

# Professor Norman Matloff's Beginner's Guide to Installing Linux

Dr. Norman Matloff  
Department of Computer Science  
University of California at Davis  
matloff@cs.ucdavis.edu  
©1999, 2000, 2001

Last Updated November 5, 2001

## Contents

<b>1 Overview</b>	<b>3</b>
1.1 Background Needed . . . . .	3
1.2 Why Linux? . . . . .	3
<b>2 Obtaining Linux</b>	<b>3</b>
<b>3 Installing Linux</b>	<b>4</b>
3.1 Assumptions . . . . .	4
3.2 If You Are Upgrading from a Previous Version of Linux . . . . .	4
3.3 Determine What Kind of Hardware You Have . . . . .	4
3.4 Run the fips Program . . . . .	5
3.5 Perform First Section of the Installation Process . . . . .	6
<b>4 Post-Installation Configuration</b>	<b>7</b>
4.1 Configuring Your Search Path ("Why can't I run my a.out?") . . . . .	7
4.2 Starting and Shutting Down Linux . . . . .	8
4.2.1 Starting Linux . . . . .	8
4.2.2 Starting X11 Windows from Text Mode . . . . .	8
4.2.3 Using X11 Windows . . . . .	9
4.2.4 Ending . . . . .	9
4.3 Configuring a Printer . . . . .	10

4.4	Using Your CD-ROM Drive from Linux . . . . .	10
4.5	Dial-Up Networking from Linux . . . . .	10
4.6	Configuring KDE/GNOME . . . . .	11
<b>5</b>	<b>More on Shells</b>	<b>11</b>
<b>6</b>	<b>Adding Other Linux Software</b>	<b>12</b>
6.1	RPMs . . . . .	12
6.2	My Favorite UNIX/Linux Utilities and Applications . . . . .	12
<b>7</b>	<b>Accessing Your Windows Files (Including on Floppy Disks) from Linux</b>	<b>13</b>
<b>8</b>	<b>Other Sources of Information on Linux</b>	<b>14</b>
8.1	Newsgroups . . . . .	14
8.2	The Web . . . . .	14
<b>A</b>	<b>Making an MS-DOS Boot Disk</b>	<b>15</b>
<b>B</b>	<b>Winmodems</b>	<b>15</b>
<b>C</b>	<b>Who Does NOT Need to Run fips</b>	<b>15</b>
<b>D</b>	<b>How to Change LILO Parameters</b>	<b>15</b>
<b>E</b>	<b>If You Wish to Remove Linux</b>	<b>16</b>
<b>F</b>	<b>How to Use wvdial</b>	<b>16</b>
<b>G</b>	<b>How Linux Names Disk Partitions</b>	<b>17</b>
<b>H</b>	<b>Making a Linux Rescue Disk</b>	<b>17</b>

# 1 Overview

## 1.1 Background Needed

I have tailored the material here to beginners. No special sophistication in computers is needed. Any Microsoft Windows user who, say, understands the difference between the C: and A: drives should be able to understand the instructions here and install Linux in an hour's time. (Do not be intimidated by the length of this document; you probably will not have to use most of it.)

## 1.2 Why Linux?

The UC Davis Dept. of Computer Science is making this help page available because it highly recommends that students install Linux on their PCs (or Macs), for the following reasons:

- Linux is becoming one of the “hottest” software systems. Virtually all of the major companies now embracing it, and Linux is a leading corporate choice for Web servers.
- Many CS courses make specific use of Unix, and thus their work cannot be done on Windows platforms. Since it is a full Unix system, Linux allows students to do their homework in the comfort of their own homes, without tying up their phone lines. If you are new to Unix, click here for my Unix tutorial Web page at <http://heather.cs.ucdavis.edu/~matloff/unix.html>
- In installing and using Linux, students learn many practical things about computers which they do not learn in coursework. This practical experience can also help you in job interviews, both for permanent jobs after graduation and for summer jobs and co-ops during your college years. Even if the job you interview for does not involve Unix, you will definitely impress the interviewer if, for example, you discuss various things you have done to use and customize your Linux system.

## 2 Obtaining Linux

There are many different Linux *distributions*, meaning packagings. In this document we concentrate one of the most common distributions, Red Hat.

Instead of Red Hat, you may wish to consider one of the other distributions. In particular, the Mandrake distribution automatically does the disk partitioning for you.

Linux CD-ROMs are sold in boxed sets available at most large bookstores, including UCD's student bookstore. Or you can get the CD-ROM alone (for about \$5 or so) from Cheapbytes, at <http://www.cheapbytes.com>.

The material here on this Web page is tailored to Red Hat, version 7.1, for the main installation procedures. The material other than the main installation should be useful even for non-Red Hat distributions of Linux.

Linux should be installable on almost any PC. (Linux runs on Macintosh platforms too.)

## 3 Installing Linux

### 3.1 Assumptions

As mentioned earlier, this discussion on the main installation procedures is tailored to Red Hat 7.1. We assume that you have a PC running Windows and that you have a single hard drive. We also assume basic (not advanced) familiarity with using Windows.

Some modifications may be necessary for other systems (e.g. using `/dev/hdb` instead of `/dev/hda` if you are installing Linux on the second of two hard drives). But the instructions here should work for most of you.

Please note: To avoid potential damage, for instance loss of files on your hard drive, it is essential that you follow the instructions here as carefully as possible. In addition, although we have endeavored to make the information here accurate, no guarantee is made of any kind.

### 3.2 If You Are Upgrading from a Previous Version of Linux

(If you are installing Linux from scratch, skip this section.) Suppose you already have Linux installed but are upgrading to a newer version. First of course you will want to make sure you save your old files, just in case sometimes goes wrong. A convenient place to save them is the Windows partition of your hard drive disk! Follow the procedures outlined in appendix Section 7.

Copying your files will take some care, though. If a file is copied to the Windows partition, its name will be truncated to Windows standard length limits, which could make things extremely for you to rebuild later on. I suggest that you use the Unix **tar** command to put everything into one big package (with suitable directories and/or README files to remind yourself where the files had been), and then copy the .tar file (compressed, if you wish) to the Windows partition. After you install the the new version of Linux, you then copy the .tar file back again, and `untar`.

(Note: You may have a directory named `.gnome-desktop` in your home directory. If so, it has a subdirectory called "Home directory", which is linked to your home directory. You must avoid `cp -r` operations here, so as to avoid a situation in which at best you have two copies of your home directory and at worst you get an infinite recursion.)

Note that in addition to any "personal" files you have, you may also have added some downloaded packages, whose files are now in places like `/usr/bin`, `/usr/lib`, `/usr/local/bin`, `/usr/local/lib`, `/usr/local/share` and so on. You may also have modified files in `/etc`, such as `/etc/resolv.conf` and `/etc/wvdial.conf`. (Don't copy the Linux system files, e.g. in `/usr/bin`, though, since you want them to be replaced by their counterparts in the new version of Linux.)

### 3.3 Determine What Kind of Hardware You Have

The Red Hat installation program will be able to sense most of your hardware information, but it is helpful if you can do so on your own too.

To check your hardware from Windows, go to My Computer, then System, then Properties to get the amount of RAM and CPU type. Then from there go to Device Manager, and click on the '+' next to each component, e.g. "Disk drives," "Display adapters" and so on. Also, determine your modem port (COM1, COM2, etc.)

by going to My Computer, then Modems, then Properties. Write down all this information.

This includes your hard drive type, such as IDE; your video card make and model; your monitor make and model; the type of connection used for your mouse, such as serial; your modem port; the make and model of your printer; and whether you have any devices connected with an SCSI interface.

### 3.4 Run the **fips** Program

Next, most of you will need to run the **fips** program, which will change a single-partitioned hard drive into a two-partitioned drive, with one partition being for Windows and the other for Linux. The program is available on the Red Hat CD-ROM.

(To see who does NOT need to run **fips**, go to appendix Section C. Also, if you have trouble with **fips**, you may wish to forsake Red Hat for the Mandrake distribution of Linux, which does the partitioning for you.)

Some Linux books, as well as the **fips** documentation, recommend that you first run the Defragment program in Windows before running **fips**. This theoretically should give **fips** more room to use for your Linux partition later on. However, I suggest that you consider NOT doing the defragmenting, at least at first, because of reports that Windows' Defragment puts some files "out in left field," thereby actually giving you LESS room for Linux. The more conservative approach would be to skip the defragmenting at first, and then see how much room **fips** can get you for Linux. If there is not enough room for Linux, you could quit **fips** and go back to run Defragment first, then come back to **fips**. The sequence to run Defragment is: go to My Computer; double right-click on Properties, then Tools.

Note that the books also recommend that you make backup copies of your important Windows files before running **fips**, just in case something goes wrong. It is up to you whether to act on this recommendation. I personally have never seen a single case in which someone has actually lost a Windows file due to not running **fips** correctly. The **fips** program has been refined quite a bit over the years, and works well. Thus I personally never do a big backup before using **fips**, but the decision is yours.

A hard drive can be "partitioned" into various virtual drives, but most PCs are set up so that the entire hard drive consists of one partition. You will need to subdivide that partition into two partitions, one of which will consist of your original Windows files and the other will be used for Linux. The **fips** program will do this.

First, get the MS-DOS prompt. This is best done in this case by booting up the system with an MS-DOS boot floppy with system files; details on how to make the floppy are in appendix Section A.

Next, switch to the dosutils directory on your CD-ROM and start **fips**, by typing:

```
d:  
cd \dosutils  
fips
```

(If your MS-DOS floppy does not support CD-ROM access, go to Windows and copy d: dosutils fips.exe to your floppy, and then boot from the floppy and runs **fips** from it.)

The **fips** program will then give you a summary of the overall capacity of your hard drive, and how much space you are currently using. It will then ask you how much of this space you wish to keep for Windows,

and how much for Linux. (The amount needed will depend on which version of Linux you install; check your manual for this.)

Using the keyboard arrow keys, go through the list of choices **fips** gives you in terms of space allocated to Windows and Linux. Then hit Enter.

When you exit **fips**, you may get an error message, "Memory allocation error." In my experience, this has been no problem.

Reboot the machine, with NO floppy disk in the drive, so that the new partitions are in effect. (Note: Make sure not to write to the hard drive before this reboot.) If Windows complains about the new partition, just ignore the message, since you will soon be changing the partition to non-Windows type anyway.

### 3.5 Perform First Section of the Installation Process

Most likely your computer is bootable from a CD-ROM; try it and see: Put your Linux CD-ROM in the drive, and reboot again, and see if the Linux installation window comes up. If not, you could change the BIOS settings to make the CD-ROM bootable (and the first device checked during the boot process); see your computer's manual on how to do this. Otherwise you will need to boot the installation process using a floppy diskette; see the Red Hat manual on how to make this.

At this point you will get a boot: prompt, giving you various choices on installation or other booting. I recommend the GUI installation method, for which you simply hit the Enter key at the boot: prompt.

Now the Linux kernel will actually be booted, and the installation program will run.

While running the installation program, in the GUI version you make selections by mouse, while in the text one you can move between major choices by pressing Tab and Alt-Tab, and move through lists using the up and down arrow keys. From this point on, we will usually assume you are using the GUI method.

When presented with a question asking whether you desire a certain option, in the GUI version you highlight the option to select it (note that dark lines on the "West" and "North" borders means "selected," while "South" and "East" means "not selected").

At this point, you will be presented with a series of screens. Here are some comments on the screens which may not be obvious:

- Mouse:

If you have only a 2-button mouse, make sure to ask for 3-button emulation. (If you have a laptop, its pointing device is 2-button.)

- Install Options:

We will assume here that you have chosen Workstation. (The Laptop version is similar to Workstation but also adds support for PCMCIA, i.e. PC cards.)

- Automatic Partitioning:

Choose "Automatic partitioning..."

This means you will skip the step involving Red Hat's Disk Druid. However, it will be useful to at least learn how Linux names partitions. It is strongly recommended that you take two minutes at this point to read the material on disk-partition names in appendix Section G.

- Security Options:  
I recommend accepting the default.
- Accounting Configuration:  
Enter the root password (and, a second account if you wish).
- Selecting Package Groups:  
I recommend choosing both GNOME and KDE.
- X Configuration, video card type:  
It guesses your video card type. Confirm if correct, or select on your own.
- X Configuration, screen type:  
It guesses your monitor type. Confirm if correct, or select on your own.
- Desktop:  
Choose your graphical desktop, either GNOME (the default) or KDE. KDE is a bit more colorful, so I suggest you choose that one.  
I recommend that you choose Text for logon, not Graphic.
- About to Install:  
Continue. It will take a minute to “think,” and then about a half hour to get all the packages installed.

The basic Linux package is now installed! Reboot to try it out. Note that at the beginning of the boot process, you will be asked which OS to boot, Linux or “DOS” (i.e. Windows), which you choose from using the up- and down-arrow keys.

## 4 Post-Installation Configuration

### 4.1 Configuring Your Search Path (“Why can’t I run my a.out?”)

The Red Hat distribution does not include your current directory, ‘.’, in the PATH variable. Thus if for example you compile a program and then type

```
a.out
```

the shell may tell you that a.out is not found. You can get around this by explicitly specifying the current directory,

```
./a.out
```

but this is inconvenient.

To remedy this problem more directly, edit the file

```
/root/.bash_profile
```

In the line which sets PATH, append “:.” (a colon and a dot) at the end of the line, with no intervening spaces. Then log out and log in again.

## 4.2 Starting and Shutting Down Linux

### 4.2.1 Starting Linux

From now on, whenever you boot up your machine, you will be presented with the LILO prompt. Answer “dos” if you wish to run Windows, or answer “linux” to run Linux. (If you answer nothing and wait a few seconds, Linux will be booted by default. If you wish to change the default to “dos”, i.e. Windows, see appendix Section D.)

You will then be asked to log in. At first you will do so as “root”, using the password you set during installation. Later, you can set up other usernames as well if you wish.

Once you log in, the action you take next will depend on whether you stated Text or Graphical logon during the installation process:

- If you specified Text logon, at this point you will be in the Linux console, and can enter text-based commands, e.g. the usual Unix cd, ls and so on.
- If you specified Graphical logon, either GNOME or KDE will now come up, depending on which one chose during installation. You will then be running X11 windows, with GNOME or KDE services and interface added.

### 4.2.2 Starting X11 Windows from Text Mode

I recommended Text logon for the sake of flexibility, but that doesn’t mean I feel you should do your work in console mode. Most of the time you run Linux, you should use X11 windows, so as to enable “doing several things at once.”

You will need to configure a windows manager to use X11. The easiest choice is **twm** (I personally prefer **icewm**). Use a text editor to create the file /root/.xinitrc, consisting of the line

```
twm
```

Start up X11 windows by typing

```
startx
```

If you wish to use KDE, type

```
switchdesk KDE
```

(Or specify GNOME if you wish.) This will move your old `.xinitrc` file (make note of the new name), and set things so that when you run **startx** now, then KDE will come up instead of **twm**. If you want **twm** back as the default, simply restore your original `.xinitrc` file.

If you do use KDE or GNOME, I recommend that you do some quick configuration which will I believe you will find very helpful; see appendix Section 4.6.

### 4.2.3 Using X11 Windows

You can then use your Linux box with X11 windows just as you do the workstations in CSIF. Please read my introduction to X11 windows at <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/XWindows.html> even if you already have experience in X11. For example, if you want an “xterm” style terminal window, where you can submit commands to a shell, do the following:

- **twm**: click and hold down the left mouse button, and choose Xterm
- **KDE**: click on the (sea)shell icon
- **GNOME**: click on the terminal icon in the toolbar

Remember, if you have only a 2-button mouse, Linux does 3-button emulation for you, so that for example you can do X11 cut-and-paste operations. To click the (emulated) “middle” button, just simultaneously push both the left and right buttons.

### 4.2.4 Ending

When you wish to end a Linux session, exit KDE or GNOME using their explicit Quit methods.

To exit **twm** and return to console mode, hit Ctrl-Alt-Backspace. Then type

```
poweroff
```

if you’re going to turn off the machine. If you merely wish to reboot and, say, use Windows, then type Ctrl-Alt-Delete.

### 4.3 Configuring a Printer

If Red Hat does not automatically configure your printer for you, you can run Red Hat's **printtool** program.

In an "emergency," you can print Postscript files without the usual process machinery (e.g. `lpd print daemon`). Say the file name is `x.ps` and you have an HP Laserjet 4 printer. Then type `gs -dNOPAUSE -sDEVICE=ljet4 -sOutputFile = - x.ps > /dev/lp0`

Specifying `'-'` as the output file means stdout.

### 4.4 Using Your CD-ROM Drive from Linux

First, set up a directory at which the CD-ROM can be "mounted" into your file system:

```
mkdir /cdrom
```

(Of course, the directory name can be anything, not just "cdrom".) Do this just once, not each time you insert a CD-ROM. But each time you do insert a CD-ROM, type

```
mount -r /dev/hdc /cdrom
```

assuming `/dev/hdc` is your CD-ROM drive; if not, you can probably specify `/dev/cdrom`.

Note: Be sure to leave the `/cdrom` directory and type `umount /cdrom`

before attempting to remove the CD-ROM. If you have any trouble (you shouldn't, but just "if"), then reboot the machine.

### 4.5 Dial-Up Networking from Linux

Important note: Beware of "Winmodems," which will not work with Linux. See our section on this elsewhere in this document, titled "Winmodems."

First, find out which COM port your modem is on. (See the material above on how to determine your hardware properties in Windows.) Then run the Linux program **modemtool**.

Then edit the file `/etc/resolv.conf` (create it if it is not there), adding the line

```
nameserver x.x.x.x
```

to it, where `x.x.x.x` is the DNS address used by your Internet service provider. (You can have up to three such lines.)

I strongly recommend using **wvdial**. It probably is already included in your Linux system, but if not, obtain it from <http://www.worldvisions.ca/wvdial/>.

You need to create the **wvdial** startup file `/etc/wvdial.conf`. You could do this simply by using the sample in the man page as a model. (Red Hat's **wvdial** configuration tool is `rp3`; you may wish to run this instead.) Or you can use **wvdial**'s own configuration tool; see Appendix F for details.

To configure Netscape for e-mail, you may find it necessary to remove the POP mail server entry. Click on Communicator, then Messenger, then Edit, then Preferences, then Identity. Fill out the information, and click on Mail Servers, and remove the "pop" entry (click on it and then click on Delete).

## 4.6 Configuring KDE/GNOME

This is optional, but I strongly recommend that you take just 1 minute to configure KDE and GNOME in a way that will greatly add to your productivity. You will configure so that windows are "auto raise," meaning that as soon as you move the mouse to a window, the window pops up, and configure to have focus-follows-mouse, meaning that when you move the mouse to a window, you can start typing there without a mouse click.

Do the following at least for the desktop you chose during installation, preferably for both:

### **KDE case:**

Click on the KDE manager icon at the bottom of the screen. Click the + sign in Window Behavior, then Properties. For focus type, select "Classic focus follows mouse." Then select Auto Raise.

Then click OK, File and Exit. (If the window is too big for your screen, and the OK box is hidden, move the window upward, by holding down the Alt key and using the left mouse button to drag the left or right border of the window upward, until you see the OK box.)

### **GNOME case:**

Click on the toolbox icon at the bottom of the GNOME screen, and then click on the "Focus behavior" subheading within "Sawfish window manager." In the section titled "When does the mouse pointer affect the input focus" choose "enter-exit."

Select "Raise windows when focused."

Click OK, and then leave the configuration tool, by clicking File and Exit. (If the window is too big for your screen, and the OK box is hidden, move the window upward, by using the middle mouse button to drag the left or right border of the window upward, until you see the OK box.)

## 5 More on Shells

The default shell in Linux is **bash**. It is very good, but if you are used to using, say, **tcsh**, you can use the **chsh** command to change your login shell.

I have an introduction to Unix shells at <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/ShellIntro.html> and a more advanced look at **tcsh** at <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/CShellIII.html>.

## 6 Adding Other Linux Software

### 6.1 RPMs

You can add more programs from your Linux CD-ROM or from the Web (most of them are free). They tend to be in RPM packages, with `.rpm` suffixes in their names. To install such a package, type

```
rpm -i package_file_name
```

If you later wish to remove, i.e. uninstall a package, you can use “rpm -e” (‘e’ stands for “erase”). You do NOT have to have the RPM file present to do this.

Some packages will have different versions for different C libraries. Red Hat uses glibc. Type

```
ls -l /lib/libc*
```

to see which version you have.

You may find that you need some library files for a program you download, and that you are missing those files. You can usually get these from the Web too. If a program complains about a missing file, try the **ldd** command (e.g. “ldd x” if the name of the program which needs the library is x); this will tell you which libraries are needed, where they were found on your system, and which ones, if any, were not found.

### 6.2 My Favorite UNIX/Linux Utilities and Applications

- text editing:

I use the modern extensions to vi: **vim** has a very nice set of features, and neat icons, and **elvis** has a great HTML mode and lots of other nice features. See my tutorials at <http://heather.cs.ucdavis.edu/~matloff/vim.html> and <http://heather.cs.ucdavis.edu/~matloff/elvis.html>.

- e-mail:

I use the **mutt** e-mail utility. It is very flexible and customizable, and I particularly like its ability to record the fact that one has already replied to a message, and the fact that it allows you to save partially-written message for a later time when you can finish writing it. It is text-based, not GUI, but the functionality it gives is what really counts, in my view. See my tutorial at <http://heather.cs.ucdavis.edu/~matloff/mutt.html>.

If you prefer a GUI-based mail utility, many nice ones exist for Linux. Check the Web for these.

- HTML editing:

I typically use **amaya**, which is a full-featured WYSIWYG HTML editor, written by the Web policy consortium. One nice feature is that you can actually use the embedded Web links, good for testing them. See my tutorial at <http://heather.cs.ucdavis.edu/~matloff/amaya.html>.

- integrated software development (IDE):

For C/C++ work, I actually don't use an IDE. I find that the **vim** editor (cited above) and the **ddd** GUI interface to the **gdb** debugging tool, work great together. See my tutorials at <http://heather.cs.ucdavis.edu/~matloff/vim.html> and <http://heather.cs.ucdavis.edu/~matloff/debug.html>.

However, a relatively new package, KDevelop, runs under KDE on Linux systems, and appears to be very nice. You may wish to give it a try.

- word processing:

I use LaTeX, because of its flexibility, its beautiful output, and its outstanding ability to do math. I typically use Lyx, which is a great WYSIWYG interface to LaTeX, especially for math work. See my tutorials at <http://heather.cs.ucdavis.edu/~matloff/latex.html> and <http://heather.cs.ucdavis.edu/~matloff/lyx.html>.

If you wish to use the Microsoft Office environment, there is a free suite of programs, Star Office, which provide Microsoft compatibility. The CD-ROM is packaged with some Linux distributions, or you can order it for about \$10 at the Sun site.

Another free Microsoft-compatible program is AbiWord; it is still under development, but works well for simple documents, and it loads much faster.

## 7 Accessing Your Windows Files (Including on Floppy Disks) from Linux

I suggest that you create a directory, say named /c, at which to “mount” your Windows file system within your Linux file system. The command to do the mount is

```
mount -r -t msdos /dev/hda1 /c
```

for read-only access; change “-r” to “-rw” if you want read/write access. (This assumes that Windows is on /dev/hda1. You saw this when you installed Linux, but if you have forgotten, you can run fdisk to check.) In Windows 98, you may have to specify “vfat” instead of “msdos” in the mount command.

Warning: If you mount your Windows partition with write access, be very careful not to copy to or remove your Windows files!

Note that Windows handles text files differently from Unix; the former ends each line with Carriage Return (CR) and Line Feed (LF) characters, while the latter uses just LF. Some Unix applications have trouble with DOS/Windows format, in which case you need to convert the format. One way to do this is with the vim editor (which is the version of vi which comes with Linux; simply open the file (say, named x) by typing

```
vi x
```

```
type
```

```
:set fileformat=unix
```

(and hit the Enter key), and save/exit by typing ZZ.

I consider it virtually mandatory that to have the mtools set of utilities for reading and writing MS-DOS floppies. For example, mcopy does what the DOS COPY command does, mdir corresponds to MDIR and so on. The mtools package is probably on your Linux CD-ROM, but if not, Click here to get it, at <http://www.tux.org/pub/knaff/mtools/>.

The free **wine** program allows you to run some Windows programs from within Linux, but this may take some detailed work in the case of some Windows applications. You can achieve full Windows programs execution from within Linux by using VMware.

## 8 Other Sources of Information on Linux

### 8.1 Newsgroups

There are various Usenet newsgroups devoted to Linux, a few of which are:

```
comp.os.linux.setup
comp.os.linux.hardware
comp.os.linux.answers
comp.os.linux.announce (excellent for news of new programs, mostly
                        free, that run under Linux)
```

By the way, if you have a problem with hardware and post a query about it to a newsgroup, it is a good idea to include the output from the **dmesg** command. It gives a record of what occurred during bootup.

### 8.2 The Web

- Linux home page, at <http://www.linux.org/> Lots and lots of information is available here.
- [www.linux.com](http://www.linux.com). Chock full of information and links.
- Google's excellent set of links to various Linux sites, [http://directory.google.com/Top/Computers/Software/Operating\\_Systems/Linux](http://directory.google.com/Top/Computers/Software/Operating_Systems/Linux)
- Another good set of Linux links, <http://www.linuxjunior.org/resources.shtml>
- If you are having trouble with specific hardware in your Linux installation, an excellent place to go for detailed information is the Linux HOW-TO documentation. (For the same reason, if you are about to purchase a machine and suspect that some of the hardware is nonstandard, you can check the corresponding Linux HOW-TO to see if there are any problems with that hardware.  
The HOW-TO documents are available at many sites, such as the one at [linux.org](http://www.linux.org).
- Linux applications page (now somewhat old, but still useful), <http://www.xnet.com/~blatura/linapps.shtml>.

## A Making an MS-DOS Boot Disk

Insert a 3.5" floppy disk. Go to My Computer, then Control Panel, then Add/Remove Programs, then Startup Disk.

## B Winmodems

Currently virtually all mass-produced PCs and laptops use *Winmodems*. These are only “partial” modems, in the sense that the hardware is incomplete. The missing hardware is replaced by software – Windows software, which presents a problem for Linux users.

Here are your choices for solving the problem:

- Install another modem. Check the Linux modem Web page, at <http://www.grapevine.net/gromitkc/winmodem.html> before you buy. Or you may wish to avoid the problem entirely by purchasing an external modem which connects to your serial port; most external modems are fine, according to the modem Web page above. (You can probably even leave your old internal modem inside the machine, as long as you unplug the phone wire from it.) Plug your modem into your serial port, and rerun **modemtool**.
- Try to find a Linux driver for your Winmodem. There is a lot of activity in this regard; see [www.linmodems.org](http://www.linmodems.org)
- Or do all your Internet access from Windows, transferring your downloaded/uploaded files to and from Linux via Windows. To do this, see page 7.

## C Who Does NOT Need to Run fips

The main types of people who do NOT need to run **fips** are:

- Those who have already have two or more partitions on their hard drive.
- Those who have two hard drives, and will be installing Linux on the second one. (Make sure you have formatted the second drive using Windows, even though you will be installing Linux there.)
- Those who have a new machine and possess the CD-ROMs with all its software, and are willing to rebuild their Windows system from scratch from those CD-ROMs. In this case, you would re-format C: then run DOS' (not Linux's) fdisk program to make two partitions on C:, then reload all your Windows software onto the first partition.

In these cases, skip the **fips** material in this Web document, and go directly to the main Linux installation section.

## D How to Change LILO Parameters

You can change LILO's parameters by editing LILO's startup file, `/etc/lilo.conf` and then running `/sbin/lilo` without command-line arguments. (The changes to the startup file won't take effect unless you then run

lilo.)

For example, LILO has a parameter named “timeout”, which specifies the number of tenths of seconds LILO will wait for your response after it issues its “LILO boot:” prompt. For instance, if you edit the timeout line in the startup file to read

```
timeout=150
```

then your default OS will automatically be booted if we don’t type anything within 15 seconds of the appearance of the “Lilo boot:” prompt.

The installation process sets Linux as the default OS. If you wish to change this to Windows (called “dos” in LILO), use your editor to change the “default” line in `/etc/lilo.conf`

## **E If You Wish to Remove Linux**

If you wish to remove Linux from your machine, first run Linux’s version of `fdisk` and remove all the Linux partitions. (Be careful not to remove the DOS/Windows partitions!)

Then boot up Windows, and remove **lilo**, by typing:

```
c:  
cd \windows\command  
fdisk /mbr
```

Subsequently Windows will boot up as it did before you installed Linux.

Finally, use Windows’ `fdisk` to recover the former Linux space as Windows partitions.

## **F How to Use wvdial**

The **wvdial** program does a great job of automating the setup of a PPP connection to the Internet. It is already there in your Red Hat distribution.

Create a configuration file by typing

```
wvdialconf /etc/wvdial.conf
```

or by simply modifying the example in the **wvdial** man page.

You are now ready to try it out! Just type

wvdial &

The program should dial out for you, sense when the user and password prompts come from your ISP, automatically send them, and then start **pppd**. You now can run **telnet**, **netscape** or whatever.

To quit your PPP connection, first bring **wvdial** back to the foreground by typing “fg”, and then hit ctrl-c to kill it.

(Do not use the Unix **kill** command for this; ctrl-c gives **wvdial** a chance to end gracefully. Otherwise, the OS kernel’s routing table will not revert to its original setting, and if you use **wvdial** again without rebooting, **wvdial** won’t work properly. If you are curious about this, try running the **route** command before, during and after running **wvdial**.)

## G How Linux Names Disk Partitions

Your entire first hard drive is called /dev/hda, and partitions within it are called /dev/hda1, /dev/hda2 and so on. If your original Windows setup had an extended partition in addition to the primary one, the latter will be /dev/hda1 and the former will be /dev/hda5. The notation for your second hard drive, if you have one, will be /dev/hdb, with partitions within it named /dev/hdb1, /dev/hdb2 etc. Your first SCSI disk, if any, is /dev/sda, etc.

## H Making a Linux Rescue Disk

If you ever have a disk crash or similar problem, you’ll be thankful that you had previously made a Linux rescue disk. If you did not do so during your Linux install process, put a floppy diskette into the drive and run the **mkbootdisk** command.