the GNU Parted Manual

GNU Parted, version 1.6.1, 13 May 2002

Andrew Clausen clausen@gnu.org
Richard M. Kreuter kreuter@anduril.rutgers.edu

Short 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 `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 [Parted boot disks], page 3.

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 [Compiling], page 2. (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 `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
   ```

2 Using Parted

2.1 Partitioning Concepts

Unfortunately, partitioning your disk is rather complicated. This is because there are interactions between many different systems that need to be taken into consideration:

- The BIOS or firmware - the program that is built into a ROM chip inside your computer, that does memory checks, etc. You can not (easily) change programs in this system. Examples of BIOS or firmware programs: AmiBIOS, Award, Phoenix, OpenFirmware. You will only have one of these programs.
- The boot loader - the program that allows you to select which operating system you want to use, and loads that operating system. Examples: LILO, GRUB, Yaboot, Quik. You may have more than one boot loader installed, especially if you have more than one type of operating system installed.
- The operating system (at the moment, this must be GNU/Linux) that runs Parted, and the other operating systems that you use.
- The file system types - the way the data stored on partitions. Examples of these are: ext2, fat32, hfs, reiserfs. You will often have partitions of of different file system types.

Parted suppports many combinations of BIOS, boot loader, operating system, and file systems, and will support more in the future. To further understand the roles of each, please see Chapter 3 [BIOSes and Firmware], page 17, Chapter 4 [Boot Loaders], page 21, Chapter 5 [Operating Systems], page 25, and Chapter 6 [File Systems], page 27.

This chapter describes how to use Parted, which is largely the same, no matter what systems you are using. You should read this chapter, then each of chapters on BIOSes, boot loaders, operating systems, and file systems. However, you only need to read the sections that are relevant to you. For example, if you are only using LILO as your boot loader, then you only need to read the introduction, and Section 4.1 [LILO], page 21.

2.2 Using GNU Parted

Parted has two modes: command line and interactive. Parted should always be started with:

```
# parted device
```

where *device* is the hard disk device to edit. (If you're lazy, Parted will attempt to guess which device you want.)

In command line mode, this is followed by one or more commands. For example:

```
# parted /dev/sda resize 1 52 104 mkfs 2 fat16
```

Options (like `--help`) can only be specified on the command line.

In interactive mode, commands are entered one at a time at a prompt, and modify the disk immediately. For example:

```
(parted) resize 1 52.0005 104.5
(parted) mkfs 2 fat16
```

Unambiguous abbreviations are allowed. For example, you can type "p" instead of "print", and "re" instead of "resize". Commands can be typed in, either in English, or your native language (if your language has been translated). This may create ambiguities.

Also note that you can specify decimal places in the numbers corresponding to partition locations (in megabytes). Negative numbers count from the end of the disk, with "-0" being the end of the disk.

If you don't give a parameter to a command, Parted will ask you for it. For example:

```
(parted) resize 1
Start? 0
End? 400
```

Parted will always warn you before doing something that is potentially dangerous, unless it's something that's obviosuly dangerous (i.e. rm, mklabel, mkfs) For example, if you attempt to shrink a partition "too much" (i.e. by more than the free space available), Parted will automatically resize to the minimum it can without losing data. If this minimum is significantly different, it will warn you that it's doing something significanlty different to what you asked. Since many partitioning systems have complicated constraints, Parted will usually do something slightly different to what you asked. (For example, create a partitiong starting at 10.352, not 10.4)

2.3 Command Line Options

When invoked from the command line, parted supports the following syntax:

```
# parted [option] device [command [argument]]
```

Available options and commands follow. For detailed explanations of the use of parted commands, see Section 2.4 [Command explanations], page 6. Options begin with a hyphen, commands do not:

Options:

'`-h`'
'`--help`' display a help message

'`-i`'
'`--interactive`'
where necessary, prompt for user intervention

'`-s`'
'`--script`'
never prompt for user intervention

'`-v`'
'`--version`'
display the version

2.4 Parted Session Commands

GNU Parted provides the following commands:

2.4.1 check

check *minor* Command

Checks if the file system on partition *minor* has any errors.

Example:

```
(parted) check 1
```

Check the file system on partition 1.

2.4.2 cp

cp [*from-device*] *from-minor to-minor* Command

Copies the file system on the partition *from-minor* to partition *to-minor*, deleting the original contents of the destination partition.

An optional device parameter, *from-device* can be given, which specifies which device the source partition is on.

Supported file systems:

ext2, ext3 (provided the destination partition is larger than the source partition)

fat16, fat32

linux-swap (equivalent to mkswap on destination partition)

reiserfs (if libreiserfs is installed)

Example:

```
(parted) cp /dev/hdb 2 3
```

Copy partition 2 of '`/dev/hdb`' (i.e. '`/dev/hdb2`') to partition on 3, on the device Parted was loaded with, destroying the original contents of partition 3.

2.4.3 help

help [*command*] Command

Prints general help, or help on *command*.

Example:

```
(parted) help resize
```

Print help for the resize command.

2.4.4 mklabel

mklabel *label-type* Command

Creates a new disk label, of type *label-type*. The new disk label will have no partitions. This command (normally) won't technically destroy your data, but it will make it will make it basically unusable, and you will need to use the rescue command (see Chapter 9 [Related Software and Info], page 35) to recover any partitions. Gpart only works for msdos disk labels (AFAIK), but is much better than parted at recovering partitions. Parted works on all partition tables.[1]

label-type must be one of these supported disk labels:

bsd

loop (raw disk access)

gpt

mac

msdos

pc98

sun

Example:

```
(parted) mklabel msdos
```

Create an msdos style disklabel.

2.4.5 mkfs

mkfs *minor fs-type* Command

Makes a file system *fs-type* on partition *minor*, destroying all data that resides on that partition.

Supported file systems:

ext2

mips

fat16

fat32

linux-swap

reiserfs (if libreiserfs is installed)

Example:

```
(parted) mkfs 2 fat32
```

Make a *fat32* file system on partition 2.

2.4.6 mkpart

mkpart *part-type* [*fs-type*] *start end* Command

Creates a new partition, *without* creating a new file system on that partition. This is useful for creating partitions for file systems (or LVM, etc.) that Parted doesn't support. You may specify a file system type, to set the appropriate partition code in

[1] Everyone seems to have a different word for "disk label" — these are all the same thing: partition table, partition map. Also, the Master Boot Record on x86 machines is stored in the same sector as Partition Tables (you don't need to know this to use Parted).

the partition table for the new partition. *fs-type* is required for data partitions (i.e., non-extended partitions). *start* and *end* are the offset from the beginning of the disk, that is, the "distance" from the start of the disk.

part-type is one of: primary, extended, logical. Extended and logical are only used for msdos and mips disk labels.

fs-type must be on of these supported file systems:

ext2

fat32

fat16

HFS

linux-swap

NTFS

reiserfs

ufs

Example:

```
(parted) mkpart logical 0.0 692.1
```

Create a logical partition that will contain an ext2 filesystem. The partition will start at the beginning of the disk, and end 692.1 megabytes into the disk.

2.4.7 mkpartfs

mkpartfs *part-type fs-type start end* Command

Creates a new partition of type *part-type* with a new file system of type *fs-type* on it. The new partition will start *start* megabytes, and end *end* megabytes from the beginning of the disk. Do not use this command to recover a deleted partition (use mkpart instead).

part-type is one of: primary, extended, logical. Extended and logical are only used for msdos and mips disk labels.

fs-type must be one of these supported file systems:

ext2

fat32

fat16

linux-swap

reiserfs (if libreiserfs is installed)

Example:

```
(parted) mkpartfs logical ext2 440 670
```

Make a logical partition and write an ext2 file system, starting 440 megabytes and ending 670 megabytes from the beginning of the disk.

2.4.8 move

move *minor start* [*end*] Command

Moves partition on the disk, by moving its beginning to *start*. Note: move never changes the minor number.

If no *end* is given, the partition's size remains the same.

Supported file systems:

ext2, ext3 (provided the destination partition is larger than the source partition)

fat32

fat16

linux-swap

reiserfs (if libreiserfs is installed)

Example:

```
(parted) move 2 150
```

Move partition with minor number 2 so that it begins 150 megabytes from the start of the disk.

2.4.9 name

name *minor name* Command

Sets the name for the partition *minor* (Mac and PC98 only). The name can be placed in quotes.

Example:

```
(parted) name 2 'Secret Documents'
```

Set the name of partition 2 to 'Secret Documents'.

2.4.10 print

print Command

Displays the partition table on the device parted is editing.

Example:

```
(parted) print
Disk geometry for /dev/hda: 0.000-2445.679 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.031    945.000  primary   FAT         boot, lba
2        945.000   2358.562  primary   ext2
3       2358.562   2445.187  primary   linux-swap
```

2.4.11 quit

quit Command

Quits Parted.

It is only after Parted exits that the Linux kernel knows about the changes Parted has made to the disks. However, the changes caused by typing your commands will *probably* be made to the disk immediately after typing a command. However, Linux's cache, and the disk's hardware cache may delay this.

2.4.12 rescue

rescue *start end* Command

rescue a lost partition that used to be about *start* and *end*

Looks for file system signatures around *start* and *end*. If one is found, it will ask you if you want to create a partition for it. This is useful if you accidently deleted a partition with parted's rm command, for example.

Example:

```
(parted) print
Disk geometry for /dev/hdc: 0.000-8063.507 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.031   8056.032  primary   ext3
(parted) rm
Partition number? 1
(parted) print
Disk geometry for /dev/hdc: 0.000-8063.507 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
```

OUCH! We deleted our ext3 partition!!! Parted comes to the rescue...

```
(parted) rescue
Start? 0
End? 8056
Information: A ext3 primary partition was found at 0.031Mb ->
8056.030Mb.  Do you want to add it to the partition table?
Yes/No/Cancel? y
(parted) print
Disk geometry for /dev/hdc: 0.000-8063.507 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.031   8056.032  primary   ext3
```

It's back! :)

2.4.13 resize

resize *minor start end* Command

Resizes the partition with number *minor*. The partition will start *start* from the beginning of the disk, and end *end* from the beginning of the disk. resize never changes the minor number. Extended partitions can be resized, so long as the new extended partition completely contains all logical partitions.

Note that Parted does not require a file system to be "defragged" (Parted can safely move data around if necessary). It's a waste of time defragging. Don't bother!

Supported file systems:

ext2, ext3 - restriction: the new *start* must be the same as the old *start*.

fat16, fat32

linux-swap

reiserfs (if libreiserfs is installed)

Example:

```
(parted) resize 3 200 850
```

Resize partition 3, so that it begins 200 megabytes and ends 850 megabytes from the beginning of the disk.

2.4.14 rm

rm *minor* Command

Removes the partition with number *minor*. If you accidently delete a partition with this command, use mkpart (*not* mkpartfs) to recover it. Also, you can use the gpart program (see Chapter 9 [Related Software and Info], page 35) to recover damaged disk labels.

Note for msdos disk labels: if you delete a logical partition, all logical partitions with a larger minor number will be renumbered. For example, if you delete a logical partition with a minor number of 6, then logical partitions that were number 7, 8 and 9 would be renumbered to 6, 7 and 8 respectively. This means, for example, that you have to update '`/etc/fstab`' on GNU/Linux systems.

Example:

```
(parted) rm 3
```

Remove partition 3.

2.4.15 select

select *device* Command

Selects the device, *device*, for Parted to edit. The device will usually be a Linux hard disk device, or, if direct access to a file system is required — a partition, software RAID device, or LVM logical volume.

Example:

```
(parted) select /dev/hdb
```

Select '`/dev/hdb`' (the slave device on the first ide controller on Linux) as the device to edit.

2.4.16 set

set *minor flag state* Command

Changes a flag on the partition with number *minor*. A flag can be either "on" or "off". Some or all of these flags will be available, depending on what disk label you are using:

'`boot`' (Mac, MSDOS, PC98) - should be enabled if you want to boot off the partition. The semantics vary between disk labels. For MSDOS disk labels, only one partition can be bootable. If you are installing LILO on a partition (see Section 4.1 [LILO], page 21), then that partition must be bootable. For PC98 disk labels, all ext2 partitions must be bootable (this is enforced by Parted).

'`lba`' (MSDOS) - this flag can be enabled, to tell MS DOS, MS Windows 9x and MS Windows ME based operating systems to use Linear (LBA) mode.

'`root`' (Mac) - this flag should be enabled if the partition is the root device to be used by Linux.

'`swap`' (Mac) - this flag should be enabled if the partition is the swap device to be used by Linux.

'`hidden`' (MSDOS, PC98) - this flag can be enabled to hide partitions from Microsoft operating systems.

'`raid`' (MSDOS) - this flag can be enabled to tell linux the partition is a software RAID partition See Chapter 7 [LVM and RAID], page 30.

'`LVM`' (MSDOS) - this flag can be enabled to tell linux the partition is a physical volume.

The print command displays all enabled flags for each partition.

Example:

```
(parted) set 1 boot on
```

Set the '`boot`' flag on partition 1.

2.5 Example Parted Sessions

These examples attempt to cover the most common circumstances, with the exception of disk imaging, which is covered in Chapter 8 [Disk Imaging], page 33.

2.5.1 Example: Growing a partition into unused space

Suppose your disk layout looks like this:

```
(parted) print
Disk geometry for /dev/hda: 0.000-1000.000 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.063   500.000  primary   ext2
2        500.000   625.000  primary   linux-swap
```

There is 375 Mb of free space at the end of the disk (after partition 2). Partition 1 has an ext2 file system, which is the root device. Partition 2 is a swap device.

Suppose you wanted to use the free space at the end of the disk for the file system on partition 1. You could do the following:

1. These steps will modify both the root file system on partition 1, and the swap device on partition 2. Therefore, you shouldn't be using either partitions. You should probably use a Parted boot disk. See Section 1.6 [Parted boot disks], page 3. From the boot disk, run Parted:

   ```
   # parted /dev/hda
   ```

2. Remove partition 2 (the swap partition). Normally, you wouldn't want to delete a partition with data on it. However, a swap partition doesn't contain data when it isn't "swapped on" (mounted), so you can remove it, and create a replacement swap partition later.

   ```
   (parted) rm 2
   ```

3. Create the new swap partition at the end of the disk:

   ```
   (parted) mkpartfs primary linux-swap 875 999.9
   (parted) print
   Disk geometry for /dev/hda: 0.000-1000.000 megabytes
   Disk label type: msdos
   Minor    Start       End     Type      Filesystem  Flags
   1          0.063    500.000  primary   ext2
   2        875.000   1000.000  primary   linux-swap
   ```

4. Grow partition 1, into the adjacent free space:

   ```
   (parted) resize 1 0.063 874.9
   ```

   All done!

   ```
   (parted) print
   Disk geometry for /dev/hda: 0.000-1000.000 megabytes
   Disk label type: msdos
   Minor    Start       End     Type      Filesystem  Flags
   1          0.063    874.999  primary   ext2
   2        875.000   1000.000  primary   linux-swap
   ```

2.5.2 Example: Resizing an ext2 partition on a crowded disk.

Suppose your disk layout looks like this:

```
(parted) print
Disk geometry for /dev/hda: 0-8063.5 megabytes
Disk label type: msdos
Minor   Start     End    Type            Filesystem     Flags
1          0.0     23.5  primary         ext2           boot
2         23.5   8056.0  extended
5         23.6   3545.6  logical         ext2
6       3545.6   7067.7  logical         ext2
7       7067.7   7326.5  logical         ext2
8       7326.5   7585.4  logical         ext2
9       7585.4   7844.2  logical         linux-swap
```

```
$ df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/hda8            251M   31M  207M  13% /
/dev/hda1             23M  2.4M   19M  11% /boot
/dev/hda5            3.4G  577M  2.7G  18% /usr
/dev/hda6            3.4G  289M  2.9G   9% /home
/dev/hda7            251M   12M  226M   5% /var
```

Suppose you wanted to increase the '`/var`' partition ('`/dev/hda7`') to 1GB, using some space from '`/home`' ('`/dev/hda6`').

To resize a partition with Parted, you use the resize command:

```
(parted) resize partition_number new start new end
```

new start must be the same as the old start for ext2 partitions (unfortunately). So this process is going to be rather complicated. It *is* possible, though.[2]

1. Shrink the '`/home`' partition ('`/dev/hda6`') by 500MB:

   ```
   # parted /dev/hda
   (parted) resize 6 3545.6 6200
   ```

2. Make a new partition in its place. This is where '`/var`' will be, eventually. This new partition will be numbered 10.

   ```
   (parted) mkpartfs logical ext2 6200 7067.7
   ```

3. Copy the old '`/var`' partition ('`/dev/hda7`') to the new one ('`/dev/hda10`').

   ```
   (parted) cp 7 10
   ```

4. Delete the old '`/var`'.

   ```
   (parted) rm 7
   ```

   At this point: all logical partitions greater than 7 just changed number. So 8, 9 and 10 become 7, 8 and 9 respectively.

   This renumbering won't take place while any partitions are mounted on that disk (this will happen when you reboot). That's what that warning message is talking about. So you should *never* attempt to mount a file system touched by Parted (resized or created by Parted), before rebooting, if you get this message.

5. Resize the new '`/var`' partition (now numbered 9), adding the space from the old '`/var`' partition:

   ```
   (parted) resize 9 6200 7326.5
   (parted) quit
   Warning: The kernel was unable to re-read the partition table on
   /dev/hda (Device or resource busy).  This means Linux knows nothing
   about any modifications you made.  You should reboot your computer
   before doing anything with /dev/hda.
   ```

[2] If Parted supported moving the start of ext2 partitions (like it does with FAT partitions), then it would be trivial:

```
(parted) resize 6 3545.6 6200
(parted) resize 7 6200 7326.5
```

6. Since the partition numbers have changed, '/etc/fstab' must be updated. This can be done before rebooting, because the root device wasn't touched by Parted. (If you want to use Parted to do something to the root device, you need to use the boot disk).

   If the old '/etc/fstab' looks like this:

   ```
   /dev/hda8     /            ext2    defaults         1 1
   /dev/hda1     /boot        ext2    defaults         1 2
   /dev/hda6     /home        ext2    grpquota,usrquota  0   2
   /dev/cdrom    /mnt/cdrom   iso9660 noauto,owner,ro  0 0
   /dev/hda5     /usr         ext2    defaults         1 2
   /dev/hda7     /var         ext2    grpquota,usrquota  0   2
   /dev/fd0      /mnt/floppy  auto    noauto,owner     0 0
   none          /proc        proc    defaults         0 0
   none          /dev/pts     devpts  gid=5,mode=620   0 0
   /dev/hda9     swap         swap    defaults         0 0
   ```

   A few lines need to be changed:

   '/var' is now /dev/hda9 (because we copied it to a new partition)

   '/dev/hda8' (the root device) has been renumbered to '/dev/hda7'

   '/dev/hda9' (the swap device) has been renumbered to '/dev/hda8'

   The new '/etc/fstab' looks like this:

   ```
   /dev/hda7     /            ext2    defaults         1 1
   /dev/hda1     /boot        ext2    defaults         1 2
   /dev/hda6     /home        ext2    grpquota,usrquota  0   2
   /dev/cdrom    /mnt/cdrom   iso9660 noauto,owner,ro  0 0
   /dev/hda5     /usr         ext2    defaults         1 2
   /dev/hda9     /var         ext2    grpquota,usrquota  0   2
   /dev/fd0      /mnt/floppy  auto    noauto,owner     0 0
   none          /proc        proc    defaults         0 0
   none          /dev/pts     devpts  gid=5,mode=620   0 0
   /dev/hda8     swap         swap    defaults         0 0
   ```

7. Reboot. That's it!

3 BIOSes and Firmware

“BIOS” (Basic Input/Output System) and “firmware” mean the same thing. However, on PC and PC98 based computers, the word BIOS is more common. On Apple Macintosh and Sun computers, the word “firmware” is more common. The BIOS or firmware program is built into a ROM chip inside your computer, that does memory checks, etc. You can not (easily) change this program. Since BIOSes today are generally compatible with BIOSes in use 20 years ago, these programs tend to have an antiquated design. Features have been added in counter-intuitive ways over the years, leading to overly complicated systems that cause a lot of confusion today.

3.1 The PC BIOS

There are a few popular PC BIOSes: AmiBIOS, Award, Phoenix, and others. They all work in a similar way. None of these BIOSes understand or know about partition tables. They affect partitioning indirectly.

The following occurs when your computer boots up:

1. These BIOSes load the first bit of the boot loader program, stored on the MBR (Master Boot Record) on the hard disk.
2. The BIOS executes the first bit of the boot loader.
3. The boot loader program uses the BIOS to load the rest of itself.
4. The boot loader uses the BIOS to load the operating system (or, perhaps another boot loader, in which case you go to step 2 again).
5. The operating system may or may not use the BIOS to do normal file system access (Windows usually does, Linux or BSD do not).

Steps (3) to (5) involve programs communicating with the BIOS, to ask it to talk to the disks. There are two ways of talking to the BIOS to do this: by using CHS (Cylinders Heads and Sectors) or LBA (Linear Block Addressing). Older BIOSes will only support CHS. New BIOSes usually support both LBA and CHS, although CHS support may be phased out in the future. (CHS is generally considered to be a horrible legacy system.)

Steps (3) and (4), performed by the boot loader, will always use the same access method — either always LBA or always CHS. In the case of the Windows boot loader, this is determined by the LBA flag on the Windows boot partition (see Section 2.4.16 [set], page 13 for information about about flags). In the case of Linux, you will probably be using either LILO or GRUB as your boot loader. GRUB uses LBA if it is available, and CHS otherwise. LILO requires that you choose when you install it (with the linear, or lba32 options).

Step (5) - IO done by the operating system - only Windows does its IO through the BIOS. [We still don’t know enough about the problems, but it seems Windows can have its own problems with CHS mode. People have told us about occassions where Windows corrupts its own file systems, etc. It’s really hard for us to know what’s going on. We strongly recommend you use LBA, if you can!]

So, there are 3 possible situations, which are all covered here:

1. You are using CHS mode, and your BIOS only supports CHS mode.

2. You are using CHS mode, and your BIOS supports both CHS and LBA. So, you want to convert from CHS mode to LBA mode As Soon As Possible TM.
3. You are using LBA mode already.

3.1.1 Using Parted with CHS mode BIOS

Linux usually detects the BIOS geometry automatically. However, sometimes it gets it wrong. In this case, you should tell it yourself, by passing a parameter to Linux. For example, if Linux thinks your hard drive '/dev/hda' has geometry 256/64/63, but the BIOS setup program says the geometry is 512/32/63, then you would pass this parameter to Linux:

```
hda=512,32,63
```

Parameters are passed in different ways, depending what boot-loader you're using. You are probably using LILO. In this case, you add this line to '/etc/lilo.conf': (You then need to run /sbin/lilo, and reboot for the change to take effect)

```
append="hda=512,32,63"
```

Parted can usually detect if Linux has detected the incorrect geometry. However, it cannot do this if there are no partitions on the disk. In this case, you should check yourself. It is very important that you do this.

Sometimes, Parted will complain that the partitions aren't aligned to cylinder boundaries. Parted will give you the option to ignore. If you ignore, then Parted will do some tricks with your partition table. Linux will have no problems. DOS and Windows will have no problems if they are using LBA mode. If DOS/Windows is using CHS mode, then reinstalling the the boot loader (see Section 4.3 [MS DOS MS Windows 9x MS Windows ME], page 22) should solve any problems - but switching to LBA mode is preferrable (see see Section 3.1.2 [CHS to LBA mode], page 18).

Partitions that are involved in the boot process must end before cylinder 1024, if CHS mode is being used. See the section on boot loaders to determine if a partition is involved in the boot process. The typical setup is to have a small Linux '/boot' partition, a Windows partition and then the Linux partition(s).

3.1.2 Converting from CHS to LBA mode BIOS

To make the Windows boot loader + operating system use LBA mode, just set the LBA flag on all FAT partitions Section 2.4.16 [set], page 13. You should have no problems. If you do have problems booting Windows, then reinstalling the Windows boot loader will fix it Section 4.3 [MS DOS MS Windows 9x MS Windows ME], page 22.

Linux doesn't use the BIOS for IO. However, the boot loader (LILO or GRUB) may. GRUB automatically uses LBA, if it is available. LILO requires the "linear" or "lba32" option. So, if LILO still boots after changing '/etc/lilo.conf' reinstalling with /sbin/lilo, then everything's done! See Section 4.1 [LILO], page 21. (If you have problems, just revert back to CHS, by removing "linear" or "lba32", and reinstalling lilo from a boot disk.)

Now that you're using LBA, read on...

3.1.3 Using Parted with LBA mode BIOS

LBA solves all of CHS's problems. However, there is no reliable way for Linux or Parted to know that you're using LBA, so Parted may give you warnings about cylinders not being aligned, or the BIOS geometry being inconsistent, etc. You can ignore these messages if you are using LBA mode. (Old versions of Parted had problems, which have all been solved).

When your disk is in LBA mode, Parted (and most other programs) will say your CHS geometry is X/255/63 - unless you have switched from CHS to LBA.

3.2 OpenFirmware of the Apple Macintosh

There are two major version of the PowerMac's OpenFirmware - one used on "old world", and one on "new world" PowerMac. "New world" refers to coloured PowerPCs manufactured since 1999. They have significant differences. However, both understand partition maps.

Both require that the user choose exactly one partition to be the boot partition (the partition with the boot loader). However, they use different mechanisms to do this.

3.2.1 Old World Macintosh OpenFirmware

The partition chosen to boot is set by various boot loaders, like Quik. So, you shouldn't need to do anything. We could add support for this in Parted, if someone screams loudly enough...

3.2.2 New World Macintosh OpenFirmware

New world OpenFirmware requires the boot partition to be HFS and marked as a boot partition. It uses a different mechanism to be marked as the boot partition. This is controlled with Parted's "boot" flag. For example:

```
(parted) set 2 boot on
```

3.3 The PC98 BIOS

The PC98 BIOS allows you to mark any number of partitions as bootable. You can mark or unmark a partition as bootable with Parted's "boot" flag. For example:

```
(parted) set 2 boot off
```

3.4 SGI / MIPS Firmware

The SGI / MIPS firmware allows booting from special boot files, which are managed by the partition table. In Parted, these boot files are treated as logical partitions inside an extended partition.

For example:

```
Disk label type: mips
Minor    Start       End     Type      Filesystem  Name        Flags
9          0.000      2.732  extended
17         0.002      0.002  logical               sgilabel
18         0.003      1.162  logical               symmon
19         1.404      1.670  logical               sash
1          2.732   8555.212  primary   xfs                     root
2       8555.213   8682.270  primary                           swap
```

Partition 9 is the extended partition (volume header in SGI/MIPS terminology) where the boot files may lie. Partitions 17, 18 and 19 are the boot files. There names can be manipulated with parted's name command. Partitions 1 and 2 are normal partitions. They can't have names.

Note that Linux doesn't see the boot files as partitions (maybe it should?). So /dev/hda17 doesn't exist in Linux. You should use dvhtool(8) to manipulate boot files.

4 Boot Loaders

The boot loader is the program that allows you to select which operating system you want to use, and loads that operating system. You may have more than one boot loader installed, especially if you have more than one type of operating system installed. It is common for boot loaders to be able to load other boot loaders.

When resizing a partition, lots of data gets moved around. Many boot loaders don't understand the file system. They just remember where on the disk the required boot loader information lies. If this information is moved, they must be told where it has been moved. This is done by reinstalling the boot loader (i.e., re-running the installer program for the boot loader, which usually involves issuing a single command at the shell). Not all boot loaders require this.

4.1 LILO: a bootloader for the Linux kernel

LILO is a popular boot loader for x86. LILO's boot loader is usually installed with:

```
# /sbin/lilo
```

If you are using a boot disk, then you should do this instead: (where '/dev/hda1' should be replaced with your root device)

```
# mount /dev/hda1 /mnt
# chroot /mnt /sbin/lilo
# umount /dev/hda1
```

Old versions of LILO don't support LBA mode (see Section 3.1 [PC BIOSes], page 17). LBA mode is enabled with the lba32 or linear option, in '/etc/lilo.conf' (see the LILO documentation for more info).

If you use LBA mode, you should have no problems, as long as your BIOS supports LBA.

If you use CHS mode, then the partition with your '/boot' directory must finish before cylinder 1024. So, if you have a large disk (say, over 8 gigabytes), you should have a '/boot' partition near the start of your disk.

4.2 GRUB: The GNU GRand Unified Bootloader

GRUB is a relatively new boot loader, for x86. Depending on how GRUB is installed, it may understand the file system, or simply remember where the boot files are stored. It understands the file system if it's using "Stage1.5". If it's not using Stage1.5, or the partition number changes, then you need to reinstall Stage2 (please see the GRUB documentation). Otherwise, you don't need to do anything.

GRUB automatically detects if LBA is available, and will use it if it is available (equivalent to LILO's "lba32" option).

4.3 Legacy Microsoft Operating System Bootloaders

DOS and Windows require you to re-install the boot loader if you change the FAT type (FAT16 or FAT32) of the boot partition. Parted will warn you before attempting to do this. To re-install the boot loader, you can either create a boot disk, or use the boot CDROM. The boot disk method does not work with Windows ME.

– BOOT DISK METHOD (DOS/Windows 9x)
  1. Create a Windows boot disk
     - Boot Windows. *This implies you should make the boot disk before you use parted.*
     - Right click on the floppy drive in Windows Explorer.
     - Click on “Format”.
     - Mark “Copy system files”.
     - Click on “Format”.
     - Copy C:\WINDOWS\COMMAND\SYS.COM to A:\ Note: you might have called C:\WINDOWS something else, like C:\WIN98.
  2. Boot off the Windows boot disk, by leaving the boot disk in the floppy drive when booting. You may need to tell your BIOS to boot off the floppy.
  3. Type at the DOS prompt:

     ```
     A:\>sys c:
     ```

– CDROM METHOD: (Windows 9x/ME)
  1. Insert the Windows CDROM, and boot from it. (Select “boot without CDROM support”).
  2. Type:

     ```
     A:\>c:
     C:\>cd \windows\command    (might be \win98\command, or similar)
     C:\WINDOWS\COMMAND>sys c:
     ```

     That’s all there is to it.

Also, DOS and Windows impose a few restrictions:

- The boot partition should be selected with the “boot” flag. Only one boot partition can be selected (sometimes called the “active” partition). For example, to set partition 3 to be the boot partition, do:

  ```
  (parted) set 3 boot on
  ```

- The MS DOS and MS Windows 9x/ME can only boot from the first FAT partition. That is, the FAT partition with the smallest minor number, that isn’t hidden. Note that boot loaders like GRUB and LILO (and some BIOSes) can change this behaviour. . .
- If you are using CHS addressing (rather than LBA addressing), then the boot partition start must be less than cylinder 1024. You can tell MS DOS to use (or not to use) LBA addressing, by enabling or disabling the LBA flag on the boot partition. For example, to enable the LBA flag on partition 2, do:

  ```
  (parted) set 2 lba on
  ```

  Note: LBA addressing is not supported in MS-DOS 6.22 and lower, as well as all versions of PC-DOS.

Warning: some BIOSes won't enable LBA addressing, unless you enable it in the BIOS as well. If for some reason, Windows doesn't boot after changing this flag, this is probably the problem.

- the "real" MS-DOS (i.e. up to version 6.22) and MS-DOS 7.0 (i.e. Windows 95/95a) don't know about FAT32. It's therefore possible to boot them from the *second fat* (FAT16 only, of course) partition, when the *first fat* partition is FAT32. Both have to be primary partitions, so you'll have to set the one you want to boot from as active partition.

4.4 The Microsoft Windows NT Bootloader

Windows NT can't read or boot from FAT32 partitions. Therefore, you should never convert FAT16 partitions to FAT32 partitions, if you want to use them with Windows NT.

4.5 The Microsoft Windows 2000 Bootloader

Windows 2000 require you to re-install the boot loader if you change the FAT type (FAT16 or FAT32) of the system partition. Parted will warn you before attempting to do this. To re-install the boot loader, do:

1. Boot off the Windows 2000 CD.
2. It will ask if you want to proceed installing. Hit Enter.
3. It will then ask you if you want to install a new system, or Repair an existing system. Choose the later (by pressing "R").
4. It will ask you if you want to do an automatic repair, or if you want to use the recovery console. Choose to use the recovery console.
5. At the console, type:

   ```
   C:\>fixboot
   ```

   The system should boot successfully now.

The NT/2000 boot loader also needs:

- its own boot sector code in a PRIMARY FAT12, FAT16 or NTFS partition (FAT32 possible with Windows 2000), which is called the "system partition". This partition should be marked with the "boot" flag in Parted.
- the files NTLDR, BOOT.INI and NTDETECT.COM within the system partition. BOOT.INI holds the information about the physical location of the primary partition or logical drive where Windows NT was installed to, called the "boot partition". The boot partition and system partition may be located together in one primary partition.
- optionally, the file NTBOOTDD.SYS within the system partition, which is the renamed disk driver for your SCSI or IDE controller, when this has no own BIOS (or its BIOS can't access large disks).
- with MS Windows NT, the system partition should end before cylinder 1024, and *must* start before cylinder 1024. If it ends after cylinder 1024 and the files necessary to boot are moved past this border MS Windows NT won't start anymore!

- both the boot and system partition may be resized, without the need for any other changes.
- if the boot partition's number changes (i.e. its minor number), then the BOOT.INI has to be updated.

4.6 Quik: a bootloader for Macintosh PowerPC's

Quik is a popular boot loader for "Old World" Macintosh PowerPCs. You need to reinstall Quik if you resize an ext2 partition, with:

```
# /sbin/quik
```

4.7 Yaboot: a boot loader for Macintosh PowerPC's

Yaboot is a popular boot loader for "new world" Macintosh PowerPCs. ("New-world" refers to coloured PowerPCs manufactured since 1999.)

Yaboot needs its own boot strap partition that must be at least 800k. So, if you are installing GNU/Linux from scratch, you would do something like:

```
(parted) mklabel mac
(parted) print
Disk geometry for /dev/sda: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name          Flags
1          0.000      0.031              Apple
(parted) mkpart primary hfs 0.032 1
(parted) print
Disk geometry for /dev/hdb: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name          Flags
1          0.000      0.031              Apple
2          0.031      1.000
(parted) set 2 boot on
(parted) print
Disk geometry for /dev/hdb: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name          Flags
1          0.000      0.031              Apple
2          0.031      1.000                            boot
```

You don't need to reinstall Yaboot after resizing a partition. Yaboot is installed with ybin Chapter 9 [Related Software and Info], page 35.

5 Operating Systems

Parted only runs under GNU/Linux and GNU/Hurd, at the moment. However, it can be used to resize partitions used by, or shared with other operating systems.

When you want to resize a file system, make sure it is not mounted. Parted can't resize mounted partitions (this may change in the future. . .).

If you modify the partition table on a disk with a partition mounted on it, you should reboot immediately. Linux won't know about the changes you made the partition table. (This will be fixed, with kernel 2.4, and when we add support for it.)

If you want to resize your root or boot partition, use a boot disk See Section 1.6 [Parted boot disks], page 3, or use Andreas Dilger's online ext2 resizer, included in the ext2resize package Section 6.2 [Ext2], page 27.

5.1 Disk labels used by GNU/Linux and FreeBSD

Both GNU/Linux and FreeBSD systems are rather flexible about disk labels, supporting many different disk label types.

Because it is rather unusual for a machine to use hard disks with labels normally used on other architectures, standard distributions of the Linux kernel often only support the popular disk labels for the architecture for which the kernel was compiled. For example, a standard Linux kernel compiled for a PC will likely not have support for Mac or Sun disk labels. To access the file systems on disks with unsupported disk labels, the kernel will have to be recompiled.

FreeBSD has a disk label system that is incompatible with MSDOS partition tables, and a partition slice system that is compatible with MSDOS partition tables. Parted only supports the BSD disk label system. It is unlikely to support the partition slice system, because the semantics are rather strange, and don't work like “normal” partition tables do.

5.2 Disk labels supported by Microsoft systems and OS/2

MS Windows and OS/2 only supports the msdos disk label. Therefore, if you create a new disk label, you should use:

```
(parted) mklabel msdos
```

5.3 Macintosh Operating System Disk Label Support

MacOS (and OpenFirmware) only understands the mac disk label. Therefore, if you create a new disk label, you should use:

```
(parted) mklabel mac
```

Note that for Mac partition maps, you should avoid leaving free space around, because free space regions take up entries in the partition map (and Linux doesn't like having more than 15 entries). For example, if you do:

```
(parted) print
Disk geometry for /dev/sda: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name           Flags
1          0.000      0.031              Apple
2          0.031      1.000                             boot
3          1.000   1000.000  ext2        root           root
(parted)  mkpartfs primary ext2 1001 2000
(parted) print
Disk geometry for /dev/sda: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name           Flags
1          0.000      0.031              Apple
2          0.031      1.000                             boot
3          1.000   1000.000  ext2        root           root
4       1001.000   2000.000  ext2
```

There is 1 megabyte of free space between partitions 3 and 4. You can avoid this, by creating partitions 0.1M apart (in which case, Parted automatically “pushes” them together). So, in the above example, you should do this instead:

```
(parted) mkpartfs primary ext2 1000.1 2000
(parted) print
Disk geometry for /dev/sda: 0.000-6149.882 megabytes
Disk label type: mac
Minor    Start       End     Filesystem  Name           Flags
1          0.000      0.031              Apple
2          0.031      1.000                             boot
3          1.000   1000.000  ext2        root           root
4       1000.000   2000.000  ext2
```

6 File Systems supported by Parted

6.1 File Systems Supported by GNU Parted

Parted has support for these operations:

Filesystem	detect	create	resize	copy	check
ext2	*	*	*1	*2	*3
ext3	*		*1	*2	*3
fat16	*	*	*4	*4	*
fat32	*	*	*	*	*
hfs	*				
jfs	*				
linux-swap	*	*	*	*	*
ntfs	*				
reiserfs	*	*5	*1,5	*5	*3,5
ufs	*				
xfs	*				

NOTES: (1) The start of the partition must stay fixed for ext2, ext3 and reiserfs.

(2) The partition you copy to must be bigger (or exactly the same size) as the partition you copy from.

(3) Limited checking is done when the file system is opened. This is the only checking at the moment. All commands (including resize) will gracefully fail, leaving the file system intact, if there are any errors in the file system (and the vast majority of errors in general).

(4) The size of the new partition, after resizing or copying, is restricted by the cluster size for fat (mainly affects FAT16). This is worse than you think, because you don't get to choose your cluster size (it's a bug in Windows, but you want compatibility, right?).

So, in practise, you can always shrink your partition (because Parted can shrink the cluster size), but you may not be able to grow the partition to the size you want. If you don't have any problems with using FAT32, you will always be able to grow the partition to the size you want.

Summary: you can always shrink your partition. If you can't use FAT32 for some reason, you may not be able to grow your partition.

(5) Reiserfs support is enabled if you install libreiserfs, available at `http://reiserfs.osdn.org.ua`. (It is likely to be availabe soon from www.namesys.com... tell us if/when this happens!)

6.2 GNU Parted and the Second Extended Filesystem

Parted doesn't directly support for copying ext2 file systems (yet). There are a few ways of achieving this, however:

Use Parted's mkfs command (or mkfs.ext2), and then run at the shell:

FIXME!!! this doesn't work quite right - but it should be OK for most people. . .

```
# mount -t ext2 /dev/hda2 /mnt/dst
# find /mnt/src -depth | cpio -pm /mnt/dst
```

If the duplicate partition is going to bigger than the original, this can also be done: first, create the new ext2 partition. Then:

```
# dd if=/dev/src-device of=/dev/dst-device bs=1024 count=old_size
# parted /dev/hda resize 2 start end
```

where *old_size* is the size of the original partition in kilobytes. *start* and *end* are the new start and end for the duplicate partition.

6.3 Legacy Microsoft File System Support

Parted can not grow the cluster size of FAT file systems (yet). This places restrictions on resizing and copying partitions. This is often quite bizarre, because Parted can convert file systems between FAT16 and FAT32, which have different restrictions on what the cluster size can be.

For example, lets say have a 100Mb partition with a cluster size 4k. This partition can't be resized to 400Mb, because the cluster size would need to be changed to 16k. However, it can be resized to 600Mb if you use FAT32. The reverse is true for 600Mb FAT32 file systems.

Note: when you copy or resize a file system, Parted will ask you if you want to convert between FAT16 and FAT32 (if it is possible). Therefore, if you just want to convert a partition to FAT32 (without resizing), you can just resize the partition to the same size.

6.3.1 MS DriveSpace partitions

MS DriveSpace is a program that comes with MS Windows 95 that can be used to compress FAT file systems. I believe this works the same way as DoubleSpace, so everything said here applies should apply to DoubleSpace as well.

It is possible to use Parted to resize and copy these partitions, but you have to do a few extra things. . .

6.3.1.1 Growing a DriveSpace partition

To increase the size of a DriveSpace partition, do the following

1. Use Parted's resize command to grow the partition to the desired size.
2. Use MS DriveSpace to shift the free space from the host drive to the compressed drive.

6.3.1.2 Shrinking a DriveSpace partition

To decrease the size of a DriveSpace partition, do the following:

1. Use MS DriveSpace to shift the free space from the compressed drive to the host drive. The amount of free space shifted corresponds to the desired amount by which to shrink the partition.
2. Use Parted's resize command to shrink the partition to the desired size. NOTE: Parted doesn't give good feedback on what the numbers, to know how much to shrink the partition by. This is on the TODO list.

6.3.1.3 Copying a DriveSpace partition

If you want to copy a DriveSpace partition to a partition that is bigger, then you can follow the instructions for growing a DriveSpace partition, except you copy rather than resize the desired partition, and make sure you use the new partition in DriveSpace.

However, if you want to copy a DriveSpace partition to a partition that is smaller, things get a bit more complicated:

1. Use MS DriveSpace to shift free space from the compressed drive to the source host drive. The amount of space shifted should be more than difference between the source partition, and the desired size of the duplicate partition.
2. Use Parted to copy the source partition to the duplicate partition.
3. Use MS DriveSpace to shift the free space from the source host drive back to the compressed drive.
4. Use MS DriveSpace to shift the free space from the duplicate host drive back to the compressed drive.

6.4 Reiserfs Journalling File System

Parted supports reiserfs if libreiserfs is installed. Parted detects this at runtime, and automatically enables support. You can download libreiserfs from:

`http://reiserfs.linux.kiev.ua`

Note that libreiserfs is new software, and hasn't been tested widely yet.

7 LVM and RAID

7.1 Logical Volume Manager and Redundant Arrays of Inexpensive Disks

LVM (Logical Volume Manager) is an alternative system to partitioning. It allows logical volumes (i.e. "virtual partitions") to be spread over many physical volumes (i.e. hard disks and/or partitions). LVM is supported on Linux version 2.4, and later.

RAID (Redundant Array of Inexpensive Disks) is a system for using many disks and/or partitions together, as a "virtual partition". There are a few different modes of utilising software RAID, that are essentially:

using multiple (small) disks for a single file system, to improve performance and to make the entire space available available on a single file system.

using multiple disks to store redundant copies of information, to improve reliability and performance.

Software RAID is supported on Linux version 2.0, and later.

Hardware RAID is supported normally by Parted - so you need not read this section if you are using hardware RAID (as opposed to software RAID).

LVM, software RAID and partitions are often used simultaneously, but they can all be used independently. LVM and software RAID are often composed of partitions, rather than raw hard disks.

GNU Parted does not support LVM and software RAID in full, but it is still useful when used in combination with their respective tools. Parted is useful for these tasks:

creating a software RAID or LVM partition

creating, resizing or copying a file system on a logical volume (or "virtual partition")

7.2 Creating RAID or LVM partitions

To create a RAID or LVM partition, you must:

1. Create a partition with the mkpart command.
2. Set the LVM or RAID flag on the partition.

For example:

```
(parted) mkpart primary ext2 0 4000
(parted) set 1 lvm on
```

Note: the LVM or RAID partition will not be ready for use yet. You still need to run mkraid(8) for RAID, or use the LVM tools to initialise the physical volume, and create logical groups, etc.

7.3 Manipulating a File System on a RAID or LVM volume

Parted can manipulate RAID and LVM logical volumes, even though it does not understand RAID or LVM. It utilises Linux's support for RAID and LVM. Therefore, you can only use these methods if your Linux kernel supports RAID and/or LVM.

To manipulate a file system on a RAID or LVM logical volume (or, a raw partition, for that matter), you can start parted by selecting the logical volume (partition) device. For example:

```
# parted /dev/md0
```

For the rest of this chapter, "virtual device" will refer to the device Parted is editting (in our example cases, '/dev/md0').

7.3.1 Creating a File System on an LVM or RAID Virtual Device

To create a file system on an LVM volume, use the following steps:

1. Create a loop disk label. This is a fake disk label, that tells Parted to treat the virtual device as a single file system. With this fake disk label, there is either zero or one partition.

   ```
   (parted) mklabel loop
   ```

2. Create the file system, by using Parted's mkpartfs command. You should make the start of the file system 0. The partition can end anywhere inside the virtual device. You can find out the size of the virtual device with the print command. For example:

   ```
   (parted) print
   Disk geometry for /dev/md0: 0.000-47.065 megabytes
   Disk label type: loop
   Minor    Start       End     Filesystem  Flags
   (parted) mkpartfs primary ext2 0 47.065
   (parted) print
   Disk geometry for /dev/md0: 0.000-47.065 megabytes
   Disk label type: loop
   Minor    Start       End     Filesystem  Flags
   1          0.000     47.065  ext2
   ```

7.3.2 Resizing a File System on an LVM or RAID Virtual Device

You usually resize the file system at the same times as you resize your virtual device. If you are growing the file system and virtual device, you should grow the device first (with the RAID or LVM tools), and then grow the file system. If you are shrinking the file system and virtual device, you should shrink the file system first, and then the virtual device afterwards.

To resize the file system in Parted, use the resize command. For example:

```
(parted) select /dev/md0
(parted) resize 1 0 20
```

7.3.3 Copying a File System from an LVM or RAID Virtual Device to a Partition

To copy a filesystem from an LVM or RAID virtual device, just use the cp command. For example:

(parted) *select /dev/hda* (parted) *cp /dev/md0 1 3*

7.3.4 Copying a File System to an LVM or RAID Virtual Device

To copy a file system from an LVM or RAID virtual device, use the following recipe:

1. Create the loop disk label on the virtual device. For example:

   ```
   (parted) select /dev/md0
   (parted) mklabel loop
   ```

2. Create a file system on the virtual device, with the mkpartfs command. For example:

   ```
   (parted) mkpartfs primary ext2 0 47.065
   ```

3. Copy the partition with the cp command:

   ```
   (parted) select /dev/hda
   (parted) cp /dev/md0 3 1
   ```

8 Disk Imaging

Disk imaging is a method for avoiding the tedious Windows install process. For example, if you want to install Windows and Office on 1000 machines, it'll probably take you about 5 times 1000 hours. Things aren't so bad with GNU/Linux, because there are programs like Red Hat's kickstart, which allow you to automate the install of other programs, or practically anything you need to do. Therefore, disk imaging is really only used for Windows machines. Obviously, we urge you to not use Windows (or any non-free software) at all, but we realise that most organisations would find it impossible switch from Windows to GNU/Linux (or other free software) without a transition period, where both systems are available.

With disk imaging, you can burn a CD with a disk image of a partition containing Windows and Office, and copy the partition directly onto the hard disks of all the computers, by sticking in a boot disk and the CD, and letting it fly. But the partition on the Windows disk is probably going to be bigger, so the partition will also have to be resized. I've had several people say that they've managed to automate this process with Linux boot floppies and Parted. It is possible to use the CDROM only, by using the floppy as the boot image on the CD. Read the CD writing HOWTO for more information. There are a few weird things you have to do to get this whole thing to work (which will be fixed in the next stable series). Anyway, this is the general process:

1. Install Windows on a machine, with the setup you want. You can have the partition as big as you want, so long as you don't use more than 640 Mb, and you leave enough room for a complete Linux install, and another 1300 Mb for two copies of the CD image.
2. Install Linux on the machine.
3. Make a directory for the CD image (e.g.: '`/root/cdimage`')
4. Create a 640 disk image file (e.g.: '`/root/cdimage/diskimage`') in the CD image directory:

   ```
   # dd if=/dev/zero of=/root/cdimage/diskimage bs=1M count=640
   ```
5. Use Parted to copy the Windows partition to the disk image:

   ```
   # parted /root/cdimage/diskimage mklabel msdos
   # parted /root/cdimage/diskimage mkpart primary fat32 0 639
   # parted /root/cdimage/diskimage cp /dev/hda 1 1
   ```
6. Create the CD image from the CD image directory, and burn it to CD with your favourite CD writing tools.
7. Compile a special version of Parted without native language support and readline support (or download the special RPM from Freshmeat):

   ```
   localhost:~/parted-1.0.0# ./configure --disable-nls --without-readline -
   -disable-shared; make
   ```
8. Create a Linux boot disk (see the Bootdisk HOWTO).
9. Put the cut-down version of Parted on the boot disk (or a supplementary root disk).
10. Write a shell script to do the following:

    ```
    mount /dev/cdrom /mnt/cdrom
    parted --script /dev/hda mklabel msdos
    ```

```
parted --script /dev/hda mkpartfs primary fat 0 some-size
parted --script /dev/hda cp /mnt/cdrom/diskimage 1 1
parted --script /dev/hda set 1 boot on
/sbin/halt
```

some-size is the size you desire the first partition to use.

11. Start installing! Stick the floppy + CD into each computer, and let it roll...

Obviously, I can and will make this process a lot easier. We're considering making a mini-distribution to do this. I wouldn't have time to maintain such a thing — any volunteers?

9 Related Software and Info

If you want to find out more information, feel free to send questions to `parted@gnu.org`. (!) indicates the information/software is probably included in your distribution.

These files in the Parted distribution contain further information:

- ABOUT-NLS - information about using Native Language Support, and the Free Translation Project
- API - the documentation for the libparted API
- AUTHORS - who wrote what
- BUGS - unfixed bugs
- ChangeLog - record of changes made to Parted
- COPYING - the GNU General Public License, the terms under which GNU Parted may be distributed
- COPYING.DOC - the GNU Free Documentation Licence, the term under which Parted's documentation may be distributed.
- FAT – information on how the FAT resizer works (for programmers)
- INSTALL — how to compile and install Parted, and most other free software
- TODO — planned features that are not yet implemented

These documents are not distributed with Parted, but you may find them useful. Most of these will probably be in your distribution. For example, on Red Hat Linux, look on the CD in '`/doc/HOWTO`' and '`/doc/FAQ`'.

- Filesystems HOWTO `http://penguin.cz/~mhi/fs`
- Hard Disk Upgrade mini-HOWTO (!): `http://sunsite.unc.edu/LDP/HOWTO`
- Large Disk HOWTO `http://www.win.tue.nl/~aeb/linux/Large-Disk.html`
- LILO mini-HOWTO (!) `http://sunsite.unc.edu/LDP/HOWTO`
- MILO HOWTO (!) `http://sunsite.unc.edu/LDP/HOWTO`
- Linux+OS mini-HOWTOs (!): Linux+DOS+Win95+OS2, Linux+FreeBSD-mini-HOWTO, Linux+Win95, Linux+FreeBSD, Linux+NT-Loader. You can get these from: `http://sunsite.unc.edu/LDP/HOWTO`
- Partition mini-HOWTO (!): `http://www.linuxdoc.org/HOWTO/mini/Partition/index.html`
- Partition Table HOWTO `http://www.win.tue.nl/~aeb/partitions/partition_tables.html`
- Partition Types list `http://www.win.tue.nl/~aeb/partitions/partition_types.html`
- Software RAID HOWTO `http://linas.org/linux/Software-RAID/Software-RAID.html`

Other related programs are listed here. Some of these also have useful documentation:

- Disk Drake. Available at `www.linux-mandrake.com/diskdrake` This is similar in functionality to Parted. The FAT code in Disk Drake is based on our Parted code. Here's how Disk Drake compares to Parted: (which we both agree to :-) Disk Drake is:

  - easier to use, protects you from making mistakes
  - more complete solution (handles '`/etc/fstab`', lilo, etc.)
  - less support for FAT (can't convert FAT16<->FAT32, can't copy partitions)
  - less support for ext2 (at the moment)
  - less attention to compatibility between (older) DOS/Windows systems
  - no support for non-PC architectures
- dvhtool (for SGI/MIPS) (!)
- dosfsck (!)
- e2fsck, resize2fs e2fsprogs (!) `http://web.mit.edu/tytso/www/linux/e2fsprogs.html`
- ext2resize - uses the same code as Parted, but includes some other goodies as well, such as an online ext2 resizer, that doesn't require unmounting. `http://ext2resize.sourceforge.net`
- fdisk (!)
- FIPS (!) (First Interactive Partition Splitter) `http://www.igd.fhg.de/~aschaefe/fips`

  RMK: standards: GPL'd
- GPart - recovers broken partition tables. `http://www.stud.uni-hannover.de/user/76201/gpart`
- GNU GRUB - GRand Unified Boot-loader `http://www.gnu.org/software/grub/grub.html`
- LILO (!) (LInux LOader) `ftp://tsx-11.mit.edu/pub/linux/packages/lilo`
- LVM `http://linux.msede.com/lvm`
- mkdosfs (!) (sometimes called mkfs.msdos)
- mke2fs (!) (sometimes called mkfs.ext2)
- mkfs (!)
- mkswap (!)
- quik (!)
- reiserfs: NOTE: a reiserfs resizer is included with normal reiserfs distribution. `http://devlinux.com/projects/reiserfs` Also, an independent reimplementation of reiserfs userland (which parted can utilize) `http://reiserfs.linux.kiev.ua`. This may be adopted by the other reiserfs people in future...
- yaboot (!) `http://ppclinux.apple.com/~benh/`
- ybin (!) `http://www.alaska.net/~erbenson/ybin/`

Appendix A Copying This Manual

A.1 GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

   B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

   C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

   D. Preserve all the copyright notices of the Document.

   E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

   F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

   G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

   H. Include an unaltered copy of this License.

   I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgments" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgments", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See `http://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being *list*"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B This manual's history

This manual is based on the file, USER, included in GNU Parted version 1.4.22 source distribution. The GNU Parted source distribution is available at `ftp.gnu.org/gnu/parted`.

Texinfo formatting by Richard M. Kreuter, 2002.

This manual is distributed under the GNU Free Documentation License, version 1.1 or later, at your discretion, with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. See Appendix A [Copying This Manual], page 37, for details.

Index