Go to the first, previous, next, last section, table of contents.

1. Introduction

1.1 Overview of GNU Parted

GNU Parted is a program for creating, destroying, resizing, checking and copying partitions, and the file systems on them.

This is useful for creating space for new operating systems, reorganising disk usage, copying data between hard disks, and "disk imaging" -- replicating installations over many computers.

This documentation assumes knowledge of partitions and file systems. If you want to learn more about these, the Partition mini-HOWTO is recommended reading. It is probably included with your distribution, or it is available at

http://www.linuxdoc.org/HOWTO/mini/Partition/index.html

GNU Parted was designed to minimize the chance of data loss. For example, it was designed to avoid data loss during interruptions (like power failure) and performs many safety checks. However there could be bugs in Parted, so you should back up your important files.

The GNU Parted homepage is www.gnu.org/software/parted. It can be downloaded from ftp.gnu.org/gnu/parted.

The Parted mailing list is parted@gnu.org. To subscribe, write to bug-parted-request@gnu.org with `` `subscribe' `` in the subject. Subscription information and archives are available at:

http://mail.gnu.org/mailman/listinfo/bug-parted

Please send bug reports to bug-parted@gnu.org. When sending bug reports, please include the version of GNU Parted. If the bug is related to partition tables, then please include the output from these commands:

```
# fdisk /dev/hda -l
# fdisk /dev/hda
Command (m for help): p
Command (m for help): x
Extended command (m for help): p
```

Feel free to ask for help on this list -- just check that your question isn't answered here first. If you don't understand the documentation, please tell us, so we can explain it better. General philosophy is: if you need to ask for help, then something needs to be fixed so you (and others) don't need to ask for help.

Also, we'd love to hear your ideas :-)

1.2 Software Required for the use of Parted

GNU Parted depends on the following packages to build correctly:

- libuuid, part of the e2fsprogs package. If you don't have this, you can get it from: http://web.mit.edu/tytso/www/linux/e2fsprogs.html If you want to compile Parted and e2fsprogs, note that you will need to `make install` and `make install-libs` e2fsprogs.
- GNU Readline (optional), available from ftp://ftp.gnu.org/gnu/readline If you are compiling Parted, and you don't have readline, you can disable Parted's readline support with the `--disable-readline` option for @command{configure}.

- GNU gettext (or compatible software) for compilation, if internationalisation support is desired. ftp://ftp.gnu.org/gnu/gettext
- libreiserfs, if you want reiserfs support: http://reiserfs.osdn.org.ua Note that parted will automatically detect libreiserfs at runtime, and enable reiserfs support. libreiserfs is new, and hasn't been widely tested yet.

1.3 Platforms on which GNU Parted runs

Hopefully, this list will grow a lot. If you do not have one of these platforms (Linux at the moment!), then you can use a boot disk. See section 1.6 Using a Parted Boot Disk.

GNU/Linux
: Linux versions 2.0 and up, on Alpha, x86 PCs, PC98, Macintosh PowerPC, Sun hardware.

GNU/Hurd
: Experimental support.

GNU libc 2.1 or higher is required. You can probably use older versions by using the `` `--disable-nls' `` option. See section 1.5 Building GNU Parted. (Note: I think we have now dropped this requirement. TODO: check if libc 2.0 works!)

1.4 Terms of distribution for GNU Parted

GNU Parted is free software, covered by the GNU General Public License Version 2. This should have been included with the Parted distribution, in the COPYING file. If not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Libparted is considered part of GNU Parted. It is covered by the GNU General Public License. It is NOT released under the GNU Lesser General Public License (LGPL).

1.5 Building GNU Parted

If you want to compile GNU Parted, this is generally done with:

```
$ ./configure
$ make
```

However, there are a few options for @command{configure}:

`--without-readline`
: turns off use of readline. This is useful for making rescue disks, etc., where few libraries are available.

`--disable-debug`
: don't include assertions

`--disable-dynamic-loading`
: disables dynamic loading of some libraries (only libreiserfs for now, although we hope to expand this). Dynamic loading is useful because it allows you to reuse libparted shared libraries even when you don't know when if some libraries will be available. It has a small overhead (mainly linking with libdl), so it may be useful to disable it on bootdisks if you don't need the flexibility.

`--disable-fs`
: disable all filesystem support

`--disable-nls`
: turns off native language support. This is useful for use with old versions of glibc, or a trimmed down version of glibc suitable for rescue disks.

`--disable-shared`
: turns off shared libraries. This may be necessary for use with old versions of GNU libc, if you get a compile error about a "spilled register". Also useful for boot/rescue disks.

`--disable-Werror`
: ignore warning messages in compilation

`--enable-all-static`

builds the Parted binary as a fully static binary. This is convienient for boot disks, because you don't need to install any libraries on the boot disk (although, other programs may require them...) Note: you will also want to run strip(1).

`--enable-discover-only`
: support only reading/probing

`--enable-mtrace`
: enable malloc() debugging

`--enable-read-only`
: disable writing (for debugging)

1.6 Using a Parted Boot Disk

If you want run Parted on a machine without GNU/Linux installed, or you want to resize a root or boot partition, you will need to use a boot disk.

A boot disk image is available from:

ftp://ftp.gnu.org/gnu/parted/bootdisk/partboot.img

To actually create the boot disk, the disk image must be written to a floppy disk. In GNU/Linux, this can be done with

```
# dd if=partboot.img of=/dev/fd0 bs=1440k
```

Or use RAWRITE.EXE under DOS.

Unfortunately, the boot disk doesn't support a very wide range of hardware. If your hard disk isn't supported, then you will need to make your own boot disk. You can copy the parted binary from the parted bootdisk onto another disk, or try other boot disks, or make your own. You may find mkparted useful, which is a shell script to make custom parted boot disks. It is available at:

ftp://ftp.tux.org/pub/people/kent-robotti/mkparted

To copy parted from the boot disk onto another disk:

1. Boot from the Parted boot disk.
2. Insert another ext2-formatted floppy disk. If it isn't formated, you can create a file system with, for example:

   ```
   $ parted /dev/fd0 mklabel loop mkpartfs primary ext2 0 1.4
   ```

3. Mount the floppy disk, e.g.,

   ```
   $ mount -t ext2 /dev/fd0 /mnt/floppy
   ```

4. Copy `` `/sbin/parted' `` to the floppy, e.g.,

   ```
   $ cp /sbin/parted /mnt/floppy
   ```

5. Copy `` `/lib/*' `` to the floppy, e.g.,

   ```
   $ cp /lib/* /mnt/floppy
   ```

6. Unmount the floppy, e.g.,

   ```
   $ umount /mnt/floppy
   ```

7. Find a boot disk that *does* support your hard disk. (Hint: try looking for rescue disks from various distributions on big mirror sites)
8. Boot off your rescue disk. Mount the disk you copied Parted onto.
9. Run Parted. For example,

```
# cd /mnt/floppy
# LD_LIBRARY_PATH=. ./parted
```

Go to the first, previous, next, last section, table of contents.