Professor Norm Matloff's Beginner's Guide to Installing and Using Linux *

Dr. Norm Matloff
Department of Computer Science
University of California at Davis
matloff@cs.ucdavis.edu

December 25, 2011

Contents

*The information contained here is accurate, to the author's knowledge. However, no guarantee is made in this regard.

1 Overview

1.1 Background Needed

I have tailored the material here to beginners. No special sophistication in computers is needed. Any typical Microsoft Windows user should be able to understand the instructions here and install Linux in less than 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 What Is Linux?

Linux is a form of the Unix operating system. Though originally Unix was used mainly by engineers and scientists and thus was not very familiar to the general public, a lot of what you take for granted on computer systems today began in Unix. A notable example is the Internet—the first major operating system to implement the TCP/IP protocol at the heart of the Internet was Unix, and that led to the general acceptance of the protocol.

In the early 1990s, computer science student Linus Torvalds decided to write his own version of Unix, which he called Linux. Other "homegrown" versions of Unix had been written, such as MINIX, but what distinguished Linux was the scale of worldwide participation involved. Torvalds innocently put a message on the Internet asking if anyone wanted to help, and he got a torrent of responses.

There are a several reasons why Linux is mainstream today. First, it became known as a very reliable, stable operating system, with one result being that Linux has become a major platform for large corporate Web servers. Another reason is that it is free, as is the vast majority of the software associated with it developed elsewhere. Many companies have found that it is cheaper to run Linux on their PCs, both for this reason and because of reduced maintenance costs.

There are several good reasons for you to use Linux:

- As mentioned, Linux is becoming one of the "hottest" software systems. Virtually all of the major companies—IBM, HP, Sun Microsystems, etc.—are promoting it, and as mentioned Linux is a leading corporate choice for Web servers. Linux is the main operating system used at Google.
- Linux is also starting to make inroads in large desktop markets, such as businesses, schools and so on, due to its high reliability, far lower rate of infection by viruses compared to Windows, and its low cost.
- The Linux community shares. That means that people online are much more willing to help you (see Section 9.2), and more open source software is available.

If you are a university computer science student, there are some very important additional advantages:

- 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. If you are new to Unix, click here for my Unix tutorial Web page at `http://heather.cs.ucdavis.edu/~matloff/unix.html`, which will introduce you to Unix file and directory commands, and so on.

- 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 internships/co-ops during your college years. Even if the job you interview for does not involve Linux, you will definitely impress the interviewer if, for example, you discuss various things you have done to use and customize your Linux system.

2 Linux Distributions

Linux comes in various *distributions*, called *distros* by Linux afficionados—but they are all Linux in terms of functionality. Some of the most popular are Ubuntu, Red Hat, Fedora, SuSE, MEPIS, PCLinuxOS and so on.

2.1 Which One Is Best?

Remember, there are tons of good distros out there. Any of the above would be fine, as would many others, but here is my short answer: **Use Ubuntu.** It is arguably one of the most user-friendly of the distros, and it has a large user community you can access in the Ubuntu forum on the Web, probably the most active one out there.

I now use Ubuntu myself on my home computers, as well as on my office computer, after years of using various other distros.

If you have an old machine, especially one with limited memory (i.e. RAM), you may wish to give Puppy Linux or Damn Small Linux a try. I installed them (one at a time) on an old 1998 laptop with only 64M of memory! And they take as little as 50M of disk space.

2.2 Obtaining Linux

You can obtain your desired distro (assuming it's one of the free distros, such as Ubuntu) by downloading from the Web and burning a CD (its basic installation form is small enough to fit on a CD).

Or you can buy a book devoted to the distro, or buy a Linux magazine that includes a CD for it.

Important note: If you download Linux from the Web and burn it to a CD or DVD, make sure that you burn the ISO image, as opposed to copying the ISO file as you would in, say, a backup operation. Your burner software should have a choice in its menu for this.

2.3 Live-CD Linux Distributions

A more recent concept in Linux distributions is that of *live CD* distribution. Here the Linux package is on a bootable CD-ROM. The user inserts the CD in the drive, reboots, and then Linux boots up.

The advantage of this approach is that one does not have to get involved in disk partitioning, a sometimes difficult process. One is using Linux without actually installing it, thus without changing the disk partitioning.

A disadvantage is that it may not allow one's application programs to save files to the hard drive, unless one has already split the Windows partition, defeating much of the purpose. However, one can save files to a USB key.

So, the approach is ideal for those who wish to just try Linux for a short period of time, but not so useful for long-term use.

The first well-known live-CD distribution was Knoppix, but there are many others today, including Ubuntu, whose CD you can use either as a live-CD or for permanent installation.

If you use the live-CD approach, you may of course skip Section 3 of this tutorial.

3 Installing Linux

3.1 Assumptions

3.1.1 Generality/Specificity of Coverage

This part of the tutorial will not go into the details for installing one particular distribution. That would be impractical, since the details for any one distribution often change substantially from one release to the next. So instead, this section on installation will discuss the major points you should watch for during the procedure. It will sometimes use Ubuntu as an example for concreteness, but the principles should be similar for most other distributions.

3.1.2 Your Machine

It is assumed that you have an Intel-compatible desktop or notebook, with a bootable CD-ROM or DVD drive. You should have at least 128M of RAM. I recommend that you have at least 10G of disk space available for Linux. It will also be assumed in some places that your machine currently runs Windows.

3.2 Determine Your Hardware Details

The Linux installation program will be able to sense most of your hardware information. So, you can probably skip our section here. But if you want to take about five minutes extra time here, it could be helpful later if you write down some of your hardware types before beginning installation.

You could download the free program **Hardinfo**, and run it to record a list of your hardware. Or to check your hardware from Windows XP, select My Computer | Control Panel | System | Hardware | Device Manager. Click General to get the amount of RAM and CPU type. Then go to Device Manager, and click on the '+' next to each component, e.g. "Disk drives," "Display adapters" and so on. Write down the information, including 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 USB or PS/2; the make and model of your printer; etc.

Do you still have the manual which came with your monitor? If so, check the specs in the back, and write down the horizontal sync and vertical refresh rate, and the make and model.

3.3 Partioning Your Hard Drive

Today most Linux distros, such as Mandriva, SuSE and Ubuntu, do the disk partitioning for you. This is a major advantage, as partitioning is a vital but delicate operation. Later in this section, I'll give you some advice for the Ubuntu case, and also give you some options to use if you have a distro that does not do automatic partitioning.

But I do suggest that even if you will have automatic partitioning done, it would still be worthwhile for you to read Section 3.3.1. This would be useful both for the installation process and later on in your role as an "informed consumer."

3.3.1 What Is Partitioning?

Again, it is probably not necessary for you to know the material here, and it is rather detailed, but you may find it useful at some point. I do recommend that you take a few minutes and read this section.

A hard drive will consist of one or more *partitions*. A partition is a set of contiguous space (sequential blocks) on the disk, and is treated as an independent disk.

So, assuming you want your system to include both Windows and Linux (termed a *dual boot* situation, since you can boot either system), you will need at least one partition for Windows and one (actually two) for Linux.

It's important to understand how the naming works: In Linux systems, all I/O devices are treated as "files." If your first hard drive is of the IDE type, the entire drive is probably called **/dev/hda**, i.e. the "file" **hda** within the directory **/dev**. In the case of SATA-type hard drives, the notation is **/dev/sda** etc.

Your first CD-ROM/DVD drive may be **/dev/hdc** (your third "hard drive"), your first USB port may be **/dev/sdf1** and so on.

Partitions within, say, **/dev/hda**, are called **/dev/hda1**, **/dev/hda2** and so on. Your original Windows single partition was probably **/dev/hda1** or **/dev/sda1**.

Within a partition you'll have some type of file system. The disk consists simply of a long stream of bytes, with no structure, so the OS needs to have a way of organizing them into files, recording where in that stream each file has its bytes. But you don't need to know the details. Windows XP and Vista use the **NTFS** file system. The standard Linux file system is **ext2** (number 0x83, sometimes called *Linux native*), or possibly **ext3**, for your main Linux partition and of type **swap** for your swap partition (number 0x82, used for temporary storage during the time the OS is running).

PCs were originally designed to have up to four "real" partitions, called *primary* partitions. After people found that to be too constraining, *logical* or *extended* partitions were invented. You should install Linux in a primary partition, for recovery reasons, but it is not necessary.

3.3.2 Before You Do the Partitioning

Before you start, give some thought as to how much of the original partition you want to keep for Windows and how much you want to leave for Linux. If you plan to become a serious Linux user,[1] you'll want to

[1] And as mentioned in Section 9.1, if you want to learn Linux, the only way to do it is to become a serious, every day user.

allocate at least half of the space for Linux.

You really ought to run Windows' **chkdsk** command first, in case you have any bad sectors on your hard drive. You may also wish to defragment.

3.3.3 Partitioning Using GParted

Today most distros will invoke a partitioning program to do your partitioning. This could be the famous GParted program, or one that the authors of your distro wrote themselves.

You can use GParted on your own by downloading and booting a GParted live CD before you install Linux, but I'll assume here that your Linux installation program invokes either GParted or another program written specifically for your distro.

Since every distro will handle this a bit differently, what I will do here is just give you an understanding of what operations need to be done, with the specific mouse clicks needed varying from one distro to another.

I'll assume that you want your Windows and Linux systems to coexist on the same hard drive. So when your distro's installer program asks you whether you want to use the entire disk, be sure to say no! Of course, if you do want to erase Windows, or if you are installing Linux on a separate drive from Windows, you can go ahead and use the whole drive.

Here are the main steps in GParted, roughly stated (you may see some variation):

- Select the disk you wish to repartition. If you have only one disk, it will be something like **/dev/hda**. (See Section 3.3.1.)
- Select the partition where Windows resides. This will typically cover the entire disk, and will almost certainly be of file system type NTFS. I'll assume that here.
- Decide how much space you want to remove from the Windows partition in order to make a partition for Linux.
- Now resize, in this case shrink, the Windows partition. The partioner will ask you how much room to make.
- Adjust the partition size according to your desired value.
- You'll need to make the main Linux partition **primary**, of type **ext2** or **ext3**, and set to be bootable.
- You'll need a smaller partition of type **linux-swap**. This is not used for files, but rather as "scratch space" by the OS, for virtual memory and for storage when your machine is in hibernate mode.
- You'll then have to commit, i.e. save, the changes to the partitions. This might take a few minutes, so be patient.
- The next time you boot Windows, you will be asked if you want a disk consistency check. Definitely say yes.

3.4 The Installation Process

3.4.1 To Begin

By the way, if you are upgrading or replacing another version or distribution of Linux, see Section 11 before beginning.

Put your Linux CD-ROM or DVD in the drive, and reboot. The installation program should begin.[2]

3.4.2 Questions You May Be Asked During the Installation Process

The trend in time is for the installation programs to actually ask you fewer and fewer questions, i.e. the process has become more and more automated. Most of the questions discussed in this section will NOT be asked—Ubuntu will probably ask none of them—but the information here will give you an idea of how to answer if they are asked.

- Some distributions will give you a choice of several installation types, which vary in terms of what kinds of application software will be installed. If you are a CS student, you need to make sure your installation will include compilers, editors, debuggers and so on. Note that you can always add more applications later on. But since most people now have plenty of disk space, it is easier to simply ask for everything.

- Assuming you'll want a *dual-boot system*, i.e. you'll be having both Windows and Linux available for booting, you need some sort of *boot loader*. This is a program which upon powerup of your computer will ask you which OS you wish to boot at that time. Your distribution will probably use the GRUB boot loader, or possibly LILO. It doesn't matter that much for a beginner, but if asked, definitely indicate that you want to be able to boot both OSs. (If you are not asked, the distro should make it dual-boot by default.) Take the defaults for everything else, e.g. the choice of bootloader program.

- If you're asked whether you want 3-button mouse emulation, say yes. If you have only a 2-button mouse (the wheel does count as a button), this emulation will enable cut-and-paste window operations.

- You'll need a GUI ("graphical user intrface") desktop manager. The two most widely-used GUI desktop managers for Linux are KDE and GNOME. Each has its band of devoted followers. I generally use GNOME these days, but both are good. It really doesn't matter which one you choose for new users, and you can always switch later if desired. Choose one (or both).

- I mentioned earlier that disk partitioning has over the years been one of the two major issues in Linux installation. The other has been configuring for the video card and monitor.

  With today's modern Linux installation programs, this is typically not a problem. They are pretty good at identifying your video card, and guessing good settings to use. Typically they will give you a chance to test those settings out before continuing with the installation process, with a test image. My experience has generally been that that is sufficient.

  If that image does not turn out well, the installation program will typically give you a chance to state the make and model of your video card, and horizontal sync, vertical refresh rate, and make and

[2] If not, you must 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.

model of your monitor. That is why I asked earlier if you still have the manual for your monitor. (On a laptop, though, you often don't have this information, since its monitor is built in.)

By the way, once a configuration has been decided on, it will be saved to a file, such as **/etc/X11/xorg.conf**. You can look at this later if you are curious as to what configuration the installer has chosen for you, and can modify it if you know what needs to be tweaked.

- You may be asked if your machine has a static Internet address. In most cases, the answer should be no; for a home machine or wireless use you probably get a dynamic Internet address, using a protocol named DHCP.

3.5 Installing Linux to a USB Key or External Hard Drive

You can install Linux to a USB key or external hard drive, and boot up Linux from there. This is especially useful for netbooks without CD drives.

3.5.1 Installation Method I

There's a wonderful program **UNetbootin** which you can use to really automate the process. All you do is specify an ISO file for the Linux distro you want (see below), and it does the rest. See `http://unetbootin.sourceforge.net/`.

You can ask **UNetbootin** to download the ISO file you want from the Web (it gives you a list of many choices), or you can download one yourself and then tell **UNetbootin** where you have it.

3.5.2 Installation Method II (for Slax Linux)

Slax is a nice, colorful and small version of Linux, at `http://www.slax.org`. Click on "Get Slax" to download, and on "Read Manuals" to see how to install onto a USB key or external hard drive. It is extremely easy!

In short:

- Download the Slax **.tar** package.
- Go to the directory (or folder, in Windows) for your USB key.
- Unpack the **.tar** file from that directory.
- Go to the **boot** subdirectory, and run either **bootinst.sh** (from Linux) or **bootinst.bat** (from Windows). In the Linux case, you may need to precede your command by **sudo**.

3.5.3 Installation Method III (from Linux)

Well, this is the long way, but you'll learn more from it. Here are the steps. Say your ISO file is **x.iso**, and your USB key or external drive (I'll jsut say "external device" for either from now on) is at **/dev/sdg1**. (You may need to run as root, say via **sudo**, for some of these steps.)

Create a single, primary ext2 partition on the external device, say using GParted.

```
% mkdir /mykey
% mount /dev/sdb1 /mykey
% mkdir myisodir
% mount -o loop x.iso myisodir
% cp -r myisodir/* mykey
% grub-install --root-directory=/mykey --no-floppy /dev/sdb1
```

The directories **mykey** and **myisodir** can be on any device, as they are just temporary.

The first mount maps whatever is, or will be, in the **mykey** directory to the external device. That way the **cp** does copy the ISO files to that device.

The second mount sets up a file system out of the ISO file, so that the files in the ISO file may be copied.

Note that we also install the GRUB bootloader to the device.

As you final step, you need to configure GRUB, by editing its **menu.lst** file, which should in the above context be in **/mykey/boot/grub/menu.lst**. The following contents will probably be enough, though may be distro-dependent:

```
title yourdistronamegoeshere
kernel /boot/vmlinuz
initrd /boot/initrd.gz
```

Don't forget to unmount **mykey** when you're done, using **umount**.

3.5.4 Usage

Note that you need your machine to try to boot from the USB key before trying to boot from the hard drive. At the very beginning of bootup, hit Esc, F1, F2 or F12 (depends on your machine), to stop the process and allow yourself to be queried as to which device you wish to boot from.

4 Post-Installation Configuration

This section describes some further steps I recommend taking after your installation is finished.

4.1 Help in Hardware Configuration

Having trouble getting some hardware component to work under Linux? I'll have some tips on that below, but keep in mind that a great source is the Web. Plug something like "Linux install XXXX," where XXXX is the type of machine you own) into Google. Actually, it would be better to specify your distro, e.g. "Ubuntu install XXXX." You'll find a number of reports of experiences by other people with your machine/distro.

4.2 Configuring Your Search Path ("Why can't I run my a.out?")

Most Linux distros do 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 are expected to explicitly specify the current directory:

```
./a.out
```

If you consider this a problem, as I do, to remedy it in the case of the BASH shell (the default shell for most distros), edit the file **/.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, or do

```
source ~/.bash_profile
```

4.3 Configuring a Printer

Your Linux distribution should have some program to help you configure your printer if something went wrong during installation. For example, if you are running the GNOME GUI, select System | Administration | Printing.

It's now easy to connect to a remote printer elsewhere on your network (even if it is on a Windows machine), using Samba.

4.4 WiFi Networking

The newer versions of the major distros handle WiFi configuration pretty well without your intervention. But if you have problems, the material in this section may be helpful.

4.4.1 General Information

Below is a five-minute crash course in WiFi. Even if you don't understand all of it, even partial understanding may be helpful.

- Recall that in Unix-family operating systems, I/O devices are represented as "files" in the directory **/dev**. Your WiFi device is probably **eth1** or **wlan0**.
- Your WiFi device needs a driver. Many, if not most, laptops use Broadcom WiFi hardware, and in older Linux distros, they needed some fiddling to work, but now it's much easier (see below).
- The names of wireless access points are called **ESSID**s.
- If you are connected to a router or a wireless access point, your machine is probably assigned an IP address via DHCP, rather than statically. An error message like "no lease offered" means that the DHCP process failed.
- DNS servers convert an "English" address like **www.google.com** to a numerical address like 209.85.171.103. So your OS needs to set up a connection to a DNS server.

4.4.2 Network Management Tools

If you are running the GNOME windows manager, select System | Administration | Network. There is also an icon you can click in the toolbar; it looks like two black monitors when you are not connected, and is a set of blue bars indicating signal strength when you are connected. Note that left- and right-clicking gives different results, so try both. In KDE, select System | Network Device Control.

You can activate/deactivate your netword card during a session. In GNOME, this is done via System | Administration | Network.

The network managers included with most Linux distros are rather primitive. An excellent alternative is WiFi Radar. In Ubuntu, install via

```
sudo apt-get install wifi-radar
```

4.4.3 Individual Linux Network Commands

Useful commands from a terminal window include:

- **iwlist:** You can determine which ESSIDs are within range of you by typing the command

  ```
  $ sudo iwlist eth1 scanning
  ```

  say if your wireless device is **eth1**.

- **ifconfig:** Shows information about all your network interfaces, i.e. their hardware addresses, IP addresses and so on. Lack of IP address on your wireless port, e.g. **wlan0** or **eth1**, may indicate that DHCP has failed. This command can also be used to set the IP address and other parameters "manually," deactivate/reactive a network interface, etc.

- **iwconfig**: Shows information about all your wireless connections.

  Also can be used by you to specify which access point you wish to use.

  For example, to select a particular wireless access point named X, type

  ```
  sudo iwconfig wlan0 essid "X"
  ```

  (assuming **wlan0** is your wireless interface).

- **dmesg:** Shows a record of your last bootup. This may show error messages regarding your WiFi card. It's pretty long, so either run it through **more**, i.e. run

  ```
  dmesg | more
  ```

  or save it to a file, say **dmesg.out**, and then explore the file at your leisure with a text editor.

- **route:** Displays the current packet routing table.

- ethtool: Running

  ```
  ethtool eth0
  ```

will give you information about your Ethernet link, e.g. link speed. To get statistics on recent usage, run

```
ethtool --statistics eth0
```

Some of these must be used with root privilege. For example, running

```
iwlist eth1 scanning
```

may produce no access points, while

```
sudo iwlist eth1 scanning
```

will show you all of them.

The file **/etc/resolv.conf** lists the IP addresses of the DNS servers. You can add more **nameserver** lines if you know of some, say from your ISP (of for that matter, other ISPs).

4.4.4 If You Have a Problem

These days, Linux generally does well with WiFi, and it might work for you "right out of the box," with no configuration on your part. If not, this section is for you.

Some wireless network cards typically sold with PCs today do not have direct Linux drivers available. A common example is the Broadcom BCM43XX series. However, you can still operate as usual after some preparation, as explained below.

Ubuntu: BCM43XX Series

Ubuntu handles Broadcom cards well, as long as you have Linux kernel 2.6.15 or newer. (Run **dmesg** if you want to check this.) You simply need to take the following action once:

First establish an Ethernet connection to the Internet, to enable download. For example, if you have a router at home, even a wireless one, connect your machine directly to the router with an Ethernet cable.

Then Select System | Administration | Hardware Devices (the last might be labeled Additional Drivers). It will ask you if you want to download the Broadcom firmware, so say yes. Check the Enable box for Firmware for Broadcom 43 Wireless Driver. You will be asked whether you want the firmware to be downloaded from the net; say yes. Then check Enabled after the download.

Know Your WiFi Card

You first need to determine which wireless card you have. On the laptop I use now, I determined this by running **dmesg** and **lspci** under Linux. Sure enough, it turned out to be a Broadcom BCM43XX series card.

Other Cards/Kernels

For other cards, go to the **ndiswrapper** home page, `http://ndiswrapper.sourceforge.net/`. The program **ndiswrapper** allows Linux to use Windows drivers.

4.5 Configuring KDE/GNOME for Convenient Window Operations

4.5.1 Autoraise Etc.

You should find that windowing operations are generally easier in Linux systems than in Windows, in the sense of requiring fewer mouse clicks, if you set things up that way. Personally, I find it annoying in Windows that, when I switch from one window to another, I need to click on that second window. In most Linux windowing systems, I can arrange things so that all I have to do is simply move the mouse to the second window, without clicking on it. The term for this is *focus follows mouse*, and we can configure most Linux windowing systems to do this.

Also when I move from one window to another, I want the second one to "come out of hiding" and be fully exposed on the screen. This is called *autoraise*, and can be configured too.

You can arrange this configuration in less than one minute's time. Again, the exact configuration steps will vary from GNOME to KDE, and from one version to another within those systems, so I can't give you the general steps here but here is how it works on a GNOME system: click System | Preferences | Windows, and check Select Windows When the Mouse Moves Over Them (this may be referred to as *focus* on your system) and Raise Selected Windows After an Interval (this may be referred as *autoraise*). I move the slider for the latter all the way to the left, for 0.0 seconds. For KDE, as of September 2007 the sequence is K | Control Center | Desktop | Window Behavior; after that, the choices are similar to those described for GNOME above: at Policy, choose Focus Follows Mouse and Auto Raise.

4.5.2 Saving Window Places Between Sessions

If upon bootup you'd like to have the same windows in the same places as in your last session, you can arrange this to occur automatically in GNOME by System | Preferences | Sessions | Session Options and then checking the proper box.

5 Some Points on Linux Usage

To log out in GNOME, select System | Shutdown. It is similar for other desktop managers.

5.1 More on Shells/Terminal Windows

In Microsoft Windows, most work done by most users is through a Graphical User Interface (GUI), rather than in a command window (Start | Run | cmd). In Linux, a lot of work is done via GUIs but also it is frequently handier to use a command window, called a *terminal* window. You should always keep two or three terminal windows on your screen for various tasks that might arise.

You can start a terminal window in GNOME by selecting Applications | Accessories | Terminal; the other desktop managers are similar.

You may be given a choice of several terminal types, say **gnome-term**, **xterm** etc., but it doesn't much matter which one you choose.[3] If you are using **gnome-term**, you may wish to reduce the font size, by

[3] You may like **gnome-term** because it is more easily configurable, as to colors, size, etc.

holding down the Control key and hitting the - key twice.

When you type commands in a terminal window, the program which reads and acts on those commands is called a *shell*. (Thus a terminal window is sometimes called a "shell window.")

I have an introduction to Unix shells, based on the T C-shell, **tcsh** at `http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/ShellIntro.html` and `http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/CShellII.html`.

The default shell in Linux is **bash**. It is very good, but if you are used to using, say, **tcsh**,[4] you can use the **chsh** command in any terminal window to change your login shell.

5.2 Cut-and-Paste Window Operations

The X11 windowing system used in Linux has its roots in 3-button mice. Today, most people have such mice (the middle wheel counts as a button), but if you don't, that's no problem, because Linux does 3-button emulation for you. The middle button is emulated by simultaneously clicking both left and right buttons.

To do a cut-and-paste operations, hold down the left mouse button and drag it to highlight the text you wish to copy. Then go to the place you wish to copy that text, and simultaneously push both the left and right buttons. Generally, more things are cut-and-pastable in Linux than Windows, so this is a big convenience.

5.3 Mounting Other Peripheral Devices

This section explains how to use DVDs, USB devices and so on under Linux. You may wish to review Section 3.3.1 before continuing.

5.3.1 Mount Points

Each I/O device that contains a file system must be *mounted*, i.e. associated with some directory. That directory is called a *mount point*. The files then appear in that directory.

These days most Linux distributions have a designated directory for mount points for DVD/CD-ROMs, USB devices, floppy disks, etc. This will vary from one distribution to another, but typical directory names are **/mnt**, **/media** etc.

You can check what is currently mounted by running the **df** command from a shell window (another good Linux learning experience). The mount points are listed along with the **/dev** files. Also, to list the **/dev** files for all your operating drives including USB flash drives and including drives not mounted, type **sudo fdisk -l**.[5] For more detailed information, such as file system types, just run **mount** without any arguments.

Your machine's internal hard drives, and possibly other devices, will be mounted automatically at boot time. This is controlled by the entries in the file **/etc/fstab**. The details are an advanced topic, but even without understanding everything, you might find it worthwhile to take a quick look at that file. Here is a line from the file on my office machine,

[4] Or if you want to use my shell tutorials, mentioned above.

[5] This might not work in some cases. If **fdisk** doesn't recognize your device, try viewing the file **/proc/partitions**. Your device may appear there, say as **sdb1**. Then run **mount** as shown below, on **/dev/sdb1**.

```
/dev/sda3       /usr/home               ext3    defaults        0 2
```

Here **/dev/sda3** is the third partition ('3') on my first SATA hard drive ('a'). The entry says that this partition has an **ext3** type filesystem in it, and is to be mounted at the directory **/usr/home**. The remaining entries concern things such as backup and file system checks.

When you attach a device to your machine *after* bootup, your system will probably recognize it immediately, and maybe pop up a window showing the device's contents. If you have trouble, you can use the Unix **mount** command. This is an advanced command, but just to give you an idea, a typical usage would be

```
mount -t iso9660 /dev/hdc /mnt/yyy
```

This tells Linux that the I/O device corresponding to **/dev/hdc**, our CD-ROM, should be mounted at the directory **/mnt/yyy**. If that directory doesn't exist, you must create it first, using **mkdir**. The field **-t iso9660** says that the file system type is ISO9660. This is standard for CD-ROMs, and you can probably omit it.

5.3.2 Reading Your DVD/CD-ROM Floppy Drive from Linux

The files are available under the mount point, as explained above. If they contain music or video, you of course will need a program to access them; see Section 6.2.8.

5.3.3 CD/DVD Burning

You can use the shell-based **cdrecord** and **dvdrecord** programs, but it is much easier to use one of the GUI-based programs. I use **gnomebaker** (or sometimes Nautilus, the built-in Linux file manager).

If you do not have that program, you can download it from the Web. Under Ubuntu, for instance, simply type

```
sudo apt-get install gnomebaker
```

Run the program by typing

```
gnomebaker
```

in a shell window. The GUI will come up.

In the bottom right-hand corner, set the size of the CD/DVD (a typical DVD has capacity 4.7G), then click Create Data Disk.

Then go to the Filesystem section in the upper-right portion of the window, and choose your directory. Then for each file you want to burn, click and drag it from the File section at the upper-right to the Data Disk (or Audio Disk) section at the bottom of the window. If you wish to copy an entire directory, just drag the directory name.

To burn an ISO image, select Actions | Burn CD/DVD Image, then select the **.iso** file, and burn.

5.3.4 Using USB Devices

USB drives, including memory sticks, should have their filesystems mounted automatically when you attach them. Use the **df** command to check where they've been mounted (it could be in the directory **/mnt/ /media** etc.).

USB mice should become automatically usable when you attach them.

5.4 A Note on Ubuntu

5.4.1 Root Operations

Ubuntu works like any other Linux distro, except for one important point: Ubuntu does not have a root user account in the classic Unix sense. Instead, whenever executing a command which requires root privileges, one precedes the command by the term **sudo** ("superuser do"). One is then prompted for a password, which is the password for the first user account created at the time of installation.

If you have a lot of root-type work to do in a session, type

```
$ sudo -s
```

to create a new superuser shell, and do your work there.

6 Linux Applications Software

6.1 GUI Vs. Text-Based

Most people prefer to use GUI-based applications. If you are one of them, rest assured that there are tons of them available for Linux.

I do wish to mention, though, that many "super hard core" Linux users prefer to use text-based applications, rather than GUI ones. For instance, I and many others like the **mutt** e-mail utility (Section 6.2.3), which is text-based. Here's why, at least in my view:

- I often access my Linux machine remotely, while traveling.[6] I might be at a university library, for instance, or at the business center in a hotel, and be "stuck" with a Windows machine, and logging in to my Linux machine via an SSH connection.[7] This limits me to text.
- It's very important to me that I use the same text editor for all my computer applications—e-mail, programming, word processing, etc.—so that I can take advantage of all the abbreviations, shortcuts and so on which I have built up over the years. This saves me huge amounts of typing. But most GUI applications, e.g. e-mail utilities, have their own built-in text editors, so I can't use mine.

[6] Which is in fact exactly the case as I write this paragraph.

[7] Though I sometimes use VNC to access a remote image of my Linux desktop. See `http://heather.cs.ucdavis.edu/~matloff/vnc.html`.

- I find that text-based applications often have more features, are better documented, etc. For example, I often wish to automate certain processes, such as uploading files to another machine, and typically text-based programs do this better.

However, in listing my favorite applications in Section 6.2 below, I've made sure to list both text-based and GUI programs.

6.2 My Favorite Unix/Linux Utilities and Applications

6.2.1 Text Editing

I use a modern extension to the **vi** editor, **vim**. This is the version of **vi** which is built in to most Linux distros. See my tutorial at `http://heather.cs.ucdavis.edu/~matloff/vim.html`.

Note: In some Fedora distros, somehow the version of **vim** that is linked to **vi** isn't configured fully correctly. I suggest using **/usr/bin/vim** directly.

Even though **vim** is text-based, it does have a GUI version too, **gvim**. This comes with nice icons, allows you to do mouse operations, etc. Unfortunately, most Linux distros seem to have only the text-based program. To get the GUI, you can download it yourself. In Ubuntu, do

```
sudo apt-get install vim-gnome
```

For this, you may need to edit **/etc/apt/sources.list** and uncommented the lines for Canonical's 'partner' repository

6.2.2 Web Browsing

Your Linux distro will come with a Web browser, probably Firefox, and possibly Konqueror in addition.

I usually use Firefox. But believe it or not, sometimes I use the famous text-based browser, **lynx**. In some cases, it is just plain quicker and easier. Moreover, you can do cool tricks, such as recording keystrokes for later playback, thus enabling one to do certain Web operations automatically.

If you use Fedora, your Firefox system may not be configured for Java. If so, see `http://www.mjmwired.net/resources/mjm-fedora-fc6.html#java`. NOTE CAREFULLY: This site has some very long shell commands, which will not be completely displayed unless you make the browser window quite wide.

Though my main browser is Firefox, I've found the Opera browser often handles special situations better, e.g. playing online videos. Don't overlook this wonderful tool.

If you are short on memory (i.e. RAM), you may wish to use a lightweight browser, such as Galeon (related to Firefox but somewhat fewer features) or Dillo (really bare-bones).

6.2.3 E-Mail

I use the **mutt** e-mail utility. It is very flexible and customizable, and excellent features. For example, it has great search capabilities, important if you are a heavy e-mail user. I 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`.

In Ubuntu, download it by typing

```
sudo apt-get install mutt
```

If you prefer a GUI-based mail utility, many nice ones exist for Linux. Check the Web for these, or use the Thunderbird e-mail utility in the Firefox Web browser suite.

6.2.4 HTML Editing

I usually use Vim, along with some macros I've written for HTML editing, but I sometimes use Amaya, which is a full-featured GUI 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`.

There are many newer and more powerful packages, such as Quanta+, Bluefish and NVu.

6.2.5 Compilers

Some distros come with the GCC suite. Ubuntu, for example, does not, but it can be downloaded via

```
sudo apt-get install build-essential
```

Under Ubuntu, you can install the Sun JDK via

```
sudo apt-get install sun-java6-jdk sun-java6-jre
```

6.2.6 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. For example in **vim** I can type `:make` (which I have aliased to just `M`, or with **gvim** click on the make icon, and the source code I'm debugging will be recompiled. And as I've mentioned, it's important to me that I use the same text editor for all applications, which most IDE would not allow me to do. I use either GDB (try CGDB!) or DDD for my debugging tool. See my tutorials at `http://heather.cs.ucdavis.edu/~matloff/vim.html` and `http://heather.cs.ucdavis.edu/~matloff/debug.html`.

DDD is also usable with my favorite programming language, Python.

However, if you love IDEs, try Eclipse. I've got a tutorial that is more complete than most, at `http://heather.cs.ucdavis.edu/~matloff/eclipse.html`. It can be used with C, C++, Java, Perl, Python and many others.

Another system that has become quite popular is NetBeans.

Also, for KDE users, there is a very well-received IDE named KDevelop. I lean toward Eclipse, though, as it is easier to learn, is cross-platform, and can be used with more programming languages.

6.2.7 Word Processing

I use LaTeX because of its flexibility, its beautiful output, and its outstanding ability to do math. You may like Lyx, which is a great GUI interface to LaTeX which is especially good 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 work with files compatible with the Microsoft Office environment, there is a free suite of programs, OpenOffice, which provide Microsoft compatibility. It is packaged with most Linux distributions.

If you would like something that quickly converts an Office file to rough text form, say to use with e-mail attacments, try Antiword. In Ubuntu, install via

```
sudo apt-get install antiword
```

6.2.8 Playing Movies, Music, Etc.

MPlayer is free and outstanding. Its capabilities are amazingly broad.

The documentation is extensive, and hard to navigate, but here are a couple of things to get you started:

Installation:

It's easy in Ubuntu:

```
sudo apt-get install mplayer
sudo apt-get install mencoder
```

Otherwise, build it yourself, as follows.

One downloads the source code, **MPlayer-1.0pre7try2.tar.bz2** and the codecs, **essential-20041107.tar.bz2**, from `www.mplayerhq.hu/design7/dload.html`.

Unpack the codecs file first,

```
tar xfj essential-20041107.tar.bz2
```

This creates a new directory. Copy the contents of that directory to the directory **/usr/local/lib/codecs** (use **mkdir** to create it if necessary). (Note: There may be legality issues with some codecs. When in doubt about a particular codec, you should obtain it from a site like Fluendo that offers it for a nominal fee, See a discussion at `http://fedoraproject.org/wiki/CodecBuddy`.

Now, unpack the source code file, and go into the directory it creates. Then go through the usual sequence for building open-source software from source:

```
configure
make
make install
```

Note that if you want to use the GUI, the **configure** command should be

```
configure --enable-gui
```

After **make install** is done, you will probably get a message something like

```
 *** Download font at http://www.mplayerhq.hu/dload.html
*** for OSD/Subtitles support and extract to
/usr/local/share/mplayer/font/
*** Download skin(s) at http://www.mplayerhq.hu/dload.html
*** for GUI, and extract to /usr/local/share/mplayer/skins/
```

The fonts are needed for the subtitles (and for the GUI, if you use it). Just the **iso1** font is needed. Download the font package, go to the indicated directory (**/usr/local/share/mplayer/font/** in the above example), and then do the unpack operation. This will produce a subdirectory, e.g. **font-arial-iso-8859-1**.

Viewing a video:

To play a video or audio file, say **x.avi**, type

```
mplayer x.avi
```

If you specify several files, as a playlist, it will play them all. Hit the Enter key if you want to skip the rest of the current file and go to the next one.

To play a DVD, put the disk in the tray (see Section 5.3.2). Then type

```
mplayer dvd://1 -dvd-device /mnt/cdrom
```

where you will have to substitute a different mount point if it is not **/mnt/cdrom** (try running **df** or rummaging around in **/media**).

You have the following controls:

- right and left arrow keys to go back or forward 10 seconds
- down and up arrow keys to go back or forward 1 minutes
- PgDown and PgUp keys to go back or forward 10 min
- left- and right-bracket keys to decrease/increase speed by 10%, or left- and right-brace for 50%; Backspace key to return to normal speed

- Space bar to pause, then . to go forward frame by frame, Space bar to resume play
- f to go full screen
- q to quit

You can use **mplayer**, actually **mencoder**, which comes with the package, to do format conversion, e.g. AVI to MPG, change aspect ratio, and even do some primitive editing.

There are many, MANY, **MANY** different options.

You may wish to try other players, e.g. VLC.

6.2.9 Video Editing

Try Kino, Cinelerra, LiVES and many others.

6.2.10 Image Viewing, Manipulation and Drawing

I use **xpdf** to view PDF files, though Acroread for Linux is available. I like the fact that **xpdf** allows me to copy ASCII text from the file.

For collections of JPEG files and the like, I use **xzgv**, **gqview** and **gwenview**; for viewing a single image, I use **qiv**.

Want something like Adobe Photoshop? The GIMP program is quite powerful, and free. It's included with most Linux distributions.

You can use GIMP to draw, but for "quick and dirty" tasks, I would suggest Dia, at `http://www.gnome.org/projects/dia/`.

6.2.11 Accessing Usenet Newsgroups

Linux distros generally come a text-based newsreader, either **slrn** or **tin**. I generally use **slrn**, but am not that happy with any known newsreader.

In the GUI arena, I sometimes use **pan**. You can download it from `pan.rebelbase.com`.

Firefox's Thunderbird program includes a newsreader too.

6.2.12 FTP

I usually use the text-based **ftp** and **sftp**, the latter being an SSH version for security.

If you do frequent uploads/downloads to/from a particular site and wish to automate them, another text-based program, **yafc**, is excellent.

A very nice GUI program, though, is **gftp**, which you can download from the Web if your Linux system doesn't already have it. In addition to the GUI, this program also has some functionality which ordinary FTP programs don't have.

6.2.13 Statistical Analysis

Use the statistical package that the professional statisticians use—R!

In my opinion from the point of view of someone with a "foot in both camps"—I'm a computer science professor who used to be a statistics professor—the R statistical package is the best one around, whether open source or commercial.[8] It is statistically modern and correct, and it also is a general-purpose programming language.

I have a tutorial on R at `http://heather.cs.ucdavis.edu/~matloff/r.html`.

6.2.14 Video Chat

Currently, this is an area in which many Linux distros need work.

Ubuntu comes with Ekiga, which works best if the person you're chatting with has Ekiga too.

Skype has a Linux version, which many people use, but some have found to have problems.

As of this writing, Google does not offer Google Talk for Linux. However, Empathy can be used.

There are driver issues for some Webcams. Those using the UVC protocol are supposed to work on Linux, with the **uvcvideo** driver that comes with Linux. But again, there may be problems.

6.2.15 Runing Windows Applications from Within Linux

I am simply not a Windows user, but on occasion there ia a Windows program I need to run from within Linux.

The simple way, if it works, is the WINE Windows emulator. Your distro may include it (type `which wine` in a terminal window to check); if not, download it, with for example the Ubuntu command being

```
udo apt-get install wine
```

For more involved applications, you may wish to try one of the virtual machine packages. See `http://heather.cs.ucdavis.edu/~matloff/vm.html` for a brief introduciton.

6.3 Downloading New Software

There is a vast wealth of free software for Linux on the Web. Here's how to obtain and install it.

6.3.1 How to Find It

These days most downloads and installs are done automatically, say with **yum** or **apt-get**, as seen in Section 6.3.2 below. That helps you find it too. If you want to find application Z, instead of plugging "Z" into Google, plug "yum install Z" or "apt-get install Z" so as to narrow down the volume of response.

[8] In some respects, it's even better than S, the commercial product it is based on.

6.3.2 Automatic Download/Installation

In recent years, most Linux distros have made it very easy to download and install new software. In Fedora, for instance, one uses the **yum** command.

For example, to download the program **yafc** mentioned above, one simply types

```
yum install yafc
```

In Ubuntu, there is the **apt-get** command, which works similarly. For instance, to download the **xpdf** PDF viewer, I typed

```
sudo apt-get install xpdf
```

(See Section 5.4.1 for an explanation of **sudo**. Ubuntu may ask you to install from your CD-ROM, but yours may be incomplete. If so, comment out the first line of **/etc/apt/sources.list**; this is the line telling Ubuntu to install from the CD-ROM.)

For those who prefer GUIs, Ubuntu offers the Synaptic package manager.

With both **yum** and **apt-get**, one can direct where to download from, by making the proper entries in the file **etc/apt/sources.list**. For instance, for the R statistical package above, **apt-get** may not find it on its own, in which case we can add a line

```
deb http://cran.stat.ucla.edu/bin/linux/ubuntu gutsy/
```

to **etc/apt/sources.list**, telling **apt-get** that here is an alternative place it can look. (This is for the Gutsy edition of Ubuntu.)

By default **apt-get** will try to retrieve your requested program from your installation CD/DVD. You can change this by commenting-out the line in **etc/apt/sources.list** that begins with

```
deb cdrom:
```

Sometimes it may not be clear which package name to use with **yum** or **apt-get**. For instance, to install the GCC compiler, C library and so on, the command is

```
sudo apt-get install build-essential
```

How did I learn this? I did a Web search for "apt-get GCC."

To install the **curses** library (and include file), do

```
sudo apt-get install libncurses5-dev
```

6.3.3 Debian/Ubuntu .deb Files

The Debian distro of Linux uses its own packaging for downloaded programs, which you'll see as files whose names have a **.deb** suffix. Ubuntu, as a derivative of Debian, uses this too.

Usually you will not need to work directly with these files, since you will use **apt-get** or Synaptic. But if you do download such a file directly from the Web, use **gdebi** to install it; the GUI version is **gdebi-gtk**.

6.3.4 Using RPMs

Though the methods in Section 6.3.2 have now made RPMs less important, you may find that the software you want comes in an RPM package, with a **.rpm** suffix in its name. 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.

7 Dual-Boot Issues

You may wish to change some parameters of your dual-boot process, e.g. change the default OS. You can do this by editing the configuration file for your bootloader.

Most distros today use GRUB as their bootloader. Its configuration file is **/boot/grub/menu.lst**. By the way, note that GRUB's notation for partitions is (drive ID, partition number), so that for instance (hd0,1) means the second partition in the first hard drive.

8 Live CDs or USB-Key Based Linux As Rescue Tools

Among other things, Knoppix has developed a reputation as being useful as an OS rescue/repair tool, including for Windows! And now, most of the live CDs or USB-key based Linux installations can be used this way.

A common usage is to either fix broken files or at least make copies of important user files. It may be, for instance, that Windows is not bootable due to corruption, but by using a Linux rescue CD/USB key, we can access individual files. Here is a typical pattern. One brings up a terminal window and then:

```
sudo -s  # get root privileges; could try su root instead
fdisk -l  # check where the partitions are
# say /dev/sda1 is of interest
mkdir x
mount /dev/sda1 x
cd x
# now you have those files at your disposal
```

In one case, I forgot my password on an Ubuntu netbook. I could fix it as root if I could boot up in Ubuntu recovery mode, but unfortunately the GRUB bootloader was configured with a timeout value of 0 seconds, giving me no way to choose recovery mode. So, I booted up Linux from a USB key (Section 3.5), mounted my Ubuntu file system as above, and then edited the GRUB startup file, **/boot/grub/menu.lst**, changing the timeout value to 5 seconds.

The preceding operations can be done by booting almost any Linux distro, but Knoppix is nicer as it comes with two very nice utilities (both can be obtained separately as well):

- **testdisk**: This does a lot of diagnostics on your hard drive, recover lost partitions, undelete deleted files, fix boot sectors and so on.
- **ntfsfix**: May be able to fix your broken NTFS partition.
- **photorec**: Quite a program! It bypasses your (possibly broken) file system, and looks for files by going through your hard drive literally bit by bit, looking for bytes that encode any of 180 known file types, e.g. .jpg, .avi, .pdf etc.

9 Learning More About Linux

9.1 Wanna Get Good at Linux? Use It for Everything!

The only way to really learn Linux is to use it on a daily basis for all your computer work—e-mail, word processing, Web work, programming, etc.

As you do this, the expertise you'll want to pick up includes: file, directory and mount operations; process operations; roles of system directories (**/usr**, **/etc**, **/dev**, **/sbin** and their various subdirectories, e.g. **/usr/lib**; search paths; network operation and utilities such as **netstat**; and so on.

Don't try to do this all at once. Instead, take your time, and learn these naturally, as the need arises. As you use Linux more and more in your daily computer application work (e-mail, word processing, etc.), the needs will arise as you go along.

And remember, there's lots of help available if you need it.

9.2 Getting Help

9.2.1 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.

9.2.2 The Web

- If you are running Ubuntu or one of its offshoots, the Ubuntu Forums, `http://ubuntuforums.org/` is an excellent resource.
- Linux home page, at http://www.linux.org/ Lots and lots of information is available here.
- `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`
- 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.

9.2.3 LUGs

There are Linux Users Groups (LUGs) in virtually every city. You can join if you wish, or just get to know them casually. They are great sources of help! And by the way, many of them hold monthly Linux Installfests, where you can see Linux being installed or have it installed on your own machine.

10 Troubleshooting

One of Linux's biggest strengths is its stability. If you are tired of getting Windows' infamous "blue screen of death," then Linux is the OS for you. (It is also subject to far fewer virus and other attacks than Windows.) So emergencies are rare, but they can happen. Here are some tips for such cases.

10.1 Tools

Here are some commands you can run in a terminal window that you can use to investigate:

- **ps**: Tells you what processes are running. Typically one uses this with something like the **ax** option.
- **dmesg**: Tells you the major events that have occurred on your machine ever since it was last booted up.
- **lsmod**: This tells you what OS modules are installed, i.e. device drivers and the like.
- **lpq**: Lists the current printer queue.
- **lsusb**: Lists what USB devices are currently plugged in.
- **ifconfig**: Lists network interfaces.
- **iwconfig**: Lists currently operating wireless devices.
- **iwlist**: Lists wireless access points in range.
- **netstat**: Lists current network connections.

10.2 A Program Freezes

If an application program freezes up and you invoked it from the command line within a shell, you can in most cases kill it by hitting Ctrl-c in the terminal window from which invoked it. If this doesn't work, run the "processes" command by typing

```
ps ax
```

in another terminal window, and noting the process number of your program. Say for concreteness that that number is 2398. Then type

```
kill -9 2398
```

to kill the program.

If you have a program named, say, **xyz**, the command

```
pkill -9 xyz
```

kills all running instances of the program.

10.3 Screen Freezes

What if your entire screen freezes up? Again, this should be quite rare, but it is possible. I recommend the following remedies, in order:

-
- In Gnome hit Alt F2, which will bring up a little window in which you can run a command, say **pkill** as above.
- In Gnome, hit Ctrl Alt T, which will create a new terminal window, from which you can kill the offending program.
- Try going to another screen! Linux allows you to switch among multiple screens. In Gnome, for instance, You can switch to the second screen from the first via Ctrl Alt Right, and go back via Ctl Alt Left. Then open a terminal window in the new screen, find the process number of the program and kill the program, as described above.
- In Gnome, try hitting Ctrl Alt Del). This should cause an exit from Linux's X11 windowing system but not an exit from Linux itself. You would then get an opportunity to log in again.

Try NOT to simply poweroff the machine, as that may do damage to your files. It may not be permanent damage, as the OS will try to fix the problems when you next reboot, but don't just pull the plug unless you have no other recourse.

10.4 Inaccessible Partition

Suppose you reinstall or upgrade your Windows OS. This will probably restore the original boot procedure, rendering your Linux files inaccessible.

You can easily access the files by booting one of the live CD distros (Section 2.3 above). Do the following after booting:

```
$ cd /
$ mkdir mylinfiles
$ mount /dev/hda2 mylinfiles
$ cd mylinfiles
$ ls
```

(Of course, you may need to type a different **/dev** file name here; see Section 3.3.1 above.)

At this point, you will be in your Linux file system! You can then go down to your Linux home directory, via **cd home** or something like that.

You can then run GRUB from your live CD. Please check the Web for instructions.

11 If You Are Upgrading or Replacing Another Version or Distribution 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 of the same distribution or changing to a different distribution. First of course you will want to make sure you back up your old files, just in case sometimes goes wrong.

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/local/**. You may also have modified files in **/etc**, such as **/etc/resolv.conf**. You may wish to tar these into a save file too. (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.)

12 Accessing Your Windows Files from Linux

At this point, most Linux distributions, except Fedora/Red Hat, give you access (at least read access) to your Windows partition from Linux. For some of them, they may do this automatically, in which case your Windows partition, say **/dev/hda1** should be visible in the file **/etc/fstab**. If not, mount it yourself:

```
mkdir /dosc
mount /dev/hda1 /dosc
cd /dosc
```

You should now see your Windows files, and should be able to access them on at least a read basis.

For more information, including concerning write access, ss the Linux-NTFS Project, `http://www.linux-ntfs.org/`.

13 If You Wish to Remove Linux

If you wish to remove Linux from your machine, first remove LILO/GRUB as follows. Boot from your the Windows recovery CD that came with your machine. (Make sure you have the boot order set for your machine so that it tries to boot from CD or DVD before a hard drive.) When asked whether you want setup or recovery, hit R for the latter. Choose whichever disk your Windows system is on, probably C:. Change directories to WINDOWS if you are not already there, and issue the FIXMBR command. It will warn you that you will be restoring the Master Boot Record (MBR), which is what you want. Then hit EXIT to finish, and reboot without the CD.

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

Finally, use GParted to recover the former Linux space into your Windows partitions. Typically, this means deleting your Linux partitions (the ones that are not of type FAT32 or NTFS), and then expanding your NTFS partition. Don't forget that the next time you boot Windows, it will ask you if you want a disk check, which you should definitely answer Yes to.