

# Professor Norm Matloff’s Beginner’s Guide to Installing and Using Linux \*

Norm Matloff  
Department of Computer Science  
University of California at Davis  
matloff@cs.ucdavis.edu  
©1999-2013

January 4, 2013

## Contents

<b>1</b>	<b>Background Needed</b>	<b>4</b>
<b>2</b>	<b>Install to Where?</b>	<b>4</b>
<b>3</b>	<b>Which Linux Distribution Is Best?</b>	<b>4</b>
<b>4</b>	<b>Installation</b>	<b>4</b>
4.1	The Short Answer . . . . .	5
4.2	Installing Linux to a USB Key or External Hard Drive . . . . .	5
4.2.1	Installation Method I (for Slax Linux) . . . . .	6
4.2.2	Other Methods . . . . .	6
<b>5</b>	<b>Post-Installation Configuration</b>	<b>6</b>
5.1	Configuring Your Search Path (“Why can’t I run my a.out?”) . . . . .	6
5.2	Configuring a Printer . . . . .	7
5.3	Switching from GNOME/Ubuntu Unity . . . . .	7
5.4	Configuring KDE/GNOME for Convenient Window Operations . . . . .	7
<b>6</b>	<b>Some Points on Linux Usage</b>	<b>7</b>

---

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

6.0.1	Ubuntu Root Operations . . . . .	7
6.1	More on Shells/Terminal Windows . . . . .	7
6.2	Cut-and-Paste Window Operations . . . . .	8
6.3	Mounting Other Peripheral Devices . . . . .	8
6.3.1	Mount Points . . . . .	8
6.3.2	Using USB Devices . . . . .	9
<b>7</b>	<b>Linux Applications Software</b>	<b>9</b>
7.1	GUI Vs. Text-Based . . . . .	9
7.2	My Favorite Unix/Linux Apps . . . . .	10
7.2.1	Text Editing . . . . .	10
7.2.2	Web Browsing and Java . . . . .	10
7.2.3	HTML Editing . . . . .	10
7.2.4	Compilers . . . . .	11
7.2.5	Integrated Software Development (IDE) . . . . .	11
7.2.6	Word Processing . . . . .	11
7.2.7	Playing Movies, Music, Etc. . . . .	12
7.2.8	Video Editing . . . . .	13
7.2.9	Image Viewing, Manipulation and Drawing . . . . .	13
7.2.10	FTP . . . . .	14
7.2.11	Statistical Analysis . . . . .	14
7.2.12	Video Chat . . . . .	14
7.2.13	Running Windows Applications from Within Linux . . . . .	14
7.3	Downloading New Software . . . . .	15
7.3.1	How to Find It . . . . .	15
7.3.2	Automatic Download/Installation . . . . .	15
7.3.3	Debian/Ubuntu .deb Files . . . . .	16
7.3.4	Using RPMs . . . . .	16
<b>8</b>	<b>Learning More About Linux</b>	<b>16</b>
<b>9</b>	<b>Advanced Linux Usage</b>	<b>17</b>
9.1	Dual-Boot Issues . . . . .	17

9.2	Live CDs or USB-Key Based Linux As Rescue Tools . . . . .	17
9.3	Troubleshooting . . . . .	18
9.3.1	Tools . . . . .	18
9.3.2	WiFi Networking . . . . .	19
9.3.3	General Information . . . . .	19
9.3.4	Network Management Tools . . . . .	19
9.3.5	Individual Linux Network Commands . . . . .	20
9.3.6	If You Have a Problem . . . . .	21
9.3.7	A Program Freezes . . . . .	21
9.3.8	Screen Freezes . . . . .	22
9.4	Accessing Your Windows Files from Linux . . . . .	22
<b>A</b>	<b>What Is Linux?</b>	<b>22</b>
<b>B</b>	<b>What Is Partitioning?</b>	<b>23</b>
B.1	Partitioning Using GParted . . . . .	24

## 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.)

Don't worry about the length of this document. You'll probably only need a small part of it.

For some background on the history and significance of the Linux operating system, go to Appendix A

## 2 Install to Where?

I recommend that you install Linux on your hard drive, so that you will *dual-boot* either your old OS (I'll assume Windows from now on) or Linux. After installation, each time you boot up, you will be given a choice of whether to boot Windows or Linux.

Another alternative is to install Linux on a USB key (memory stick) or external hard drive.

Still another choice is to run Linux as a virtual machine. This is not covered in this document, but if you are a UCD student, see <http://csifdocs.cs.ucdavis.edu/documentation/archives/csif-fedora-on-a-> for the easy steps.

By the way, if you have a Mac, you may have the capabilities you need without Linux, since both the Mac OS and Linux are Unix systems.

## 3 Which Linux Distribution Is Best?

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

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** (or one of the many Ubuntu derivatives, such as Linux Mint). 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.

## 4 Installation

Here is the short way to install Linux on your hard drive, dual-booting with Windows.

## 4.1 The Short Answer

Here you will install the Ubuntu distro, using UNetbootin as your installation tool.<sup>1</sup>

For simplicity, I'll assume you wish to install Linux to your laptop.

- Download UNetbootin from its home page <http://unetbootin.sourceforge.net/> to your hard drive. (For further informat on UNetbootin, see <http://sourceforge.net/apps/trac/unetbootin/wiki/guide>.)
- Insert a USB key (memory stick). It needs to have FAT32 format. It probably came that way, but if not then check the Web for how to fix that using your OS.
- Run UNetbootin.
- Click Select Distribution, and choose Ubuntu. Then choose the latest Live version in the window to the right.
- For Type, choose USB Drive, and for Drive, choose the drive in which your USB key is inserted.
- Click OK.
- After the installation to your USB key finishes, choose Exit.
- Leaving your key in the drive, restart your laptop.
- Select the choice labeled something like Try Ubuntu.
- Once Ubuntu boots up, try to use the WiFi: Click on the proper icon at the top right of the screen, and select your wireless network. If none appears, then for now, connect your machine to an Ethernet jack, say at a public library or copy shop.
- Follow directions. If asked whether you want third-party software to be installed, say yes.
- Reboot (remove the USB key when the screen goes dark).
- If WiFi didn't work above, it should work now. You may have to click a pop-up window that asks if you want to use the proprietary drivers.

During the installation process, there may be some mention of disk partitions. You should not have to take action, but if you wish to know about partitions (very useful!), see Appendix B.

## 4.2 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 whenever you want to use Linux. (This is not the same as the USB key created from UNetbootin, which is only temporary.)

---

<sup>1</sup>Unfortunately, UNetbootin does not produce Mac-bootable USB keys.

### 4.2.1 Installation Method I (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**.

### 4.2.2 Other Methods

You can use UNetbootin (Section 4.1), but you’ll need to make your USB installation **persistent**; see <http://sourceforge.net/apps/trac/unetbootin/wiki/guide>.

There are methods to construct your USB installation “by hand” from an ISO file. This is complicated, and will not be pursued here.

## 5 Post-Installation Configuration

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

### 5.1 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
```

## 5.2 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.

## 5.3 Switching from GNOME/Ubuntu Unity

I personally don't like the Unity window manager in GNOME. Many others feel the same way. So, Ubuntu gives us other choices. To set them up, do

```
sudo apt-get install gnome-shell
```

At your next start, the login screen will show a symbol next to choices of login names; choose Gnome Classic (No Effects) or whatever you like; experiment to find one that suits you.

## 5.4 Configuring KDE/GNOME for Convenient Window Operations

### 5.4.1 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 GNOME in Ubuntu 12.10 or later:

Open a terminal window (ctrl-alt-t), and type

```
sudo gsettings set org.gnome.desktop.wm.preferences auto-raise true
sudo gsettings set org.gnome.desktop.wm.preferences focus-mode 'mouse'
```

Then log out and back in. You only need do this once.

You get check these settings using **get** instead of **set**, or use **reset** to revert to the original values (**false** and **'click'**), e.g.

```
sudo gsettings reset org.gnome.desktop.wm.preferences auto-raise
```

## 6 Some Points on Linux Usage

### 6.0.2 Ubuntu 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.

### 6.1 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 typing ctrl-alt-t.

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, at <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/ShellIntro.html> and <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/CShellIII.html>. These are based on the T C-shell, **tcsh**, but at least in the case of the first tutorials, most of the material also applies to the more popular **bash** shell.

On the other hand, **bash** is more popular, so you

### 6.2 Cut-and-Paste Window Operations

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.

### 6.3 Mounting Other Peripheral Devices

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

### 6.3.1 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**.<sup>2</sup> 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,

```
/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.

Use **umount** to unmount. It's not safe to remove a USB device without running this first.

### 6.3.2 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.

---

<sup>2</sup>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**.

## 7 Linux Applications Software

### 7.1 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 ??), which is text-based. Here’s why, at least in my view:

- I often access my Linux machine remotely, while traveling.<sup>3</sup> 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.<sup>4</sup> 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.
- 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 7.2 below, I’ve made sure to list both text-based and GUI programs.

### 7.2 My Favorite Unix/Linux Apps

In Ubuntu, one downloads new apps using **apt-get**, which I’ll use in my examples here. The same is true for other distros derived from Debian. In Fedora, use **yum**.

#### 7.2.1 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

---

<sup>3</sup>Which is in fact exactly the case as I write this paragraph.

<sup>4</sup>Though I sometimes use VNC to access a remote image of my Linux desktop. See <http://heather.cs.ucdavis.edu/~matloff/vnc.html>.

```
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

### 7.2.2 Web Browsing and Java

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

I usually use Firefox. Chrome is nice, but I really like the plugins available for 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 Ubuntu, your system may not be configured for Java in Web browsing. If so, do

```
sudo apt-get install openjdk-7-jre
sudo apt-get install icedtea-7-plugin
```

### 7.2.3 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.

### 7.2.4 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
# may need to do this separately:
sudo apt-get install libc6-dev # C library
```

### 7.2.5 Integrated Software Development (IDE)

For programming work, I rarely use IDEs, as they are slow to load, take up too much space, and often don't allow me to use my own text favorite editor.

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](http://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.

For R programming, RStudio and StatET are both first-rate.

## 7.2.6 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>.

Install by running

```
sudo apt-get install texlive
# you may also need:
sudo apt-get install texlive-fonts-recommended
```

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 attachments, try Antiword. In Ubuntu, install via

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

## 7.2.7 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](http://www.mplayerhq.hu/design7/dload.html).

Unpack the codecs file first,

```
tar xvj 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.

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.

### 7.2.8 Video Editing

Try Kino, Cinelerra, LiVES and many others.

### 7.2.9 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. Others popular with Linux are **evince**, **okular** and **MuPDF**.

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/>.

### 7.2.10 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.

### 7.2.11 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.<sup>5</sup> 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>.

Install via

```
sudo apt-get install r-base
```

### 7.2.12 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.

### 7.2.13 Running Windows Applications from Within Linux

I am simply not a Windows user, but on occasion there is 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 introduction.

## 7.3 Downloading New Software

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

### 7.3.1 How to Find It

These days most downloads and installs are done automatically, say with **yum** or **apt-get**, as seen in Section 7.3.2 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.

---

<sup>5</sup>In some respects, it's even better than S, the commercial product it is based on.

### 7.3.2 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 6.0.1 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
```

### 7.3.3 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**.

### 7.3.4 Using RPMs

Though the methods in Section 7.3.2 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.

## 8 Learning More About Linux

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.

- 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](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://linux.org).

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.

## 9 Advanced Linux Usage

### 9.1 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.

### 9.2 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 4.2), 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.

## 9.3 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.

### 9.3.1 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.

### 9.3.2 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.

### 9.3.3 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 **ESSIDs**.
- 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.

### 9.3.4 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 network 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
```

### 9.3.5 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).

### 9.3.6 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.

### 9.3.7 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.

### 9.3.8 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 **pskill** 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.

## 9.4 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/>.

## A 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. The Apple Macintosh operating system is based on a form of Unix, and the Android system is based on Linux.

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 are promoting it, and as mentioned Linux is a leading corporate choice for Web servers. Linux is the main operating system used at [Google](#), and in fact they developed their own version of Linux, Goobuntu (a play on Ubuntu, one of the most popular versions of Linux).
- 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, 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.

## **B What Is Partitioning?**

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 (not just for Linux) 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.

## B.1 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 (or USB key), 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 B.)
- 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 partitioner 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.