The Linux Installation HOWTO

by Matt Welsh, `mdw@sunsite.unc.edu`

v3.3, 11 December 1994

This document describes how to obtain and install the Linux software, focusing on the popular Slackware distribution (version 2.0.1). It is the first document which a new Linux user should read to get started.

Contents

1 Introduction

Linux is a freely-distributable implementation of UNIX for 80386 and 80486 machines. It supports a wide range of software, including X Windows, Emacs, TCP/IP networking (including SLIP), the works. This document assumes that you have heard of and know about Linux, and just want to sit down and install it.

1.1 Other sources of information

If you have never heard of Linux before, there are several sources of basic information about the system. One is the Linux Frequently Asked Questions list (FAQ), available from `sunsite.unc.edu:/pub/Linux/docs/FAQ`. This document contains many common questions (and answers!) about Linux—it is a "must read" for new users.

In the directory `/pub/Linux/docs` on `sunsite.unc.edu` you'll find a number of other documents about Linux, including the Linux `INFO-SHEET` and `META-FAQ`, both of which you should read. Also take a look at the USENET newsgroups `comp.os.linux.help` and `comp.os.linux.announce`.

Another source of online Linux documentation is the Linux HOWTO archive, on `sunsite.unc.edu:/pub/Linux/docs/HOWTO`. The file `HOWTO-INDEX` in that directory explains what Linux HOWTOs are available.

The Linux Documentation Project is writing a set of manuals and books about Linux, all of which are freely distributable on the net. The directory `/pub/Linux/docs/LDP` on `sunsite.unc.edu` contains the current set of LDP manuals.

The book *"Linux Installation and Getting Started"* is a complete guide to getting and installing Linux, as well as how to use the system once you've installed it. It contains a complete tutorial to using and running the system, and much more information than is contained here. This HOWTO is simply a condensation of some of the most important information in that book. You can get *"Linux Installation and Getting Started"* from `sunsite.unc.edu` in `/pub/Linux/docs/LDP/install-guide`. The `README` file there describes how you can order a printed copy of the book (about 180 pages).

1.2 New versions of this document

New versions of the Linux Installation HOWTO will be periodically posted to `comp.os.linux.announce`, `comp.os.linux.help`, and `news.answers`. They will also be uploaded to various Linux FTP sites, including `sunsite.unc.edu:/pub/Linux/docs/HOWTO`.

1.3 Feedback

If you have questions or comments about this document, please feel free to mail Matt Welsh, at `mdw@sunsite.unc.edu`. I welcome any suggestions, criticism, or postcards. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

2 Hardware Requirements

What kind of system is needed to run Linux? This is a good question; the actual hardware requirements for the system change periodically. The Linux Hardware-HOWTO gives a (more or less) complete listing of hardware supported by Linux. The Linux INFO-SHEET provides another list.

At the very least, a hardware configuration that looks like the following is required:

Any ISA, EISA or VESA Local Bus 80386, 80486, or Pentium system will do. Currently, the MicroChannel (MCA) architecture (found on IBM PS/2 machines) is not supported. Many PCI bus systems are supported (see the Linux PCI HOWTO for details). Any CPU from the 386SX to the Pentium will work. You do not need a math coprocessor, although it is nice to have one.

You need at least 4 megabytes of memory in your machine. Technically, Linux will run with only 2 megs, but most installations and software require 4. The more memory you have, the happier you'll be. I suggest 8 or 16 megabytes if you're planning to use X-Windows.

Of course, you'll need a hard drive and an AT-standard drive controller. All MFM, RLL, and IDE drives and controllers should work. Many SCSI drives and adaptors are supported as well; the Linux SCSI-HOWTO contains more information on SCSI.

Linux can actually run on a single 5.25" HD floppy, but that's only useful for installation and maintenance.

Free space on your hard drive is needed as well. The amount of space needed depends on how much software you plan to install. Most installations require somewhere in the ballpark of 40 to 80 megs. This includes space for the software, swap space (used as virtual RAM on your machine), and free space for users, and so on.

It's conceivable that you could run a minimal Linux system in 10 megs or less, and it's conceivable that you could use well over 100 megs or more for all of your Linux software. The amount varies greatly depending on the amount of software you install and how much space you require. More about this later.

Linux will co-exist with other operating systems, such as MS-DOS, Microsoft Windows, or OS/2, on your hard drive. (In fact you can even access MS-DOS files and run some MS-DOS programs from Linux.) In other words, when partitioning your drive for Linux, MS-DOS or OS/2 live on their own partitions, and Linux exists on its own. We'll go into more detail later.

You do NOT need to be running MS-DOS, OS/2, or any other operating system to use Linux. Linux is a completely different, stand-alone operating system and does not rely on other OS's for installation and use.

You also need a Hercules, CGA, EGA, VGA, or Super VGA video card and monitor. In general, if your video card and monitor work under MS-DOS then it should work under Linux. However, if you wish to run X Windows, there are other restrictions on the supported video hardware. The Linux XFree86-HOWTO contains more information about running X and its requirements.

In all, the minimal setup for Linux is not much more than is required for most MS-DOS or MS Windows systems sold today. If you have a 386 or 486 with at least 4 megs of RAM, then you'll be happy running Linux. Linux does not require huge amounts of diskspace, memory, or processor speed. I (used to) run Linux on a 386/16 MHz (the slowest machine you can get) with 4 megs of RAM, and was quite happy. The more you want to do, the more memory (and faster processor) you'll need. In my experience a 486 with 16 megabytes of RAM running Linux outdoes several models of workstation.

3 Getting Linux

In this section we'll cover how to obtain the Linux software.

3.1 Linux Distributions

Before you can install Linux, you need to decide on one of the "distributions" of Linux which are available. There is no single, standard release of the Linux software—there are many such releases. Each release has its own documentation and installation instructions.

Linux distributions are available both via anonymous FTP and via mail order on diskette, tape, and CD-ROM. The Linux Distribution HOWTO (see `sunsite.unc.edu` in the file `/pub/Linux/docs/HOWTO/Distribution-HOWTO`) includes a list of many Linux distributions available via FTP and mail order.

The release of Linux covered in this HOWTO is the Slackware distribution, maintained by Patrick J. Volkerding (`volkerdi@mhd1.moorhead.msus.edu`). It is one of the most popular distributions available; it is very up-to-date and includes a good amount of software including X-Windows, TeX, and others. The Slackware distribution consists of a number of "disk sets", each one containing a particular type of software (for example, the `d` disk set contains development tools such as the `gcc` compiler, and so forth). You can elect to install whatever disk sets you like, and can easily install new ones later.

Slackware is also easy to install; it is *very* self-explanatory. (So self-explanatory, in fact, that this HOWTO may not be necessary.)

The version of Slackware described here is 2.0.0, of 25 June 1994. Installation of later versions of Slackware should be very similar to the information given here.

Information on other releases can be found in the Linux Installation and Getting Started manual from the LDP. You can also find other releases of Linux on various FTP sites, including `sunsite.unc.edu:/pub/Linux/distributions` See the Distribution-HOWTO (mentioned above) for details.

The instructions here should be general enough to be applicable to releases other than Slackware. I hate to be biased towards a single release, but I don't have time to keep up with them all! And Slackware appears to have what most Linux users are looking for.

3.2 Slackware Space Requirements

Unfortunately, Slackware does not maintain a complete list of diskspace requirements for each disk set. You need at least 7 megabytes to install just the A series of disks; a very rough estimate of the required diskspace would be 2 or 2.5 megabytes per disk.

The following disk sets are available:

A

The base system. Enough to get up and running and have elvis and comm programs available. Based around the 1.0.9 Linux kernel, and the new filesystem standard (FSSTND).

These disks are known to fit on 1.2M disks, although the rest of Slackware won't. If you have only a 1.2M floppy, you can still install the base system, download other disks you want and install them from your hard drive.

AP

Various applications and add ons, such as the manual pages, `groff`, `ispell` (GNU and international versions), `term`, `joe`, `jove`, `ghostscript`, `sc`, `bc`, and the quota patches.

D

Program development. GCC/G++/Objective C 2.5.8, `make` (GNU and BSD), `byacc` and GNU `bison`, `flex`, the 4.5.26 C libraries, `gdb`, kernel source for 1.0.9, `SVGAlib`, `ncurses`, `clisp`, `f2c`, `p2c`, `m4`, `perl`, `rcs`.

E

GNU Emacs 19.25.

F

A collection of FAQs and other documentation.

I

Info pages for GNU software. Documentation for various programs readable by `info` or Emacs.

N

Networking. TCP/IP, UUCP, `mailx`, `dip`, `deliver`, `elm`, `pine`, `smail`, `cnews`, `nn`, `tin`, `trn`.

OOP

Object Oriented Programming. GNU Smalltalk 1.1.1, and the Smalltalk Interface to X (STIX).

Q

Alpha kernel source and images (currently contains Linux 1.1.18).

TCL

Tcl, Tk, TclX, blt, itcl.

Y

Games. The BSD games collection, and Tetris for terminals.

X

The base XFree86 2.1.1 system, with `libXpm`, `fvwm` 1.20, and `xlock` added.

XAP

X applications: X11 `ghostscript`, `libgr13`, `seyon`, `workman`, `xfilemanager`, `xv` 3.01, GNU `chess` and `xboard`, `xfm` 1.2, `ghostview`, and various X games.

XD

X11 program development. X11 libraries, server linkkit, PEX support.

XV

Xview 3.2 release 5. XView libraries, and the Open Look virtual and non-virtual window managers.

IV

Interviews libraries, include files, and the `doc` and `idraw` apps. These run unreasonably slow on my machine, but they might still be worth looking at.

OI

ParcPlace's Object Builder 2.0 and Object Interface Library 4.0, generously made available for Linux developers according to the terms in the "copying" notice found in these directories. Note that these only work with `libc`-4.4.4, but a new version may be released once `gcc` 2.5.9 is available.

T

The TeX and LaTeX2e text formatting systems.

You must get the "`a`" disk set; the rest are optional. I suggest at least installing the `a`, `ap`, and `d` sets, as well as the `x` set if you plan to run X Windows.

3.3 Getting Slackware via Mail Order

Slackware is available for free from the Internet, as well as via mail order (if you don't have Internet access, or don't want to take the time to download it yourself). The next section describes how to download Slackware from the Internet.

The various mail order distributors for Slackware (and other Linux distributions) are listed in the *Linux Distribution HOWTO*, from `sunsite.unc.edu` in the directory `/pub/Linux/docs/HOWTO`.

3.4 Getting Slackware from the Internet

The Slackware release of Linux may be found on any number of FTP sites worldwide. The Linux META-FAQ lists several of the Linux FTP sites; we suggest that you try to find the software on the FTP site nearest you, to reduce net traffic. However, two of the major Linux FTP sites are `sunsite.unc.edu` and `tsx-11.mit.edu`.

The Slackware release may be found on the following FTP sites:

- `sunsite.unc.edu:/pub/Linux/distributions/slackware`
- `tsx-11.mit.edu:/pub/linux/packages/slackware`
- `ftp.cdrom.com:/pub/linux/slackware`

`ftp.cdrom.com` is Slackware's home site.

3.4.1 Downloading the files

You need to download the following files from the FTP sites listed above. Make sure that you use binary mode when FTPing them!

- The various `README` files, as well as `SLACKWARE_FAQ`. Be sure to read these files before attempting to install the software, to get any updates or changes to this document.
- A bootdisk image. This is a file that you will write to a floppy to create the Slackware boot disk. If you have a 1.44 megabyte boot floppy (3.5"), look in the directory `bootdsks.144`. If you have a 1.2 megabyte boot floppy (5.25"), look in the directory `bootdsks.12`. The `README` files in these directories describes what the files are.

  You need one of the following files. (The file `READ.ME` in these directories contains an up-to-date list of boot disk images.)

  - `bare.gz`. This is a boot floppy that has only IDE hard drive drivers. (No SCSI, CD-ROM, or networking support.) Use this if you only have an IDE hard drive controller and aren't going to be installing over the network or from CD-ROM.
  - `xt.gz`. This is a boot floppy with IDE and XT hard drive support.
  - `cd.gz`. Contains IDE hard drive and non-SCSI CD-ROM drivers. Get this if you're going to be installing from a (non-SCSI) CD-ROM (only relevant if you bought a Slackware CD-ROM of some kind).
  - `cdscsi.gz`. Contains IDE and SCSI hard drive support, and non-SCSI CD-ROM drivers.
  - `scsi.gz`. Contains IDE and SCSI hard drive support, and SCSI CD-ROM drivers.
  - `net.gz`. Contains IDE hard drive and TCP/IP networking support. Get this if you are going to install over the network using NFS.
  - `scsinet.gz`. Contains IDE and SCSI hard drive support, SCSI CD-ROM drivers, and TCP/IP networking support. You might want to try this one first; use `bare.gz` or one of the other boot floppies if this doesn't work for you.

  You need only **one** of the above bootdisk images, depending on the type hardware that you have in your system.

  The issue here is that some hardware drivers conflict with each other in strange ways, and instead of attempting to debug hardware problems on your system it's easier to use a boot floppy image with only certain drivers enabled. Most users should try `scsi.gz` or `bare.gz`.

- A rootdisk image. This is a file that you will write to a floppy to create the Slackware installation disk. As with the bootdisk image, look in `rootdsks.144` or `rootdsks.12` depending on the type of boot floppy drive that you have.

  You need one of the following files:

  - `color144.gz`. The menu-based color installation disk for 1.44 meg drives. Most users should use this rootdisk.
  - `umsds144.gz`. A version of the `color144` disk for installing with the UMSDOS filesystem, which allows you to install Linux onto a directory of an MS-DOS filesystem. This installation method is not discussed in detail here, but it will prevent you from having to repartition your drive. More on this later.
  - `tty144.gz`. The terminal-based installation disk for 1.44 meg drives. You should use `color144.gz`, but a few people have reported problems with it on their system. If `color144.gz` doesn't work for you, try `tty144.gz` instead. It is a bit dated and the installation procedure isn't identical, but it should work if `color144.gz` doesn't.
  - `colrlite.gz`. The menu-based color installation disk for 1.2 meg drives. Some things have been trimmed off of this disk to make it fit on a 1.2 meg floppy, but it should work if you only have a 1.2 meg drive.
  - `umsds12.gz`. A version of the `colrlite` disk for installing with the UMSDOS filesystem. See the description of `umsds144.gz`, above.
  - `tty12.gz`. The terminal-based installation disk for 1.2 meg drives. Use this rootdisk if you have a 1.2 meg boot floppy and `colrlite.gz` doesn't work for you.

  Again, you need only **one** of the above rootdisk images, depending on the type of boot floppy drive that you have.

- `GZIP.EXE`. This is an MS-DOS executable of the `gzip` compression program used to compress the boot and rootdisk files (the `.gz` extension on the filenames indicates this). This can be found in the `install` directory.

- `RAWRITE.EXE`. This is an MS-DOS program that will write the contents of a file (such as the boot and rootdisk images) directly to a floppy, without regard to format. You will use `RAWRITE.EXE` to create the boot and root floppies. This can be found in the `install` directory as well.

  You only need `RAWRITE.EXE` and `GZIP.EXE` if you plan to create the boot and root floppies from an MS-DOS system. If you have access to a UNIX workstation with a floppy drive instead, you can create the floppies from there, using the `dd` command. See the man page for `dd` and ask your local UNIX gurus for assistance.

- The files in the directories `slakware/a1`, `slakware/a2`, `slakware/a3`, and `slakware/a4`. These files make up the "a" disk set of the Slackware distribution. They are required. Later, you will copy these files to MS-DOS floppies for installation (or, you can install from your hard drive). Therefore, when you download these files, keep them in separate directories; don't mix the `a1` files with the `a2` files, and so on.

  Be sure that you get the files without periods in the filenames as well. That is, within FTP, use the command "`mget *`" instead of "`mget *.*`".

- The files in the directories `ap1`, `ap2`, etc., depending on what disk sets you are installing. For example, if you are installing the "`x`" disk series, get the files in the directories `x1` through `x5`. As with the "`a`" disk set, above, be sure to keep the files in separate directories when you download them.

3.4.2 Installation methods

Slackware provides several different means of installing the software. The most popular is installing from an MS-DOS partition on your hard drive; the other is to install from a set of MS-DOS floppies created from the disk sets that you downloaded.

You can also install Slackware from an NFS-mounted filesystem over a TCP/IP network. This is somewhat involved, however, and a discussion of how to do this is out of the range of this document. If you need help, ask your local UNIX gurus how to set up your system to mount an NFS filesystem which you can install Slackware from. (For this, you'll need another system on the network which has Slackware available on an exported filesystem.)

First we'll describe how to create the boot and root floppies, and then we'll describe how to set things up for either a hard drive or floppy installation.

Creating the boot and root floppies You must create floppies from the bootdisk and rootdisk images that you downloaded, no matter what type of installation you will be doing. This is where the MS-DOS programs `GZIP.EXE` and `RAWRITE.EXE` come into play.

First you must uncompress the bootdisk and rootdisk images using `GZIP.EXE` (on an MS-DOS system, of course). For example, if you're using the `bare.gz` bootdisk image, issue the MS-DOS command:

```
C:\> GZIP -D BARE.GZ
```

which will uncompress `bare.gz` and leave you with the file `bare`. You must similarly uncompress the rootdisk image. For example, if you are using the rootdisk `color144.gz`, issue the command:

```
C:\> GZIP -D COLOR144.GZ
```

which will uncompress the file and leave you with `color144`.

Next, you must have two *high-density* MS-DOS formatted floppies. (They must be of the same type; that is, if your boot floppy drive is a 3.5" drive, both floppies must be high-density 3.5" disks.) You will use `RAWRITE.EXE` to write the boot and rootdisk images to the floppies.

For example, if you're using the `bare.gz` bootdisk, use the command:

```
C:\> RAWRITE
```

Answer the prompts for the name of the file to write (such as `BAREBOOT`) and the floppy to write it to (such as `A:`). `RAWRITE` will copy the file, block-by-block, directly to the floppy. Also use `RAWRITE` for the root disk image (such as `COLOR144`). When you're done, you'll have two floppies: one containing the boot disk, the other containing the root disk. Note that these two floppies will no longer be readable by MS-DOS (they are "Linux format" floppies, in some sense).

Be sure that you're using brand-new, error-free floppies. The floppies must have no bad blocks on them.

Note that you do not need to be running MS-DOS in order to install Slackware. However, running MS-DOS makes it easier to create the boot and root floppies, and it makes it easier to install the software (as you can install directly from an MS-DOS partition on your system). If you are not running MS-DOS on your system, you can use someone else's MS-DOS system just to create the floppies, and install from there.

It is not necessary to use `GZIP.EXE` and `RAWRITE.EXE` under MS-DOS to create the boot and root floppies, either. You can use the `gzip` and `dd` commands on a UNIX system to do the same job. (For this, you will need a UNIX workstation with a floppy drive, of course.) For example, on a Sun workstation with the floppy drive on device `/dev/rfd0`, you can use the commands:

```
$ gunzip bare.gz
$ dd if=bare of=/dev/rfd0 obs=18k
```

You must provide the appropriate block size argument (the `obs` argument) on some workstations (e.g., Suns) or this will fail. If you have problems the man page for `dd` will be instructive.

Preparing for installation from hard drive If you're planning on installing the Slackware software directly from the hard drive (which is much faster and more reliable than a floppy installation), you will need an MS-DOS partition on the system that you're installing Slackware to. (That is, you must already be running MS-DOS on the system.)

Note: If you plan to install Slackware from an MS-DOS partition, that partition must NOT be compressed with DoubleSpace, Stacker, or any other MS-DOS drive compression utility. Linux cannot currently read DoubleSpace/Stacker MS-DOS partitions directly. (You can access them via the MS-DOS Emulator, but that is not an option when installing the Linux software.)

To prepare for hard drive installation, simply create a directory on the hard drive to store the Slackware files. For example,

```
C:\> MKDIR SLACKWAR
```

will create the directory `C:\SLACKWAR` to hold the Slackware files. Under this directory, you should create the subdirectories `A1`, `A2`, and so on, for each disk set that you downloaded. All of the files from the `A1` disk should go into the directory `SLACKWAR\A1`, and so forth.

Now you're ready to go on and install the software; skip to the section "Installing the Software".

Preparing for floppy installation If you wish to install Slackware from floppies instead of the hard drive, you'll need to have one blank, MS-DOS formatted floppy for each Slackware disk that you downloaded. These disks must be high-density format.

The `A` disk set (disks `A1` through `A4`) may be either 3.5" or 5.25" floppies. However, the rest of the disk sets must be 3.5" disks. Therefore, if you only have a 5.25" floppy drive, you'll need to borrow a 3.5" drive from someone in order to install disk sets other than `A`. (Or, you can install from the hard drive, as explained in the previous section.)

To make the disks, simply copy the files from each Slackware directory onto an MS-DOS formatted floppy, using the MS-DOS `COPY` command. As so:

```
C:\> COPY A1\*.* A:
```

will copy the contents of the `A1` disk to the floppy in drive `A:`. You should repeat this for each disk that you downloaded.

You do *not* need to modify or uncompress the files on the disks in any way; you merely need to copy them to MS-DOS floppies. The Slackware installation procedure takes care of uncompressing the files for you.

4 Installing the Software

In this section we'll describe how to prepare your system for installing Slackware, and finally how to go about installing it.

4.1 Repartitioning

On most systems, the hard drive is already dedicated to partitions for MS-DOS, OS/2, and so on. You need to resize these partitions in order to make space for Linux.

NOTE: If you use one of the `umsds` root diskettes, you can install Slackware TO a directory on your MS-DOS partition. (This is different than installing FROM an MS-DOS partition.) Instead, you use the "UMSDOS filesystem", which allows you to treat a directory of your MS-DOS partition as a Linux filesystem. In this way, you don't have to repartition your drive.

I only suggest using this method if your drive has four partitions and repartitioning would be more trouble than it's worth. Or, if you want to try Slackware out before repartitioning, this is a good way to do so. But in most cases you should repartition, as described here. If you do plan to use UMSDOS, you are on your own—it is not documented in detail here. From now on, we assume that you are NOT using UMSDOS, and that you will be repartitioning.

A *partition* is just a section of the hard drive set aside for a particular operating system to use. If you only have MS-DOS installed, your hard drive probably has just one partition, entirely for MS-DOS. To use Linux, however, you'll need to repartition the drive, so that you have one partition for MS-DOS, and one (or more) for Linux.

Partitions come in three flavors: *primary*, *extended*, and *logical*. Briefly, primary partitions are one of the four main partitions on your drive. However, if you wish to have more than four partitions per drive, you need to create an extended partition, which can contain many logical partitions. You don't store data directly on an extended partition—it is used only as a container for logical partitions. Data is stored only on either primary or logical partitions.

To put this another way, most people use only primary partitions. However, if you need more than four partitions on a drive, you create an extended partition. Logical partitions are then created on top of the extended partition, and there you have it—more than four partitions per drive.

Note that you can easily install Linux on the second drive on your system (known as `D:` to MS-DOS). You simply specify the appropriate device name when creating Linux partitions. This is described in detail below.

Back to repartitioning your drive: The problem with resizing partitions is that there is no way to do it (easily) without deleting the data on those partitions. Therefore, you will need to make a full backup of your system before repartitioning. In order to resize a partition, we simply delete the partition(s), and re-create them with smaller sizes.

NOTE: There is a non-destructive disk repartitioner available for MS-DOS, called FIPS. Look on `sunsite.unc.edu` in the directory `/pub/Linux/system/Install`. With FIPS, a disk optimizer (such as Norton Speed Disk), and a little bit of luck, you should be able to resize MS-DOS partitions without destroying the data on them. It's still suggested that you make a full backup before attempting this.

If you're not using FIPS, however, the classic way to modify partitions is with the program `FDISK`. For example, let's say that you have an 80 meg hard drive, dedicated to MS-DOS. You'd like to split it in half—40 megs for MS-DOS and 40 megs for Linux. In order to do this, you run `FDISK` under MS-DOS, delete the 80 meg MS-DOS partition, and re-create a 40 meg MS-DOS partition in its place. You can then format the new partition and reinstall your MS-DOS software from backups. 40 megabytes of the drive is left empty. Later, you create Linux partitions on the unused portion of the drive.

In short, you should do the following to resize MS-DOS partitions with FDISK:

1. Make a full backup of your system.
2. Create an MS-DOS bootable floppy, using a command such as

   ```
   FORMAT /S A:
   ```

3. Copy the files `FDISK.EXE` and `FORMAT.COM` to this floppy, as well as any other utilities that you need. (For example, utilities to recover your system from backup.)
4. Boot the MS-DOS system floppy.
5. Run `FDISK`, possibly specifying the drive to modify (such as `C:` or `D:`).
6. Use the `FDISK` menu options to delete the partitions which you wish to resize. **This will destroy all data on the affected partitions.**
7. Use the `FDISK` menu options to re-create those partitions, with smaller sizes.
8. Exit `FDISK` and re-format the new partitions with the `FORMAT` command.
9. Restore the original files from backup.

Note that MS-DOS `FDISK` will give you an option to create a "logical DOS drive". A logical DOS drive is just a logical partition on your hard drive. You can install Linux on a logical partition, but you don't want to create that logical partition with MS-DOS fdisk. So, if you're currently using a logical DOS drive, and want to install Linux in its place, you should delete the logical drive with MS-DOS `FDISK`, and (later) create a logical partition for Linux in its place.

The mechanism used to repartition for OS/2 and other operating systems is similar. See the documentation for those operating systems for details.

4.2 Creating partitions for Linux

After repartitioning your drive, you need to create partitions for Linux. Before describing how to do that, we'll talk about partitions and filesystems under Linux.

4.2.1 Filesystems and swap space

Linux requires at least one partition, for the *root filesystem*, which will hold the Linux software itself.

You can think of a *filesystem* as a partition formatted for Linux. Filesystems are used to hold files. Every system must have a root filesystem, at least. However, many users prefer to use multiple filesystems—one for each major part of the directory tree. For example, you may wish to create a separate filesystem to hold all files under the `/usr` directory. (Note that on UNIX systems, forward slashes are used to delimit directories, not backslashes as with MS-DOS.) In this case you have both a root filesystem, and a `/usr` filesystem.

Each filesystem requires its own partition. Therefore, if you're using both root and `/usr` filesystems, you'll need to create two Linux partitions.

In addition, most users create a *swap partition*, which is used for virtual RAM. If you have, say, 4 megabytes of memory on your machine, and a 10-megabyte swap partition, as far as Linux is concerned you have 14 megabytes of virtual memory.

When using swap space, Linux moves unused pages of memory out to disk, allowing you to run more applications at once on your system. However, because swapping is often slow, it's no replacement for real physical RAM. But applications that require a great deal of memory (such as the X Window System) often rely on swap space if you don't have enough physical RAM.

Nearly all Linux users employ a swap partition. If you have 4 megabytes of RAM or less, a swap partition is required to install the software. It is strongly recommended that you have a swap partition anyway, unless you have a great amount of physical RAM.

The size of your swap partition depends on how much virtual memory you need. It's often suggested that you have at least 16 megabytes of virtual memory total. Therefore, if you have 8 megs of physical RAM, you might want to create an 8-megabyte swap partition. Note that swap partitions can be no larger than 128 megabytes in size. Therefore, if you need more than 128 megs of swap, you must create multiple swap partitions. You may have up to 16 swap partitions in all.

4.2.2 Booting the installation disk

Specifying hardware parameters The first step is to boot the Slackware bootdisk. After the system boots, you will see the message:

```
Welcome to the Slackware Linux 2.0.0 Bootkernel disk!
```

Here, you are given the opportunity to specify various hardware parameters, such as your SCSI controller IRQ and address, or drive geometry, before booting the Linux kernel. This is necessary in case Linux does not detect your SCSI controller or hard drive geometry, for example.

In particular, many BIOS-less SCSI controllers require you to specify the port address and IRQ at boot time. Likewise, IBM PS/1, ThinkPad, and ValuePoint machines do not store drive geometry in the CMOS, and you must specify it at boot time.

To try booting the kernel without any special parameters, just press `enter` at the boot prompt.

Watch the messages as the system boots. If you have a SCSI controller, you should see a listing of the SCSI hosts detected. If you see the message

```
SCSI: 0 hosts
```

Then your SCSI controller was not detected, and you will have to use the following procedure.

Also, the system will display information on the drive partitions and devices detected. If any of this information is incorrect or missing, you will have to force hardware detection.

On the other hand, if all goes well and you hardware seems to be detected, you can skip to the following section, "Loading the root disk."

To force hardware detection, you must enter the appropriate parameters at the boot prompt, using the following syntax:

```
ramdisk <parameters...>
```

There are a number of such parameters available; here are some of the most common.

- `hd=cylinders,heads,sectors` Specify the drive geometry. Required for systems such as the IBM PS/1, ValuePoint, and ThinkPad. For example, if your drive has 683 cylinders, 16 heads, and 32 sectors per track, enter

  ```
  ramdisk hd=683,16,32
  ```

- `tmc8xx=memaddr,irq` Specify address and IRQ for BIOS-less Future Domain TMC-8xx SCSI controller. For example,

  ```
  ramdisk tmc8xx=0xca000,5
  ```

  Note that the `0x` prefix must be used for all values given in hex. This is true for all of the following options.

- `st0x=memaddr,irq` Specify address and IRQ for BIOS-less Seagate ST02 controller.
- `t128=memaddr,irq` Specify address and IRQ for BIOS-less Trantor T128B controller.
- `ncr5380=port,irq,dma` Specify port, IRQ, and DMA channel for generic NCR5380 controller.
- `aha152x=port,irq,scsi_id,1` Specify port, IRQ, and SCSI ID for BIOS-less AIC-6260 controllers. This includes Adaptec 1510, 152x, and Soundblaster-SCSI controllers.

For each of these, you must enter '`ramdisk`' followed by the parameter that you wish to use.

If you have questions about these boot-time options, please read the Linux *SCSI HOWTO*, which should be available on any Linux FTP archive site (or from wherever you obtained this document). The *SCSI HOWTO* explains Linux SCSI compatibility in much more detail.

Loading the rootdisk After booting the kernel, you will be prompted to enter the Slackware root disk:

```
Please remove the boot kernel disk from your floppy drive,
insert a disk to be loaded into the ramdisk, and press
[enter] to continue.
```

At this point you should remove the bootdisk from the drive and insert the rootdisk. Then press `enter` to go on.

The rootdisk will be loaded into memory, and you should be presented with a login prompt. Login as “`root`”.

```
slackware login: root
#
```

4.2.3 Using `fdisk`

To create Linux partitions, we'll use the Linux version of the `fdisk` program. After logging in as root, run the command

`fdisk` ⟨*drive*⟩

where ⟨*drive*⟩ is the name of the drive that you wish to create Linux partitions on. Hard drive device names are:

- `/dev/hda` First IDE drive
- `/dev/hdb` Second IDE drive
- `/dev/sda` First SCSI drive
- `/dev/sdb` Second SCSI drive

For example, to create Linux partitions on the first SCSI drive in your system, use the command

```
fdisk /dev/sda
```

If you use `fdisk` without an argument, it will assume `/dev/hda`.

To create Linux partitions on the second drive on your system, simply specify either `/dev/hdb` (for IDE drives) or `/dev/sdb` (for SCSI drives) when running `fdisk`.

Your Linux partitions don't all have to be on the same drive. You might want to create your root filesystem partition on `/dev/hda` and your swap partition on `/dev/hdb`, for example. In order to do so just run `fdisk` once for each drive.

Use of `fdisk` is simple. The command “`p`” displays your current partition table. “`n`” creates a new partition, and “`d`” deletes a partition.

To Linux, partitions are given a name based on the drive which they belong to. For example, the first partition on the drive `/dev/hda` is `/dev/hda1`, the second is `/dev/hda2`, and so on. If you have any logical partitions, they are numbered starting with `/dev/hda5`, `/dev/hda6` and so on up.

NOTE: You should not create or delete partitions for operating systems other than Linux with Linux `fdisk`. That is, don't create or delete MS-DOS partitions with this version of `fdisk`; use MS-DOS's version of `FDISK` instead. If you try to create MS-DOS partitions with Linux `fdisk`, chances are MS-DOS will not recognize the partition and not boot correctly.

Here's an example of using `fdisk`. Here, we have a single MS-DOS partition using 61693 blocks on the drive, and the rest of the drive is free for Linux. (Under Linux, one block is 1024 bytes. Therefore, 61693 blocks is about 61 megabytes.) We will create two Linux partitions: one for swap, and one for the root filesystem.

First, we use the "`p`" command to display the current partition table. As you can see, `/dev/hda1` (the first partition on `/dev/hda`) is a DOS partition of 61693 blocks.

```
Command (m for help):   p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

     Device Boot  Begin   Start     End  Blocks   Id  System
  /dev/hda1   *       1       1     203   61693    6  DOS 16-bit >=32M

Command (m for help):
```

Next, we use the "`n`" command to create a new partition. The Linux root partition will be 80 megs in size.

```
Command (m for help):  n
Command action
    e   extended
    p   primary partition (1-4)
p
```

Here we're being asked if we want to create an extended or primary partition. In most cases you want to use primary partitions, unless you need more than four partitions on a drive. See the section "Repartitioning", above, for more information.

```
Partition number (1-4): 2
First cylinder (204-683):  204
Last cylinder or +size or +sizeM or +sizeK (204-683): +80M
```

The first cylinder should be the cylinder AFTER where the last partition left off. In this case, `/dev/hda1` ended on cylinder 203, so we start the new partition at cylinder 204.

As you can see, if we use the notation "`+80M`", it specifies a partition of 80 megs in size. Likewise, the notation "`+80K`" would specify an 80 kilobyte partition, and "`+80`" would specify just an 80 byte partition.

```
Warning: Linux cannot currently use 33090 sectors of this partition
```

If you see this warning, you can ignore it. It is left over from an old restriction that Linux filesystems could only be 64 megs in size. However, with newer filesystem types, that is no longer the case... partitions can now be up to 4 terabytes in size.

Next, we create our 10 megabyte swap partition, `/dev/hda3`.

```
Command (m for help): n
Command action
    e   extended
    p   primary partition (1-4)
p

Partition number (1-4): 3
First cylinder (474-683):  474
Last cylinder or +size or +sizeM or +sizeK (474-683):  +10M
```

Again, we display the contents of the partition table. Be sure to write down the information here, especially the size of each partition in blocks. You need this information later.

```
Command (m for help): p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

     Device Boot  Begin   Start     End  Blocks   Id  System
   /dev/hda1   *      1       1     203   61693    6  DOS 16-bit >=32M
   /dev/hda2        204     204     473   82080   83  Linux native
   /dev/hda3        474     474     507   10336   83  Linux native
```

Note that the Linux swap partition (here, `/dev/hda3`) has type “Linux native”. We need to change the type of the swap partition to “Linux swap” so that the installation program will recognize it as such. In order to do this, use the `fdisk` “`t`” command:

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
```

If you use “`L`” to list the type codes, you’ll find that 82 is the type corresponding to Linux swap.

To quit `fdisk` and save the changes to the partition table, use the “`w`” command. To quit `fdisk` WITHOUT saving changes, use the “`q`” command.

After quitting `fdisk`, the system may tell you to reboot to make sure that the changes took effect. In general there is no reason to reboot after using `fdisk`—the version of `fdisk` on the Slackware distribution is smart enough to update the partitions without rebooting.

4.3 Preparing the swap space

If you have 4 megabytes of RAM (or less) in your machine, you need to create a swap partition (using `fdisk`) and enable it for use before installing the software. Here, we describe how to format and enable your swap partition(s).

If you have more than 4 megs of RAM, you need only create your partition(s)—it's not necessary to format and enable them before installing the software. In this case you can skip down to the section "Installing the Software".

If you get any "`out of memory`" errors during the installation procedure you should create a swap partition and enable it as described here.

To prepare the swap space for use, we use the `mkswap` command. It takes the form:

```
mkswap -c ⟨partition⟩ ⟨size⟩
```

where ⟨*partition*⟩ is the partition name, such as `/dev/hda3`, and ⟨*size*⟩ is the size of the partition in blocks.

For example, if you created the swap partition `/dev/hda3` of size 10336 blocks, use the command

```
mkswap -c /dev/hda3 10336
```

The `-c` option tells mkswap to check for bad blocks on the partition when preparing the swap space. If you see any "`read_intr`" error messages during the mkswap operation, this means that bad blocks were found (and flagged). So you can ignore these errors.

To enable swapping on the new device, use the command

```
swapon ⟨partition⟩
```

For example, for our swap space on `/dev/hda3`, we use

```
swapon /dev/hda3
```

We're now swapping with about 10 megabytes more virtual memory.

You should execute `mkswap` and `swapon` for each swap partition that you created.

4.4 Installing the software

Installing the Slackware release is very simple; it's almost automatic. You use the `setup` command, which guides you through a series of menus which allow you to specify the means of installation, the partitions to use, and so forth. Almost everything is automatic.

Here, we're not going to document many of the specifics of using `setup`, because it changes from time to time. `setup` is very self-explanatory; it contains its own documentation. Just to give you an idea of what it's like, however, we'll describe what most installations are like using `setup`.

Before you begin, be sure that you have a high-density MS-DOS formatted floppy on hand. You will use this floppy to create a Linux boot diskette.

After running `fdisk` (and, perhaps, `mkswap` and `swapon` as described above), issue the command

```
# setup
```

This will present you with a colourful menu with various options such as “Addswap” (to set up your swap space), “Source” (to specify the source of the software to install, such as floppy or hard drive), “Target” (to specify where to install the software), and so on.

In general, you should go through the menu commands in the following order:

1. **Addswap.** If you created a swap partition (using `fdisk`), use the `addswap` menu option to tell the system about it. This option will present you with a list of possible swap partitions; just type in the name of the swap partition(s) that you wish to use (such as `/dev/hda3`). The system will then ask you if you want to format the swap partition, which you should do *unless* you already ran `mkswap` and `swapon` on it. That is, you should format the swap partition unless you already formatted and enabled it by hand as described in the previous section.

2. **Source.** This menu option lets you specify the source for the software to install. You can select several means of installation, such as from floppy or from hard drive. If you are installing from floppies, the system will ask you which floppy drive to use. If you are installing from hard drive, the system will ask you what partition the files are stored on, and what directory they are in.

   For example, if you are installing from an MS-DOS partition on your hard drive, and the Slackware files are under the directory `C:\SLACKWAR`, you should enter the name of the MS-DOS partition (such as `/dev/hda1`) and the name of the directory (such as `/slackwar`). Note that you should use forward slashes (`/`), not backslashes (`\`), in the directory name.

   There are other means of installation, such as CD-ROM. These should be self-explanatory as well.

3. **Target.** This menu item lets you specify what partition(s) to install the software on. The system will display a list of possible partitions. First you will be asked to enter the name of the *root partition*, such as `/dev/hda2`. You will be asked if you want to format the partition; unless you are installing on a partition previously formatted for Linux you should do so. You should use the Second Extended Filesystem (`ext2fs`) type for the partition.

   You will also be given a chance to use additional partitions for different parts of the directory tree. For example, if you created a separate partition for the `/usr` filesystem, you should enter the name of that partition and the directory that it corresponds to (`/usr`) when asked.

4. **Disksets.** This option allows you to specify the disksets you wish to install. Use the arrow keys to scroll through the list; pressing the spacebar selects or deselects a set. Press `return` when you're done selecting disk sets.

   You may wish to only install a minimal system at this time. That's fine. Only the `A` diskset is required. After you have installed the software you may run `setup` to install other disksets.

5. **Install.** After setting up all of the parameters above, you're ready to install the software. First the system will ask you what type of prompting to use; you should use the "normal" prompting method (unless you're an expert and have modified the installation tagfiles in some way).

   The system will simply go through each disk set and install the software. For each software package, a dialog box will be displayed describing the software. Software packages that are required will be installed automatically. For optional software packages you will be given the option of either installing or not installing the package. (If you don't wish to install a certain package now, you can always use `setup` on your system to install it later).

   While the software is installing, watch out for error messages that may be displayed. The most common error that you're likely to run into is "`device full`", which means that you have run out of space on your Linux partitions. Unfortunately, the Slackware installation procedure is not quite smart enough to detect this, and will attempt to continue installing the software regardless. If you get any kind of error messages during the installation procedure, you may wish to break out of the installation program (using `Ctrl-C`) to record them. The only solution for the "`device full`" problem is to re-create your Linux partitions with different sizes, or attempt to reinstall the software without several of the optional software packages.

4.5 After installation

After installation is complete, and if all goes well, you will be given the option of creating a "standard boot disk", which you can use to boot your newly-installed Linux system. For this you will need a blank, high-density MS-DOS formatted diskette of the type that you boot with on your system. Simply insert the disk when prompted and a boot diskette will be created.

You will also be given the chance to install LILO on your hard drive. LILO (which stands for LInux LOader) is a program that will allow you to boot Linux (as well as other operating systems, such as MS-DOS) from your hard drive. If you wish to do this, just select the appropriate menu option and follow the prompts.

If you are using OS/2's Boot Manager, the menu will include an option for configuring LILO for use with the Boot Manager, so that you can boot Linux from it.

Note that this automated LILO installation procedure is not foolproof; there are situations in which this can fail. Be sure that you have a way to boot MS-DOS, Linux, and other operating systems from floppy before you attempt to install LILO. If the LILO installation fails you will be able to boot your system from floppy and correct the problem.

More information on configuring LILO is given below.

The postinstallation procedure will also take you through several menu items allowing you to configure your system. This includes specifying your modem and mouse device, as well as your time zone. Just follow the menu options.

4.6 Booting your new system

If everything went as planned, you should be able to boot your Linux boot floppy (not the Slackware installation floppy, but the floppy created after installing the software). Or, if you installed LILO, you should be able to boot from the hard drive. After booting, login as `root`. Congratulations! You have your very own Linux system.

If you are booting using LILO, try holding down `shift` or `control` during boot. This will present you with a boot prompt; press `tab` to see a list of options. In this way you can boot Linux, MS-DOS, or whatever directly from LILO.

After booting your system and logging in as root, one of the first things you should do is create an account for yourself. The `adduser` command may be used for this purpose. For example,

```
# adduser
Login to add (^C to quit): ebersol
Full Name: Norbert Ebersol
GID [100]: 100
UID [501]: 501
Home Directory [/home/ebersol]: /home/ebersol
Shell [/bin/bash]: /bin/bash
Password [ebersol]: new.password

Information for new user [ebersol]:
Home directory: [/home/ebersol]  Shell: [/bin/bash]
Password: [new.password]  UID: [502] GID:[100]
Is this correct? [y/n]: y
```

`adduser` will prompt you for various parameters, such as the username, full name, GID (group ID), UID (user ID), and so on. For the most part you can use the defaults. If you're unfamiliar with creating users on a UNIX system, I strongly suggest getting a book on UNIX systems administration. It will help you greatly in setting up and using your new system.

You can now login as the new user. You can use the keys `Alt-F1` through `Alt-F8` to switch between *virtual consoles*, which will allow you to login multiple times from the console. The `passwd` command can be used to set the passwords on your new accounts; you should set a password for `root` and any new users that you create.

Also, the hostname of your machine is set at boot time in the file `/etc/rc.d/rc.M`. You should edit this file (as `root`) to change the hostname of the machine. You should edit the lines in this file which run the commands `hostname` or `hostname_notcp`. (The default hostname is `darkstar`.) You may also wish to edit the `domainname` commands in this file, if you are on a TCP/IP network. (On most Slackware systems the hostname and domain name are set in the file `/etc/HOSTNAME`, so editing this file will probably suffice.)

Obviously, there are many more things to setup and configure. A good book on UNIX systems administration should help. (I suggest *Essential Systems Administration* from O'Reilly and Associates.) You will pick these things up as time goes by. You should read various other Linux HOWTOs, such as the *NET-2-HOWTO* and *Printing-HOWTO*, for information on other configuration tasks.

After that, the system is all yours... have fun!

5 Configuring LILO

LILO is a boot loader, which can be used to select either Linux, MS-DOS, or some other operating system at boot time. If you install LILO as the *primary* boot loader, it will handle the first-stage booting process for all operating systems on your drive. This works well if MS-DOS is the only other operating system that you have installed. However, you might be running OS/2, which has it's own Boot Manager. In this case, you want OS/2's Boot Manager to be the primary boot loader, and use LILO just to boot Linux (as the *secondary* boot loader).

The Slackware installation procedure allows you to install and configure LILO. However, this method doesn't seem to be smart enough to handle several peculiar situations. It might be easier in some cases to configure LILO by hand.

In order to set up LILO for your system, just edit the file `/etc/lilo.conf`. Below we present an example of a LILO configuration file, where the Linux root partition is on `/dev/hda2`, and MS-DOS is installed on `/dev/hdb1` (on the second hard drive).

```
# Tell LILO to install itself as the primary boot loader on /dev/hda.
boot = /dev/hda
# The boot image to install; you probably shouldn't change this
install = /boot/boot.b
# Do some optimization. Doesn't work on all systems.
compact

# The stanza for booting Linux.
image = /vmlinuz       # The kernel is in /vmlinuz
  label = linux        # Give it the name "linux"
  root = /dev/hda2     # Use /dev/hda2 as the root filesystem
  vga = ask            # Prompt for VGA mode
  append = "aha152x=0x340,11,7,1"  # Add this to the boot options,
                                   # for detecting the SCSI controller

# The stanza for booting MS-DOS
other = /dev/hdb1      # This is the MS-DOS partition
  label = msdos        # Give it the name "msdos"
  table = /dev/hdb     # The partition table for the second drive
```

Once you have edited the `/etc/lilo.conf` file, run `/sbin/lilo` as `root`. This will install LILO on your drive. Note that you must rerun `/sbin/lilo` anytime that you recompile your kernel (something that you don't need to worry about just now, but keep it in mind).

Note how we use the `append` option in `/etc/lilo.conf` to specify boot parameters as we did when booting the Slackware bootdisk.

You can now reboot your system from the hard drive. By default LILO will boot the operating system listed first in the configuration file, which in this case is Linux. In order to bring up a boot menu, in order to select another operating system, hold down `shift` or `ctrl` while the system boots; you should see a prompt such as

```
Boot:
```

Here, enter either the name of the operating system to boot (given by the `label` line in the configuration file; in this case, either `linux` or `msdos`), or press `tab` to get a list.

Now let's say that you want to use LILO as the secondary boot loader; if you want to boot Linux from OS/2's Boot Manager, for example. In order to boot a Linux partition from OS/2 Boot Manager, unfortunately, you must create the partition using OS/2's `FDISK` (not Linux's), and format the partition as FAT or HPFS, so that OS/2 knows about it. (That's IBM for you.)

In order to have LILO boot Linux from OS/2 Boot Manager, you only want to install LILO on your Linux root filesystem (in the above example, `/dev/hda2`). In this case, your LILO config file should look something like:

```
boot = /dev/hda2
install = /boot/boot.b
compact

image = /vmlinuz
  label = linux
  root = /dev/hda2
  vga = ask
```

Note the change in the `boot` line. After running `/sbin/lilo` you should be able to add the Linux partition to Boot Manager. This mechanism should work for boot loaders used by other operating systems as well.

6 Miscellaneous

I don't like to be biased towards the Slackware release, however, in order to document multiple releases of Linux, this file would be much, much longer. It is simpler and more coherent to cover the specific instructions for a single release of Linux. The book *Linux Installation and Getting Started* currently includes *general* installation instructions which should be applicable to "any" distribution of Linux. However, because the installation procedures are so varied, covering them all would be very confusing both to myself and to the reader.

The basic concepts in this document still hold, no matter what release of Linux you choose. For example, all releases require you to run `fdisk`, and all of them (to my knowledge) include some kind of installation menu similar to the `setup` program. If you choose to use a release of Linux other than Slackware, the READMEs and installation instructions that come with that release should be easy to understand in the context of the material presented here.

If you would like a more complete discussion of Linux installation (instead of the "quick" examples given here), read the book *Linux Installation and Getting Started*, from `sunsite.unc.edu` in `/pub/Linux/docs/LDP`. This book includes a complete discussion of how to obtain and install Linux, as well as a basic UNIX and systems administration tutorial for new users.

Please mail me at `mdw@sunsite.unc.edu` if any part of this document is confusing or incorrect. I depend on feedback from readers in order to maintain this document! I also like to help answer questions about Linux installation, if you have any.

I'd like to thank Patrick Volkerding for his work on the Slackware distribution and assistance in preparing this document.

Best of luck with your new Linux system!

Cheers, mdw