(char *ptr, struct sysfs_devattr_values *v, unsigned int idx);
        int free;
} sysfs_devattrs[SYSFS_DEVATTR_NUM] = {
        [SYSFS_DEVATTR_IFINDEX] = {
                .name           = "ifindex",
                .handler        = sysfs_devattr_ulong_dec,
        },
        [SYSFS_DEVATTR_ADDR_LEN] = {
                .name           = "addr_len",
                .handler        = sysfs_devattr_ulong_dec,
        },
        [SYSFS_DEVATTR_FLAGS] = {
                .name           = "flags",
                .handler        = sysfs_devattr_ulong_hex,
        },
#if 0
        [SYSFS_DEVATTR_TYPE] = {
                .name           = "type",
                .handler        = sysfs_devattr_ulong_dec,
        },
        [SYSFS_DEVATTR_ADDRESS] = {
                .name           = "address",
                .handler        = sysfs_devattr_macaddr,
                .free           = 1,
        },
#endif
        [SYSFS_DEVATTR_BROADCAST] = {
                .name           = "broadcast",
                .handler        = sysfs_devattr_macaddr,
                .free           = 1,
        },
};
#endif

/*
 * find_device()
 *
 * This function checks 1) if the device (if given) is okay for ARP,
 * or 2) find fist appropriate device on the system.
 *
 * Return value:
 *      >0      : Succeeded, and appropriate device not found.
 *                device.ifindex remains 0.
 *      0       : Succeeded, and approptiate device found.
 *                device.ifindex is set.
 *      <0      : Failed.  Support not found, or other
 *                : system error.  Try other method.
 *
 * If an appropriate device found, it is recorded inside the
 * "device" variable for later reference.
 *
 * We have several implementations for this.
 *      by_ifaddrs():   requires getifaddr() in glibc, and rtnetlink in
 *                      kernel. default and recommended for recent systems.
 *      by_sysfs():     requires libsysfs , and sysfs in kernel.
 *      by_ioctl():     unable to list devices without ipv4 address; this
 *                      means, you need to supply the device name for
 *                      DAD purpose.
 */
/* Common check for ifa->ifa_flags */
static int check_ifflags(unsigned int ifflags, int fatal)
{
        if (!(ifflags & IFF_UP)) {
                if (fatal) {
                        if (!quiet)
                                printf("Interface \"%s\" is down\n", device.name);
                        exit(2);
                }
                return -1;
        }
        if (ifflags & (IFF_NOARP | IFF_LOOPBACK)) {
                if (fatal) {
```

```
                        if (!quiet)
                                printf("Interface \"%s\" is not ARPable\n", device.name);
                        exit(dad ? 0 : 2);
                }
                return -1;
        }
        return 0;
}

static int find_device_by_ifaddrs(void)
{
#ifndef WITHOUT_IFADDRS
        int rc;
        struct ifaddrs *ifa0, *ifa;
        int count = 0;

        rc = getifaddrs(&ifa0);
        if (rc) {
                perror("getifaddrs");
                return -1;
        }

        for (ifa = ifa0; ifa; ifa = ifa->ifa_next) {
                if (!ifa->ifa_addr)
                        continue;
                if (ifa->ifa_addr->sa_family != AF_PACKET)
                        continue;
                if (device.name && ifa->ifa_name && strcmp(ifa->ifa_name, device.name))
                        continue;

                if (check_ifflags(ifa->ifa_flags, device.name != NULL) < 0)
                        continue;

                if (!((struct sockaddr_ll *)ifa->ifa_addr)->sll_halen)
                        continue;
                if (!ifa->ifa_broadaddr)
                        continue;

                device.ifa = ifa;

                if (count++)
                        break;
        }

        if (count == 1 && device.ifa) {
                device.ifindex = if_nametoindex(device.ifa->ifa_name);
                if (!device.ifindex) {
                        perror("arping: if_nametoindex");
                        freeifaddrs(ifa0);
                        return -1;
                }
                device.name  = device.ifa->ifa_name;
                return 0;
        }
        return 1;
#else
        return -1;
#endif
}

#ifdef USE_SYSFS
static void sysfs_devattr_values_init(struct sysfs_devattr_values *v, int do_free)
{
        int i;
        if (do_free) {
                free(v->ifname);
                for (i = 0; i < SYSFS_DEVATTR_NUM; i++) {
                        if (sysfs_devattrs[i].free)
                                free(v->value[i].ptr);
                }
        }
        memset(v, 0, sizeof(*v));
}

static int sysfs_devattr_ulong(char *ptr, struct sysfs_devattr_values *v, unsigned int idx,
                                    unsigned int base)
{
        unsigned long *p;
        char *ep;

        if (!ptr || !v)
                return -1;

        p = &v->value[idx].ulong;
        errno = 0;
        *p = strtoul(ptr, &ep, base);
        if ((*ptr && isspace(*ptr & 0xff)) || errno || (*ep != '\0' && *ep != '\n'))
                goto out;

        return 0;
out:
        return -1;
}

static int sysfs_devattr_ulong_dec(char *ptr, struct sysfs_devattr_values *v, unsigned int idx)
{
        int rc = sysfs_devattr_ulong(ptr, v, idx, 10);
        return rc;
```

```
}

static int sysfs_devattr_ulong_hex(char *ptr, struct sysfs_devattr_values *v, unsigned int idx)
{
        int rc = sysfs_devattr_ulong(ptr, v, idx, 16);
        return rc;
}

static int sysfs_devattr_macaddr(char *ptr, struct sysfs_devattr_values *v, unsigned int idx)
{
        unsigned char *m;
        int i;
        unsigned int addrlen;

        if (!ptr || !v)
                return -1;

        addrlen = v->value[SYSFS_DEVATTR_ADDR_LEN].ulong;
        m = malloc(addrlen);

        for (i = 0; i < addrlen; i++) {
                if (i && *(ptr + i * 3 - 1) != ':')
                        goto out;
                if (sscanf(ptr + i * 3, "%02hhx", &m[i]) != 1)
                        goto out;
        }

        v->value[idx].ptr = m;
        return 0;
out:
        free(m);
        return -1;
}
#endif

int find_device_by_sysfs(void)
{
        int rc = -1;
#ifdef USE_SYSFS
        struct sysfs_class *cls_net;
        struct dlist *dev_list;
        struct sysfs_class_device *dev;
        struct sysfs_attribute *dev_attr;
        struct sysfs_devattr_values sysfs_devattr_values;
        int count = 0;

        if (!device.sysfs) {
                device.sysfs = malloc(sizeof(*device.sysfs));
                sysfs_devattr_values_init(device.sysfs, 0);
        }

        cls_net = sysfs_open_class("net");
        if (!cls_net) {
                perror("sysfs_open_class");
                return -1;
        }

        dev_list = sysfs_get_class_devices(cls_net);
        if (!dev_list) {
                perror("sysfs_get_class_devices");
                goto out;
        }

        sysfs_devattr_values_init(&sysfs_devattr_values, 0);

        dlist_for_each_data(dev_list, dev, struct sysfs_class_device) {
                int i;
                int rc = -1;

                if (device.name && strcmp(dev->name, device.name))
                        goto do_next;

                sysfs_devattr_values_init(&sysfs_devattr_values, 1);

                for (i = 0; i < SYSFS_DEVATTR_NUM; i++) {

                        dev_attr = sysfs_get_classdev_attr(dev, sysfs_devattrs[i].name);
                        if (!dev_attr) {
                                perror("sysfs_get_classdev_attr");
                                rc = -1;
                                break;
                        }
                        if (sysfs_read_attribute(dev_attr)) {
                                perror("sysfs_read_attribute");
                                rc = -1;
                                break;
                        }
                        rc = sysfs_devattrs[i].handler(dev_attr->value, &sysfs_devattr_values, i);

                        if (rc < 0)
                                break;
                }

                if (rc < 0)
                        goto do_next;

                if (check_ifflags(sysfs_devattr_values.value[SYSFS_DEVATTR_FLAGS].ulong,
                                  device.name != NULL) < 0)
```

```
                        goto do_next;

                if (!sysfs_devattr_values.value[SYSFS_DEVATTR_ADDR_LEN].ulong)
                        goto do_next;

                if (device.sysfs->value[SYSFS_DEVATTR_IFINDEX].ulong) {
                        if (device.sysfs->value[SYSFS_DEVATTR_FLAGS].ulong & IFF_RUNNING)
                                goto do_next;
                }

                sysfs_devattr_values.ifname = strdup(dev->name);
                if (!sysfs_devattr_values.ifname) {
                        perror("malloc");
                        goto out;
                }

                sysfs_devattr_values_init(device.sysfs, 1);
                memcpy(device.sysfs, &sysfs_devattr_values, sizeof(*device.sysfs));
                sysfs_devattr_values_init(&sysfs_devattr_values, 0);

                if (count++)
                        break;

                continue;
do_next:
                sysfs_devattr_values_init(&sysfs_devattr_values, 1);
        }

        if (count == 1) {
                device.ifindex = device.sysfs->value[SYSFS_DEVATTR_IFINDEX].ulong;
                device.name = device.sysfs->ifname;
        }
        rc = !device.ifindex;
out:
        sysfs_close_class(cls_net);
#endif
        return rc;
}

static int check_device_by_ioctl(int s, struct ifreq *ifr)
{
        if (ioctl(s, SIOCGIFFLAGS, ifr) < 0) {
                perror("ioctl(SIOCGIFINDEX");
                return -1;
        }

        if (check_ifflags(ifr->ifr_flags, device.name != NULL) < 0)
                return 1;

        if (ioctl(s, SIOCGIFINDEX, ifr) < 0) {
                perror("ioctl(SIOCGIFINDEX");
                return -1;
        }

        return 0;
}

static int find_device_by_ioctl(void)
{
        int s;
        struct ifreq *ifr0, *ifr, *ifr_end;
        size_t ifrsize = sizeof(*ifr);
        struct ifconf ifc;
        static struct ifreq ifrbuf;
        int count = 0;

        s = socket(AF_INET, SOCK_DGRAM, 0);
        if (s < 0) {
                perror("socket");
                return -1;
        }

        memset(&ifrbuf, 0, sizeof(ifrbuf));

        if (device.name) {
                strncpy(ifrbuf.ifr_name, device.name, sizeof(ifrbuf.ifr_name) - 1);
                if (check_device_by_ioctl(s, &ifrbuf))
                        goto out;
                count++;
        } else {
                do {
                        int rc;
                        ifr0 = malloc(ifrsize);
                        if (!ifr0) {
                                perror("malloc");
                                goto out;
                        }

                        ifc.ifc_buf = (char *)ifr0;
                        ifc.ifc_len = ifrsize;

                        rc = ioctl(s, SIOCGIFCONF, &ifc);
                        if (rc < 0) {
                                perror("ioctl(SIOCFIFCONF");
                                goto out;
                        }

                        if (ifc.ifc_len + sizeof(*ifr0) + sizeof(struct sockaddr_storage) - sizeof(struct sockaddr) <= ifrsize)
```

```
                                break;
                        ifrsize *= 2;
                        free(ifr0);
                        ifr0 = NULL;
                } while(ifrsize < INT_MAX / 2);

                if (!ifr0) {
                        fprintf(stderr, "arping: too many interfaces!?\n");
                        goto out;
                }

                ifr_end = (struct ifreq *)(((char *)ifr0) + ifc.ifc_len - sizeof(*ifr0));
                for (ifr = ifr0; ifr <= ifr_end; ifr++) {
                        if (check_device_by_ioctl(s, &ifrbuf))
                                continue;
                        memcpy(&ifrbuf.ifr_name, ifr->ifr_name, sizeof(ifrbuf.ifr_name));
                        if (count++)
                                break;
                }
        }

        close(s);

        if (count == 1) {
                device.ifindex = ifrbuf.ifr_ifindex;
                device.name = ifrbuf.ifr_name;
        }
        return !device.ifindex;
out:
        close(s);
        return -1;
}

static int find_device(void)
{
        int rc;
        rc = find_device_by_ifaddrs();
        if (rc >= 0)
                goto out;
        rc = find_device_by_sysfs();
        if (rc >= 0)
                goto out;
        rc = find_device_by_ioctl();
out:
        return rc;
}

/*
 * set_device_broadcast()
 *
 * This fills the device "broadcast address"
 * based on information found by find_device() funcion.
 */
static int set_device_broadcast_ifaddrs_one(struct device *device, unsigned char *ba, size_t balen, int fatal)
{
#ifndef WITHOUT_IFADDRS
        struct ifaddrs *ifa;
        struct sockaddr_ll *sll;

        if (!device)
                return -1;

        ifa = device->ifa;
        if (!ifa)
                return -1;

        sll = (struct sockaddr_ll *)ifa->ifa_broadaddr;

        if (sll->sll_halen != balen) {
                if (fatal) {
                        if (!quiet)
                                printf("Address length does not match...\n");
                        exit(2);
                }
                return -1;
        }
        memcpy(ba, sll->sll_addr, sll->sll_halen);
        return 0;
#else
        return -1;
#endif
}
int set_device_broadcast_sysfs(struct device *device, unsigned char *ba, size_t balen)
{
#ifdef USE_SYSFS
        struct sysfs_devattr_values *v;
        if (!device)
                return -1;
        v = device->sysfs;
        if (!v)
                return -1;
        if (v->value[SYSFS_DEVATTR_ADDR_LEN].ulong != balen)
                return -1;
        memcpy(ba, v->value[SYSFS_DEVATTR_BROADCAST].ptr, balen);
        return 0;
#else
        return -1;
#endif
```

```
}

static int set_device_broadcast_fallback(struct device *device, unsigned char *ba, size_t balen)
{
        if (!quiet)
                fprintf(stderr, "WARNING: using default broadcast address.\n");
        memset(ba, -1, balen);
        return 0;
}

static void set_device_broadcast(struct device *dev, unsigned char *ba, size_t balen)
{
        if (!set_device_broadcast_ifaddrs_one(dev, ba, balen, 0))
                return;
        if (!set_device_broadcast_sysfs(dev, ba, balen))
                return;
        set_device_broadcast_fallback(dev, ba, balen);
}

int
main(int argc, char **argv)
{
        int socket_errno;
        int ch;

        limit_capabilities();

#ifdef USE_IDN
        setlocale(LC_ALL, "");
#endif

        enable_capability_raw();

        s = socket(PF_PACKET, SOCK_DGRAM, 0);
        socket_errno = errno;

        disable_capability_raw();

        while ((ch = getopt(argc, argv, "h?bfDUAqc:w:s:I:V")) != EOF) {
                switch(ch) {
                case 'b':
                        broadcast_only=1;
                        break;
                case 'D':
                        dad++;
                        quit_on_reply=1;
                        break;
                case 'U':
                        unsolicited++;
                        break;
                case 'A':
                        advert++;
                        unsolicited++;
                        break;
                case 'q':
                        quiet++;
                        break;
                case 'c':
                        count = atoi(optarg);
                        break;
                case 'w':
                        timeout = atoi(optarg);
                        break;
                case 'I':
                        device.name = optarg;
                        break;
                case 'f':
                        quit_on_reply=1;
                        break;
                case 's':
                        source = optarg;
                        break;
                case 'V':
                        printf("arping utility, iputils-%s\n", SNAPSHOT);
                        exit(0);
                case 'h':
                case '?':
                default:
                        usage();
                }
        }
        argc -= optind;
        argv += optind;

        if (argc != 1)
                usage();

        target = *argv;

        if (device.name && !*device.name)
                device.name = NULL;

        if (s < 0) {
                errno = socket_errno;
                perror("arping: socket");
                exit(2);
        }
```

```
        if (find_device() < 0)
                exit(2);

        if (!device.ifindex) {
                if (device.name) {
                        fprintf(stderr, "arping: Device %s not available.\n", device.name);
                        exit(2);
                }
                fprintf(stderr, "arping: device (option -I) is required.\n");
                usage();
        }

        if (inet_aton(target, &dst) != 1) {
                struct hostent *hp;
                char *idn = target;
#ifdef USE_IDN
                int rc;

                rc = idna_to_ascii_lz(target, &idn, 0);

                if (rc != IDNA_SUCCESS) {
                        fprintf(stderr, "arping: IDN encoding failed: %s\n", idna_strerror(rc));
                        exit(2);
                }
#endif

                hp = gethostbyname2(idn, AF_INET);
                if (!hp) {
                        fprintf(stderr, "arping: unknown host %s\n", target);
                        exit(2);
                }

#ifdef USE_IDN
                free(idn);
#endif

                memcpy(&dst, hp->h_addr, 4);
        }

        if (source && inet_aton(source, &src) != 1) {
                fprintf(stderr, "arping: invalid source %s\n", source);
                exit(2);
        }

        if (!dad && unsolicited && src.s_addr == 0)
                src = dst;

        if (!dad || src.s_addr) {
                struct sockaddr_in saddr;
                int probe_fd = socket(AF_INET, SOCK_DGRAM, 0);

                if (probe_fd < 0) {
                        perror("socket");
                        exit(2);
                }
                if (device.name) {
                        enable_capability_raw();

                        if (setsockopt(probe_fd, SOL_SOCKET, SO_BINDTODEVICE, device.name, strlen(device.name)+1) == -1)
                                perror("WARNING: interface is ignored");

                        disable_capability_raw();
                }
                memset(&saddr, 0, sizeof(saddr));
                saddr.sin_family = AF_INET;
                if (src.s_addr) {
                        saddr.sin_addr = src;
                        if (bind(probe_fd, (struct sockaddr*)&saddr, sizeof(saddr)) == -1) {
                                perror("bind");
                                exit(2);
                        }
                } else if (!dad) {
                        int on = 1;
                        socklen_t alen = sizeof(saddr);

                        saddr.sin_port = htons(1025);
                        saddr.sin_addr = dst;

                        if (setsockopt(probe_fd, SOL_SOCKET, SO_DONTROUTE, (char*)&on, sizeof(on)) == -1)
                                perror("WARNING: setsockopt(SO_DONTROUTE)");
                        if (connect(probe_fd, (struct sockaddr*)&saddr, sizeof(saddr)) == -1) {
                                perror("connect");
                                exit(2);
                        }
                        if (getsockname(probe_fd, (struct sockaddr*)&saddr, &alen) == -1) {
                                perror("getsockname");
                                exit(2);
                        }
                        src = saddr.sin_addr;
                }
                close(probe_fd);
        };

        ((struct sockaddr_ll *)&me)->sll_family = AF_PACKET;
        ((struct sockaddr_ll *)&me)->sll_ifindex = device.ifindex;
        ((struct sockaddr_ll *)&me)->sll_protocol = htons(ETH_P_ARP);
        if (bind(s, (struct sockaddr*)&me, sizeof(me)) == -1) {
                perror("bind");
```

```
                exit(2);
        }

        if (1) {
                socklen_t alen = sizeof(me);
                if (getsockname(s, (struct sockaddr*)&me, &alen) == -1) {
                        perror("getsockname");
                        exit(2);
                }
        }
        if (((struct sockaddr_ll *)&me)->sll_halen == 0) {
                if (!quiet)
                        printf("Interface \"%s\" is not ARPable (no ll address)\n", device.name);
                exit(dad?0:2);
        }

        he = me;

        set_device_broadcast(&device, ((struct sockaddr_ll *)&he)->sll_addr,
                             ((struct sockaddr_ll *)&he)->sll_halen);

        if (!quiet) {
                printf("ARPING %s ", inet_ntoa(dst));
                printf("from %s %s\n",  inet_ntoa(src), device.name ? : "");
        }

        if (!src.s_addr && !dad) {
                fprintf(stderr, "arping: no source address in not-DAD mode\n");
                exit(2);
        }

        drop_capabilities();

        set_signal(SIGINT, finish);
        set_signal(SIGALRM, catcher);

        catcher();

        while(1) {
                sigset_t sset, osset;
                unsigned char packet[4096];
                struct sockaddr_storage from;
                socklen_t alen = sizeof(from);
                int cc;

                if ((cc = recvfrom(s, packet, sizeof(packet), 0,
                                   (struct sockaddr *)&from, &alen)) < 0) {
                        perror("arping: recvfrom");
                        continue;
                }

                sigemptyset(&sset);
                sigaddset(&sset, SIGALRM);
                sigaddset(&sset, SIGINT);
                sigprocmask(SIG_BLOCK, &sset, &osset);
                recv_pack(packet, cc, (struct sockaddr_ll *)&from);
                sigprocmask(SIG_SETMASK, &osset, NULL);
        }
}
```

Notice for package(s)

iputils

```
/*
 * Copyright (c) 1983 Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the University of
 *      California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
```

```
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

#ifndef lint
char copyright[] =
"@(#) Copyright (c) 1983 Regents of the University of California.\n\
 All rights reserved.\n";
#endif /* not lint */

#ifndef lint
/*static char sccsid[] = "from: @(#)tftpd.c     5.13 (Berkeley) 2/26/91";*/
/*static char rcsid[] = "$Id: tftpd.c,v 1.3 1993/08/01 18:28:53 mycroft Exp $";*/
#endif /* not lint */

/*
 * Trivial file transfer protocol server.
 *
 * This version includes many modifications by Jim Guyton <guyton@rand-unix>
 */

#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <signal.h>
#include <fcntl.h>

#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#include <setjmp.h>
#include <syslog.h>
#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

#include "tftp.h"

#ifndef MSG_CONFIRM
#define MSG_CONFIRM 0
#warning Please, upgrade kernel, otherwise this tftpd has no advantages.
#endif

#define TIMEOUT         5

int     peer;
int     rexmtval = TIMEOUT;
int     maxtimeout = 5*TIMEOUT;

#define PKTSIZE SEGSIZE+4
char    buf[PKTSIZE];
char    ackbuf[PKTSIZE];
union {
        struct  sockaddr     sa;
        struct  sockaddr_in  sin;
        struct  sockaddr_in6 sin6;
} from;
socklen_t       fromlen;

#define MAXARG  1
char    *dirs[MAXARG+1];

void tftp(struct tftphdr *tp, int size) __attribute__((noreturn));
void nak(int error);
int validate_access(char *filename, int mode);

struct formats;

void sendfile(struct formats *pf);
void recvfile(struct formats *pf);


int main(int ac, char **av)
{
        register struct tftphdr *tp;
        register int n = 0;
        int on = 1;

        /* Sanity. If parent forgot to setuid() on us. */
        if (geteuid() == 0) {
                setgid(65534);
                setuid(65534);
        }

        ac--; av++;
        while (ac-- > 0 && n < MAXARG)
                dirs[n++] = *av++;

        openlog("tftpd", LOG_PID, LOG_DAEMON);
        if (ioctl(0, FIONBIO, &on) < 0) {
```

```
                syslog(LOG_ERR, "ioctl(FIONBIO): %m\n");
                exit(1);
        }
        fromlen = sizeof (from);
        n = recvfrom(0, buf, sizeof (buf), 0,
            (struct sockaddr *)&from, &fromlen);
        if (n < 0) {
                if (errno != EAGAIN)
                        syslog(LOG_ERR, "recvfrom: %m\n");
                exit(1);
        }
        /*
         * Now that we have read the message out of the UDP
         * socket, we fork and exit.  Thus, inetd will go back
         * to listening to the tftp port, and the next request
         * to come in will start up a new instance of tftpd.
         *
         * We do this so that inetd can run tftpd in "wait" mode.
         * The problem with tftpd running in "nowait" mode is that
         * inetd may get one or more successful "selects" on the
         * tftp port before we do our receive, so more than one
         * instance of tftpd may be started up.  Worse, if tftpd
         * break before doing the above "recvfrom", inetd would
         * spawn endless instances, clogging the system.
         */
        {
                int pid;
                int i;
                socklen_t j;

                for (i = 1; i < 20; i++) {
                    pid = fork();
                    if (pid < 0) {
                                sleep(i);
                                /*
                                 * flush out to most recently sent request.
                                 *
                                 * This may drop some request, but those
                                 * will be resent by the clients when
                                 * they timeout.  The positive effect of
                                 * this flush is to (try to) prevent more
                                 * than one tftpd being started up to service
                                 * a single request from a single client.
                                 */
                                j = sizeof from;
                                i = recvfrom(0, buf, sizeof (buf), 0,
                                    (struct sockaddr *)&from, &j);
                                if (i > 0) {
                                        n = i;
                                        fromlen = j;
                                }
                    } else {
                                break;
                    }
                }
                if (pid < 0) {
                        syslog(LOG_ERR, "fork: %m\n");
                        exit(1);
                } else if (pid != 0) {
                        exit(0);
                }
        }
        alarm(0);
        close(0);
        close(1);
        peer = socket(from.sa.sa_family, SOCK_DGRAM, 0);
        if (peer < 0) {
                syslog(LOG_ERR, "socket: %m\n");
                exit(1);
        }
        if (connect(peer, (struct sockaddr *)&from, sizeof(from)) < 0) {
                syslog(LOG_ERR, "connect: %m\n");
                exit(1);
        }
        tp = (struct tftphdr *)buf;
        tp->th_opcode = ntohs(tp->th_opcode);
        if (tp->th_opcode == RRQ || tp->th_opcode == WRQ)
                tftp(tp, n);
        exit(1);
}

struct formats {
        char    *f_mode;
        int     (*f_validate)(char *filename, int mode);
        void    (*f_send)(struct formats*);
        void    (*f_recv)(struct formats*);
        int     f_convert;
} formats[] = {
        { "netascii",   validate_access,        sendfile,       recvfile, 1 },
        { "octet",      validate_access,        sendfile,       recvfile, 0 },
#ifdef notdef
        { "mail",       validate_user,          sendmail,       recvmail, 1 },
#endif
        { 0 }
};

/*
 * Handle initial connection protocol.
```

```
 */
void tftp(struct tftphdr *tp, int size)
{
        register char *cp;
        int first = 1, ecode;
        register struct formats *pf;
        char *filename, *mode = NULL;

        filename = cp = tp->th_stuff;
again:
        while (cp < buf + size) {
                if (*cp == '\0')
                        break;
                cp++;
        }
        if (*cp != '\0') {
                nak(EBADOP);
                exit(1);
        }
        if (first) {
                mode = ++cp;
                first = 0;
                goto again;
        }
        for (cp = mode; *cp; cp++)
                if (isupper(*cp))
                        *cp = tolower(*cp);
        for (pf = formats; pf->f_mode; pf++)
                if (strcmp(pf->f_mode, mode) == 0)
                        break;
        if (pf->f_mode == 0) {
                nak(EBADOP);
                exit(1);
        }
        ecode = (*pf->f_validate)(filename, tp->th_opcode);
        if (ecode) {
                nak(ecode);
                exit(1);
        }
        if (tp->th_opcode == WRQ)
                (*pf->f_recv)(pf);
        else
                (*pf->f_send)(pf);
        exit(0);
}


FILE *file;

/*
 * Validate file access.  Since we
 * have no uid or gid, for now require
 * file to exist and be publicly
 * readable/writable.
 * If we were invoked with arguments
 * from inetd then the file must also be
 * in one of the given directory prefixes.
 * Note also, full path name must be
 * given as we have no login directory.
 */
int validate_access(char *filename, int mode)
{
        struct stat stbuf;
        int    fd;
        char  *cp;
        char   fnamebuf[1024+512];

        for (cp = filename; *cp; cp++) {
                if(*cp == '.' && (cp == filename || strncmp(cp-1, "/../", 4) == 0)) {
                        syslog(LOG_ERR, "bad path %s", filename);
                        return(EACCESS);
                }
        }

        if (*filename == '/')
                filename++;

        if (!*dirs) {
                syslog(LOG_ERR, "no dirs");
                return EACCESS;
        }
        snprintf(fnamebuf, sizeof(fnamebuf)-1, "%s/%s", *dirs, filename);
        filename = fnamebuf;

        if (stat(filename, &stbuf) < 0) {
                syslog(LOG_ERR, "stat %s : %m", filename);
                return (errno == ENOENT ? ENOTFOUND : EACCESS);
        }
        if (mode == RRQ) {
                if ((stbuf.st_mode&(S_IREAD >> 6)) == 0) {
                        syslog(LOG_ERR, "not readable %s", filename);
                        return (EACCESS);
                }
        } else {
                if ((stbuf.st_mode&(S_IWRITE >> 6)) == 0) {
                        syslog(LOG_ERR, "not writable %s", filename);
                        return (EACCESS);
                }
```

```
        }
        fd = open(filename, mode == RRQ ? 0 : 1);
        if (fd < 0) {
                syslog(LOG_ERR, "cannot open %s: %m", filename);
                return (errno + 100);
        }
        file = fdopen(fd, (mode == RRQ)? "r":"w");
        if (file == NULL) {
                return errno+100;
        }
        return (0);
}

int     confirmed;
int     timeout;
jmp_buf timeoutbuf;

void timer(int signo)
{
        confirmed = 0;
        timeout += rexmtval;
        if (timeout >= maxtimeout)
                exit(1);
        longjmp(timeoutbuf, 1);
}

/*
 * Send the requested file.
 */
void sendfile(struct formats *pf)
{
        struct tftphdr *dp;
        register struct tftphdr *ap;    /* ack packet */
        volatile int block = 1;
        int size, n;

        confirmed = 0;
        signal(SIGALRM, timer);
        dp = r_init();
        ap = (struct tftphdr *)ackbuf;
        do {
                size = readit(file, &dp, pf->f_convert);
                if (size < 0) {
                        nak(errno + 100);
                        goto abort;
                }
                dp->th_opcode = htons((u_short)DATA);
                dp->th_block = htons((u_short)block);
                timeout = 0;
                (void) setjmp(timeoutbuf);

send_data:
                if (send(peer, dp, size + 4, confirmed) != size + 4) {
                        syslog(LOG_ERR, "tftpd: write: %m\n");
                        goto abort;
                }
                confirmed = 0;
                read_ahead(file, pf->f_convert);
                for ( ; ; ) {
                        alarm(rexmtval);        /* read the ack */
                        n = recv(peer, ackbuf, sizeof (ackbuf), 0);
                        alarm(0);
                        if (n < 0) {
                                syslog(LOG_ERR, "tftpd: read: %m\n");
                                goto abort;
                        }
                        ap->th_opcode = ntohs((u_short)ap->th_opcode);
                        ap->th_block = ntohs((u_short)ap->th_block);

                        if (ap->th_opcode == ERROR)
                                goto abort;

                        if (ap->th_opcode == ACK) {
                                if (ap->th_block == block) {
                                        confirmed = MSG_CONFIRM;
                                        break;
                                }
                                /* Re-synchronize with the other side */
                                synchnet(peer);
                                if (ap->th_block == (block -1)) {
                                        goto send_data;
                                }
                        }

                }
                block++;
        } while (size == SEGSIZE);
abort:
        (void) fclose(file);
}

void justquit(int signo)
{
        exit(0);
}


/*
```

```
 * Receive a file.
 */
void recvfile(struct formats *pf)
{
        struct tftphdr *dp;
        register struct tftphdr *ap;    /* ack buffer */
        volatile int block = 0, n, size;

        confirmed = 0;
        signal(SIGALRM, timer);
        dp = w_init();
        ap = (struct tftphdr *)ackbuf;
        do {
                timeout = 0;
                ap->th_opcode = htons((u_short)ACK);
                ap->th_block = htons((u_short)block);
                block++;
                (void) setjmp(timeoutbuf);
send_ack:
                if (send(peer, ackbuf, 4, confirmed) != 4) {
                        syslog(LOG_ERR, "tftpd: write: %m\n");
                        goto abort;
                }
                confirmed = 0;
                write_behind(file, pf->f_convert);
                for ( ; ; ) {
                        alarm(rexmtval);
                        n = recv(peer, dp, PKTSIZE, 0);
                        alarm(0);
                        if (n < 0) {            /* really? */
                                syslog(LOG_ERR, "tftpd: read: %m\n");
                                goto abort;
                        }
                        dp->th_opcode = ntohs((u_short)dp->th_opcode);
                        dp->th_block = ntohs((u_short)dp->th_block);
                        if (dp->th_opcode == ERROR)
                                goto abort;
                        if (dp->th_opcode == DATA) {
                                if (dp->th_block == block) {
                                        confirmed = MSG_CONFIRM;
                                        break;   /* normal */
                                }
                                /* Re-synchronize with the other side */
                                (void) synchnet(peer);
                                if (dp->th_block == (block-1))
                                        goto send_ack;          /* rexmit */
                        }
                }
                /*  size = write(file, dp->th_data, n - 4); */
                size = writeit(file, &dp, n - 4, pf->f_convert);
                if (size != (n-4)) {                    /* ahem */
                        if (size < 0) nak(errno + 100);
                        else nak(ENOSPACE);
                        goto abort;
                }
        } while (size == SEGSIZE);
        write_behind(file, pf->f_convert);
        (void) fclose(file);            /* close data file */

        ap->th_opcode = htons((u_short)ACK);    /* send the "final" ack */
        ap->th_block = htons((u_short)(block));
        (void) send(peer, ackbuf, 4, confirmed);

        signal(SIGALRM, justquit);      /* just quit on timeout */
        alarm(rexmtval);
        n = recv(peer, buf, sizeof (buf), 0); /* normally times out and quits */
        alarm(0);
        if (n >= 4 &&                   /* if read some data */
            dp->th_opcode == DATA &&    /* and got a data block */
            block == dp->th_block) {    /* then my last ack was lost */
                (void) send(peer, ackbuf, 4, 0);     /* resend final ack */
        }
abort:
        return;
}

struct errmsg {
        int     e_code;
        char    *e_msg;
} errmsgs[] = {
        { EUNDEF,       "Undefined error code" },
        { ENOTFOUND,    "File not found" },
        { EACCESS,      "Access violation" },
        { ENOSPACE,     "Disk full or allocation exceeded" },
        { EBADOP,       "Illegal TFTP operation" },
        { EBADID,       "Unknown transfer ID" },
        { EEXISTS,      "File already exists" },
        { ENOUSER,      "No such user" },
        { -1,           0 }
};

/*
 * Send a nak packet (error message).
 * Error code passed in is one of the
 * standard TFTP codes, or a UNIX errno
 * offset by 100.
 */
void nak(int error)
```

```
{
        register struct tftphdr *tp;
        int length;
        register struct errmsg *pe;

        tp = (struct tftphdr *)buf;
        tp->th_opcode = htons((u_short)ERROR);
        tp->th_code = htons((u_short)error);
        for (pe = errmsgs; pe->e_code >= 0; pe++)
                if (pe->e_code == error)
                        break;
        if (pe->e_code < 0) {
                pe->e_msg = strerror(error - 100);
                tp->th_code = EUNDEF;   /* set 'undef' errorcode */
        }
        strcpy(tp->th_msg, pe->e_msg);
        length = strlen(pe->e_msg);
        tp->th_msg[length] = '\0';
        length += 5;
        if (send(peer, buf, length, 0) != length)
                syslog(LOG_ERR, "nak: %m\n");
}
```

Notice for package(s)

m4

```
The files in this directory provide example uses of GNU M4.
The following copyright notice applies to each of these
description files.

Copyright (C) 2006, 2010-2012 Free Software Foundation, Inc.
This file is free software; the Free Software Foundation
gives unlimited permission to copy and/or distribute it,
with or without modifications, as long as this notice is preserved.
```

Notice for package(s)

flex

```
Flex carries the copyright used for BSD software, slightly modified
because it originated at the Lawrence Berkeley (not Livermore!) Laboratory,
which operates under a contract with the Department of Energy:

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007 The Flex Project.

Copyright (c) 1990, 1997 The Regents of the University of California.
All rights reserved.

This code is derived from software contributed to Berkeley by
Vern Paxson.

The United States Government has rights in this work pursuant
to contract no. DE-AC03-76SF00098 between the United States
Department of Energy and the University of California.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.

This basically says "do whatever you please with this software except
remove this notice or take advantage of the University's (or the flex
authors') name".

Note that the "flex.skl" scanner skeleton carries no copyright notice.
You are free to do whatever you please with scanners generated using flex;
for them, you are not even bound by the above copyright.
```

Notice for package(s)

bc

---

encourage the widest possible use of a certain library, so that it
becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

  Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.
^L
                  GNU LESSER GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control
compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

  2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices
    stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no
    charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application. Therefore, Subsection 2d requires that any

application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.
^L
Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
^L
6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one

of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked
    with the Library, with the complete machine-readable "work that
    uses the Library", as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library. (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these
    materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

  It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.
^L
  7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work
    based on the Library, uncombined with any other library
    facilities. This must be distributed under the terms of the
    Sections above.

    b) Give prominent notice with the combined library of the fact
    that part of it is a work based on the Library, and explaining
    where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.
^L
  11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent

license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.
^L
14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
^L
How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

```
    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307  USA

Also add information on how to contact you by electronic and paper
mail.

You should also get your employer (if you work as a programmer) or
your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James
Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

bc

```
/* bcdefs.h:  The single file to include all constants and type definitions. */

/*  This file is part of GNU bc.
    Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation, Inc.

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License , or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; see the file COPYING.  If not, write to
      The Free Software Foundation, Inc.
      59 Temple Place, Suite 330
      Boston, MA 02111 USA

    You may contact the author by:
       e-mail:  philnelson@acm.org
      us-mail:  Philip A. Nelson
                Computer Science Department, 9062
                Western Washington University
                Bellingham, WA 98226-9062

*************************************************************************/

/* Include the configuration file. */
#include "config.h"

/* Standard includes for all files. */
#include <stdio.h>
#include <sys/types.h>
#include <ctype.h>
#ifdef HAVE_STRINGS_H
#include <strings.h>
#else
#include <string.h>
#endif
#ifdef HAVE_LIMITS_H
#include <limits.h>
#endif

#if defined(LIBEDIT)
#include <histedit.h>
#endif

#if defined(READLINE)
#include <readline/readline.h>
#include <readline/history.h>
#endif
```

```
/* Include the other definitions. */
#include "const.h"
#include "number.h"

/* These definitions define all the structures used in
   code and data storage.  This includes the representation of
   labels.   The "guiding" principle is to make structures that
   take a minimum of space when unused but can be built to contain
   the full structures.  */

/* Labels are first.  Labels are generated sequentially in functions
   and full code.  They just "point" to a single bye in the code.  The
   "address" is the byte number.  The byte number is used to get an
   actual character pointer. */

typedef struct bc_label_group
    {
      long l_adrs [ BC_LABEL_GROUP ];
      struct bc_label_group *l_next;
    } bc_label_group;

/* Argument list.  Recorded in the function so arguments can
   be checked at call time. */

typedef struct arg_list
    {
      int av_name;
      int arg_is_var;           /* Extension ... variable parameters. */
      struct arg_list *next;
    } arg_list;

/* Each function has its own code segments and labels.  There can be
   no jumps between functions so labels are unique to a function. */

typedef struct
    {
      char f_defined;   /* Is this function defined yet. */
      char *f_body;
      int  f_body_size;  /* Size of body.  Power of 2. */
      int  f_code_size;
      bc_label_group *f_label;
      arg_list *f_params;
      arg_list *f_autos;
    } bc_function;

/* Code addresses. */
typedef struct {
      int pc_func;
      int pc_addr;
    } program_counter;


/* Variables are "pushable" (auto) and thus we need a stack mechanism.
   This is built into the variable record. */

typedef struct bc_var
    {
      bc_num v_value;
      struct bc_var *v_next;
    }  bc_var;


/* bc arrays can also be "auto" variables and thus need the same
   kind of stacking mechanisms. */

typedef struct bc_array_node
    {
      union
        {
          bc_num n_num [NODE_SIZE];
          struct bc_array_node *n_down [NODE_SIZE];
        } n_items;
    } bc_array_node;

typedef struct bc_array
    {
      bc_array_node *a_tree;
      short a_depth;
    } bc_array;

typedef struct bc_var_array
    {
      bc_array *a_value;
      char      a_param;
      struct bc_var_array *a_next;
    } bc_var_array;


/* For the stacks, execution and function, we need records to allow
   for arbitrary size. */

typedef struct estack_rec {
        bc_num s_num;
        struct estack_rec *s_next;
} estack_rec;

typedef struct fstack_rec {
```

```
        int  s_val;
        struct fstack_rec *s_next;
} fstack_rec;


/* The following are for the name tree. */

typedef struct id_rec {
        char  *id;      /* The program name. */
                        /* A name == 0 => nothing assigned yet. */
        int   a_name;   /* The array variable name (number). */
        int   f_name;   /* The function name (number).  */
        int   v_name;   /* The variable name (number).  */
        short balance;  /* For the balanced tree. */
        struct id_rec *left, *right; /* Tree pointers. */
} id_rec;


/* A list of files to process. */

typedef struct file_node {
        char *name;
        struct file_node *next;
} file_node;

/* Macro Definitions */

#if defined(LIBEDIT)
#define HISTORY_SIZE(n) history(hist, &histev, H_SETSIZE, n)
#define UNLIMIT_HISTORY history(hist, &histev, H_SETSIZE, INT_MAX)
#endif

#if defined(READLINE)
#define HISTORY_SIZE(n) stifle_history(n)
#define UNLIMIT_HISTORY unstifle_history()
#endif
```

Notice for package(s)

bc

```
/*
 * Header file for dc routines
 *
 * Copyright (C) 1994, 1997, 1998 Free Software Foundation, Inc.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2, or (at your option)
 * any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, you can either send email to this
 * program's author (see below) or write to:
 *
 *    The Free Software Foundation, Inc.
 *    59 Temple Place, Suite 330
 *    Boston, MA 02111 USA
 */

#ifndef DC_DEFS_H
#define DC_DEFS_H

/* 'I' is a command, and bases 17 and 18 are quite
 * unusual, so we limit ourselves to bases 2 to 16
 */
#define DC_IBASE_MAX    16

#define DC_SUCCESS              0
#define DC_DOMAIN_ERROR 1
#define DC_FAIL                 2       /* generic failure */


#ifndef __STDC__
# define DC_PROTO(x)                    ()
# define DC_DECLVOID()                  ()
# define DC_DECLARG(arglist)    arglist
# define DC_DECLSEP                             ;
# define DC_DECLEND                             ;
#else /* __STDC__ */
# define DC_PROTO(x)                    x
# define DC_DECLVOID()                  (void)
# define DC_DECLARG(arglist)    (
# define DC_DECLSEP                             ,
# define DC_DECLEND                             )
#endif /* __STDC__ */
```

```
typedef enum {DC_TOSS, DC_KEEP}   dc_discard;
typedef enum {DC_NONL, DC_WITHNL} dc_newline;


/* type discriminant for dc_data */
typedef enum {DC_UNINITIALIZED, DC_NUMBER, DC_STRING} dc_value_type;

/* only numeric.c knows what dc_num's *really* look like */
typedef struct dc_number *dc_num;

/* only string.c knows what dc_str's *really* look like */
typedef struct dc_string *dc_str;


/* except for the two implementation-specific modules, all
 * dc functions only know of this one generic type of object
 */
typedef struct {
        dc_value_type dc_type;  /* discriminant for union */
        union {
                dc_num number;
                dc_str string;
        } v;
} dc_data;


/* This is dc's only global variable: */
extern const char *progname;    /* basename of program invocation */

#endif /* not DC_DEFS_H */
```

Notice for package(s)

bc

```
/* number.c: Implements arbitrary precision numbers. */
/*
    Copyright (C) 1991, 1992, 1993, 1994, 1997, 2000 Free Software Foundation, Inc.

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License , or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; see the file COPYING.  If not, write to:

      The Free Software Foundation, Inc.
      59 Temple Place, Suite 330
      Boston, MA 02111-1307 USA.


    You may contact the author by:
       e-mail:  philnelson@acm.org
      us-mail:  Philip A. Nelson
                Computer Science Department, 9062
                Western Washington University
                Bellingham, WA 98226-9062

************************************************************************/

#include <stdio.h>
#include <config.h>
#include <number.h>
#include <assert.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>/* Prototypes needed for external utility routines. */

#define bc_rt_warn rt_warn
#define bc_rt_error rt_error
#define bc_out_of_memory out_of_memory

_PROTOTYPE(void rt_warn, (char *mesg ,...));
_PROTOTYPE(void rt_error, (char *mesg ,...));
_PROTOTYPE(void out_of_memory, (void));

/* Storage used for special numbers. */
bc_num _zero_;
bc_num _one_;
bc_num _two_;

static bc_num _bc_Free_list = NULL;
```

```
/* new_num allocates a number and sets fields to known values. */

bc_num
bc_new_num (length, scale)
     int length, scale;
{
  bc_num temp;

  if (_bc_Free_list != NULL) {
    temp = _bc_Free_list;
    _bc_Free_list = temp->n_next;
  } else {
    temp = (bc_num) malloc (sizeof(bc_struct));
    if (temp == NULL) bc_out_of_memory ();
  }
  temp->n_sign = PLUS;
  temp->n_len = length;
  temp->n_scale = scale;
  temp->n_refs = 1;
  temp->n_ptr = (char *) malloc (length+scale);
  if (temp->n_ptr == NULL) bc_out_of_memory();
  temp->n_value = temp->n_ptr;
  memset (temp->n_ptr, 0, length+scale);
  return temp;
}

/* "Frees" a bc_num NUM.  Actually decreases reference count and only
   frees the storage if reference count is zero. */

void
bc_free_num (num)
    bc_num *num;
{
  if (*num == NULL) return;
  (*num)->n_refs--;
  if ((*num)->n_refs == 0) {
    if ((*num)->n_ptr)
      free ((*num)->n_ptr);
    (*num)->n_next = _bc_Free_list;
    _bc_Free_list = *num;
  }
  *num = NULL;
}


/* Intitialize the number package! */

void
bc_init_numbers ()
{
  _zero_ = bc_new_num (1,0);
  _one_  = bc_new_num (1,0);
  _one_->n_value[0] = 1;
  _two_  = bc_new_num (1,0);
  _two_->n_value[0] = 2;
}


/* Make a copy of a number!  Just increments the reference count! */

bc_num
bc_copy_num (num)
     bc_num num;
{
  num->n_refs++;
  return num;
}


/* Initialize a number NUM by making it a copy of zero. */

void
bc_init_num (num)
     bc_num *num;
{
  *num = bc_copy_num (_zero_);
}

/* For many things, we may have leading zeros in a number NUM.
   _bc_rm_leading_zeros just moves the data "value" pointer to the
   correct place and adjusts the length. */

static void
_bc_rm_leading_zeros (num)
     bc_num num;
{
  /* We can move n_value to point to the first non zero digit! */
  while (*num->n_value == 0 && num->n_len > 1) {
    num->n_value++;
    num->n_len--;
  }
}


/* Compare two bc numbers.  Return value is 0 if equal, -1 if N1 is less
   than N2 and +1 if N1 is greater than N2.  If USE_SIGN is false, just
   compare the magnitudes. */
```

```
static int
_bc_do_compare (n1, n2, use_sign, ignore_last)
     bc_num n1, n2;
     int use_sign;
     int ignore_last;
{
  char *n1ptr, *n2ptr;
  int  count;

  /* First, compare signs. */
  if (use_sign && n1->n_sign != n2->n_sign)
    {
      if (n1->n_sign == PLUS)
        return (1);     /* Positive N1 > Negative N2 */
      else
        return (-1);    /* Negative N1 < Positive N1 */
    }

  /* Now compare the magnitude. */
  if (n1->n_len != n2->n_len)
    {
      if (n1->n_len > n2->n_len)
        {
          /* Magnitude of n1 > n2. */
          if (!use_sign || n1->n_sign == PLUS)
            return (1);
          else
            return (-1);
        }
      else
        {
          /* Magnitude of n1 < n2. */
          if (!use_sign || n1->n_sign == PLUS)
            return (-1);
          else
            return (1);
        }
    }

  /* If we get here, they have the same number of integer digits.
     check the integer part and the equal length part of the fraction. */
  count = n1->n_len + MIN (n1->n_scale, n2->n_scale);
  n1ptr = n1->n_value;
  n2ptr = n2->n_value;

  while ((count > 0) && (*n1ptr == *n2ptr))
    {
      n1ptr++;
      n2ptr++;
      count--;
    }
  if (ignore_last && count == 1 && n1->n_scale == n2->n_scale)
    return (0);
  if (count != 0)
    {
      if (*n1ptr > *n2ptr)
        {
          /* Magnitude of n1 > n2. */
          if (!use_sign || n1->n_sign == PLUS)
            return (1);
          else
            return (-1);
        }
      else
        {
          /* Magnitude of n1 < n2. */
          if (!use_sign || n1->n_sign == PLUS)
            return (-1);
          else
            return (1);
        }
    }

  /* They are equal up to the last part of the equal part of the fraction. */
  if (n1->n_scale != n2->n_scale)
    {
      if (n1->n_scale > n2->n_scale)
        {
          for (count = n1->n_scale-n2->n_scale; count>0; count--)
            if (*n1ptr++ != 0)
              {
                /* Magnitude of n1 > n2. */
                if (!use_sign || n1->n_sign == PLUS)
                  return (1);
                else
                  return (-1);
              }
        }
      else
        {
          for (count = n2->n_scale-n1->n_scale; count>0; count--)
            if (*n2ptr++ != 0)
              {
                /* Magnitude of n1 < n2. */
                if (!use_sign || n1->n_sign == PLUS)
                  return (-1);
                else
                  return (1);
```

```
            }
        }
    }

  /* They must be equal! */
  return (0);
}


/* This is the "user callable" routine to compare numbers N1 and N2. */

int
bc_compare (n1, n2)
     bc_num n1, n2;
{
  return _bc_do_compare (n1, n2, TRUE, FALSE);
}

/* In some places we need to check if the number is negative. */

char
bc_is_neg (num)
     bc_num num;
{
  return num->n_sign == MINUS;
}

/* In some places we need to check if the number NUM is zero. */

char
bc_is_zero (num)
     bc_num num;
{
  int  count;
  char *nptr;

  /* Quick check. */
  if (num == _zero_) return TRUE;

  /* Initialize */
  count = num->n_len + num->n_scale;
  nptr = num->n_value;

  /* The check */
  while ((count > 0) && (*nptr++ == 0)) count--;

  if (count != 0)
    return FALSE;
  else
    return TRUE;
}

/* In some places we need to check if the number NUM is almost zero.
   Specifically, all but the last digit is 0 and the last digit is 1.
   Last digit is defined by scale. */

char
bc_is_near_zero (num, scale)
     bc_num num;
     int scale;
{
  int  count;
  char *nptr;

  /* Error checking */
  if (scale > num->n_scale)
    scale = num->n_scale;

  /* Initialize */
  count = num->n_len + scale;
  nptr = num->n_value;

  /* The check */
  while ((count > 0) && (*nptr++ == 0)) count--;

  if (count != 0 && (count != 1 || *--nptr != 1))
    return FALSE;
  else
    return TRUE;
}


/* Perform addition: N1 is added to N2 and the value is
   returned.  The signs of N1 and N2 are ignored.
   SCALE_MIN is to set the minimum scale of the result. */

static bc_num
_bc_do_add (n1, n2, scale_min)
     bc_num n1, n2;
     int scale_min;
{
  bc_num sum;
  int sum_scale, sum_digits;
  char *n1ptr, *n2ptr, *sumptr;
  int carry, n1bytes, n2bytes;
  int count;

  /* Prepare sum. */
```

```
  sum_scale = MAX (n1->n_scale, n2->n_scale);
  sum_digits = MAX (n1->n_len, n2->n_len) + 1;
  sum = bc_new_num (sum_digits, MAX(sum_scale, scale_min));

  /* Zero extra digits made by scale_min. */
  if (scale_min > sum_scale)
    {
      sumptr = (char *) (sum->n_value + sum_scale + sum_digits);
      for (count = scale_min - sum_scale; count > 0; count--)
        *sumptr++ = 0;
    }

  /* Start with the fraction part.  Initialize the pointers. */
  n1bytes = n1->n_scale;
  n2bytes = n2->n_scale;
  n1ptr = (char *) (n1->n_value + n1->n_len + n1bytes - 1);
  n2ptr = (char *) (n2->n_value + n2->n_len + n2bytes - 1);
  sumptr = (char *) (sum->n_value + sum_scale + sum_digits - 1);

  /* Add the fraction part.  First copy the longer fraction.*/
  if (n1bytes != n2bytes)
    {
      if (n1bytes > n2bytes)
        while (n1bytes>n2bytes)
          { *sumptr-- = *n1ptr--; n1bytes--;}
      else
        while (n2bytes>n1bytes)
          { *sumptr-- = *n2ptr--; n2bytes--;}
    }

  /* Now add the remaining fraction part and equal size integer parts. */
  n1bytes += n1->n_len;
  n2bytes += n2->n_len;
  carry = 0;
  while ((n1bytes > 0) && (n2bytes > 0))
    {
      *sumptr = *n1ptr-- + *n2ptr-- + carry;
      if (*sumptr > (BASE-1))
        {
           carry = 1;
           *sumptr -= BASE;
        }
      else
        carry = 0;
      sumptr--;
      n1bytes--;
      n2bytes--;
    }

  /* Now add carry the longer integer part. */
  if (n1bytes == 0)
    { n1bytes = n2bytes; n1ptr = n2ptr; }
  while (n1bytes-- > 0)
    {
      *sumptr = *n1ptr-- + carry;
      if (*sumptr > (BASE-1))
        {
           carry = 1;
           *sumptr -= BASE;
         }
      else
        carry = 0;
      sumptr--;
    }

  /* Set final carry. */
  if (carry == 1)
    *sumptr += 1;

  /* Adjust sum and return. */
  _bc_rm_leading_zeros (sum);
  return sum;
}


/* Perform subtraction: N2 is subtracted from N1 and the value is
   returned.  The signs of N1 and N2 are ignored.  Also, N1 is
   assumed to be larger than N2.  SCALE_MIN is the minimum scale
   of the result. */

static bc_num
_bc_do_sub (n1, n2, scale_min)
     bc_num n1, n2;
     int scale_min;
{
  bc_num diff;
  int diff_scale, diff_len;
  int min_scale, min_len;
  char *n1ptr, *n2ptr, *diffptr;
  int borrow, count, val;

  /* Allocate temporary storage. */
  diff_len = MAX (n1->n_len, n2->n_len);
  diff_scale = MAX (n1->n_scale, n2->n_scale);
  min_len = MIN  (n1->n_len, n2->n_len);
  min_scale = MIN (n1->n_scale, n2->n_scale);
  diff = bc_new_num (diff_len, MAX(diff_scale, scale_min));
```

```
  /* Zero extra digits made by scale_min. */
  if (scale_min > diff_scale)
    {
      diffptr = (char *) (diff->n_value + diff_len + diff_scale);
      for (count = scale_min - diff_scale; count > 0; count--)
        *diffptr++ = 0;
    }

  /* Initialize the subtract. */
  n1ptr = (char *) (n1->n_value + n1->n_len + n1->n_scale -1);
  n2ptr = (char *) (n2->n_value + n2->n_len + n2->n_scale -1);
  diffptr = (char *) (diff->n_value + diff_len + diff_scale -1);

  /* Subtract the numbers. */
  borrow = 0;

  /* Take care of the longer scaled number. */
  if (n1->n_scale != min_scale)
    {
      /* n1 has the longer scale */
      for (count = n1->n_scale - min_scale; count > 0; count--)
        *diffptr-- = *n1ptr--;
    }
  else
    {
      /* n2 has the longer scale */
      for (count = n2->n_scale - min_scale; count > 0; count--)
        {
          val = - *n2ptr-- - borrow;
          if (val < 0)
            {
              val += BASE;
              borrow = 1;
            }
          else
            borrow = 0;
          *diffptr-- = val;
        }
    }

  /* Now do the equal length scale and integer parts. */

  for (count = 0; count < min_len + min_scale; count++)
    {
      val = *n1ptr-- - *n2ptr-- - borrow;
      if (val < 0)
        {
          val += BASE;
          borrow = 1;
        }
      else
        borrow = 0;
      *diffptr-- = val;
    }

  /* If n1 has more digits then n2, we now do that subtract. */
  if (diff_len != min_len)
    {
      for (count = diff_len - min_len; count > 0; count--)
        {
          val = *n1ptr-- - borrow;
          if (val < 0)
            {
              val += BASE;
              borrow = 1;
            }
          else
            borrow = 0;
          *diffptr-- = val;
        }
    }

  /* Clean up and return. */
  _bc_rm_leading_zeros (diff);
  return diff;
}


/* Here is the full subtract routine that takes care of negative numbers.
   N2 is subtracted from N1 and the result placed in RESULT.  SCALE_MIN
   is the minimum scale for the result. */

void
bc_sub (n1, n2, result, scale_min)
     bc_num n1, n2, *result;
     int scale_min;
{
  bc_num diff = NULL;
  int cmp_res;
  int res_scale;

  if (n1->n_sign != n2->n_sign)
    {
      diff = _bc_do_add (n1, n2, scale_min);
      diff->n_sign = n1->n_sign;
    }
  else
    {
```

```
      /* subtraction must be done. */
      /* Compare magnitudes. */
      cmp_res = _bc_do_compare (n1, n2, FALSE, FALSE);
      switch (cmp_res)
        {
        case -1:
          /* n1 is less than n2, subtract n1 from n2. */
          diff = _bc_do_sub (n2, n1, scale_min);
          diff->n_sign = (n2->n_sign == PLUS ? MINUS : PLUS);
          break;
        case  0:
          /* They are equal! return zero! */
          res_scale = MAX (scale_min, MAX(n1->n_scale, n2->n_scale));
          diff = bc_new_num (1, res_scale);
          memset (diff->n_value, 0, res_scale+1);
          break;
        case  1:
          /* n2 is less than n1, subtract n2 from n1. */
          diff = _bc_do_sub (n1, n2, scale_min);
          diff->n_sign = n1->n_sign;
          break;
        }
    }

  /* Clean up and return. */
  bc_free_num (result);
  *result = diff;
}


/* Here is the full add routine that takes care of negative numbers.
   N1 is added to N2 and the result placed into RESULT.  SCALE_MIN
   is the minimum scale for the result. */

void
bc_add (n1, n2, result, scale_min)
     bc_num n1, n2, *result;
     int scale_min;
{
  bc_num sum = NULL;
  int cmp_res;
  int res_scale;

  if (n1->n_sign == n2->n_sign)
    {
      sum = _bc_do_add (n1, n2, scale_min);
      sum->n_sign = n1->n_sign;
    }
  else
    {
      /* subtraction must be done. */
      cmp_res = _bc_do_compare (n1, n2, FALSE, FALSE);  /* Compare magnitudes. */
      switch (cmp_res)
        {
        case -1:
          /* n1 is less than n2, subtract n1 from n2. */
          sum = _bc_do_sub (n2, n1, scale_min);
          sum->n_sign = n2->n_sign;
          break;
        case  0:
          /* They are equal! return zero with the correct scale! */
          res_scale = MAX (scale_min, MAX(n1->n_scale, n2->n_scale));
          sum = bc_new_num (1, res_scale);
          memset (sum->n_value, 0, res_scale+1);
          break;
        case  1:
          /* n2 is less than n1, subtract n2 from n1. */
          sum = _bc_do_sub (n1, n2, scale_min);
          sum->n_sign = n1->n_sign;
        }
    }

  /* Clean up and return. */
  bc_free_num (result);
  *result = sum;
}

/* Recursive vs non-recursive multiply crossover ranges. */
#if defined(MULDIGITS)
#include "muldigits.h"
#else
#define MUL_BASE_DIGITS 80
#endif

int mul_base_digits = MUL_BASE_DIGITS;
#define MUL_SMALL_DIGITS mul_base_digits/4

/* Multiply utility routines */

static bc_num
new_sub_num (length, scale, value)
     int length, scale;
     char *value;
{
  bc_num temp;

  if (_bc_Free_list != NULL) {
    temp = _bc_Free_list;
```

```
    _bc_Free_list = temp->n_next;
  } else {
    temp = (bc_num) malloc (sizeof(bc_struct));
    if (temp == NULL) bc_out_of_memory ();
  }
  temp->n_sign = PLUS;
  temp->n_len = length;
  temp->n_scale = scale;
  temp->n_refs = 1;
  temp->n_ptr = NULL;
  temp->n_value = value;
  return temp;
}

static void
_bc_simp_mul (bc_num n1, int n1len, bc_num n2, int n2len, bc_num *prod,
              int full_scale)
{
  char *n1ptr, *n2ptr, *pvptr;
  char *n1end, *n2end;          /* To the end of n1 and n2. */
  int indx, sum, prodlen;

  prodlen = n1len+n2len+1;

  *prod = bc_new_num (prodlen, 0);

  n1end = (char *) (n1->n_value + n1len - 1);
  n2end = (char *) (n2->n_value + n2len - 1);
  pvptr = (char *) ((*prod)->n_value + prodlen - 1);
  sum = 0;

  /* Here is the loop... */
  for (indx = 0; indx < prodlen-1; indx++)
    {
      n1ptr = (char *) (n1end - MAX(0, indx-n2len+1));
      n2ptr = (char *) (n2end - MIN(indx, n2len-1));
      while ((n1ptr >= n1->n_value) && (n2ptr <= n2end))
        sum += *n1ptr-- * *n2ptr++;
      *pvptr-- = sum % BASE;
      sum = sum / BASE;
    }
  *pvptr = sum;
}


/* A special adder/subtractor for the recursive divide and conquer
   multiply algorithm.  Note: if sub is called, accum must
   be larger that what is being subtracted.  Also, accum and val
   must have n_scale = 0.  (e.g. they must look like integers. *) */
static void
_bc_shift_addsub (bc_num accum, bc_num val, int shift, int sub)
{
  signed char *accp, *valp;
  int  count, carry;

  count = val->n_len;
  if (val->n_value[0] == 0)
    count--;
  assert (accum->n_len+accum->n_scale >= shift+count);

  /* Set up pointers and others */
  accp = (signed char *)(accum->n_value +
                         accum->n_len + accum->n_scale - shift - 1);
  valp = (signed char *)(val->n_value + val->n_len - 1);
  carry = 0;

  if (sub) {
    /* Subtraction, carry is really borrow. */
    while (count--) {
      *accp -= *valp-- + carry;
      if (*accp < 0) {
        carry = 1;
        *accp-- += BASE;
      } else {
        carry = 0;
        accp--;
      }
    }
    while (carry) {
      *accp -= carry;
      if (*accp < 0)
        *accp-- += BASE;
      else
        carry = 0;
    }
  } else {
    /* Addition */
    while (count--) {
      *accp += *valp-- + carry;
      if (*accp > (BASE-1)) {
        carry = 1;
        *accp-- -= BASE;
      } else {
        carry = 0;
        accp--;
      }
    }
    while (carry) {
```

```
      *accp += carry;
      if (*accp > (BASE-1))
        *accp-- -= BASE;
      else
        carry = 0;
    }
  }
}

/* Recursive divide and conquer multiply algorithm.
   Based on
   Let u = u0 + u1*(b^n)
   Let v = v0 + v1*(b^n)
   Then uv = (B^2n+B^n)*u1*v1 + B^n*(u1-u0)*(v0-v1) + (B^n+1)*u0*v0

   B is the base of storage, number of digits in u1,u0 close to equal.
*/
static void
_bc_rec_mul (bc_num u, int ulen, bc_num v, int vlen, bc_num *prod,
             int full_scale)
{
  bc_num u0, u1, v0, v1;
  int u0len, v0len;
  bc_num m1, m2, m3, d1, d2;
  int n, prodlen, m1zero;
  int d1len, d2len;

  /* Base case? */
  if ((ulen+vlen) < mul_base_digits
      || ulen < MUL_SMALL_DIGITS
      || vlen < MUL_SMALL_DIGITS ) {
    _bc_simp_mul (u, ulen, v, vlen, prod, full_scale);
    return;
  }

  /* Calculate n -- the u and v split point in digits. */
  n = (MAX(ulen, vlen)+1) / 2;

  /* Split u and v. */
  if (ulen < n) {
    u1 = bc_copy_num (_zero_);
    u0 = new_sub_num (ulen,0, u->n_value);
  } else {
    u1 = new_sub_num (ulen-n, 0, u->n_value);
    u0 = new_sub_num (n, 0, u->n_value+ulen-n);
  }
  if (vlen < n) {
    v1 = bc_copy_num (_zero_);
    v0 = new_sub_num (vlen,0, v->n_value);
  } else {
    v1 = new_sub_num (vlen-n, 0, v->n_value);
    v0 = new_sub_num (n, 0, v->n_value+vlen-n);
    }
  _bc_rm_leading_zeros (u1);
  _bc_rm_leading_zeros (u0);
  u0len = u0->n_len;
  _bc_rm_leading_zeros (v1);
  _bc_rm_leading_zeros (v0);
  v0len = v0->n_len;

  m1zero = bc_is_zero(u1) || bc_is_zero(v1);

  /* Calculate sub results ... */

  bc_init_num(&d1);
  bc_init_num(&d2);
  bc_sub (u1, u0, &d1, 0);
  d1len = d1->n_len;
  bc_sub (v0, v1, &d2, 0);
  d2len = d2->n_len;


  /* Do recursive multiplies and shifted adds. */
  if (m1zero)
    m1 = bc_copy_num (_zero_);
  else
    _bc_rec_mul (u1, u1->n_len, v1, v1->n_len, &m1, 0);

  if (bc_is_zero(d1) || bc_is_zero(d2))
    m2 = bc_copy_num (_zero_);
  else
    _bc_rec_mul (d1, d1len, d2, d2len, &m2, 0);

  if (bc_is_zero(u0) || bc_is_zero(v0))
    m3 = bc_copy_num (_zero_);
  else
    _bc_rec_mul (u0, u0->n_len, v0, v0->n_len, &m3, 0);

  /* Initialize product */
  prodlen = ulen+vlen+1;
  *prod = bc_new_num(prodlen, 0);

  if (!m1zero) {
    _bc_shift_addsub (*prod, m1, 2*n, 0);
    _bc_shift_addsub (*prod, m1, n, 0);
  }
  _bc_shift_addsub (*prod, m3, n, 0);
  _bc_shift_addsub (*prod, m3, 0, 0);
```

```
  _bc_shift_addsub (*prod, m2, n, d1->n_sign != d2->n_sign);

  /* Now clean up! */
  bc_free_num (&u1);
  bc_free_num (&u0);
  bc_free_num (&v1);
  bc_free_num (&m1);
  bc_free_num (&v0);
  bc_free_num (&m2);
  bc_free_num (&m3);
  bc_free_num (&d1);
  bc_free_num (&d2);
}

/* The multiply routine.  N2 times N1 is put int PROD with the scale of
   the result being MIN(N2 scale+N1 scale, MAX (SCALE, N2 scale, N1 scale)).
   */

void
bc_multiply (n1, n2, prod, scale)
     bc_num n1, n2, *prod;
     int scale;
{
  bc_num pval;
  int len1, len2;
  int full_scale, prod_scale;

  /* Initialize things. */
  len1 = n1->n_len + n1->n_scale;
  len2 = n2->n_len + n2->n_scale;
  full_scale = n1->n_scale + n2->n_scale;
  prod_scale = MIN(full_scale,MAX(scale,MAX(n1->n_scale,n2->n_scale)));

  /* Do the multiply */
  _bc_rec_mul (n1, len1, n2, len2, &pval, full_scale);

  /* Assign to prod and clean up the number. */
  pval->n_sign = ( n1->n_sign == n2->n_sign ? PLUS : MINUS );
  pval->n_value = pval->n_ptr;
  pval->n_len = len2 + len1 + 1 - full_scale;
  pval->n_scale = prod_scale;
  _bc_rm_leading_zeros (pval);
  if (bc_is_zero (pval))
    pval->n_sign = PLUS;
  bc_free_num (prod);
  *prod = pval;
}

/* Some utility routines for the divide:  First a one digit multiply.
   NUM (with SIZE digits) is multiplied by DIGIT and the result is
   placed into RESULT.  It is written so that NUM and RESULT can be
   the same pointers.  */

static void
_one_mult (num, size, digit, result)
     unsigned char *num;
     int size, digit;
     unsigned char *result;
{
  int carry, value;
  unsigned char *nptr, *rptr;

  if (digit == 0)
    memset (result, 0, size);
  else
    {
      if (digit == 1)
        memcpy (result, num, size);
      else
        {
          /* Initialize */
          nptr = (unsigned char *) (num+size-1);
          rptr = (unsigned char *) (result+size-1);
          carry = 0;

          while (size-- > 0)
            {
              value = *nptr-- * digit + carry;
              *rptr-- = value % BASE;
              carry = value / BASE;
            }

          if (carry != 0) *rptr = carry;
        }
    }
}


/* The full division routine. This computes N1 / N2.  It returns
   0 if the division is ok and the result is in QUOT.  The number of
   digits after the decimal point is SCALE. It returns -1 if division
   by zero is tried.  The algorithm is found in Knuth Vol 2. p237. */

int
bc_divide (n1, n2, quot, scale)
     bc_num n1, n2, *quot;
     int scale;
{
```

```
bc_num qval;
unsigned char *num1, *num2;
unsigned char *ptr1, *ptr2, *n2ptr, *qptr;
int  scale1, val;
unsigned int  len1, len2, scale2, qdigits, extra, count;
unsigned int  qdig, qguess, borrow, carry;
unsigned char *mval;
char zero;
unsigned int  norm;

/* Test for divide by zero. */
if (bc_is_zero (n2)) return -1;

/* Test for divide by 1.  If it is we must truncate. */
if (n2->n_scale == 0)
  {
    if (n2->n_len == 1 && *n2->n_value == 1)
      {
        qval = bc_new_num (n1->n_len, scale);
        qval->n_sign = (n1->n_sign == n2->n_sign ? PLUS : MINUS);
        memset (&qval->n_value[n1->n_len],0,scale);
        memcpy (qval->n_value, n1->n_value,
                n1->n_len + MIN(n1->n_scale,scale));
        bc_free_num (quot);
        *quot = qval;
      }
  }

/* Set up the divide.  Move the decimal point on n1 by n2's scale.
   Remember, zeros on the end of num2 are wasted effort for dividing. */
scale2 = n2->n_scale;
n2ptr = (unsigned char *) n2->n_value+n2->n_len+scale2-1;
while ((scale2 > 0) && (*n2ptr-- == 0)) scale2--;

len1 = n1->n_len + scale2;
scale1 = n1->n_scale - scale2;
if (scale1 < scale)
  extra = scale - scale1;
else
  extra = 0;
num1 = (unsigned char *) malloc (n1->n_len+n1->n_scale+extra+2);
if (num1 == NULL) bc_out_of_memory();
memset (num1, 0, n1->n_len+n1->n_scale+extra+2);
memcpy (num1+1, n1->n_value, n1->n_len+n1->n_scale);

len2 = n2->n_len + scale2;
num2 = (unsigned char *) malloc (len2+1);
if (num2 == NULL) bc_out_of_memory();
memcpy (num2, n2->n_value, len2);
*(num2+len2) = 0;
n2ptr = num2;
while (*n2ptr == 0)
  {
    n2ptr++;
    len2--;
  }

/* Calculate the number of quotient digits. */
if (len2 > len1+scale)
  {
    qdigits = scale+1;
    zero = TRUE;
  }
else
  {
    zero = FALSE;
    if (len2>len1)
      qdigits = scale+1;      /* One for the zero integer part. */
    else
      qdigits = len1-len2+scale+1;
  }

/* Allocate and zero the storage for the quotient. */
qval = bc_new_num (qdigits-scale,scale);
memset (qval->n_value, 0, qdigits);

/* Allocate storage for the temporary storage mval. */
mval = (unsigned char *) malloc (len2+1);
if (mval == NULL) bc_out_of_memory ();

/* Now for the full divide algorithm. */
if (!zero)
  {
    /* Normalize */
    norm =  10 / ((int)*n2ptr + 1);
    if (norm != 1)
      {
        _one_mult (num1, len1+scale1+extra+1, norm, num1);
        _one_mult (n2ptr, len2, norm, n2ptr);
      }

    /* Initialize divide loop. */
    qdig = 0;
    if (len2 > len1)
      qptr = (unsigned char *) qval->n_value+len2-len1;
    else
      qptr = (unsigned char *) qval->n_value;
```

```
      /* Loop */
      while (qdig <= len1+scale-len2)
        {
          /* Calculate the quotient digit guess. */
          if (*n2ptr == num1[qdig])
            qguess = 9;
          else
            qguess = (num1[qdig]*10 + num1[qdig+1]) / *n2ptr;

          /* Test qguess. */
          if (n2ptr[1]*qguess >
              (num1[qdig]*10 + num1[qdig+1] - *n2ptr*qguess)*10
               + num1[qdig+2])
            {
              qguess--;
              /* And again. */
              if (n2ptr[1]*qguess >
                  (num1[qdig]*10 + num1[qdig+1] - *n2ptr*qguess)*10
                  + num1[qdig+2])
                qguess--;
            }

          /* Multiply and subtract. */
          borrow = 0;
          if (qguess != 0)
            {
              *mval = 0;
              _one_mult (n2ptr, len2, qguess, mval+1);
              ptr1 = (unsigned char *) num1+qdig+len2;
              ptr2 = (unsigned char *) mval+len2;
              for (count = 0; count < len2+1; count++)
                {
                  val = (int) *ptr1 - (int) *ptr2-- - borrow;
                  if (val < 0)
                    {
                      val += 10;
                      borrow = 1;
                    }
                  else
                    borrow = 0;
                  *ptr1-- = val;
                }
            }

          /* Test for negative result. */
          if (borrow == 1)
            {
              qguess--;
              ptr1 = (unsigned char *) num1+qdig+len2;
              ptr2 = (unsigned char *) n2ptr+len2-1;
              carry = 0;
              for (count = 0; count < len2; count++)
                {
                  val = (int) *ptr1 + (int) *ptr2-- + carry;
                  if (val > 9)
                    {
                      val -= 10;
                      carry = 1;
                    }
                  else
                    carry = 0;
                  *ptr1-- = val;
                }
              if (carry == 1) *ptr1 = (*ptr1 + 1) % 10;
            }

          /* We now know the quotient digit. */
          *qptr++ =  qguess;
          qdig++;
        }
    }

  /* Clean up and return the number. */
  qval->n_sign = ( n1->n_sign == n2->n_sign ? PLUS : MINUS );
  if (bc_is_zero (qval)) qval->n_sign = PLUS;
  _bc_rm_leading_zeros (qval);
  bc_free_num (quot);
  *quot = qval;

  /* Clean up temporary storage. */
  free (mval);
  free (num1);
  free (num2);

  return 0;     /* Everything is OK. */
}


/* Division *and* modulo for numbers.  This computes both NUM1 / NUM2 and
   NUM1 % NUM2  and puts the results in QUOT and REM, except that if QUOT
   is NULL then that store will be omitted.
 */

int
bc_divmod (num1, num2, quot, rem, scale)
     bc_num num1, num2, *quot, *rem;
     int scale;
{
```

```
  bc_num quotient = NULL;
  bc_num temp;
  int rscale;

  /* Check for correct numbers. */
  if (bc_is_zero (num2)) return -1;

  /* Calculate final scale. */
  rscale = MAX (num1->n_scale, num2->n_scale+scale);
  bc_init_num(&temp);

  /* Calculate it. */
  bc_divide (num1, num2, &temp, scale);
  if (quot)
    quotient = bc_copy_num (temp);
  bc_multiply (temp, num2, &temp, rscale);
  bc_sub (num1, temp, rem, rscale);
  bc_free_num (&temp);

  if (quot)
    {
      bc_free_num (quot);
      *quot = quotient;
    }

  return 0;     /* Everything is OK. */
}


/* Modulo for numbers.  This computes NUM1 % NUM2  and puts the
   result in RESULT.   */

int
bc_modulo (num1, num2, result, scale)
     bc_num num1, num2, *result;
     int scale;
{
  return bc_divmod (num1, num2, NULL, result, scale);
}

/* Raise BASE to the EXPO power, reduced modulo MOD.  The result is
   placed in RESULT.  If a EXPO is not an integer,
   only the integer part is used.  */

int
bc_raisemod (base, expo, mod, result, scale)
     bc_num base, expo, mod, *result;
     int scale;
{
  bc_num power, exponent, parity, temp;
  int rscale;

  /* Check for correct numbers. */
  if (bc_is_zero(mod)) return -1;
  if (bc_is_neg(expo)) return -1;

  /* Set initial values.  */
  power = bc_copy_num (base);
  exponent = bc_copy_num (expo);
  temp = bc_copy_num (_one_);
  bc_init_num(&parity);

  /* Check the base for scale digits. */
  if (base->n_scale != 0)
     bc_rt_warn ("non-zero scale in base");

  /* Check the exponent for scale digits. */
  if (exponent->n_scale != 0)
    {
      bc_rt_warn ("non-zero scale in exponent");
      bc_divide (exponent, _one_, &exponent, 0); /*truncate */
    }

  /* Check the modulus for scale digits. */
  if (mod->n_scale != 0)
     bc_rt_warn ("non-zero scale in modulus");

  /* Do the calculation. */
  rscale = MAX(scale, base->n_scale);
  while ( !bc_is_zero(exponent) )
    {
      (void) bc_divmod (exponent, _two_, &exponent, &parity, 0);
      if ( !bc_is_zero(parity) )
        {
          bc_multiply (temp, power, &temp, rscale);
          (void) bc_modulo (temp, mod, &temp, scale);
        }

      bc_multiply (power, power, &power, rscale);
      (void) bc_modulo (power, mod, &power, scale);
    }

  /* Assign the value. */
  bc_free_num (&power);
  bc_free_num (&exponent);
  bc_free_num (result);
  *result = temp;
  return 0;     /* Everything is OK. */
```

```
}

/* Raise NUM1 to the NUM2 power.  The result is placed in RESULT.
   Maximum exponent is LONG_MAX.  If a NUM2 is not an integer,
   only the integer part is used.  */

void
bc_raise (num1, num2, result, scale)
     bc_num num1, num2, *result;
     int scale;
{
   bc_num temp, power;
   long exponent;
   int rscale;
   int pwrscale;
   int calcscale;
   char neg;

   /* Check the exponent for scale digits and convert to a long. */
   if (num2->n_scale != 0)
     bc_rt_warn ("non-zero scale in exponent");
   exponent = bc_num2long (num2);
   if (exponent == 0 && (num2->n_len > 1 || num2->n_value[0] != 0))
       bc_rt_error ("exponent too large in raise");

   /* Special case if exponent is a zero. */
   if (exponent == 0)
     {
       bc_free_num (result);
       *result = bc_copy_num (_one_);
       return;
     }

   /* Other initializations. */
   if (exponent < 0)
     {
       neg = TRUE;
       exponent = -exponent;
       rscale = scale;
     }
   else
     {
       neg = FALSE;
       rscale = MIN (num1->n_scale*exponent, MAX(scale, num1->n_scale));
     }

   /* Set initial value of temp.  */
   power = bc_copy_num (num1);
   pwrscale = num1->n_scale;
   while ((exponent & 1) == 0)
     {
       pwrscale = 2*pwrscale;
       bc_multiply (power, power, &power, pwrscale);
       exponent = exponent >> 1;
     }
   temp = bc_copy_num (power);
   calcscale = pwrscale;
   exponent = exponent >> 1;

   /* Do the calculation. */
   while (exponent > 0)
     {
       pwrscale = 2*pwrscale;
       bc_multiply (power, power, &power, pwrscale);
       if ((exponent & 1) == 1) {
         calcscale = pwrscale + calcscale;
         bc_multiply (temp, power, &temp, calcscale);
       }
       exponent = exponent >> 1;
     }

   /* Assign the value. */
   if (neg)
     {
       bc_divide (_one_, temp, result, rscale);
       bc_free_num (&temp);
     }
   else
     {
       bc_free_num (result);
       *result = temp;
       if ((*result)->n_scale > rscale)
         (*result)->n_scale = rscale;
     }
   bc_free_num (&power);
}

/* Take the square root NUM and return it in NUM with SCALE digits
   after the decimal place. */

int
bc_sqrt (num, scale)
     bc_num *num;
     int scale;
{
  int rscale, cmp_res, done;
  int cscale;
  bc_num guess, guess1, point5, diff;
```

```
  /* Initial checks. */
  cmp_res = bc_compare (*num, _zero_);
  if (cmp_res < 0)
    return 0;           /* error */
  else
    {
      if (cmp_res == 0)
        {
          bc_free_num (num);
          *num = bc_copy_num (_zero_);
          return 1;
        }
    }
  cmp_res = bc_compare (*num, _one_);
  if (cmp_res == 0)
    {
      bc_free_num (num);
      *num = bc_copy_num (_one_);
      return 1;
    }

  /* Initialize the variables. */
  rscale = MAX (scale, (*num)->n_scale);
  bc_init_num(&guess);
  bc_init_num(&guess1);
  bc_init_num(&diff);
  point5 = bc_new_num (1,1);
  point5->n_value[1] = 5;


  /* Calculate the initial guess. */
  if (cmp_res < 0)
    {
      /* The number is between 0 and 1.  Guess should start at 1. */
      guess = bc_copy_num (_one_);
      cscale = (*num)->n_scale;
    }
  else
    {
      /* The number is greater than 1.  Guess should start at 10^(exp/2). */
      bc_int2num (&guess,10);

      bc_int2num (&guess1,(*num)->n_len);
      bc_multiply (guess1, point5, &guess1, 0);
      guess1->n_scale = 0;
      bc_raise (guess, guess1, &guess, 0);
      bc_free_num (&guess1);
      cscale = 3;
    }

  /* Find the square root using Newton's algorithm. */
  done = FALSE;
  while (!done)
    {
      bc_free_num (&guess1);
      guess1 = bc_copy_num (guess);
      bc_divide (*num, guess, &guess, cscale);
      bc_add (guess, guess1, &guess, 0);
      bc_multiply (guess, point5, &guess, cscale);
      bc_sub (guess, guess1, &diff, cscale+1);
      if (bc_is_near_zero (diff, cscale))
        {
          if (cscale < rscale+1)
            cscale = MIN (cscale*3, rscale+1);
          else
            done = TRUE;
        }
    }

  /* Assign the number and clean up. */
  bc_free_num (num);
  bc_divide (guess,_one_,num,rscale);
  bc_free_num (&guess);
  bc_free_num (&guess1);
  bc_free_num (&point5);
  bc_free_num (&diff);
  return 1;
}


/* The following routines provide output for bcd numbers package
   using the rules of POSIX bc for output. */

/* This structure is used for saving digits in the conversion process. */
typedef struct stk_rec {
        long  digit;
        struct stk_rec *next;
} stk_rec;

/* The reference string for digits. */
static char ref_str[] = "0123456789ABCDEF";


/* A special output routine for "multi-character digits."  Exactly
   SIZE characters must be output for the value VAL.  If SPACE is
   non-zero, we must output one space before the number.  OUT_CHAR
   is the actual routine for writing the characters. */
```

```
void
bc_out_long (val, size, space, out_char)
     long val;
     int size, space;
#ifdef __STDC__
     void (*out_char)(int);
#else
     void (*out_char)();
#endif
{
  char digits[40];
  int len, ix;

  if (space) (*out_char) (' ');
  sprintf (digits, "%ld", val);
  len = strlen (digits);
  while (size > len)
    {
      (*out_char) ('0');
      size--;
    }
  for (ix=0; ix < len; ix++)
    (*out_char) (digits[ix]);
}

/* Output of a bcd number.  NUM is written in base O_BASE using OUT_CHAR
   as the routine to do the actual output of the characters. */

void
bc_out_num (num, o_base, out_char, leading_zero)
     bc_num num;
     int o_base;
#ifdef __STDC__
     void (*out_char)(int);
#else
     void (*out_char)();
#endif
     int leading_zero;
{
  char *nptr;
  int  index, fdigit, pre_space;
  stk_rec *digits, *temp;
  bc_num int_part, frac_part, base, cur_dig, t_num, max_o_digit;

  /* The negative sign if needed. */
  if (num->n_sign == MINUS) (*out_char) ('-');

  /* Output the number. */
  if (bc_is_zero (num))
    (*out_char) ('0');
  else
    if (o_base == 10)
      {
        /* The number is in base 10, do it the fast way. */
        nptr = num->n_value;
        if (num->n_len > 1 || *nptr != 0)
          for (index=num->n_len; index>0; index--)
            (*out_char) (BCD_CHAR(*nptr++));
        else
          nptr++;

        if (leading_zero && bc_is_zero (num))
          (*out_char) ('0');

        /* Now the fraction. */
        if (num->n_scale > 0)
          {
            (*out_char) ('.');
            for (index=0; index<num->n_scale; index++)
              (*out_char) (BCD_CHAR(*nptr++));
          }
      }
    else
      {
        /* special case ... */
        if (leading_zero && bc_is_zero (num))
          (*out_char) ('0');

        /* The number is some other base. */
        digits = NULL;
        bc_init_num (&int_part);
        bc_divide (num, _one_, &int_part, 0);
        bc_init_num (&frac_part);
        bc_init_num (&cur_dig);
        bc_init_num (&base);
        bc_sub (num, int_part, &frac_part, 0);
        /* Make the INT_PART and FRAC_PART positive. */
        int_part->n_sign = PLUS;
        frac_part->n_sign = PLUS;
        bc_int2num (&base, o_base);
        bc_init_num (&max_o_digit);
        bc_int2num (&max_o_digit, o_base-1);


        /* Get the digits of the integer part and push them on a stack. */
        while (!bc_is_zero (int_part))
          {
```

```
            bc_modulo (int_part, base, &cur_dig, 0);
            temp = (stk_rec *) malloc (sizeof(stk_rec));
            if (temp == NULL) bc_out_of_memory();
            temp->digit = bc_num2long (cur_dig);
            temp->next = digits;
            digits = temp;
            bc_divide (int_part, base, &int_part, 0);
          }

        /* Print the digits on the stack. */
        if (digits != NULL)
          {
            /* Output the digits. */
            while (digits != NULL)
              {
                temp = digits;
                digits = digits->next;
                if (o_base <= 16)
                  (*out_char) (ref_str[ (int) temp->digit]);
                else
                  bc_out_long (temp->digit, max_o_digit->n_len, 1, out_char);
                free (temp);
              }
          }

        /* Get and print the digits of the fraction part. */
        if (num->n_scale > 0)
          {
            (*out_char) ('.');
            pre_space = 0;
            t_num = bc_copy_num (_one_);
            while (t_num->n_len <= num->n_scale) {
              bc_multiply (frac_part, base, &frac_part, num->n_scale);
              fdigit = bc_num2long (frac_part);
              bc_int2num (&int_part, fdigit);
              bc_sub (frac_part, int_part, &frac_part, 0);
              if (o_base <= 16)
                (*out_char) (ref_str[fdigit]);
              else {
                bc_out_long (fdigit, max_o_digit->n_len, pre_space, out_char);
                pre_space = 1;
              }
              bc_multiply (t_num, base, &t_num, 0);
            }
            bc_free_num (&t_num);
          }

        /* Clean up. */
        bc_free_num (&int_part);
        bc_free_num (&frac_part);
        bc_free_num (&base);
        bc_free_num (&cur_dig);
        bc_free_num (&max_o_digit);
      }
}
/* Convert a number NUM to a long.  The function returns only the integer
   part of the number.  For numbers that are too large to represent as
   a long, this function returns a zero.  This can be detected by checking
   the NUM for zero after having a zero returned. */

long
bc_num2long (num)
     bc_num num;
{
  long val;
  char *nptr;
  int  index;

  /* Extract the int value, ignore the fraction. */
  val = 0;
  nptr = num->n_value;
  for (index=num->n_len; (index>0) && (val<=(LONG_MAX/BASE)); index--)
    val = val*BASE + *nptr++;

  /* Check for overflow.  If overflow, return zero. */
  if (index>0) val = 0;
  if (val < 0) val = 0;

  /* Return the value. */
  if (num->n_sign == PLUS)
    return (val);
  else
    return (-val);
}


/* Convert an integer VAL to a bc number NUM. */

void
bc_int2num (num, val)
     bc_num *num;
     int val;
{
  char buffer[30];
  char *bptr, *vptr;
  int  ix = 1;
  char neg = 0;
```

```
  /* Sign. */
  if (val < 0)
    {
      neg = 1;
      val = -val;
    }

  /* Get things going. */
  bptr = buffer;
  *bptr++ = val % BASE;
  val = val / BASE;

  /* Extract remaining digits. */
  while (val != 0)
    {
      *bptr++ = val % BASE;
      val = val / BASE;
      ix++;             /* Count the digits. */
    }

  /* Make the number. */
  bc_free_num (num);
  *num = bc_new_num (ix, 0);
  if (neg) (*num)->n_sign = MINUS;

  /* Assign the digits. */
  vptr = (*num)->n_value;
  while (ix-- > 0)
    *vptr++ = *--bptr;
}

/* Convert a numbers to a string.  Base 10 only.*/

char
*num2str (num)
      bc_num num;
{
  char *str, *sptr;
  char *nptr;
  int  index, signch;

  /* Allocate the string memory. */
  signch = ( num->n_sign == PLUS ? 0 : 1 );  /* Number of sign chars. */
  if (num->n_scale > 0)
    str = (char *) malloc (num->n_len + num->n_scale + 2 + signch);
  else
    str = (char *) malloc (num->n_len + 1 + signch);
  if (str == NULL) bc_out_of_memory();

  /* The negative sign if needed. */
  sptr = str;
  if (signch) *sptr++ = '-';

  /* Load the whole number. */
  nptr = num->n_value;
  for (index=num->n_len; index>0; index--)
    *sptr++ = BCD_CHAR(*nptr++);

  /* Now the fraction. */
  if (num->n_scale > 0)
    {
      *sptr++ = '.';
      for (index=0; index<num->n_scale; index++)
        *sptr++ = BCD_CHAR(*nptr++);
    }

  /* Terminate the string and return it! */
  *sptr = '\0';
  return (str);
}
/* Convert strings to bc numbers.  Base 10 only.*/

void
bc_str2num (num, str, scale)
     bc_num *num;
     char *str;
     int scale;
{
  int digits, strscale;
  char *ptr, *nptr;
  char zero_int;

  /* Prepare num. */
  bc_free_num (num);

  /* Check for valid number and count digits. */
  ptr = str;
  digits = 0;
  strscale = 0;
  zero_int = FALSE;
  if ( (*ptr == '+') || (*ptr == '-'))  ptr++;  /* Sign */
  while (*ptr == '0') ptr++;                    /* Skip leading zeros. */
  while (isdigit((int)*ptr)) ptr++, digits++;   /* digits */
  if (*ptr == '.') ptr++;                       /* decimal point */
  while (isdigit((int)*ptr)) ptr++, strscale++; /* digits */
  if ((*ptr != '\0') || (digits+strscale == 0))
    {
      *num = bc_copy_num (_zero_);
```

```
      return;
    }

  /* Adjust numbers and allocate storage and initialize fields. */
  strscale = MIN(strscale, scale);
  if (digits == 0)
    {
      zero_int = TRUE;
      digits = 1;
    }
  *num = bc_new_num (digits, strscale);

  /* Build the whole number. */
  ptr = str;
  if (*ptr == '-')
    {
      (*num)->n_sign = MINUS;
      ptr++;
    }
  else
    {
      (*num)->n_sign = PLUS;
      if (*ptr == '+') ptr++;
    }
  while (*ptr == '0') ptr++;                    /* Skip leading zeros. */
  nptr = (*num)->n_value;
  if (zero_int)
    {
      *nptr++ = 0;
      digits = 0;
    }
  for (;digits > 0; digits--)
    *nptr++ = CH_VAL(*ptr++);


  /* Build the fractional part. */
  if (strscale > 0)
    {
      ptr++;  /* skip the decimal point! */
      for (;strscale > 0; strscale--)
        *nptr++ = CH_VAL(*ptr++);
    }
}

/* pn prints the number NUM in base 10. */

static void
out_char (int c)
{
  putchar(c);
}


void
pn (num)
     bc_num num;
{
  bc_out_num (num, 10, out_char, 0);
  out_char ('\n');
}


/* pv prints a character array as if it was a string of bcd digits. */
void
pv (name, num, len)
     char *name;
     unsigned char *num;
     int len;
{
  int i;
  printf ("%s=", name);
  for (i=0; i<len; i++) printf ("%c",BCD_CHAR(num[i]));
  printf ("\n");
}
```

Notice for package(s)

grep

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a

menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

  A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

  The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

  The Corresponding Source for a work in source code form is that same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

  When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product

is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the

form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License. Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program. Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance. However,
nothing other than this License grants you permission to propagate or
modify any covered work. These actions infringe copyright if you do
not accept this License. Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License. You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations. If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License. For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a

publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

Notice for package(s)

procps

```
                  GNU LIBRARY GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1991 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the library GPL.  It is
 numbered 2 because it goes with version 2 of the ordinary GPL.]
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for

that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

```
  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission.  For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this.  Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

                            NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU.  SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Library General Public
    License as published by the Free Software Foundation; either
    version 2 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Library General Public License for more details.

    You should have received a copy of the GNU Library General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

coreutils

```
/* 'dir', 'vdir' and 'ls' directory listing programs for GNU.
   Copyright (C) 1985-2015 Free Software Foundation, Inc.

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation, either version 3 of the License, or
   (at your option) any later version.
```

```
   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program.  If not, see <http://www.gnu.org/licenses/>.  */

/* If ls_mode is LS_MULTI_COL,
   the multi-column format is the default regardless
   of the type of output device.
   This is for the 'dir' program.

   If ls_mode is LS_LONG_FORMAT,
   the long format is the default regardless of the
   type of output device.
   This is for the 'vdir' program.

   If ls_mode is LS_LS,
   the output format depends on whether the output
   device is a terminal.
   This is for the 'ls' program.  */

/* Written by Richard Stallman and David MacKenzie.  */

/* Color support by Peter Anvin <Peter.Anvin@linux.org> and Dennis
   Flaherty <dennisf@denix.elk.miles.com> based on original patches by
   Greg Lee <lee@uhunix.uhcc.hawaii.edu>.  */

#include <config.h>
#include <sys/types.h>

#include <termios.h>
#if HAVE_STROPTS_H
# include <stropts.h>
#endif
#include <sys/ioctl.h>

#ifdef WINSIZE_IN_PTEM
# include <sys/stream.h>
# include <sys/ptem.h>
#endif

#include <stdio.h>
#include <assert.h>
#include <setjmp.h>
#include <pwd.h>
#include <getopt.h>
#include <signal.h>
#include <selinux/selinux.h>
#include <wchar.h>

#if HAVE_LANGINFO_CODESET
# include <langinfo.h>
#endif

/* Use SA_NOCLDSTOP as a proxy for whether the sigaction machinery is
   present.  */
#ifndef SA_NOCLDSTOP
# define SA_NOCLDSTOP 0
# define sigprocmask(How, Set, Oset) /* empty */
# define sigset_t int
# if ! HAVE_SIGINTERRUPT
#  define siginterrupt(sig, flag) /* empty */
# endif
#endif

/* NonStop circa 2011 lacks both SA_RESTART and siginterrupt, so don't
   restart syscalls after a signal handler fires.  This may cause
   colors to get messed up on the screen if 'ls' is interrupted, but
   that's the best we can do on such a platform.  */
#ifndef SA_RESTART
# define SA_RESTART 0
#endif

#include "system.h"
#include <fnmatch.h>

#include "acl.h"
#include "argmatch.h"
#include "dev-ino.h"
#include "error.h"
#include "filenamecat.h"
#include "hard-locale.h"
#include "hash.h"
#include "human.h"
#include "filemode.h"
#include "filevercmp.h"
#include "idcache.h"
#include "ls.h"
#include "mbswidth.h"
#include "mpsort.h"
#include "obstack.h"
#include "quote.h"
#include "quotearg.h"
#include "smack.h"
#include "stat-size.h"
#include "stat-time.h"
```

```
#include "strftime.h"
#include "xdectoint.h"
#include "xstrtol.h"
#include "areadlink.h"
#include "mbsalign.h"
#include "dircolors.h"

/* Include <sys/capability.h> last to avoid a clash of <sys/types.h>
   include guards with some premature versions of libcap.
   For more details, see <http://bugzilla.redhat.com/483548>.  */
#ifdef HAVE_CAP
# include <sys/capability.h>
#endif

#define PROGRAM_NAME (ls_mode == LS_LS ? "ls" \
                      : (ls_mode == LS_MULTI_COL \
                         ? "dir" : "vdir"))

#define AUTHORS \
  proper_name ("Richard M. Stallman"), \
  proper_name ("David MacKenzie")

#define obstack_chunk_alloc malloc
#define obstack_chunk_free free

/* Return an int indicating the result of comparing two integers.
   Subtracting doesn't always work, due to overflow.  */
#define longdiff(a, b) ((a) < (b) ? -1 : (a) > (b))

/* Unix-based readdir implementations have historically returned a dirent.d_ino
   value that is sometimes not equal to the stat-obtained st_ino value for
   that same entry.  This error occurs for a readdir entry that refers
   to a mount point.  readdir's error is to return the inode number of
   the underlying directory -- one that typically cannot be stat'ed, as
   long as a file system is mounted on that directory.  RELIABLE_D_INO
   encapsulates whether we can use the more efficient approach of relying
   on readdir-supplied d_ino values, or whether we must incur the cost of
   calling stat or lstat to obtain each guaranteed-valid inode number.  */

#ifndef READDIR_LIES_ABOUT_MOUNTPOINT_D_INO
# define READDIR_LIES_ABOUT_MOUNTPOINT_D_INO 1
#endif

#if READDIR_LIES_ABOUT_MOUNTPOINT_D_INO
# define RELIABLE_D_INO(dp) NOT_AN_INODE_NUMBER
#else
# define RELIABLE_D_INO(dp) D_INO (dp)
#endif

#if ! HAVE_STRUCT_STAT_ST_AUTHOR
# define st_author st_uid
#endif

enum filetype
  {
    unknown,
    fifo,
    chardev,
    directory,
    blockdev,
    normal,
    symbolic_link,
    sock,
    whiteout,
    arg_directory
  };

/* Display letters and indicators for each filetype.
   Keep these in sync with enum filetype.  */
static char const filetype_letter[] = "?pcdb-lswd";

/* Ensure that filetype and filetype_letter have the same
   number of elements.  */
verify (sizeof filetype_letter - 1 == arg_directory + 1);

#define FILETYPE_INDICATORS                                 \
  {                                                         \
    C_ORPHAN, C_FIFO, C_CHR, C_DIR, C_BLK, C_FILE,          \
    C_LINK, C_SOCK, C_FILE, C_DIR                           \
  }

enum acl_type
  {
    ACL_T_NONE,
    ACL_T_LSM_CONTEXT_ONLY,
    ACL_T_YES
  };

struct fileinfo
  {
    /* The file name.  */
    char *name;

    /* For symbolic link, name of the file linked to, otherwise zero.  */
    char *linkname;

    struct stat stat;
```

```
    enum filetype filetype;

    /* For symbolic link and long listing, st_mode of file linked to, otherwise
       zero.  */
    mode_t linkmode;

    /* security context.  */
    char *scontext;

    bool stat_ok;

    /* For symbolic link and color printing, true if linked-to file
       exists, otherwise false.  */
    bool linkok;

    /* For long listings, true if the file has an access control list,
       or a security context.  */
    enum acl_type acl_type;

    /* For color listings, true if a regular file has capability info.  */
    bool has_capability;
  };

#define LEN_STR_PAIR(s) sizeof (s) - 1, s

/* Null is a valid character in a color indicator (think about Epson
   printers, for example) so we have to use a length/buffer string
   type.  */

struct bin_str
  {
    size_t len;                 /* Number of bytes */
    const char *string;         /* Pointer to the same */
  };

#if ! HAVE_TCGETPGRP
# define tcgetpgrp(Fd) 0
#endif

static size_t quote_name (FILE *out, const char *name,
                          struct quoting_options const *options,
                          size_t *width);
static char *make_link_name (char const *name, char const *linkname);
static int decode_switches (int argc, char **argv);
static bool file_ignored (char const *name);
static uintmax_t gobble_file (char const *name, enum filetype type,
                              ino_t inode, bool command_line_arg,
                              char const *dirname);
static bool print_color_indicator (const struct fileinfo *f,
                                   bool symlink_target);
static void put_indicator (const struct bin_str *ind);
static void add_ignore_pattern (const char *pattern);
static void attach (char *dest, const char *dirname, const char *name);
static void clear_files (void);
static void extract_dirs_from_files (char const *dirname,
                                     bool command_line_arg);
static void get_link_name (char const *filename, struct fileinfo *f,
                           bool command_line_arg);
static void indent (size_t from, size_t to);
static size_t calculate_columns (bool by_columns);
static void print_current_files (void);
static void print_dir (char const *name, char const *realname,
                       bool command_line_arg);
static size_t print_file_name_and_frills (const struct fileinfo *f,
                                          size_t start_col);
static void print_horizontal (void);
static int format_user_width (uid_t u);
static int format_group_width (gid_t g);
static void print_long_format (const struct fileinfo *f);
static void print_many_per_line (void);
static size_t print_name_with_quoting (const struct fileinfo *f,
                                       bool symlink_target,
                                       struct obstack *stack,
                                       size_t start_col);
static void prep_non_filename_text (void);
static bool print_type_indicator (bool stat_ok, mode_t mode,
                                  enum filetype type);
static void print_with_commas (void);
static void queue_directory (char const *name, char const *realname,
                             bool command_line_arg);
static void sort_files (void);
static void parse_ls_color (void);

static void getenv_quoting_style (void);

/* Initial size of hash table.
   Most hierarchies are likely to be shallower than this.  */
#define INITIAL_TABLE_SIZE 30

/* The set of 'active' directories, from the current command-line argument
   to the level in the hierarchy at which files are being listed.
   A directory is represented by its device and inode numbers (struct dev_ino).
   A directory is added to this set when ls begins listing it or its
   entries, and it is removed from the set just after ls has finished
   processing it.  This set is used solely to detect loops, e.g., with
   mkdir loop; cd loop; ln -s ../loop sub; ls -RL  */
static Hash_table *active_dir_set;
```

```
#define LOOP_DETECT (!!active_dir_set)

/* The table of files in the current directory:

   'cwd_file' points to a vector of 'struct fileinfo', one per file.
   'cwd_n_alloc' is the number of elements space has been allocated for.
   'cwd_n_used' is the number actually in use.  */

/* Address of block containing the files that are described.  */
static struct fileinfo *cwd_file;

/* Length of block that 'cwd_file' points to, measured in files.  */
static size_t cwd_n_alloc;

/* Index of first unused slot in 'cwd_file'.  */
static size_t cwd_n_used;

/* Vector of pointers to files, in proper sorted order, and the number
   of entries allocated for it.  */
static void **sorted_file;
static size_t sorted_file_alloc;

/* When true, in a color listing, color each symlink name according to the
   type of file it points to.  Otherwise, color them according to the 'ln'
   directive in LS_COLORS.  Dangling (orphan) symlinks are treated specially,
   regardless.  This is set when 'ln=target' appears in LS_COLORS.  */

static bool color_symlink_as_referent;

/* mode of appropriate file for colorization */
#define FILE_OR_LINK_MODE(File) \
    ((color_symlink_as_referent && (File)->linkok) \
     ? (File)->linkmode : (File)->stat.st_mode)


/* Record of one pending directory waiting to be listed.  */

struct pending
  {
    char *name;
    /* If the directory is actually the file pointed to by a symbolic link we
       were told to list, 'realname' will contain the name of the symbolic
       link, otherwise zero.  */
    char *realname;
    bool command_line_arg;
    struct pending *next;
  };

static struct pending *pending_dirs;

/* Current time in seconds and nanoseconds since 1970, updated as
   needed when deciding whether a file is recent.  */

static struct timespec current_time;

static bool print_scontext;
static char UNKNOWN_SECURITY_CONTEXT[] = "?";

/* Whether any of the files has an ACL.  This affects the width of the
   mode column.  */

static bool any_has_acl;

/* The number of columns to use for columns containing inode numbers,
   block sizes, link counts, owners, groups, authors, major device
   numbers, minor device numbers, and file sizes, respectively.  */

static int inode_number_width;
static int block_size_width;
static int nlink_width;
static int scontext_width;
static int owner_width;
static int group_width;
static int author_width;
static int major_device_number_width;
static int minor_device_number_width;
static int file_size_width;

/* Option flags */

/* long_format for lots of info, one per line.
   one_per_line for just names, one per line.
   many_per_line for just names, many per line, sorted vertically.
   horizontal for just names, many per line, sorted horizontally.
   with_commas for just names, many per line, separated by commas.

   -l (and other options that imply -l), -1, -C, -x and -m control
   this parameter.  */

enum format
  {
    long_format,                /* -l and other options that imply -l */
    one_per_line,               /* -1 */
    many_per_line,              /* -C */
    horizontal,                 /* -x */
    with_commas                 /* -m */
  };
```

```
static enum format format;

/* 'full-iso' uses full ISO-style dates and times.  'long-iso' uses longer
   ISO-style time stamps, though shorter than 'full-iso'.  'iso' uses shorter
   ISO-style time stamps.  'locale' uses locale-dependent time stamps.  */
enum time_style
  {
    full_iso_time_style,        /* --time-style=full-iso */
    long_iso_time_style,        /* --time-style=long-iso */
    iso_time_style,             /* --time-style=iso */
    locale_time_style           /* --time-style=locale */
  };

static char const *const time_style_args[] =
{
  "full-iso", "long-iso", "iso", "locale", NULL
};
static enum time_style const time_style_types[] =
{
  full_iso_time_style, long_iso_time_style, iso_time_style,
  locale_time_style
};
ARGMATCH_VERIFY (time_style_args, time_style_types);

/* Type of time to print or sort by.  Controlled by -c and -u.
   The values of each item of this enum are important since they are
   used as indices in the sort functions array (see sort_files()).  */

enum time_type
  {
    time_mtime,                 /* default */
    time_ctime,                 /* -c */
    time_atime,                 /* -u */
    time_numtypes               /* the number of elements of this enum */
  };

static enum time_type time_type;

/* The file characteristic to sort by.  Controlled by -t, -S, -U, -X, -v.
   The values of each item of this enum are important since they are
   used as indices in the sort functions array (see sort_files()).  */

enum sort_type
  {
    sort_none = -1,             /* -U */
    sort_name,                  /* default */
    sort_extension,             /* -X */
    sort_size,                  /* -S */
    sort_version,               /* -v */
    sort_time,                  /* -t */
    sort_numtypes               /* the number of elements of this enum */
  };

static enum sort_type sort_type;

/* Direction of sort.
   false means highest first if numeric,
   lowest first if alphabetic;
   these are the defaults.
   true means the opposite order in each case.  -r  */

static bool sort_reverse;

/* True means to display owner information.  -g turns this off.  */

static bool print_owner = true;

/* True means to display author information.  */

static bool print_author;

/* True means to display group information.  -G and -o turn this off.  */

static bool print_group = true;

/* True means print the user and group id's as numbers rather
   than as names.  -n  */

static bool numeric_ids;

/* True means mention the size in blocks of each file.  -s  */

static bool print_block_size;

/* Human-readable options for output, when printing block counts.  */
static int human_output_opts;

/* The units to use when printing block counts.  */
static uintmax_t output_block_size;

/* Likewise, but for file sizes.  */
static int file_human_output_opts;
static uintmax_t file_output_block_size = 1;

/* Follow the output with a special string.  Using this format,
   Emacs' dired mode starts up twice as fast, and can handle all
   strange characters in file names.  */
static bool dired;
```

```
/* 'none' means don't mention the type of files.
   'slash' means mention directories only, with a '/'.
   'file_type' means mention file types.
   'classify' means mention file types and mark executables.

   Controlled by -F, -p, and --indicator-style.  */

enum indicator_style
  {
    none,       /*     --indicator-style=none */
    slash,      /* -p, --indicator-style=slash */
    file_type,  /*     --indicator-style=file-type */
    classify    /* -F, --indicator-style=classify */
  };

static enum indicator_style indicator_style;

/* Names of indicator styles.  */
static char const *const indicator_style_args[] =
{
  "none", "slash", "file-type", "classify", NULL
};
static enum indicator_style const indicator_style_types[] =
{
  none, slash, file_type, classify
};
ARGMATCH_VERIFY (indicator_style_args, indicator_style_types);

/* True means use colors to mark types.  Also define the different
   colors as well as the stuff for the LS_COLORS environment variable.
   The LS_COLORS variable is now in a termcap-like format.  */

static bool print_with_color;

/* Whether we used any colors in the output so far.  If so, we will
   need to restore the default color later.  If not, we will need to
   call prep_non_filename_text before using color for the first time. */

static bool used_color = false;

enum color_type
  {
    color_never,                /* 0: default or --color=never */
    color_always,               /* 1: --color=always */
    color_if_tty                /* 2: --color=tty */
  };

enum Dereference_symlink
  {
    DEREF_UNDEFINED = 1,
    DEREF_NEVER,
    DEREF_COMMAND_LINE_ARGUMENTS,       /* -H */
    DEREF_COMMAND_LINE_SYMLINK_TO_DIR,  /* the default, in certain cases */
    DEREF_ALWAYS                        /* -L */
  };

enum indicator_no
  {
    C_LEFT, C_RIGHT, C_END, C_RESET, C_NORM, C_FILE, C_DIR, C_LINK,
    C_FIFO, C_SOCK,
    C_BLK, C_CHR, C_MISSING, C_ORPHAN, C_EXEC, C_DOOR, C_SETUID, C_SETGID,
    C_STICKY, C_OTHER_WRITABLE, C_STICKY_OTHER_WRITABLE, C_CAP, C_MULTIHARDLINK,
    C_CLR_TO_EOL
  };

static const char *const indicator_name[]=
  {
    "lc", "rc", "ec", "rs", "no", "fi", "di", "ln", "pi", "so",
    "bd", "cd", "mi", "or", "ex", "do", "su", "sg", "st",
    "ow", "tw", "ca", "mh", "cl", NULL
  };

struct color_ext_type
  {
    struct bin_str ext;         /* The extension we're looking for */
    struct bin_str seq;         /* The sequence to output when we do */
    struct color_ext_type *next;        /* Next in list */
  };

static struct bin_str color_indicator[] =
  {
    { LEN_STR_PAIR ("\033[") },         /* lc: Left of color sequence */
    { LEN_STR_PAIR ("m") },             /* rc: Right of color sequence */
    { 0, NULL },                        /* ec: End color (replaces lc+rs+rc) */
    { LEN_STR_PAIR ("0") },             /* rs: Reset to ordinary colors */
    { 0, NULL },                        /* no: Normal */
    { 0, NULL },                        /* fi: File: default */
    { LEN_STR_PAIR ("01;34") },         /* di: Directory: bright blue */
    { LEN_STR_PAIR ("01;36") },         /* ln: Symlink: bright cyan */
    { LEN_STR_PAIR ("33") },            /* pi: Pipe: yellow/brown */
    { LEN_STR_PAIR ("01;35") },         /* so: Socket: bright magenta */
    { LEN_STR_PAIR ("01;33") },         /* bd: Block device: bright yellow */
    { LEN_STR_PAIR ("01;33") },         /* cd: Char device: bright yellow */
    { 0, NULL },                        /* mi: Missing file: undefined */
    { 0, NULL },                        /* or: Orphaned symlink: undefined */
    { LEN_STR_PAIR ("01;32") },         /* ex: Executable: bright green */
    { LEN_STR_PAIR ("01;35") },         /* do: Door: bright magenta */
```

```
    { LEN_STR_PAIR ("37;41") },         /* su: setuid: white on red */
    { LEN_STR_PAIR ("30;43") },         /* sg: setgid: black on yellow */
    { LEN_STR_PAIR ("37;44") },         /* st: sticky: black on blue */
    { LEN_STR_PAIR ("34;42") },         /* ow: other-writable: blue on green */
    { LEN_STR_PAIR ("30;42") },         /* tw: ow w/ sticky: black on green */
    { LEN_STR_PAIR ("30;41") },         /* ca: black on red */
    { 0, NULL },                        /* mh: disabled by default */
    { LEN_STR_PAIR ("\033[K") },        /* cl: clear to end of line */
  };

/* FIXME: comment  */
static struct color_ext_type *color_ext_list = NULL;

/* Buffer for color sequences */
static char *color_buf;

/* True means to check for orphaned symbolic link, for displaying
   colors.  */

static bool check_symlink_color;

/* True means mention the inode number of each file.  -i  */

static bool print_inode;

/* What to do with symbolic links.  Affected by -d, -F, -H, -l (and
   other options that imply -l), and -L.  */

static enum Dereference_symlink dereference;

/* True means when a directory is found, display info on its
   contents.  -R  */

static bool recursive;

/* True means when an argument is a directory name, display info
   on it itself.  -d  */

static bool immediate_dirs;

/* True means that directories are grouped before files. */

static bool directories_first;

/* Which files to ignore.  */

static enum
{
  /* Ignore files whose names start with '.', and files specified by
     --hide and --ignore.  */
  IGNORE_DEFAULT,

  /* Ignore '.', '..', and files specified by --ignore.  */
  IGNORE_DOT_AND_DOTDOT,

  /* Ignore only files specified by --ignore.  */
  IGNORE_MINIMAL
} ignore_mode;

/* A linked list of shell-style globbing patterns.  If a non-argument
   file name matches any of these patterns, it is ignored.
   Controlled by -I.  Multiple -I options accumulate.
   The -B option adds '*~' and '.*~' to this list.  */

struct ignore_pattern
  {
    const char *pattern;
    struct ignore_pattern *next;
  };

static struct ignore_pattern *ignore_patterns;

/* Similar to IGNORE_PATTERNS, except that -a or -A causes this
   variable itself to be ignored.  */
static struct ignore_pattern *hide_patterns;

/* True means output nongraphic chars in file names as '?'.
   (-q, --hide-control-chars)
   qmark_funny_chars and the quoting style (-Q, --quoting-style=WORD) are
   independent.  The algorithm is: first, obey the quoting style to get a
   string representing the file name;  then, if qmark_funny_chars is set,
   replace all nonprintable chars in that string with '?'.  It's necessary
   to replace nonprintable chars even in quoted strings, because we don't
   want to mess up the terminal if control chars get sent to it, and some
   quoting methods pass through control chars as-is.  */
static bool qmark_funny_chars;

/* Quoting options for file and dir name output.  */

static struct quoting_options *filename_quoting_options;
static struct quoting_options *dirname_quoting_options;

/* The number of chars per hardware tab stop.  Setting this to zero
   inhibits the use of TAB characters for separating columns.  -T */
static size_t tabsize;

/* True means print each directory name before listing it.  */
```

```
static bool print_dir_name;

/* The line length to use for breaking lines in many-per-line format.
   Can be set with -w.  */

static size_t line_length;

/* If true, the file listing format requires that stat be called on
   each file.  */

static bool format_needs_stat;

/* Similar to 'format_needs_stat', but set if only the file type is
   needed.  */

static bool format_needs_type;

/* An arbitrary limit on the number of bytes in a printed time stamp.
   This is set to a relatively small value to avoid the need to worry
   about denial-of-service attacks on servers that run "ls" on behalf
   of remote clients.  1000 bytes should be enough for any practical
   time stamp format.  */

enum { TIME_STAMP_LEN_MAXIMUM = MAX (1000, INT_STRLEN_BOUND (time_t)) };

/* strftime formats for non-recent and recent files, respectively, in
   -l output.  */

static char const *long_time_format[2] =
  {
    /* strftime format for non-recent files (older than 6 months), in
       -l output.  This should contain the year, month and day (at
       least), in an order that is understood by people in your
       locale's territory.  Please try to keep the number of used
       screen columns small, because many people work in windows with
       only 80 columns.  But make this as wide as the other string
       below, for recent files.  */
    /* TRANSLATORS: ls output needs to be aligned for ease of reading,
       so be wary of using variable width fields from the locale.
       Note %b is handled specially by ls and aligned correctly.
       Note also that specifying a width as in %5b is erroneous as strftime
       will count bytes rather than characters in multibyte locales.  */
    N_("%b %e  %Y"),
    /* strftime format for recent files (younger than 6 months), in -l
       output.  This should contain the month, day and time (at
       least), in an order that is understood by people in your
       locale's territory.  Please try to keep the number of used
       screen columns small, because many people work in windows with
       only 80 columns.  But make this as wide as the other string
       above, for non-recent files.  */
    /* TRANSLATORS: ls output needs to be aligned for ease of reading,
       so be wary of using variable width fields from the locale.
       Note %b is handled specially by ls and aligned correctly.
       Note also that specifying a width as in %5b is erroneous as strftime
       will count bytes rather than characters in multibyte locales.  */
    N_("%b %e %H:%M")
  };

/* The set of signals that are caught.  */

static sigset_t caught_signals;

/* If nonzero, the value of the pending fatal signal.  */

static sig_atomic_t volatile interrupt_signal;

/* A count of the number of pending stop signals that have been received.  */

static sig_atomic_t volatile stop_signal_count;

/* Desired exit status.  */

static int exit_status;

/* Exit statuses.  */
enum
  {
    /* "ls" had a minor problem.  E.g., while processing a directory,
       ls obtained the name of an entry via readdir, yet was later
       unable to stat that name.  This happens when listing a directory
       in which entries are actively being removed or renamed.  */
    LS_MINOR_PROBLEM = 1,

    /* "ls" had more serious trouble (e.g., memory exhausted, invalid
       option or failure to stat a command line argument.  */
    LS_FAILURE = 2
  };

/* For long options that have no equivalent short option, use a
   non-character as a pseudo short option, starting with CHAR_MAX + 1.  */
enum
{
  AUTHOR_OPTION = CHAR_MAX + 1,
  BLOCK_SIZE_OPTION,
  COLOR_OPTION,
  DEREFERENCE_COMMAND_LINE_SYMLINK_TO_DIR_OPTION,
  FILE_TYPE_INDICATOR_OPTION,
  FORMAT_OPTION,
```

```
  FULL_TIME_OPTION,
  GROUP_DIRECTORIES_FIRST_OPTION,
  HIDE_OPTION,
  INDICATOR_STYLE_OPTION,
  QUOTING_STYLE_OPTION,
  SHOW_CONTROL_CHARS_OPTION,
  SI_OPTION,
  SORT_OPTION,
  TIME_OPTION,
  TIME_STYLE_OPTION
};

static struct option const long_options[] =
{
  {"all", no_argument, NULL, 'a'},
  {"escape", no_argument, NULL, 'b'},
  {"directory", no_argument, NULL, 'd'},
  {"dired", no_argument, NULL, 'D'},
  {"full-time", no_argument, NULL, FULL_TIME_OPTION},
  {"group-directories-first", no_argument, NULL,
   GROUP_DIRECTORIES_FIRST_OPTION},
  {"human-readable", no_argument, NULL, 'h'},
  {"inode", no_argument, NULL, 'i'},
  {"kibibytes", no_argument, NULL, 'k'},
  {"numeric-uid-gid", no_argument, NULL, 'n'},
  {"no-group", no_argument, NULL, 'G'},
  {"hide-control-chars", no_argument, NULL, 'q'},
  {"reverse", no_argument, NULL, 'r'},
  {"size", no_argument, NULL, 's'},
  {"width", required_argument, NULL, 'w'},
  {"almost-all", no_argument, NULL, 'A'},
  {"ignore-backups", no_argument, NULL, 'B'},
  {"classify", no_argument, NULL, 'F'},
  {"file-type", no_argument, NULL, FILE_TYPE_INDICATOR_OPTION},
  {"si", no_argument, NULL, SI_OPTION},
  {"dereference-command-line", no_argument, NULL, 'H'},
  {"dereference-command-line-symlink-to-dir", no_argument, NULL,
   DEREFERENCE_COMMAND_LINE_SYMLINK_TO_DIR_OPTION},
  {"hide", required_argument, NULL, HIDE_OPTION},
  {"ignore", required_argument, NULL, 'I'},
  {"indicator-style", required_argument, NULL, INDICATOR_STYLE_OPTION},
  {"dereference", no_argument, NULL, 'L'},
  {"literal", no_argument, NULL, 'N'},
  {"quote-name", no_argument, NULL, 'Q'},
  {"quoting-style", required_argument, NULL, QUOTING_STYLE_OPTION},
  {"recursive", no_argument, NULL, 'R'},
  {"format", required_argument, NULL, FORMAT_OPTION},
  {"show-control-chars", no_argument, NULL, SHOW_CONTROL_CHARS_OPTION},
  {"sort", required_argument, NULL, SORT_OPTION},
  {"tabsize", required_argument, NULL, 'T'},
  {"time", required_argument, NULL, TIME_OPTION},
  {"time-style", required_argument, NULL, TIME_STYLE_OPTION},
  {"color", optional_argument, NULL, COLOR_OPTION},
  {"block-size", required_argument, NULL, BLOCK_SIZE_OPTION},
  {"context", no_argument, 0, 'Z'},
  {"author", no_argument, NULL, AUTHOR_OPTION},
  {GETOPT_HELP_OPTION_DECL},
  {GETOPT_VERSION_OPTION_DECL},
  {NULL, 0, NULL, 0}
};

static char const *const format_args[] =
{
  "verbose", "long", "commas", "horizontal", "across",
  "vertical", "single-column", NULL
};
static enum format const format_types[] =
{
  long_format, long_format, with_commas, horizontal, horizontal,
  many_per_line, one_per_line
};
ARGMATCH_VERIFY (format_args, format_types);

static char const *const sort_args[] =
{
  "none", "time", "size", "extension", "version", NULL
};
static enum sort_type const sort_types[] =
{
  sort_none, sort_time, sort_size, sort_extension, sort_version
};
ARGMATCH_VERIFY (sort_args, sort_types);

static char const *const time_args[] =
{
  "atime", "access", "use", "ctime", "status", NULL
};
static enum time_type const time_types[] =
{
  time_atime, time_atime, time_atime, time_ctime, time_ctime
};
ARGMATCH_VERIFY (time_args, time_types);

static char const *const color_args[] =
{
  /* force and none are for compatibility with another color-ls version */
  "always", "yes", "force",
  "never", "no", "none",
```

```
  "auto", "tty", "if-tty", NULL
};
static enum color_type const color_types[] =
{
  color_always, color_always, color_always,
  color_never, color_never, color_never,
  color_if_tty, color_if_tty, color_if_tty
};
ARGMATCH_VERIFY (color_args, color_types);

/* Information about filling a column.  */
struct column_info
{
  bool valid_len;
  size_t line_len;
  size_t *col_arr;
};

/* Array with information about column filledness.  */
static struct column_info *column_info;

/* Maximum number of columns ever possible for this display.  */
static size_t max_idx;

/* The minimum width of a column is 3: 1 character for the name and 2
   for the separating white space.  */
#define MIN_COLUMN_WIDTH        3


/* This zero-based index is used solely with the --dired option.
   When that option is in effect, this counter is incremented for each
   byte of output generated by this program so that the beginning
   and ending indices (in that output) of every file name can be recorded
   and later output themselves.  */
static size_t dired_pos;

#define DIRED_PUTCHAR(c) do {putchar ((c)); ++dired_pos;} while (0)

/* Write S to STREAM and increment DIRED_POS by S_LEN.  */
#define DIRED_FPUTS(s, stream, s_len) \
    do {fputs (s, stream); dired_pos += s_len;} while (0)

/* Like DIRED_FPUTS, but for use when S is a literal string.  */
#define DIRED_FPUTS_LITERAL(s, stream) \
    do {fputs (s, stream); dired_pos += sizeof (s) - 1;} while (0)

#define DIRED_INDENT()                                                  \
    do                                                                  \
      {                                                                 \
        if (dired)                                                      \
          DIRED_FPUTS_LITERAL ("  ", stdout);                           \
      }                                                                 \
    while (0)

/* With --dired, store pairs of beginning and ending indices of filenames.  */
static struct obstack dired_obstack;

/* With --dired, store pairs of beginning and ending indices of any
   directory names that appear as headers (just before 'total' line)
   for lists of directory entries.  Such directory names are seen when
   listing hierarchies using -R and when a directory is listed with at
   least one other command line argument.  */
static struct obstack subdired_obstack;

/* Save the current index on the specified obstack, OBS.  */
#define PUSH_CURRENT_DIRED_POS(obs)                                     \
  do                                                                    \
    {                                                                   \
      if (dired)                                                        \
        obstack_grow (obs, &dired_pos, sizeof (dired_pos));             \
    }                                                                   \
  while (0)

/* With -R, this stack is used to help detect directory cycles.
   The device/inode pairs on this stack mirror the pairs in the
   active_dir_set hash table.  */
static struct obstack dev_ino_obstack;

/* Push a pair onto the device/inode stack.  */
static void
dev_ino_push (dev_t dev, ino_t ino)
{
  void *vdi;
  struct dev_ino *di;
  int dev_ino_size = sizeof *di;
  obstack_blank (&dev_ino_obstack, dev_ino_size);
  vdi = obstack_next_free (&dev_ino_obstack);
  di = vdi;
  di--;
  di->st_dev = dev;
  di->st_ino = ino;
}

/* Pop a dev/ino struct off the global dev_ino_obstack
   and return that struct.  */
static struct dev_ino
dev_ino_pop (void)
{
```

```
  void *vdi;
  struct dev_ino *di;
  int dev_ino_size = sizeof *di;
  assert (dev_ino_size <= obstack_object_size (&dev_ino_obstack));
  obstack_blank_fast (&dev_ino_obstack, -dev_ino_size);
  vdi = obstack_next_free (&dev_ino_obstack);
  di = vdi;
  return *di;
}

/* Note the use commented out below:
#define ASSERT_MATCHING_DEV_INO(Name, Di)       \
  do                                            \
    {                                           \
      struct stat sb;                           \
      assert (Name);                            \
      assert (0 <= stat (Name, &sb));           \
      assert (sb.st_dev == Di.st_dev);          \
      assert (sb.st_ino == Di.st_ino);          \
    }                                           \
  while (0)
*/

/* Write to standard output PREFIX, followed by the quoting style and
   a space-separated list of the integers stored in OS all on one line.  */

static void
dired_dump_obstack (const char *prefix, struct obstack *os)
{
  size_t n_pos;

  n_pos = obstack_object_size (os) / sizeof (dired_pos);
  if (n_pos > 0)
    {
      size_t i;
      size_t *pos;

      pos = (size_t *) obstack_finish (os);
      fputs (prefix, stdout);
      for (i = 0; i < n_pos; i++)
        printf (" %lu", (unsigned long int) pos[i]);
      putchar ('\n');
    }
}

/* Read the abbreviated month names from the locale, to align them
   and to determine the max width of the field and to truncate names
   greater than our max allowed.
   Note even though this handles multibyte locales correctly
   it's not restricted to them as single byte locales can have
   variable width abbreviated months and also precomputing/caching
   the names was seen to increase the performance of ls significantly.  */

/* max number of display cells to use */
enum { MAX_MON_WIDTH = 5 };
/* In the unlikely event that the abmon[] storage is not big enough
   an error message will be displayed, and we revert to using
   unmodified abbreviated month names from the locale database.  */
static char abmon[12][MAX_MON_WIDTH * 2 * MB_LEN_MAX + 1];
/* minimum width needed to align %b, 0 => don't use precomputed values.  */
static size_t required_mon_width;

static size_t
abmon_init (void)
{
#ifdef HAVE_NL_LANGINFO
  required_mon_width = MAX_MON_WIDTH;
  size_t curr_max_width;
  do
    {
      curr_max_width = required_mon_width;
      required_mon_width = 0;
      for (int i = 0; i < 12; i++)
        {
          size_t width = curr_max_width;

          size_t req = mbsalign (nl_langinfo (ABMON_1 + i),
                                 abmon[i], sizeof (abmon[i]),
                                 &width, MBS_ALIGN_LEFT, 0);

          if (req == (size_t) -1 || req >= sizeof (abmon[i]))
            {
              required_mon_width = 0; /* ignore precomputed strings.  */
              return required_mon_width;
            }

          required_mon_width = MAX (required_mon_width, width);
        }
    }
  while (curr_max_width > required_mon_width);
#endif

  return required_mon_width;
}

static size_t
dev_ino_hash (void const *x, size_t table_size)
{
```

```
  struct dev_ino const *p = x;
  return (uintmax_t) p->st_ino % table_size;
}

static bool
dev_ino_compare (void const *x, void const *y)
{
  struct dev_ino const *a = x;
  struct dev_ino const *b = y;
  return SAME_INODE (*a, *b) ? true : false;
}

static void
dev_ino_free (void *x)
{
  free (x);
}

/* Add the device/inode pair (P->st_dev/P->st_ino) to the set of
   active directories.  Return true if there is already a matching
   entry in the table.  */

static bool
visit_dir (dev_t dev, ino_t ino)
{
  struct dev_ino *ent;
  struct dev_ino *ent_from_table;
  bool found_match;

  ent = xmalloc (sizeof *ent);
  ent->st_ino = ino;
  ent->st_dev = dev;

  /* Attempt to insert this entry into the table.  */
  ent_from_table = hash_insert (active_dir_set, ent);

  if (ent_from_table == NULL)
    {
      /* Insertion failed due to lack of memory.  */
      xalloc_die ();
    }

  found_match = (ent_from_table != ent);

  if (found_match)
    {
      /* ent was not inserted, so free it.  */
      free (ent);
    }

  return found_match;
}

static void
free_pending_ent (struct pending *p)
{
  free (p->name);
  free (p->realname);
  free (p);
}

static bool
is_colored (enum indicator_no type)
{
  size_t len = color_indicator[type].len;
  char const *s = color_indicator[type].string;
  return ! (len == 0
            || (len == 1 && STRNCMP_LIT (s, "0") == 0)
            || (len == 2 && STRNCMP_LIT (s, "00") == 0));
}

static void
restore_default_color (void)
{
  put_indicator (&color_indicator[C_LEFT]);
  put_indicator (&color_indicator[C_RIGHT]);
}

static void
set_normal_color (void)
{
  if (print_with_color && is_colored (C_NORM))
    {
      put_indicator (&color_indicator[C_LEFT]);
      put_indicator (&color_indicator[C_NORM]);
      put_indicator (&color_indicator[C_RIGHT]);
    }
}

/* An ordinary signal was received; arrange for the program to exit.  */

static void
sighandler (int sig)
{
  if (! SA_NOCLDSTOP)
    signal (sig, SIG_IGN);
  if (! interrupt_signal)
    interrupt_signal = sig;
```

```
}

/* A SIGTSTP was received; arrange for the program to suspend itself.  */

static void
stophandler (int sig)
{
  if (! SA_NOCLDSTOP)
    signal (sig, stophandler);
  if (! interrupt_signal)
    stop_signal_count++;
}

/* Process any pending signals.  If signals are caught, this function
   should be called periodically.  Ideally there should never be an
   unbounded amount of time when signals are not being processed.
   Signal handling can restore the default colors, so callers must
   immediately change colors after invoking this function.  */

static void
process_signals (void)
{
  while (interrupt_signal || stop_signal_count)
    {
      int sig;
      int stops;
      sigset_t oldset;

      if (used_color)
        restore_default_color ();
      fflush (stdout);

      sigprocmask (SIG_BLOCK, &caught_signals, &oldset);

      /* Reload interrupt_signal and stop_signal_count, in case a new
         signal was handled before sigprocmask took effect.  */
      sig = interrupt_signal;
      stops = stop_signal_count;

      /* SIGTSTP is special, since the application can receive that signal
         more than once.  In this case, don't set the signal handler to the
         default.  Instead, just raise the uncatchable SIGSTOP.  */
      if (stops)
        {
          stop_signal_count = stops - 1;
          sig = SIGSTOP;
        }
      else
        signal (sig, SIG_DFL);

      /* Exit or suspend the program.  */
      raise (sig);
      sigprocmask (SIG_SETMASK, &oldset, NULL);

      /* If execution reaches here, then the program has been
         continued (after being suspended).  */
    }
}

int
main (int argc, char **argv)
{
  int i;
  struct pending *thispend;
  int n_files;

  /* The signals that are trapped, and the number of such signals.  */
  static int const sig[] =
    {
      /* This one is handled specially.  */
      SIGTSTP,

      /* The usual suspects.  */
      SIGALRM, SIGHUP, SIGINT, SIGPIPE, SIGQUIT, SIGTERM,
#ifdef SIGPOLL
      SIGPOLL,
#endif
#ifdef SIGPROF
      SIGPROF,
#endif
#ifdef SIGVTALRM
      SIGVTALRM,
#endif
#ifdef SIGXCPU
      SIGXCPU,
#endif
#ifdef SIGXFSZ
      SIGXFSZ,
#endif
    };
  enum { nsigs = ARRAY_CARDINALITY (sig) };

#if ! SA_NOCLDSTOP
  bool caught_sig[nsigs];
#endif

  initialize_main (&argc, &argv);
  set_program_name (argv[0]);
```

```
  setlocale (LC_ALL, "");
  bindtextdomain (PACKAGE, LOCALEDIR);
  textdomain (PACKAGE);

  initialize_exit_failure (LS_FAILURE);
  atexit (close_stdout);

  assert (ARRAY_CARDINALITY (color_indicator) + 1
          == ARRAY_CARDINALITY (indicator_name));

  exit_status = EXIT_SUCCESS;
  print_dir_name = true;
  pending_dirs = NULL;

  current_time.tv_sec = TYPE_MINIMUM (time_t);
  current_time.tv_nsec = -1;

  i = decode_switches (argc, argv);

  if (print_with_color)
    parse_ls_color ();

  /* Test print_with_color again, because the call to parse_ls_color
     may have just reset it -- e.g., if LS_COLORS is invalid.  */
  if (print_with_color)
    {
      /* Avoid following symbolic links when possible.  */
      if (is_colored (C_ORPHAN)
          || (is_colored (C_EXEC) && color_symlink_as_referent)
          || (is_colored (C_MISSING) && format == long_format))
        check_symlink_color = true;

      /* If the standard output is a controlling terminal, watch out
         for signals, so that the colors can be restored to the
         default state if "ls" is suspended or interrupted.  */

      if (0 <= tcgetpgrp (STDOUT_FILENO))
        {
          int j;
#if SA_NOCLDSTOP
          struct sigaction act;

          sigemptyset (&caught_signals);
          for (j = 0; j < nsigs; j++)
            {
              sigaction (sig[j], NULL, &act);
              if (act.sa_handler != SIG_IGN)
                sigaddset (&caught_signals, sig[j]);
            }

          act.sa_mask = caught_signals;
          act.sa_flags = SA_RESTART;

          for (j = 0; j < nsigs; j++)
            if (sigismember (&caught_signals, sig[j]))
              {
                act.sa_handler = sig[j] == SIGTSTP ? stophandler : sighandler;
                sigaction (sig[j], &act, NULL);
              }
#else
          for (j = 0; j < nsigs; j++)
            {
              caught_sig[j] = (signal (sig[j], SIG_IGN) != SIG_IGN);
              if (caught_sig[j])
                {
                  signal (sig[j], sig[j] == SIGTSTP ? stophandler : sighandler);
                  siginterrupt (sig[j], 0);
                }
            }
#endif
        }
    }

  if (dereference == DEREF_UNDEFINED)
    dereference = ((immediate_dirs
                    || indicator_style == classify
                    || format == long_format)
                   ? DEREF_NEVER
                   : DEREF_COMMAND_LINE_SYMLINK_TO_DIR);

  /* When using -R, initialize a data structure we'll use to
     detect any directory cycles.  */
  if (recursive)
    {
      active_dir_set = hash_initialize (INITIAL_TABLE_SIZE, NULL,
                                        dev_ino_hash,
                                        dev_ino_compare,
                                        dev_ino_free);
      if (active_dir_set == NULL)
        xalloc_die ();

      obstack_init (&dev_ino_obstack);
    }

  format_needs_stat = sort_type == sort_time || sort_type == sort_size
    || format == long_format
    || print_scontext
    || print_block_size;
```

```
format_needs_type = (! format_needs_stat
                     && (recursive
                         || print_with_color
                         || indicator_style != none
                         || directories_first));

if (dired)
  {
    obstack_init (&dired_obstack);
    obstack_init (&subdired_obstack);
  }

cwd_n_alloc = 100;
cwd_file = xnmalloc (cwd_n_alloc, sizeof *cwd_file);
cwd_n_used = 0;

clear_files ();

n_files = argc - i;

if (n_files <= 0)
  {
    if (immediate_dirs)
      gobble_file (".", directory, NOT_AN_INODE_NUMBER, true, "");
    else
      queue_directory (".", NULL, true);
  }
else
  do
    gobble_file (argv[i++], unknown, NOT_AN_INODE_NUMBER, true, "");
  while (i < argc);

if (cwd_n_used)
  {
    sort_files ();
    if (!immediate_dirs)
      extract_dirs_from_files (NULL, true);
    /* 'cwd_n_used' might be zero now.  */
  }

/* In the following if/else blocks, it is sufficient to test 'pending_dirs'
   (and not pending_dirs->name) because there may be no markers in the queue
   at this point.  A marker may be enqueued when extract_dirs_from_files is
   called with a non-empty string or via print_dir.  */
if (cwd_n_used)
  {
    print_current_files ();
    if (pending_dirs)
      DIRED_PUTCHAR ('\n');
  }
else if (n_files <= 1 && pending_dirs && pending_dirs->next == 0)
  print_dir_name = false;

while (pending_dirs)
  {
    thispend = pending_dirs;
    pending_dirs = pending_dirs->next;

    if (LOOP_DETECT)
      {
        if (thispend->name == NULL)
          {
            /* thispend->name == NULL means this is a marker entry
               indicating we've finished processing the directory.
               Use its dev/ino numbers to remove the corresponding
               entry from the active_dir_set hash table.  */
            struct dev_ino di = dev_ino_pop ();
            struct dev_ino *found = hash_delete (active_dir_set, &di);
            /* ASSERT_MATCHING_DEV_INO (thispend->realname, di); */
            assert (found);
            dev_ino_free (found);
            free_pending_ent (thispend);
            continue;
          }
      }

    print_dir (thispend->name, thispend->realname,
               thispend->command_line_arg);

    free_pending_ent (thispend);
    print_dir_name = true;
  }

if (print_with_color)
  {
    int j;

    if (used_color)
      {
        /* Skip the restore when it would be a no-op, i.e.,
           when left is "\033[" and right is "m".  */
        if (!(color_indicator[C_LEFT].len == 2
              && memcmp (color_indicator[C_LEFT].string, "\033[", 2) == 0
              && color_indicator[C_RIGHT].len == 1
              && color_indicator[C_RIGHT].string[0] == 'm'))
          restore_default_color ();
      }
    fflush (stdout);
```

```
      /* Restore the default signal handling.  */
#if SA_NOCLDSTOP
      for (j = 0; j < nsigs; j++)
        if (sigismember (&caught_signals, sig[j]))
          signal (sig[j], SIG_DFL);
#else
      for (j = 0; j < nsigs; j++)
        if (caught_sig[j])
          signal (sig[j], SIG_DFL);
#endif

      /* Act on any signals that arrived before the default was restored.
         This can process signals out of order, but there doesn't seem to
         be an easy way to do them in order, and the order isn't that
         important anyway.  */
      for (j = stop_signal_count; j; j--)
        raise (SIGSTOP);
      j = interrupt_signal;
      if (j)
        raise (j);
    }

  if (dired)
    {
      /* No need to free these since we're about to exit.  */
      dired_dump_obstack ("//DIRED//", &dired_obstack);
      dired_dump_obstack ("//SUBDIRED//", &subdired_obstack);
      printf ("//DIRED-OPTIONS// --quoting-style=%s\n",
              quoting_style_args[get_quoting_style (filename_quoting_options)]);
    }

  if (LOOP_DETECT)
    {
      assert (hash_get_n_entries (active_dir_set) == 0);
      hash_free (active_dir_set);
    }

  return exit_status;
}

/* Set all the option flags according to the switches specified.
   Return the index of the first non-option argument.  */

static int
decode_switches (int argc, char **argv)
{
  char *time_style_option = NULL;

  bool sort_type_specified = false;
  bool kibibytes_specified = false;

  qmark_funny_chars = false;

  /* initialize all switches to default settings */

  switch (ls_mode)
    {
    case LS_MULTI_COL:
      /* This is for the 'dir' program.  */
      format = many_per_line;
      set_quoting_style (NULL, escape_quoting_style);
      break;

    case LS_LONG_FORMAT:
      /* This is for the 'vdir' program.  */
      format = long_format;
      set_quoting_style (NULL, escape_quoting_style);
      break;

    case LS_LS:
      /* This is for the 'ls' program.  */
      if (isatty (STDOUT_FILENO))
        {
          format = many_per_line;
          /* See description of qmark_funny_chars, above.  */
          qmark_funny_chars = true;
        }
      else
        {
          format = one_per_line;
          qmark_funny_chars = false;
        }
      break;

    default:
      abort ();
    }

  time_type = time_mtime;
  sort_type = sort_name;
  sort_reverse = false;
  numeric_ids = false;
  print_block_size = false;
  indicator_style = none;
  print_inode = false;
  dereference = DEREF_UNDEFINED;
  recursive = false;
```

```
  immediate_dirs = false;
  ignore_mode = IGNORE_DEFAULT;
  ignore_patterns = NULL;
  hide_patterns = NULL;
  print_scontext = false;

  getenv_quoting_style ();

  line_length = 80;
  {
    char const *p = getenv ("COLUMNS");
    if (p && *p)
      {
        unsigned long int tmp_ulong;
        if (xstrtoul (p, NULL, 0, &tmp_ulong, NULL) == LONGINT_OK
            && 0 < tmp_ulong && tmp_ulong <= SIZE_MAX)
          {
            line_length = tmp_ulong;
          }
        else
          {
            error (0, 0,
               _("ignoring invalid width in environment variable COLUMNS: %s"),
                   quotearg (p));
          }
      }
  }

#ifdef TIOCGWINSZ
  {
    struct winsize ws;

    if (ioctl (STDOUT_FILENO, TIOCGWINSZ, &ws) != -1
        && 0 < ws.ws_col && ws.ws_col == (size_t) ws.ws_col)
      line_length = ws.ws_col;
  }
#endif

  {
    char const *p = getenv ("TABSIZE");
    tabsize = 8;
    if (p)
      {
        unsigned long int tmp_ulong;
        if (xstrtoul (p, NULL, 0, &tmp_ulong, NULL) == LONGINT_OK
            && tmp_ulong <= SIZE_MAX)
          {
            tabsize = tmp_ulong;
          }
        else
          {
            error (0, 0,
              _("ignoring invalid tab size in environment variable TABSIZE: %s"),
                   quotearg (p));
          }
      }
  }

  while (true)
    {
      int oi = -1;
      int c = getopt_long (argc, argv,
                           "abcdfghiklmnopqrstuvw:xABCDFGHI:LNQRST:UXZ1",
                           long_options, &oi);
      if (c == -1)
        break;

      switch (c)
        {
        case 'a':
          ignore_mode = IGNORE_MINIMAL;
          break;

        case 'b':
          set_quoting_style (NULL, escape_quoting_style);
          break;

        case 'c':
          time_type = time_ctime;
          break;

        case 'd':
          immediate_dirs = true;
          break;

        case 'f':
          /* Same as enabling -a -U and disabling -l -s.  */
          ignore_mode = IGNORE_MINIMAL;
          sort_type = sort_none;
          sort_type_specified = true;
          /* disable -l */
          if (format == long_format)
            format = (isatty (STDOUT_FILENO) ? many_per_line : one_per_line);
          print_block_size = false;     /* disable -s */
          print_with_color = false;     /* disable --color */
          break;

        case FILE_TYPE_INDICATOR_OPTION: /* --file-type */
```

```
  indicator_style = file_type;
  break;

case 'g':
  format = long_format;
  print_owner = false;
  break;

case 'h':
  file_human_output_opts = human_output_opts =
    human_autoscale | human_SI | human_base_1024;
  file_output_block_size = output_block_size = 1;
  break;

case 'i':
  print_inode = true;
  break;

case 'k':
  kibibytes_specified = true;
  break;

case 'l':
  format = long_format;
  break;

case 'm':
  format = with_commas;
  break;

case 'n':
  numeric_ids = true;
  format = long_format;
  break;

case 'o':  /* Just like -l, but don't display group info.  */
  format = long_format;
  print_group = false;
  break;

case 'p':
  indicator_style = slash;
  break;

case 'q':
  qmark_funny_chars = true;
  break;

case 'r':
  sort_reverse = true;
  break;

case 's':
  print_block_size = true;
  break;

case 't':
  sort_type = sort_time;
  sort_type_specified = true;
  break;

case 'u':
  time_type = time_atime;
  break;

case 'v':
  sort_type = sort_version;
  sort_type_specified = true;
  break;

case 'w':
  line_length = xnumtoumax (optarg, 0, 1, SIZE_MAX, "",
                            _("invalid line width"), LS_FAILURE);
  break;

case 'x':
  format = horizontal;
  break;

case 'A':
  if (ignore_mode == IGNORE_DEFAULT)
    ignore_mode = IGNORE_DOT_AND_DOTDOT;
  break;

case 'B':
  add_ignore_pattern ("*~");
  add_ignore_pattern (".*~");
  break;

case 'C':
  format = many_per_line;
  break;

case 'D':
  dired = true;
  break;

case 'F':
```

```
  indicator_style = classify;
  break;

case 'G':               /* inhibit display of group info */
  print_group = false;
  break;

case 'H':
  dereference = DEREF_COMMAND_LINE_ARGUMENTS;
  break;

case DEREFERENCE_COMMAND_LINE_SYMLINK_TO_DIR_OPTION:
  dereference = DEREF_COMMAND_LINE_SYMLINK_TO_DIR;
  break;

case 'I':
  add_ignore_pattern (optarg);
  break;

case 'L':
  dereference = DEREF_ALWAYS;
  break;

case 'N':
  set_quoting_style (NULL, literal_quoting_style);
  break;

case 'Q':
  set_quoting_style (NULL, c_quoting_style);
  break;

case 'R':
  recursive = true;
  break;

case 'S':
  sort_type = sort_size;
  sort_type_specified = true;
  break;

case 'T':
  tabsize = xnumtoumax (optarg, 0, 0, SIZE_MAX, "",
                        _("invalid tab size"), LS_FAILURE);
  break;

case 'U':
  sort_type = sort_none;
  sort_type_specified = true;
  break;

case 'X':
  sort_type = sort_extension;
  sort_type_specified = true;
  break;

case '1':
  /* -1 has no effect after -l.  */
  if (format != long_format)
    format = one_per_line;
  break;

case AUTHOR_OPTION:
  print_author = true;
  break;

case HIDE_OPTION:
  {
    struct ignore_pattern *hide = xmalloc (sizeof *hide);
    hide->pattern = optarg;
    hide->next = hide_patterns;
    hide_patterns = hide;
  }
  break;

case SORT_OPTION:
  sort_type = XARGMATCH ("--sort", optarg, sort_args, sort_types);
  sort_type_specified = true;
  break;

case GROUP_DIRECTORIES_FIRST_OPTION:
  directories_first = true;
  break;

case TIME_OPTION:
  time_type = XARGMATCH ("--time", optarg, time_args, time_types);
  break;

case FORMAT_OPTION:
  format = XARGMATCH ("--format", optarg, format_args, format_types);
  break;

case FULL_TIME_OPTION:
  format = long_format;
  time_style_option = bad_cast ("full-iso");
  break;

case COLOR_OPTION:
  {
```

```
          int i;
          if (optarg)
            i = XARGMATCH ("--color", optarg, color_args, color_types);
          else
            /* Using --color with no argument is equivalent to using
               --color=always.  */
            i = color_always;

          print_with_color = (i == color_always
                              || (i == color_if_tty
                                  && isatty (STDOUT_FILENO)));

          if (print_with_color)
            {
              /* Don't use TAB characters in output.  Some terminal
                 emulators can't handle the combination of tabs and
                 color codes on the same line.  */
              tabsize = 0;
            }
          break;
        }

      case INDICATOR_STYLE_OPTION:
        indicator_style = XARGMATCH ("--indicator-style", optarg,
                                     indicator_style_args,
                                     indicator_style_types);
        break;

      case QUOTING_STYLE_OPTION:
        set_quoting_style (NULL,
                           XARGMATCH ("--quoting-style", optarg,
                                      quoting_style_args,
                                      quoting_style_vals));
        break;

      case TIME_STYLE_OPTION:
        time_style_option = optarg;
        break;

      case SHOW_CONTROL_CHARS_OPTION:
        qmark_funny_chars = false;
        break;

      case BLOCK_SIZE_OPTION:
        {
          enum strtol_error e = human_options (optarg, &human_output_opts,
                                               &output_block_size);
          if (e != LONGINT_OK)
            xstrtol_fatal (e, oi, 0, long_options, optarg);
          file_human_output_opts = human_output_opts;
          file_output_block_size = output_block_size;
        }
        break;

      case SI_OPTION:
        file_human_output_opts = human_output_opts =
          human_autoscale | human_SI;
        file_output_block_size = output_block_size = 1;
        break;

      case 'Z':
        print_scontext = true;
        break;

      case_GETOPT_HELP_CHAR;

      case_GETOPT_VERSION_CHAR (PROGRAM_NAME, AUTHORS);

      default:
        usage (LS_FAILURE);
      }
  }

if (! output_block_size)
  {
    char const *ls_block_size = getenv ("LS_BLOCK_SIZE");
    human_options (ls_block_size,
                   &human_output_opts, &output_block_size);
    if (ls_block_size || getenv ("BLOCK_SIZE"))
      {
        file_human_output_opts = human_output_opts;
        file_output_block_size = output_block_size;
      }
    if (kibibytes_specified)
      {
        human_output_opts = 0;
        output_block_size = 1024;
      }
  }

max_idx = MAX (1, line_length / MIN_COLUMN_WIDTH);

filename_quoting_options = clone_quoting_options (NULL);
if (get_quoting_style (filename_quoting_options) == escape_quoting_style)
  set_char_quoting (filename_quoting_options, ' ', 1);
if (file_type <= indicator_style)
  {
    char const *p;
```

```
    for (p = &"*=>@|"[indicator_style - file_type]; *p; p++)
      set_char_quoting (filename_quoting_options, *p, 1);
  }

dirname_quoting_options = clone_quoting_options (NULL);
set_char_quoting (dirname_quoting_options, ':', 1);

/* --dired is meaningful only with --format=long (-l).
   Otherwise, ignore it.  FIXME: warn about this?
   Alternatively, make --dired imply --format=long?  */
if (dired && format != long_format)
  dired = false;

/* If -c or -u is specified and not -l (or any other option that implies -l),
   and no sort-type was specified, then sort by the ctime (-c) or atime (-u).
   The behavior of ls when using either -c or -u but with neither -l nor -t
   appears to be unspecified by POSIX.  So, with GNU ls, '-u' alone means
   sort by atime (this is the one that's not specified by the POSIX spec),
   -lu means show atime and sort by name, -lut means show atime and sort
   by atime.  */

if ((time_type == time_ctime || time_type == time_atime)
    && !sort_type_specified && format != long_format)
  {
    sort_type = sort_time;
  }

if (format == long_format)
  {
    char *style = time_style_option;
    static char const posix_prefix[] = "posix-";

    if (! style)
      if (! (style = getenv ("TIME_STYLE")))
        style = bad_cast ("locale");

    while (STREQ_LEN (style, posix_prefix, sizeof posix_prefix - 1))
      {
        if (! hard_locale (LC_TIME))
          return optind;
        style += sizeof posix_prefix - 1;
      }

    if (*style == '+')
      {
        char *p0 = style + 1;
        char *p1 = strchr (p0, '\n');
        if (! p1)
          p1 = p0;
        else
          {
            if (strchr (p1 + 1, '\n'))
              error (LS_FAILURE, 0, _("invalid time style format %s"),
                     quote (p0));
            *p1++ = '\0';
          }
        long_time_format[0] = p0;
        long_time_format[1] = p1;
      }
    else
      {
        ptrdiff_t res = argmatch (style, time_style_args,
                                  (char const *) time_style_types,
                                  sizeof (*time_style_types));
        if (res < 0)
          {
            /* This whole block used to be a simple use of XARGMATCH.
               but that didn't print the "posix-"-prefixed variants or
               the "+"-prefixed format string option upon failure.  */
            argmatch_invalid ("time style", style, res);

            /* The following is a manual expansion of argmatch_valid,
               but with the added "+ ..." description and the [posix-]
               prefixes prepended.  Note that this simplification works
               only because all four existing time_style_types values
               are distinct.  */
            fputs (_("Valid arguments are:\n"), stderr);
            char const *const *p = time_style_args;
            while (*p)
              fprintf (stderr, "  - [posix-]%s\n", *p++);
            fputs (_("  - +FORMAT (e.g., +%H:%M) for a 'date'-style"
                     " format\n"), stderr);
            usage (LS_FAILURE);
          }
        switch (res)
          {
          case full_iso_time_style:
            long_time_format[0] = long_time_format[1] =
              "%Y-%m-%d %H:%M:%S.%N %z";
            break;

          case long_iso_time_style:
            long_time_format[0] = long_time_format[1] = "%Y-%m-%d %H:%M";
            break;

          case iso_time_style:
            long_time_format[0] = "%Y-%m-%d ";
            long_time_format[1] = "%m-%d %H:%M";
```

```
              break;

            case locale_time_style:
              if (hard_locale (LC_TIME))
                {
                  int i;
                  for (i = 0; i < 2; i++)
                    long_time_format[i] =
                      dcgettext (NULL, long_time_format[i], LC_TIME);
                }
            }
        }

      /* Note we leave %5b etc. alone so user widths/flags are honored.  */
      if (strstr (long_time_format[0], "%b")
          || strstr (long_time_format[1], "%b"))
        if (!abmon_init ())
          error (0, 0, _("error initializing month strings"));
    }

  return optind;
}

/* Parse a string as part of the LS_COLORS variable; this may involve
   decoding all kinds of escape characters.  If equals_end is set an
   unescaped equal sign ends the string, otherwise only a : or \0
   does.  Set *OUTPUT_COUNT to the number of bytes output.  Return
   true if successful.

   The resulting string is *not* null-terminated, but may contain
   embedded nulls.

   Note that both dest and src are char **; on return they point to
   the first free byte after the array and the character that ended
   the input string, respectively.  */

static bool
get_funky_string (char **dest, const char **src, bool equals_end,
                  size_t *output_count)
{
  char num;                     /* For numerical codes */
  size_t count;                 /* Something to count with */
  enum {
    ST_GND, ST_BACKSLASH, ST_OCTAL, ST_HEX, ST_CARET, ST_END, ST_ERROR
  } state;
  const char *p;
  char *q;

  p = *src;                     /* We don't want to double-indirect */
  q = *dest;                    /* the whole darn time.  */

  count = 0;                    /* No characters counted in yet.  */
  num = 0;

  state = ST_GND;               /* Start in ground state.  */
  while (state < ST_END)
    {
      switch (state)
        {
        case ST_GND:            /* Ground state (no escapes) */
          switch (*p)
            {
            case ':':
            case '\0':
              state = ST_END;   /* End of string */
              break;
            case '\\':
              state = ST_BACKSLASH; /* Backslash scape sequence */
              ++p;
              break;
            case '^':
              state = ST_CARET; /* Caret escape */
              ++p;
              break;
            case '=':
              if (equals_end)
                {
                  state = ST_END; /* End */
                  break;
                }
              /* else fall through */
            default:
              *(q++) = *(p++);
              ++count;
              break;
            }
          break;

        case ST_BACKSLASH:      /* Backslash escaped character */
          switch (*p)
            {
            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
```

```
    case '7':
      state = ST_OCTAL; /* Octal sequence */
      num = *p - '0';
      break;
    case 'x':
    case 'X':
      state = ST_HEX;   /* Hex sequence */
      num = 0;
      break;
    case 'a':           /* Bell */
      num = '\a';
      break;
    case 'b':           /* Backspace */
      num = '\b';
      break;
    case 'e':           /* Escape */
      num = 27;
      break;
    case 'f':           /* Form feed */
      num = '\f';
      break;
    case 'n':           /* Newline */
      num = '\n';
      break;
    case 'r':           /* Carriage return */
      num = '\r';
      break;
    case 't':           /* Tab */
      num = '\t';
      break;
    case 'v':           /* Vtab */
      num = '\v';
      break;
    case '?':           /* Delete */
      num = 127;
      break;
    case '_':           /* Space */
      num = ' ';
      break;
    case '\0':          /* End of string */
      state = ST_ERROR; /* Error! */
      break;
    default:            /* Escaped character like \ ^ : = */
      num = *p;
      break;
    }
  if (state == ST_BACKSLASH)
    {
      *(q++) = num;
      ++count;
      state = ST_GND;
    }
  ++p;
  break;

case ST_OCTAL:          /* Octal sequence */
  if (*p < '0' || *p > '7')
    {
      *(q++) = num;
      ++count;
      state = ST_GND;
    }
  else
    num = (num << 3) + (*(p++) - '0');
  break;

case ST_HEX:            /* Hex sequence */
  switch (*p)
    {
    case '0':
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
      num = (num << 4) + (*(p++) - '0');
      break;
    case 'a':
    case 'b':
    case 'c':
    case 'd':
    case 'e':
    case 'f':
      num = (num << 4) + (*(p++) - 'a') + 10;
      break;
    case 'A':
    case 'B':
    case 'C':
    case 'D':
    case 'E':
    case 'F':
      num = (num << 4) + (*(p++) - 'A') + 10;
      break;
    default:
```

```
              *(q++) = num;
              ++count;
              state = ST_GND;
              break;
            }
          break;

        case ST_CARET:          /* Caret escape */
          state = ST_GND;       /* Should be the next state... */
          if (*p >= '@' && *p <= '~')
            {
              *(q++) = *(p++) & 037;
              ++count;
            }
          else if (*p == '?')
            {
              *(q++) = 127;
              ++count;
            }
          else
            state = ST_ERROR;
          break;

        default:
          abort ();
        }
    }

  *dest = q;
  *src = p;
  *output_count = count;

  return state != ST_ERROR;
}

enum parse_state
  {
    PS_START = 1,
    PS_2,
    PS_3,
    PS_4,
    PS_DONE,
    PS_FAIL
  };


/* Check if the content of TERM is a valid name in dircolors.  */

static bool
known_term_type (void)
{
  char const *term = getenv ("TERM");
  if (! term || ! *term)
    return false;

  char const *line = G_line;
  while (line - G_line < sizeof (G_line))
    {
      if (STRNCMP_LIT (line, "TERM ") == 0)
        {
          if (STREQ (term, line + 5))
            return true;
        }
      line += strlen (line) + 1;
    }

  return false;
}

static void
parse_ls_color (void)
{
  const char *p;                /* Pointer to character being parsed */
  char *buf;                    /* color_buf buffer pointer */
  int ind_no;                   /* Indicator number */
  char label[3];                /* Indicator label */
  struct color_ext_type *ext;   /* Extension we are working on */

  if ((p = getenv ("LS_COLORS")) == NULL || *p == '\0')
    {
      /* LS_COLORS takes precedence, but if that's not set then
         honor the COLORTERM and TERM env variables so that
         we only go with the internal ANSI color codes if the
         former is non empty or the latter is set to a known value.  */
      char const *colorterm = getenv ("COLORTERM");
      if (! (colorterm && *colorterm) && ! known_term_type ())
        print_with_color = false;
      return;
    }

  ext = NULL;
  strcpy (label, "??");

  /* This is an overly conservative estimate, but any possible
     LS_COLORS string will *not* generate a color_buf longer than
     itself, so it is a safe way of allocating a buffer in
     advance.  */
  buf = color_buf = xstrdup (p);
```

```
 enum parse_state state = PS_START;
 while (true)
   {
     switch (state)
       {
       case PS_START:          /* First label character */
         switch (*p)
           {
           case ':':
             ++p;
             break;

           case '*':
             /* Allocate new extension block and add to head of
                linked list (this way a later definition will
                override an earlier one, which can be useful for
                having terminal-specific defs override global).  */

             ext = xmalloc (sizeof *ext);
             ext->next = color_ext_list;
             color_ext_list = ext;

             ++p;
             ext->ext.string = buf;

             state = (get_funky_string (&buf, &p, true, &ext->ext.len)
                      ? PS_4 : PS_FAIL);
             break;

           case '\0':
             state = PS_DONE;  /* Done! */
             goto done;

           default:    /* Assume it is file type label */
             label[0] = *(p++);
             state = PS_2;
             break;
           }
         break;

       case PS_2:              /* Second label character */
         if (*p)
           {
             label[1] = *(p++);
             state = PS_3;
           }
         else
           state = PS_FAIL;    /* Error */
         break;

       case PS_3:              /* Equal sign after indicator label */
         state = PS_FAIL;      /* Assume failure...  */
         if (*(p++) == '=')/* It *should* be...  */
           {
             for (ind_no = 0; indicator_name[ind_no] != NULL; ++ind_no)
               {
                 if (STREQ (label, indicator_name[ind_no]))
                   {
                     color_indicator[ind_no].string = buf;
                     state = (get_funky_string (&buf, &p, false,
                                                &color_indicator[ind_no].len)
                              ? PS_START : PS_FAIL);
                     break;
                   }
               }
             if (state == PS_FAIL)
               error (0, 0, _("unrecognized prefix: %s"), quotearg (label));
           }
         break;

       case PS_4:              /* Equal sign after *.ext */
         if (*(p++) == '=')
           {
             ext->seq.string = buf;
             state = (get_funky_string (&buf, &p, false, &ext->seq.len)
                      ? PS_START : PS_FAIL);
           }
         else
           state = PS_FAIL;
         break;

       case PS_FAIL:
         goto done;

       default:
         abort ();
       }
   }
done:

 if (state == PS_FAIL)
   {
     struct color_ext_type *e;
     struct color_ext_type *e2;

     error (0, 0,
            _("unparsable value for LS_COLORS environment variable"));
```

```
      free (color_buf);
      for (e = color_ext_list; e != NULL; /* empty */)
        {
          e2 = e;
          e = e->next;
          free (e2);
        }
      print_with_color = false;
    }

  if (color_indicator[C_LINK].len == 6
      && !STRNCMP_LIT (color_indicator[C_LINK].string, "target"))
    color_symlink_as_referent = true;
}

/* Set the quoting style default if the environment variable
   QUOTING_STYLE is set.  */

static void
getenv_quoting_style (void)
{
  char const *q_style = getenv ("QUOTING_STYLE");
  if (q_style)
    {
      int i = ARGMATCH (q_style, quoting_style_args, quoting_style_vals);
      if (0 <= i)
        set_quoting_style (NULL, quoting_style_vals[i]);
      else
        error (0, 0,
       _("ignoring invalid value of environment variable QUOTING_STYLE: %s"),
               quotearg (q_style));
    }
}

/* Set the exit status to report a failure.  If SERIOUS, it is a
   serious failure; otherwise, it is merely a minor problem.  */

static void
set_exit_status (bool serious)
{
  if (serious)
    exit_status = LS_FAILURE;
  else if (exit_status == EXIT_SUCCESS)
    exit_status = LS_MINOR_PROBLEM;
}

/* Assuming a failure is serious if SERIOUS, use the printf-style
   MESSAGE to report the failure to access a file named FILE.  Assume
   errno is set appropriately for the failure.  */

static void
file_failure (bool serious, char const *message, char const *file)
{
  error (0, errno, message, quotearg_colon (file));
  set_exit_status (serious);
}

/* Request that the directory named NAME have its contents listed later.
   If REALNAME is nonzero, it will be used instead of NAME when the
   directory name is printed.  This allows symbolic links to directories
   to be treated as regular directories but still be listed under their
   real names.  NAME == NULL is used to insert a marker entry for the
   directory named in REALNAME.
   If NAME is non-NULL, we use its dev/ino information to save
   a call to stat -- when doing a recursive (-R) traversal.
   COMMAND_LINE_ARG means this directory was mentioned on the command line.  */

static void
queue_directory (char const *name, char const *realname, bool command_line_arg)
{
  struct pending *new = xmalloc (sizeof *new);
  new->realname = realname ? xstrdup (realname) : NULL;
  new->name = name ? xstrdup (name) : NULL;
  new->command_line_arg = command_line_arg;
  new->next = pending_dirs;
  pending_dirs = new;
}

/* Read directory NAME, and list the files in it.
   If REALNAME is nonzero, print its name instead of NAME;
   this is used for symbolic links to directories.
   COMMAND_LINE_ARG means this directory was mentioned on the command line.  */

static void
print_dir (char const *name, char const *realname, bool command_line_arg)
{
  DIR *dirp;
  struct dirent *next;
  uintmax_t total_blocks = 0;
  static bool first = true;

  errno = 0;
  dirp = opendir (name);
  if (!dirp)
    {
      file_failure (command_line_arg, _("cannot open directory %s"), name);
      return;
    }
```

```
  if (LOOP_DETECT)
    {
      struct stat dir_stat;
      int fd = dirfd (dirp);

      /* If dirfd failed, endure the overhead of using stat.  */
      if ((0 <= fd
           ? fstat (fd, &dir_stat)
           : stat (name, &dir_stat)) < 0)
        {
          file_failure (command_line_arg,
                        _("cannot determine device and inode of %s"), name);
          closedir (dirp);
          return;
        }

      /* If we've already visited this dev/inode pair, warn that
         we've found a loop, and do not process this directory.  */
      if (visit_dir (dir_stat.st_dev, dir_stat.st_ino))
        {
          error (0, 0, _("%s: not listing already-listed directory"),
                 quotearg_colon (name));
          closedir (dirp);
          set_exit_status (true);
          return;
        }

      dev_ino_push (dir_stat.st_dev, dir_stat.st_ino);
    }

  if (recursive || print_dir_name)
    {
      if (!first)
        DIRED_PUTCHAR ('\n');
      first = false;
      DIRED_INDENT ();
      PUSH_CURRENT_DIRED_POS (&subdired_obstack);
      dired_pos += quote_name (stdout, realname ? realname : name,
                               dirname_quoting_options, NULL);
      PUSH_CURRENT_DIRED_POS (&subdired_obstack);
      DIRED_FPUTS_LITERAL (":\n", stdout);
    }

  /* Read the directory entries, and insert the subfiles into the 'cwd_file'
     table.  */

  clear_files ();

  while (1)
    {
      /* Set errno to zero so we can distinguish between a readdir failure
         and when readdir simply finds that there are no more entries.  */
      errno = 0;
      next = readdir (dirp);
      if (next)
        {
          if (! file_ignored (next->d_name))
            {
              enum filetype type = unknown;

#if HAVE_STRUCT_DIRENT_D_TYPE
              switch (next->d_type)
                {
                case DT_BLK:  type = blockdev;          break;
                case DT_CHR:  type = chardev;           break;
                case DT_DIR:  type = directory;         break;
                case DT_FIFO: type = fifo;              break;
                case DT_LNK:  type = symbolic_link;     break;
                case DT_REG:  type = normal;            break;
                case DT_SOCK: type = sock;              break;
# ifdef DT_WHT
                case DT_WHT:  type = whiteout;          break;
# endif
                }
#endif
              total_blocks += gobble_file (next->d_name, type,
                                           RELIABLE_D_INO (next),
                                           false, name);

              /* In this narrow case, print out each name right away, so
                 ls uses constant memory while processing the entries of
                 this directory.  Useful when there are many (millions)
                 of entries in a directory.  */
              if (format == one_per_line && sort_type == sort_none
                  && !print_block_size && !recursive)
                {
                  /* We must call sort_files in spite of
                     "sort_type == sort_none" for its initialization
                     of the sorted_file vector.  */
                  sort_files ();
                  print_current_files ();
                  clear_files ();
                }
            }
        }
      else if (errno != 0)
        {
```

```
          file_failure (command_line_arg, _("reading directory %s"), name);
          if (errno != EOVERFLOW)
            break;
        }
      else
        break;

      /* When processing a very large directory, and since we've inhibited
         interrupts, this loop would take so long that ls would be annoyingly
         uninterruptible.  This ensures that it handles signals promptly.  */
      process_signals ();
    }

  if (closedir (dirp) != 0)
    {
      file_failure (command_line_arg, _("closing directory %s"), name);
      /* Don't return; print whatever we got.  */
    }

  /* Sort the directory contents.  */
  sort_files ();

  /* If any member files are subdirectories, perhaps they should have their
     contents listed rather than being mentioned here as files.  */

  if (recursive)
    extract_dirs_from_files (name, false);

  if (format == long_format || print_block_size)
    {
      const char *p;
      char buf[LONGEST_HUMAN_READABLE + 1];

      DIRED_INDENT ();
      p = _("total");
      DIRED_FPUTS (p, stdout, strlen (p));
      DIRED_PUTCHAR (' ');
      p = human_readable (total_blocks, buf, human_output_opts,
                          ST_NBLOCKSIZE, output_block_size);
      DIRED_FPUTS (p, stdout, strlen (p));
      DIRED_PUTCHAR ('\n');
    }

  if (cwd_n_used)
    print_current_files ();
}

/* Add 'pattern' to the list of patterns for which files that match are
   not listed.  */

static void
add_ignore_pattern (const char *pattern)
{
  struct ignore_pattern *ignore;

  ignore = xmalloc (sizeof *ignore);
  ignore->pattern = pattern;
  /* Add it to the head of the linked list.  */
  ignore->next = ignore_patterns;
  ignore_patterns = ignore;
}

/* Return true if one of the PATTERNS matches FILE.  */

static bool
patterns_match (struct ignore_pattern const *patterns, char const *file)
{
  struct ignore_pattern const *p;
  for (p = patterns; p; p = p->next)
    if (fnmatch (p->pattern, file, FNM_PERIOD) == 0)
      return true;
  return false;
}

/* Return true if FILE should be ignored.  */

static bool
file_ignored (char const *name)
{
  return ((ignore_mode != IGNORE_MINIMAL
           && name[0] == '.'
           && (ignore_mode == IGNORE_DEFAULT || ! name[1 + (name[1] == '.')]))
          || (ignore_mode == IGNORE_DEFAULT
              && patterns_match (hide_patterns, name))
          || patterns_match (ignore_patterns, name));
}

/* POSIX requires that a file size be printed without a sign, even
   when negative.  Assume the typical case where negative sizes are
   actually positive values that have wrapped around.  */

static uintmax_t
unsigned_file_size (off_t size)
{
  return size + (size < 0) * ((uintmax_t) OFF_T_MAX - OFF_T_MIN + 1);
}

#ifdef HAVE_CAP
```

```
/* Return true if NAME has a capability (see linux/capability.h) */
static bool
has_capability (char const *name)
{
  char *result;
  bool has_cap;

  cap_t cap_d = cap_get_file (name);
  if (cap_d == NULL)
    return false;

  result = cap_to_text (cap_d, NULL);
  cap_free (cap_d);
  if (!result)
    return false;

  /* check if human-readable capability string is empty */
  has_cap = !!*result;

  cap_free (result);
  return has_cap;
}
#else
static bool
has_capability (char const *name _GL_UNUSED)
{
  errno = ENOTSUP;
  return false;
}
#endif

/* Enter and remove entries in the table 'cwd_file'.  */

static void
free_ent (struct fileinfo *f)
{
  free (f->name);
  free (f->linkname);
  if (f->scontext != UNKNOWN_SECURITY_CONTEXT)
    {
      if (is_smack_enabled ())
        free (f->scontext);
      else
        freecon (f->scontext);
    }
}

/* Empty the table of files.  */
static void
clear_files (void)
{
  size_t i;

  for (i = 0; i < cwd_n_used; i++)
    {
      struct fileinfo *f = sorted_file[i];
      free_ent (f);
    }

  cwd_n_used = 0;
  any_has_acl = false;
  inode_number_width = 0;
  block_size_width = 0;
  nlink_width = 0;
  owner_width = 0;
  group_width = 0;
  author_width = 0;
  scontext_width = 0;
  major_device_number_width = 0;
  minor_device_number_width = 0;
  file_size_width = 0;
}

/* Return true if ERR implies lack-of-support failure by a
   getxattr-calling function like getfilecon or file_has_acl.  */
static bool
errno_unsupported (int err)
{
  return (err == EINVAL || err == ENOSYS || is_ENOTSUP (err));
}

/* Cache *getfilecon failure, when it's trivial to do so.
   Like getfilecon/lgetfilecon, but when F's st_dev says it's doesn't
   support getting the security context, fail with ENOTSUP immediately.  */
static int
getfilecon_cache (char const *file, struct fileinfo *f, bool deref)
{
  /* st_dev of the most recently processed device for which we've
     found that [l]getfilecon fails indicating lack of support.  */
  static dev_t unsupported_device;

  if (f->stat.st_dev == unsupported_device)
    {
      errno = ENOTSUP;
      return -1;
    }
  int r = 0;
#ifdef HAVE_SMACK
```

```
  if (is_smack_enabled ())
    r = smack_new_label_from_path (file, "security.SMACK64", deref,
                                   &f->scontext);
  else
#endif
    r = (deref
         ? getfilecon (file, &f->scontext)
         : lgetfilecon (file, &f->scontext));
  if (r < 0 && errno_unsupported (errno))
    unsupported_device = f->stat.st_dev;
  return r;
}

/* Cache file_has_acl failure, when it's trivial to do.
   Like file_has_acl, but when F's st_dev says it's on a file
   system lacking ACL support, return 0 with ENOTSUP immediately.  */
static int
file_has_acl_cache (char const *file, struct fileinfo *f)
{
  /* st_dev of the most recently processed device for which we've
     found that file_has_acl fails indicating lack of support.  */
  static dev_t unsupported_device;

  if (f->stat.st_dev == unsupported_device)
    {
      errno = ENOTSUP;
      return 0;
    }

  /* Zero errno so that we can distinguish between two 0-returning cases:
     "has-ACL-support, but only a default ACL" and "no ACL support". */
  errno = 0;
  int n = file_has_acl (file, &f->stat);
  if (n <= 0 && errno_unsupported (errno))
    unsupported_device = f->stat.st_dev;
  return n;
}

/* Cache has_capability failure, when it's trivial to do.
   Like has_capability, but when F's st_dev says it's on a file
   system lacking capability support, return 0 with ENOTSUP immediately.  */
static bool
has_capability_cache (char const *file, struct fileinfo *f)
{
  /* st_dev of the most recently processed device for which we've
     found that has_capability fails indicating lack of support.  */
  static dev_t unsupported_device;

  if (f->stat.st_dev == unsupported_device)
    {
      errno = ENOTSUP;
      return 0;
    }

  bool b = has_capability (file);
  if ( !b && errno_unsupported (errno))
    unsupported_device = f->stat.st_dev;
  return b;
}

/* Add a file to the current table of files.
   Verify that the file exists, and print an error message if it does not.
   Return the number of blocks that the file occupies.  */
static uintmax_t
gobble_file (char const *name, enum filetype type, ino_t inode,
             bool command_line_arg, char const *dirname)
{
  uintmax_t blocks = 0;
  struct fileinfo *f;

  /* An inode value prior to gobble_file necessarily came from readdir,
     which is not used for command line arguments.  */
  assert (! command_line_arg || inode == NOT_AN_INODE_NUMBER);

  if (cwd_n_used == cwd_n_alloc)
    {
      cwd_file = xnrealloc (cwd_file, cwd_n_alloc, 2 * sizeof *cwd_file);
      cwd_n_alloc *= 2;
    }

  f = &cwd_file[cwd_n_used];
  memset (f, '\0', sizeof *f);
  f->stat.st_ino = inode;
  f->filetype = type;

  if (command_line_arg
      || format_needs_stat
      /* When coloring a directory (we may know the type from
         direct.d_type), we have to stat it in order to indicate
         sticky and/or other-writable attributes.  */
      || (type == directory && print_with_color
          && (is_colored (C_OTHER_WRITABLE)
              || is_colored (C_STICKY)
              || is_colored (C_STICKY_OTHER_WRITABLE)))
      /* When dereferencing symlinks, the inode and type must come from
         stat, but readdir provides the inode and type of lstat.  */
      || ((print_inode || format_needs_type)
          && (type == symbolic_link || type == unknown)
```

```
      && (dereference == DEREF_ALWAYS
          || color_symlink_as_referent || check_symlink_color))
  /* Command line dereferences are already taken care of by the above
     assertion that the inode number is not yet known.  */
  || (print_inode && inode == NOT_AN_INODE_NUMBER)
  || (format_needs_type
      && (type == unknown || command_line_arg
          /* --indicator-style=classify (aka -F)
             requires that we stat each regular file
             to see if it's executable.  */
          || (type == normal && (indicator_style == classify
                                 /* This is so that --color ends up
                                    highlighting files with these mode
                                    bits set even when options like -F are
                                    not specified.  Note we do a redundant
                                    stat in the very unlikely case where
                                    C_CAP is set but not the others. */
                                 || (print_with_color
                                     && (is_colored (C_EXEC)
                                         || is_colored (C_SETUID)
                                         || is_colored (C_SETGID)
                                         || is_colored (C_CAP)))
                                 )))))

{
  /* Absolute name of this file.  */
  char *absolute_name;
  bool do_deref;
  int err;

  if (name[0] == '/' || dirname[0] == 0)
    absolute_name = (char *) name;
  else
    {
      absolute_name = alloca (strlen (name) + strlen (dirname) + 2);
      attach (absolute_name, dirname, name);
    }

  switch (dereference)
    {
    case DEREF_ALWAYS:
      err = stat (absolute_name, &f->stat);
      do_deref = true;
      break;

    case DEREF_COMMAND_LINE_ARGUMENTS:
    case DEREF_COMMAND_LINE_SYMLINK_TO_DIR:
      if (command_line_arg)
        {
          bool need_lstat;
          err = stat (absolute_name, &f->stat);
          do_deref = true;

          if (dereference == DEREF_COMMAND_LINE_ARGUMENTS)
            break;

          need_lstat = (err < 0
                        ? errno == ENOENT
                        : ! S_ISDIR (f->stat.st_mode));
          if (!need_lstat)
            break;

          /* stat failed because of ENOENT, maybe indicating a dangling
             symlink.  Or stat succeeded, ABSOLUTE_NAME does not refer to a
             directory, and --dereference-command-line-symlink-to-dir is
             in effect.  Fall through so that we call lstat instead.  */
        }

    default: /* DEREF_NEVER */
      err = lstat (absolute_name, &f->stat);
      do_deref = false;
      break;
    }

  if (err != 0)
    {
      /* Failure to stat a command line argument leads to
         an exit status of 2.  For other files, stat failure
         provokes an exit status of 1.  */
      file_failure (command_line_arg,
                    _("cannot access %s"), absolute_name);
      if (command_line_arg)
        return 0;

      f->name = xstrdup (name);
      cwd_n_used++;

      return 0;
    }

  f->stat_ok = true;

  /* Note has_capability() adds around 30% runtime to 'ls --color'  */
  if ((type == normal || S_ISREG (f->stat.st_mode))
      && print_with_color && is_colored (C_CAP))
    f->has_capability = has_capability_cache (absolute_name, f);

  if (format == long_format || print_scontext)
```

```
  {
    bool have_scontext = false;
    bool have_acl = false;
    int attr_len = getfilecon_cache (absolute_name, f, do_deref);
    err = (attr_len < 0);

    if (err == 0)
      {
        if (is_smack_enabled ())
          have_scontext = ! STREQ ("_", f->scontext);
        else
          have_scontext = ! STREQ ("unlabeled", f->scontext);
      }
    else
      {
        f->scontext = UNKNOWN_SECURITY_CONTEXT;

        /* When requesting security context information, don't make
           ls fail just because the file (even a command line argument)
           isn't on the right type of file system.  I.e., a getfilecon
           failure isn't in the same class as a stat failure.  */
        if (is_ENOTSUP (errno) || errno == ENODATA)
          err = 0;
      }

    if (err == 0 && format == long_format)
      {
        int n = file_has_acl_cache (absolute_name, f);
        err = (n < 0);
        have_acl = (0 < n);
      }

    f->acl_type = (!have_scontext && !have_acl
                   ? ACL_T_NONE
                   : (have_scontext && !have_acl
                      ? ACL_T_LSM_CONTEXT_ONLY
                      : ACL_T_YES));
    any_has_acl |= f->acl_type != ACL_T_NONE;

    if (err)
      error (0, errno, "%s", quotearg_colon (absolute_name));
  }

if (S_ISLNK (f->stat.st_mode)
    && (format == long_format || check_symlink_color))
  {
    struct stat linkstats;

    get_link_name (absolute_name, f, command_line_arg);
    char *linkname = make_link_name (absolute_name, f->linkname);

    /* Avoid following symbolic links when possible, ie, when
       they won't be traced and when no indicator is needed.  */
    if (linkname
        && (file_type <= indicator_style || check_symlink_color)
        && stat (linkname, &linkstats) == 0)
      {
        f->linkok = true;

        /* Symbolic links to directories that are mentioned on the
           command line are automatically traced if not being
           listed as files.  */
        if (!command_line_arg || format == long_format
            || !S_ISDIR (linkstats.st_mode))
          {
            /* Get the linked-to file's mode for the filetype indicator
               in long listings.  */
            f->linkmode = linkstats.st_mode;
          }
      }
    free (linkname);
  }

if (S_ISLNK (f->stat.st_mode))
  f->filetype = symbolic_link;
else if (S_ISDIR (f->stat.st_mode))
  {
    if (command_line_arg && !immediate_dirs)
      f->filetype = arg_directory;
    else
      f->filetype = directory;
  }
else
  f->filetype = normal;

blocks = ST_NBLOCKS (f->stat);
if (format == long_format || print_block_size)
  {
    char buf[LONGEST_HUMAN_READABLE + 1];
    int len = mbswidth (human_readable (blocks, buf, human_output_opts,
                                        ST_NBLOCKSIZE, output_block_size),
                        0);
    if (block_size_width < len)
      block_size_width = len;
  }

if (format == long_format)
  {
```

```
          if (print_owner)
            {
              int len = format_user_width (f->stat.st_uid);
              if (owner_width < len)
                owner_width = len;
            }

          if (print_group)
            {
              int len = format_group_width (f->stat.st_gid);
              if (group_width < len)
                group_width = len;
            }

          if (print_author)
            {
              int len = format_user_width (f->stat.st_author);
              if (author_width < len)
                author_width = len;
            }
        }

      if (print_scontext)
        {
          int len = strlen (f->scontext);
          if (scontext_width < len)
            scontext_width = len;
        }

      if (format == long_format)
        {
          char b[INT_BUFSIZE_BOUND (uintmax_t)];
          int b_len = strlen (umaxtostr (f->stat.st_nlink, b));
          if (nlink_width < b_len)
            nlink_width = b_len;

          if (S_ISCHR (f->stat.st_mode) || S_ISBLK (f->stat.st_mode))
            {
              char buf[INT_BUFSIZE_BOUND (uintmax_t)];
              int len = strlen (umaxtostr (major (f->stat.st_rdev), buf));
              if (major_device_number_width < len)
                major_device_number_width = len;
              len = strlen (umaxtostr (minor (f->stat.st_rdev), buf));
              if (minor_device_number_width < len)
                minor_device_number_width = len;
              len = major_device_number_width + 2 + minor_device_number_width;
              if (file_size_width < len)
                file_size_width = len;
            }
          else
            {
              char buf[LONGEST_HUMAN_READABLE + 1];
              uintmax_t size = unsigned_file_size (f->stat.st_size);
              int len = mbswidth (human_readable (size, buf,
                                                  file_human_output_opts,
                                                  1, file_output_block_size),
                                  0);
              if (file_size_width < len)
                file_size_width = len;
            }
        }
    }

  if (print_inode)
    {
      char buf[INT_BUFSIZE_BOUND (uintmax_t)];
      int len = strlen (umaxtostr (f->stat.st_ino, buf));
      if (inode_number_width < len)
        inode_number_width = len;
    }

  f->name = xstrdup (name);
  cwd_n_used++;

  return blocks;
}

/* Return true if F refers to a directory.  */
static bool
is_directory (const struct fileinfo *f)
{
  return f->filetype == directory || f->filetype == arg_directory;
}

/* Put the name of the file that FILENAME is a symbolic link to
   into the LINKNAME field of 'f'.  COMMAND_LINE_ARG indicates whether
   FILENAME is a command-line argument.  */

static void
get_link_name (char const *filename, struct fileinfo *f, bool command_line_arg)
{
  f->linkname = areadlink_with_size (filename, f->stat.st_size);
  if (f->linkname == NULL)
    file_failure (command_line_arg, _("cannot read symbolic link %s"),
                  filename);
}

/* If LINKNAME is a relative name and NAME contains one or more
```

```
   leading directories, return LINKNAME with those directories
   prepended; otherwise, return a copy of LINKNAME.
   If LINKNAME is NULL, return NULL.  */

static char *
make_link_name (char const *name, char const *linkname)
{
  if (!linkname)
    return NULL;

  if (IS_ABSOLUTE_FILE_NAME (linkname))
    return xstrdup (linkname);

  /* The link is to a relative name.  Prepend any leading directory
     in 'name' to the link name.  */
  size_t prefix_len = dir_len (name);
  if (prefix_len == 0)
    return xstrdup (linkname);

  char *p = xmalloc (prefix_len + 1 + strlen (linkname) + 1);

  /* PREFIX_LEN usually specifies a string not ending in slash.
     In that case, extend it by one, since the next byte *is* a slash.
     Otherwise, the prefix is "/", so leave the length unchanged.  */
  if ( ! ISSLASH (name[prefix_len - 1]))
    ++prefix_len;

  stpcpy (stpncpy (p, name, prefix_len), linkname);
  return p;
}

/* Return true if the last component of NAME is '.' or '..'
   This is so we don't try to recurse on './././. ...' */

static bool
basename_is_dot_or_dotdot (const char *name)
{
  char const *base = last_component (name);
  return dot_or_dotdot (base);
}

/* Remove any entries from CWD_FILE that are for directories,
   and queue them to be listed as directories instead.
   DIRNAME is the prefix to prepend to each dirname
   to make it correct relative to ls's working dir;
   if it is null, no prefix is needed and "." and ".." should not be ignored.
   If COMMAND_LINE_ARG is true, this directory was mentioned at the top level,
   This is desirable when processing directories recursively.  */

static void
extract_dirs_from_files (char const *dirname, bool command_line_arg)
{
  size_t i;
  size_t j;
  bool ignore_dot_and_dot_dot = (dirname != NULL);

  if (dirname && LOOP_DETECT)
    {
      /* Insert a marker entry first.  When we dequeue this marker entry,
         we'll know that DIRNAME has been processed and may be removed
         from the set of active directories.  */
      queue_directory (NULL, dirname, false);
    }

  /* Queue the directories last one first, because queueing reverses the
     order.  */
  for (i = cwd_n_used; i-- != 0; )
    {
      struct fileinfo *f = sorted_file[i];

      if (is_directory (f)
          && (! ignore_dot_and_dot_dot
              || ! basename_is_dot_or_dotdot (f->name)))
        {
          if (!dirname || f->name[0] == '/')
            queue_directory (f->name, f->linkname, command_line_arg);
          else
            {
              char *name = file_name_concat (dirname, f->name, NULL);
              queue_directory (name, f->linkname, command_line_arg);
              free (name);
            }
          if (f->filetype == arg_directory)
            free_ent (f);
        }
    }

  /* Now delete the directories from the table, compacting all the remaining
     entries.  */

  for (i = 0, j = 0; i < cwd_n_used; i++)
    {
      struct fileinfo *f = sorted_file[i];
      sorted_file[j] = f;
      j += (f->filetype != arg_directory);
    }
  cwd_n_used = j;
}
```

```
/* Use strcoll to compare strings in this locale.  If an error occurs,
   report an error and longjmp to failed_strcoll.  */

static jmp_buf failed_strcoll;

static int
xstrcoll (char const *a, char const *b)
{
  int diff;
  errno = 0;
  diff = strcoll (a, b);
  if (errno)
    {
      error (0, errno, _("cannot compare file names %s and %s"),
             quote_n (0, a), quote_n (1, b));
      set_exit_status (false);
      longjmp (failed_strcoll, 1);
    }
  return diff;
}

/* Comparison routines for sorting the files.  */

typedef void const *V;
typedef int (*qsortFunc)(V a, V b);

/* Used below in DEFINE_SORT_FUNCTIONS for _df_ sort function variants.
   The do { ... } while(0) makes it possible to use the macro more like
   a statement, without violating C89 rules: */
#define DIRFIRST_CHECK(a, b)                                            \
  do                                                                    \
    {                                                                   \
      bool a_is_dir = is_directory ((struct fileinfo const *) a);       \
      bool b_is_dir = is_directory ((struct fileinfo const *) b);       \
      if (a_is_dir && !b_is_dir)                                        \
        return -1;          /* a goes before b */                       \
      if (!a_is_dir && b_is_dir)                                        \
        return 1;           /* b goes before a */                       \
    }                                                                   \
  while (0)

/* Define the 8 different sort function variants required for each sortkey.
   KEY_NAME is a token describing the sort key, e.g., ctime, atime, size.
   KEY_CMP_FUNC is a function to compare records based on that key, e.g.,
   ctime_cmp, atime_cmp, size_cmp.  Append KEY_NAME to the string,
   '[rev_][x]str{cmp|coll}[_df]_', to create each function name.  */
#define DEFINE_SORT_FUNCTIONS(key_name, key_cmp_func)                   \
  /* direct, non-dirfirst versions */                                   \
  static int xstrcoll_##key_name (V a, V b)                             \
  { return key_cmp_func (a, b, xstrcoll); }                             \
  static int strcmp_##key_name (V a, V b)                               \
  { return key_cmp_func (a, b, strcmp); }                               \
                                                                        \
  /* reverse, non-dirfirst versions */                                  \
  static int rev_xstrcoll_##key_name (V a, V b)                         \
  { return key_cmp_func (b, a, xstrcoll); }                             \
  static int rev_strcmp_##key_name (V a, V b)                           \
  { return key_cmp_func (b, a, strcmp); }                               \
                                                                        \
  /* direct, dirfirst versions */                                       \
  static int xstrcoll_df_##key_name (V a, V b)                          \
  { DIRFIRST_CHECK (a, b); return key_cmp_func (a, b, xstrcoll); }      \
  static int strcmp_df_##key_name (V a, V b)                            \
  { DIRFIRST_CHECK (a, b); return key_cmp_func (a, b, strcmp); }        \
                                                                        \
  /* reverse, dirfirst versions */                                      \
  static int rev_xstrcoll_df_##key_name (V a, V b)                      \
  { DIRFIRST_CHECK (a, b); return key_cmp_func (b, a, xstrcoll); }      \
  static int rev_strcmp_df_##key_name (V a, V b)                        \
  { DIRFIRST_CHECK (a, b); return key_cmp_func (b, a, strcmp); }

static inline int
cmp_ctime (struct fileinfo const *a, struct fileinfo const *b,
           int (*cmp) (char const *, char const *))
{
  int diff = timespec_cmp (get_stat_ctime (&b->stat),
                           get_stat_ctime (&a->stat));
  return diff ? diff : cmp (a->name, b->name);
}

static inline int
cmp_mtime (struct fileinfo const *a, struct fileinfo const *b,
           int (*cmp) (char const *, char const *))
{
  int diff = timespec_cmp (get_stat_mtime (&b->stat),
                           get_stat_mtime (&a->stat));
  return diff ? diff : cmp (a->name, b->name);
}

static inline int
cmp_atime (struct fileinfo const *a, struct fileinfo const *b,
           int (*cmp) (char const *, char const *))
{
  int diff = timespec_cmp (get_stat_atime (&b->stat),
                           get_stat_atime (&a->stat));
  return diff ? diff : cmp (a->name, b->name);
}
```

```
static inline int
cmp_size (struct fileinfo const *a, struct fileinfo const *b,
          int (*cmp) (char const *, char const *))
{
  int diff = longdiff (b->stat.st_size, a->stat.st_size);
  return diff ? diff : cmp (a->name, b->name);
}

static inline int
cmp_name (struct fileinfo const *a, struct fileinfo const *b,
          int (*cmp) (char const *, char const *))
{
  return cmp (a->name, b->name);
}

/* Compare file extensions.  Files with no extension are 'smallest'.
   If extensions are the same, compare by filenames instead.  */

static inline int
cmp_extension (struct fileinfo const *a, struct fileinfo const *b,
               int (*cmp) (char const *, char const *))
{
  char const *base1 = strrchr (a->name, '.');
  char const *base2 = strrchr (b->name, '.');
  int diff = cmp (base1 ? base1 : "", base2 ? base2 : "");
  return diff ? diff : cmp (a->name, b->name);
}

DEFINE_SORT_FUNCTIONS (ctime, cmp_ctime)
DEFINE_SORT_FUNCTIONS (mtime, cmp_mtime)
DEFINE_SORT_FUNCTIONS (atime, cmp_atime)
DEFINE_SORT_FUNCTIONS (size, cmp_size)
DEFINE_SORT_FUNCTIONS (name, cmp_name)
DEFINE_SORT_FUNCTIONS (extension, cmp_extension)

/* Compare file versions.
   Unlike all other compare functions above, cmp_version depends only
   on filevercmp, which does not fail (even for locale reasons), and does not
   need a secondary sort key. See lib/filevercmp.h for function description.

   All the other sort options, in fact, need xstrcoll and strcmp variants,
   because they all use a string comparison (either as the primary or secondary
   sort key), and xstrcoll has the ability to do a longjmp if strcoll fails for
   locale reasons.  Lastly, filevercmp is ALWAYS available with gnulib.  */
static inline int
cmp_version (struct fileinfo const *a, struct fileinfo const *b)
{
  return filevercmp (a->name, b->name);
}

static int xstrcoll_version (V a, V b)
{ return cmp_version (a, b); }
static int rev_xstrcoll_version (V a, V b)
{ return cmp_version (b, a); }
static int xstrcoll_df_version (V a, V b)
{ DIRFIRST_CHECK (a, b); return cmp_version (a, b); }
static int rev_xstrcoll_df_version (V a, V b)
{ DIRFIRST_CHECK (a, b); return cmp_version (b, a); }


/* We have 2^3 different variants for each sort-key function
   (for 3 independent sort modes).
   The function pointers stored in this array must be dereferenced as:

    sort_variants[sort_key][use_strcmp][reverse][dirs_first]

   Note that the order in which sort keys are listed in the function pointer
   array below is defined by the order of the elements in the time_type and
   sort_type enums!  */

#define LIST_SORTFUNCTION_VARIANTS(key_name)                        \
  {                                                                 \
    {                                                               \
      { xstrcoll_##key_name, xstrcoll_df_##key_name },              \
      { rev_xstrcoll_##key_name, rev_xstrcoll_df_##key_name },      \
    },                                                              \
    {                                                               \
      { strcmp_##key_name, strcmp_df_##key_name },                  \
      { rev_strcmp_##key_name, rev_strcmp_df_##key_name },          \
    }                                                               \
  }

static qsortFunc const sort_functions[][2][2][2] =
  {
    LIST_SORTFUNCTION_VARIANTS (name),
    LIST_SORTFUNCTION_VARIANTS (extension),
    LIST_SORTFUNCTION_VARIANTS (size),

    {
      {
        { xstrcoll_version, xstrcoll_df_version },
        { rev_xstrcoll_version, rev_xstrcoll_df_version },
      },

      /* We use NULL for the strcmp variants of version comparison
         since as explained in cmp_version definition, version comparison
         does not rely on xstrcoll, so it will never longjmp, and never
```

```
         need to try the strcmp fallback. */
      {
        { NULL, NULL },
        { NULL, NULL },
      }
    },

    /* last are time sort functions */
    LIST_SORTFUNCTION_VARIANTS (mtime),
    LIST_SORTFUNCTION_VARIANTS (ctime),
    LIST_SORTFUNCTION_VARIANTS (atime)
  };

/* The number of sort keys is calculated as the sum of
     the number of elements in the sort_type enum (i.e., sort_numtypes)
     the number of elements in the time_type enum (i.e., time_numtypes) - 1
   This is because when sort_type==sort_time, we have up to
   time_numtypes possible sort keys.

   This line verifies at compile-time that the array of sort functions has been
   initialized for all possible sort keys. */
verify (ARRAY_CARDINALITY (sort_functions)
        == sort_numtypes + time_numtypes - 1 );

/* Set up SORTED_FILE to point to the in-use entries in CWD_FILE, in order.  */

static void
initialize_ordering_vector (void)
{
  size_t i;
  for (i = 0; i < cwd_n_used; i++)
    sorted_file[i] = &cwd_file[i];
}

/* Sort the files now in the table.  */

static void
sort_files (void)
{
  bool use_strcmp;

  if (sorted_file_alloc < cwd_n_used + cwd_n_used / 2)
    {
      free (sorted_file);
      sorted_file = xnmalloc (cwd_n_used, 3 * sizeof *sorted_file);
      sorted_file_alloc = 3 * cwd_n_used;
    }

  initialize_ordering_vector ();

  if (sort_type == sort_none)
    return;

  /* Try strcoll.  If it fails, fall back on strcmp.  We can't safely
     ignore strcoll failures, as a failing strcoll might be a
     comparison function that is not a total order, and if we ignored
     the failure this might cause qsort to dump core.  */

  if (! setjmp (failed_strcoll))
    use_strcmp = false;      /* strcoll() succeeded */
  else
    {
      use_strcmp = true;
      assert (sort_type != sort_version);
      initialize_ordering_vector ();
    }

  /* When sort_type == sort_time, use time_type as subindex.  */
  mpsort ((void const **) sorted_file, cwd_n_used,
          sort_functions[sort_type + (sort_type == sort_time ? time_type : 0)]
                        [use_strcmp][sort_reverse]
                        [directories_first]);
}

/* List all the files now in the table.  */

static void
print_current_files (void)
{
  size_t i;

  switch (format)
    {
    case one_per_line:
      for (i = 0; i < cwd_n_used; i++)
        {
          print_file_name_and_frills (sorted_file[i], 0);
          putchar ('\n');
        }
      break;

    case many_per_line:
      print_many_per_line ();
      break;

    case horizontal:
      print_horizontal ();
      break;
```

```
    case with_commas:
      print_with_commas ();
      break;

    case long_format:
      for (i = 0; i < cwd_n_used; i++)
        {
          set_normal_color ();
          print_long_format (sorted_file[i]);
          DIRED_PUTCHAR ('\n');
        }
      break;
    }
}

/* Replace the first %b with precomputed aligned month names.
   Note on glibc-2.7 at least, this speeds up the whole 'ls -lU'
   process by around 17%, compared to letting strftime() handle the %b.  */

static size_t
align_nstrftime (char *buf, size_t size, char const *fmt, struct tm const *tm,
                 int __utc, int __ns)
{
  const char *nfmt = fmt;
  /* In the unlikely event that rpl_fmt below is not large enough,
     the replacement is not done.  A malloc here slows ls down by 2%  */
  char rpl_fmt[sizeof (abmon[0]) + 100];
  const char *pb;
  if (required_mon_width && (pb = strstr (fmt, "%b"))
      && 0 <= tm->tm_mon && tm->tm_mon <= 11)
    {
      if (strlen (fmt) < (sizeof (rpl_fmt) - sizeof (abmon[0]) + 2))
        {
          char *pfmt = rpl_fmt;
          nfmt = rpl_fmt;

          pfmt = mempcpy (pfmt, fmt, pb - fmt);
          pfmt = stpcpy (pfmt, abmon[tm->tm_mon]);
          strcpy (pfmt, pb + 2);
        }
    }
  size_t ret = nstrftime (buf, size, nfmt, tm, __utc, __ns);
  return ret;
}

/* Return the expected number of columns in a long-format time stamp,
   or zero if it cannot be calculated.  */

static int
long_time_expected_width (void)
{
  static int width = -1;

  if (width < 0)
    {
      time_t epoch = 0;
      struct tm const *tm = localtime (&epoch);
      char buf[TIME_STAMP_LEN_MAXIMUM + 1];

      /* In case you're wondering if localtime can fail with an input time_t
         value of 0, let's just say it's very unlikely, but not inconceivable.
         The TZ environment variable would have to specify a time zone that
         is 2**31-1900 years or more ahead of UTC.  This could happen only on
         a 64-bit system that blindly accepts e.g., TZ=UTC+20000000000000.
         However, this is not possible with Solaris 10 or glibc-2.3.5, since
         their implementations limit the offset to 167:59 and 24:00, resp.  */
      if (tm)
        {
          size_t len =
            align_nstrftime (buf, sizeof buf, long_time_format[0], tm, 0, 0);
          if (len != 0)
            width = mbsnwidth (buf, len, 0);
        }

      if (width < 0)
        width = 0;
    }

  return width;
}

/* Print the user or group name NAME, with numeric id ID, using a
   print width of WIDTH columns.  */

static void
format_user_or_group (char const *name, unsigned long int id, int width)
{
  size_t len;

  if (name)
    {
      int width_gap = width - mbswidth (name, 0);
      int pad = MAX (0, width_gap);
      fputs (name, stdout);
      len = strlen (name) + pad;

      do
```

```
        putchar (' ');
      while (pad--);
    }
  else
    {
      printf ("%*lu ", width, id);
      len = width;
    }

  dired_pos += len + 1;
}

/* Print the name or id of the user with id U, using a print width of
   WIDTH.  */

static void
format_user (uid_t u, int width, bool stat_ok)
{
  format_user_or_group (! stat_ok ? "?" :
                        (numeric_ids ? NULL : getuser (u)), u, width);
}

/* Likewise, for groups.  */

static void
format_group (gid_t g, int width, bool stat_ok)
{
  format_user_or_group (! stat_ok ? "?" :
                        (numeric_ids ? NULL : getgroup (g)), g, width);
}

/* Return the number of columns that format_user_or_group will print.  */

static int
format_user_or_group_width (char const *name, unsigned long int id)
{
  if (name)
    {
      int len = mbswidth (name, 0);
      return MAX (0, len);
    }
  else
    {
      char buf[INT_BUFSIZE_BOUND (id)];
      sprintf (buf, "%lu", id);
      return strlen (buf);
    }
}

/* Return the number of columns that format_user will print.  */

static int
format_user_width (uid_t u)
{
  return format_user_or_group_width (numeric_ids ? NULL : getuser (u), u);
}

/* Likewise, for groups.  */

static int
format_group_width (gid_t g)
{
  return format_user_or_group_width (numeric_ids ? NULL : getgroup (g), g);
}

/* Return a pointer to a formatted version of F->stat.st_ino,
   possibly using buffer, BUF, of length BUFLEN, which must be at least
   INT_BUFSIZE_BOUND (uintmax_t) bytes.  */
static char *
format_inode (char *buf, size_t buflen, const struct fileinfo *f)
{
  assert (INT_BUFSIZE_BOUND (uintmax_t) <= buflen);
  return (f->stat_ok && f->stat.st_ino != NOT_AN_INODE_NUMBER
          ? umaxtostr (f->stat.st_ino, buf)
          : (char *) "?");
}

/* Print information about F in long format.  */
static void
print_long_format (const struct fileinfo *f)
{
  char modebuf[12];
  char buf
    [LONGEST_HUMAN_READABLE + 1         /* inode */
     + LONGEST_HUMAN_READABLE + 1       /* size in blocks */
     + sizeof (modebuf) - 1 + 1         /* mode string */
     + INT_BUFSIZE_BOUND (uintmax_t)    /* st_nlink */
     + LONGEST_HUMAN_READABLE + 2       /* major device number */
     + LONGEST_HUMAN_READABLE + 1       /* minor device number */
     + TIME_STAMP_LEN_MAXIMUM + 1       /* max length of time/date */
     ];
  size_t s;
  char *p;
  struct timespec when_timespec;
  struct tm *when_local;

  /* Compute the mode string, except remove the trailing space if no
     file in this directory has an ACL or security context.  */
```

```
if (f->stat_ok)
  filemodestring (&f->stat, modebuf);
else
  {
    modebuf[0] = filetype_letter[f->filetype];
    memset (modebuf + 1, '?', 10);
    modebuf[11] = '\0';
  }
if (! any_has_acl)
  modebuf[10] = '\0';
else if (f->acl_type == ACL_T_LSM_CONTEXT_ONLY)
  modebuf[10] = '.';
else if (f->acl_type == ACL_T_YES)
  modebuf[10] = '+';

switch (time_type)
  {
  case time_ctime:
    when_timespec = get_stat_ctime (&f->stat);
    break;
  case time_mtime:
    when_timespec = get_stat_mtime (&f->stat);
    break;
  case time_atime:
    when_timespec = get_stat_atime (&f->stat);
    break;
  default:
    abort ();
  }

p = buf;

if (print_inode)
  {
    char hbuf[INT_BUFSIZE_BOUND (uintmax_t)];
    sprintf (p, "%*s ", inode_number_width,
             format_inode (hbuf, sizeof hbuf, f));
    /* Increment by strlen (p) here, rather than by inode_number_width + 1.
       The latter is wrong when inode_number_width is zero.  */
    p += strlen (p);
  }

if (print_block_size)
  {
    char hbuf[LONGEST_HUMAN_READABLE + 1];
    char const *blocks =
      (! f->stat_ok
       ? "?"
       : human_readable (ST_NBLOCKS (f->stat), hbuf, human_output_opts,
                         ST_NBLOCKSIZE, output_block_size));
    int pad;
    for (pad = block_size_width - mbswidth (blocks, 0); 0 < pad; pad--)
      *p++ = ' ';
    while ((*p++ = *blocks++))
      continue;
    p[-1] = ' ';
  }

/* The last byte of the mode string is the POSIX
   "optional alternate access method flag".  */
{
  char hbuf[INT_BUFSIZE_BOUND (uintmax_t)];
  sprintf (p, "%s %*s ", modebuf, nlink_width,
           ! f->stat_ok ? "?" : umaxtostr (f->stat.st_nlink, hbuf));
}
/* Increment by strlen (p) here, rather than by, e.g.,
   sizeof modebuf - 2 + any_has_acl + 1 + nlink_width + 1.
   The latter is wrong when nlink_width is zero.  */
p += strlen (p);

DIRED_INDENT ();

if (print_owner || print_group || print_author || print_scontext)
  {
    DIRED_FPUTS (buf, stdout, p - buf);

    if (print_owner)
      format_user (f->stat.st_uid, owner_width, f->stat_ok);

    if (print_group)
      format_group (f->stat.st_gid, group_width, f->stat_ok);

    if (print_author)
      format_user (f->stat.st_author, author_width, f->stat_ok);

    if (print_scontext)
      format_user_or_group (f->scontext, 0, scontext_width);

    p = buf;
  }

if (f->stat_ok
    && (S_ISCHR (f->stat.st_mode) || S_ISBLK (f->stat.st_mode)))
  {
    char majorbuf[INT_BUFSIZE_BOUND (uintmax_t)];
    char minorbuf[INT_BUFSIZE_BOUND (uintmax_t)];
    int blanks_width = (file_size_width
                        - (major_device_number_width + 2
```

```
                          + minor_device_number_width));
    sprintf (p, "%*s, %*s ",
             major_device_number_width + MAX (0, blanks_width),
             umaxtostr (major (f->stat.st_rdev), majorbuf),
             minor_device_number_width,
             umaxtostr (minor (f->stat.st_rdev), minorbuf));
    p += file_size_width + 1;
  }
else
  {
    char hbuf[LONGEST_HUMAN_READABLE + 1];
    char const *size =
      (! f->stat_ok
       ? "?"
       : human_readable (unsigned_file_size (f->stat.st_size),
                         hbuf, file_human_output_opts, 1,
                         file_output_block_size));
    int pad;
    for (pad = file_size_width - mbswidth (size, 0); 0 < pad; pad--)
      *p++ = ' ';
    while ((*p++ = *size++))
      continue;
    p[-1] = ' ';
  }

when_local = localtime (&when_timespec.tv_sec);
s = 0;
*p = '\1';

if (f->stat_ok && when_local)
  {
    struct timespec six_months_ago;
    bool recent;
    char const *fmt;

    /* If the file appears to be in the future, update the current
       time, in case the file happens to have been modified since
       the last time we checked the clock.  */
    if (timespec_cmp (current_time, when_timespec) < 0)
      {
        /* Note that gettime may call gettimeofday which, on some non-
           compliant systems, clobbers the buffer used for localtime's result.
           But it's ok here, because we use a gettimeofday wrapper that
           saves and restores the buffer around the gettimeofday call.  */
        gettime (¤t_time);
      }

    /* Consider a time to be recent if it is within the past six months.
       A Gregorian year has 365.2425 * 24 * 60 * 60 == 31556952 seconds
       on the average.  Write this value as an integer constant to
       avoid floating point hassles.  */
    six_months_ago.tv_sec = current_time.tv_sec - 31556952 / 2;
    six_months_ago.tv_nsec = current_time.tv_nsec;

    recent = (timespec_cmp (six_months_ago, when_timespec) < 0
              && (timespec_cmp (when_timespec, current_time) < 0));
    fmt = long_time_format[recent];

    /* We assume here that all time zones are offset from UTC by a
       whole number of seconds.  */
    s = align_nstrftime (p, TIME_STAMP_LEN_MAXIMUM + 1, fmt,
                         when_local, 0, when_timespec.tv_nsec);
  }

if (s || !*p)
  {
    p += s;
    *p++ = ' ';

    /* NUL-terminate the string -- fputs (via DIRED_FPUTS) requires it.  */
    *p = '\0';
  }
else
  {
    /* The time cannot be converted using the desired format, so
       print it as a huge integer number of seconds.  */
    char hbuf[INT_BUFSIZE_BOUND (intmax_t)];
    sprintf (p, "%*s ", long_time_expected_width (),
             (! f->stat_ok
              ? "?"
              : timetostr (when_timespec.tv_sec, hbuf)));
    /* FIXME: (maybe) We discarded when_timespec.tv_nsec. */
    p += strlen (p);
  }

DIRED_FPUTS (buf, stdout, p - buf);
size_t w = print_name_with_quoting (f, false, &dired_obstack, p - buf);

if (f->filetype == symbolic_link)
  {
    if (f->linkname)
      {
        DIRED_FPUTS_LITERAL (" -> ", stdout);
        print_name_with_quoting (f, true, NULL, (p - buf) + w + 4);
        if (indicator_style != none)
          print_type_indicator (true, f->linkmode, unknown);
      }
  }
```

```
  else if (indicator_style != none)
    print_type_indicator (f->stat_ok, f->stat.st_mode, f->filetype);
}

/* Output to OUT a quoted representation of the file name NAME,
   using OPTIONS to control quoting.  Produce no output if OUT is NULL.
   Store the number of screen columns occupied by NAME's quoted
   representation into WIDTH, if non-NULL.  Return the number of bytes
   produced.  */

static size_t
quote_name (FILE *out, const char *name, struct quoting_options const *options,
            size_t *width)
{
  char smallbuf[BUFSIZ];
  size_t len = quotearg_buffer (smallbuf, sizeof smallbuf, name, -1, options);
  char *buf;
  size_t displayed_width IF_LINT ( = 0);

  if (len < sizeof smallbuf)
    buf = smallbuf;
  else
    {
      buf = alloca (len + 1);
      quotearg_buffer (buf, len + 1, name, -1, options);
    }

  if (qmark_funny_chars)
    {
      if (MB_CUR_MAX > 1)
        {
          char const *p = buf;
          char const *plimit = buf + len;
          char *q = buf;
          displayed_width = 0;

          while (p < plimit)
            switch (*p)
              {
                case ' ': case '!': case '"': case '#': case '%':
                case '&': case '\'': case '(': case ')': case '*':
                case '+': case ',': case '-': case '.': case '/':
                case '0': case '1': case '2': case '3': case '4':
                case '5': case '6': case '7': case '8': case '9':
                case ':': case ';': case '<': case '=': case '>':
                case '?':
                case 'A': case 'B': case 'C': case 'D': case 'E':
                case 'F': case 'G': case 'H': case 'I': case 'J':
                case 'K': case 'L': case 'M': case 'N': case 'O':
                case 'P': case 'Q': case 'R': case 'S': case 'T':
                case 'U': case 'V': case 'W': case 'X': case 'Y':
                case 'Z':
                case '[': case '\\': case ']': case '^': case '_':
                case 'a': case 'b': case 'c': case 'd': case 'e':
                case 'f': case 'g': case 'h': case 'i': case 'j':
                case 'k': case 'l': case 'm': case 'n': case 'o':
                case 'p': case 'q': case 'r': case 's': case 't':
                case 'u': case 'v': case 'w': case 'x': case 'y':
                case 'z': case '{': case '|': case '}': case '~':
                  /* These characters are printable ASCII characters.  */
                  *q++ = *p++;
                  displayed_width += 1;
                  break;
                default:
                  /* If we have a multibyte sequence, copy it until we
                     reach its end, replacing each non-printable multibyte
                     character with a single question mark.  */
                  {
                    mbstate_t mbstate = { 0, };
                    do
                      {
                        wchar_t wc;
                        size_t bytes;
                        int w;

                        bytes = mbrtowc (&wc, p, plimit - p, &mbstate);

                        if (bytes == (size_t) -1)
                          {
                            /* An invalid multibyte sequence was
                               encountered.  Skip one input byte, and
                               put a question mark.  */
                            p++;
                            *q++ = '?';
                            displayed_width += 1;
                            break;
                          }

                        if (bytes == (size_t) -2)
                          {
                            /* An incomplete multibyte character
                               at the end.  Replace it entirely with
                               a question mark.  */
                            p = plimit;
                            *q++ = '?';
                            displayed_width += 1;
                            break;
                          }
```

```
                        if (bytes == 0)
                          /* A null wide character was encountered.  */
                          bytes = 1;

                        w = wcwidth (wc);
                        if (w >= 0)
                          {
                            /* A printable multibyte character.
                               Keep it.  */
                            for (; bytes > 0; --bytes)
                              *q++ = *p++;
                            displayed_width += w;
                          }
                        else
                          {
                            /* An unprintable multibyte character.
                               Replace it entirely with a question
                               mark.  */
                            p += bytes;
                            *q++ = '?';
                            displayed_width += 1;
                          }
                      }
                    while (! mbsinit (&mbstate));
                  }
                  break;
              }

          /* The buffer may have shrunk.  */
          len = q - buf;
        }
      else
        {
          char *p = buf;
          char const *plimit = buf + len;

          while (p < plimit)
            {
              if (! isprint (to_uchar (*p)))
                *p = '?';
              p++;
            }
          displayed_width = len;
        }
    }
  else if (width != NULL)
    {
      if (MB_CUR_MAX > 1)
        displayed_width = mbsnwidth (buf, len, 0);
      else
        {
          char const *p = buf;
          char const *plimit = buf + len;

          displayed_width = 0;
          while (p < plimit)
            {
              if (isprint (to_uchar (*p)))
                displayed_width++;
              p++;
            }
        }
    }

  if (out != NULL)
    fwrite (buf, 1, len, out);
  if (width != NULL)
    *width = displayed_width;
  return len;
}

static size_t
print_name_with_quoting (const struct fileinfo *f,
                         bool symlink_target,
                         struct obstack *stack,
                         size_t start_col)
{
  const char* name = symlink_target ? f->linkname : f->name;

  bool used_color_this_time
    = (print_with_color
        && (print_color_indicator (f, symlink_target)
            || is_colored (C_NORM)));

  if (stack)
    PUSH_CURRENT_DIRED_POS (stack);

  size_t width = quote_name (stdout, name, filename_quoting_options, NULL);
  dired_pos += width;

  if (stack)
    PUSH_CURRENT_DIRED_POS (stack);

  process_signals ();
  if (used_color_this_time)
    {
      prep_non_filename_text ();
```

```
      if (start_col / line_length != (start_col + width - 1) / line_length)
        put_indicator (&color_indicator[C_CLR_TO_EOL]);
    }

  return width;
}

static void
prep_non_filename_text (void)
{
  if (color_indicator[C_END].string != NULL)
    put_indicator (&color_indicator[C_END]);
  else
    {
      put_indicator (&color_indicator[C_LEFT]);
      put_indicator (&color_indicator[C_RESET]);
      put_indicator (&color_indicator[C_RIGHT]);
    }
}

/* Print the file name of 'f' with appropriate quoting.
   Also print file size, inode number, and filetype indicator character,
   as requested by switches.  */

static size_t
print_file_name_and_frills (const struct fileinfo *f, size_t start_col)
{
  char buf[MAX (LONGEST_HUMAN_READABLE + 1, INT_BUFSIZE_BOUND (uintmax_t))];

  set_normal_color ();

  if (print_inode)
    printf ("%*s ", format == with_commas ? 0 : inode_number_width,
            format_inode (buf, sizeof buf, f));

  if (print_block_size)
    printf ("%*s ", format == with_commas ? 0 : block_size_width,
            ! f->stat_ok ? "?"
            : human_readable (ST_NBLOCKS (f->stat), buf, human_output_opts,
                              ST_NBLOCKSIZE, output_block_size));

  if (print_scontext)
    printf ("%*s ", format == with_commas ? 0 : scontext_width, f->scontext);

  size_t width = print_name_with_quoting (f, false, NULL, start_col);

  if (indicator_style != none)
    width += print_type_indicator (f->stat_ok, f->stat.st_mode, f->filetype);

  return width;
}

/* Given these arguments describing a file, return the single-byte
   type indicator, or 0.  */
static char
get_type_indicator (bool stat_ok, mode_t mode, enum filetype type)
{
  char c;

  if (stat_ok ? S_ISREG (mode) : type == normal)
    {
      if (stat_ok && indicator_style == classify && (mode & S_IXUGO))
        c = '*';
      else
        c = 0;
    }
  else
    {
      if (stat_ok ? S_ISDIR (mode) : type == directory || type == arg_directory)
        c = '/';
      else if (indicator_style == slash)
        c = 0;
      else if (stat_ok ? S_ISLNK (mode) : type == symbolic_link)
        c = '@';
      else if (stat_ok ? S_ISFIFO (mode) : type == fifo)
        c = '|';
      else if (stat_ok ? S_ISSOCK (mode) : type == sock)
        c = '=';
      else if (stat_ok && S_ISDOOR (mode))
        c = '>';
      else
        c = 0;
    }
  return c;
}

static bool
print_type_indicator (bool stat_ok, mode_t mode, enum filetype type)
{
  char c = get_type_indicator (stat_ok, mode, type);
  if (c)
    DIRED_PUTCHAR (c);
  return !!c;
}

/* Returns whether any color sequence was printed. */
static bool
print_color_indicator (const struct fileinfo *f, bool symlink_target)
```

```
{
  enum indicator_no type;
  struct color_ext_type *ext;   /* Color extension */
  size_t len;                   /* Length of name */

  const char* name;
  mode_t mode;
  int linkok;
  if (symlink_target)
    {
      name = f->linkname;
      mode = f->linkmode;
      linkok = f->linkok ? 0 : -1;
    }
  else
    {
      name = f->name;
      mode = FILE_OR_LINK_MODE (f);
      linkok = f->linkok;
    }

  /* Is this a nonexistent file?  If so, linkok == -1.  */

  if (linkok == -1 && is_colored (C_MISSING))
    type = C_MISSING;
  else if (!f->stat_ok)
    {
      static enum indicator_no filetype_indicator[] = FILETYPE_INDICATORS;
      type = filetype_indicator[f->filetype];
    }
  else
    {
      if (S_ISREG (mode))
        {
          type = C_FILE;

          if ((mode & S_ISUID) != 0 && is_colored (C_SETUID))
            type = C_SETUID;
          else if ((mode & S_ISGID) != 0 && is_colored (C_SETGID))
            type = C_SETGID;
          else if (is_colored (C_CAP) && f->has_capability)
            type = C_CAP;
          else if ((mode & S_IXUGO) != 0 && is_colored (C_EXEC))
            type = C_EXEC;
          else if ((1 < f->stat.st_nlink) && is_colored (C_MULTIHARDLINK))
            type = C_MULTIHARDLINK;
        }
      else if (S_ISDIR (mode))
        {
          type = C_DIR;

          if ((mode & S_ISVTX) && (mode & S_IWOTH)
              && is_colored (C_STICKY_OTHER_WRITABLE))
            type = C_STICKY_OTHER_WRITABLE;
          else if ((mode & S_IWOTH) != 0 && is_colored (C_OTHER_WRITABLE))
            type = C_OTHER_WRITABLE;
          else if ((mode & S_ISVTX) != 0 && is_colored (C_STICKY))
            type = C_STICKY;
        }
      else if (S_ISLNK (mode))
        type = C_LINK;
      else if (S_ISFIFO (mode))
        type = C_FIFO;
      else if (S_ISSOCK (mode))
        type = C_SOCK;
      else if (S_ISBLK (mode))
        type = C_BLK;
      else if (S_ISCHR (mode))
        type = C_CHR;
      else if (S_ISDOOR (mode))
        type = C_DOOR;
      else
        {
          /* Classify a file of some other type as C_ORPHAN.  */
          type = C_ORPHAN;
        }
    }

  /* Check the file's suffix only if still classified as C_FILE.  */
  ext = NULL;
  if (type == C_FILE)
    {
      /* Test if NAME has a recognized suffix.  */

      len = strlen (name);
      name += len;              /* Pointer to final \0.  */
      for (ext = color_ext_list; ext != NULL; ext = ext->next)
        {
          if (ext->ext.len <= len
              && STREQ_LEN (name - ext->ext.len, ext->ext.string,
                            ext->ext.len))
            break;
        }
    }

  /* Adjust the color for orphaned symlinks.  */
  if (type == C_LINK && !linkok)
    {
```

```
      if (color_symlink_as_referent || is_colored (C_ORPHAN))
        type = C_ORPHAN;
    }

  {
    const struct bin_str *const s
      = ext ? &(ext->seq) : &color_indicator[type];
    if (s->string != NULL)
      {
        /* Need to reset so not dealing with attribute combinations */
        if (is_colored (C_NORM))
          restore_default_color ();
        put_indicator (&color_indicator[C_LEFT]);
        put_indicator (s);
        put_indicator (&color_indicator[C_RIGHT]);
        return true;
      }
    else
      return false;
  }
}

/* Output a color indicator (which may contain nulls).  */
static void
put_indicator (const struct bin_str *ind)
{
  if (! used_color)
    {
      used_color = true;
      prep_non_filename_text ();
    }

  fwrite (ind->string, ind->len, 1, stdout);
}

static size_t
length_of_file_name_and_frills (const struct fileinfo *f)
{
  size_t len = 0;
  size_t name_width;
  char buf[MAX (LONGEST_HUMAN_READABLE + 1, INT_BUFSIZE_BOUND (uintmax_t))];

  if (print_inode)
    len += 1 + (format == with_commas
                ? strlen (umaxtostr (f->stat.st_ino, buf))
                : inode_number_width);

  if (print_block_size)
    len += 1 + (format == with_commas
                ? strlen (! f->stat_ok ? "?"
                          : human_readable (ST_NBLOCKS (f->stat), buf,
                                            human_output_opts, ST_NBLOCKSIZE,
                                            output_block_size))
                : block_size_width);

  if (print_scontext)
    len += 1 + (format == with_commas ? strlen (f->scontext) : scontext_width);

  quote_name (NULL, f->name, filename_quoting_options, &name_width);
  len += name_width;

  if (indicator_style != none)
    {
      char c = get_type_indicator (f->stat_ok, f->stat.st_mode, f->filetype);
      len += (c != 0);
    }

  return len;
}

static void
print_many_per_line (void)
{
  size_t row;                   /* Current row.  */
  size_t cols = calculate_columns (true);
  struct column_info const *line_fmt = &column_info[cols - 1];

  /* Calculate the number of rows that will be in each column except possibly
     for a short column on the right.  */
  size_t rows = cwd_n_used / cols + (cwd_n_used % cols != 0);

  for (row = 0; row < rows; row++)
    {
      size_t col = 0;
      size_t filesno = row;
      size_t pos = 0;

      /* Print the next row.  */
      while (1)
        {
          struct fileinfo const *f = sorted_file[filesno];
          size_t name_length = length_of_file_name_and_frills (f);
          size_t max_name_length = line_fmt->col_arr[col++];
          print_file_name_and_frills (f, pos);

          filesno += rows;
          if (filesno >= cwd_n_used)
            break;
```

```
          indent (pos + name_length, pos + max_name_length);
          pos += max_name_length;
        }
      putchar ('\n');
    }
}

static void
print_horizontal (void)
{
  size_t filesno;
  size_t pos = 0;
  size_t cols = calculate_columns (false);
  struct column_info const *line_fmt = &column_info[cols - 1];
  struct fileinfo const *f = sorted_file[0];
  size_t name_length = length_of_file_name_and_frills (f);
  size_t max_name_length = line_fmt->col_arr[0];

  /* Print first entry.  */
  print_file_name_and_frills (f, 0);

  /* Now the rest.  */
  for (filesno = 1; filesno < cwd_n_used; ++filesno)
    {
      size_t col = filesno % cols;

      if (col == 0)
        {
          putchar ('\n');
          pos = 0;
        }
      else
        {
          indent (pos + name_length, pos + max_name_length);
          pos += max_name_length;
        }

      f = sorted_file[filesno];
      print_file_name_and_frills (f, pos);

      name_length = length_of_file_name_and_frills (f);
      max_name_length = line_fmt->col_arr[col];
    }
  putchar ('\n');
}

static void
print_with_commas (void)
{
  size_t filesno;
  size_t pos = 0;

  for (filesno = 0; filesno < cwd_n_used; filesno++)
    {
      struct fileinfo const *f = sorted_file[filesno];
      size_t len = length_of_file_name_and_frills (f);

      if (filesno != 0)
        {
          char separator;

          if (pos + len + 2 < line_length)
            {
              pos += 2;
              separator = ' ';
            }
          else
            {
              pos = 0;
              separator = '\n';
            }

          putchar (',');
          putchar (separator);
        }

      print_file_name_and_frills (f, pos);
      pos += len;
    }
  putchar ('\n');
}

/* Assuming cursor is at position FROM, indent up to position TO.
   Use a TAB character instead of two or more spaces whenever possible.  */

static void
indent (size_t from, size_t to)
{
  while (from < to)
    {
      if (tabsize != 0 && to / tabsize > (from + 1) / tabsize)
        {
          putchar ('\t');
          from += tabsize - from % tabsize;
        }
      else
        {
```

```
          putchar (' ');
          from++;
        }
    }
}

/* Put DIRNAME/NAME into DEST, handling '.' and '/' properly.  */
/* FIXME: maybe remove this function someday.  See about using a
   non-malloc'ing version of file_name_concat.  */

static void
attach (char *dest, const char *dirname, const char *name)
{
  const char *dirnamep = dirname;

  /* Copy dirname if it is not ".".  */
  if (dirname[0] != '.' || dirname[1] != 0)
    {
      while (*dirnamep)
        *dest++ = *dirnamep++;
      /* Add '/' if 'dirname' doesn't already end with it.  */
      if (dirnamep > dirname && dirnamep[-1] != '/')
        *dest++ = '/';
    }
  while (*name)
    *dest++ = *name++;
  *dest = 0;
}

/* Allocate enough column info suitable for the current number of
   files and display columns, and initialize the info to represent the
   narrowest possible columns.  */

static void
init_column_info (void)
{
  size_t i;
  size_t max_cols = MIN (max_idx, cwd_n_used);

  /* Currently allocated columns in column_info.  */
  static size_t column_info_alloc;

  if (column_info_alloc < max_cols)
    {
      size_t new_column_info_alloc;
      size_t *p;

      if (max_cols < max_idx / 2)
        {
          /* The number of columns is far less than the display width
             allows.  Grow the allocation, but only so that it's
             double the current requirements.  If the display is
             extremely wide, this avoids allocating a lot of memory
             that is never needed.  */
          column_info = xnrealloc (column_info, max_cols,
                                   2 * sizeof *column_info);
          new_column_info_alloc = 2 * max_cols;
        }
      else
        {
          column_info = xnrealloc (column_info, max_idx, sizeof *column_info);
          new_column_info_alloc = max_idx;
        }

      /* Allocate the new size_t objects by computing the triangle
         formula n * (n + 1) / 2, except that we don't need to
         allocate the part of the triangle that we've already
         allocated.  Check for address arithmetic overflow.  */
      {
        size_t column_info_growth = new_column_info_alloc - column_info_alloc;
        size_t s = column_info_alloc + 1 + new_column_info_alloc;
        size_t t = s * column_info_growth;
        if (s < new_column_info_alloc || t / column_info_growth != s)
          xalloc_die ();
        p = xnmalloc (t / 2, sizeof *p);
      }

      /* Grow the triangle by parceling out the cells just allocated.  */
      for (i = column_info_alloc; i < new_column_info_alloc; i++)
        {
          column_info[i].col_arr = p;
          p += i + 1;
        }

      column_info_alloc = new_column_info_alloc;
    }

  for (i = 0; i < max_cols; ++i)
    {
      size_t j;

      column_info[i].valid_len = true;
      column_info[i].line_len = (i + 1) * MIN_COLUMN_WIDTH;
      for (j = 0; j <= i; ++j)
        column_info[i].col_arr[j] = MIN_COLUMN_WIDTH;
    }
}
```

```
/* Calculate the number of columns needed to represent the current set
   of files in the current display width.  */

static size_t
calculate_columns (bool by_columns)
{
  size_t filesno;               /* Index into cwd_file.  */
  size_t cols;                  /* Number of files across.  */

  /* Normally the maximum number of columns is determined by the
     screen width.  But if few files are available this might limit it
     as well.  */
  size_t max_cols = MIN (max_idx, cwd_n_used);

  init_column_info ();

  /* Compute the maximum number of possible columns.  */
  for (filesno = 0; filesno < cwd_n_used; ++filesno)
    {
      struct fileinfo const *f = sorted_file[filesno];
      size_t name_length = length_of_file_name_and_frills (f);
      size_t i;

      for (i = 0; i < max_cols; ++i)
        {
          if (column_info[i].valid_len)
            {
              size_t idx = (by_columns
                            ? filesno / ((cwd_n_used + i) / (i + 1))
                            : filesno % (i + 1));
              size_t real_length = name_length + (idx == i ? 0 : 2);

              if (column_info[i].col_arr[idx] < real_length)
                {
                  column_info[i].line_len += (real_length
                                              - column_info[i].col_arr[idx]);
                  column_info[i].col_arr[idx] = real_length;
                  column_info[i].valid_len = (column_info[i].line_len
                                              < line_length);
                }
            }
        }
    }

  /* Find maximum allowed columns.  */
  for (cols = max_cols; 1 < cols; --cols)
    {
      if (column_info[cols - 1].valid_len)
        break;
    }

  return cols;
}

void
usage (int status)
{
  if (status != EXIT_SUCCESS)
    emit_try_help ();
  else
    {
      printf (_("Usage: %s [OPTION]... [FILE]...\n"), program_name);
      fputs (_("\
List information about the FILEs (the current directory by default).\n\
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.\n\
"), stdout);

      emit_mandatory_arg_note ();

      fputs (_("\
  -a, --all                  do not ignore entries starting with .\n\
  -A, --almost-all           do not list implied . and ..\n\
      --author               with -l, print the author of each file\n\
  -b, --escape               print C-style escapes for nongraphic characters\n\
"), stdout);
      fputs (_("\
      --block-size=SIZE      scale sizes by SIZE before printing them; e.g.,\n\
                               '--block-size=M' prints sizes in units of\n\
                               1,048,576 bytes; see SIZE format below\n\
  -B, --ignore-backups       do not list implied entries ending with ~\n\
  -c                         with -lt: sort by, and show, ctime (time of last\n\
                               modification of file status information);\n\
                               with -l: show ctime and sort by name;\n\
                               otherwise: sort by ctime, newest first\n\
"), stdout);
      fputs (_("\
  -C                         list entries by columns\n\
      --color[=WHEN]         colorize the output; WHEN can be 'always' (default\
\n\
                               if omitted), 'auto', or 'never'; more info below\
\n\
  -d, --directory            list directories themselves, not their contents\n\
  -D, --dired                generate output designed for Emacs' dired mode\n\
"), stdout);
      fputs (_("\
  -f                         do not sort, enable -aU, disable -ls --color\n\
  -F, --classify             append indicator (one of */=>@|) to entries\n\
      --file-type            likewise, except do not append '*'\n\
```

```
      --format=WORD          across -x, commas -m, horizontal -x, long -l,\n\
                               single-column -1, verbose -l, vertical -C\n\
      --full-time            like -l --time-style=full-iso\n\
"), stdout);
      fputs (_("\
  -g                         like -l, but do not list owner\n\
"), stdout);
      fputs (_("\
      --group-directories-first\n\
                             group directories before files;\n\
                               can be augmented with a --sort option, but any\n\
                               use of --sort=none (-U) disables grouping\n\
"), stdout);
      fputs (_("\
  -G, --no-group             in a long listing, don't print group names\n\
  -h, --human-readable       with -l and/or -s, print human readable sizes\n\
                               (e.g., 1K 234M 2G)\n\
      --si                   likewise, but use powers of 1000 not 1024\n\
"), stdout);
      fputs (_("\
  -H, --dereference-command-line\n\
                             follow symbolic links listed on the command line\n\
      --dereference-command-line-symlink-to-dir\n\
                             follow each command line symbolic link\n\
                               that points to a directory\n\
      --hide=PATTERN         do not list implied entries matching shell PATTERN\
\n\
                               (overridden by -a or -A)\n\
"), stdout);
      fputs (_("\
      --indicator-style=WORD  append indicator with style WORD to entry names:\
\n\
                               none (default), slash (-p),\n\
                               file-type (--file-type), classify (-F)\n\
  -i, --inode                print the index number of each file\n\
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN\
\n\
  -k, --kibibytes            default to 1024-byte blocks for disk usage\n\
"), stdout);
      fputs (_("\
  -l                         use a long listing format\n\
  -L, --dereference          when showing file information for a symbolic\n\
                               link, show information for the file the link\n\
                               references rather than for the link itself\n\
  -m                         fill width with a comma separated list of entries\
\n\
"), stdout);
      fputs (_("\
  -n, --numeric-uid-gid      like -l, but list numeric user and group IDs\n\
  -N, --literal              print raw entry names (don't treat e.g. control\n\
                               characters specially)\n\
  -o                         like -l, but do not list group information\n\
  -p, --indicator-style=slash\n\
                             append / indicator to directories\n\
"), stdout);
      fputs (_("\
  -q, --hide-control-chars   print ? instead of nongraphic characters\n\
      --show-control-chars   show nongraphic characters as-is (the default,\n\
                               unless program is 'ls' and output is a terminal)\
\n\
  -Q, --quote-name           enclose entry names in double quotes\n\
      --quoting-style=WORD   use quoting style WORD for entry names:\n\
                               literal, locale, shell, shell-always, c, escape\
\n\
"), stdout);
      fputs (_("\
  -r, --reverse              reverse order while sorting\n\
  -R, --recursive            list subdirectories recursively\n\
  -s, --size                 print the allocated size of each file, in blocks\n\
"), stdout);
      fputs (_("\
  -S                         sort by file size, largest first\n\
      --sort=WORD            sort by WORD instead of name: none (-U), size (-S)\
,\n\
                               time (-t), version (-v), extension (-X)\n\
      --time=WORD            with -l, show time as WORD instead of default\n\
                               modification time: atime or access or use (-u);\
\n\
                               ctime or status (-c); also use specified time\n\
                               as sort key if --sort=time (newest first)\n\
"), stdout);
      fputs (_("\
      --time-style=STYLE     with -l, show times using style STYLE:\n\
                               full-iso, long-iso, iso, locale, or +FORMAT;\n\
                               FORMAT is interpreted like in 'date'; if FORMAT\
\n\
                               is FORMAT1<newline>FORMAT2, then FORMAT1 applies\
\n\
                               to non-recent files and FORMAT2 to recent files;\
\n\
                               if STYLE is prefixed with 'posix-', STYLE\n\
                               takes effect only outside the POSIX locale\n\
"), stdout);
      fputs (_("\
  -t                         sort by modification time, newest first\n\
  -T, --tabsize=COLS         assume tab stops at each COLS instead of 8\n\
"), stdout);
      fputs (_("\
```

```
  -u                         with -lt: sort by, and show, access time;\n\
                               with -l: show access time and sort by name;\n\
                               otherwise: sort by access time, newest first\n\
  -U                         do not sort; list entries in directory order\n\
  -v                         natural sort of (version) numbers within text\n\
"), stdout);
      fputs (_("\
  -w, --width=COLS           assume screen width instead of current value\n\
  -x                         list entries by lines instead of by columns\n\
  -X                         sort alphabetically by entry extension\n\
  -Z, --context              print any security context of each file\n\
  -1                         list one file per line.  Avoid '\\n' with -q or -b\
\n\
"), stdout);
      fputs (HELP_OPTION_DESCRIPTION, stdout);
      fputs (VERSION_OPTION_DESCRIPTION, stdout);
      emit_size_note ();
      fputs (_("\
\n\
Using color to distinguish file types is disabled both by default and\n\
with --color=never.  With --color=auto, ls emits color codes only when\n\
standard output is connected to a terminal.  The LS_COLORS environment\n\
variable can change the settings.  Use the dircolors command to set it.\n\
"), stdout);
      fputs (_("\
\n\
Exit status:\n\
 0  if OK,\n\
 1  if minor problems (e.g., cannot access subdirectory),\n\
 2  if serious trouble (e.g., cannot access command-line argument).\n\
"), stdout);
      emit_ancillary_info (PROGRAM_NAME);
    }
  exit (status);
}
```

Notice for package(s)

sed

---

modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding

Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met. This License explicitly affirms your unlimited
permission to run the unmodified Program. The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work. This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force. You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright. Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below. Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7. This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy. This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged. This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit. Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do

not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
```

```
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

Notice for package(s)

sed

```
/*  GNU SED, a batch stream editor.
    Copyright (C) 1989,90,91,92,93,94,95,98,99,2002,2003
    Free Software Foundation, Inc.

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 3, or (at your option)
    any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. */

#include "config.h"
#include "basicdefs.h"
#include "regex.h"
#include <stdio.h>
#include "unlocked-io.h"

#include "utils.h"

/* Struct vector is used to describe a compiled sed program. */
struct vector {
  struct sed_cmd *v;    /* a dynamically allocated array */
  size_t v_allocated;   /* ... number slots allocated */
  size_t v_length;      /* ... number of slots in use */
};

/* This structure tracks files used by sed so that they may all be
   closed cleanly at normal program termination.  A flag is kept that tells
   if a missing newline was encountered, so that it is added on the
   next line and the two lines are not concatenated.  */
struct output {
  char *name;
  bool missing_newline;
  FILE *fp;
  struct output *link;
};

struct text_buf {
  char *text;
  size_t text_length;
};
```

```
struct regex {
  regex_t pattern;
  int flags;
  size_t sz;
  char re[1];
};

enum replacement_types {
  REPL_ASIS = 0,
  REPL_UPPERCASE = 1,
  REPL_LOWERCASE = 2,
  REPL_UPPERCASE_FIRST = 4,
  REPL_LOWERCASE_FIRST = 8,
  REPL_MODIFIERS = REPL_UPPERCASE_FIRST | REPL_LOWERCASE_FIRST,

  /* These are given to aid in debugging */
  REPL_UPPERCASE_UPPERCASE = REPL_UPPERCASE_FIRST | REPL_UPPERCASE,
  REPL_UPPERCASE_LOWERCASE = REPL_UPPERCASE_FIRST | REPL_LOWERCASE,
  REPL_LOWERCASE_UPPERCASE = REPL_LOWERCASE_FIRST | REPL_UPPERCASE,
  REPL_LOWERCASE_LOWERCASE = REPL_LOWERCASE_FIRST | REPL_LOWERCASE
};

enum text_types {
  TEXT_BUFFER,
  TEXT_REPLACEMENT,
  TEXT_REGEX
};

enum posixicity_types {
  POSIXLY_EXTENDED,     /* with GNU extensions */
  POSIXLY_CORRECT,      /* with POSIX-compatible GNU extensions */
  POSIXLY_BASIC         /* pedantically POSIX */
};

enum addr_state {
  RANGE_INACTIVE,       /* never been active */
  RANGE_ACTIVE,         /* between first and second address */
  RANGE_CLOSED          /* like RANGE_INACTIVE, but range has ended once */
};

enum addr_types {
  ADDR_IS_NULL,         /* null address */
  ADDR_IS_REGEX,        /* a.addr_regex is valid */
  ADDR_IS_NUM,          /* a.addr_number is valid */
  ADDR_IS_NUM_MOD,      /* a.addr_number is valid, addr_step is modulo */
  ADDR_IS_STEP,         /* address is +N (only valid for addr2) */
  ADDR_IS_STEP_MOD,     /* address is ~N (only valid for addr2) */
  ADDR_IS_LAST          /* address is $ */
};

struct addr {
  enum addr_types addr_type;
  countT addr_number;
  countT addr_step;
  struct regex *addr_regex;
};


struct replacement {
  char *prefix;
  size_t prefix_length;
  int subst_id;
  enum replacement_types repl_type;
  struct replacement *next;
};

struct subst {
  struct regex *regx;
  struct replacement *replacement;
  countT numb;          /* if >0, only substitute for match number "numb" */
  struct output *outf;  /* 'w' option given */
  unsigned global : 1;  /* 'g' option given */
  unsigned print : 2;   /* 'p' option given (before/after eval) */
  unsigned eval : 1;    /* 'e' option given */
  unsigned max_id : 4;  /* maximum backreference on the RHS */
};

#ifdef REG_PERL
/* This is the structure we store register match data in.  See
   regex.texinfo for a full description of what registers match.  */
struct re_registers
{
  unsigned num_regs;
  regoff_t *start;
  regoff_t *end;
};
#endif



struct sed_cmd {
  struct addr *a1;      /* save space: usually is NULL */
  struct addr *a2;

  /* See description the enum, above.  */
  enum addr_state range_state;

  /* Non-zero if command is to be applied to non-matches. */
```

```
  char addr_bang;

  /* The actual command character. */
  char cmd;

  /* auxiliary data for various commands */
  union {
    /* This structure is used for a, i, and c commands. */
    struct text_buf cmd_txt;

    /* This is used for the l, q and Q commands. */
    int int_arg;

    /* This is used for the {}, b, and t commands. */
    countT jump_index;

    /* This is used for the r command. */
    char *fname;

    /* This is used for the hairy s command. */
    struct subst *cmd_subst;

    /* This is used for the w command. */
    struct output *outf;

    /* This is used for the R command. */
    FILE *fp;

    /* This is used for the y command. */
    unsigned char *translate;
    char **translatemb;
  } x;
};



void bad_prog (const char *why);
size_t normalize_text (char *text, size_t len, enum text_types buftype);
struct vector *compile_string (struct vector *, char *str, size_t len);
struct vector *compile_file (struct vector *, const char *cmdfile);
void check_final_program (struct vector *);
void rewind_read_files (void);
void finish_program (struct vector *);

struct regex *compile_regex (struct buffer *b, int flags, int needed_sub);
int match_regex (struct regex *regex,
                 char *buf, size_t buflen, size_t buf_start_offset,
                 struct re_registers *regarray, int regsize);
#ifdef DEBUG_LEAKS
void release_regex (struct regex *);
#endif

int process_files (struct vector *, char **argv);

int main (int, char **);

extern void fmt (const char *line, const char *line_end, int max_length, FILE *output_file);

extern int extended_regexp_flags;

/* one-byte buffer delimiter */
extern char buffer_delimiter;

/* If set, fflush(stdout) on every line output,
   and turn off stream buffering on inputs.  */
extern bool unbuffered;

/* If set, don't write out the line unless explicitly told to. */
extern bool no_default_output;

/* If set, reset line counts on every new file. */
extern bool separate_files;

/* If set, follow symlinks when invoked with -i option */
extern bool follow_symlinks;

/* Do we need to be pedantically POSIX compliant? */
extern enum posixicity_types posixicity;

/* How long should the `l' command's output line be? */
extern countT lcmd_out_line_len;

/* How do we edit files in-place? (we don't if NULL) */
extern char *in_place_extension;

/* The mode to use to read and write files, either "rt"/"w" or "rb"/"wb".  */
extern char *read_mode;
extern char *write_mode;

/* Should we use EREs? */
extern bool use_extended_syntax_p;

/* Declarations for multibyte character sets.  */
extern int mb_cur_max;
extern bool is_utf8;

#define MBRTOWC(pwc, s, n, ps) \
  (mb_cur_max == 1 ? \
```

```
   (*(pwc) = btowc (*(unsigned char *) (s)), 1) : \
   mbrtowc ((pwc), (s), (n), (ps)))

#define WCRTOMB(s, wc, ps) \
  (mb_cur_max == 1 ? \
   (*(s) = wctob ((wint_t) (wc)), 1) : \
   wcrtomb ((s), (wc), (ps)))

#define MBSINIT(s) \
  (mb_cur_max == 1 ? 1 : mbsinit ((s)))

#define MBRLEN(s, n, ps) \
  (mb_cur_max == 1 ? 1 : mbrtowc (NULL, s, n, ps))

#define BRLEN(ch, ps) \
  (mb_cur_max == 1 ? 1 : brlen (ch, ps))

extern int brlen (int ch, mbstate_t *ps);
extern void initialize_mbcs (void);
```

Notice for package(s)

openvswitch

```
This file is a summary of the licensing of files in this distribution.
Some files may be marked specifically with a different license, in
which case that license applies to the file in question.

Most files are licensed under the Apache License, Version 2.0:

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at:

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.

Files under the datapath directory are licensed under the GNU General
Public License, version 2.

File build-aux/cccl is licensed under the GNU General Public
License, version 2.

The following files are licensed under the 2-clause BSD license.
    include/windows/getopt.h
    lib/getopt_long.c
    lib/conntrack-tcp.c

The following files are licensed under the 3-clause BSD-license
    include/windows/netinet/icmp6.h
    include/windows/netinet/ip6.h
    lib/strsep.c

Files under the xenserver directory are licensed on a file-by-file
basis.  Refer to each file for details.

Files lib/sflow*.[ch] are licensed under the terms of either the
Sun Industry Standards Source License 1.1, that is available at:
        http://host-sflow.sourceforge.net/sissl.html
or the InMon sFlow License, that is available at:
        http://www.inmon.com/technology/sflowlicense.txt
```

Notice for package(s)

iptables

```
/* Code to take an iptables-style command line and do it. */

/*
 * Author: Paul.Russell@rustcorp.com.au and mneuling@radlogic.com.au
 *
 * (C) 2000-2002 by the netfilter coreteam <coreteam@netfilter.org>:
 *                  Paul 'Rusty' Russell <rusty@rustcorp.com.au>
 *                  Marc Boucher <marc+nf@mbsi.ca>
 *                  James Morris <jmorris@intercode.com.au>
 *                  Harald Welte <laforge@gnumonks.org>
 *                  Jozsef Kadlecsik <kadlec@blackhole.kfki.hu>
 *
 *      This program is free software; you can redistribute it and/or modify
```

```
 *      it under the terms of the GNU General Public License as published by
 *      the Free Software Foundation; either version 2 of the License, or
 *      (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful,
 *      but WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *      GNU General Public License for more details.
 *
 *      You should have received a copy of the GNU General Public License
 *      along with this program; if not, write to the Free Software
 *      Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <getopt.h>
#include <string.h>
#include <netdb.h>
#include <errno.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdarg.h>
#include <limits.h>
#include <unistd.h>
#include <iptables.h>
#include <xtables.h>
#include <fcntl.h>
#include "xshared.h"

#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif

#define CMD_NONE                0x0000U
#define CMD_INSERT              0x0001U
#define CMD_DELETE              0x0002U
#define CMD_DELETE_NUM          0x0004U
#define CMD_REPLACE             0x0008U
#define CMD_APPEND              0x0010U
#define CMD_LIST                0x0020U
#define CMD_FLUSH               0x0040U
#define CMD_ZERO                0x0080U
#define CMD_NEW_CHAIN           0x0100U
#define CMD_DELETE_CHAIN        0x0200U
#define CMD_SET_POLICY          0x0400U
#define CMD_RENAME_CHAIN        0x0800U
#define CMD_LIST_RULES          0x1000U
#define CMD_ZERO_NUM            0x2000U
#define CMD_CHECK               0x4000U
#define NUMBER_OF_CMD   16
static const char cmdflags[] = { 'I', 'D', 'D', 'R', 'A', 'L', 'F', 'Z',
                                 'N', 'X', 'P', 'E', 'S', 'Z', 'C' };

#define OPT_FRAGMENT    0x00800U
#define NUMBER_OF_OPT   ARRAY_SIZE(optflags)
static const char optflags[]
= { 'n', 's', 'd', 'p', 'j', 'v', 'x', 'i', 'o', '0', 'c', 'f'};

static struct option original_opts[] = {
        {.name = "append",        .has_arg = 1, .val = 'A'},
        {.name = "delete",        .has_arg = 1, .val = 'D'},
        {.name = "check",         .has_arg = 1, .val = 'C'},
        {.name = "insert",        .has_arg = 1, .val = 'I'},
        {.name = "replace",       .has_arg = 1, .val = 'R'},
        {.name = "list",          .has_arg = 2, .val = 'L'},
        {.name = "list-rules",    .has_arg = 2, .val = 'S'},
        {.name = "flush",         .has_arg = 2, .val = 'F'},
        {.name = "zero",          .has_arg = 2, .val = 'Z'},
        {.name = "new-chain",     .has_arg = 1, .val = 'N'},
        {.name = "delete-chain",  .has_arg = 2, .val = 'X'},
        {.name = "rename-chain",  .has_arg = 1, .val = 'E'},
        {.name = "policy",        .has_arg = 1, .val = 'P'},
        {.name = "source",        .has_arg = 1, .val = 's'},
        {.name = "destination",   .has_arg = 1, .val = 'd'},
        {.name = "src",           .has_arg = 1, .val = 's'}, /* synonym */
        {.name = "dst",           .has_arg = 1, .val = 'd'}, /* synonym */
        {.name = "protocol",      .has_arg = 1, .val = 'p'},
        {.name = "in-interface",  .has_arg = 1, .val = 'i'},
        {.name = "jump",          .has_arg = 1, .val = 'j'},
        {.name = "table",         .has_arg = 1, .val = 't'},
        {.name = "match",         .has_arg = 1, .val = 'm'},
        {.name = "numeric",       .has_arg = 0, .val = 'n'},
        {.name = "out-interface", .has_arg = 1, .val = 'o'},
        {.name = "verbose",       .has_arg = 0, .val = 'v'},
        {.name = "wait",          .has_arg = 0, .val = 'w'},
        {.name = "exact",         .has_arg = 0, .val = 'x'},
        {.name = "fragments",     .has_arg = 0, .val = 'f'},
        {.name = "version",       .has_arg = 0, .val = 'V'},
        {.name = "help",          .has_arg = 2, .val = 'h'},
        {.name = "line-numbers",  .has_arg = 0, .val = '0'},
        {.name = "modprobe",      .has_arg = 1, .val = 'M'},
        {.name = "set-counters",  .has_arg = 1, .val = 'c'},
        {.name = "goto",          .has_arg = 1, .val = 'g'},
        {.name = "ipv4",          .has_arg = 0, .val = '4'},
```

```
        {.name = "ipv6",          .has_arg = 0, .val = '6'},
        {NULL},
};

void iptables_exit_error(enum xtables_exittype status, const char *msg, ...) __attribute__((noreturn, format(printf,2,3)));

struct xtables_globals iptables_globals = {
        .option_offset = 0,
        .program_version = IPTABLES_VERSION,
        .orig_opts = original_opts,
        .exit_err = iptables_exit_error,
};

/* Table of legal combinations of commands and options.  If any of the
 * given commands make an option legal, that option is legal (applies to
 * CMD_LIST and CMD_ZERO only).
 * Key:
 *  +  compulsory
 *  x  illegal
 *     optional
 */

static const char commands_v_options[NUMBER_OF_CMD][NUMBER_OF_OPT] =
/* Well, it's better than "Re: Linux vs FreeBSD" */
{
        /*     -n  -s  -d  -p  -j  -v  -x  -i  -o --line -c -f */
/*INSERT*/    {'x',' ',' ',' ',' ',' ','x',' ',' ','x',' ',' '},
/*DELETE*/    {'x',' ',' ',' ',' ',' ','x',' ',' ','x','x',' '},
/*DELETE_NUM*/{'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*REPLACE*/   {'x',' ',' ',' ',' ',' ','x',' ',' ','x',' ',' '},
/*APPEND*/    {'x',' ',' ',' ',' ',' ','x',' ',' ','x',' ',' '},
/*LIST*/      {' ','x','x','x','x','x',' ',' ','x','x','x',' ','x','x'},
/*FLUSH*/     {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*ZERO*/      {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*NEW_CHAIN*/ {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*DEL_CHAIN*/ {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*SET_POLICY*/{'x','x','x','x','x','x',' ','x','x','x','x',' ','x'},
/*RENAME*/    {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*LIST_RULES*/{'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*ZERO_NUM*/  {'x','x','x','x','x','x',' ','x','x','x','x','x','x'},
/*CHECK*/     {'x',' ',' ',' ',' ',' ','x',' ',' ','x','x',' '},
};

static const int inverse_for_options[NUMBER_OF_OPT] =
{
/* -n */ 0,
/* -s */ IPT_INV_SRCIP,
/* -d */ IPT_INV_DSTIP,
/* -p */ XT_INV_PROTO,
/* -j */ 0,
/* -v */ 0,
/* -x */ 0,
/* -i */ IPT_INV_VIA_IN,
/* -o */ IPT_INV_VIA_OUT,
/*--line*/ 0,
/* -c */ 0,
/* -f */ IPT_INV_FRAG,
};

#define opts iptables_globals.opts
#define prog_name iptables_globals.program_name
#define prog_vers iptables_globals.program_version

static void __attribute__((noreturn))
exit_tryhelp(int status)
{
        if (line != -1)
                fprintf(stderr, "Error occurred at line: %d\n", line);
        fprintf(stderr, "Try `%s -h' or '%s --help' for more information.\n",
                        prog_name, prog_name);
        xtables_free_opts(1);
        exit(status);
}

static void
exit_printhelp(const struct xtables_rule_match *matches)
{
        printf("%s v%s\n\n"
"Usage: %s -[ACD] chain rule-specification [options]\n"
"       %s -I chain [rulenum] rule-specification [options]\n"
"       %s -R chain rulenum rule-specification [options]\n"
"       %s -D chain rulenum [options]\n"
"       %s -[LS] [chain [rulenum]] [options]\n"
"       %s -[FZ] [chain] [options]\n"
"       %s -[NX] chain\n"
"       %s -E old-chain-name new-chain-name\n"
"       %s -P chain target [options]\n"
"       %s -h (print this help information)\n\n",
               prog_name, prog_vers, prog_name, prog_name,
               prog_name, prog_name, prog_name, prog_name,
               prog_name, prog_name, prog_name, prog_name);

        printf(
"Commands:\n"
"Either long or short options are allowed.\n"
"  --append  -A chain           Append to chain\n"
"  --check   -C chain           Check for the existence of a rule\n"
"  --delete  -D chain           Delete matching rule from chain\n"
```

```
"  --delete  -D chain rulenum\n"
"                               Delete rule rulenum (1 = first) from chain\n"
"  --insert  -I chain [rulenum]\n"
"                               Insert in chain as rulenum (default 1=first)\n"
"  --replace -R chain rulenum\n"
"                               Replace rule rulenum (1 = first) in chain\n"
"  --list    -L [chain [rulenum]]\n"
"                               List the rules in a chain or all chains\n"
"  --list-rules -S [chain [rulenum]]\n"
"                               Print the rules in a chain or all chains\n"
"  --flush   -F [chain]         Delete all rules in  chain or all chains\n"
"  --zero    -Z [chain [rulenum]]\n"
"                               Zero counters in chain or all chains\n"
"  --new     -N chain           Create a new user-defined chain\n"
"  --delete-chain\n"
"            -X [chain]         Delete a user-defined chain\n"
"  --policy  -P chain target\n"
"                               Change policy on chain to target\n"
"  --rename-chain\n"
"            -E old-chain new-chain\n"
"                               Change chain name, (moving any references)\n"

"Options:\n"
"    --ipv4     -4              Nothing (line is ignored by ip6tables-restore)\n"
"    --ipv6     -6              Error (line is ignored by iptables-restore)\n"
"[!] --protocol -p proto        protocol: by number or name, eg. `tcp'\n"
"[!] --source   -s address[/mask][...]\n"
"                               source specification\n"
"[!] --destination -d address[/mask][...]\n"
"                               destination specification\n"
"[!] --in-interface -i input name[+]\n"
"                               network interface name ([+] for wildcard)\n"
" --jump        -j target\n"
"                               target for rule (may load target extension)\n"
#ifdef IPT_F_GOTO
"  --goto      -g chain\n"
"                              jump to chain with no return\n"
#endif
"  --match      -m match\n"
"                               extended match (may load extension)\n"
"  --numeric    -n              numeric output of addresses and ports\n"
"[!] --out-interface -o output name[+]\n"
"                               network interface name ([+] for wildcard)\n"
"  --table      -t table        table to manipulate (default: `filter')\n"
"  --verbose    -v              verbose mode\n"
"  --wait       -w              wait for the xtables lock\n"
"  --line-numbers               print line numbers when listing\n"
"  --exact      -x              expand numbers (display exact values)\n"
"[!] --fragment -f              match second or further fragments only\n"
"  --modprobe=<command>         try to insert modules using this command\n"
"  --set-counters PKTS BYTES    set the counter during insert/append\n"
"[!] --version  -V              print package version.\n");

        print_extension_helps(xtables_targets, matches);
        exit(0);
}

void
iptables_exit_error(enum xtables_exittype status, const char *msg, ...)
{
        va_list args;

        va_start(args, msg);
        fprintf(stderr, "%s v%s: ", prog_name, prog_vers);
        vfprintf(stderr, msg, args);
        va_end(args);
        fprintf(stderr, "\n");
        if (status == PARAMETER_PROBLEM)
                exit_tryhelp(status);
        if (status == VERSION_PROBLEM)
                fprintf(stderr,
                        "Perhaps iptables or your kernel needs to be upgraded.\n");
        /* On error paths, make sure that we don't leak memory */
        xtables_free_opts(1);
        exit(status);
}

static void
generic_opt_check(int command, int options)
{
        int i, j, legal = 0;

        /* Check that commands are valid with options.  Complicated by the
         * fact that if an option is legal with *any* command given, it is
         * legal overall (ie. -z and -l).
         */
        for (i = 0; i < NUMBER_OF_OPT; i++) {
                legal = 0; /* -1 => illegal, 1 => legal, 0 => undecided. */

                for (j = 0; j < NUMBER_OF_CMD; j++) {
                        if (!(command & (1<<j)))
                                continue;

                        if (!(options & (1<<i))) {
                                if (commands_v_options[j][i] == '+')
                                        xtables_error(PARAMETER_PROBLEM,
                                                      "You need to supply the `-%c' "
                                                      "option for this command\n",
```

```
                                                optflags[i]);
                        } else {
                                if (commands_v_options[j][i] != 'x')
                                        legal = 1;
                                else if (legal == 0)
                                        legal = -1;
                        }
                }
                if (legal == -1)
                        xtables_error(PARAMETER_PROBLEM,
                                   "Illegal option `-%c' with this command\n",
                                   optflags[i]);
        }
}

static char
opt2char(int option)
{
        const char *ptr;
        for (ptr = optflags; option > 1; option >>= 1, ptr++);

        return *ptr;
}

static char
cmd2char(int option)
{
        const char *ptr;
        for (ptr = cmdflags; option > 1; option >>= 1, ptr++);

        return *ptr;
}

static void
add_command(unsigned int *cmd, const int newcmd, const int othercmds,
            int invert)
{
        if (invert)
                xtables_error(PARAMETER_PROBLEM, "unexpected ! flag");
        if (*cmd & (~othercmds))
                xtables_error(PARAMETER_PROBLEM, "Cannot use -%c with -%c\n",
                           cmd2char(newcmd), cmd2char(*cmd & (~othercmds)));
        *cmd |= newcmd;
}

/*
 *      All functions starting with "parse" should succeed, otherwise
 *      the program fails.
 *      Most routines return pointers to static data that may change
 *      between calls to the same or other routines with a few exceptions:
 *      "host_to_addr", "parse_hostnetwork", and "parse_hostnetworkmask"
 *      return global static data.
*/

/* Christophe Burki wants `-p 6' to imply `-m tcp'.  */
/* Can't be zero. */
static int
parse_rulenumber(const char *rule)
{
        unsigned int rulenum;

        if (!xtables_strtoui(rule, NULL, &rulenum, 1, INT_MAX))
                xtables_error(PARAMETER_PROBLEM,
                           "Invalid rule number `%s'", rule);

        return rulenum;
}

static void
parse_chain(const char *chainname)
{
        const char *ptr;

        if (strlen(chainname) >= XT_EXTENSION_MAXNAMELEN)
                xtables_error(PARAMETER_PROBLEM,
                           "chain name `%s' too long (must be under %u chars)",
                           chainname, XT_EXTENSION_MAXNAMELEN);

        if (*chainname == '-' || *chainname == '!')
                xtables_error(PARAMETER_PROBLEM,
                           "chain name not allowed to start "
                           "with `%c'\n", *chainname);

        if (xtables_find_target(chainname, XTF_TRY_LOAD))
                xtables_error(PARAMETER_PROBLEM,
                           "chain name may not clash "
                           "with target name\n");

        for (ptr = chainname; *ptr; ptr++)
                if (isspace(*ptr))
                        xtables_error(PARAMETER_PROBLEM,
                                   "Invalid chain name `%s'", chainname);
}

static const char *
parse_target(const char *targetname)
{
        const char *ptr;
```

```
        if (strlen(targetname) < 1)
                xtables_error(PARAMETER_PROBLEM,
                           "Invalid target name (too short)");

        if (strlen(targetname) >= XT_EXTENSION_MAXNAMELEN)
                xtables_error(PARAMETER_PROBLEM,
                           "Invalid target name `%s' (%u chars max)",
                           targetname, XT_EXTENSION_MAXNAMELEN - 1);

        for (ptr = targetname; *ptr; ptr++)
                if (isspace(*ptr))
                        xtables_error(PARAMETER_PROBLEM,
                                   "Invalid target name `%s'", targetname);
        return targetname;
}

static void
set_option(unsigned int *options, unsigned int option, uint8_t *invflg,
           int invert)
{
        if (*options & option)
                xtables_error(PARAMETER_PROBLEM, "multiple -%c flags not allowed",
                           opt2char(option));
        *options |= option;

        if (invert) {
                unsigned int i;
                for (i = 0; 1 << i != option; i++);

                if (!inverse_for_options[i])
                        xtables_error(PARAMETER_PROBLEM,
                                   "cannot have ! before -%c",
                                   opt2char(option));
                *invflg |= inverse_for_options[i];
        }
}

static void
print_header(unsigned int format, const char *chain, struct xtc_handle *handle)
{
        struct xt_counters counters;
        const char *pol = iptc_get_policy(chain, &counters, handle);
        printf("Chain %s", chain);
        if (pol) {
                printf(" (policy %s", pol);
                if (!(format & FMT_NOCOUNTS)) {
                        fputc(' ', stdout);
                        xtables_print_num(counters.pcnt, (format|FMT_NOTABLE));
                        fputs("packets, ", stdout);
                        xtables_print_num(counters.bcnt, (format|FMT_NOTABLE));
                        fputs("bytes", stdout);
                }
                printf(")\n");
        } else {
                unsigned int refs;
                if (!iptc_get_references(&refs, chain, handle))
                        printf(" (ERROR obtaining refs)\n");
                else
                        printf(" (%u references)\n", refs);
        }

        if (format & FMT_LINENUMBERS)
                printf(FMT("%-4s ", "%s "), "num");
        if (!(format & FMT_NOCOUNTS)) {
                if (format & FMT_KILOMEGAGIGA) {
                        printf(FMT("%5s ","%s "), "pkts");
                        printf(FMT("%5s ","%s "), "bytes");
                } else {
                        printf(FMT("%8s ","%s "), "pkts");
                        printf(FMT("%10s ","%s "), "bytes");
                }
        }
        if (!(format & FMT_NOTARGET))
                printf(FMT("%-9s ","%s "), "target");
        fputs(" prot ", stdout);
        if (format & FMT_OPTIONS)
                fputs("opt", stdout);
        if (format & FMT_VIA) {
                printf(FMT(" %-6s ","%s "), "in");
                printf(FMT("%-6s ","%s "), "out");
        }
        printf(FMT(" %-19s ","%s "), "source");
        printf(FMT(" %-19s ","%s "), "destination");
        printf("\n");
}


static int
print_match(const struct xt_entry_match *m,
            const struct ipt_ip *ip,
            int numeric)
{
        const struct xtables_match *match =
                xtables_find_match(m->u.user.name, XTF_TRY_LOAD, NULL);

        if (match) {
                if (match->print)
```

```
                        match->print(ip, m, numeric);
                else
                        printf("%s ", match->name);
        } else {
                if (m->u.user.name[0])
                        printf("UNKNOWN match `%s' ", m->u.user.name);
        }
        /* Don't stop iterating. */
        return 0;
}

/* e is called `fw' here for historical reasons */
static void
print_firewall(const struct ipt_entry *fw,
               const char *targname,
               unsigned int num,
               unsigned int format,
               struct xtc_handle *const handle)
{
        const struct xtables_target *target = NULL;
        const struct xt_entry_target *t;
        uint8_t flags;
        char buf[BUFSIZ];

        if (!iptc_is_chain(targname, handle))
                target = xtables_find_target(targname, XTF_TRY_LOAD);
        else
                target = xtables_find_target(XT_STANDARD_TARGET,
                         XTF_LOAD_MUST_SUCCEED);

        t = ipt_get_target((struct ipt_entry *)fw);
        flags = fw->ip.flags;

        if (format & FMT_LINENUMBERS)
                printf(FMT("%-4u ", "%u "), num);

        if (!(format & FMT_NOCOUNTS)) {
                xtables_print_num(fw->counters.pcnt, format);
                xtables_print_num(fw->counters.bcnt, format);
        }

        if (!(format & FMT_NOTARGET))
                printf(FMT("%-9s ", "%s "), targname);

        fputc(fw->ip.invflags & XT_INV_PROTO ? '!' : ' ', stdout);
        {
                const char *pname = proto_to_name(fw->ip.proto, format&FMT_NUMERIC);
                if (pname)
                        printf(FMT("%-5s", "%s "), pname);
                else
                        printf(FMT("%-5hu", "%hu "), fw->ip.proto);
        }

        if (format & FMT_OPTIONS) {
                if (format & FMT_NOTABLE)
                        fputs("opt ", stdout);
                fputc(fw->ip.invflags & IPT_INV_FRAG ? '!' : '-', stdout);
                fputc(flags & IPT_F_FRAG ? 'f' : '-', stdout);
                fputc(' ', stdout);
        }

        if (format & FMT_VIA) {
                char iface[IFNAMSIZ+2];

                if (fw->ip.invflags & IPT_INV_VIA_IN) {
                        iface[0] = '!';
                        iface[1] = '\0';
                }
                else iface[0] = '\0';

                if (fw->ip.iniface[0] != '\0') {
                        strcat(iface, fw->ip.iniface);
                }
                else if (format & FMT_NUMERIC) strcat(iface, "*");
                else strcat(iface, "any");
                printf(FMT(" %-6s ","in %s "), iface);

                if (fw->ip.invflags & IPT_INV_VIA_OUT) {
                        iface[0] = '!';
                        iface[1] = '\0';
                }
                else iface[0] = '\0';

                if (fw->ip.outiface[0] != '\0') {
                        strcat(iface, fw->ip.outiface);
                }
                else if (format & FMT_NUMERIC) strcat(iface, "*");
                else strcat(iface, "any");
                printf(FMT("%-6s ","out %s "), iface);
        }

        fputc(fw->ip.invflags & IPT_INV_SRCIP ? '!' : ' ', stdout);
        if (fw->ip.smsk.s_addr == 0L && !(format & FMT_NUMERIC))
                printf(FMT("%-19s ","%s "), "anywhere");
        else {
                if (format & FMT_NUMERIC)
                        strcpy(buf, xtables_ipaddr_to_numeric(&fw->ip.src));
                else
```

```
                        strcpy(buf, xtables_ipaddr_to_anyname(&fw->ip.src));
                strcat(buf, xtables_ipmask_to_numeric(&fw->ip.smsk));
                printf(FMT("%-19s ","%s "), buf);
        }

        fputc(fw->ip.invflags & IPT_INV_DSTIP ? '!' : ' ', stdout);
        if (fw->ip.dmsk.s_addr == 0L && !(format & FMT_NUMERIC))
                printf(FMT("%-19s ","-> %s"), "anywhere");
        else {
                if (format & FMT_NUMERIC)
                        strcpy(buf, xtables_ipaddr_to_numeric(&fw->ip.dst));
                else
                        strcpy(buf, xtables_ipaddr_to_anyname(&fw->ip.dst));
                strcat(buf, xtables_ipmask_to_numeric(&fw->ip.dmsk));
                printf(FMT("%-19s ","-> %s"), buf);
        }

        if (format & FMT_NOTABLE)
                fputs("  ", stdout);

#ifdef IPT_F_GOTO
        if(fw->ip.flags & IPT_F_GOTO)
                printf("[goto] ");
#endif

        IPT_MATCH_ITERATE(fw, print_match, &fw->ip, format & FMT_NUMERIC);

        if (target) {
                if (target->print)
                        /* Print the target information. */
                        target->print(&fw->ip, t, format & FMT_NUMERIC);
        } else if (t->u.target_size != sizeof(*t))
                printf("[%u bytes of unknown target data] ",
                       (unsigned int)(t->u.target_size - sizeof(*t)));

        if (!(format & FMT_NONEWLINE))
                fputc('\n', stdout);
}

static void
print_firewall_line(const struct ipt_entry *fw,
                    struct xtc_handle *const h)
{
        struct xt_entry_target *t;

        t = ipt_get_target((struct ipt_entry *)fw);
        print_firewall(fw, t->u.user.name, 0, FMT_PRINT_RULE, h);
}

static int
append_entry(const xt_chainlabel chain,
             struct ipt_entry *fw,
             unsigned int nsaddrs,
             const struct in_addr saddrs[],
             const struct in_addr smasks[],
             unsigned int ndaddrs,
             const struct in_addr daddrs[],
             const struct in_addr dmasks[],
             int verbose,
             struct xtc_handle *handle)
{
        unsigned int i, j;
        int ret = 1;

        for (i = 0; i < nsaddrs; i++) {
                fw->ip.src.s_addr = saddrs[i].s_addr;
                fw->ip.smsk.s_addr = smasks[i].s_addr;
                for (j = 0; j < ndaddrs; j++) {
                        fw->ip.dst.s_addr = daddrs[j].s_addr;
                        fw->ip.dmsk.s_addr = dmasks[j].s_addr;
                        if (verbose)
                                print_firewall_line(fw, handle);
                        ret &= iptc_append_entry(chain, fw, handle);
                }
        }

        return ret;
}

static int
replace_entry(const xt_chainlabel chain,
              struct ipt_entry *fw,
              unsigned int rulenum,
              const struct in_addr *saddr, const struct in_addr *smask,
              const struct in_addr *daddr, const struct in_addr *dmask,
              int verbose,
              struct xtc_handle *handle)
{
        fw->ip.src.s_addr = saddr->s_addr;
        fw->ip.dst.s_addr = daddr->s_addr;
        fw->ip.smsk.s_addr = smask->s_addr;
        fw->ip.dmsk.s_addr = dmask->s_addr;

        if (verbose)
                print_firewall_line(fw, handle);
        return iptc_replace_entry(chain, fw, rulenum, handle);
}
```

```
static int
insert_entry(const xt_chainlabel chain,
             struct ipt_entry *fw,
             unsigned int rulenum,
             unsigned int nsaddrs,
             const struct in_addr saddrs[],
             const struct in_addr smasks[],
             unsigned int ndaddrs,
             const struct in_addr daddrs[],
             const struct in_addr dmasks[],
             int verbose,
             struct xtc_handle *handle)
{
        unsigned int i, j;
        int ret = 1;

        for (i = 0; i < nsaddrs; i++) {
                fw->ip.src.s_addr = saddrs[i].s_addr;
                fw->ip.smsk.s_addr = smasks[i].s_addr;
                for (j = 0; j < ndaddrs; j++) {
                        fw->ip.dst.s_addr = daddrs[j].s_addr;
                        fw->ip.dmsk.s_addr = dmasks[j].s_addr;
                        if (verbose)
                                print_firewall_line(fw, handle);
                        ret &= iptc_insert_entry(chain, fw, rulenum, handle);
                }
        }

        return ret;
}

static unsigned char *
make_delete_mask(const struct xtables_rule_match *matches,
                 const struct xtables_target *target)
{
        /* Establish mask for comparison */
        unsigned int size;
        const struct xtables_rule_match *matchp;
        unsigned char *mask, *mptr;

        size = sizeof(struct ipt_entry);
        for (matchp = matches; matchp; matchp = matchp->next)
                size += XT_ALIGN(sizeof(struct xt_entry_match)) + matchp->match->size;

        mask = xtables_calloc(1, size
                        + XT_ALIGN(sizeof(struct xt_entry_target))
                        + target->size);

        memset(mask, 0xFF, sizeof(struct ipt_entry));
        mptr = mask + sizeof(struct ipt_entry);

        for (matchp = matches; matchp; matchp = matchp->next) {
                memset(mptr, 0xFF,
                       XT_ALIGN(sizeof(struct xt_entry_match))
                       + matchp->match->userspacesize);
                mptr += XT_ALIGN(sizeof(struct xt_entry_match)) + matchp->match->size;
        }

        memset(mptr, 0xFF,
               XT_ALIGN(sizeof(struct xt_entry_target))
               + target->userspacesize);

        return mask;
}

static int
delete_entry(const xt_chainlabel chain,
             struct ipt_entry *fw,
             unsigned int nsaddrs,
             const struct in_addr saddrs[],
             const struct in_addr smasks[],
             unsigned int ndaddrs,
             const struct in_addr daddrs[],
             const struct in_addr dmasks[],
             int verbose,
             struct xtc_handle *handle,
             struct xtables_rule_match *matches,
             const struct xtables_target *target)
{
        unsigned int i, j;
        int ret = 1;
        unsigned char *mask;

        mask = make_delete_mask(matches, target);
        for (i = 0; i < nsaddrs; i++) {
                fw->ip.src.s_addr = saddrs[i].s_addr;
                fw->ip.smsk.s_addr = smasks[i].s_addr;
                for (j = 0; j < ndaddrs; j++) {
                        fw->ip.dst.s_addr = daddrs[j].s_addr;
                        fw->ip.dmsk.s_addr = dmasks[j].s_addr;
                        if (verbose)
                                print_firewall_line(fw, handle);
                        ret &= iptc_delete_entry(chain, fw, mask, handle);
                }
        }
        free(mask);

        return ret;
```

```
}

static int
check_entry(const xt_chainlabel chain, struct ipt_entry *fw,
            unsigned int nsaddrs, const struct in_addr *saddrs,
            const struct in_addr *smasks, unsigned int ndaddrs,
            const struct in_addr *daddrs, const struct in_addr *dmasks,
            bool verbose, struct xtc_handle *handle,
            struct xtables_rule_match *matches,
            const struct xtables_target *target)
{
        unsigned int i, j;
        int ret = 1;
        unsigned char *mask;

        mask = make_delete_mask(matches, target);
        for (i = 0; i < nsaddrs; i++) {
                fw->ip.src.s_addr = saddrs[i].s_addr;
                fw->ip.smsk.s_addr = smasks[i].s_addr;
                for (j = 0; j < ndaddrs; j++) {
                        fw->ip.dst.s_addr = daddrs[j].s_addr;
                        fw->ip.dmsk.s_addr = dmasks[j].s_addr;
                        if (verbose)
                                print_firewall_line(fw, handle);
                        ret &= iptc_check_entry(chain, fw, mask, handle);
                }
        }

        free(mask);
        return ret;
}

int
for_each_chain4(int (*fn)(const xt_chainlabel, int, struct xtc_handle *),
               int verbose, int builtinstoo, struct xtc_handle *handle)
{
        int ret = 1;
        const char *chain;
        char *chains;
        unsigned int i, chaincount = 0;

        chain = iptc_first_chain(handle);
        while (chain) {
                chaincount++;
                chain = iptc_next_chain(handle);
        }

        chains = xtables_malloc(sizeof(xt_chainlabel) * chaincount);
        i = 0;
        chain = iptc_first_chain(handle);
        while (chain) {
                strcpy(chains + i*sizeof(xt_chainlabel), chain);
                i++;
                chain = iptc_next_chain(handle);
        }

        for (i = 0; i < chaincount; i++) {
                if (!builtinstoo
                    && iptc_builtin(chains + i*sizeof(xt_chainlabel),
                                    handle) == 1)
                        continue;
                ret &= fn(chains + i*sizeof(xt_chainlabel), verbose, handle);
        }

        free(chains);
        return ret;
}

int
flush_entries4(const xt_chainlabel chain, int verbose,
               struct xtc_handle *handle)
{
        if (!chain)
                return for_each_chain4(flush_entries4, verbose, 1, handle);

        if (verbose)
                fprintf(stdout, "Flushing chain `%s'\n", chain);
        return iptc_flush_entries(chain, handle);
}

static int
zero_entries(const xt_chainlabel chain, int verbose,
             struct xtc_handle *handle)
{
        if (!chain)
                return for_each_chain4(zero_entries, verbose, 1, handle);

        if (verbose)
                fprintf(stdout, "Zeroing chain `%s'\n", chain);
        return iptc_zero_entries(chain, handle);
}

int
delete_chain4(const xt_chainlabel chain, int verbose,
              struct xtc_handle *handle)
{
        if (!chain)
                return for_each_chain4(delete_chain4, verbose, 0, handle);
```

```
        if (verbose)
                fprintf(stdout, "Deleting chain `%s'\n", chain);
        return iptc_delete_chain(chain, handle);
}

static int
list_entries(const xt_chainlabel chain, int rulenum, int verbose, int numeric,
             int expanded, int linenumbers, struct xtc_handle *handle)
{
        int found = 0;
        unsigned int format;
        const char *this;

        format = FMT_OPTIONS;
        if (!verbose)
                format |= FMT_NOCOUNTS;
        else
                format |= FMT_VIA;

        if (numeric)
                format |= FMT_NUMERIC;

        if (!expanded)
                format |= FMT_KILOMEGAGIGA;

        if (linenumbers)
                format |= FMT_LINENUMBERS;

        for (this = iptc_first_chain(handle);
             this;
             this = iptc_next_chain(handle)) {
                const struct ipt_entry *i;
                unsigned int num;

                if (chain && strcmp(chain, this) != 0)
                        continue;

                if (found) printf("\n");

                if (!rulenum)
                        print_header(format, this, handle);
                i = iptc_first_rule(this, handle);

                num = 0;
                while (i) {
                        num++;
                        if (!rulenum || num == rulenum)
                                print_firewall(i,
                                               iptc_get_target(i, handle),
                                               num,
                                               format,
                                               handle);
                        i = iptc_next_rule(i, handle);
                }
                found = 1;
        }

        errno = ENOENT;
        return found;
}

static void print_proto(uint16_t proto, int invert)
{
        if (proto) {
                unsigned int i;
                const char *invertstr = invert ? " !" : "";

                const struct protoent *pent = getprotobynumber(proto);
                if (pent) {
                        printf("%s -p %s", invertstr, pent->p_name);
                        return;
                }

                for (i = 0; xtables_chain_protos[i].name != NULL; ++i)
                        if (xtables_chain_protos[i].num == proto) {
                                printf("%s -p %s",
                                       invertstr, xtables_chain_protos[i].name);
                                return;
                        }

                printf("%s -p %u", invertstr, proto);
        }
}

#define IP_PARTS_NATIVE(n)                      \
(unsigned int)((n)>>24)&0xFF,                   \
(unsigned int)((n)>>16)&0xFF,                   \
(unsigned int)((n)>>8)&0xFF,                    \
(unsigned int)((n)&0xFF)

#define IP_PARTS(n) IP_PARTS_NATIVE(ntohl(n))

/* This assumes that mask is contiguous, and byte-bounded. */
static void
print_iface(char letter, const char *iface, const unsigned char *mask,
            int invert)
{
```

```
        unsigned int i;

        if (mask[0] == 0)
                return;

        printf("%s -%c ", invert ? " !" : "", letter);

        for (i = 0; i < IFNAMSIZ; i++) {
                if (mask[i] != 0) {
                        if (iface[i] != '\0')
                                printf("%c", iface[i]);
                } else {
                        /* we can access iface[i-1] here, because
                         * a few lines above we make sure that mask[0] != 0 */
                        if (iface[i-1] != '\0')
                                printf("+");
                        break;
                }
        }
}

static int print_match_save(const struct xt_entry_match *e,
                        const struct ipt_ip *ip)
{
        const struct xtables_match *match =
                xtables_find_match(e->u.user.name, XTF_TRY_LOAD, NULL);

        if (match) {
                printf(" -m %s",
                        match->alias ? match->alias(e) : e->u.user.name);

                /* some matches don't provide a save function */
                if (match->save)
                        match->save(ip, e);
        } else {
                if (e->u.match_size) {
                        fprintf(stderr,
                                "Can't find library for match `%s'\n",
                                e->u.user.name);
                        exit(1);
                }
        }
        return 0;
}

/* print a given ip including mask if neccessary */
static void print_ip(const char *prefix, uint32_t ip,
                     uint32_t mask, int invert)
{
        uint32_t bits, hmask = ntohl(mask);
        int i;

        if (!mask && !ip && !invert)
                return;

        printf("%s %s %u.%u.%u.%u",
                invert ? " !" : "",
                prefix,
                IP_PARTS(ip));

        if (mask == 0xFFFFFFFFU) {
                printf("/32");
                return;
        }

        i    = 32;
        bits = 0xFFFFFFFEU;
        while (--i >= 0 && hmask != bits)
                bits <<= 1;
        if (i >= 0)
                printf("/%u", i);
        else
                printf("/%u.%u.%u.%u", IP_PARTS(mask));
}

/* We want this to be readable, so only print out neccessary fields.
 * Because that's the kind of world I want to live in.  */
void print_rule4(const struct ipt_entry *e,
                struct xtc_handle *h, const char *chain, int counters)
{
        const struct xt_entry_target *t;
        const char *target_name;

        /* print counters for iptables-save */
        if (counters > 0)
                printf("[%llu:%llu] ", (unsigned long long)e->counters.pcnt, (unsigned long long)e->counters.bcnt);

        /* print chain name */
        printf("-A %s", chain);

        /* Print IP part. */
        print_ip("-s", e->ip.src.s_addr,e->ip.smsk.s_addr,
                        e->ip.invflags & IPT_INV_SRCIP);

        print_ip("-d", e->ip.dst.s_addr, e->ip.dmsk.s_addr,
                        e->ip.invflags & IPT_INV_DSTIP);

        print_iface('i', e->ip.iniface, e->ip.iniface_mask,
```

```
                e->ip.invflags & IPT_INV_VIA_IN);

        print_iface('o', e->ip.outiface, e->ip.outiface_mask,
                e->ip.invflags & IPT_INV_VIA_OUT);

        print_proto(e->ip.proto, e->ip.invflags & XT_INV_PROTO);

        if (e->ip.flags & IPT_F_FRAG)
                printf("%s -f",
                       e->ip.invflags & IPT_INV_FRAG ? " !" : "");

        /* Print matchinfo part */
        if (e->target_offset) {
                IPT_MATCH_ITERATE(e, print_match_save, &e->ip);
        }

        /* print counters for iptables -R */
        if (counters < 0)
                printf(" -c %llu %llu", (unsigned long long)e->counters.pcnt, (unsigned long long)e->counters.bcnt);

        /* Print target name and targinfo part */
        target_name = iptc_get_target(e, h);
        t = ipt_get_target((struct ipt_entry *)e);
        if (t->u.user.name[0]) {
                const struct xtables_target *target =
                        xtables_find_target(t->u.user.name, XTF_TRY_LOAD);

                if (!target) {
                        fprintf(stderr, "Can't find library for target `%s'\n",
                                t->u.user.name);
                        exit(1);
                }

                printf(" -j %s", target->alias ? target->alias(t) : target_name);
                if (target->save)
                        target->save(&e->ip, t);
                else {
                        /* If the target size is greater than xt_entry_target
                         * there is something to be saved, we just don't know
                         * how to print it */
                        if (t->u.target_size !=
                            sizeof(struct xt_entry_target)) {
                                fprintf(stderr, "Target `%s' is missing "
                                                "save function\n",
                                        t->u.user.name);
                                exit(1);
                        }
                }
        } else if (target_name && (*target_name != '\0'))
#ifdef IPT_F_GOTO
                printf(" -%c %s", e->ip.flags & IPT_F_GOTO ? 'g' : 'j', target_name);
#else
                printf(" -j %s", target_name);
#endif

        printf("\n");
}

static int
list_rules(const xt_chainlabel chain, int rulenum, int counters,
             struct xtc_handle *handle)
{
        const char *this = NULL;
        int found = 0;

        if (counters)
            counters = -1;              /* iptables -c format */

        /* Dump out chain names first,
         * thereby preventing dependency conflicts */
        if (!rulenum) for (this = iptc_first_chain(handle);
             this;
             this = iptc_next_chain(handle)) {
                if (chain && strcmp(this, chain) != 0)
                        continue;

                if (iptc_builtin(this, handle)) {
                        struct xt_counters count;
                        printf("-P %s %s", this, iptc_get_policy(this, &count, handle));
                        if (counters)
                            printf(" -c %llu %llu", (unsigned long long)count.pcnt, (unsigned long long)count.bcnt);
                        printf("\n");
                } else {
                        printf("-N %s\n", this);
                }
        }

        for (this = iptc_first_chain(handle);
             this;
             this = iptc_next_chain(handle)) {
                const struct ipt_entry *e;
                int num = 0;

                if (chain && strcmp(this, chain) != 0)
                        continue;

                /* Dump out rules */
                e = iptc_first_rule(this, handle);
```

```
                while(e) {
                        num++;
                        if (!rulenum || num == rulenum)
                            print_rule4(e, handle, this, counters);
                        e = iptc_next_rule(e, handle);
                }
                found = 1;
        }

        errno = ENOENT;
        return found;
}

static struct ipt_entry *
generate_entry(const struct ipt_entry *fw,
               struct xtables_rule_match *matches,
               struct xt_entry_target *target)
{
        unsigned int size;
        struct xtables_rule_match *matchp;
        struct ipt_entry *e;

        size = sizeof(struct ipt_entry);
        for (matchp = matches; matchp; matchp = matchp->next)
                size += matchp->match->m->u.match_size;

        e = xtables_malloc(size + target->u.target_size);
        *e = *fw;
        e->target_offset = size;
        e->next_offset = size + target->u.target_size;

        size = 0;
        for (matchp = matches; matchp; matchp = matchp->next) {
                memcpy(e->elems + size, matchp->match->m, matchp->match->m->u.match_size);
                size += matchp->match->m->u.match_size;
        }
        memcpy(e->elems + size, target, target->u.target_size);

        return e;
}

static void command_jump(struct iptables_command_state *cs)
{
        size_t size;

        set_option(&cs->options, OPT_JUMP, &cs->fw.ip.invflags, cs->invert);
        cs->jumpto = parse_target(optarg);
        /* TRY_LOAD (may be chain name) */
        cs->target = xtables_find_target(cs->jumpto, XTF_TRY_LOAD);

        if (cs->target == NULL)
                return;

        size = XT_ALIGN(sizeof(struct xt_entry_target))
                + cs->target->size;

        cs->target->t = xtables_calloc(1, size);
        cs->target->t->u.target_size = size;
        if (cs->target->real_name == NULL) {
                strcpy(cs->target->t->u.user.name, cs->jumpto);
        } else {
                /* Alias support for userspace side */
                strcpy(cs->target->t->u.user.name, cs->target->real_name);
                if (!(cs->target->ext_flags & XTABLES_EXT_ALIAS))
                        fprintf(stderr, "Notice: The %s target is converted into %s target "
                                "in rule listing and saving.\n",
                                cs->jumpto, cs->target->real_name);
        }
        cs->target->t->u.user.revision = cs->target->revision;

        xs_init_target(cs->target);

        if (cs->target->x6_options != NULL)
                opts = xtables_options_xfrm(iptables_globals.orig_opts, opts,
                                            cs->target->x6_options,
                                            &cs->target->option_offset);
        else
                opts = xtables_merge_options(iptables_globals.orig_opts, opts,
                                             cs->target->extra_opts,
                                             &cs->target->option_offset);
        if (opts == NULL)
                xtables_error(OTHER_PROBLEM, "can't alloc memory!");
}

static void command_match(struct iptables_command_state *cs)
{
        struct xtables_match *m;
        size_t size;

        if (cs->invert)
                xtables_error(PARAMETER_PROBLEM,
                           "unexpected ! flag before --match");

        m = xtables_find_match(optarg, XTF_LOAD_MUST_SUCCEED, &cs->matches);
        size = XT_ALIGN(sizeof(struct xt_entry_match)) + m->size;
        m->m = xtables_calloc(1, size);
        m->m->u.match_size = size;
        if (m->real_name == NULL) {
```

```
                strcpy(m->m->u.user.name, m->name);
        } else {
                strcpy(m->m->u.user.name, m->real_name);
                if (!(m->ext_flags & XTABLES_EXT_ALIAS))
                        fprintf(stderr, "Notice: the %s match is converted into %s match "
                                "in rule listing and saving.\n", m->name, m->real_name);
        }
        m->m->u.user.revision = m->revision;

        xs_init_match(m);
        if (m == m->next)
                return;
        /* Merge options for non-cloned matches */
        if (m->x6_options != NULL)
                opts = xtables_options_xfrm(iptables_globals.orig_opts, opts,
                                            m->x6_options, &m->option_offset);
        else if (m->extra_opts != NULL)
                opts = xtables_merge_options(iptables_globals.orig_opts, opts,
                                             m->extra_opts, &m->option_offset);
        if (opts == NULL)
                xtables_error(OTHER_PROBLEM, "can't alloc memory!");
}

int do_command4(int argc, char *argv[], char **table,
                struct xtc_handle **handle, bool restore)
{
        struct iptables_command_state cs;
        struct ipt_entry *e = NULL;
        unsigned int nsaddrs = 0, ndaddrs = 0;
        struct in_addr *saddrs = NULL, *smasks = NULL;
        struct in_addr *daddrs = NULL, *dmasks = NULL;

        int verbose = 0;
        bool wait = false;
        const char *chain = NULL;
        const char *shostnetworkmask = NULL, *dhostnetworkmask = NULL;
        const char *policy = NULL, *newname = NULL;
        unsigned int rulenum = 0, command = 0;
        const char *pcnt = NULL, *bcnt = NULL;
        int ret = 1;
        struct xtables_match *m;
        struct xtables_rule_match *matchp;
        struct xtables_target *t;
        unsigned long long cnt;

        memset(&cs, 0, sizeof(cs));
        cs.jumpto = "";
        cs.argv = argv;

        /* re-set optind to 0 in case do_command4 gets called
         * a second time */
        optind = 0;

        /* clear mflags in case do_command4 gets called a second time
         * (we clear the global list of all matches for security)*/
        for (m = xtables_matches; m; m = m->next)
                m->mflags = 0;

        for (t = xtables_targets; t; t = t->next) {
                t->tflags = 0;
                t->used = 0;
        }

        /* Suppress error messages: we may add new options if we
           demand-load a protocol. */
        opterr = 0;

        opts = xt_params->orig_opts;
        while ((cs.c = getopt_long(argc, argv,
           "-:A:C:D:R:I:L::S::M:F::Z::N:X::E:P:Vh::o:p:s:d:j:i:fbvwnt:m:xc:g:46",
                                   opts, NULL)) != -1) {
                switch (cs.c) {
                        /*
                         * Command selection
                         */
                case 'A':
                        add_command(&command, CMD_APPEND, CMD_NONE,
                                    cs.invert);
                        chain = optarg;
                        break;

                case 'C':
                        add_command(&command, CMD_CHECK, CMD_NONE,
                                    cs.invert);
                        chain = optarg;
                        break;

                case 'D':
                        add_command(&command, CMD_DELETE, CMD_NONE,
                                    cs.invert);
                        chain = optarg;
                        if (optind < argc && argv[optind][0] != '-'
                            && argv[optind][0] != '!') {
                                rulenum = parse_rulenumber(argv[optind++]);
                                command = CMD_DELETE_NUM;
                        }
                        break;
```

```
case 'R':
       add_command(&command, CMD_REPLACE, CMD_NONE,
                   cs.invert);
       chain = optarg;
       if (optind < argc && argv[optind][0] != '-'
           && argv[optind][0] != '!')
               rulenum = parse_rulenumber(argv[optind++]);
       else
               xtables_error(PARAMETER_PROBLEM,
                             "-%c requires a rule number",
                             cmd2char(CMD_REPLACE));
       break;

case 'I':
       add_command(&command, CMD_INSERT, CMD_NONE,
                   cs.invert);
       chain = optarg;
       if (optind < argc && argv[optind][0] != '-'
           && argv[optind][0] != '!')
               rulenum = parse_rulenumber(argv[optind++]);
       else rulenum = 1;
       break;

case 'L':
       add_command(&command, CMD_LIST,
                   CMD_ZERO | CMD_ZERO_NUM, cs.invert);
       if (optarg) chain = optarg;
       else if (optind < argc && argv[optind][0] != '-'
                && argv[optind][0] != '!')
               chain = argv[optind++];
       if (optind < argc && argv[optind][0] != '-'
           && argv[optind][0] != '!')
               rulenum = parse_rulenumber(argv[optind++]);
       break;

case 'S':
       add_command(&command, CMD_LIST_RULES,
                   CMD_ZERO|CMD_ZERO_NUM, cs.invert);
       if (optarg) chain = optarg;
       else if (optind < argc && argv[optind][0] != '-'
                && argv[optind][0] != '!')
               chain = argv[optind++];
       if (optind < argc && argv[optind][0] != '-'
           && argv[optind][0] != '!')
               rulenum = parse_rulenumber(argv[optind++]);
       break;

case 'F':
       add_command(&command, CMD_FLUSH, CMD_NONE,
                   cs.invert);
       if (optarg) chain = optarg;
       else if (optind < argc && argv[optind][0] != '-'
                && argv[optind][0] != '!')
               chain = argv[optind++];
       break;

case 'Z':
       add_command(&command, CMD_ZERO, CMD_LIST|CMD_LIST_RULES,
                   cs.invert);
       if (optarg) chain = optarg;
       else if (optind < argc && argv[optind][0] != '-'
               && argv[optind][0] != '!')
               chain = argv[optind++];
       if (optind < argc && argv[optind][0] != '-'
               && argv[optind][0] != '!') {
               rulenum = parse_rulenumber(argv[optind++]);
               command = CMD_ZERO_NUM;
       }
       break;

case 'N':
       parse_chain(optarg);
       add_command(&command, CMD_NEW_CHAIN, CMD_NONE,
                   cs.invert);
       chain = optarg;
       break;

case 'X':
       add_command(&command, CMD_DELETE_CHAIN, CMD_NONE,
                   cs.invert);
       if (optarg) chain = optarg;
       else if (optind < argc && argv[optind][0] != '-'
                && argv[optind][0] != '!')
               chain = argv[optind++];
       break;

case 'E':
       add_command(&command, CMD_RENAME_CHAIN, CMD_NONE,
                   cs.invert);
       chain = optarg;
       if (optind < argc && argv[optind][0] != '-'
           && argv[optind][0] != '!')
               newname = argv[optind++];
       else
               xtables_error(PARAMETER_PROBLEM,
                             "-%c requires old-chain-name and "
                             "new-chain-name",
                              cmd2char(CMD_RENAME_CHAIN));
```

```
                        break;

                case 'P':
                        add_command(&command, CMD_SET_POLICY, CMD_NONE,
                                    cs.invert);
                        chain = optarg;
                        if (optind < argc && argv[optind][0] != '-'
                            && argv[optind][0] != '!')
                                policy = argv[optind++];
                        else
                                xtables_error(PARAMETER_PROBLEM,
                                           "-%c requires a chain and a policy",
                                           cmd2char(CMD_SET_POLICY));
                        break;

                case 'h':
                        if (!optarg)
                                optarg = argv[optind];

                        /* iptables -p icmp -h */
                        if (!cs.matches && cs.protocol)
                                xtables_find_match(cs.protocol,
                                        XTF_TRY_LOAD, &cs.matches);

                        exit_printhelp(cs.matches);

                        /*
                         * Option selection
                         */
                case 'p':
                        set_option(&cs.options, OPT_PROTOCOL, &cs.fw.ip.invflags,
                                   cs.invert);

                        /* Canonicalize into lower case */
                        for (cs.protocol = optarg; *cs.protocol; cs.protocol++)
                                *cs.protocol = tolower(*cs.protocol);

                        cs.protocol = optarg;
                        cs.fw.ip.proto = xtables_parse_protocol(cs.protocol);

                        if (cs.fw.ip.proto == 0
                            && (cs.fw.ip.invflags & XT_INV_PROTO))
                                xtables_error(PARAMETER_PROBLEM,
                                           "rule would never match protocol");
                        break;

                case 's':
                        set_option(&cs.options, OPT_SOURCE, &cs.fw.ip.invflags,
                                   cs.invert);
                        shostnetworkmask = optarg;
                        break;

                case 'd':
                        set_option(&cs.options, OPT_DESTINATION, &cs.fw.ip.invflags,
                                   cs.invert);
                        dhostnetworkmask = optarg;
                        break;

#ifdef IPT_F_GOTO
                case 'g':
                        set_option(&cs.options, OPT_JUMP, &cs.fw.ip.invflags,
                                   cs.invert);
                        cs.fw.ip.flags |= IPT_F_GOTO;
                        cs.jumpto = parse_target(optarg);
                        break;
#endif

                case 'j':
                        command_jump(&cs);
                        break;


                case 'i':
                        if (*optarg == '\0')
                                xtables_error(PARAMETER_PROBLEM,
                                        "Empty interface is likely to be "
                                        "undesired");
                        set_option(&cs.options, OPT_VIANAMEIN, &cs.fw.ip.invflags,
                                   cs.invert);
                        xtables_parse_interface(optarg,
                                        cs.fw.ip.iniface,
                                        cs.fw.ip.iniface_mask);
                        break;

                case 'o':
                        if (*optarg == '\0')
                                xtables_error(PARAMETER_PROBLEM,
                                        "Empty interface is likely to be "
                                        "undesired");
                        set_option(&cs.options, OPT_VIANAMEOUT, &cs.fw.ip.invflags,
                                   cs.invert);
                        xtables_parse_interface(optarg,
                                        cs.fw.ip.outiface,
                                        cs.fw.ip.outiface_mask);
                        break;

                case 'f':
                        set_option(&cs.options, OPT_FRAGMENT, &cs.fw.ip.invflags,
```

```
                cs.invert);
        cs.fw.ip.flags |= IPT_F_FRAG;
        break;

case 'v':
        if (!verbose)
                set_option(&cs.options, OPT_VERBOSE,
                           &cs.fw.ip.invflags, cs.invert);
        verbose++;
        break;

case 'w':
        if (restore) {
                xtables_error(PARAMETER_PROBLEM,
                              "You cannot use `-w' from "
                              "iptables-restore");
        }
        wait = true;
        break;

case 'm':
        command_match(&cs);
        break;

case 'n':
        set_option(&cs.options, OPT_NUMERIC, &cs.fw.ip.invflags,
                cs.invert);
        break;

case 't':
        if (cs.invert)
                xtables_error(PARAMETER_PROBLEM,
                              "unexpected ! flag before --table");
        *table = optarg;
        break;

case 'x':
        set_option(&cs.options, OPT_EXPANDED, &cs.fw.ip.invflags,
                cs.invert);
        break;

case 'V':
        if (cs.invert)
                printf("Not %s ;-)\n", prog_vers);
        else
                printf("%s v%s\n",
                       prog_name, prog_vers);
        exit(0);

case '0':
        set_option(&cs.options, OPT_LINENUMBERS, &cs.fw.ip.invflags,
                cs.invert);
        break;

case 'M':
        xtables_modprobe_program = optarg;
        break;

case 'c':

        set_option(&cs.options, OPT_COUNTERS, &cs.fw.ip.invflags,
                cs.invert);
        pcnt = optarg;
        bcnt = strchr(pcnt + 1, ',');
        if (bcnt)
            bcnt++;
        if (!bcnt && optind < argc && argv[optind][0] != '-'
            && argv[optind][0] != '!')
                bcnt = argv[optind++];
        if (!bcnt)
                xtables_error(PARAMETER_PROBLEM,
                        "-%c requires packet and byte counter",
                        opt2char(OPT_COUNTERS));

        if (sscanf(pcnt, "%llu", &cnt) != 1)
                xtables_error(PARAMETER_PROBLEM,
                        "-%c packet counter not numeric",
                        opt2char(OPT_COUNTERS));
        cs.fw.counters.pcnt = cnt;

        if (sscanf(bcnt, "%llu", &cnt) != 1)
                xtables_error(PARAMETER_PROBLEM,
                        "-%c byte counter not numeric",
                        opt2char(OPT_COUNTERS));
        cs.fw.counters.bcnt = cnt;
        break;

case '4':
        /* This is indeed the IPv4 iptables */
        break;

case '6':
        /* This is not the IPv6 ip6tables */
        if (line != -1)
                return 1; /* success: line ignored */
        fprintf(stderr, "This is the IPv4 version of iptables.\n");
        exit_tryhelp(2);
```

```
        case 1: /* non option */
                if (optarg[0] == '!' && optarg[1] == '\0') {
                        if (cs.invert)
                                xtables_error(PARAMETER_PROBLEM,
                                              "multiple consecutive ! not"
                                              " allowed");
                        cs.invert = TRUE;
                        optarg[0] = '\0';
                        continue;
                }
                fprintf(stderr, "Bad argument `%s'\n", optarg);
                exit_tryhelp(2);

        default:
                if (command_default(&cs, &iptables_globals) == 1)
                        /* cf. ip6tables.c */
                        continue;
                break;
        }
        cs.invert = FALSE;
}

if (strcmp(*table, "nat") == 0 &&
    ((policy != NULL && strcmp(policy, "DROP") == 0) ||
    (cs.jumpto != NULL && strcmp(cs.jumpto, "DROP") == 0)))
        xtables_error(PARAMETER_PROBLEM,
                "\nThe \"nat\" table is not intended for filtering, "
                "the use of DROP is therefore inhibited.\n\n");

for (matchp = cs.matches; matchp; matchp = matchp->next)
        xtables_option_mfcall(matchp->match);
if (cs.target != NULL)
        xtables_option_tfcall(cs.target);

/* Fix me: must put inverse options checking here --MN */

if (optind < argc)
        xtables_error(PARAMETER_PROBLEM,
                   "unknown arguments found on commandline");
if (!command)
        xtables_error(PARAMETER_PROBLEM, "no command specified");
if (cs.invert)
        xtables_error(PARAMETER_PROBLEM,
                   "nothing appropriate following !");

if (command & (CMD_REPLACE | CMD_INSERT | CMD_DELETE | CMD_APPEND | CMD_CHECK)) {
        if (!(cs.options & OPT_DESTINATION))
                dhostnetworkmask = "0.0.0.0/0";
        if (!(cs.options & OPT_SOURCE))
                shostnetworkmask = "0.0.0.0/0";
}

if (shostnetworkmask)
        xtables_ipparse_multiple(shostnetworkmask, &saddrs,
                                 &smasks, &nsaddrs);

if (dhostnetworkmask)
        xtables_ipparse_multiple(dhostnetworkmask, &daddrs,
                                 &dmasks, &ndaddrs);

if ((nsaddrs > 1 || ndaddrs > 1) &&
    (cs.fw.ip.invflags & (IPT_INV_SRCIP | IPT_INV_DSTIP)))
        xtables_error(PARAMETER_PROBLEM, "! not allowed with multiple"
                   " source or destination IP addresses");

if (command == CMD_REPLACE && (nsaddrs != 1 || ndaddrs != 1))
        xtables_error(PARAMETER_PROBLEM, "Replacement rule does not "
                   "specify a unique address");

generic_opt_check(command, cs.options);

/* Attempt to acquire the xtables lock */
if (!restore && !xtables_lock(wait)) {
        fprintf(stderr, "Another app is currently holding the xtables lock. "
                "Perhaps you want to use the -w option?\n");
        xtables_free_opts(1);
        exit(RESOURCE_PROBLEM);
}

/* only allocate handle if we weren't called with a handle */
if (!*handle)
        *handle = iptc_init(*table);

/* try to insmod the module if iptc_init failed */
if (!*handle && xtables_load_ko(xtables_modprobe_program, false) != -1)
        *handle = iptc_init(*table);

if (!*handle)
        xtables_error(VERSION_PROBLEM,
                   "can't initialize iptables table `%s': %s",
                   *table, iptc_strerror(errno));

if (command == CMD_APPEND
    || command == CMD_DELETE
    || command == CMD_CHECK
    || command == CMD_INSERT
    || command == CMD_REPLACE) {
        if (strcmp(chain, "PREROUTING") == 0
```

```
                || strcmp(chain, "INPUT") == 0) {
                    /* -o not valid with incoming packets. */
                    if (cs.options & OPT_VIANAMEOUT)
                            xtables_error(PARAMETER_PROBLEM,
                                       "Can't use -%c with %s\n",
                                       opt2char(OPT_VIANAMEOUT),
                                       chain);
            }

            if (strcmp(chain, "POSTROUTING") == 0
                || strcmp(chain, "OUTPUT") == 0) {
                    /* -i not valid with outgoing packets */
                    if (cs.options & OPT_VIANAMEIN)
                            xtables_error(PARAMETER_PROBLEM,
                                       "Can't use -%c with %s\n",
                                       opt2char(OPT_VIANAMEIN),
                                       chain);
            }

            if (cs.target && iptc_is_chain(cs.jumpto, *handle)) {
                    fprintf(stderr,
                            "Warning: using chain %s, not extension\n",
                            cs.jumpto);

                    if (cs.target->t)
                            free(cs.target->t);

                    cs.target = NULL;
            }

            /* If they didn't specify a target, or it's a chain
               name, use standard. */
            if (!cs.target
                && (strlen(cs.jumpto) == 0
                    || iptc_is_chain(cs.jumpto, *handle))) {
                    size_t size;

                    cs.target = xtables_find_target(XT_STANDARD_TARGET,
                                    XTF_LOAD_MUST_SUCCEED);

                    size = sizeof(struct xt_entry_target)
                            + cs.target->size;
                    cs.target->t = xtables_calloc(1, size);
                    cs.target->t->u.target_size = size;
                    strcpy(cs.target->t->u.user.name, cs.jumpto);
                    if (!iptc_is_chain(cs.jumpto, *handle))
                            cs.target->t->u.user.revision = cs.target->revision;
                    xs_init_target(cs.target);
            }

            if (!cs.target) {
                    /* it is no chain, and we can't load a plugin.
                     * We cannot know if the plugin is corrupt, non
                     * existant OR if the user just misspelled a
                     * chain. */
#ifdef IPT_F_GOTO
                    if (cs.fw.ip.flags & IPT_F_GOTO)
                            xtables_error(PARAMETER_PROBLEM,
                                       "goto '%s' is not a chain\n",
                                       cs.jumpto);
#endif
                    xtables_find_target(cs.jumpto, XTF_LOAD_MUST_SUCCEED);
            } else {
                    e = generate_entry(&cs.fw, cs.matches, cs.target->t);
                    free(cs.target->t);
            }
    }

    switch (command) {
    case CMD_APPEND:
            ret = append_entry(chain, e,
                               nsaddrs, saddrs, smasks,
                               ndaddrs, daddrs, dmasks,
                               cs.options&OPT_VERBOSE,
                               *handle);
            break;
    case CMD_DELETE:
            ret = delete_entry(chain, e,
                               nsaddrs, saddrs, smasks,
                               ndaddrs, daddrs, dmasks,
                               cs.options&OPT_VERBOSE,
                               *handle, cs.matches, cs.target);
            break;
    case CMD_DELETE_NUM:
            ret = iptc_delete_num_entry(chain, rulenum - 1, *handle);
            break;
    case CMD_CHECK:
            ret = check_entry(chain, e,
                               nsaddrs, saddrs, smasks,
                               ndaddrs, daddrs, dmasks,
                               cs.options&OPT_VERBOSE,
                               *handle, cs.matches, cs.target);
            break;
    case CMD_REPLACE:
            ret = replace_entry(chain, e, rulenum - 1,
                                saddrs, smasks, daddrs, dmasks,
                                cs.options&OPT_VERBOSE, *handle);
            break;
```

```
        case CMD_INSERT:
                ret = insert_entry(chain, e, rulenum - 1,
                                   nsaddrs, saddrs, smasks,
                                   ndaddrs, daddrs, dmasks,
                                   cs.options&OPT_VERBOSE,
                                   *handle);
                break;
        case CMD_FLUSH:
                ret = flush_entries4(chain, cs.options&OPT_VERBOSE, *handle);
                break;
        case CMD_ZERO:
                ret = zero_entries(chain, cs.options&OPT_VERBOSE, *handle);
                break;
        case CMD_ZERO_NUM:
                ret = iptc_zero_counter(chain, rulenum, *handle);
                break;
        case CMD_LIST:
        case CMD_LIST|CMD_ZERO:
        case CMD_LIST|CMD_ZERO_NUM:
                ret = list_entries(chain,
                                   rulenum,
                                   cs.options&OPT_VERBOSE,
                                   cs.options&OPT_NUMERIC,
                                   cs.options&OPT_EXPANDED,
                                   cs.options&OPT_LINENUMBERS,
                                   *handle);
                if (ret && (command & CMD_ZERO))
                        ret = zero_entries(chain,
                                           cs.options&OPT_VERBOSE, *handle);
                if (ret && (command & CMD_ZERO_NUM))
                        ret = iptc_zero_counter(chain, rulenum, *handle);
                break;
        case CMD_LIST_RULES:
        case CMD_LIST_RULES|CMD_ZERO:
        case CMD_LIST_RULES|CMD_ZERO_NUM:
                ret = list_rules(chain,
                                 rulenum,
                                 cs.options&OPT_VERBOSE,
                                 *handle);
                if (ret && (command & CMD_ZERO))
                        ret = zero_entries(chain,
                                           cs.options&OPT_VERBOSE, *handle);
                if (ret && (command & CMD_ZERO_NUM))
                        ret = iptc_zero_counter(chain, rulenum, *handle);
                break;
        case CMD_NEW_CHAIN:
                ret = iptc_create_chain(chain, *handle);
                break;
        case CMD_DELETE_CHAIN:
                ret = delete_chain4(chain, cs.options&OPT_VERBOSE, *handle);
                break;
        case CMD_RENAME_CHAIN:
                ret = iptc_rename_chain(chain, newname, *handle);
                break;
        case CMD_SET_POLICY:
                ret = iptc_set_policy(chain, policy, cs.options&OPT_COUNTERS ? &cs.fw.counters : NULL, *handle);
                break;
        default:
                /* We should never reach this... */
                exit_tryhelp(2);
        }

        if (verbose > 1)
                dump_entries(*handle);

        xtables_rule_matches_free(&cs.matches);

        if (e != NULL) {
                free(e);
                e = NULL;
        }

        free(saddrs);
        free(smasks);
        free(daddrs);
        free(dmasks);
        xtables_free_opts(1);

        return ret;
}
```

Notice for package(s)

libnfsidmap

Notice for package(s)

nfs-utils

---

a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code. (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

```
  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.  SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```

Notice for package(s)

tipcutils

```
/*
 * tipc-config.c: TIPC configuration management tool
 *
 * Copyright (c) 2004-2005, Ericsson Research Canada
 * Copyright (c) 2004-2006, Ericsson AB
```

```
 * Copyright (c) 2005-2008,2010-2011, Wind River Systems
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the names of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <getopt.h>
#include <unistd.h>
#include <poll.h>
#include <string.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <linux/tipc.h>
#include <linux/tipc_config.h>
#include <linux/genetlink.h>
#include <linux/version.h>
#include <ifaddrs.h>
#include <netdb.h>

#include "config.h"

/* typedefs */

typedef void (*VOIDFUNCPTR) ();

/* constants */
#define MEDIA_NAME_UDP "udp"
#define MAX_COMMANDS 8
#define MAX_TLVS_SPACE 33000            /* must be a multiple of 4 bytes */
#define ADDR_AREA 30
#define REPLY_LEN 256
#define ARGS_SIZE 128

/* local variables */

static int verbose = 0;
static int interactive = 0;
static __u32 dest = 0;
static __u32 tlv_area[MAX_TLVS_SPACE / sizeof(__u32)];
static __u32 tlv_list_area[MAX_TLVS_SPACE / sizeof(__u32)];

/* forward declarations */

static char usage[];

/* macros */

#define cprintf(fmt, arg...)    do { if (verbose) printf(fmt, ##arg); } while (0)

#define fatal(fmt, arg...)      do { printf(fmt, ##arg); exit(EXIT_FAILURE); } while (0)

#define confirm(fmt, arg...) do { \
                char c; \
                int ret; \
                if (interactive) { \
                        printf(fmt, ##arg); \
                        ret = scanf(" %1c", &c); /* leading blank skips whitespace */ \
                        if ((ret != 1) || ((c != '\n') && (c != 'Y') && (c != 'y'))) { \
                                printf("Exiting...\n"); \
                                exit(EXIT_SUCCESS); \
                        } \
                } \
        } while (0)

/* local variables */

static char *err_string[] = {
        "incorrect message format",
        "must be network administrator to perform operation",
        "must be zone master to perform operation",
```

```
        "remote management not enabled on destination node",
        "operation not supported",
        "invalid argument"
};

/*****************************************************************************
 *
 * Utility routines used in executing command options
 *
 */

static inline int delimit(int val, int min, int max)
{
        if (val > max)
                return max;
        if (val < min)
                return min;
        return val;
}

static __u32 own_node(void)
{
        struct sockaddr_tipc addr;
        socklen_t sz = sizeof(addr);
        int sd;

        sd = socket(AF_TIPC, SOCK_RDM, 0);
        if (sd < 0)
                fatal("TIPC module not installed\n");
        if (getsockname(sd, (struct sockaddr *)&addr, &sz) < 0)
                fatal("failed to get TIPC socket address\n");
        close(sd);
        return addr.addr.id.node;
}

static const char *addr2str(__u32 addr)
{
        static char addr_area[4][16];   /* allow up to 4 uses in one printf() */
        static int addr_crs = 0;

        addr_crs = (addr_crs + 1) & 3;
        sprintf(&addr_area[addr_crs][0], "<%u.%u.%u>",
                tipc_zone(addr), tipc_cluster(addr), tipc_node(addr));
        return &addr_area[addr_crs][0];
}

static const char *for_dest(void)
{
        static char addr_area[ADDR_AREA];

        if (dest == own_node())
                return "";
        sprintf(addr_area, " for node %s", addr2str(dest));
        return addr_area;
}

static const char *for_domain(const char *string, __u32 domain)
{
        static char addr_area[ADDR_AREA];

        if (domain == 0)
                return "";
        sprintf(addr_area, "%s%s", string, addr2str(domain));
        return addr_area;
}

static void print_title(const char *main_title, const char *extra_title)
{
        printf(main_title, for_dest(), extra_title);
}

static void print_title_opt(const char *main_title, const char *extra_title)
{
        if ((dest == own_node()) && (extra_title[0] == '\0'))
                return;

        printf(main_title, for_dest(), extra_title);
}

char *get_arg(char **args)
{
        char *ret;
        char *comma;

        ret = *args;
        comma = strchr(ret, ',');
        if (comma) {
                *comma = '\0';
                *args = comma + 1;
        } else
                *args = NULL;
        return ret;
}

static __u32 str2addr(char *str)
{
        uint z, c, n;
        char dummy;
```

```
        if (sscanf(str, "%u.%u.%u%c", &z, &c, &n, &dummy) != 3)
                fatal("invalid network address, use syntax: Z.C.N\n");
        if ((z != delimit(z, 0, 255)) ||
                        (c != delimit(c, 0, 4095)) ||
                        (n != delimit(n, 0, 4095)))
                fatal("network address field value(s) too large\n");
        return tipc_addr(z, c, n);
}


/* ************************************************************************
 *
 * Routines used to exchange messages over Netlink sockets
 *
 */

#define NLA_SIZE(type)  (NLA_HDRLEN + NLA_ALIGN(sizeof(type)))

#define nla_for_each_attr(pos, head, len, rem) \
        for (pos = head, rem = len; nla_ok(pos, rem); pos = nla_next(pos, &(rem)))

static inline void *nla_data(const struct nlattr *nla)
{
        return (char *) nla + NLA_HDRLEN;
}

static inline int nla_ok(const struct nlattr *nla, int remaining)
{
        return remaining >= sizeof(*nla) &&
               nla->nla_len >= sizeof(*nla) &&
               nla->nla_len <= remaining;
}

static inline struct nlattr *nla_next(const struct nlattr *nla, int *remaining) {
        int totlen = NLA_ALIGN(nla->nla_len);

        *remaining -= totlen;
        return (struct nlattr *) ((char *) nla + totlen);
}

static inline int nla_put_string(struct nlattr *nla, int type, const char *str)
{
        int attrlen = strlen(str) + 1;

        nla->nla_len = NLA_HDRLEN + attrlen;
        nla->nla_type = type;
        memcpy(nla_data(nla), str, attrlen);

        return NLA_HDRLEN + NLA_ALIGN(attrlen);
}

static inline __u16 nla_get_u16(struct nlattr *nla)
{
        return *(__u16 *) nla_data(nla);
}

static int write_uninterrupted(int sk, const char *buf, int len)
{
        int c;

        while ((c = write(sk, buf, len)) < len) {
                if (c == -1) {
                        if (errno == EINTR)
                                continue;
                        return -1;
                }

                buf += c;
                len -= c;
        }

        return 0;
}

static int genetlink_call(__u16 family_id, __u8 cmd, void *header,
                          size_t header_len, void *request, size_t request_len,
                          void *reply, size_t reply_len)
{
        struct msg {
                struct nlmsghdr n;
                struct genlmsghdr g;
                char payload[0];
        };

        struct msg *request_msg;
        struct msg *reply_msg;
        int request_msg_size;
        int reply_msg_size;

        struct sockaddr_nl local;
        struct pollfd pfd;
        int sndbuf = 32*1024; /* 32k */
        int rcvbuf = 32*1024; /* 32k */
        int len;
        int sk;

        /*
```

```
         * Prepare request/reply messages
         */
        request_msg_size = NLMSG_LENGTH(GENL_HDRLEN + header_len + request_len);
        request_msg = malloc(request_msg_size);
        request_msg->n.nlmsg_len = request_msg_size;
        request_msg->n.nlmsg_type = family_id;
        request_msg->n.nlmsg_flags = NLM_F_REQUEST;
        request_msg->n.nlmsg_seq = 0;
        request_msg->n.nlmsg_pid = getpid();
        request_msg->g.cmd = cmd;
        request_msg->g.version = 0;
        if (header_len)
                memcpy(&request_msg->payload[0], header, header_len);
        if (request_len)
                memcpy(&request_msg->payload[header_len], request, request_len);

        reply_msg_size = NLMSG_LENGTH(GENL_HDRLEN + header_len + reply_len);
        reply_msg = malloc(reply_msg_size);

        /*
         * Create socket
         */
        memset(&local, 0, sizeof(local));
        local.nl_family = AF_NETLINK;

        if ((sk = socket(AF_NETLINK, SOCK_DGRAM, NETLINK_GENERIC)) == -1)
                fatal("error creating Netlink socket\n");

        if ((bind(sk, (struct sockaddr*)&local, sizeof(local)) == -1) ||
                        (setsockopt(sk, SOL_SOCKET, SO_SNDBUF, &sndbuf, sizeof(sndbuf)) == -1) ||
                        (setsockopt(sk, SOL_SOCKET, SO_RCVBUF, &rcvbuf, sizeof(rcvbuf)) == -1)) {
                fatal("error creating Netlink socket\n");
        }

        /*
         * Send request
         */
        if (write_uninterrupted(sk, (char*)request_msg, request_msg_size) < 0)
                fatal("error sending message via Netlink\n");

        /*
         * Wait for reply
         */
        pfd.fd = sk;
        pfd.events = ~POLLOUT;
        if ((poll(&pfd, 1, 3000) != 1) || !(pfd.revents & POLLIN))
                fatal("no reply detected from Netlink\n");

        /*
         * Read reply
         */
        len = recv(sk, (char*)reply_msg, reply_msg_size, 0);
        if (len < 0)
                fatal("error receiving reply message via Netlink\n");

        close(sk);

        /*
         * Validate response
         */
        if (!NLMSG_OK(&reply_msg->n, len))
                fatal("invalid reply message received via Netlink\n");

        if (reply_msg->n.nlmsg_type == NLMSG_ERROR) {
                len = -1;
                goto out;
        }

        if ((request_msg->n.nlmsg_type != reply_msg->n.nlmsg_type) ||
                        (request_msg->n.nlmsg_seq != reply_msg->n.nlmsg_seq))
                fatal("unexpected message received via Netlink\n");

        /*
         * Copy reply header
         */
        len -= NLMSG_LENGTH(GENL_HDRLEN);
        if (len < header_len)
                fatal("too small reply message received via Netlink\n");
        if (header_len > 0)
                memcpy(header, &reply_msg->payload[0], header_len);

        /*
         * Copy reply payload
         */
        len -= header_len;
        if (len > reply_len)
                fatal("reply message too large to copy\n");
        if (len > 0)
                memcpy(reply, &reply_msg->payload[header_len], len);

out:
        free(request_msg);
        free(reply_msg);

        return len;
}

static int get_genl_family_id(const char* name)
```

```
{
        struct nlattr_family_name {
                char value[GENL_NAMSIZ];
        };

        struct nlattr_family_id {
                __u16 value;
        };

        /*
         * Create request/reply buffers
         *
         * Note that the reply buffer is larger than necessary in case future
         * versions of Netlink return additional protocol family attributes
         */
        char request[NLA_SIZE(struct nlattr_family_name)];
        int request_len = nla_put_string((struct nlattr *)request, CTRL_ATTR_FAMILY_NAME, name);

        char reply[REPLY_LEN];

        /*
         * Call control service
         */
        int len = genetlink_call(GENL_ID_CTRL, CTRL_CMD_GETFAMILY,
                                 0, 0,
                                 request, request_len,
                                 reply, sizeof(reply));

        if (len == -1)
                return -1;

        /*
         * Parse reply
         */
        struct nlattr *head = (struct nlattr *) reply;
        struct nlattr *nla;
        int rem;

        nla_for_each_attr(nla, head, len, rem) {
                if (nla->nla_type == CTRL_ATTR_FAMILY_ID)
                        return nla_get_u16(nla);
        }

        if (rem > 0)
                fatal("%d bytes leftover after parsing Netlink attributes\n", rem);

        return -1;
}

static int do_command_netlink(__u16 cmd, void *req_tlv, __u32 req_tlv_space,
                              void *rep_tlv, __u32 rep_tlv_space)
{
        struct tipc_genlmsghdr header;
        int family_id;
        int len;

        /*
         * Request header
         */
        header.dest = dest;
        header.cmd = cmd;

        /*
         * Get TIPC family id
         */
        if ((family_id = get_genl_family_id(TIPC_GENL_NAME)) == -1)
                fatal("no Netlink service registered for %s\n", TIPC_GENL_NAME);

        /*
         * Call control service
         */
        len = genetlink_call(family_id, TIPC_GENL_CMD,
                             &header, sizeof(header),
                             req_tlv, req_tlv_space,
                             rep_tlv, rep_tlv_space);

        return len;
}

/*****************************************************************************
 *
 * Routines used to exchange messages over TIPC sockets
 *
 */

static int do_command_tipc(__u16 cmd, void *req_tlv, __u32 req_tlv_space,
                           void *rep_tlv, __u32 rep_tlv_space)
{
        struct {
                struct tipc_cfg_msg_hdr hdr;
                char buf[MAX_TLVS_SPACE];
        } req, ans;
        int msg_space;
        int tsd;
        struct sockaddr_tipc tipc_dest;
        int imp = TIPC_CRITICAL_IMPORTANCE;
        struct pollfd pfd;
        int pollres;
```

```
        if ((tsd = socket(AF_TIPC, SOCK_RDM, 0)) < 0)
                fatal("TIPC module not installed\n");

        msg_space = TCM_SET(&req.hdr, cmd, TCM_F_REQUEST,
                            req_tlv, req_tlv_space);

        setsockopt(tsd, SOL_TIPC, TIPC_IMPORTANCE, &imp, sizeof(imp));

        tipc_dest.family = AF_TIPC;
        tipc_dest.addrtype = TIPC_ADDR_NAME;
        tipc_dest.addr.name.name.type = TIPC_CFG_SRV;
        tipc_dest.addr.name.name.instance = dest;
        tipc_dest.addr.name.domain = dest;

        if (sendto(tsd, &req, msg_space, 0,
                        (struct sockaddr *)&tipc_dest, sizeof(tipc_dest)) < 0)
                fatal("unable to send command to node %s\n", addr2str(dest));

        /* Wait for response message */

        pfd.events = 0xffff & ~POLLOUT;
        pfd.fd = tsd;
        pollres = poll(&pfd, 1, 3000);
        if ((pollres < 0) || !(pfd.revents & POLLIN))
                fatal("no reply detected from TIPC\n");
        msg_space = recv(tsd, &ans, sizeof(ans), 0);
        if (msg_space < 0)
                fatal("error receiving reply message via TIPC\n");

        /* Validate response message */

        if ((msg_space < TCM_SPACE(0)) || (ntohl(ans.hdr.tcm_len) > msg_space))
                fatal("invalid reply message received via TIPC\n");
        if ((ntohs(ans.hdr.tcm_type) != cmd) ||
                        (ntohs(ans.hdr.tcm_flags) != 0))
                fatal("unexpected message received via TIPC\n");

        msg_space = ntohl(ans.hdr.tcm_len) - TCM_SPACE(0);
        if (msg_space > rep_tlv_space)
                fatal("reply message too large to copy\n");
        memcpy(rep_tlv, ans.buf, msg_space);
        return msg_space;
}


/*******************************************************************************
 *
 * Routines used to process commands requested by user
 *
 */

static __u32 do_command(__u16 cmd, void *req_tlv, __u32 req_tlv_space,
                        void *rep_tlv, __u32 rep_tlv_space)
{
        int rep_len;

        if (dest == own_node())
                rep_len = do_command_netlink(cmd, req_tlv, req_tlv_space,
                                             rep_tlv, rep_tlv_space);
        else
                rep_len = do_command_tipc(cmd, req_tlv, req_tlv_space,
                                          rep_tlv, rep_tlv_space);

        if (TLV_CHECK(rep_tlv, rep_len, TIPC_TLV_ERROR_STRING)) {
                char *c = (char *)TLV_DATA(rep_tlv);
                char code = *c;
                char max_code = sizeof(err_string)/sizeof(err_string[0]);

                if (code & 0x80) {
                        code &= 0x7F;
                        printf("%s",(code < max_code) ? err_string[(int)code]
                                : "unknown error");
                        c++;
                }
                fatal("%s\n", c);
        }

        return rep_len;
}

static __u32 do_get_unsigned(__u16 cmd)
{
        int tlv_space;
        __u32 value;

        tlv_space = do_command(cmd, NULL, 0, tlv_area, sizeof(tlv_area));

        if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_UNSIGNED))
                fatal("corrupted reply message\n");

        value = *(__u32 *)TLV_DATA(tlv_area);
        return ntohl(value);
}

static void do_set_unsigned(char *args, __u16 cmd, char *attr_name,
                            char *attr_warn)
{
```

```
        __u32 attr_val;
        __u32 attr_val_net;
        int tlv_space;
        char dummy;

        if (sscanf(args, "%u%c", &attr_val, &dummy) != 1)
                fatal("invalid numeric argument for %s\n", attr_name);

        confirm("set %s to %u%s?%s [Y/n]\n", attr_name, attr_val,
                for_dest(), attr_warn);

        attr_val_net = htonl(attr_val);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_UNSIGNED,
                            &attr_val_net, sizeof(attr_val_net));
        do_command(cmd, tlv_area, tlv_space, tlv_area, sizeof(tlv_area));

        cprintf("%s%s now set to %u\n", attr_name, for_dest(), attr_val);
}

static void set_node_addr(char *args)
{
        __u32 new_addr;
        __u32 new_addr_net;
        int tlv_space;

        if (!*args) {
                do_command(TIPC_CMD_NOOP, NULL, 0, tlv_area, sizeof(tlv_area));
                printf("node address: %s\n", addr2str(dest));
                return;
        }

        new_addr = str2addr(args);

        confirm("change node address%s to %s? "
                "(this will delete all links) [Y/n]\n",
                for_dest(), addr2str(new_addr));

        new_addr_net = htonl(new_addr);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_NET_ADDR,
                            &new_addr_net, sizeof(new_addr_net));
        do_command(TIPC_CMD_SET_NODE_ADDR, tlv_area, tlv_space,
                   tlv_area, sizeof(tlv_area));

        cprintf("node address%s now set to %s\n",
                for_dest(), addr2str(new_addr));
        dest = new_addr;
}

static void set_remote_mng(char *args)
{
        __u32 attr_val;
        __u32 attr_val_net;
        int tlv_space;

        if (!*args) {
                printf("remote management%s: %s\n", for_dest(),
                       do_get_unsigned(TIPC_CMD_GET_REMOTE_MNG) ?
                       "enabled" : "disabled");
                return;
        }

        if (!strcmp(args, "enable"))
                attr_val = 1;
        else if (!strcmp(args, "disable"))
                attr_val = 0;
        else
                fatal("invalid argument for remote management\n");

        confirm("%s remote management%s? [Y/n]\n",
                attr_val ? "enable" : "disable", for_dest());

        attr_val_net = htonl(attr_val);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_UNSIGNED,
                            &attr_val_net, sizeof(attr_val_net));
        do_command(TIPC_CMD_SET_REMOTE_MNG, tlv_area, tlv_space,
                   tlv_area, sizeof(tlv_area));

        cprintf("remote management%s %s\n", for_dest(),
                attr_val ? "enabled" : "disabled");
}

static void set_max_ports(char *args)
{
        if (!*args)
                printf("maximum allowed ports%s: %u\n", for_dest(),
                       do_get_unsigned(TIPC_CMD_GET_MAX_PORTS));
        else
                do_set_unsigned(args, TIPC_CMD_SET_MAX_PORTS,
                                "max ports", "");
}

static void set_max_publ(char *args)
{
        if (!*args)
                printf("maximum allowed publications%s: %u\n", for_dest(),
                       do_get_unsigned(TIPC_CMD_GET_MAX_PUBL));
        else
                do_set_unsigned(args, TIPC_CMD_SET_MAX_PUBL,
```

```
                                "max publications", "");
}

static void set_max_subscr(char *args)
{
        if (!*args)
                printf("maximum allowed subscriptions%s: %u\n", for_dest(),
                       do_get_unsigned(TIPC_CMD_GET_MAX_SUBSCR));
        else
                do_set_unsigned(args, TIPC_CMD_SET_MAX_SUBSCR,
                                "max subscriptions", "");
}

static void set_netid(char *args)
{
        if (!*args)
                printf("current network id%s: %u\n", for_dest(),
                       do_get_unsigned(TIPC_CMD_GET_NETID));
        else
                do_set_unsigned(args, TIPC_CMD_SET_NETID,
                                "network identity", "");
}

static void get_nodes(char *args)
{
        int tlv_space;
        __u32 domain;
        __u32 domain_net;
        struct tlv_list_desc tlv_list;
        struct tipc_node_info *node_info;

        domain = (*args != 0) ? str2addr(args) : 0;
        domain_net = htonl(domain);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_NET_ADDR,
                            &domain_net, sizeof(domain_net));
        tlv_space = do_command(TIPC_CMD_GET_NODES, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        print_title("Neighbors%s%s:\n", for_domain(" within domain ", domain));
        if (!tlv_space) {
                printf("No nodes found\n");
                return;
        }

        TLV_LIST_INIT(&tlv_list, tlv_area, tlv_space);
        while (!TLV_LIST_EMPTY(&tlv_list)) {
                if (!TLV_LIST_CHECK(&tlv_list, TIPC_TLV_NODE_INFO))
                        fatal("corrupted reply message\n");
                node_info = (struct tipc_node_info *)TLV_LIST_DATA(&tlv_list);
                printf("%s: %s\n", addr2str(ntohl(node_info->addr)),
                       ntohl(node_info->up) ? "up" : "down");
                TLV_LIST_STEP(&tlv_list);
        }
}

/**
 * do_these_links - perform operation on specified set of links
 * @funcToRun: operation to be performed on link
 * @domain: network domain of interest (0.0.0 if not used)
 * @str: link name pattern of interest (NULL if not used)
 * @vname: name of the parameter being set (optional arg to 'funcToRun')
 * @cmd: command to execute (optional arg to 'funcToRun')
 * @val: new value to be set (optional arg to 'funcToRun')
 *
 * This routine first retrieves the names of all links in the specified
 * network domain, eliminates those that don't match the specified search
 * pattern, and then performs the requestion operation on each remaining link.
 */

static void do_these_links(VOIDFUNCPTR funcToRun, __u32 domain, const char *str,
                           const char *vname, int cmd, int val)
{
        int tlv_space;
        int numLinks = 0;
        __u32 domain_net;
        struct tlv_list_desc tlv_list;
        struct tipc_link_info *local_link_info;

        domain_net = htonl(domain);
        tlv_space = TLV_SET(tlv_list_area, TIPC_TLV_NET_ADDR,
                            &domain_net, sizeof(domain_net));
        tlv_space = do_command(TIPC_CMD_GET_LINKS, tlv_list_area, tlv_space,
                               tlv_list_area, sizeof(tlv_list_area));

        TLV_LIST_INIT(&tlv_list, tlv_list_area, tlv_space);

        while (!TLV_LIST_EMPTY(&tlv_list)) {
                if (!TLV_LIST_CHECK(&tlv_list, TIPC_TLV_LINK_INFO))
                        fatal("corrupted reply message\n");
                local_link_info = (struct tipc_link_info *)TLV_LIST_DATA(&tlv_list);
                if ((str == NULL) ||
                                (strstr(local_link_info->str, str) != NULL)) {
                        funcToRun(local_link_info->str, local_link_info->up,
                                  vname, cmd, val);
                        numLinks++;
                }
                TLV_LIST_STEP(&tlv_list);
        }
```

```
        if (numLinks == 0) {
                if (str == NULL)
                        printf("No links found\n");
                else
                        printf("No links found matching pattern '%s'\n", str);
        }
}

static void get_link(char *linkName, __u32 up)
{
        printf("%s: %s\n", linkName, ntohl(up) ? "up" : "down");
}

static void get_linkset(char *args)
{
        char *strp = NULL;                      /* list all links by default */
        __u32 domain = 0;

        if (*args != 0) {
                if (args[0] == '?')
                        strp = args + 1;        /* list links matching pattern */
                else
                        domain = str2addr(args);/* list links in domain */
        }

        print_title("Links%s%s:\n", for_domain(" within domain ", domain));

        do_these_links(get_link, domain, strp, "", 0, 0);
}

static void show_link_stats(char *linkName)
{
        int tlv_space;

        tlv_space = TLV_SET(tlv_area, TIPC_TLV_LINK_NAME,
                            linkName, TIPC_MAX_LINK_NAME);
        tlv_space = do_command(TIPC_CMD_SHOW_LINK_STATS, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_ULTRA_STRING))
                fatal("corrupted reply message\n");

        printf("%s\n", (char *)TLV_DATA(tlv_area));
}

static void show_linkset_stats(char *args)
{
        print_title("Link statistics%s:\n", NULL);

        if (*args == 0)                 /* show for all links */
                do_these_links(show_link_stats, 0, NULL, NULL, 0, 0);
        else if (args[0] == '?')        /* show for all links matching pattern */
                do_these_links(show_link_stats, 0, args+1, NULL, 0, 0);
        else                            /* show for specified link */
                show_link_stats(args);
}

static void reset_link_stats(char *linkName)
{
        int tlv_space;

        tlv_space = TLV_SET(tlv_area, TIPC_TLV_LINK_NAME,
                            linkName, TIPC_MAX_LINK_NAME);
        tlv_space = do_command(TIPC_CMD_RESET_LINK_STATS, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        cprintf("Link %s statistics reset\n", linkName);
}

static void reset_linkset_stats(char *args)
{
        if (args[0] == '?')
                do_these_links(reset_link_stats, 0, args+1, NULL, 0, 0);
        else
                reset_link_stats(args);
}

static void show_name_table(char *args)
{
        int tlv_space;
        __u32 depth;
        __u32 type;
        __u32 lowbound;
        __u32 upbound;
        char dummy;
        struct tipc_name_table_query query_info;

        /* process (optional) depth argument */

        if (!*args)
                depth = 0;
        else if (args[0] == 'a')
                depth = 4;
        else if (args[0] == 'p')
                depth = 3;
        else if (args[0] == 'n')
                depth = 2;
```

```
        else if (args[0] == 't')
                depth = 1;
        else
                depth = 0;

        if (depth > 0) {
                args += strcspn(args, ",");
                if (*args)
                        args++;   /* skip over comma */
        } else {
                depth = 4;
        }

        /* process (optional) type arguments */

        if (!*args) {
                depth |= TIPC_NTQ_ALLTYPES;
                type = lowbound = upbound = 0;
        } else if (sscanf(args, "%u,%u,%u%c", &type, &lowbound, &upbound,
                          &dummy) == 3) {
                /* do nothing more */
        } else if (sscanf(args, "%u,%u%c", &type, &lowbound, &dummy) == 2) {
                upbound = lowbound;
        } else if (sscanf(args, "%u%c", &type, &dummy) == 1) {
                lowbound = 0;
                upbound = ~0;
        } else
                fatal("%s",usage);

        /* issue query & process response */

        query_info.depth = htonl(depth);
        query_info.type = htonl(type);
        query_info.lowbound = htonl(lowbound);
        query_info.upbound = htonl(upbound);

        tlv_space = TLV_SET(tlv_area, TIPC_TLV_NAME_TBL_QUERY,
                            &query_info, sizeof(query_info));
        tlv_space = do_command(TIPC_CMD_SHOW_NAME_TABLE, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_ULTRA_STRING))
                fatal("corrupted reply message\n");

        print_title_opt("Names%s:\n", "");
        printf("%s", (char *)TLV_DATA(tlv_area));
}

static void get_media(char *dummy)
{
        int tlv_space;
        struct tlv_list_desc tlv_list;

        tlv_space = do_command(TIPC_CMD_GET_MEDIA_NAMES, NULL, 0,
                               tlv_area, sizeof(tlv_area));

        print_title("Media%s:\n", NULL);
        if (!tlv_space) {
                printf("No registered media\n");
                return;
        }

        TLV_LIST_INIT(&tlv_list, tlv_area, tlv_space);
        while (!TLV_LIST_EMPTY(&tlv_list)) {
                if (!TLV_LIST_CHECK(&tlv_list, TIPC_TLV_MEDIA_NAME))
                        fatal("corrupted reply message\n");
                printf("%s\n", (char *)TLV_LIST_DATA(&tlv_list));
                TLV_LIST_STEP(&tlv_list);
        }
}


/**
 * do_these_bearers - perform operation on specified set of bearers
 * @funcToRun: operation to be performed on bearer
 * @str: bearer name pattern (if NULL, do operation on all bearers)
 */

static void do_these_bearers(VOIDFUNCPTR funcToRun, const char *str)
{
        int numBearers = 0;
        int tlv_space;
        struct tlv_list_desc tlv_list;
        char *bname;

        tlv_space = do_command(TIPC_CMD_GET_BEARER_NAMES, NULL, 0,
                               tlv_list_area, sizeof(tlv_list_area));

        TLV_LIST_INIT(&tlv_list, tlv_list_area, tlv_space);

        while (!TLV_LIST_EMPTY(&tlv_list)) {
                if (!TLV_LIST_CHECK(&tlv_list, TIPC_TLV_BEARER_NAME))
                        fatal("corrupted reply message\n");
                bname = (char *)TLV_LIST_DATA(&tlv_list);
                if ((str == NULL) || (strstr(bname, str) != NULL)) {
                        funcToRun(bname);
                        numBearers++;
                }
```

```
                TLV_LIST_STEP(&tlv_list);
        }

        if (numBearers == 0) {
                if (str == NULL)
                        printf("No active bearers\n");
                else
                        printf("No bearers found matching pattern '%s'\n", str);
        }
}

static void get_bearer(char *bname)
{
        printf("%s\n", bname);
}

static void get_bearerset(char *args)
{
        print_title("Bearers%s:\n", NULL);

        if (*args == 0)
                do_these_bearers(get_bearer, NULL);     /* list all bearers */
        else if (args[0] == '?')
                do_these_bearers(get_bearer, args+1);   /* list matching ones */
        else
                fatal("Invalid argument '%s' \n", args);
}

static void show_ports(char *dummy)
{
        int tlv_space;

        tlv_space = do_command(TIPC_CMD_SHOW_PORTS, NULL, 0,
                               tlv_area, sizeof(tlv_area));

        if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_ULTRA_STRING))
                fatal("corrupted reply message\n");

        print_title("Ports%s:\n", NULL);
        printf("%s", (char *)TLV_DATA(tlv_area));
}

static void set_log_size(char *args)
{
        int tlv_space;

        if (!*args) {
                tlv_space = do_command(TIPC_CMD_DUMP_LOG, NULL, 0,
                                       tlv_area, sizeof(tlv_area));

                if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_ULTRA_STRING))
                        fatal("corrupted reply message\n");

                printf("Log dump%s:\n%s", for_dest(), (char *)TLV_DATA(tlv_area));
        } else {
                do_set_unsigned(args, TIPC_CMD_SET_LOG_SIZE, "log size",
                                " (this will discard current log contents)");
        }
}
#if (LINUX_VERSION_CODE > KERNEL_VERSION(2,6,34))
static void show_stats(char *args)
{
        __u32 attr_val_net;
        int tlv_space;

        /*
         * In future, may allow user to control what info is returned;
         * for now, just hard code 0 as command argument to get default info
         */

        attr_val_net = htonl(0);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_UNSIGNED,
                            &attr_val_net, sizeof(attr_val_net));

        tlv_space = do_command(TIPC_CMD_SHOW_STATS, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        if (!TLV_CHECK(tlv_area, tlv_space, TIPC_TLV_ULTRA_STRING))
                fatal("corrupted reply message\n");

        print_title_opt("Status%s:\n", "");
        printf("%s", (char *)TLV_DATA(tlv_area));
}
#endif

static void set_link_value(char *linkName, __u32 dummy, const char *vname,
                           int cmd, int val)
{
        struct tipc_link_config req_tlv;
        int tlv_space;

        req_tlv.value = htonl(val);
        strcpy(req_tlv.name, linkName);
        req_tlv.name[TIPC_MAX_LINK_NAME - 1] = '\0';

        confirm("Change %s of link <%s>%s to %u? [Y/n]\n",
                vname, req_tlv.name, for_dest(), val);
```

```
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_LINK_CONFIG,
                            &req_tlv, sizeof(req_tlv));
        tlv_space = do_command(cmd, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        cprintf("Link <%s>%s changed %s to %u\n",
                req_tlv.name, for_dest(), vname, val);
}

static void set_linkset_value(char *args, const char *vname, int cmd)
{
        int  val;
        char dummy;
        char *s = strchr(args, '/');

        if (!s)
                fatal("Syntax: tipcConfig -l%c=<link-name>|<pattern>/<%s>\n",
                      vname[0], vname);

        *s++ = 0;

        if (sscanf(s, "%u%c", &val, &dummy) != 1)
                fatal("non-numeric link %s specified\n", vname);

        if (args[0] == '?')
                do_these_links(set_link_value, 0, args+1, vname, cmd, val);
        else
                set_link_value(args, 0, vname, cmd, val);
}

static void set_linkset_tolerance(char *args)
{
        set_linkset_value(args, "tolerance", TIPC_CMD_SET_LINK_TOL);
}

static void set_linkset_priority(char *args)
{
        set_linkset_value(args, "priority", TIPC_CMD_SET_LINK_PRI);
}

static void set_linkset_window(char *args)
{
        set_linkset_value(args, "window", TIPC_CMD_SET_LINK_WINDOW);
}

static int get_local_address(char *arg)
{
        char *opt;
        struct ifaddrs *ifap, *ifa;
        int i;
        struct sockaddr_in *addr = NULL;
        char ifaddr[16];
        char tmp[TIPC_MAX_BEARER_NAME];
        char *savep = tmp;

        memcpy(tmp, arg, TIPC_MAX_BEARER_NAME);

        opt = strsep(&savep , ":");
        if (!opt || !savep)
                return -EINVAL;
        if (strcmp(opt, MEDIA_NAME_UDP) != 0)
                        return 0;
        opt = strsep(&savep, ":");

        /*If an IP address was specified, use it directly*/
        if (inet_pton(AF_INET, opt, &addr))
                return 0;

        if (getifaddrs(&ifap)) {
                perror("getifaddrs");
                return -EINVAL;
        }
        /*Get the interface address*/
        for(ifa = ifap; ifa != NULL; ifa = ifa->ifa_next) {
                if ((ifa->ifa_addr->sa_family == AF_INET) &&
                   (strcmp(ifa->ifa_name, opt) == 0)) {
                        addr = ifa->ifa_addr;
                        break;
                }
        }
        if (!addr) {
                freeifaddrs(ifap);
                return -ENODEV;
        }
        if (NULL==inet_ntop(AF_INET, &addr->sin_addr, ifaddr,
            sizeof(struct sockaddr_in))) {
                freeifaddrs(ifap);
                perror("ntop");
                return -EINVAL;
        }
        if(savep)
                sprintf(arg, "%s:%s:%s\0", MEDIA_NAME_UDP, ifaddr, savep);
        else
                sprintf(arg, "%s:%s\0", MEDIA_NAME_UDP, ifaddr);
        freeifaddrs(ifap);
        return 0;
}
```

```
static int resolve_bearer_endpoint(char *arg)
{
        char tmp[TIPC_MAX_BEARER_NAME];
        char raddr[16];
        char *delim;
        struct sockaddr_in si_remote;
        struct addrinfo hints = {0};
        struct addrinfo *remote_info;
        int i;

        memcpy(tmp, arg, TIPC_MAX_BEARER_NAME);
        /*Get the fourth token (remote address)*/
        delim = strtok(tmp, ":");
        for (i = 0; i < 3; i++)
                if (!(delim = strtok(NULL, ":")))
                        return 0;
        if (inet_pton(AF_INET, delim, &si_remote.sin_addr))
                return 0;
        hints.ai_family = AF_INET;
        if (getaddrinfo(delim, NULL, &hints, &remote_info) != 0)
                return -EINVAL;
        /*Multiple addresses may be returned, but we just use the first one*/
        inet_ntop(AF_INET, &((struct sockaddr_in*)remote_info->ai_addr)->sin_addr,
                  raddr, 255);
        delim = tmp;
        for (i = 0; i < 3; i++)
                delim = strchr(delim, ':')+1;
        i = delim - tmp;
        delim = strchr(delim,':');
        snprintf(arg, i, tmp);
        if (delim)
                sprintf(arg + i - 1, ":%s%s",raddr,delim);
        else
                sprintf(arg + i - 1,":%s",raddr);
        return 0;
}


static void enable_bearer(char *args)
{
        struct tipc_bearer_config req_tlv;
        int tlv_space;
        char *a;
        char dummy;
        int err;

        while (args) {
                __u32 domain = dest & 0xfffff000; /* defaults to own cluster */
                uint pri = TIPC_MEDIA_LINK_PRI; /* defaults to media priority */
                char *domain_str, *pri_str;

                a = get_arg(&args);
                if ((domain_str = strchr(a, '/'))) {
                        *domain_str++ = 0;
                        if ((pri_str = strchr(domain_str, '/'))) {
                                *pri_str++ = 0;
                                if ((*pri_str != 0) &&
                                                sscanf(pri_str, "%u%c", &pri, &dummy) != 1)
                                        fatal("non-numeric bearer priority specified\n");
                        }
                        if (*domain_str != 0)
                                domain = str2addr(domain_str);
                }

                confirm("Enable bearer <%s>%s with detection domain %s and "
                        "priority %u? [Y/n]",
                        a, for_dest(), addr2str(domain), pri);

                req_tlv.priority = htonl(pri);
#if (LINUX_VERSION_CODE < KERNEL_VERSION(2,6,38))
                req_tlv.detect_scope = htonl(domain);
#else
                req_tlv.disc_domain = htonl(domain);
#endif
                if (err = get_local_address(a) != 0)
                        fatal("Invalid bearer parameters (%d)\n",err);
                if (err = resolve_bearer_endpoint(a) != 0) {
                        fatal("Could not resolve remote bearer endpoint name (%d)\n",
                              err);
                }
                strncpy(req_tlv.name, a, TIPC_MAX_BEARER_NAME - 1);
                req_tlv.name[TIPC_MAX_BEARER_NAME - 1] = '\0';

                tlv_space = TLV_SET(tlv_area, TIPC_TLV_BEARER_CONFIG,
                                    &req_tlv, sizeof(req_tlv));
                tlv_space = do_command(TIPC_CMD_ENABLE_BEARER, tlv_area, tlv_space,
                                       tlv_area, sizeof(tlv_area));

                cprintf("Bearer <%s> enabled%s\n", a, for_dest());
        }
}

static void disable_bearer(char *bname)
{
        char bearer_name[TIPC_MAX_BEARER_NAME];
        int tlv_space;
        int err;
```

```
        strncpy(bearer_name, bname, TIPC_MAX_BEARER_NAME - 1);
        bearer_name[TIPC_MAX_BEARER_NAME - 1] = '\0';

        confirm("Disable bearer <%s>%s ? [Y/n]", bearer_name, for_dest());

        if (err = get_local_address(bearer_name) != 0)
                fatal("Invalid bearer parameters (%d)\n",err);
        tlv_space = TLV_SET(tlv_area, TIPC_TLV_BEARER_NAME,
                            bearer_name, sizeof(bearer_name));
        tlv_space = do_command(TIPC_CMD_DISABLE_BEARER, tlv_area, tlv_space,
                               tlv_area, sizeof(tlv_area));

        cprintf("Bearer <%s> disabled%s\n", bearer_name, for_dest());
}

static void disable_bearerset(char *args)
{
        if (args[0] == '?')
                do_these_bearers(disable_bearer, args+1); /* name pattern */
        else {
                while (args) {
                        disable_bearer(get_arg(&args)); /* list of names */
                }
        }
}


/*****************************************************************************
 *
 * Basic data structures and routines associated with command/option processing
 *
 * Terminology note: The arguments passed to tipc-config are usually referred
 * to as "commands", since most of them are actually requests that are passed
 * on to TIPC rather than directives that are executed by tipc-config itself.
 * However, since tipc-config utilizes Linux's command line library to parse
 * the commands as if they were options, the latter term is also acceptable.
 *
 */

#define OPT_BASE '@'

struct command {
        void (*fcn) (char *args);
        char args[ARGS_SIZE];
};

/*
 * Help string generated by tipc-config application;
 * command entries are listed alphabetically
 */

static char usage[] =
        "Usage: \n"
        "       tipc-config command [command ...]\n"
        "  \n"
        "  valid commands:\n"
        "  -addr [=<addr>]                              Get/set node address\n"
        "  -b    [=<bearerpat>]                         Get bearers\n"
        "  -bd    =<bearerpat>                          Disable bearer\n"
        "  -be    =<bearer>[/<domain>[/<priority>]]]    Enable bearer\n"
        "  -dest  =<addr>                               Command destination node\n"
        "  -help                                        This usage list\n"
        "  -i                                           Interactive set operations\n"
        "  -l    [=<domain>|<linkpat>]                  Get links to domain\n"
        "  -log  [=<size>]                              Dump/resize log\n"
        "  -lp    =<linkpat>|<bearer>|<media>/<value>   Set link priority\n"
        "  -ls   [=<linkpat>]                           Get link statistics\n"
        "  -lsr   =<linkpat>                            Reset link statistics\n"
        "  -lt    =<linkpat>|<bearer>|<media>/<value>   Set link tolerance\n"
        "  -lw    =<linkpat>|<bearer>|<media>/<value>   Set link window\n"
        "  -m                                           Get media\n"
        "  -max_ports    [=<value>]                     Get/set max number of ports\n"
        "  -max_publ     [=<value>]                     Get/set max publications\n"
        "  -max_subscr   [=<value>]                     Get/set max subscriptions\n"
        "  -mng  [=enable|disable]                      Get/set remote management\n"
        "  -n    [=<domain>]                            Get nodes in domain\n"
        "  -netid[=<value>]                             Get/set network id\n"
        "  -nt   [=[<depth>,]<type>[,<low>[,<up>]]]     Get name table\n"
        "         where <depth> = types|names|ports|all\n"
        "  -p                                           Get port info\n"
        "  -s                                           Get TIPC status info\n"
        "  -v                                           Verbose output\n"
        "  -V                                           Get tipc-config version info\n"
        ; /* end of concatenated string literal */

/*
 * Option structure field usage in tipc-config application:
 *      1) option name
 *      2) argument count
 *              0 if argument is not allowed
 *              1 if argument is required
 *              2 if argument is optional
 *      3) always set to 0
 *      4) value to return
 *
 * Note 1: Option name field must match the info in "usage" (above).
 * Note 2: Entries need not be stored alphabetically, but "value to return"
 *         field must reflect ordering used in "cmd_array" (below).
```

```
 */

static struct option options[] = {
        {"help",         0, 0, '0'},
        {"v",            0, 0, '1'},
        {"i",            0, 0, '2'},
        {"dest",         1, 0, '3'},
        {"V",            0, 0, '4'},
        {"addr",         2, 0, OPT_BASE + 0},
        {"netid",        2, 0, OPT_BASE + 1},
        {"mng",          2, 0, OPT_BASE + 2},
        {"nt",           2, 0, OPT_BASE + 3},
        {"p",            0, 0, OPT_BASE + 4},
        {"m",            0, 0, OPT_BASE + 5},
        {"b",            2, 0, OPT_BASE + 6},
        {"be",           1, 0, OPT_BASE + 7},
        {"bd",           1, 0, OPT_BASE + 8},
        {"n",            2, 0, OPT_BASE + 9},
        {"l",            2, 0, OPT_BASE + 10},
        {"ls",           2, 0, OPT_BASE + 11},
        {"lsr",          1, 0, OPT_BASE + 12},
        {"lp",           1, 0, OPT_BASE + 13},
        {"lw",           1, 0, OPT_BASE + 14},
        {"lt",           1, 0, OPT_BASE + 15},
        {"max_ports",    2, 0, OPT_BASE + 16},
        {"max_subscr",   2, 0, OPT_BASE + 17},
        {"max_publ",     2, 0, OPT_BASE + 18},
        {"log",          2, 0, OPT_BASE + 19},
        {"s",            0, 0, OPT_BASE + 20},
        {0, 0, 0, 0}
};

void (*cmd_array[])(char *args) = {
        set_node_addr,
        set_netid,
        set_remote_mng,
        show_name_table,
        show_ports,
        get_media,
        get_bearerset,
        enable_bearer,
        disable_bearerset,
        get_nodes,
        get_linkset,
        show_linkset_stats,
        reset_linkset_stats,
        set_linkset_priority,
        set_linkset_window,
        set_linkset_tolerance,
        set_max_ports,
        set_max_subscr,
        set_max_publ,
        set_log_size,
#if (LINUX_VERSION_CODE > KERNEL_VERSION(2,6,34))
        show_stats,
#endif
        NULL
};

/*
 * Mainline parses option list and processes each command.  Most commands are
 * not actually executed until parsing is complete in case they are impacted
 * by commands that appear later in the list.
 */

int main(int argc, char *argv[], char *dummy[])
{
        struct command commands[MAX_COMMANDS];
        int cno, cno2;
        int c;

        if (argc == 1)
                fatal("%s",usage);

        dest = own_node();

        cno = 0;
        while ((c = getopt_long_only(argc, argv, "", options, NULL)) != EOF) {

                if (c >= OPT_BASE) {
                        if (cno >= MAX_COMMANDS)
                                fatal("too many commands specified\n");

                        commands[cno].fcn = cmd_array[c - OPT_BASE];
                        if (optarg)
                                strcpy(commands[cno].args, optarg);
                        else
                                commands[cno].args[0] = '\0';
                        cno++;
                } else {
                        switch (c) {
                        case '0':
                                fatal("%s", usage);
                                break;
                        case '1':
                                verbose = 1;
                                break;
                        case '2':
```

```
                                interactive = 1;
                                break;
                        case '3':
                                dest = str2addr(optarg);
                                break;
                        case '4':
                                printf("TIPC configuration tool version "
                                       VERSION "\n");
                                break;
                        default:
                                /* getopt_long_only() generates the error msg */
                                exit(EXIT_FAILURE);
                                break;
                        }
                }

        }

        if (optind < argc) {
                /* detects arguments that don't start with a '-' sign */
                fatal("unexpected command argument '%s'\n", argv[optind]);
        }

        for (cno2 = 0; cno2 < cno; cno2++) {
                if (!commands[cno2].fcn)
                        fatal("command table error\n");
                commands[cno2].fcn(commands[cno2].args);
        }

        return 0;
}
```

Notice for package(s)

bridge-utils

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) 19yy name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

Notice for package(s)

libpcap
tcpdump

```
License: BSD

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in
     the documentation and/or other materials provided with the
     distribution.
  3. The names of the authors may not be used to endorse or promote
     products derived from this software without specific prior
     written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
```

Notice for package(s)

libpcap

```
/*
 * Copyright (c) 1993, 1994, 1995, 1996, 1997
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the Computer Systems
 *      Engineering Group at Lawrence Berkeley Laboratory.
 * 4. Neither the name of the University nor of the Laboratory may be used
 *    to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

/*
 * For backwards compatibility.
 *
 * Note to OS vendors: do NOT get rid of this file!  Many applications
 * expect to be able to include <pcap.h>, and at least some of them
 * go through contortions in their configure scripts to try to detect
 * OSes that have "helpfully" moved pcap.h to <pcap/pcap.h> without
 * leaving behind a <pcap.h> file.
 */
#include <pcap/pcap.h>
```

Notice for package(s)

util-linux

```
The project util-linux doesn't use the same license for all of the code.
There is code under:

   * GPLv2+ (GNU General Public License version 2, or any later version)

   * GPLv2 (GNU General Public License version 2)

   * LGPLv2+ (GNU Lesser General Public License v2 (or 2.1) or any later version)
```

```
   * BSD with advertising

   * Public Domain

Please, check the source code for more details. A license is usually at the start
of each source file.

The ./COPYING file (GPLv2+) is the default license for code without an explicitly
defined license.
```

Notice for package(s)

util-linux

```
/*
 * Copyright (c) 1989 The Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the University of
 *      California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
```

Notice for package(s)

util-linux

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later
version.

The complete text of the license is available in the
../Documentation/licenses/COPYING.LGPLv2.1 file.
```

Notice for package(s)

gzip

```
/* gzip.h -- common declarations for all gzip modules

   Copyright (C) 1997-1999, 2001, 2006-2007, 2009-2013 Free Software
   Foundation, Inc.

   Copyright (C) 1992-1993 Jean-loup Gailly.

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 3, or (at your option)
   any later version.

   This program is distributed in the hope that it will be useful,
```

```
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software Foundation,
   Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.  */

#ifdef __STDC__
   typedef void *voidp;
#else
   typedef char *voidp;
#endif

#ifndef __attribute__
# if __GNUC__ < 2 || (__GNUC__ == 2 && __GNUC_MINOR__ < 8) || __STRICT_ANSI__
#  define __attribute__(x)
# endif
#endif

#ifndef ATTRIBUTE_NORETURN
# define ATTRIBUTE_NORETURN __attribute__ ((__noreturn__))
#endif

/* I don't like nested includes, but the following headers are used
 * too often
 */
#include <stdio.h>
#include <sys/types.h> /* for off_t */
#include <time.h>
#include <string.h>
#define memzero(s, n) memset ((voidp)(s), 0, (n))

#ifndef RETSIGTYPE
#  define RETSIGTYPE void
#endif

#define local static

typedef unsigned char  uch;
typedef unsigned short ush;
typedef unsigned long  ulg;

/* Return codes from gzip */
#define OK      0
#define ERROR   1
#define WARNING 2

/* Compression methods (see algorithm.doc) */
#define STORED      0
#define COMPRESSED  1
#define PACKED      2
#define LZHED       3
/* methods 4 to 7 reserved */
#define DEFLATED    8
#define MAX_METHODS 9
extern int method;         /* compression method */

/* To save memory for 16 bit systems, some arrays are overlaid between
 * the various modules:
 * deflate:  prev+head   window      d_buf  l_buf  outbuf
 * unlzw:    tab_prefix  tab_suffix  stack  inbuf  outbuf
 * inflate:              window              inbuf
 * unpack:               window              inbuf  prefix_len
 * unlzh:    left+right  window      c_table inbuf c_len
 * For compression, input is done in window[]. For decompression, output
 * is done in window except for unlzw.
 */

#ifndef INBUFSIZ
#  ifdef SMALL_MEM
#    define INBUFSIZ  0x2000  /* input buffer size */
#  else
#    define INBUFSIZ  0x8000  /* input buffer size */
#  endif
#endif
#define INBUF_EXTRA  64     /* required by unlzw() */

#ifndef OUTBUFSIZ
#  ifdef SMALL_MEM
#    define OUTBUFSIZ   8192  /* output buffer size */
#  else
#    define OUTBUFSIZ  16384  /* output buffer size */
#  endif
#endif
#define OUTBUF_EXTRA 2048   /* required by unlzw() */

#ifndef DIST_BUFSIZE
#  ifdef SMALL_MEM
#    define DIST_BUFSIZE 0x2000 /* buffer for distances, see trees.c */
#  else
#    define DIST_BUFSIZE 0x8000 /* buffer for distances, see trees.c */
#  endif
#endif

#ifdef DYN_ALLOC
#  define EXTERN(type, array)  extern type * near array
#  define DECLARE(type, array, size)  type * near array
```

```
#  define ALLOC(type, array, size) { \
      array = (type*)fcalloc((size_t)(((size)+1L)/2), 2*sizeof(type)); \
      if (!array) xalloc_die (); \
   }
#  define FREE(array) {if (array != NULL) fcfree(array), array=NULL;}
#else
#  define EXTERN(type, array)  extern type array[]
#  define DECLARE(type, array, size)  type array[size]
#  define ALLOC(type, array, size)
#  define FREE(array)
#endif

EXTERN(uch, inbuf);          /* input buffer */
EXTERN(uch, outbuf);         /* output buffer */
EXTERN(ush, d_buf);          /* buffer for distances, see trees.c */
EXTERN(uch, window);         /* Sliding window and suffix table (unlzw) */
#define tab_suffix window
#ifndef MAXSEG_64K
#  define tab_prefix prev    /* hash link (see deflate.c) */
#  define head (prev+WSIZE)  /* hash head (see deflate.c) */
   EXTERN(ush, tab_prefix);  /* prefix code (see unlzw.c) */
#else
#  define tab_prefix0 prev
#  define head tab_prefix1
   EXTERN(ush, tab_prefix0); /* prefix for even codes */
   EXTERN(ush, tab_prefix1); /* prefix for odd  codes */
#endif

extern unsigned insize; /* valid bytes in inbuf */
extern unsigned inptr;  /* index of next byte to be processed in inbuf */
extern unsigned outcnt; /* bytes in output buffer */

extern off_t bytes_in;   /* number of input bytes */
extern off_t bytes_out;  /* number of output bytes */
extern off_t header_bytes;/* number of bytes in gzip header */

extern int  ifd;        /* input file descriptor */
extern int  ofd;        /* output file descriptor */
extern char ifname[];   /* input file name or "stdin" */
extern char ofname[];   /* output file name or "stdout" */
extern char *program_name;  /* program name */

extern struct timespec time_stamp; /* original time stamp (modification time) */
extern off_t ifile_size; /* input file size, -1 for devices (debug only) */

typedef int file_t;     /* Do not use stdio */
#define NO_FILE  (-1)   /* in memory compression */


#define PACK_MAGIC     "\037\036" /* Magic header for packed files */
#define GZIP_MAGIC     "\037\213" /* Magic header for gzip files, 1F 8B */
#define OLD_GZIP_MAGIC "\037\236" /* Magic header for gzip 0.5 = freeze 1.x */
#define LZH_MAGIC      "\037\240" /* Magic header for SCO LZH Compress files*/
#define PKZIP_MAGIC    "\120\113\003\004" /* Magic header for pkzip files */

/* gzip flag byte */
#define ASCII_FLAG   0x01 /* bit 0 set: file probably ascii text */
#define HEADER_CRC   0x02 /* bit 1 set: CRC16 for the gzip header */
#define EXTRA_FIELD  0x04 /* bit 2 set: extra field present */
#define ORIG_NAME    0x08 /* bit 3 set: original file name present */
#define COMMENT      0x10 /* bit 4 set: file comment present */
#define ENCRYPTED    0x20 /* bit 5 set: file is encrypted */
#define RESERVED     0xC0 /* bit 6,7:   reserved */

/* internal file attribute */
#define UNKNOWN 0xffff
#define BINARY  0
#define ASCII   1

#ifndef WSIZE
#  define WSIZE 0x8000     /* window size--must be a power of two, and */
#endif                     /*  at least 32K for zip's deflate method */

#define MIN_MATCH  3
#define MAX_MATCH  258
/* The minimum and maximum match lengths */

#define MIN_LOOKAHEAD (MAX_MATCH+MIN_MATCH+1)
/* Minimum amount of lookahead, except at the end of the input file.
 * See deflate.c for comments about the MIN_MATCH+1.
 */

#define MAX_DIST  (WSIZE-MIN_LOOKAHEAD)
/* In order to simplify the code, particularly on 16 bit machines, match
 * distances are limited to MAX_DIST instead of WSIZE.
 */

extern int exit_code;      /* program exit code */
extern int verbose;        /* be verbose (-v) */
extern int quiet;          /* be quiet (-q) */
extern int level;          /* compression level */
extern int test;           /* check .z file integrity */
extern int to_stdout;      /* output to stdout (-c) */
extern int save_orig_name; /* set if original name must be saved */

#define get_byte()  (inptr < insize ? inbuf[inptr++] : fill_inbuf(0))
#define try_byte()  (inptr < insize ? inbuf[inptr++] : fill_inbuf(1))
```

```
/* put_byte is used for the compressed output, put_ubyte for the
 * uncompressed output. However unlzw() uses window for its
 * suffix table instead of its output buffer, so it does not use put_ubyte
 * (to be cleaned up).
 */
#define put_byte(c) {outbuf[outcnt++]=(uch)(c); if (outcnt==OUTBUFSIZ)\
   flush_outbuf();}
#define put_ubyte(c) {window[outcnt++]=(uch)(c); if (outcnt==WSIZE)\
   flush_window();}

/* Output a 16 bit value, lsb first */
#define put_short(w) \
{ if (outcnt < OUTBUFSIZ-2) { \
    outbuf[outcnt++] = (uch) ((w) & 0xff); \
    outbuf[outcnt++] = (uch) ((ush)(w) >> 8); \
  } else { \
    put_byte((uch)((w) & 0xff)); \
    put_byte((uch)((ush)(w) >> 8)); \
  } \
}

/* Output a 32 bit value to the bit stream, lsb first */
#define put_long(n) { \
    put_short((n) & 0xffff); \
    put_short(((ulg)(n)) >> 16); \
}

#define seekable()    0  /* force sequential output */
#define translate_eol 0  /* no option -a yet */

#define tolow(c)  (isupper (c) ? tolower (c) : (c))  /* force to lower case */

/* Macros for getting two-byte and four-byte header values */
#define SH(p) ((ush)(uch)((p)[0]) | ((ush)(uch)((p)[1]) << 8))
#define LG(p) ((ulg)(SH(p)) | ((ulg)(SH((p)+2)) << 16))

/* Diagnostic functions */
#ifdef DEBUG
#  define Assert(cond,msg) {if (!(cond)) gzip_error (msg);}
#  define Trace(x) fprintf x
#  define Tracev(x) {if (verbose) fprintf x ;}
#  define Tracevv(x) {if (verbose>1) fprintf x ;}
#  define Tracec(c,x) {if (verbose && (c)) fprintf x ;}
#  define Tracecv(c,x) {if (verbose>1 && (c)) fprintf x ;}
#else
#  define Assert(cond,msg)
#  define Trace(x)
#  define Tracev(x)
#  define Tracevv(x)
#  define Tracec(c,x)
#  define Tracecv(c,x)
#endif

#define WARN(msg) {if (!quiet) fprintf msg ; \
                   if (exit_code == OK) exit_code = WARNING;}

        /* in zip.c: */
extern int zip        (int in, int out);
extern int file_read  (char *buf,  unsigned size);

        /* in unzip.c */
extern int unzip      (int in, int out);
extern int check_zipfile (int in);

        /* in unpack.c */
extern int unpack     (int in, int out);

        /* in unlzh.c */
extern int unlzh      (int in, int out);

        /* in gzip.c */
extern void abort_gzip (void) ATTRIBUTE_NORETURN;

        /* in deflate.c */
extern void lm_init (int pack_level, ush *flags);
extern off_t deflate (void);

        /* in trees.c */
extern void ct_init     (ush *attr, int *method);
extern int  ct_tally    (int dist, int lc);
extern off_t flush_block (char *buf, ulg stored_len, int eof);

        /* in bits.c */
extern void     bi_init    (file_t zipfile);
extern void     send_bits  (int value, int length);
extern unsigned bi_reverse (unsigned value, int length) _GL_ATTRIBUTE_CONST;
extern void     bi_windup  (void);
extern void     copy_block (char *buf, unsigned len, int header);
extern int     (*read_buf) (char *buf, unsigned size);

        /* in util.c: */
extern int copy           (int in, int out);
extern ulg  updcrc        (uch *s, unsigned n);
extern void clear_bufs    (void);
extern int  fill_inbuf    (int eof_ok);
extern void flush_outbuf  (void);
extern void flush_window  (void);
extern void write_buf     (int fd, voidp buf, unsigned cnt);
```

```
extern int read_buffer    (int fd, voidp buf, unsigned int cnt);
extern char *strlwr       (char *s);
extern char *gzip_base_name (char *fname) _GL_ATTRIBUTE_PURE;
extern int xunlink        (char *fname);
extern void make_simple_name (char *name);
extern char *add_envopt   (int *argcp, char ***argvp, char const *env);
extern void gzip_error    (char const *m) ATTRIBUTE_NORETURN;
extern void xalloc_die    (void) ATTRIBUTE_NORETURN;
extern void warning       (char const *m);
extern void read_error    (void) ATTRIBUTE_NORETURN;
extern void write_error   (void) ATTRIBUTE_NORETURN;
extern void display_ratio (off_t num, off_t den, FILE *file);
extern void fprint_off    (FILE *, off_t, int);

        /* in inflate.c */
extern int inflate (void);
```

Notice for package(s)

attr

```
Most components of the "attr" package are licensed under
Version 2.1 of the GNU Lesser General Public License (see COPYING.LGPL).

Some components (as annotated in the source) are licensed
under Version 2 of the GNU General Public License (see below),

------------------------------------------------------------------------

                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and
modification follow.

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
```

a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code. (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

```
  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.  SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```

Notice for package(s)

attr

```
/*
 * Copyright (c) 2000-2002,2004 Silicon Graphics, Inc.
 * All Rights Reserved.
 *
 * This program is free software: you can redistribute it and/or modify it
```

```
 * under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/stat.h>
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <locale.h>

#include <attr/attributes.h>
#include "config.h"

#define SETOP           1               /* do a SET operation */
#define GETOP           2               /* do a GET operation */
#define REMOVEOP        3               /* do a REMOVE operation */
#define LISTOP          4               /* do a LIST operation */

#define BUFSIZE         (60*1024)       /* buffer size for LIST operations */

static char *progname;

void
usage(void)
{
        fprintf(stderr, _(
"Usage: %s [-LRSq] -s attrname [-V attrvalue] pathname  # set value\n"
"       %s [-LRSq] -g attrname pathname                 # get value\n"
"       %s [-LRSq] -r attrname pathname                 # remove attr\n"
"       %s [-LRq]  -l pathname                          # list attrs \n"
"      -s reads a value from stdin and -g writes a value to stdout\n"),
                progname, progname, progname, progname);
        exit(1);
}

int
main(int argc, char **argv)
{
        char *attrname, *attrvalue, *filename, *buffer;
        int attrlength, attrflags;
        int opflag, i, ch, error, follow, verbose, rootflag, secureflag;
        attrlist_t *alist;
        attrlist_ent_t *aep;
        attrlist_cursor_t cursor;

        progname = basename(argv[0]);

        setlocale(LC_CTYPE, "");
        setlocale(LC_MESSAGES, "");
        bindtextdomain(PACKAGE, LOCALEDIR);
        textdomain(PACKAGE);

        /*
         * Pick up and validate the arguments.
         */
        verbose = 1;
        follow = opflag = rootflag = secureflag = 0;
        attrname = attrvalue = NULL;
        while ((ch = getopt(argc, argv, "s:V:g:r:lqLRS")) != EOF) {
                switch (ch) {
                case 's':
                        if ((opflag != 0) && (opflag != SETOP)) {
                                fprintf(stderr,
                                  _("Only one of -s, -g, -r, or -l allowed\n"));
                                usage();
                        }
                        opflag = SETOP;
                        attrname = optarg;
                        break;
                case 'V':
                        if ((opflag != 0) && (opflag != SETOP)) {
                                fprintf(stderr, _("-V only allowed with -s\n"));
                                usage();
                        }
                        opflag = SETOP;
                        attrvalue = optarg;
                        break;
                case 'g':
                        if (opflag) {
                                fprintf(stderr,
                                  _("Only one of -s, -g, -r, or -l allowed\n"));
                                usage();
                        }
                        opflag = GETOP;
```

```
                attrname = optarg;
                break;
        case 'r':
                if (opflag) {
                        fprintf(stderr,
                          _("Only one of -s, -g, -r, or -l allowed\n"));
                        usage();
                }
                opflag = REMOVEOP;
                attrname = optarg;
                break;
        case 'l':
                if (opflag) {
                        fprintf(stderr,
                          _("Only one of -s, -g, -r, or -l allowed\n"));
                        usage();
                }
                opflag = LISTOP;
                break;
        case 'L':
                follow++;
                break;
        case 'R':
                rootflag++;
                break;
        case 'S':
                secureflag++;
                break;
        case 'q':
                verbose = 0;
                break;
        default:
                fprintf(stderr, _("Unrecognized option: %c\n"),
                        (char)ch);
                usage();
                break;
        }
}
if (optind != argc-1) {
        fprintf(stderr, _("A filename to operate on is required\n"));
        usage();
}
filename = argv[optind];

attrflags = ((!follow ? ATTR_DONTFOLLOW : 0) |
             (secureflag ? ATTR_SECURE : 0) |
             (rootflag ? ATTR_ROOT : 0));
/*
 * Break out into option-specific processing.
 */
switch (opflag) {
case SETOP:
        if (attrvalue == NULL) {
                attrvalue = malloc(ATTR_MAX_VALUELEN);
                if (attrvalue == NULL) {
                        perror("malloc");
                        exit(1);
                }
                attrlength =
                        fread(attrvalue, 1, ATTR_MAX_VALUELEN, stdin);
        } else {
                attrlength = strlen(attrvalue);
        }
        error = attr_set(filename, attrname, attrvalue,
                                 attrlength, attrflags);
        if (error) {
                perror("attr_set");
                fprintf(stderr, _("Could not set \"%s\" for %s\n"),
                                attrname, filename);
                exit(1);
        }
        if (verbose) {
                printf(_("Attribute \"%s\" set to a %d byte value "
                       "for %s:\n"), attrname, attrlength, filename);
                fwrite(attrvalue, 1, attrlength, stdout);
                printf("\n");
        }
        break;

case GETOP:
        attrvalue = malloc(ATTR_MAX_VALUELEN);
        if (attrvalue == NULL) {
                perror("malloc");
                exit(1);
        }
        attrlength = ATTR_MAX_VALUELEN;
        error = attr_get(filename, attrname, attrvalue,
                                 &attrlength, attrflags);
        if (error) {
                perror("attr_get");
                fprintf(stderr, _("Could not get \"%s\" for %s\n"),
                                attrname, filename);
                exit(1);
        }
        if (verbose) {
                printf(_("Attribute \"%s\" had a %d byte value "
                        "for %s:\n"), attrname, attrlength, filename);
        }
```

```
                fwrite(attrvalue, 1, attrlength, stdout);
                if (verbose) {
                        printf("\n");
                }
                break;

        case REMOVEOP:
                error = attr_remove(filename, attrname, attrflags);
                if (error) {
                        perror("attr_remove");
                        fprintf(stderr, _("Could not remove \"%s\" for %s\n"),
                                        attrname, filename);
                        exit(1);
                }
                break;

        case LISTOP:
                if ((buffer = malloc(BUFSIZE)) == NULL) {
                        perror("malloc");
                        exit(1);
                }
                bzero((char *)&cursor, sizeof(cursor));
                do {
                        error = attr_list(filename, buffer, BUFSIZE,
                                        attrflags, &cursor);
                        if (error) {
                                perror("attr_list");
                                fprintf(stderr,
                                        _("Could not list \"%s\" for %s\n"),
                                        attrname, filename);
                                exit(1);
                        }

                        alist = (attrlist_t *)buffer;
                        for (i = 0; i < alist->al_count; i++) {
                                aep = (attrlist_ent_t *)&buffer[ alist->al_offset[i] ];
                                if (verbose) {
                                        printf(
                        _("Attribute \"%s\" has a %d byte value for %s\n"),
                                                aep->a_name, aep->a_valuelen,
                                                filename);
                                } else {
                                        printf("%s\n", aep->a_name);
                                }
                        }
                } while (alist->al_more);
                break;

        default:
                fprintf(stderr,
                        _("At least one of -s, -g, -r, or -l is required\n"));
                usage();
                break;
        }

        return(0);
}
```

Notice for package(s)

attr

```
/*
 * Copyright (c) 2001-2003,2005 Silicon Graphics, Inc.
 * All Rights Reserved.
 *
 * This program is free software: you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as published
 * by the Free Software Foundation, either version 2.1 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */

#include <errno.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>

#include <attr/xattr.h>
#include <attr/attributes.h>

#undef MAXNAMELEN
#define MAXNAMELEN 256
```

```
#undef MAXLISTLEN
#define MAXLISTLEN 65536

#undef roundup
#define roundup(x,y) ((((x)+((y)-1))/(y))*(y))

static const char *user_name = "user.";
static const char *secure_name = "security.";
static const char *trusted_name = "trusted.";
static const char *xfsroot_name = "xfsroot.";

/*
 * Convert IRIX API components into Linux/XFS API components,
 * and vice-versa.
 */
static int
api_convert(char *name, const char *irixname, int irixflags, int compat)
{
        if (strlen(irixname) >= MAXNAMELEN) {
                errno = EINVAL;
                return -1;
        }
        if (irixflags & ATTR_ROOT) {
                if (compat)
                        strcpy(name, xfsroot_name);
                else
                        strcpy(name, trusted_name);
        } else if (irixflags & ATTR_SECURE) {
                strcpy(name, secure_name);
        } else {
                strcpy(name, user_name);
        }
        strcat(name, irixname);
        return 0;
}

static int
api_unconvert(char *name, const char *linuxname, int irixflags)
{
        int type, length;

        length = strlen(user_name);
        if (strncmp(linuxname, user_name, length) == 0) {
                type = 0; /*ATTR_USER*/
                goto found;
        }
        length = strlen(secure_name);
        if (strncmp(linuxname, secure_name, length) == 0) {
                type = ATTR_SECURE;
                goto found;
        }
        length = strlen(trusted_name);
        if (strncmp(linuxname, trusted_name, length) == 0) {
                type = ATTR_ROOT;
                goto found;
        }
        length = strlen(xfsroot_name);
        if (strncmp(linuxname, xfsroot_name, length) == 0) {
                type = ATTR_ROOT;
                goto found;
        }
        return 1;

found:
        if ((irixflags & ATTR_SECURE) != 0 && (type != ATTR_SECURE))
                return 1;
        if ((irixflags & ATTR_ROOT) != 0 && (type != ATTR_ROOT))
                return 1;
        strcpy(name, linuxname + length);
        return 0;
}


int
attr_get(const char *path, const char *attrname, char *attrvalue,
         int *valuelength, int flags)
{
        int c, compat;
        char name[MAXNAMELEN+16];

        for (compat = 0; compat < 2; compat++) {
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                if (flags & ATTR_DONTFOLLOW)
                        c = lgetxattr(path, name, attrvalue, *valuelength);
                else
                        c =  getxattr(path, name, attrvalue, *valuelength);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        if (c < 0)
                return c;
        *valuelength = c;
        return 0;
}

int
```

```
attr_getf(int fd, const char *attrname, char *attrvalue,
          int *valuelength, int flags)
{
        int c, compat;
        char name[MAXNAMELEN+16];

        for (compat = 0; compat < 2; compat++) {
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                c = fgetxattr(fd, name, attrvalue, *valuelength);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        if (c < 0)
                return c;
        *valuelength = c;
        return 0;
}

int
attr_set(const char *path, const char *attrname, const char *attrvalue,
         const int valuelength, int flags)
{
        int c, compat, lflags = 0;
        char name[MAXNAMELEN+16];
        void *buffer = (void *)attrvalue;

        if (flags & ATTR_CREATE)
                lflags = XATTR_CREATE;
        else if (flags & ATTR_REPLACE)
                lflags = XATTR_REPLACE;

        for (compat = 0; compat < 2; compat++) {
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                if (flags & ATTR_DONTFOLLOW)
                        c = lsetxattr(path, name, buffer, valuelength, lflags);
                else
                        c = setxattr(path, name, buffer, valuelength, lflags);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        return c;
}

int
attr_setf(int fd, const char *attrname,
          const char *attrvalue, const int valuelength, int flags)
{
        int c, compat, lflags = 0;
        char name[MAXNAMELEN+16];
        void *buffer = (void *)attrvalue;

        if (flags & ATTR_CREATE)
                lflags = XATTR_CREATE;
        else if (flags & ATTR_REPLACE)
                lflags = XATTR_REPLACE;

        for (compat = 0; compat < 2; compat++) {
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                c = fsetxattr(fd, name, buffer, valuelength, lflags);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        return c;
}

int
attr_remove(const char *path, const char *attrname, int flags)
{
        int c, compat;
        char name[MAXNAMELEN+16];

        for (compat = 0; compat < 2; compat++) {
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                if (flags & ATTR_DONTFOLLOW)
                        c = lremovexattr(path, name);
                else
                        c = removexattr(path, name);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        return c;
}

int
attr_removef(int fd, const char *attrname, int flags)
{
        int c, compat;
        char name[MAXNAMELEN+16];

        for (compat = 0; compat < 2; compat++) {
```

```
                if ((c = api_convert(name, attrname, flags, compat)) < 0)
                        return c;
                c = fremovexattr(fd, name);
                if (c < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                break;
        }
        return c;
}


/*
 * Helper routine for attr_list functions.
 */

static int
attr_list_pack(const char *name, const int valuelen,
                char *buffer, const int buffersize,
                int *start_offset, int *end_offset)
{
        attrlist_ent_t *aentp;
        attrlist_t *alist = (attrlist_t *)buffer;
        int size = roundup(strlen(name) + 1 + sizeof(aentp->a_valuelen), 8);

        if ((*end_offset - size) < (*start_offset + sizeof(alist->al_count))) {
                alist->al_more = 1;
                return 1;
        }

        *end_offset -= size;
        aentp = (attrlist_ent_t *)&buffer[ *end_offset ];
        aentp->a_valuelen = valuelen;
        strncpy(aentp->a_name, name, size - sizeof(aentp->a_valuelen));

        *start_offset += sizeof(alist->al_offset);
        alist->al_offset[alist->al_count] = *end_offset;
        alist->al_count++;
        return 0;
}

int
attr_list(const char *path, char *buffer, const int buffersize, int flags,
          attrlist_cursor_t *cursor)
{
        const char *l;
        int length, vlength, count = 0;
        char lbuf[MAXLISTLEN];
        char name[MAXNAMELEN+16];
        int start_offset, end_offset;

        if (buffersize < sizeof(attrlist_t)) {
                errno = EINVAL;
                return -1;
        }
        bzero(buffer, sizeof(attrlist_t));

        if (flags & ATTR_DONTFOLLOW)
                length = llistxattr(path, lbuf, sizeof(lbuf));
        else
                length = listxattr(path, lbuf, sizeof(lbuf));
        if (length <= 0)
                return length;

        start_offset = sizeof(attrlist_t);
        end_offset = buffersize & ~(8-1);       /* 8 byte align */

        for (l = lbuf; l != lbuf + length; l = strchr(l, '\0') + 1) {
                if (api_unconvert(name, l, flags))
                        continue;
                if (flags & ATTR_DONTFOLLOW)
                        vlength = lgetxattr(path, l, NULL, 0);
                else
                        vlength =  getxattr(path, l, NULL, 0);
                if (vlength < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                if (count++ < cursor->opaque[0])
                        continue;
                if (attr_list_pack(name, vlength, buffer, buffersize,
                                   &start_offset, &end_offset)) {
                        cursor->opaque[0] = count;
                        break;
                }
        }
        return 0;
}

int
attr_listf(int fd, char *buffer, const int buffersize, int flags,
           attrlist_cursor_t *cursor)
{
        const char *l;
        int length, vlength, count = 0;
        char lbuf[MAXLISTLEN];
        char name[MAXNAMELEN+16];
        int start_offset, end_offset;

        if (buffersize < sizeof(attrlist_t)) {
                errno = EINVAL;
```

```
                return -1;
        }
        bzero(buffer, sizeof(attrlist_t));

        length = flistxattr(fd, lbuf, sizeof(lbuf));
        if (length < 0)
                return length;

        start_offset = sizeof(attrlist_t);
        end_offset = buffersize & ~(8-1);       /* 8 byte align */

        for (l = lbuf; l != lbuf + length; l = strchr(l, '\0') + 1) {
                if (api_unconvert(name, l, flags))
                        continue;
                vlength = fgetxattr(fd, l, NULL, 0);
                if (vlength < 0 && (errno == ENOATTR || errno == ENOTSUP))
                        continue;
                if (count++ < cursor->opaque[0])
                        continue;
                if (attr_list_pack(name, vlength, buffer, buffersize,
                                &start_offset, &end_offset)) {
                        cursor->opaque[0] = count;
                        break;
                }
        }
        return 0;
}


/*
 * Helper routines for the attr_multi functions.  In IRIX, the
 * multi routines are a single syscall - in Linux, we break em
 * apart in userspace and make individual syscalls for each.
 */

static int
attr_single(const char *path, attr_multiop_t *op, int flags)
{
        int r = -1;

        errno = -EINVAL;
        flags |= op->am_flags;
        if (op->am_opcode & ATTR_OP_GET)
                r = attr_get(path, op->am_attrname, op->am_attrvalue,
                             &op->am_length, flags);
        else if (op->am_opcode & ATTR_OP_SET)
                r = attr_set(path, op->am_attrname, op->am_attrvalue,
                             op->am_length, flags);
        else if (op->am_opcode & ATTR_OP_REMOVE)
                r = attr_remove(path, op->am_attrname, flags);
        return r;
}

static int
attr_singlef(const int fd, attr_multiop_t *op, int flags)
{
        int r = -1;

        errno = -EINVAL;
        flags |= op->am_flags;
        if (op->am_opcode & ATTR_OP_GET)
                r = attr_getf(fd, op->am_attrname, op->am_attrvalue,
                              &op->am_length, flags);
        else if (op->am_opcode & ATTR_OP_SET)
                r = attr_setf(fd, op->am_attrname, op->am_attrvalue,
                              op->am_length, flags);
        else if (op->am_opcode & ATTR_OP_REMOVE)
                r = attr_removef(fd, op->am_attrname, flags);
        return r;
}

/*
 * Operate on multiple attributes of the same object simultaneously
 *
 * From the manpage: "attr_multi will fail if ... a bit other than
 * ATTR_DONTFOLLOW was set in the flag argument." flags must be
 * checked here as they are not passed into the kernel.
 */
int
attr_multi(const char *path, attr_multiop_t *multiops, int count, int flags)
{
        int i, tmp, r = -1;

        errno = EINVAL;
        if ((flags & ATTR_DONTFOLLOW) != flags)
                return r;

        r = errno = 0;
        for (i = 0; i < count; i++) {
                tmp = attr_single(path, &multiops[i], flags);
                if (tmp) r = tmp;
        }
        return r;
}

int
attr_multif(int fd, attr_multiop_t *multiops, int count, int flags)
{
```

```
        int i, tmp, r = -1;

        errno = EINVAL;
        if ((flags & ATTR_DONTFOLLOW) != flags)
                return r;

        r = errno = 0;
        for (i = 0; i < count; i++) {
                tmp = attr_singlef(fd, &multiops[i], flags);
                if (tmp) r = tmp;
        }
        return r;
}
```

Notice for package(s)

ncurses

```
/****************************************************************************
 * Copyright (c) 1999-2004,2005 Free Software Foundation, Inc.              *
 *                                                                          *
 * Permission is hereby granted, free of charge, to any person obtaining a  *
 * copy of this software and associated documentation files (the            *
 * "Software"), to deal in the Software without restriction, including      *
 * without limitation the rights to use, copy, modify, merge, publish,      *
 * distribute, distribute with modifications, sublicense, and/or sell       *
 * copies of the Software, and to permit persons to whom the Software is    *
 * furnished to do so, subject to the following conditions:                 *
 *                                                                          *
 * The above copyright notice and this permission notice shall be included  *
 * in all copies or substantial portions of the Software.                   *
 *                                                                          *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  *
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF               *
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.   *
 * IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,   *
 * DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR    *
 * OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR    *
 * THE USE OR OTHER DEALINGS IN THE SOFTWARE.                               *
 *                                                                          *
 * Except as contained in this notice, the name(s) of the above copyright   *
 * holders shall not be used in advertising or otherwise to promote the     *
 * sale, use or other dealings in this Software without prior written       *
 * authorization.                                                           *
 ****************************************************************************/

/****************************************************************************
 *  Author: Thomas E. Dickey <dickey@clark.net> 1999                        *
 ****************************************************************************/

#include <curses.priv.h>

MODULE_ID("$Id: version.c,v 1.6 2005/01/02 01:23:54 tom Exp $")

NCURSES_EXPORT(const char *)
curses_version(void)
{
    T((T_CALLED("curses_version()")));
    returnCPtr("ncurses " NCURSES_VERSION_STRING);
}
```

Notice for package(s)

elfutils

```
   This file describes the limits of the Exception under which you are allowed
   to distribute Non-GPL Code in linked combination with Red Hat elfutils.
   For the full text of the license, please see one of the header files
   included with the source distribution or the file COPYING found in the
   top level directory of the source.

   The Approved Interfaces are the functions declared in the files:

   libelf.h
   libdw.h
   libdwfl.h
```

Notice for package(s)

bzip2

```
--------------------------------------------------------------------------

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2010 Julian R Seward.  All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
   not claim that you wrote the original software.  If you use this
   software in a product, an acknowledgment in the product
   documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
   not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org
bzip2/libbzip2 version 1.0.6 of 6 September 2010

--------------------------------------------------------------------------
```

Notice for package(s)

libcheck

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
        51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations
below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
```

you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

  Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

  When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

  We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

  For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

  Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control

compilation and installation of the library.

  Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

  You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not

compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

  It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

  7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined

library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

    b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

  11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing

and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James
  Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice
```

That's all there is to it!

Notice for package(s)

acl

Most components of the "acl" package are licensed under
Version 2.1 of the GNU Lesser General Public License (see COPYING.LGPL).

Some components (as annotated in the source) are licensed
under Version 2 of the GNU General Public License (see below),

---

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
```

```
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License along
    with this program; if not, write to the Free Software Foundation, Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
```

Notice for package(s)

acl

```
Most components of the "acl" package are licensed under
Version 2.1 of the GNU Lesser General Public License (see below).
below.

Some components (as annotated in the source) are licensed
under Version 2 of the GNU General Public License (see COPYING).

------------------------------------------------------------------------

                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
```

distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is

included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a

medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot

use both them and the Library together in an executable that you
distribute.

  7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work
    based on the Library, uncombined with any other library
    facilities. This must be distributed under the terms of the
    Sections above.

    b) Give prominent notice with the combined library of the fact
    that part of it is a work based on the Library, and explaining
    where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

  11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,

write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

                            NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!

Notice for package(s)

db

```
/*-
 * $Id$
 */

The following is the license that applies to this copy of the Berkeley DB
software.  For a license to use the Berkeley DB software under conditions
other than those described here, or to purchase support for this software,
please contact Oracle at berkeleydb-info_us@oracle.com.
```

If you were looking for the license that applies to Berkeley DB XML, click here.
http://www.oracle.com/technetwork/products/berkeleydb/downloads/xmloslicense-086890.html

If you were looking for the license that applies to Berkeley DB Java Edition, click here.
http://www.oracle.com/technetwork/products/berkeleydb/downloads/jeoslicense-086837.html

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
/*
 * The Oracle Berkeley DB product is licensed
 * under the GNU AFFERO GENERAL PUBLIC LICENSE:
 *
 *
 *                    GNU AFFERO GENERAL PUBLIC LICENSE
 *                       Version 3, 19 November 2007
 *
 * Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 * Everyone is permitted to copy and distribute verbatim copies
 * of this license document, but changing it is not allowed.
 *
 *                            Preamble
 *
 *  The GNU Affero General Public License is a free, copyleft license for
 * software and other kinds of works, specifically designed to ensure
 * cooperation with the community in the case of network server software.
 *
 *  The licenses for most software and other practical works are designed
 * to take away your freedom to share and change the works.  By contrast,
 * our General Public Licenses are intended to guarantee your freedom to
 * share and change all versions of a program--to make sure it remains free
 * software for all its users.
 *
 *  When we speak of free software, we are referring to freedom, not
 * price.  Our General Public Licenses are designed to make sure that you
 * have the freedom to distribute copies of free software (and charge for
 * them if you wish), that you receive source code or can get it if you
 * want it, that you can change the software or use pieces of it in new
 * free programs, and that you know you can do these things.
 *
 *  Developers that use our General Public Licenses protect your rights
 * with two steps: (1) assert copyright on the software, and (2) offer
 * you this License which gives you legal permission to copy, distribute
 * and/or modify the software.
 *
 *  A secondary benefit of defending all users' freedom is that
 * improvements made in alternate versions of the program, if they
 * receive widespread use, become available for other developers to
 * incorporate.  Many developers of free software are heartened and
 * encouraged by the resulting cooperation.  However, in the case of
 * software used on network servers, this result may fail to come about.
 * The GNU General Public License permits making a modified version and
 * letting the public access it on a server without ever releasing its
 * source code to the public.
 *
 *  The GNU Affero General Public License is designed specifically to
 * ensure that, in such cases, the modified source code becomes available
 * to the community.  It requires the operator of a network server to
 * provide the source code of the modified version running there to the
 * users of that server.  Therefore, public use of a modified version, on
 * a publicly accessible server, gives the public access to the source
 * code of the modified version.
 *
 *  An older license, called the Affero General Public License and
 * published by Affero, was designed to accomplish similar goals.  This is
 * a different license, not a version of the Affero GPL, but Affero has
 * released a new version of the Affero GPL which permits relicensing under
 * this license.
 *
 *  The precise terms and conditions for copying, distribution and
 * modification follow.
 *
 *                       TERMS AND CONDITIONS
 *
 *  0. Definitions.
 *
 *  "This License" refers to version 3 of the GNU Affero General Public License.
 *
 *  "Copyright" also means copyright-like laws that apply to other kinds of
 * works, such as semiconductor masks.
 *
 *  "The Program" refers to any copyrightable work licensed under this
 * License.  Each licensee is addressed as "you".  "Licensees" and
 * "recipients" may be individuals or organizations.
 *
 *  To "modify" a work means to copy from or adapt all or part of the work
 * in a fashion requiring copyright permission, other than the making of an
 * exact copy.  The resulting work is called a "modified version" of the
 * earlier work or a work "based on" the earlier work.
 *
 *  A "covered work" means either the unmodified Program or a work based
 * on the Program.
 *
 *  To "propagate" a work means to do anything with it that, without
 * permission, would make you directly or secondarily liable for
 * infringement under applicable copyright law, except executing it on a
 * computer or modifying a private copy.  Propagation includes copying,
 * distribution (with or without modification), making available to the
 * public, and in some countries other activities as well.
 *
```

```
*  To "convey" a work means any kind of propagation that enables other
* parties to make or receive copies.  Mere interaction with a user through
* a computer network, with no transfer of a copy, is not conveying.
*
*  An interactive user interface displays "Appropriate Legal Notices"
* to the extent that it includes a convenient and prominently visible
* feature that (1) displays an appropriate copyright notice, and (2)
* tells the user that there is no warranty for the work (except to the
* extent that warranties are provided), that licensees may convey the
* work under this License, and how to view a copy of this License.  If
* the interface presents a list of user commands or options, such as a
* menu, a prominent item in the list meets this criterion.
*
*  1. Source Code.
*
*  The "source code" for a work means the preferred form of the work
* for making modifications to it.  "Object code" means any non-source
* form of a work.
*
*  A "Standard Interface" means an interface that either is an official
* standard defined by a recognized standards body, or, in the case of
* interfaces specified for a particular programming language, one that
* is widely used among developers working in that language.
*
*  The "System Libraries" of an executable work include anything, other
* than the work as a whole, that (a) is included in the normal form of
* packaging a Major Component, but which is not part of that Major
* Component, and (b) serves only to enable use of the work with that
* Major Component, or to implement a Standard Interface for which an
* implementation is available to the public in source code form.  A
* "Major Component", in this context, means a major essential component
* (kernel, window system, and so on) of the specific operating system
* (if any) on which the executable work runs, or a compiler used to
* produce the work, or an object code interpreter used to run it.
*
*  The "Corresponding Source" for a work in object code form means all
* the source code needed to generate, install, and (for an executable
* work) run the object code and to modify the work, including scripts to
* control those activities.  However, it does not include the work's
* System Libraries, or general-purpose tools or generally available free
* programs which are used unmodified in performing those activities but
* which are not part of the work.  For example, Corresponding Source
* includes interface definition files associated with source files for
* the work, and the source code for shared libraries and dynamically
* linked subprograms that the work is specifically designed to require,
* such as by intimate data communication or control flow between those
* subprograms and other parts of the work.
*
*  The Corresponding Source need not include anything that users
* can regenerate automatically from other parts of the Corresponding
* Source.
*
*  The Corresponding Source for a work in source code form is that
* same work.
*
*  2. Basic Permissions.
*
*  All rights granted under this License are granted for the term of
* copyright on the Program, and are irrevocable provided the stated
* conditions are met.  This License explicitly affirms your unlimited
* permission to run the unmodified Program.  The output from running a
* covered work is covered by this License only if the output, given its
* content, constitutes a covered work.  This License acknowledges your
* rights of fair use or other equivalent, as provided by copyright law.
*
* You may make, run and propagate covered works that you do not
* convey, without conditions so long as your license otherwise remains
* in force.  You may convey covered works to others for the sole purpose
* of having them make modifications exclusively for you, or provide you
* with facilities for running those works, provided that you comply with
* the terms of this License in conveying all material for which you do
* not control copyright.  Those thus making or running the covered works
* for you must do so exclusively on your behalf, under your direction
* and control, on terms that prohibit them from making any copies of
* your copyrighted material outside their relationship with you.
*
*   Conveying under any other circumstances is permitted solely under
* the conditions stated below.  Sublicensing is not allowed; section 10
* makes it unnecessary.
*
*   3. Protecting Users' Legal Rights From Anti-Circumvention Law.
*
*   No covered work shall be deemed part of an effective technological
* measure under any applicable law fulfilling obligations under article
* 11 of the WIPO copyright treaty adopted on 20 December 1996, or
* similar laws prohibiting or restricting circumvention of such
* measures.
*
*   When you convey a covered work, you waive any legal power to forbid
* circumvention of technological measures to the extent such circumvention
* is effected by exercising rights under this License with respect to
* the covered work, and you disclaim any intention to limit operation or
* modification of the work as a means of enforcing, against the work's
* users, your or third parties' legal rights to forbid circumvention of
* technological measures.
*
*   4. Conveying Verbatim Copies.
*
```

```
*   You may convey verbatim copies of the Program's source code as you
* receive it, in any medium, provided that you conspicuously and
* appropriately publish on each copy an appropriate copyright notice;
* keep intact all notices stating that this License and any
* non-permissive terms added in accord with section 7 apply to the code;
* keep intact all notices of the absence of any warranty; and give all
* recipients a copy of this License along with the Program.
*
*   You may charge any price or no price for each copy that you convey,
* and you may offer support or warranty protection for a fee.
*
*   5. Conveying Modified Source Versions.
*
*   You may convey a work based on the Program, or the modifications to
* produce it from the Program, in the form of source code under the
* terms of section 4, provided that you also meet all of these conditions:
*
*     a) The work must carry prominent notices stating that you modified
*     it, and giving a relevant date.
*
*     b) The work must carry prominent notices stating that it is
*     released under this License and any conditions added under section
*     7.  This requirement modifies the requirement in section 4 to
*     "keep intact all notices".
*
*     c) You must license the entire work, as a whole, under this
*     License to anyone who comes into possession of a copy.  This
*     License will therefore apply, along with any applicable section 7
*     additional terms, to the whole of the work, and all its parts,
*     regardless of how they are packaged.  This License gives no
*     permission to license the work in any other way, but it does not
*     invalidate such permission if you have separately received it.
*
*     d) If the work has interactive user interfaces, each must display
*     Appropriate Legal Notices; however, if the Program has interactive
*     interfaces that do not display Appropriate Legal Notices, your
*     work need not make them do so.
*
*  A compilation of a covered work with other separate and independent
* works, which are not by their nature extensions of the covered work,
* and which are not combined with it such as to form a larger program,
* in or on a volume of a storage or distribution medium, is called an
* "aggregate" if the compilation and its resulting copyright are not
* used to limit the access or legal rights of the compilation's users
* beyond what the individual works permit.  Inclusion of a covered work
* in an aggregate does not cause this License to apply to the other
* parts of the aggregate.
*
*   6. Conveying Non-Source Forms.
*
*   You may convey a covered work in object code form under the terms
* of sections 4 and 5, provided that you also convey the
* machine-readable Corresponding Source under the terms of this License,
* in one of these ways:
*
*     a) Convey the object code in, or embodied in, a physical product
*     (including a physical distribution medium), accompanied by the
*     Corresponding Source fixed on a durable physical medium
*     customarily used for software interchange.
*
*     b) Convey the object code in, or embodied in, a physical product
*     (including a physical distribution medium), accompanied by a
*     written offer, valid for at least three years and valid for as
*     long as you offer spare parts or customer support for that product
*     model, to give anyone who possesses the object code either (1) a
*     copy of the Corresponding Source for all the software in the
*     product that is covered by this License, on a durable physical
*     medium customarily used for software interchange, for a price no
*     more than your reasonable cost of physically performing this
*     conveying of source, or (2) access to copy the
*     Corresponding Source from a network server at no charge.
*
*     c) Convey individual copies of the object code with a copy of the
*     written offer to provide the Corresponding Source.  This
*     alternative is allowed only occasionally and noncommercially, and
*     only if you received the object code with such an offer, in accord
*     with subsection 6b.
*
*     d) Convey the object code by offering access from a designated
*     place (gratis or for a charge), and offer equivalent access to the
*     Corresponding Source in the same way through the same place at no
*     further charge.  You need not require recipients to copy the
*     Corresponding Source along with the object code.  If the place to
*     copy the object code is a network server, the Corresponding Source
*     may be on a different server (operated by you or a third party)
*     that supports equivalent copying facilities, provided you maintain
*     clear directions next to the object code saying where to find the
*     Corresponding Source.  Regardless of what server hosts the
*     Corresponding Source, you remain obligated to ensure that it is
*     available for as long as needed to satisfy these requirements.
*
*     e) Convey the object code using peer-to-peer transmission, provided
*     you inform other peers where the object code and Corresponding
*     Source of the work are being offered to the general public at no
*     charge under subsection 6d.
*
*   A separable portion of the object code, whose source code is excluded
* from the Corresponding Source as a System Library, need not be
```

* included in conveying the object code work.
*
*   A "User Product" is either (1) a "consumer product", which means any
* tangible personal property which is normally used for personal, family,
* or household purposes, or (2) anything designed or sold for incorporation
* into a dwelling. In determining whether a product is a consumer product,
* doubtful cases shall be resolved in favor of coverage. For a particular
* product received by a particular user, "normally used" refers to a
* typical or common use of that class of product, regardless of the status
* of the particular user or of the way in which the particular user
* actually uses, or expects or is expected to use, the product. A product
* is a consumer product regardless of whether the product has substantial
* commercial, industrial or non-consumer uses, unless such uses represent
* the only significant mode of use of the product.
*
*   "Installation Information" for a User Product means any methods,
* procedures, authorization keys, or other information required to install
* and execute modified versions of a covered work in that User Product from
* a modified version of its Corresponding Source. The information must
* suffice to ensure that the continued functioning of the modified object
* code is in no case prevented or interfered with solely because
* modification has been made.
*
*   If you convey an object code work under this section in, or with, or
* specifically for use in, a User Product, and the conveying occurs as
* part of a transaction in which the right of possession and use of the
* User Product is transferred to the recipient in perpetuity or for a
* fixed term (regardless of how the transaction is characterized), the
* Corresponding Source conveyed under this section must be accompanied
* by the Installation Information. But this requirement does not apply
* if neither you nor any third party retains the ability to install
* modified object code on the User Product (for example, the work has
* been installed in ROM).
*
*   The requirement to provide Installation Information does not include a
* requirement to continue to provide support service, warranty, or updates
* for a work that has been modified or installed by the recipient, or for
* the User Product in which it has been modified or installed. Access to a
* network may be denied when the modification itself materially and
* adversely affects the operation of the network or violates the rules and
* protocols for communication across the network.
*
*   Corresponding Source conveyed, and Installation Information provided,
* in accord with this section must be in a format that is publicly
* documented (and with an implementation available to the public in
* source code form), and must require no special password or key for
* unpacking, reading or copying.
*
*   7. Additional Terms.
*
*   "Additional permissions" are terms that supplement the terms of this
* License by making exceptions from one or more of its conditions.
* Additional permissions that are applicable to the entire Program shall
* be treated as though they were included in this License, to the extent
* that they are valid under applicable law. If additional permissions
* apply only to part of the Program, that part may be used separately
* under those permissions, but the entire Program remains governed by
* this License without regard to the additional permissions.
*
*   When you convey a copy of a covered work, you may at your option
* remove any additional permissions from that copy, or from any part of
* it. (Additional permissions may be written to require their own
* removal in certain cases when you modify the work.) You may place
* additional permissions on material, added by you to a covered work,
* for which you have or can give appropriate copyright permission.
*
*   Notwithstanding any other provision of this License, for material you
* add to a covered work, you may (if authorized by the copyright holders of
* that material) supplement the terms of this License with terms:
*
*   a) Disclaiming warranty or limiting liability differently from the
*   terms of sections 15 and 16 of this License; or
*
*   b) Requiring preservation of specified reasonable legal notices or
*   author attributions in that material or in the Appropriate Legal
*   Notices displayed by works containing it; or
*
*   c) Prohibiting misrepresentation of the origin of that material, or
*   requiring that modified versions of such material be marked in
*   reasonable ways as different from the original version; or
*
*   d) Limiting the use for publicity purposes of names of licensors or
*   authors of the material; or
*
*   e) Declining to grant rights under trademark law for use of some
*   trade names, trademarks, or service marks; or
*
*   f) Requiring indemnification of licensors and authors of that
*   material by anyone who conveys the material (or modified versions of
*   it) with contractual assumptions of liability to the recipient, for
*   any liability that these contractual assumptions directly impose on
*   those licensors and authors.
*
*   All other non-permissive additional terms are considered "further
* restrictions" within the meaning of section 10. If the Program as you
* received it, or any part of it, contains a notice stating that it is
* governed by this License along with a term that is a further
* restriction, you may remove that term. If a license document contains

```
* a further restriction but permits relicensing or conveying under this
* License, you may add to a covered work material governed by the terms
* of that license document, provided that the further restriction does
* not survive such relicensing or conveying.
*
*   If you add terms to a covered work in accord with this section, you
* must place, in the relevant source files, a statement of the
* additional terms that apply to those files, or a notice indicating
* where to find the applicable terms.
*
*   Additional terms, permissive or non-permissive, may be stated in the
* form of a separately written license, or stated as exceptions;
* the above requirements apply either way.
*
*   8. Termination.
*
*   You may not propagate or modify a covered work except as expressly
* provided under this License.  Any attempt otherwise to propagate or
* modify it is void, and will automatically terminate your rights under
* this License (including any patent licenses granted under the third
* paragraph of section 11).
*
*   However, if you cease all violation of this License, then your
* license from a particular copyright holder is reinstated (a)
* provisionally, unless and until the copyright holder explicitly and
* finally terminates your license, and (b) permanently, if the copyright
* holder fails to notify you of the violation by some reasonable means
* prior to 60 days after the cessation.
*
*   Moreover, your license from a particular copyright holder is
* reinstated permanently if the copyright holder notifies you of the
* violation by some reasonable means, this is the first time you have
* received notice of violation of this License (for any work) from that
* copyright holder, and you cure the violation prior to 30 days after
* your receipt of the notice.
*
*   Termination of your rights under this section does not terminate the
* licenses of parties who have received copies or rights from you under
* this License.  If your rights have been terminated and not permanently
* reinstated, you do not qualify to receive new licenses for the same
* material under section 10.
*
*   9. Acceptance Not Required for Having Copies.
*
*   You are not required to accept this License in order to receive or
* run a copy of the Program.  Ancillary propagation of a covered work
* occurring solely as a consequence of using peer-to-peer transmission
* to receive a copy likewise does not require acceptance.  However,
* nothing other than this License grants you permission to propagate or
* modify any covered work.  These actions infringe copyright if you do
* not accept this License.  Therefore, by modifying or propagating a
* covered work, you indicate your acceptance of this License to do so.
*
*   10. Automatic Licensing of Downstream Recipients.
*
*   Each time you convey a covered work, the recipient automatically
* receives a license from the original licensors, to run, modify and
* propagate that work, subject to this License.  You are not responsible
* for enforcing compliance by third parties with this License.
*
*   An "entity transaction" is a transaction transferring control of an
* organization, or substantially all assets of one, or subdividing an
* organization, or merging organizations.  If propagation of a covered
* work results from an entity transaction, each party to that
* transaction who receives a copy of the work also receives whatever
* licenses to the work the party's predecessor in interest had or could
* give under the previous paragraph, plus a right to possession of the
* Corresponding Source of the work from the predecessor in interest, if
* the predecessor has it or can get it with reasonable efforts.
*
*   You may not impose any further restrictions on the exercise of the
* rights granted or affirmed under this License.  For example, you may
* not impose a license fee, royalty, or other charge for exercise of
* rights granted under this License, and you may not initiate litigation
* (including a cross-claim or counterclaim in a lawsuit) alleging that
* any patent claim is infringed by making, using, selling, offering for
* sale, or importing the Program or any portion of it.
*
*   11. Patents.
*
*   A "contributor" is a copyright holder who authorizes use under this
* License of the Program or a work on which the Program is based.  The
* work thus licensed is called the contributor's "contributor version".
*
*   A contributor's "essential patent claims" are all patent claims
* owned or controlled by the contributor, whether already acquired or
* hereafter acquired, that would be infringed by some manner, permitted
* by this License, of making, using, or selling its contributor version,
* but do not include claims that would be infringed only as a
* consequence of further modification of the contributor version.  For
* purposes of this definition, "control" includes the right to grant
* patent sublicenses in a manner consistent with the requirements of
* this License.
*
*   Each contributor grants you a non-exclusive, worldwide, royalty-free
* patent license under the contributor's essential patent claims, to
* make, use, sell, offer for sale, import and otherwise run, modify and
* propagate the contents of its contributor version.
```

```
*
*   In the following three paragraphs, a "patent license" is any express
* agreement or commitment, however denominated, not to enforce a patent
* (such as an express permission to practice a patent or covenant not to
* sue for patent infringement).  To "grant" such a patent license to a
* party means to make such an agreement or commitment not to enforce a
* patent against the party.
*
*   If you convey a covered work, knowingly relying on a patent license,
* and the Corresponding Source of the work is not available for anyone
* to copy, free of charge and under the terms of this License, through a
* publicly available network server or other readily accessible means,
* then you must either (1) cause the Corresponding Source to be so
* available, or (2) arrange to deprive yourself of the benefit of the
* patent license for this particular work, or (3) arrange, in a manner
* consistent with the requirements of this License, to extend the patent
* license to downstream recipients.  "Knowingly relying" means you have
* actual knowledge that, but for the patent license, your conveying the
* covered work in a country, or your recipient's use of the covered work
* in a country, would infringe one or more identifiable patents in that
* country that you have reason to believe are valid.
*
*   If, pursuant to or in connection with a single transaction or
* arrangement, you convey, or propagate by procuring conveyance of, a
* covered work, and grant a patent license to some of the parties
* receiving the covered work authorizing them to use, propagate, modify
* or convey a specific copy of the covered work, then the patent license
* you grant is automatically extended to all recipients of the covered
* work and works based on it.
*
*   A patent license is "discriminatory" if it does not include within
* the scope of its coverage, prohibits the exercise of, or is
* conditioned on the non-exercise of one or more of the rights that are
* specifically granted under this License.  You may not convey a covered
* work if you are a party to an arrangement with a third party that is
* in the business of distributing software, under which you make payment
* to the third party based on the extent of your activity of conveying
* the work, and under which the third party grants, to any of the
* parties who would receive the covered work from you, a discriminatory
* patent license (a) in connection with copies of the covered work
* conveyed by you (or copies made from those copies), or (b) primarily
* for and in connection with specific products or compilations that
* contain the covered work, unless you entered into that arrangement,
* or that patent license was granted, prior to 28 March 2007.
*
*   Nothing in this License shall be construed as excluding or limiting
* any implied license or other defenses to infringement that may
* otherwise be available to you under applicable patent law.
*
*   12. No Surrender of Others' Freedom.
*
*   If conditions are imposed on you (whether by court order, agreement or
* otherwise) that contradict the conditions of this License, they do not
* excuse you from the conditions of this License.  If you cannot convey a
* covered work so as to satisfy simultaneously your obligations under this
* License and any other pertinent obligations, then as a consequence you may
* not convey it at all.  For example, if you agree to terms that obligate you
* to collect a royalty for further conveying from those to whom you convey
* the Program, the only way you could satisfy both those terms and this
* License would be to refrain entirely from conveying the Program.
*
*   13. Remote Network Interaction; Use with the GNU General Public License.
*
*   Notwithstanding any other provision of this License, if you modify the
* Program, your modified version must prominently offer all users
* interacting with it remotely through a computer network (if your version
* supports such interaction) an opportunity to receive the Corresponding
* Source of your version by providing access to the Corresponding Source
* from a network server at no charge, through some standard or customary
* means of facilitating copying of software.  This Corresponding Source
* shall include the Corresponding Source for any work covered by version 3
* of the GNU General Public License that is incorporated pursuant to the
* following paragraph.
*
*   Notwithstanding any other provision of this License, you have
* permission to link or combine any covered work with a work licensed
* under version 3 of the GNU General Public License into a single
* combined work, and to convey the resulting work.  The terms of this
* License will continue to apply to the part which is the covered work,
* but the work with which it is combined will remain governed by version
* 3 of the GNU General Public License.
*
*   14. Revised Versions of this License.
*
*   The Free Software Foundation may publish revised and/or new versions of
* the GNU Affero General Public License from time to time.  Such new versions
* will be similar in spirit to the present version, but may differ in detail to
* address new problems or concerns.
*
*   Each version is given a distinguishing version number.  If the
* Program specifies that a certain numbered version of the GNU Affero General
* Public License "or any later version" applies to it, you have the
* option of following the terms and conditions either of that numbered
* version or of any later version published by the Free Software
* Foundation.  If the Program does not specify a version number of the
* GNU Affero General Public License, you may choose any version ever published
* by the Free Software Foundation.
*
```

```
 *   If the Program specifies that a proxy can decide which future
 * versions of the GNU Affero General Public License can be used, that proxy's
 * public statement of acceptance of a version permanently authorizes you
 * to choose that version for the Program.
 *
 *   Later license versions may give you additional or different
 * permissions.  However, no additional obligations are imposed on any
 * author or copyright holder as a result of your choosing to follow a
 * later version.
 *
 *   15. Disclaimer of Warranty.
 *
 *   THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
 * APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
 * HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
 * OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
 * IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
 * ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
 *
 *   16. Limitation of Liability.
 *
 *   IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
 * WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
 * THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
 * GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
 * USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
 * DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
 * PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
 * EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGES.
 *
 *   17. Interpretation of Sections 15 and 16.
 *
 *   If the disclaimer of warranty and limitation of liability provided
 * above cannot be given local legal effect according to their terms,
 * reviewing courts shall apply local law that most closely approximates
 * an absolute waiver of all civil liability in connection with the
 * Program, unless a warranty or assumption of liability accompanies a
 * copy of the Program in return for a fee.
 *
 *                      END OF TERMS AND CONDITIONS
 *
 *             How to Apply These Terms to Your New Programs
 *
 *   If you develop a new program, and you want it to be of the greatest
 * possible use to the public, the best way to achieve this is to make it
 * free software which everyone can redistribute and change under these terms.
 *
 *   To do so, attach the following notices to the program.  It is safest
 * to attach them to the start of each source file to most effectively
 * state the exclusion of warranty; and each file should have at least
 * the "copyright" line and a pointer to where the full notice is found.
 *
 *     <one line to give the program's name and a brief idea of what it does.>
 *     Copyright (C) <year>  <name of author>
 *
 *     This program is free software: you can redistribute it and/or modify
 *   it under the terms of the GNU Affero General Public License as published by
 *   the Free Software Foundation, either version 3 of the License, or
 *   (at your option) any later version.
 *
 *     This program is distributed in the hope that it will be useful,
 *     but WITHOUT ANY WARRANTY; without even the implied warranty of
 *     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *     GNU Affero General Public License for more details.
 *
 *     You should have received a copy of the GNU Affero General Public License
 *     along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 * Also add information on how to contact you by electronic and paper mail.
 *
 *   If your software can interact with users remotely through a computer
 * network, you should also make sure that it provides a way for users to
 * get its source.  For example, if your program is a web application, its
 * interface could display a "Source" link that leads users to an archive
 * of the code.  There are many ways you could offer source, and different
 * solutions will be better for different programs; see section 13 for the
 * specific requirements.
 *
 *   You should also get your employer (if you work as a programmer) or school,
 * if any, to sign a "copyright disclaimer" for the program, if necessary.
 * For more information on this, and how to apply and follow the GNU AGPL, see
 * <http://www.gnu.org/licenses/>.
 */

ADDITIONAL THIRD PARTY NOTICES:

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
/*
 * Copyright (c) 1990, 1993, 1994, 1995
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
```

```
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
/*
 * Copyright (c) 1995, 1996
 *      The President and Fellows of Harvard University.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
/***
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2005 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
```

Notice for package(s)

db

```
                    GNU AFFERO GENERAL PUBLIC LICENSE
                       Version 3, 19 November 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble
```

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official

standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object

code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third

paragraph of section 11).

  However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

  Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

  If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that

country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

  A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

  13. Remote Network Interaction; Use with the GNU General Public License.

  Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

  Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

  Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

  Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If your software can interact with users remotely through a computer
network, you should also make sure that it provides a way for users to
get its source. For example, if your program is a web application, its
interface could display a "Source" link that leads users to an archive
of the code. There are many ways you could offer source, and different
solutions will be better for different programs; see section 13 for the
specific requirements.

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU AGPL, see
<http://www.gnu.org/licenses/>.

Notice for package(s)

libusb1

                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the

Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

  To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

  Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

  When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

  We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

  For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

  Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or

other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in

these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked
    with the Library, with the complete machine-readable "work that
    uses the Library", as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library. (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these

materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new

versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

                            NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!

Notice for package(s)

xz

```
XZ Utils Licensing
==================

    Different licenses apply to different files in this package. Here
    is a rough summary of which licenses apply to which parts of this
    package (but check the individual files to be sure!):

      - liblzma is in the public domain.

      - xz, xzdec, and lzmadec command line tools are in the public
        domain unless GNU getopt_long had to be compiled and linked
        in from the lib directory. The getopt_long code is under
        GNU LGPLv2.1+.

      - The scripts to grep, diff, and view compressed files have been
        adapted from gzip. These scripts and their documentation are
        under GNU GPLv2+.

      - All the documentation in the doc directory and most of the
        XZ Utils specific documentation files in other directories
        are in the public domain.

      - Translated messages are in the public domain.

      - The build system contains public domain files, and files that
        are under GNU GPLv2+ or GNU GPLv3+. None of these files end up
        in the binaries being built.

      - Test files and test code in the tests directory, and debugging
        utilities in the debug directory are in the public domain.

      - The extra directory may contain public domain files, and files
        that are under various free software licenses.

    You can do whatever you want with the files that have been put into
    the public domain. If you find public domain legally problematic,
    take the previous sentence as a license grant. If you still find
    the lack of copyright legally problematic, you have too many
    lawyers.

    As usual, this software is provided "as is", without any warranty.

    If you copy significant amounts of public domain code from XZ Utils
    into your project, acknowledging this somewhere in your software is
    polite (especially if it is proprietary, non-free software), but
    naturally it is not legally required. Here is an example of a good
    notice to put into "about box" or into documentation:

        This software includes code from XZ Utils <http://tukaani.org/xz/>.

    The following license texts are included in the following files:
      - COPYING.LGPLv2.1: GNU Lesser General Public License version 2.1
      - COPYING.GPLv2: GNU General Public License version 2
      - COPYING.GPLv3: GNU General Public License version 3

    Note that the toolchain (compiler, linker etc.) may add some code
    pieces that are copyrighted. Thus, it is possible that e.g. liblzma
    binary wouldn't actually be in the public domain in its entirety
    even though it contains no copyrighted code from the XZ Utils source
    package.

    If you have questions, don't hesitate to ask the author(s) for more
    information.
```

Notice for package(s)

xz

```
/* Getopt for GNU.
   NOTE: getopt is now part of the C library, so if you don't know what
   "Keep this file name-space clean" means, talk to drepper@gnu.org
   before changing it!
   Copyright (C) 1987,88,89,90,91,92,93,94,95,96,98,99,2000,2001,2002,2003,2004,2006
        Free Software Foundation, Inc.
   This file is part of the GNU C Library.

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU Lesser General Public License as published by
   the Free Software Foundation; either version 2.1, or (at your option)
   any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU Lesser General Public License for more details.
```

```
   You should have received a copy of the GNU Lesser General Public License along
   with this program; if not, write to the Free Software Foundation,
   Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.  */

#ifndef _LIBC
# include <config.h>
#endif

#include "getopt.h"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#ifdef __VMS
# include <unixlib.h>
#endif

/* Completely disable NLS for getopt. We won't include translations for it
   anyway. If the system lacks getopt_long, missing translations probably
   aren't a problem.  */
/*
#ifdef _LIBC
# include <libintl.h>
#else
# include "gettext.h"
# define _(msgid) gettext (msgid)
#endif
*/
#define _(msgid) (msgid)

#if defined _LIBC && defined USE_IN_LIBIO
# include <wchar.h>
#endif

#ifndef attribute_hidden
# define attribute_hidden
#endif

/* Unlike standard Unix `getopt', functions like `getopt_long'
   let the user intersperse the options with the other arguments.

   As `getopt_long' works, it permutes the elements of ARGV so that,
   when it is done, all the options precede everything else.  Thus
   all application programs are extended to handle flexible argument order.

   Using `getopt' or setting the environment variable POSIXLY_CORRECT
   disables permutation.
   Then the application's behavior is completely standard.

   GNU application programs can use a third alternative mode in which
   they can distinguish the relative order of options and other arguments.  */

#include "getopt_int.h"

/* For communication from `getopt' to the caller.
   When `getopt' finds an option that takes an argument,
   the argument value is returned here.
   Also, when `ordering' is RETURN_IN_ORDER,
   each non-option ARGV-element is returned here.  */

char *optarg;

/* Index in ARGV of the next element to be scanned.
   This is used for communication to and from the caller
   and for communication between successive calls to `getopt'.

   On entry to `getopt', zero means this is the first call; initialize.

   When `getopt' returns -1, this is the index of the first of the
   non-option elements that the caller should itself scan.

   Otherwise, `optind' communicates from one call to the next
   how much of ARGV has been scanned so far.  */

/* 1003.2 says this must be 1 before any call.  */
int optind = 1;

/* Callers store zero here to inhibit the error message
   for unrecognized options.  */

int opterr = 1;

/* Set to an option character which was unrecognized.
   This must be initialized on some systems to avoid linking in the
   system's own getopt implementation.  */

int optopt = '?';

/* Keep a global copy of all internal members of getopt_data.  */

static struct _getopt_data getopt_data;


#if defined HAVE_DECL_GETENV && !HAVE_DECL_GETENV
extern char *getenv ();
#endif
```

```
#ifdef _LIBC
/* Stored original parameters.
   XXX This is no good solution.  We should rather copy the args so
   that we can compare them later.  But we must not use malloc(3).  */
extern int __libc_argc;
extern char **__libc_argv;

/* Bash 2.0 gives us an environment variable containing flags
   indicating ARGV elements that should not be considered arguments.  */

# ifdef USE_NONOPTION_FLAGS
/* Defined in getopt_init.c  */
extern char *__getopt_nonoption_flags;
# endif

# ifdef USE_NONOPTION_FLAGS
#  define SWAP_FLAGS(ch1, ch2) \
  if (d->__nonoption_flags_len > 0)                                      \
    {                                                                    \
      char __tmp = __getopt_nonoption_flags[ch1];                        \
      __getopt_nonoption_flags[ch1] = __getopt_nonoption_flags[ch2];     \
      __getopt_nonoption_flags[ch2] = __tmp;                             \
    }
# else
#  define SWAP_FLAGS(ch1, ch2)
# endif
#else   /* !_LIBC */
# define SWAP_FLAGS(ch1, ch2)
#endif  /* _LIBC */

/* Exchange two adjacent subsequences of ARGV.
   One subsequence is elements [first_nonopt,last_nonopt)
   which contains all the non-options that have been skipped so far.
   The other is elements [last_nonopt,optind), which contains all
   the options processed since those non-options were skipped.

   `first_nonopt' and `last_nonopt' are relocated so that they describe
   the new indices of the non-options in ARGV after they are moved.  */

static void
exchange (char **argv, struct _getopt_data *d)
{
  int bottom = d->__first_nonopt;
  int middle = d->__last_nonopt;
  int top = d->optind;
  char *tem;

  /* Exchange the shorter segment with the far end of the longer segment.
     That puts the shorter segment into the right place.
     It leaves the longer segment in the right place overall,
     but it consists of two parts that need to be swapped next.  */

#if defined _LIBC && defined USE_NONOPTION_FLAGS
  /* First make sure the handling of the `__getopt_nonoption_flags'
     string can work normally.  Our top argument must be in the range
     of the string.  */
  if (d->__nonoption_flags_len > 0 && top >= d->__nonoption_flags_max_len)
    {
      /* We must extend the array.  The user plays games with us and
         presents new arguments.  */
      char *new_str = malloc (top + 1);
      if (new_str == NULL)
        d->__nonoption_flags_len = d->__nonoption_flags_max_len = 0;
      else
        {
          memset (__mempcpy (new_str, __getopt_nonoption_flags,
                             d->__nonoption_flags_max_len),
                  '\0', top + 1 - d->__nonoption_flags_max_len);
          d->__nonoption_flags_max_len = top + 1;
          __getopt_nonoption_flags = new_str;
        }
    }
#endif

  while (top > middle && middle > bottom)
    {
      if (top - middle > middle - bottom)
        {
          /* Bottom segment is the short one.  */
          int len = middle - bottom;
          register int i;

          /* Swap it with the top part of the top segment.  */
          for (i = 0; i < len; i++)
            {
              tem = argv[bottom + i];
              argv[bottom + i] = argv[top - (middle - bottom) + i];
              argv[top - (middle - bottom) + i] = tem;
              SWAP_FLAGS (bottom + i, top - (middle - bottom) + i);
            }
          /* Exclude the moved bottom segment from further swapping.  */
          top -= len;
        }
      else
        {
          /* Top segment is the short one.  */
          int len = top - middle;
```

```
          register int i;

          /* Swap it with the bottom part of the bottom segment.  */
          for (i = 0; i < len; i++)
            {
              tem = argv[bottom + i];
              argv[bottom + i] = argv[middle + i];
              argv[middle + i] = tem;
              SWAP_FLAGS (bottom + i, middle + i);
            }
          /* Exclude the moved top segment from further swapping.  */
          bottom += len;
        }
    }

  /* Update records for the slots the non-options now occupy.  */

  d->__first_nonopt += (d->optind - d->__last_nonopt);
  d->__last_nonopt = d->optind;
}

/* Initialize the internal data when the first call is made.  */

static const char *
_getopt_initialize (int argc, char **argv, const char *optstring,
                    int posixly_correct, struct _getopt_data *d)
{
  /* Start processing options with ARGV-element 1 (since ARGV-element 0
     is the program name); the sequence of previously skipped
     non-option ARGV-elements is empty.  */

  d->__first_nonopt = d->__last_nonopt = d->optind;

  d->__nextchar = NULL;

  d->__posixly_correct = posixly_correct || !!getenv ("POSIXLY_CORRECT");

  /* Determine how to handle the ordering of options and nonoptions.  */

  if (optstring[0] == '-')
    {
      d->__ordering = RETURN_IN_ORDER;
      ++optstring;
    }
  else if (optstring[0] == '+')
    {
      d->__ordering = REQUIRE_ORDER;
      ++optstring;
    }
  else if (d->__posixly_correct)
    d->__ordering = REQUIRE_ORDER;
  else
    d->__ordering = PERMUTE;

#if defined _LIBC && defined USE_NONOPTION_FLAGS
  if (!d->__posixly_correct
      && argc == __libc_argc && argv == __libc_argv)
    {
      if (d->__nonoption_flags_max_len == 0)
        {
          if (__getopt_nonoption_flags == NULL
              || __getopt_nonoption_flags[0] == '\0')
            d->__nonoption_flags_max_len = -1;
          else
            {
              const char *orig_str = __getopt_nonoption_flags;
              int len = d->__nonoption_flags_max_len = strlen (orig_str);
              if (d->__nonoption_flags_max_len < argc)
                d->__nonoption_flags_max_len = argc;
              __getopt_nonoption_flags =
                (char *) malloc (d->__nonoption_flags_max_len);
              if (__getopt_nonoption_flags == NULL)
                d->__nonoption_flags_max_len = -1;
              else
                memset (__mempcpy (__getopt_nonoption_flags, orig_str, len),
                        '\0', d->__nonoption_flags_max_len - len);
            }
        }
      d->__nonoption_flags_len = d->__nonoption_flags_max_len;
    }
  else
    d->__nonoption_flags_len = 0;
#endif

  return optstring;
}

/* Scan elements of ARGV (whose length is ARGC) for option characters
   given in OPTSTRING.

   If an element of ARGV starts with '-', and is not exactly "-" or "--",
   then it is an option element.  The characters of this element
   (aside from the initial '-') are option characters.  If `getopt'
   is called repeatedly, it returns successively each of the option characters
   from each of the option elements.

   If `getopt' finds another option character, it returns that character,
   updating `optind' and `nextchar' so that the next call to `getopt' can
```

```
   resume the scan with the following option character or ARGV-element.

   If there are no more option characters, `getopt' returns -1.
   Then `optind' is the index in ARGV of the first ARGV-element
   that is not an option.  (The ARGV-elements have been permuted
   so that those that are not options now come last.)

   OPTSTRING is a string containing the legitimate option characters.
   If an option character is seen that is not listed in OPTSTRING,
   return '?' after printing an error message.  If you set `opterr' to
   zero, the error message is suppressed but we still return '?'.

   If a char in OPTSTRING is followed by a colon, that means it wants an arg,
   so the following text in the same ARGV-element, or the text of the following
   ARGV-element, is returned in `optarg'.  Two colons mean an option that
   wants an optional arg; if there is text in the current ARGV-element,
   it is returned in `optarg', otherwise `optarg' is set to zero.

   If OPTSTRING starts with `-' or `+', it requests different methods of
   handling the non-option ARGV-elements.
   See the comments about RETURN_IN_ORDER and REQUIRE_ORDER, above.

   Long-named options begin with `--' instead of `-'.
   Their names may be abbreviated as long as the abbreviation is unique
   or is an exact match for some defined option.  If they have an
   argument, it follows the option name in the same ARGV-element, separated
   from the option name by a `=', or else the in next ARGV-element.
   When `getopt' finds a long-named option, it returns 0 if that option's
   `flag' field is nonzero, the value of the option's `val' field
   if the `flag' field is zero.

   LONGOPTS is a vector of `struct option' terminated by an
   element containing a name which is zero.

   LONGIND returns the index in LONGOPT of the long-named option found.
   It is only valid when a long-named option has been found by the most
   recent call.

   If LONG_ONLY is nonzero, '-' as well as '--' can introduce
   long-named options.

   If POSIXLY_CORRECT is nonzero, behave as if the POSIXLY_CORRECT
   environment variable were set.  */

int
_getopt_internal_r (int argc, char **argv, const char *optstring,
                    const struct option *longopts, int *longind,
                    int long_only, int posixly_correct, struct _getopt_data *d)
{
  int print_errors = d->opterr;
  if (optstring[0] == ':')
    print_errors = 0;

  if (argc < 1)
    return -1;

  d->optarg = NULL;

  if (d->optind == 0 || !d->__initialized)
    {
      if (d->optind == 0)
        d->optind = 1;  /* Don't scan ARGV[0], the program name.  */
      optstring = _getopt_initialize (argc, argv, optstring,
                                      posixly_correct, d);
      d->__initialized = 1;
    }

  /* Test whether ARGV[optind] points to a non-option argument.
     Either it does not have option syntax, or there is an environment flag
     from the shell indicating it is not an option.  The later information
     is only used when the used in the GNU libc.  */
#if defined _LIBC && defined USE_NONOPTION_FLAGS
# define NONOPTION_P (argv[d->optind][0] != '-' || argv[d->optind][1] == '\0' \
                      || (d->optind < d->__nonoption_flags_len                \
                          && __getopt_nonoption_flags[d->optind] == '1'))
#else
# define NONOPTION_P (argv[d->optind][0] != '-' || argv[d->optind][1] == '\0')
#endif

  if (d->__nextchar == NULL || *d->__nextchar == '\0')
    {
      /* Advance to the next ARGV-element.  */

      /* Give FIRST_NONOPT & LAST_NONOPT rational values if OPTIND has been
         moved back by the user (who may also have changed the arguments).  */
      if (d->__last_nonopt > d->optind)
        d->__last_nonopt = d->optind;
      if (d->__first_nonopt > d->optind)
        d->__first_nonopt = d->optind;

      if (d->__ordering == PERMUTE)
        {
          /* If we have just processed some options following some non-options,
             exchange them so that the options come first.  */

          if (d->__first_nonopt != d->__last_nonopt
              && d->__last_nonopt != d->optind)
            exchange ((char **) argv, d);
```

```
        else if (d->__last_nonopt != d->optind)
          d->__first_nonopt = d->optind;

        /* Skip any additional non-options
           and extend the range of non-options previously skipped.  */

        while (d->optind < argc && NONOPTION_P)
          d->optind++;
        d->__last_nonopt = d->optind;
      }

    /* The special ARGV-element `--' means premature end of options.
       Skip it like a null option,
       then exchange with previous non-options as if it were an option,
       then skip everything else like a non-option.  */

    if (d->optind != argc && !strcmp (argv[d->optind], "--"))
      {
        d->optind++;

        if (d->__first_nonopt != d->__last_nonopt
            && d->__last_nonopt != d->optind)
          exchange ((char **) argv, d);
        else if (d->__first_nonopt == d->__last_nonopt)
          d->__first_nonopt = d->optind;
        d->__last_nonopt = argc;

        d->optind = argc;
      }

    /* If we have done all the ARGV-elements, stop the scan
       and back over any non-options that we skipped and permuted.  */

    if (d->optind == argc)
      {
        /* Set the next-arg-index to point at the non-options
           that we previously skipped, so the caller will digest them.  */
        if (d->__first_nonopt != d->__last_nonopt)
          d->optind = d->__first_nonopt;
        return -1;
      }

    /* If we have come to a non-option and did not permute it,
       either stop the scan or describe it to the caller and pass it by.  */

    if (NONOPTION_P)
      {
        if (d->__ordering == REQUIRE_ORDER)
          return -1;
        d->optarg = argv[d->optind++];
        return 1;
      }

    /* We have found another option-ARGV-element.
       Skip the initial punctuation.  */

    d->__nextchar = (argv[d->optind] + 1
                     + (longopts != NULL && argv[d->optind][1] == '-'));
  }

/* Decode the current option-ARGV-element.  */

/* Check whether the ARGV-element is a long option.

   If long_only and the ARGV-element has the form "-f", where f is
   a valid short option, don't consider it an abbreviated form of
   a long option that starts with f.  Otherwise there would be no
   way to give the -f short option.

   On the other hand, if there's a long option "fubar" and
   the ARGV-element is "-fu", do consider that an abbreviation of
   the long option, just like "--fu", and not "-f" with arg "u".

   This distinction seems to be the most useful approach.  */

if (longopts != NULL
    && (argv[d->optind][1] == '-'
        || (long_only && (argv[d->optind][2]
                          || !strchr (optstring, argv[d->optind][1])))))
  {
    char *nameend;
    const struct option *p;
    const struct option *pfound = NULL;
    int exact = 0;
    int ambig = 0;
    int indfound = -1;
    int option_index;

    for (nameend = d->__nextchar; *nameend && *nameend != '='; nameend++)
      /* Do nothing.  */ ;

    /* Test all long options for either exact match
       or abbreviated matches.  */
    for (p = longopts, option_index = 0; p->name; p++, option_index++)
      if (!strncmp (p->name, d->__nextchar, nameend - d->__nextchar))
        {
          if ((unsigned int) (nameend - d->__nextchar)
              == (unsigned int) strlen (p->name))
```

```
              {
                /* Exact match found.  */
                pfound = p;
                indfound = option_index;
                exact = 1;
                break;
              }
            else if (pfound == NULL)
              {
                /* First nonexact match found.  */
                pfound = p;
                indfound = option_index;
              }
            else if (long_only
                     || pfound->has_arg != p->has_arg
                     || pfound->flag != p->flag
                     || pfound->val != p->val)
              /* Second or later nonexact match found.  */
              ambig = 1;
          }

      if (ambig && !exact)
        {
          if (print_errors)
            {
#if defined _LIBC && defined USE_IN_LIBIO
              char *buf;

              if (__asprintf (&buf, _("%s: option `%s' is ambiguous\n"),
                              argv[0], argv[d->optind]) >= 0)
                {
                  _IO_flockfile (stderr);

                  int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                  ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                  __fxprintf (NULL, "%s", buf);

                  ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                  _IO_funlockfile (stderr);

                  free (buf);
                }
#else
              fprintf (stderr, _("%s: option `%s' is ambiguous\n"),
                       argv[0], argv[d->optind]);
#endif
            }
          d->__nextchar += strlen (d->__nextchar);
          d->optind++;
          d->optopt = 0;
          return '?';
        }

      if (pfound != NULL)
        {
          option_index = indfound;
          d->optind++;
          if (*nameend)
            {
              /* Don't test has_arg with >, because some C compilers don't
                 allow it to be used on enums.  */
              if (pfound->has_arg)
                d->optarg = nameend + 1;
              else
                {
                  if (print_errors)
                    {
#if defined _LIBC && defined USE_IN_LIBIO
                      char *buf;
                      int n;
#endif

                      if (argv[d->optind - 1][1] == '-')
                        {
                          /* --option */
#if defined _LIBC && defined USE_IN_LIBIO
                          n = __asprintf (&buf, _("\
%s: option `--%s' doesn't allow an argument\n"),
                                          argv[0], pfound->name);
#else
                          fprintf (stderr, _("\
%s: option `--%s' doesn't allow an argument\n"),
                                   argv[0], pfound->name);
#endif
                        }
                      else
                        {
                          /* +option or -option */
#if defined _LIBC && defined USE_IN_LIBIO
                          n = __asprintf (&buf, _("\
%s: option `%c%s' doesn't allow an argument\n"),
                                          argv[0], argv[d->optind - 1][0],
                                          pfound->name);
#else
                          fprintf (stderr, _("\
%s: option `%c%s' doesn't allow an argument\n"),
                                   argv[0], argv[d->optind - 1][0],
```

```
                                    pfound->name);
#endif
                        }

#if defined _LIBC && defined USE_IN_LIBIO
                      if (n >= 0)
                        {
                          _IO_flockfile (stderr);

                          int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                          ((_IO_FILE *) stderr)->_flags2
                            |= _IO_FLAGS2_NOTCANCEL;

                          __fxprintf (NULL, "%s", buf);

                          ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                          _IO_funlockfile (stderr);

                          free (buf);
                        }
#endif
                    }

                  d->__nextchar += strlen (d->__nextchar);

                  d->optopt = pfound->val;
                  return '?';
                }
            }
          else if (pfound->has_arg == 1)
            {
              if (d->optind < argc)
                d->optarg = argv[d->optind++];
              else
                {
                  if (print_errors)
                    {
#if defined _LIBC && defined USE_IN_LIBIO
                      char *buf;

                      if (__asprintf (&buf, _("\
%s: option `%s' requires an argument\n"),
                                      argv[0], argv[d->optind - 1]) >= 0)
                        {
                          _IO_flockfile (stderr);

                          int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                          ((_IO_FILE *) stderr)->_flags2
                            |= _IO_FLAGS2_NOTCANCEL;

                          __fxprintf (NULL, "%s", buf);

                          ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                          _IO_funlockfile (stderr);

                          free (buf);
                        }
#else
                      fprintf (stderr,
                               _("%s: option `%s' requires an argument\n"),
                               argv[0], argv[d->optind - 1]);
#endif
                    }
                  d->__nextchar += strlen (d->__nextchar);
                  d->optopt = pfound->val;
                  return optstring[0] == ':' ? ':' : '?';
                }
            }
          d->__nextchar += strlen (d->__nextchar);
          if (longind != NULL)
            *longind = option_index;
          if (pfound->flag)
            {
              *(pfound->flag) = pfound->val;
              return 0;
            }
          return pfound->val;
        }

      /* Can't find it as a long option.  If this is not getopt_long_only,
         or the option starts with '--' or is not a valid short
         option, then it's an error.
         Otherwise interpret it as a short option.  */
      if (!long_only || argv[d->optind][1] == '-'
          || strchr (optstring, *d->__nextchar) == NULL)
        {
          if (print_errors)
            {
#if defined _LIBC && defined USE_IN_LIBIO
              char *buf;
              int n;
#endif

              if (argv[d->optind][1] == '-')
                {
                  /* --option */
#if defined _LIBC && defined USE_IN_LIBIO
                  n = __asprintf (&buf, _("%s: unrecognized option `--%s'\n"),
```

```
                                argv[0], d->__nextchar);
#else
                  fprintf (stderr, _("%s: unrecognized option `--%s'\n"),
                           argv[0], d->__nextchar);
#endif
                }
              else
                {
                  /* +option or -option */
#if defined _LIBC && defined USE_IN_LIBIO
                  n = __asprintf (&buf, _("%s: unrecognized option `%c%s'\n"),
                                  argv[0], argv[d->optind][0], d->__nextchar);
#else
                  fprintf (stderr, _("%s: unrecognized option `%c%s'\n"),
                           argv[0], argv[d->optind][0], d->__nextchar);
#endif
                }

#if defined _LIBC && defined USE_IN_LIBIO
              if (n >= 0)
                {
                  _IO_flockfile (stderr);

                  int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                  ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                  __fxprintf (NULL, "%s", buf);

                  ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                  _IO_funlockfile (stderr);

                  free (buf);
                }
#endif
            }
          d->__nextchar = (char *) "";
          d->optind++;
          d->optopt = 0;
          return '?';
        }
    }

  /* Look at and handle the next short option-character.  */

  {
    char c = *d->__nextchar++;
    char *temp = strchr (optstring, c);

    /* Increment `optind' when we start to process its last character.  */
    if (*d->__nextchar == '\0')
      ++d->optind;

    if (temp == NULL || c == ':')
      {
        if (print_errors)
          {
#if defined _LIBC && defined USE_IN_LIBIO
            char *buf;
            int n;
#endif

            if (d->__posixly_correct)
              {
                /* 1003.2 specifies the format of this message.  */
#if defined _LIBC && defined USE_IN_LIBIO
                n = __asprintf (&buf, _("%s: illegal option -- %c\n"),
                                argv[0], c);
#else
                fprintf (stderr, _("%s: illegal option -- %c\n"), argv[0], c);
#endif
              }
            else
              {
#if defined _LIBC && defined USE_IN_LIBIO
                n = __asprintf (&buf, _("%s: invalid option -- %c\n"),
                                argv[0], c);
#else
                fprintf (stderr, _("%s: invalid option -- %c\n"), argv[0], c);
#endif
              }

#if defined _LIBC && defined USE_IN_LIBIO
            if (n >= 0)
              {
                _IO_flockfile (stderr);

                int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                __fxprintf (NULL, "%s", buf);

                ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                _IO_funlockfile (stderr);

                free (buf);
              }
#endif
          }
```

```
        d->optopt = c;
        return '?';
      }
    /* Convenience. Treat POSIX -W foo same as long option --foo */
    if (temp[0] == 'W' && temp[1] == ';')
      {
        char *nameend;
        const struct option *p;
        const struct option *pfound = NULL;
        int exact = 0;
        int ambig = 0;
        int indfound = 0;
        int option_index;

        /* This is an option that requires an argument.  */
        if (*d->__nextchar != '\0')
          {
            d->optarg = d->__nextchar;
            /* If we end this ARGV-element by taking the rest as an arg,
               we must advance to the next element now.  */
            d->optind++;
          }
        else if (d->optind == argc)
          {
            if (print_errors)
              {
                /* 1003.2 specifies the format of this message.  */
#if defined _LIBC && defined USE_IN_LIBIO
                char *buf;

                if (__asprintf (&buf,
                                _("%s: option requires an argument -- %c\n"),
                                argv[0], c) >= 0)
                  {
                    _IO_flockfile (stderr);

                    int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                    ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                    __fxprintf (NULL, "%s", buf);

                    ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                    _IO_funlockfile (stderr);

                    free (buf);
                  }
#else
                fprintf (stderr, _("%s: option requires an argument -- %c\n"),
                         argv[0], c);
#endif
              }
            d->optopt = c;
            if (optstring[0] == ':')
              c = ':';
            else
              c = '?';
            return c;
          }
        else
          /* We already incremented `d->optind' once;
             increment it again when taking next ARGV-elt as argument.  */
          d->optarg = argv[d->optind++];

        /* optarg is now the argument, see if it's in the
           table of longopts.  */

        for (d->__nextchar = nameend = d->optarg; *nameend && *nameend != '=';
             nameend++)
          /* Do nothing.  */ ;

        /* Test all long options for either exact match
           or abbreviated matches.  */
        for (p = longopts, option_index = 0; p->name; p++, option_index++)
          if (!strncmp (p->name, d->__nextchar, nameend - d->__nextchar))
            {
              if ((unsigned int) (nameend - d->__nextchar) == strlen (p->name))
                {
                  /* Exact match found.  */
                  pfound = p;
                  indfound = option_index;
                  exact = 1;
                  break;
                }
              else if (pfound == NULL)
                {
                  /* First nonexact match found.  */
                  pfound = p;
                  indfound = option_index;
                }
              else
                /* Second or later nonexact match found.  */
                ambig = 1;
            }
        if (ambig && !exact)
          {
            if (print_errors)
              {
#if defined _LIBC && defined USE_IN_LIBIO
```

```
                char *buf;

                if (__asprintf (&buf, _("%s: option `-W %s' is ambiguous\n"),
                                argv[0], argv[d->optind]) >= 0)
                  {
                    _IO_flockfile (stderr);

                    int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                    ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                    __fxprintf (NULL, "%s", buf);

                    ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                    _IO_funlockfile (stderr);

                    free (buf);
                  }
#else
                fprintf (stderr, _("%s: option `-W %s' is ambiguous\n"),
                         argv[0], argv[d->optind]);
#endif
              }
            d->__nextchar += strlen (d->__nextchar);
            d->optind++;
            return '?';
          }
        if (pfound != NULL)
          {
            option_index = indfound;
            if (*nameend)
              {
                /* Don't test has_arg with >, because some C compilers don't
                   allow it to be used on enums.  */
                if (pfound->has_arg)
                  d->optarg = nameend + 1;
                else
                  {
                    if (print_errors)
                      {
#if defined _LIBC && defined USE_IN_LIBIO
                        char *buf;

                        if (__asprintf (&buf, _("\
%s: option `-W %s' doesn't allow an argument\n"),
                                        argv[0], pfound->name) >= 0)
                          {
                            _IO_flockfile (stderr);

                            int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                            ((_IO_FILE *) stderr)->_flags2
                              |= _IO_FLAGS2_NOTCANCEL;

                            __fxprintf (NULL, "%s", buf);

                            ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                            _IO_funlockfile (stderr);

                            free (buf);
                          }
#else
                        fprintf (stderr, _("\
%s: option `-W %s' doesn't allow an argument\n"),
                                 argv[0], pfound->name);
#endif
                      }

                    d->__nextchar += strlen (d->__nextchar);
                    return '?';
                  }
              }
            else if (pfound->has_arg == 1)
              {
                if (d->optind < argc)
                  d->optarg = argv[d->optind++];
                else
                  {
                    if (print_errors)
                      {
#if defined _LIBC && defined USE_IN_LIBIO
                        char *buf;

                        if (__asprintf (&buf, _("\
%s: option `%s' requires an argument\n"),
                                        argv[0], argv[d->optind - 1]) >= 0)
                          {
                            _IO_flockfile (stderr);

                            int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                            ((_IO_FILE *) stderr)->_flags2
                              |= _IO_FLAGS2_NOTCANCEL;

                            __fxprintf (NULL, "%s", buf);

                            ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                            _IO_funlockfile (stderr);

                            free (buf);
                          }
```

```
#else
                      fprintf (stderr,
                               _("%s: option `%s' requires an argument\n"),
                               argv[0], argv[d->optind - 1]);
#endif
                    }
                  d->__nextchar += strlen (d->__nextchar);
                  return optstring[0] == ':' ? ':' : '?';
                }
            }
          d->__nextchar += strlen (d->__nextchar);
          if (longind != NULL)
            *longind = option_index;
          if (pfound->flag)
            {
              *(pfound->flag) = pfound->val;
              return 0;
            }
          return pfound->val;
        }
        d->__nextchar = NULL;
        return 'W';   /* Let the application handle it.   */
    }
  if (temp[1] == ':')
    {
      if (temp[2] == ':')
        {
          /* This is an option that accepts an argument optionally.  */
          if (*d->__nextchar != '\0')
            {
              d->optarg = d->__nextchar;
              d->optind++;
            }
          else
            d->optarg = NULL;
          d->__nextchar = NULL;
        }
      else
        {
          /* This is an option that requires an argument.  */
          if (*d->__nextchar != '\0')
            {
              d->optarg = d->__nextchar;
              /* If we end this ARGV-element by taking the rest as an arg,
                 we must advance to the next element now.  */
              d->optind++;
            }
          else if (d->optind == argc)
            {
              if (print_errors)
                {
                  /* 1003.2 specifies the format of this message.  */
#if defined _LIBC && defined USE_IN_LIBIO
                  char *buf;

                  if (__asprintf (&buf, _("\
%s: option requires an argument -- %c\n"),
                                  argv[0], c) >= 0)
                    {
                      _IO_flockfile (stderr);

                      int old_flags2 = ((_IO_FILE *) stderr)->_flags2;
                      ((_IO_FILE *) stderr)->_flags2 |= _IO_FLAGS2_NOTCANCEL;

                      __fxprintf (NULL, "%s", buf);

                      ((_IO_FILE *) stderr)->_flags2 = old_flags2;
                      _IO_funlockfile (stderr);

                      free (buf);
                    }
#else
                  fprintf (stderr,
                           _("%s: option requires an argument -- %c\n"),
                           argv[0], c);
#endif
                }
              d->optopt = c;
              if (optstring[0] == ':')
                c = ':';
              else
                c = '?';
            }
          else
            /* We already incremented `optind' once;
               increment it again when taking next ARGV-elt as argument.  */
            d->optarg = argv[d->optind++];
          d->__nextchar = NULL;
        }
    }
  return c;
}

int
_getopt_internal (int argc, char **argv, const char *optstring,
                  const struct option *longopts, int *longind,
                  int long_only, int posixly_correct)
```

```
{
  int result;

  getopt_data.optind = optind;
  getopt_data.opterr = opterr;

  result = _getopt_internal_r (argc, argv, optstring, longopts, longind,
                               long_only, posixly_correct, &getopt_data);

  optind = getopt_data.optind;
  optarg = getopt_data.optarg;
  optopt = getopt_data.optopt;

  return result;
}

/* glibc gets a LSB-compliant getopt.
   Standalone applications get a POSIX-compliant getopt.  */
#if _LIBC
enum { POSIXLY_CORRECT = 0 };
#else
enum { POSIXLY_CORRECT = 1 };
#endif

int
getopt (int argc, char *const *argv, const char *optstring)
{
  return _getopt_internal (argc, (char **) argv, optstring, NULL, NULL, 0,
                           POSIXLY_CORRECT);
}


#ifdef TEST

/* Compile with -DTEST to make an executable for use in testing
   the above definition of `getopt'.  */

int
main (int argc, char **argv)
{
  int c;
  int digit_optind = 0;

  while (1)
    {
      int this_option_optind = optind ? optind : 1;

      c = getopt (argc, argv, "abc:d:0123456789");
      if (c == -1)
        break;

      switch (c)
        {
        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
          if (digit_optind != 0 && digit_optind != this_option_optind)
            printf ("digits occur in two different argv-elements.\n");
          digit_optind = this_option_optind;
          printf ("option %c\n", c);
          break;

        case 'a':
          printf ("option a\n");
          break;

        case 'b':
          printf ("option b\n");
          break;

        case 'c':
          printf ("option c with value `%s'\n", optarg);
          break;

        case '?':
          break;

        default:
          printf ("?? getopt returned character code 0%o ??\n", c);
        }
    }

  if (optind < argc)
    {
      printf ("non-option ARGV-elements: ");
      while (optind < argc)
        printf ("%s ", argv[optind++]);
      printf ("\n");
    }

  exit (0);
```

```
}

#endif /* TEST */
```

Notice for package(s)

xz

```
insert GPL v3 text here

AUTOCONF CONFIGURE SCRIPT EXCEPTION

Version 3.0, 18 August 2009

Copyright © 2009 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This Exception is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file

The purpose of this Exception is to allow distribution of Autoconf`s typical output under terms of the recipient`s choice (including proprieta

0. Definitions.
"Covered Code" is the source or object code of a version of Autoconf that is a covered work under this License.

"Normally Copied Code" for a version of Autoconf means all parts of its Covered Code which that version can copy from its code (i.e., not from

"Ineligible Code" is Covered Code that is not Normally Copied Code.

1. Grant of Additional Permission.
You have permission to propagate output of Autoconf, even if such propagation would otherwise violate the terms of GPLv3. However, if by modif

2. No Weakening of Autoconf Copyleft.
The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements
```

Notice for package(s)

quota

```
/*
 * Copyright (c) 1980, 1990 Regents of the University of California. All
 * rights reserved.
 *
 * This code is derived from software contributed to Berkeley by Robert Elz at
 * The University of Melbourne.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met: 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer. 2.
 * Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution. 3. All advertising
 * materials mentioning features or use of this software must display the
 * following acknowledgement: This product includes software developed by the
 * University of California, Berkeley and its contributors. 4. Neither the
 * name of the University nor the names of its contributors may be used to
 * endorse or promote products derived from this software without specific
 * prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

#include "config.h"

/*
 * Disk quota reporting program.
 */
#include <sys/types.h>
#include <sys/param.h>
#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>
```

```
#include <pwd.h>
#include <grp.h>
#include <time.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>
#ifdef RPC
#include <rpc/rpc.h>
#include "rquota.h"
#endif

#include "quota.h"
#include "quotaops.h"
#include "quotasys.h"
#include "pot.h"
#include "common.h"

#define FL_QUIET 1
#define FL_VERBOSE 2
#define FL_USER 4
#define FL_GROUP 8
#define FL_SMARTSIZE 16
#define FL_LOCALONLY 32
#define FL_QUIETREFUSE 64
#define FL_NOAUTOFS 128
#define FL_NOWRAP 256
#define FL_FSLIST 512
#define FL_NUMNAMES 1024
#define FL_NFSALL 2048
#define FL_RAWGRACE 4096
#define FL_NO_MIXED_PATHS 8192
#define FL_SHOW_MNTPOINT 16384
#define FL_SHOW_DEVICE 32768

static int flags, fmt = -1;
char *progname;

static void usage(void)
{
        errstr( "%s%s%s%s%s",
                _("Usage: quota [-guqvswim] [-l | [-Q | -A]] [-F quotaformat]\n"),
                _("\tquota [-qvswim] [-l | [-Q | -A]] [-F quotaformat] -u username ...\n"),
                _("\tquota [-qvswim] [-l | [-Q | -A]] [-F quotaformat] -g groupname ...\n"),
                _("\tquota [-qvswugQm] [-F quotaformat] -f filesystem ...\n"),
                _("\n\
-u, --user                 display quota for user\n\
-g, --group                display quota for group\n\
-q, --quiet                print more terse message\n\
-v, --verbose              print more verbose message\n\
-s, --human-readable       display numbers in human friendly units (MB, GB...)\n\
    --always-resolve       always try to translate name to id, even if it is\n\
                           composed of only digits\n\
-w, --no-wrap              do not wrap long lines\n\
-p, --raw-grace            print grace time in seconds since epoch\n\
-l, --local-only           do not query NFS filesystems\n\
-Q, --quiet-refuse         do not print error message when NFS server does\n\
                           not respond\n\
-i, --no-autofs            do not query autofs mountpoints\n\
-F, --format=formatname    display quota of a specific format\n\
-f, --filesystem-list      display quota information only for given filesystems\n\
-A, --all-nfs              display quota for all NFS mountpoints\n\
-m, --no-mixed-pathnames   trim leading slashes from NFSv4 mountpoints\n\
    --show-mntpoint        show mount point of the file system in output\n\
    --hide-device          do not show file system device in output\n\
-h, --help                 display this help message and exit\n\
-V, --version              display version information and exit\n\n"));
        fprintf(stderr, _("Bugs to: %s\n"), MY_EMAIL);
        exit(1);
}

static void heading(int type, qid_t id, char *name, char *tag)
{
        char *spacehdr;

        if (flags & FL_SMARTSIZE)
                spacehdr = _("space");
        else
                spacehdr = _("blocks");

        printf(_("Disk quotas for %s %s (%cid %u): %s\n"), _(type2name(type)),
               name, *type2name(type), (uint) id, tag);
        if (!(flags & FL_QUIET) && !tag[0]) {
                printf("%15s%8s %7s%8s%8s%8s %7s%8s%8s\n", _("Filesystem"),
                       spacehdr, _("quota"), _("limit"), _("grace"),
                       _("files"), _("quota"), _("limit"), _("grace"));
        }
}

static void print_fs_location(struct dquot *q)
{
        struct quota_handle *h = q->dq_h;

        if (flags & FL_QUIET) {
                if (flags & FL_SHOW_DEVICE)
                        printf(" %s", h->qh_quotadev);
                if (flags & FL_SHOW_MNTPOINT)
                        printf(" %s", h->qh_dir);
                putchar('\n');
```

```
        } else {
                int wrap = 0;

                if (flags & FL_SHOW_DEVICE && flags & FL_SHOW_MNTPOINT &&
                    !(flags & FL_NOWRAP))
                        wrap = 1;
                else if (flags & FL_SHOW_DEVICE && strlen(h->qh_quotadev) > 15 &&
                    !(flags & FL_NOWRAP))
                        wrap = 1;
                else if (flags & FL_SHOW_MNTPOINT && strlen(h->qh_dir) > 15 &&
                    !(flags & FL_NOWRAP))
                        wrap = 1;

                if (flags & FL_SHOW_DEVICE) {
                        if (wrap || flags & FL_SHOW_MNTPOINT)
                                printf("%s", h->qh_quotadev);
                        else
                                printf("%15s", h->qh_quotadev);
                }
                if (flags & FL_SHOW_MNTPOINT) {
                        if (flags & FL_SHOW_DEVICE)
                                putchar(' ');
                        if (wrap || flags & FL_SHOW_DEVICE)
                                printf("%s", h->qh_dir);
                        else
                                printf("%15s", h->qh_dir);
                }
                if (wrap)
                        printf("\n%15s", "");
        }
}

static int showquotas(int type, qid_t id, int mntcnt, char **mnt)
{
        struct dquot *qlist, *q;
        char *msgi, *msgb;
        char timebuf[MAXTIMELEN];
        char name[MAXNAMELEN];
        struct quota_handle **handles;
        int lines = 0, bover, iover, over;
        time_t now;

        time(&now);
        id2name(id, type, name);
        handles = create_handle_list(mntcnt, mnt, type, fmt,
                IOI_READONLY | ((flags & FL_NO_MIXED_PATHS) ? 0 : IOI_NFS_MIXED_PATHS),
                ((flags & FL_NOAUTOFS) ? MS_NO_AUTOFS : 0)
                | ((flags & FL_LOCALONLY) ? MS_LOCALONLY : 0)
                | ((flags & FL_NFSALL) ? MS_NFS_ALL : 0));
        qlist = getprivs(id, handles, !!(flags & FL_QUIETREFUSE));
        over = 0;
        for (q = qlist; q; q = q->dq_next) {
                bover = iover = 0;
                if (!(flags & FL_VERBOSE) && !q->dq_dqb.dqb_isoftlimit && !q->dq_dqb.dqb_ihardlimit
                    && !q->dq_dqb.dqb_bsoftlimit && !q->dq_dqb.dqb_bhardlimit)
                        continue;
                msgi = NULL;
                if (q->dq_dqb.dqb_ihardlimit && q->dq_dqb.dqb_curinodes >= q->dq_dqb.dqb_ihardlimit) {
                        msgi = _("File limit reached on");
                        iover = 1;
                }
                else if (q->dq_dqb.dqb_isoftlimit
                         && q->dq_dqb.dqb_curinodes > q->dq_dqb.dqb_isoftlimit) {
                        if (q->dq_dqb.dqb_itime > now) {
                                msgi = _("In file grace period on");
                                iover = 2;
                        }
                        else {
                                msgi = _("Over file quota on");
                                iover = 3;
                        }
                }
                msgb = NULL;
                if (q->dq_dqb.dqb_bhardlimit && toqb(q->dq_dqb.dqb_curspace) >= q->dq_dqb.dqb_bhardlimit) {
                                msgb = _("Block limit reached on");
                                bover = 1;
                }
                else if (q->dq_dqb.dqb_bsoftlimit
                         && toqb(q->dq_dqb.dqb_curspace) > q->dq_dqb.dqb_bsoftlimit) {
                        if (q->dq_dqb.dqb_btime > now) {
                                msgb = _("In block grace period on");
                                bover = 2;
                        }
                        else {
                                msgb = _("Over block quota on");
                                bover = 3;
                        }
                }
                over |= bover | iover;
                if (flags & FL_QUIET) {
                        if ((msgi || msgb) && !lines++)
                                heading(type, id, name, "");
                        if (msgi) {
                                printf("\t%s", msgi);
                                print_fs_location(q);
                        }
                        if (msgb) {
                                printf("\t%s", msgb);
```

```
                        print_fs_location(q);
                }
                continue;
        }
        if ((flags & FL_VERBOSE) || q->dq_dqb.dqb_curspace || q->dq_dqb.dqb_curinodes) {
                char numbuf[3][MAXNUMLEN];

                if (!lines++)
                        heading(type, id, name, "");
                print_fs_location(q);
                if (!(flags & FL_RAWGRACE)) {
                        if (bover)
                                difftime2str(q->dq_dqb.dqb_btime, timebuf);
                        else
                                timebuf[0] = 0;
                }
                else {
                        if (bover)
                                sprintf(timebuf, "%llu", (long long unsigned int)q->dq_dqb.dqb_btime);
                        else
                                strcpy(timebuf, "0");
                }
                space2str(toqb(q->dq_dqb.dqb_curspace), numbuf[0], !!(flags & FL_SMARTSIZE));
                space2str(q->dq_dqb.dqb_bsoftlimit, numbuf[1], !!(flags & FL_SMARTSIZE));
                space2str(q->dq_dqb.dqb_bhardlimit, numbuf[2], !!(flags & FL_SMARTSIZE));
                printf(" %7s%c %6s %7s %7s", numbuf[0], bover ? '*' : ' ', numbuf[1],
                       numbuf[2], timebuf);

                if (!(flags & FL_RAWGRACE)) {
                        if (iover)
                                difftime2str(q->dq_dqb.dqb_itime, timebuf);
                        else
                                timebuf[0] = 0;
                }
                else {
                        if (iover)
                                sprintf(timebuf, "%llu", (long long unsigned int)q->dq_dqb.dqb_itime);
                        else
                                strcpy(timebuf, "0");
                }
                number2str(q->dq_dqb.dqb_curinodes, numbuf[0], !!(flags & FL_SMARTSIZE));
                number2str(q->dq_dqb.dqb_isoftlimit, numbuf[1], !!(flags & FL_SMARTSIZE));
                number2str(q->dq_dqb.dqb_ihardlimit, numbuf[2], !!(flags & FL_SMARTSIZE));
                printf(" %7s%c %6s %7s %7s\n", numbuf[0], iover ? '*' : ' ', numbuf[1],
                       numbuf[2], timebuf);
                continue;
        }
    }
    if (!(flags & FL_QUIET) && !lines && qlist)
            heading(type, id, name, _("none"));
    freeprivs(qlist);
    dispose_handle_list(handles);
    return over > 0 ? 1 : 0;
}

int main(int argc, char **argv)
{
    int ngroups;
    gid_t gidset[NGROUPS], *gidsetp;
    int i, ret;
    struct option long_opts[] = {
            { "help", 0, NULL, 'h' },
            { "version", 0, NULL, 'V' },
            { "user", 0, NULL, 'u' },
            { "group", 0, NULL, 'g' },
            { "quiet", 0, NULL, 'q' },
            { "verbose", 0, NULL, 'v' },
            { "human-readable", 0, NULL, 's' },
            { "always-resolve", 0, NULL, 256 },
            { "raw-grace", 0, NULL, 'p' },
            { "local-only", 0, NULL, 'l' },
            { "no-autofs", 0, NULL, 'i' },
            { "quiet-refuse", 0, NULL, 'Q' },
            { "format", 1, NULL, 'F' },
            { "no-wrap", 0, NULL, 'w' },
            { "filesystem-list", 0, NULL, 'f' },
            { "all-nfs", 0, NULL, 'A' },
            { "no-mixed-pathnames", 0, NULL, 'm' },
            { "show-mntpoint", 0, NULL, 257 },
            { "hide-device", 0, NULL, 258 },
            { NULL, 0, NULL, 0 }
    };

    gettexton();
    progname = basename(argv[0]);

    flags |= FL_SHOW_DEVICE;
    while ((ret = getopt_long(argc, argv, "hguqvsVliQF:wfApm", long_opts, NULL)) != -1) {
            switch (ret) {
              case 'g':
                      flags |= FL_GROUP;
                      break;
              case 'u':
                      flags |= FL_USER;
                      break;
              case 'q':
                      flags |= FL_QUIET;
                      break;
```

```
		case 'v':
			flags |= FL_VERBOSE;
			break;
		case 'F':
			if ((fmt = name2fmt(optarg)) == QF_ERROR)	/* Error? */
				exit(1);
			break;
		case 's':
			flags |= FL_SMARTSIZE;
			break;
		case 'p':
			flags |= FL_RAWGRACE;
			break;
		case 256:
			flags |= FL_NUMNAMES;
			break;
		case 'l':
			flags |= FL_LOCALONLY;
			break;
		case 'Q':
			flags |= FL_QUIETREFUSE;
			break;
		case 'i':
			flags |= FL_NOAUTOFS;
			break;
		case 'w':
			flags |= FL_NOWRAP;
			break;
		case 'f':
			flags |= FL_FSLIST;
			break;
		case 'A':
			flags |= FL_NFSALL;
			break;
		case 'm':
			flags |= FL_NO_MIXED_PATHS;
			break;
		case 257:
			flags |= FL_SHOW_MNTPOINT;
			break;
		case 258:
			flags &= ~FL_SHOW_DEVICE;
			break;
		case 'V':
			version();
			exit(0);
		case 'h':
		default:
			usage();
		}
	}
	argc -= optind;
	argv += optind;

	if (!(flags & FL_USER) && !(flags & FL_GROUP))
		flags |= FL_USER;
	if (flags & FL_FSLIST && flags & (FL_LOCALONLY | FL_NOAUTOFS))
		errstr(_("Warning: Ignoring -%c when filesystem list specified.\n"), flags & FL_LOCALONLY ? 'l' : 'i');

	init_kernel_interface();

	ret = 0;
	if (argc == 0 || flags & FL_FSLIST) {
		if (flags & FL_FSLIST && argc == 0)
			die(1, _("No filesystem specified.\n"));
		if (flags & FL_USER)
			ret |= showquotas(USRQUOTA, getuid(), argc, argv);
		if (flags & FL_GROUP) {
			ngroups = sysconf(_SC_NGROUPS_MAX);
			if (ngroups > NGROUPS) {
				gidsetp = malloc(ngroups * sizeof(gid_t));
				if (!gidsetp)
					die(1, _("Gid set allocation (%d): %s\n"), ngroups, strerror(errno));
			} else {
				gidsetp = &gidset[0];
			}
			ngroups = getgroups(ngroups, gidsetp);
			if (ngroups < 0)
				die(1, _("getgroups(): %s\n"), strerror(errno));
			for (i = 0; i < ngroups; i++)
				ret |= showquotas(GRPQUOTA, gidsetp[i], argc, argv);
		}
		exit(ret);
	}

	if ((flags & FL_USER) && (flags & FL_GROUP))
		usage();

	if (flags & FL_USER)
		for (; argc > 0; argc--, argv++)
			ret |= showquotas(USRQUOTA, user2uid(*argv, !!(flags & FL_NUMNAMES), NULL), 0, NULL);
	else if (flags & FL_GROUP)
		for (; argc > 0; argc--, argv++)
			ret |= showquotas(GRPQUOTA, group2gid(*argv, !!(flags & FL_NUMNAMES), NULL), 0, NULL);
	return ret;
}
```

Notice for package(s)

quota

```
/*
 * QUOTA    An implementation of the diskquota system for the LINUX
 *          operating system. QUOTA is implemented using the BSD systemcall
 *          interface as the means of communication with the user level.
 *          Should work for all filesystems because of integration into the
 *          VFS layer of the operating system.
 *          This is based on the Melbourne quota system wich uses both user and
 *          group quota files.
 *
 *          This part does the lookup of the info.
 *
 * Author:  Marco van Wieringen <mvw@planets.elm.net>
 *
 *          This program is free software; you can redistribute it and/or
 *          modify it under the terms of the GNU General Public License
 *          as published by the Free Software Foundation; either version
 *          2 of the License, or (at your option) any later version.
 */

#include "config.h"

#include <rpc/rpc.h>
#include <arpa/inet.h>
#include <paths.h>
#include <stdio.h>
#include <syslog.h>
#include <time.h>
#include <stdint.h>

#include "mntopt.h"
#include "quotaops.h"
#include "bylabel.h"
#include "rquota.h"
#include "quotaio.h"
#include "quotasys.h"
#include "dqblk_rpc.h"
#include "common.h"

#define STDIN_FILENO    0

#define TYPE_EXTENDED   0x01
#define ACTIVE          0x02

#define FACILITY        LOG_LOCAL7

#ifndef MAXPATHNAMELEN
#define MAXPATHNAMELEN BUFSIZ
#endif

#define NETTYPE AF_INET

/* Options from rquota_svc.c */
#define FL_AUTOFS 4
extern int flags;

extern char nfs_pseudoroot[PATH_MAX];

/*
 * Global unix authentication credentials.
 */
extern struct authunix_parms *unix_cred;

int in_group(gid_t * gids, uint32_t len, gid_t gid)
{
        gid_t *gidsp = gids + len;

        while (gidsp > gids)
                if (*(--gids) == gid)
                        return 1;

        return 0;
}

static inline void servnet2utildqblk(struct util_dqblk *u, sq_dqblk * n)
{
        time_t now;

        time(&now);
        u->dqb_bhardlimit = n->rq_bhardlimit;
        u->dqb_bsoftlimit = n->rq_bsoftlimit;
        u->dqb_ihardlimit = n->rq_fhardlimit;
        u->dqb_isoftlimit = n->rq_fsoftlimit;
        u->dqb_curspace = ((qsize_t)n->rq_curblocks) << RPC_DQBLK_SIZE_BITS;
        u->dqb_curinodes = n->rq_curfiles;
        if (n->rq_btimeleft)
                u->dqb_btime = (int32_t)n->rq_btimeleft + now;
        else
                u->dqb_btime = 0;
        if (n->rq_ftimeleft)
```

```
                u->dqb_itime = (int32_t)n->rq_ftimeleft + now;
        else
                u->dqb_itime = 0;
}

/* XDR transports 32b variables exactly. Find smallest needed shift to fit
 * 64b variable into into 32 bits and to preserve precision as high as
 * possible. */
static int find_block_shift(qsize_t hard, qsize_t soft, qsize_t cur)
{
        int shift;
        qsize_t value = hard;

        if (value < soft)
                value = soft;
        if (value < cur)
                value = cur;
        value >>= 32;
        for (shift = QUOTABLOCK_BITS; value; shift++)
                value >>= 1;

        return shift;
}

static inline void servutil2netdqblk(struct rquota *n, struct util_dqblk *u)
{
        time_t now;
        int shift;

        shift = find_block_shift(u->dqb_bhardlimit, u->dqb_bsoftlimit,
                toqb(u->dqb_curspace));
        n->rq_bsize = 1 << shift;
        n->rq_bhardlimit = u->dqb_bhardlimit >> (shift - QUOTABLOCK_BITS);
        n->rq_bsoftlimit = u->dqb_bsoftlimit >> (shift - QUOTABLOCK_BITS);
        n->rq_fhardlimit = u->dqb_ihardlimit;
        n->rq_fsoftlimit = u->dqb_isoftlimit;
        n->rq_curblocks = toqb(u->dqb_curspace) >> (shift - QUOTABLOCK_BITS);
        n->rq_curfiles = u->dqb_curinodes;

        time(&now);
        if (u->dqb_btime)
                n->rq_btimeleft = difftime2net(u->dqb_btime, now);
        else
                n->rq_btimeleft = 0;
        if (u->dqb_itime)
                n->rq_ftimeleft = difftime2net(u->dqb_itime, now);
        else
                n->rq_ftimeleft = 0;
}

setquota_rslt *setquotainfo(int lflags, caddr_t * argp, struct svc_req *rqstp)
{
        static setquota_rslt result;

#if defined(RPC_SETQUOTA)
        union {
                setquota_args *args;
                ext_setquota_args *ext_args;
        } arguments;
        struct util_dqblk dqblk;
        struct dquot *dquot;
        struct mount_entry *mnt;
        char pathname[PATH_MAX] = {0};
        char *pathp = pathname;
        int id, qcmd, type;
        struct quota_handle *handles[2] = { NULL, NULL };

        /*
         * First check authentication.
         */
        if (lflags & TYPE_EXTENDED) {
                arguments.ext_args = (ext_setquota_args *) argp;

                id = arguments.ext_args->sqa_id;
                if (unix_cred->aup_uid != 0) {
                        result.status = Q_EPERM;
                        return (&result);
                }

                qcmd = arguments.ext_args->sqa_qcmd;
                type = arguments.ext_args->sqa_type;
                if (arguments.ext_args->sqa_pathp[0] != '/')
                        sstrncpy(pathname, nfs_pseudoroot, PATH_MAX);
                sstrncat(pathname, arguments.ext_args->sqa_pathp, PATH_MAX);
                servnet2utildqblk(&dqblk, &arguments.ext_args->sqa_dqblk);
        }
        else {
                arguments.args = (setquota_args *) argp;

                id = arguments.args->sqa_id;
                if (unix_cred->aup_uid != 0) {
                        result.status = Q_EPERM;
                        return (&result);
                }

                qcmd = arguments.args->sqa_qcmd;
                type = USRQUOTA;
                if (arguments.args->sqa_pathp[0] != '/')
```

```
                        sstrncpy(pathname, nfs_pseudoroot, PATH_MAX);
                sstrncat(pathname, arguments.args->sqa_pathp, PATH_MAX);
                servnet2utildqblk(&dqblk, &arguments.args->sqa_dqblk);
        }

        result.status = Q_NOQUOTA;
        result.setquota_rslt_u.sqr_rquota.rq_bsize = RPC_DQBLK_SIZE;

        if (init_mounts_scan(1, &pathp, MS_QUIET | MS_NO_MNTPOINT | MS_NFS_ALL | ((flags & FL_AUTOFS) ? 0 : MS_NO_AUTOFS)) < 0)
                goto out;
        if (!(mnt = get_next_mount())) {
                end_mounts_scan();
                goto out;
        }
        if (!(handles[0] = init_io(mnt, type, -1, 0))) {
                end_mounts_scan();
                goto out;
        }
        end_mounts_scan();
        if (!(dquot = handles[0]->qh_ops->read_dquot(handles[0], id)))
                goto out;
        if (qcmd == QCMD(Q_RPC_SETQLIM, type) || qcmd == QCMD(Q_RPC_SETQUOTA, type)) {
                dquot->dq_dqb.dqb_bsoftlimit = dqblk.dqb_bsoftlimit;
                dquot->dq_dqb.dqb_bhardlimit = dqblk.dqb_bhardlimit;
                dquot->dq_dqb.dqb_isoftlimit = dqblk.dqb_isoftlimit;
                dquot->dq_dqb.dqb_ihardlimit = dqblk.dqb_ihardlimit;
                dquot->dq_dqb.dqb_btime = dqblk.dqb_btime;
                dquot->dq_dqb.dqb_itime = dqblk.dqb_itime;
        }
        if (qcmd == QCMD(Q_RPC_SETUSE, type) || qcmd == QCMD(Q_RPC_SETQUOTA, type)) {
                dquot->dq_dqb.dqb_curspace = dqblk.dqb_curspace;
                dquot->dq_dqb.dqb_curinodes = dqblk.dqb_curinodes;
        }
        if (handles[0]->qh_ops->commit_dquot(dquot, COMMIT_LIMITS) == -1) {
                free(dquot);
                goto out;
        }
        free(dquot);
        result.status = Q_OK;
out:
        dispose_handle_list(handles);
#else
        result.status = Q_EPERM;
#endif
        return (&result);
}

getquota_rslt *getquotainfo(int lflags, caddr_t * argp, struct svc_req * rqstp)
{
        static getquota_rslt result;
        union {
                getquota_args *args;
                ext_getquota_args *ext_args;
        } arguments;
        struct dquot *dquot = NULL;
        struct mount_entry *mnt;
        char pathname[PATH_MAX] = {0};
        char *pathp = pathname;
        int id, type;
        struct quota_handle *handles[2] = { NULL, NULL };

        /*
         * First check authentication.
         */
        if (lflags & TYPE_EXTENDED) {
                arguments.ext_args = (ext_getquota_args *) argp;
                id = arguments.ext_args->gqa_id;
                type = arguments.ext_args->gqa_type;
                if (arguments.ext_args->gqa_pathp[0] != '/')
                        sstrncpy(pathname, nfs_pseudoroot, PATH_MAX);
                sstrncat(pathname, arguments.ext_args->gqa_pathp, PATH_MAX);

                if (type == USRQUOTA && unix_cred->aup_uid && unix_cred->aup_uid != id) {
                        result.status = Q_EPERM;
                        return (&result);
                }

                if (type == GRPQUOTA && unix_cred->aup_uid && unix_cred->aup_gid != id &&
                    !in_group((gid_t *) unix_cred->aup_gids, unix_cred->aup_len, id)) {
                        result.status = Q_EPERM;
                        return (&result);
                }
        }
        else {
                arguments.args = (getquota_args *) argp;
                id = arguments.args->gqa_uid;
                type = USRQUOTA;
                if (arguments.ext_args->gqa_pathp[0] != '/')
                        sstrncpy(pathname, nfs_pseudoroot, PATH_MAX);
                sstrncat(pathname, arguments.args->gqa_pathp, PATH_MAX);

                if (unix_cred->aup_uid && unix_cred->aup_uid != id) {
                        result.status = Q_EPERM;
                        return (&result);
                }
        }

        result.status = Q_NOQUOTA;
```

```
        if (init_mounts_scan(1, &pathp, MS_QUIET | MS_NO_MNTPOINT | MS_NFS_ALL | ((flags & FL_AUTOFS) ? 0 : MS_NO_AUTOFS)) < 0)
                goto out;
        if (!(mnt = get_next_mount())) {
                end_mounts_scan();
                goto out;
        }
        if (!(handles[0] = init_io(mnt, type, -1, IOI_READONLY))) {
                end_mounts_scan();
                goto out;
        }
        end_mounts_scan();
        if (!(lflags & ACTIVE) || QIO_ENABLED(handles[0]))
                dquot = handles[0]->qh_ops->read_dquot(handles[0], id);
        if (dquot) {
                result.status = Q_OK;
                result.getquota_rslt_u.gqr_rquota.rq_active =
                        QIO_ENABLED(handles[0]) ? TRUE : FALSE;
                servutil2netdqblk(&result.getquota_rslt_u.gqr_rquota, &dquot->dq_dqb);
                free(dquot);
        }
out:
        dispose_handle_list(handles);
        return (&result);
}

/*
 * Map RPC-entrypoints to local function names.
 */
getquota_rslt *rquotaproc_getquota_1_svc(getquota_args * argp, struct svc_req * rqstp)
{
        return (getquotainfo(0, (caddr_t *) argp, rqstp));
}

getquota_rslt *rquotaproc_getactivequota_1_svc(getquota_args * argp, struct svc_req * rqstp)
{
        return (getquotainfo(ACTIVE, (caddr_t *) argp, rqstp));
}

getquota_rslt *rquotaproc_getquota_2_svc(ext_getquota_args * argp, struct svc_req * rqstp)
{
        return (getquotainfo(TYPE_EXTENDED, (caddr_t *) argp, rqstp));
}

getquota_rslt *rquotaproc_getactivequota_2_svc(ext_getquota_args * argp, struct svc_req * rqstp)
{
        return (getquotainfo(TYPE_EXTENDED | ACTIVE, (caddr_t *) argp, rqstp));
}

setquota_rslt *rquotaproc_setquota_1_svc(setquota_args * argp, struct svc_req * rqstp)
{
        return (setquotainfo(0, (caddr_t *) argp, rqstp));
}

setquota_rslt *rquotaproc_setactivequota_1_svc(setquota_args * argp, struct svc_req * rqstp)
{
        return (setquotainfo(ACTIVE, (caddr_t *) argp, rqstp));
}

setquota_rslt *rquotaproc_setquota_2_svc(ext_setquota_args * argp, struct svc_req * rqstp)
{
        return (setquotainfo(TYPE_EXTENDED, (caddr_t *) argp, rqstp));
}

setquota_rslt *rquotaproc_setactivequota_2_svc(ext_setquota_args * argp, struct svc_req * rqstp)
{
        return (setquotainfo(TYPE_EXTENDED | ACTIVE, (caddr_t *) argp, rqstp));
}
```

Notice for package(s)

quota

```
#include "config.h"

#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <netdb.h>
#include <errno.h>
#include <rpc/rpc.h>
#include <sys/socket.h>

#include "common.h"
#include "pot.h"

static int svc_socket (u_long number, int type, int protocol, int port, int reuse)
{
        struct sockaddr_in addr;
        char rpcdata [1024], servdata [1024];
        struct rpcent rpcbuf, *rpcp = NULL;
        struct servent servbuf, *servp = NULL;
        int sock, ret;
        const char *proto = protocol == IPPROTO_TCP ? "tcp" : "udp";

        if ((sock = socket (AF_INET, type, protocol)) < 0) {
                errstr(_("Cannot create socket: %s\n"), strerror(errno));
                return -1;
        }

        if (reuse) {
                ret = 1;
                if (setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &ret, sizeof(ret)) < 0) {
                        errstr(_("Cannot set socket options: %s\n"), strerror(errno));
                        return -1;
                }
        }

        memset(&addr, 0, sizeof(addr));
        addr.sin_family = AF_INET;

        if (!port) {
                ret = getrpcbynumber_r(number, &rpcbuf, rpcdata, sizeof(rpcdata), &rpcp);
                if (ret == 0 && rpcp != NULL) {
                        /* First try name */
                        ret = getservbyname_r(rpcp->r_name, proto, &servbuf, servdata,
                                              sizeof servdata, &servp);
                        if ((ret != 0 || servp == NULL) && rpcp->r_aliases) {
                                const char **a;

                                /* Then we try aliases. */
                                for (a = (const char **) rpcp->r_aliases; *a != NULL; a++) {
                                        ret = getservbyname_r(*a, proto, &servbuf, servdata,
                                                              sizeof servdata, &servp);
                                        if (ret == 0 && servp != NULL)
                                                break;
                                }
                        }
                        if (ret == 0 && servp != NULL)
                                port = servp->s_port;
                }
        }
        else
                port = htons(port);

        if (port) {
                addr.sin_port = port;
                if (bind(sock, (struct sockaddr *) &addr, sizeof(struct sockaddr_in)) < 0) {
                        errstr(_("Cannot bind to given address: %s\n"), strerror(errno));
                        close (sock);
                        return -1;
                }
        }
        else {
                /* Service not found? */
                close(sock);
                return -1;
        }

        return sock;
}

/*
 * Create and bind a TCP socket based on program number
 */
int svctcp_socket(u_long number, int port, int reuse)
{
        return svc_socket(number, SOCK_STREAM, IPPROTO_TCP, port, reuse);
}

/*
 * Create and bind a UDP socket based on program number
 */
int svcudp_socket(u_long number, int port, int reuse)
{
        return svc_socket(number, SOCK_DGRAM, IPPROTO_UDP, port, reuse);
}
```

Notice for package(s)

stat

```
Copyright (C) 1996-2001 Michael Meskes

STAT is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free
Software Foundation; either version 1, or (at your option) any later
version.

STAT is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for
more details.
```

Notice for package(s)

opkg-utils

```
#!/usr/bin/env python
#   Copyright (C) 2001 Alexander S. Guy <a7r@andern.org>
#                      Andern Research Labs
#
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2, or (at your option)
#   any later version.
#
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#   GNU General Public License for more details.
#
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 59 Temple Place - Suite 330,
#   Boston, MA 02111-1307, USA.  */
#
#   Copyright 2001, Russell Nelson <opkg.py@russnelson.com>
#   Added reading in of packages.
#   Added missing package information fields.
#   Changed render_control() to __repr__().
#
# Current Issues:
#    The API doesn't validate package information fields.  It should be
#        throwing exceptions in the right places.
#    Executions of tar could silently fail.
#    Executions of tar *do* fail, and loudly, because you have to specify a full filename,
#        and tar complains if any files are missing, and the opkg spec doesn't require
#        people to say "./control.tar.gz" or "./control" when they package files.
#        It would be much better to require ./control or disallow ./control (either)
#        rather than letting people pick.  Some freedoms aren't worth their cost.

import tempfile
import os
import sys
import glob
import hashlib
import re
import subprocess
from stat import ST_SIZE
import arfile
import tarfile
import textwrap

class Version:
    """A class for holding parsed package version information."""
    def __init__(self, epoch, version):
        self.epoch = epoch
        self.version = version

    def _versioncompare(self, selfversion, refversion):
        if not selfversion: selfversion = ""
        if not refversion: refversion = ""
        while 1:
            ## first look for non-numeric version component
            selfm = re.match('([^0-9]*)(.*)', selfversion)
            #print(('selfm', selfm.groups()))
            (selfalpha, selfversion) = selfm.groups()
            refm = re.match('([^0-9]*)(.*)', refversion)
            #print(('refm', refm.groups())
            (refalpha, refversion) = refm.groups()
            if (selfalpha > refalpha):
                return 1
            elif (selfalpha < refalpha):
                return -1
            ## now look for numeric version component
            (selfnum, selfversion) = re.match('([0-9]*)(.*)', selfversion).groups()
```

```
            (refnum, refversion) = re.match('([0-9]*)(.*)', refversion).groups()
            #print(('selfnum', selfnum, selfversion)
            #print(('refnum', refnum, refversion)
            if (selfnum != ''):
                selfnum = int(selfnum)
            else:
                selfnum = -1
            if (refnum != ''):
                refnum = int(refnum)
            else:
                refnum = -1
            if (selfnum > refnum):
                return 1
            elif (selfnum < refnum):
                return -1
            if selfversion == '' and refversion == '':
                return 0

    def compare(self, ref):
        if (self.epoch > ref.epoch):
            return 1
        elif (self.epoch < ref.epoch):
            return -1
        else:
            self_ver_comps = re.match(r"(.+?)(-r.+)?$", self.version)
            ref_ver_comps = re.match(r"(.+?)(-r.+)?$", ref.version)
            #print((self_ver_comps.group(1), self_ver_comps.group(2)))
            #print((ref_ver_comps.group(1), ref_ver_comps.group(2)))
            r = self._versioncompare(self_ver_comps.group(1), ref_ver_comps.group(1))
            if r == 0:
                r = self._versioncompare(self_ver_comps.group(2), ref_ver_comps.group(2))
            #print("compare: %s vs %s = %d" % (self, ref, r))
            return r

    def __str__(self):
        return str(self.epoch) + ":" + self.version

def parse_version(versionstr):
    epoch = 0
    # check for epoch
    m = re.match('([0-9]*):(.*)', versionstr)
    if m:
        (epochstr, versionstr) = m.groups()
        epoch = int(epochstr)
    return Version(epoch, versionstr)

class Package:
    """A class for creating objects to manipulate (e.g. create) opkg
       packages."""

    # fn: Package file path
    # relpath: If this argument is set, the file path is given relative to this
    #   path when a string representation of the Package object is created. If
    #   this argument is not set, the basename of the file path is given.
    def __init__(self, fn=None, relpath=None):
        self.package = None
        self.version = 'none'
        self.parsed_version = None
        self.architecture = None
        self.maintainer = None
        self.source = None
        self.description = None
        self.depends = None
        self.provides = None
        self.replaces = None
        self.conflicts = None
        self.recommends = None
        self.suggests = None
        self.section = None
        self.filename_header = None
        self.file_list = []
        # md5 and size is lazy attribute, computed on demand
        #self.md5 = None
        #self.size = None
        self.installed_size = None
        self.filename = None
        self.file_ext_opk = "ipk"
        self.homepage = None
        self.oe = None
        self.priority = None
        self.tags = None
        self.fn = fn
        self.license = None

        if fn:
            # see if it is deb format
            f = open(fn, "rb")

            if relpath:
                self.filename = os.path.relpath(fn, relpath)
            else:
                self.filename = os.path.basename(fn)

            ## sys.stderr.write("  extracting control.tar.gz from %s\n"% (fn,))

            ar = arfile.ArFile(f, fn)
            tarStream = ar.open("control.tar.gz")
            tarf = tarfile.open("control.tar.gz", "r", tarStream)
```

```
        try:
            control = tarf.extractfile("control")
        except KeyError:
            control = tarf.extractfile("./control")
        try:
            self.read_control(control)
        except TypeError as e:
            sys.stderr.write("Cannot read control file '%s' - %s\n" % (fn, e))
        control.close()

    self.scratch_dir = None
    self.file_dir = None
    self.meta_dir = None

def __getattr__(self, name):
    if name == "md5":
        self._computeFileMD5()
        return self.md5
    elif name == 'size':
        return self._get_file_size()
    else:
        raise AttributeError(name)

def _computeFileMD5(self):
    # compute the MD5.
    if not self.fn:
        self.md5 = 'Unknown'
    else:
        f = open(self.fn, "rb")
        sum = hashlib.md5()
        while True:
           data = f.read(1024)
           if not data: break
           sum.update(data)
        f.close()
        self.md5 = sum.hexdigest()

def _get_file_size(self):
    if not self.fn:
        self.size = 0;
    else:
        stat = os.stat(self.fn)
        self.size = stat[ST_SIZE]
    return int(self.size)

def read_control(self, control):
    import os

    line = control.readline()
    while 1:
        if not line: break
        line = line.rstrip()
        lineparts = re.match(r'([\w-]*?):\s*(.*)', str(line))
        if lineparts:
            name = lineparts.group(1).lower()
            value = lineparts.group(2)
            while 1:
                line = control.readline()
                if not line: break
                if line[0] != ' ': break
                value = value + '\n' + line
            if name == 'size':
                self.size = int(value)
            elif name == 'md5sum':
                self.md5 = value
            elif name in self.__dict__:
                self.__dict__[name] = value
            else:
                print("Lost field %s, %s" % (name,value))
                pass

            if line and line[0] == '\n':
                return # consumes one blank line at end of package descriptoin
        else:
            line = control.readline()
            pass
    return

def _setup_scratch_area(self):
    self.scratch_dir = "%s/%sopkg" % (tempfile.gettempdir(),
                                      tempfile.gettempprefix())
    self.file_dir = "%s/files" % (self.scratch_dir)
    self.meta_dir = "%s/meta" % (self.scratch_dir)

    os.mkdir(self.scratch_dir)
    os.mkdir(self.file_dir)
    os.mkdir(self.meta_dir)

def set_package(self, package):
    self.package = package

def get_package(self):
    return self.package

def set_version(self, version):
    self.version = version
    self.parsed_version = parse_version(version)
```

```
    def get_version(self):
        return self.version

    def set_architecture(self, architecture):
        self.architecture = architecture

    def get_architecture(self):
        return self.architecture

    def set_maintainer(self, maintainer):
        self.maintainer = maintainer

    def get_maintainer(self):
        return self.maintainer

    def set_source(self, source):
        self.source = source

    def get_source(self):
        return self.source

    def set_description(self, description):
        self.description = description

    def get_description(self):
        return self.description

    def set_depends(self, depends):
        self.depends = depends

    def get_depends(self, depends):
        return self.depends

    def set_provides(self, provides):
        self.provides = provides

    def get_provides(self, provides):
        return self.provides

    def set_replaces(self, replaces):
        self.replaces = replaces

    def get_replaces(self, replaces):
        return self.replaces

    def set_conflicts(self, conflicts):
        self.conflicts = conflicts

    def get_conflicts(self, conflicts):
        return self.conflicts

    def set_suggests(self, suggests):
        self.suggests = suggests

    def get_suggests(self, suggests):
        return self.suggests

    def set_section(self, section):
        self.section = section

    def get_section(self, section):
        return self.section

    def set_license(self, license):
        self.license = license

    def get_license(self, license):
        return self.license

    def get_file_list_dir(self, directory):
        def check_output(*popenargs, **kwargs):
            """Run command with arguments and return its output as a byte string.

            Backported from Python 2.7 as it's implemented as pure python on stdlib.

            >>> check_output(['/usr/bin/python', '--version'])
            Python 2.6.2
            """
            process = subprocess.Popen(stdout=subprocess.PIPE, *popenargs, **kwargs)
            output, unused_err = process.communicate()
            retcode = process.poll()
            if retcode:
                cmd = kwargs.get("args")
                if cmd is None:
                    cmd = popenargs[0]
                error = subprocess.CalledProcessError(retcode, cmd)
                error.output = output
                raise error
            return output

        if not self.fn:
            try:
                cmd = "find %s -name %s | head -n 1" % (directory, self.filename)
                rc = check_output(cmd, shell=True)
                if rc != "":
                    newfn = str(rc).split()[0]
#                    sys.stderr.write("Package '%s' with empty fn and filename is '%s' was found in '%s', updating fn\n" % (self.package, self
```

```
                self.fn = newfn
            except OSError as e:
                sys.stderr.write("Cannot find current fn for package '%s' filename '%s' in dir '%s'\n(%s)\n" % (self.package, self.filename, d
            except IOError as e:
                sys.stderr.write("Cannot find current fn for package '%s' filename '%s' in dir '%s'\n(%s)\n" % (self.package, self.filename, d
        return self.get_file_list()


    def get_file_list(self):
        if not self.fn:
            sys.stderr.write("Package '%s' has empty fn, returning empty filelist\n" % (self.package))
            return []
        f = open(self.fn, "rb")
        ar = arfile.ArFile(f, self.fn)
        tarStream = ar.open("data.tar.gz")
        tarf = tarfile.open("data.tar.gz", "r", tarStream)
        self.file_list = tarf.getnames()
        self.file_list = map(lambda a: ["./", ""][a.startswith("./")] + a, self.file_list)

        f.close()
        return self.file_list

    def set_package_extension(self, ext="ipk"):
        self.file_ext_opk = ext

    def get_package_extension(self):
        return self.file_ext_opk

    def write_package(self, dirname):
        self._setup_scratch_area()
        file = open("%s/control" % self.meta_dir, 'w')
        file.write(str(self))
        file.close()

        cmd = "cd %s ; tar cvz --format=gnu -f %s/control.tar.gz control" % (self.meta_dir,
                                                              self.scratch_dir)

        cmd_out, cmd_in, cmd_err = os.popen3(cmd)

        while cmd_err.readline() != "":
            pass

        cmd_out.close()
        cmd_in.close()
        cmd_err.close()

        bits = "control.tar.gz"

        if self.file_list:
                cmd = "cd %s ; tar cvz --format=gnu -f %s/data.tar.gz" % (self.file_dir,
                                                              self.scratch_dir)

                cmd_out, cmd_in, cmd_err = os.popen3(cmd)

                while cmd_err.readline() != "":
                    pass

                cmd_out.close()
                cmd_in.close()
                cmd_err.close()

                bits = bits + " data.tar.gz"

        file = "%s_%s_%s.%s" % (self.package, self.version, self.architecture, self.get_package_extension())
        cmd = "cd %s ; tar cvz --format=gnu -f %s/%s %s" % (self.scratch_dir,
                                            dirname,
                                            file,
                                            bits)

        cmd_out, cmd_in, cmd_err = os.popen3(cmd)

        while cmd_err.readline() != "":
            pass

        cmd_out.close()
        cmd_in.close()
        cmd_err.close()

    def compare_version(self, ref):
        """Compare package versions of self and ref"""
        if not self.version:
            print('No version for package %s' % self.package)
        if not ref.version:
            print('No version for package %s' % ref.package)
        if not self.parsed_version:
            self.parsed_version = parse_version(self.version)
        if not ref.parsed_version:
            ref.parsed_version = parse_version(ref.version)
        return self.parsed_version.compare(ref.parsed_version)

    def __str__(self):
        out = ""

        # XXX - Some checks need to be made, and some exceptions
        #       need to be thrown. -- a7r

        if self.package: out = out + "Package: %s\n" % (self.package)
        if self.version: out = out + "Version: %s\n" % (self.version)
```

```
        if self.depends: out = out + "Depends: %s\n" % (self.depends)
        if self.provides: out = out + "Provides: %s\n" % (self.provides)
        if self.replaces: out = out + "Replaces: %s\n" % (self.replaces)
        if self.conflicts: out = out + "Conflicts: %s\n" % (self.conflicts)
        if self.suggests: out = out + "Suggests: %s\n" % (self.suggests)
        if self.recommends: out = out + "Recommends: %s\n" % (self.recommends)
        if self.section: out = out + "Section: %s\n" % (self.section)
        if self.architecture: out = out + "Architecture: %s\n" % (self.architecture)
        if self.maintainer: out = out + "Maintainer: %s\n" % (self.maintainer)
        if self.md5: out = out + "MD5Sum: %s\n" % (self.md5)
        if self.size: out = out + "Size: %d\n" % int(self.size)
        if self.installed_size: out = out + "InstalledSize: %d\n" % int(self.installed_size)
        if self.filename: out = out + "Filename: %s\n" % (self.filename)
        if self.source: out = out + "Source: %s\n" % (self.source)
        if self.description:
            printable_description = textwrap.dedent(self.description).strip()
            out = out + "Description: %s\n" % textwrap.fill(printable_description, width=74, initial_indent=' ', subsequent_indent=' ')
        if self.oe: out = out + "OE: %s\n" % (self.oe)
        if self.homepage: out = out + "HomePage: %s\n" % (self.homepage)
        if self.license: out = out + "License: %s\n" % (self.license)
        if self.priority: out = out + "Priority: %s\n" % (self.priority)
        if self.tags: out = out + "Tags: %s\n" % (self.tags)
        out = out + "\n"

        return out

    def __del__(self):
        # XXX - Why is the `os' module being yanked out before Package objects
        #       are being destroyed?  -- a7r
        pass

class Packages:
    """A currently unimplemented wrapper around the opkg utility."""
    def __init__(self):
        self.packages = {}
        return

    def add_package(self, pkg):
        package = pkg.package
        arch = pkg.architecture
        name = ("%s:%s" % (package, arch))
        if (name not in self.packages):
            self.packages[name] = pkg

        if pkg.compare_version(self.packages[name]) >= 0:
            self.packages[name] = pkg
            return 0
        else:
            return 1

    def read_packages_file(self, fn):
        f = open(fn, "r")
        while True:
            pkg = Package()
            try:
                pkg.read_control(f)
            except TypeError as e:
                sys.stderr.write("Cannot read control file '%s' - %s\n" % (fn, e))
                continue
            if pkg.get_package():
                self.add_package(pkg)
            else:
                break
        f.close()
        return

    def write_packages_file(self, fn):
        f = open(fn, "w")
        names = list(self.packages.keys())
        names.sort()
        for name in names:
            f.write(self.packages[name].__repr__())
        return

    def keys(self):
        return list(self.packages.keys())

    def __getitem__(self, key):
        return self.packages[key]

if __name__ == "__main__":

    assert Version(0, "1.2.2-r1").compare(Version(0, "1.2.3-r0")) == -1
    assert Version(0, "1.2.2-r0").compare(Version(0, "1.2.2+cvs20070308-r0")) == -1
    assert Version(0, "1.2.2+cvs20070308").compare(Version(0, "1.2.2-r0")) == 1
    assert Version(0, "1.2.2-r0").compare(Version(0, "1.2.2-r0")) == 0
    assert Version(0, "1.2.2-r5").compare(Version(0, "1.2.2-r0")) == 1

    package = Package()

    package.set_package("FooBar")
    package.set_version("0.1-fam1")
    package.set_architecture("arm")
    package.set_maintainer("Testing <testing@testing.testing>")
    package.set_depends("libc")
    package.set_description("A test of the APIs. And very long descriptions so often used in oe-core\nfoo\n\n\nbar")

    print("<")
```

```
    sys.stdout.write(str(package))
    print(">")
    f = open("/tmp/control", "w")
    f.write(str(package))
    f.close()

    f = open("/tmp/control", "r")
    package2 = Package()
    package2.read_control(f)
    print("<")
    sys.stdout.write(str(package2))
    print(">")

    package.write_package("/tmp")
```

Notice for package(s)

lsbinitscripts

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code. (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not

compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

  7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

  10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

Notice for package(s)

lsb

LSB version query program (lsb_release) by Dominique Massonie
(mdomi@users.sourceforge.net)

This program forms part of the required functionality of the LSB
(Linux Standard Base) specification.

The program queries the installed state of the distribution to display
certain properties such as the version of the LSB against which the
distribution claims compliance as well. It can also attempt to display
the name and release of the distribution along with an identifier of
who produces the distribution.

Notice for package(s)

lzo

```
/* lzo_init.c -- initialization of the LZO library

   This file is part of the LZO real-time data compression library.

   Copyright (C) 1996-2015 Markus Franz Xaver Johannes Oberhumer
   All Rights Reserved.

   The LZO library is free software; you can redistribute it and/or
   modify it under the terms of the GNU General Public License as
   published by the Free Software Foundation; either version 2 of
   the License, or (at your option) any later version.

   The LZO library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with the LZO library; see the file COPYING.
   If not, write to the Free Software Foundation, Inc.,
   51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

   Markus F.X.J. Oberhumer
   <markus@oberhumer.com>
   http://www.oberhumer.com/opensource/lzo/
 */


#include "lzo_conf.h"


/*************************************************************************
// Runtime check of the assumptions about the size of builtin types,
// memory model, byte order and other low-level constructs.
//
// We are really paranoid here - LZO should either fail
// at startup or not at all.
//
// Because of inlining much of these functions evaluates to nothing.
//
// And while many of the tests seem highly obvious and redundant they are
// here to catch compiler/optimizer bugs. Yes, these do exist.
**************************************************************************/

#if !defined(__LZO_IN_MINILZO)

#define LZO_WANT_ACC_CHK_CH 1
#undef LZOCHK_ASSERT
#include "lzo_supp.h"

    LZOCHK_ASSERT((LZO_UINT32_C(1) << (int)(8*sizeof(LZO_UINT32_C(1))-1)) > 0)
    LZOCHK_ASSERT_IS_SIGNED_T(lzo_int)
    LZOCHK_ASSERT_IS_UNSIGNED_T(lzo_uint)
#if !(__LZO_UINTPTR_T_IS_POINTER)
    LZOCHK_ASSERT_IS_UNSIGNED_T(lzo_uintptr_t)
#endif
    LZOCHK_ASSERT(sizeof(lzo_uintptr_t) >= sizeof(lzo_voidp))
    LZOCHK_ASSERT_IS_UNSIGNED_T(lzo_xint)

#endif
#undef LZOCHK_ASSERT


/*************************************************************************
//
**************************************************************************/

union lzo_config_check_union {
    lzo_uint a[2];
    unsigned char b[2*LZO_MAX(8,sizeof(lzo_uint))];
#if defined(lzo_uint64_t)
    lzo_uint64_t c[2];
#endif
};


#if 0
#define u2p(ptr,off) ((lzo_voidp) (((lzo_bytep)(lzo_voidp)(ptr)) + (off)))
#else
static __lzo_noinline lzo_voidp u2p(lzo_voidp ptr, lzo_uint off)
{
    return (lzo_voidp) ((lzo_bytep) ptr + off);
}
#endif


LZO_PUBLIC(int)
_lzo_config_check(void)
{
#if (LZO_CC_CLANG && (LZO_CC_CLANG >= 0x030100ul && LZO_CC_CLANG < 0x030300ul))
```

```
# if 0
    /* work around a clang 3.1 and clang 3.2 compiler bug; clang 3.3 and 3.4 work */
    volatile
# endif
#endif
    union lzo_config_check_union u;
    lzo_voidp p;
    unsigned r = 1;

    u.a[0] = u.a[1] = 0;
    p = u2p(&u, 0);
    r &= ((* (lzo_bytep) p) == 0);
#if !(LZO_CFG_NO_CONFIG_CHECK)
#if (LZO_ABI_BIG_ENDIAN)
    u.a[0] = u.a[1] = 0; u.b[sizeof(lzo_uint) - 1] = 128;
    p = u2p(&u, 0);
    r &= ((* (lzo_uintp) p) == 128);
#endif
#if (LZO_ABI_LITTLE_ENDIAN)
    u.a[0] = u.a[1] = 0; u.b[0] = 128;
    p = u2p(&u, 0);
    r &= ((* (lzo_uintp) p) == 128);
#endif
    u.a[0] = u.a[1] = 0;
    u.b[0] = 1; u.b[3] = 2;
    p = u2p(&u, 1);
    r &= UA_GET_NE16(p) == 0;
    r &= UA_GET_LE16(p) == 0;
    u.b[1] = 128;
    r &= UA_GET_LE16(p) == 128;
    u.b[2] = 129;
    r &= UA_GET_LE16(p) == LZO_UINT16_C(0x8180);
#if (LZO_ABI_BIG_ENDIAN)
    r &= UA_GET_NE16(p) == LZO_UINT16_C(0x8081);
#endif
#if (LZO_ABI_LITTLE_ENDIAN)
    r &= UA_GET_NE16(p) == LZO_UINT16_C(0x8180);
#endif
    u.a[0] = u.a[1] = 0;
    u.b[0] = 3; u.b[5] = 4;
    p = u2p(&u, 1);
    r &= UA_GET_NE32(p) == 0;
    r &= UA_GET_LE32(p) == 0;
    u.b[1] = 128;
    r &= UA_GET_LE32(p) == 128;
    u.b[2] = 129; u.b[3] = 130; u.b[4] = 131;
    r &= UA_GET_LE32(p) == LZO_UINT32_C(0x83828180);
#if (LZO_ABI_BIG_ENDIAN)
    r &= UA_GET_NE32(p) == LZO_UINT32_C(0x80818283);
#endif
#if (LZO_ABI_LITTLE_ENDIAN)
    r &= UA_GET_NE32(p) == LZO_UINT32_C(0x83828180);
#endif
#if defined(UA_GET_NE64)
    u.c[0] = u.c[1] = 0;
    u.b[0] = 5; u.b[9] = 6;
    p = u2p(&u, 1);
    u.c[0] = u.c[1] = 0;
    r &= UA_GET_NE64(p) == 0;
#if defined(UA_GET_LE64)
    r &= UA_GET_LE64(p) == 0;
    u.b[1] = 128;
    r &= UA_GET_LE64(p) == 128;
#endif
#endif
#if defined(lzo_bitops_ctlz32)
    { unsigned i = 0; lzo_uint32_t v;
    for (v = 1; v != 0 && r == 1; v <<= 1, i++) {
        r &= lzo_bitops_ctlz32(v) == 31 - i;
        r &= lzo_bitops_ctlz32_func(v) == 31 - i;
    }}
#endif
#if defined(lzo_bitops_ctlz64)
    { unsigned i = 0; lzo_uint64_t v;
    for (v = 1; v != 0 && r == 1; v <<= 1, i++) {
        r &= lzo_bitops_ctlz64(v) == 63 - i;
        r &= lzo_bitops_ctlz64_func(v) == 63 - i;
    }}
#endif
#if defined(lzo_bitops_cttz32)
    { unsigned i = 0; lzo_uint32_t v;
    for (v = 1; v != 0 && r == 1; v <<= 1, i++) {
        r &= lzo_bitops_cttz32(v) == i;
        r &= lzo_bitops_cttz32_func(v) == i;
    }}
#endif
#if defined(lzo_bitops_cttz64)
    { unsigned i = 0; lzo_uint64_t v;
    for (v = 1; v != 0 && r == 1; v <<= 1, i++) {
        r &= lzo_bitops_cttz64(v) == i;
        r &= lzo_bitops_cttz64_func(v) == i;
    }}
#endif
#endif
    LZO_UNUSED_FUNC(lzo_bitops_unused_funcs);

    return r == 1 ? LZO_E_OK : LZO_E_ERROR;
}
```

```
/*************************************************************************
//
*************************************************************************/

LZO_PUBLIC(int)
__lzo_init_v2(unsigned v, int s1, int s2, int s3, int s4, int s5,
                          int s6, int s7, int s8, int s9)
{
    int r;

#if defined(__LZO_IN_MINILZO)
#elif (LZO_CC_MSC && ((_MSC_VER) < 700))
#else
#define LZO_WANT_ACC_CHK_CH 1
#undef LZOCHK_ASSERT
#define LZOCHK_ASSERT(expr)  LZO_COMPILE_TIME_ASSERT(expr)
#include "lzo_supp.h"
#endif
#undef LZOCHK_ASSERT

    if (v == 0)
        return LZO_E_ERROR;

    r = (s1 == -1 || s1 == (int) sizeof(short)) &&
        (s2 == -1 || s2 == (int) sizeof(int)) &&
        (s3 == -1 || s3 == (int) sizeof(long)) &&
        (s4 == -1 || s4 == (int) sizeof(lzo_uint32_t)) &&
        (s5 == -1 || s5 == (int) sizeof(lzo_uint)) &&
        (s6 == -1 || s6 == (int) lzo_sizeof_dict_t) &&
        (s7 == -1 || s7 == (int) sizeof(char *)) &&
        (s8 == -1 || s8 == (int) sizeof(lzo_voidp)) &&
        (s9 == -1 || s9 == (int) sizeof(lzo_callback_t));
    if (!r)
        return LZO_E_ERROR;

    r = _lzo_config_check();
    if (r != LZO_E_OK)
        return r;

    return r;
}


#if !defined(__LZO_IN_MINILZO)
#include "lzo_dll.ch"
#endif


/* vim:set ts=4 sw=4 et: */
```

Notice for package(s)

lldpd

```
/*
 * Copyright (c) 2008 Vincent Bernat <bernat@luffy.cx>
 *
 * Permission to use, copy, modify, and/or distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 */
```

Notice for package(s)

v2lin

```
        Copyright (C) 2000  Monta Vista Software Inc.
        Copyright (C) 2004, 2005, 2006 v2lin Team <http://v2lin.sf.net>

        This file is part of the v2lin Library.

        Initial implementation Gary S. Robertson, 2000, 2001.
        Contributed by Andrew Skiba, skibochka@sourceforge.net, 2004-09.
        Contributed by Mike Kemelmakher, mike@ubxess.com, 2005-08.
```

Contributed by Constantine Shulyupin, conan.sh@gmail.com, 2006.

The v2lin library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

In addition to the permissions in the GNU Lesser General Public
License, the Free Software Foundation gives you unlimited
permission to link the compiled version of this file with other
programs, and to distribute those programs without any restriction
coming from the use of this file. (The GNU Lesser General Public
License restrictions do apply in other respects; for example, they
cover modification of the file, and distribution when not linked
into another program.)

Note that people who make modified versions of this file are not
obligated to grant this special exception for their modified
versions; it is their choice whether to do so. The GNU Lesser
General Public License gives permission to release a modified
version without this exception; this exception also makes it
possible to release a modified version which carries forward this
exception.

The v2lin Library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with the v2lin Library; if not get it from
http://www.gnu.org/licenses/licenses.html#LGPL

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and

distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices
    stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no
    charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application. Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object

file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

```
                    END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

Notice for package(s)

zlib

```
 (C) 1995-2004 Jean-loup Gailly and Mark Adler

  This software is provided 'as-is', without any express or implied
  warranty.  In no event will the authors be held liable for any damages
  arising from the use of this software.

  Permission is granted to anyone to use this software for any purpose,
  including commercial applications, and to alter it and redistribute it
  freely, subject to the following restrictions:

  1. The origin of this software must not be misrepresented; you must not
     claim that you wrote the original software. If you use this software
     in a product, an acknowledgment in the product documentation would be
     appreciated but is not required.
  2. Altered source versions must be plainly marked as such, and must not be
     misrepresented as being the original software.
  3. This notice may not be removed or altered from any source distribution.

  Jean-loup Gailly        Mark Adler
  jloup@gzip.org          madler@alumni.caltech.edu
```

Notice for package(s)

Apache HTTP Server

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
```

```
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Apache" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
* Portions of this software are based upon public domain software
* originally written at the National Center for Supercomputing Applications,
* University of Illinois, Urbana-Champaign.
*/
```

Notice for package(s)

Apache-HTTP Server

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
```

editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,

```
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

Notice for package(s)

2d3D, LSIIT Laboratory, MontaVista Software, Ted Lemon, The Regents of the University of California, The Regents of the University of California, The Regents of the University of California, The Regents of the University of California, VxWorks, WIDE Project, WIDE Project

---

```
BSD 1.0
Copyright (c) 1982, 1986, 1990, 1991, 1993
The Regents of the University of California. All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must
display the following acknowledgement:
This product includes software developed by the University of California, Berkeley
and its contributors.
4. Neither the name of the University nor the names of its contributors may be used
to endorse or promote products derived from this software without specific prior
written permission.


THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
```

```
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Notice for package(s)

bcrypt - blowfish file encryption, flex: the fast lexical analyser, FreeBSD, IP Utils, IPMItool, Net SNMP - net-snmp, Net SNMP - net-snmp, NetBSD, NetBSD, NetBSD, NetBSD, Open BSD, OpenSSH, OpenSSH, OpenSSH, S Trace, Sudo, TCP Dump, The libpcap project, The libpcap project, The Regents of the University of California, Trace Route

---

```
Copyright (c) <year>, <copyright holder>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
   This product includes software developed by the <organization>.
4. Neither the name of the <organization> nor the
   names of its contributors may be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY <COPYRIGHT HOLDER> ''AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Notice for package(s)

Bzip2

```
bzip2 License
This program, "bzip2", the associated library "libbzip2", and all documentation,
are copyright (C) 1996-2005 Julian R Seward. All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim
that you wrote the original software. If you use this software in a product, an
acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented
as being the original software.
4. The name of the author may not be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Notice for package(s)

Carnegie Mellon University

```
Carnegie Mellon University License
CMU libsasl
Tim Martin
Rob Earhart
Rob Siemborski


Copyright (c) 2001 Carnegie Mellon University. All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

3. The name "Carnegie Mellon University" must not be used to endorse or promote
products derived from this software without prior written permission. For
permission or any other legal details, please contact

Office of Technology Transfer
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
(412) 268-4387, fax: (412) 268-7395
tech-transfer@andrew.cmu.edu

4. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/)."

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Notice for package(s)

Cron

Cron License
Copyright 1988,1990,1993,1994 by Paul Vixie
All rights reserved
Distribute freely, except: don't remove my name from the source or documentation
(don't take credit for my work), mark your changes (don't get me blamed for your
possible bugs), don't alter or remove this notice. May be sold if buildable source
is provided to buyer. use at your own risk, responsibility for damages (if any) to
anyone resulting from the use of this software rests entirely with the user.
Send bug reports, bug fixes, enhancements, requests, flames, etc., and I'll try to
keep a version up to date. I can be reached as follows:

Paul Vixie <paul@vix.com> uunet!decwrl!vixie!paul

Notice for package(s)

Bee Crypt Cryptography Library, BeeCrypt Cryptography Library, CrackLib, OpenSAF, Redblack Balanced Tree, Simple DirectMedia Layer - SDL,

---

                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave

you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

  Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

  When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

  We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

  For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

  Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation

and installation of the library.

  Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

  You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

  Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not

compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

    a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

    e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

  It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

  7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined

library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

    b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

  11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice
```

That's all there is to it!

Notice for package(s)

Gregory M. Christy

```
Copyright (c) 1991 Gregory M. Christy
All rights reserved.

Redistribution and use in source and binary forms are permitted
provided that the above copyright notice and this paragraph are
duplicated in all such forms and that any documentation,
advertising materials, and other materials related to such
distribution and use acknowledge that the software was developed
by the author.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
```

Notice for package(s)

IBM Corporation

```
This source code has been made available to you by IBM on an AS-IS
basis. Anyone receiving this source is licensed under IBM
copyrights to use it in any way he or she deems fit, including
copying it, modifying it, compiling it, and redistributing it either
with or without modifications. No license under IBM patents or
patent applications is to be implied by the copyright license.

Any user of this software should understand that IBM cannot provide
technical support for this software and will not be responsible for
any consequences resulting from the use of this software.

Any person who transfers this source code or any derivative work
must include the IBM copyright notice, this paragraph, and the
preceding two paragraphs in the transferred software.

COPYRIGHT I B M CORPORATION 1995
LICENSED MATERIAL - PROGRAM PROPERTY OF I B M
```

Notice for package(s)

Intel Corporation

```
Copyright (c) 1990,1991 Intel Corporation

Intel hereby grants you permission to copy, modify, and
distribute this software and its documentation. Intel grants
this permission provided that the above copyright notice
appears in all copies and that both the copyright notice and
this permission notice appear in supporting documentation. In
addition, Intel grants this permission provided that you
prominently mark as not part of the original any modifications
made to this software or documentation, and that the name of
Intel Corporation not be used in advertising or publicity
pertaining to distribution of the software or the documentation
without specific, written prior permission.

Intel Corporation does not warrant, guarantee or make any
representations regarding the use of, or the results of the use
of, the software and documentation in terms of correctness,
accuracy, reliability, currentness, or otherwise; and you rely
on the software, documentation and results solely at your own risk.
```

Notice for package(s)

MIPS Computer Systems

```
Copyright (c) 1992, 1991, 1990 MIPS Computer Systems, Inc.
MIPS Computer Systems, Inc. grants reproduction and use
rights to all parties, PROVIDED that this comment is
maintained in the copy.
```

Notice for package(s)

RSA Data Security, Inc. MD5 Message-Digest Algorithm

```
Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that
it is identified as the "RSA Data Security, Inc. MD5 Message-
Digest Algorithm" in all material mentioning or referencing this
software or this function.

License is also granted to make and use derivative works
provided that such works are identified as "derived from the RSA
Data Security, Inc. MD5 Message-Digest Algorithm" in all
material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning
either the merchantability of this software or the suitability
of this software for any particular purpose. It is provided "as
is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this
documentation and/or software.
```

Notice for package(s)

RSVPD - ReSerVation Protocol Daemon

```
Copyright (c) 1996 by the University of Southern California
All rights reserved.
```

```
Permission to use, copy, modify, and distribute this software and its
documentation in source and binary forms for any purpose and without
fee is hereby granted, provided that both the above copyright notice
and this permission notice appear in all copies. and that any
documentation, advertising materials, and other materials related to
such distribution and use acknowledge that the software was developed
in part by the University of Southern California, Information
Sciences Institute. The name of the University may not be used to
endorse or promote products derived from this software without
specific prior written permission.

THE UNIVERSITY OF SOUTHERN CALIFORNIA makes no representations about
the suitability of this software for any purpose. THIS SOFTWARE IS
PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Other copyrights might apply to parts of this software and are so
noted when applicable.
```

Notice for package(s)

Sun Microsystems

```
Developed at SunPro, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this
software is freely granted, provided that this notice
is preserved.
```

Notice for package(s)

Todd C. Miller

```
Copyright (c) 1998 Todd C. Miller Permission to use, copy,
modify, and distribute this software for any purpose with or
without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies. THE
SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE
FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY
DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE
OF THIS SOFTWARE.
```

Notice for package(s)

University of Oregon

```
Copyright (c) 1998 by the University of Oregon.
All rights reserved.

Permission to use, copy, modify, and distribute this software and
its documentation in source and binary forms for lawful
purposes and without fee is hereby granted, provided
that the above copyright notice appear in all copies and that both
the copyright notice and this permission notice appear in supporting
documentation, and that any documentation, advertising materials,
and other materials related to such distribution and use acknowledge
that the software was developed by the University of Oregon.
The name of the University of Oregon may not be used to endorse or
promote products derived from this software without specific prior
written permission.

THE UNIVERSITY OF OREGON DOES NOT MAKE ANY REPRESENTATIONS
ABOUT THE SUITABILITY OF THIS SOFTWARE FOR ANY PURPOSE. THIS SOFTWARE IS
PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND
NON-INFRINGEMENT.

IN NO EVENT SHALL UO, OR ANY OTHER CONTRIBUTOR BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, WHETHER IN CONTRACT,
TORT, OR OTHER FORM OF ACTION, ARISING OUT OF OR IN CONNECTION WITH,
THE USE OR PERFORMANCE OF THIS SOFTWARE.

Other copyrights might apply to parts of this software and are so
noted when applicable.
```

Notice for package(s)

University of Southern California

Notice for package(s)

US Naval Research Laboratory (NRL)

Notice for package(s)

OSSP mm, OSSP mm

---

3. All advertising materials mentioning features or use of this software
must display the following acknowledgment:
"This product includes software developed by Ralf S. Engelschall ."
4. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by Ralf S. Engelschall ."
THIS SOFTWARE IS PROVIDED BY RALF S. ENGELSCHALL "AS IS" AND ANY EXPRESSED
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL RALF S. ENGELSCHALL OR ITS CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notice for package(s)

AbsoluteValue Software, Inc.

MOZILLA PUBLIC LICENSE
Version 1.1

1. Definitions.
1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. ''Contributor'' means each entity that creates or contributes to the creation of Modifications.

1.2. ''Contributor Version'' means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made

1.3. ''Covered Code'' means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case includi

1.4. ''Electronic Distribution Mechanism'' means a mechanism generally accepted in the software development community for the electronic trans

1.5. ''Executable'' means Covered Code in any form other than Source Code.

1.6. ''Initial Developer'' means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. ''Larger Work'' means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. ''License'' means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently

1.9. ''Modifications'' means any addition to or deletion from the substance or structure of either the Original Code or any previous Modificat
A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.
1.10. ''Original Code'' means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Orig

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparat

1.11. ''Source Code'' means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any

1.12. "You'' (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License o
2. Source Code License.
2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, pe

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sa

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from t
2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, no
 (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perfor

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combinatio

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Ver
3. Distribution Obligations.
3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, includir

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form unde

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made

3.4. Intellectual Property Matters
(a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise

(b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent lice
 (c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are C
3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice ir

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distri
4. Inability to Comply Due to Statute or Regulation.
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, ju
5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.
6. Versions of the License.
6.1. New Versions. Netscape Communications Corporation (''Netscape'') may publish revised and/or new versions of the License from time to time

6.2. Effect of New Versions. Once Covered Code has been published under a particular version of the License, You may always continue to use it

6.3. Derivative Works. If You create or use a modified version of this License (which you may only do in order to apply it to code which is no
7. DISCLAIMER OF WARRANTY. COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS'' BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR
8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then an

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which
9. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHA
The Covered Code is a ''commercial item,'' as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ''commercial computer softwar
11. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable,
12. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of it
13. MULTIPLE-LICENSED CODE.
Initial Developer may designate portions of the Covered Code as Multiple-Licensed. Multiple-Licensedmeans that the Initial Developer permits y
EXHIBIT A -Mozilla Public License.
``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliа
Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the Li
The Original Code is _____.
The Initial Developer of the Original Code is _____. Portions created by _____ are Copyright (C) _____ __
Contributor(s): _____.
Alternatively, the contents of this file may be used under the terms of the _____ license (the [___] License), in which case the provisions of
[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should u

---

Notice for package(s)

Netperf

Copyright (C) 1993 Hewlett-Packard Company
ALL RIGHTS RESERVED.

The enclosed software and documentation includes copyrighted works of Hewlett-Packard Co. For as long as you comply with the following limitat
1. The enclosed software and documentation is made available at no charge in order to advance the general development of high-performance netw
2. You may not delete any copyright notices contained in the software or documentation. All hard copies, and copies in source code or object c
3. The enclosed software and documentation has not been subjected to testing and quality control and is not a Hewlett-Packard Co. product. At
4. THE SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS". HEWLETT-PACKARD COMPANY DOES NOT WARRANT THAT THE USE, REPRODUCTION, MODIFICATION OR DI
5. HEWLETT-PACKARD COMPANY WILL NOT IN ANY EVENT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING I

Notice for package(s)

OpenSSL, OpenSSL

LICENSE ISSUES
==============
The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the t
OpenSSL License

==
Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the docum
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of
6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"
THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED W
==
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (

Notice for package(s)

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.
This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.
This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply

If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.
This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the docum
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
"This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowled
THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES O
The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot si

---

Notice for package(s)

Sun RPC

Sun RPC License
Sun RPC is a product of Sun Microsystems, Inc. and is provided for unrestricted use provided that this legend is included on all tape media ar
SUN RPC IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURP
Sun RPC is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modific
SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY SUN RPC OR ANY
In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even i
Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, California 94043

Notice for package(s)

Tcl/Tk

The following terms apply to all versions of the core Tcl/Tk releases, the Tcl/Tk browser plug-in version 2.0, and TclBlend and Jacl version 1
TCL/TK LICENSE TERMS
This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties.
The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provide
IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISI
THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, F
GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the