A Basic Linux Tutorial

John E. Floyd
University of Toronto[1]

April 27, 2006

[1]I would like to thank some local gurus who have helped me figure out enough about Linux over the years to enable me to write this. Thanks go to Dan Derkach and, in an earlier era, Adam Iles, as well as to Joe Porrovecchio, Do An Vu, Pete St. Onge and my colleague John Maheu. I will gladly supply my LaTeX code to anyone who wishes to extend and refine this document with a view to making the revised version also freely available.

Contents

1 Introduction

This document provides a discussion of the basic features of the Linux operating system for individuals who have Linux on their computers and are in the process of learning how to use it and for Linux aspirants who are considering setting up Linux on their machines. Since, for practical purposes, Linux works the same as its distinguished ancestor UNIX and its cousin FreeBSD, the material here will be equally useful for people who find themselves logged in over the Web to computers that use these related operating systems. You may find it useful to temporarily set up a mini-version of Linux in memory on a computer that uses another operating system in order to experiment with some of the commands discussed here, or to copy files from the Web to a computer on which you plan to set up Linux. This can be done using either Tom's Root Boot Diskette or BasicLinux without interfering with the operating system (or even having one) already installed on that computer.

Tom's Root Boot Diskette and BasicLinux are mini-distributions that fit onto two floppy disks in the case of BasicLinux or one floppy disk in the case of Tom's Diskette. You can boot either mini-distribution from Drive A on any IBM compatible computer with 8 megs or more of RAM. Both work entirely out of a RAM disk (i.e., a virtual disk they create in the memory of the computer). Because the operating system and associated RAM disk is merely a 'figment of the computer's imagination', you can not do any damage to the file system or operating system actually on the host computer as long as you do not mount any of that computer's actual disks with the file system in RAM or use the `fdisk` program to repartition or reformat those disks.

Each of the two mini-distributions has its advantages. If you are going to set up your own 'real' Linux distribution on an old discarded or virtually free computer and are going to have to access the Internet using a modem, the BasicLinux distribution is the one to use since it contains a relatively easy way to set up a connection to your ISP so that you can download the necessary files for your 'real' Linux distribution from the Web. If your computer has an Ethernet card and can be physically connected to the Web through a network that supports DHCP, your best bet is to use Tom's Root Boot Diskette. When you connect your computer and boot it with Tom's Diskette it will in many cases automatically detect your Ethernet card and set up for you the Internet connection you will need to download the files required for the 'real' Linux distribution you plan to set up on

that computer.[1] BasicLinux includes a browser while Tom's Diskette does not, and it also has a more easy-to-use editor than Tom's disk. This entire document assumes that you have a floppy drive on your computer.

2 What Everyone Should Know About DOS

Before becoming acquainted with Linux, everyone should have a minimal working command over the original widely-used PC operating system, DOS. If you are working in some version of MS-Windows, you will know you are in DOS when you have on the screen a text window called 'MS-DOS Prompt', or 'Command Prompt' or 'Terminal Window' containing the prompt

```
C:\>
```

Alternatively, if you happen to have booted up DOS using a floppy disk in Drive A you will see the prompt

```
A:\>
```

From that point you can use the command

```
C:\> dir
```

to find out what files or directories (called 'folders' in Windows-speak) are on that disk. Or you can choose variants of the `dir` command such as

```
C:\> dir *.exe
C:\> dir *.
C:\> dir /p
C:\> dir /w
```

to respectively list all files that have the suffix `.exe`, list all the files that have no suffix, most of which are typically directories or subdirectories of the directory you are currently in, list the files and directories a screen-full at a time (`/p`), or list the files and directories in columns on the screen (`/w` means 'wide'). Upper-case and lower-case letters can be used interchangeably in DOS.

You can move from `C:\>` to a directory called, say, `OLDWORK` on the C: drive by entering

[1] It finds the Ethernet cards on my two computers connected to the University of Toronto backbone as well as my home computer connected to Rogers cable through a router and establishes an Internet connection in every case.

```
C:\> cd oldwork
```

and then to work you are doing for Bob by moving further down the directory tree to the subdirectory `BOB` by entering

```
C:\oldwork>cd bob
```

You can create a new directory called `FINISHED` and move to it using the commands

```
C:\oldwork\bob> mkdir finished
C:\oldwork\bob> cd finished
```

To move back up the directory tree you simply type `cd ..` at the prompt. One dot by itself on the command line is a reference to the current directory you are in (the working directory) while two dots together refer to the directory 'above' that current or working directory. For example, you can move back up from the directory `FINISHED` to the directory `BOB` and then erase or remove the directory `FINISHED` (provided you haven't put files in it) using the commands

```
C:\oldwork\bob\finished> cd ..
C:\oldwork\bob> rmdir finished
```

You can copy, rename, and erase files in a directory using the following types of commands

```
C:\myfiles> rename newfile.txt newfile.jnk
C:\myfiles> copy oldfile.txt C:\oldwork\bob\*.*
C:\myfiles> copy C:\oldfile\bob\badprog.exe .
A:
A:\> erase bigfile.doc
C:
C:\myfiles> erase A:oldjunk.wp5
C:\myfiles> cd ..
C:\> erase *.jnk
```

The first line above renames a file called `newfile.txt` as `newfile.jnk`. You are in the directory `C:\myfiles` when you execute this command. The second command copies a file named `oldfile.txt` from the current working directory `C:\myfiles` to the subdirectory `bob` of the directory `oldwork`. The third line copies the file `badprog.exe` from the subdirectory `bob` of the directory `oldwork` to the current working directory, referred to by the dot

at the end of the command line.[2] The next command switches the working directory to the floppy disk in Drive A and the command that follows erases the file `bigfile.doc` on that disk. Then we switch the working directory back to `C:\>`, which puts us back in the directory `C:\myfiles>` where we were when we switched to A, and then erase the file `oldjunk.wp5` from the floppy disk in Drive A. Then we move up the directory tree to the root directory `C:\>` and erase all files having the suffix `.jnk`.

To format a floppy disk in Drive A we execute the command

```
C:\> format A:
```

To format that disk and in the process make it into a boot disk, we enter the command

```
C:\> format A:/s
```

The `/s` switch tells the operating system to copy the file `COMMAND.COM` together with some hidden files to the floppy disk being formatted in Drive A. This will not work on some of the newer versions of MS-Windows, such as XP, which no longer are based on DOS. However, most old computers will have on them a version of DOS from which bootable floppy disks can be made. And all versions of MS-Windows contain a command prompt screen from which most of the commands in this section have the meanings here assigned to them. If you are working on a computer whose operating system is Windows-XP, you can still obtain a DOS boot disk using FreeDOS. Use Internet Explorer to obtain the files `fdos1440.img` and `rawwrite.exe` from my ftp site by entering the URLs

```
ftp://islamey.economics.utoronto.ca/pub/jfloyd/linux/fdos1440.img
ftp://islamey.economics.uroronto.ca/pub/jfloyd/linux/rawwrite.exe
ftp://islamey.economics.uroronto.ca/pub/jfloyd/linux/ne.com
```

The last command obtains an old text editor that will work with FreeDos. Then put a fresh floppy disk in Drive A: and execute the command `rawrite` and follow the prompts to write the image on `fdos1440.img` to the floppy disk. After doing this, boot the system with the newly created floppy. A ram-disk called Drive A: will be created in the computer's memory and the floppy Drive A: will be renamed Drive B:. Move to the FreeDos directory on the new Drive A: and, after putting a fresh new floppy in what is now Drive B:, execute the command `sys b:`. This will make this new floppy disk into a boot disk. Copy `sys.com`, `format.exe`, `fdisk.exe` and `fdisk.ini`

[2] In DOS, this dot can be omitted although a comparable dot is required in Linux.

to your new boot disk in Drive B: and then move to the drivers directory on Drive A: and copy the file `himem.exe`. Your FreeDos boot disk will then be complete. You can add an editor to this disk by copying to it the file `ne.com`. If we then take the boot floppy disk and insert it into Drive A of any IBM compatible computer and turn that computer on it will boot the DOS operating system that is on the floppy and leave us at the prompt

```
A:\>
```

If we copy the file `SYS.COM` to the floppy disk from the computer we created it on, as was done in the case of FreeDos above, we can set up that version of DOS on Drive C: of a second computer by booting it with the floppy and then entering the command

```
A:\> sys C:
```

To check the quality of a floppy disk in Drive A and correct any errors on it we can insert that floppy into Drive A of a booted and properly configured DOS or MS-Windows computer and execute the command (in a DOS window)

```
C:\> scandisk A:
```

and follow the prompts.

Sometimes we will need to create a text file using DOS. We do this by entering the command

```
C:\> edit filename.txt
```

to load the standard DOS editor and then typing in the appropriate text. The standard DOS editor can be operated by pressing the `ALT` key plus the first letter of the relevant menu item. Of course, we could also make a text file using the program NotePad on a computer booted with Windows, version 95 or later. In FreeDos we would edit a file using the command

```
C:\> ne filename.txt
```

Text files are extremely important in working with computers at a very basic level. Not only can you read and create text files using a text editor, you can have the computer print a text file on the screen by executing the command

```
C:\> type filename.txt
```

We usually name text files with the suffix `.txt` although there are important exceptions. Text files with the suffix `.bat` are called batch files. They are executable files that contain a group (or batch) of DOS commands of the sort outlined above which the computer will execute whenever we type the root name of the batch file—i.e., the part of the filename that comes before the suffix `.bat`. The most important batch file is the file `autoexec.bat`. Whenever the computer is booted into DOS (or a DOS window is opened) it looks for the `autoexec.bat` file and executes the commands that file contains if it is present. A typical `autoexec.bat` file will contain the following type of material:

```
@ECHO OFF
SET PATH=C:\;C:\utils;C:\rats;C:\shazam;C:\gnuplot;C:\wp;
DOSKEY
```

The line `@ECHO OFF` tells the computer not to print on the screen what is happening as the file executes. The line starting with `SET PATH` tells the computer where to look to find files that the user is trying to execute. The first place it will look is on the root drive `C:\`, and then it will look in the directory `C:\utils`, and then in the directory `C:\rats`, and so on. If you want DOS to look in a particular directory for executable files, you must load the `autoexec.bat` file into your text editor (create a backup copy of it first!) and add that directory to the path command. The command `DOSKEY` loads a neat little program that allows you to access all commands previously made at the DOS prompt by pressing the up-arrow on the keyboard.

Executable files in DOS all have one of three suffixes—`.bat`, `.com` or `.exe`. Files with the last two suffixes are not text files—they are binary files (i.e., files written in machine language).[3]

One other important text file on a DOS based computer is `config.sys`. This file contains information about how the computer is to be configured—the computer always reads and follows the instructions in this file when it boots.

A very important DOS program that you may need to use when setting up a Linux computer from a DOS partition is FDISK. You may need this program, which is contained in the file `FDISK.EXE`, to create a DOS partition on your target computer before setting up Linux partitions. Of course, you will set up Linux partitions using the fdisk program in Linux. You can use

[3]If you are checking your mail using a DOS-Windows computer and someone sends you a file with one of the above three suffixes, don't open it! It probably contains a virus which wreak havoc on your computer when you click on the file and open (i.e., execute) it!

BasicLinux or Tom's Linux to copy necessary files from the Web to your DOS partition and then, if you choose that route, boot the computer into DOS and install Linux from your DOS partition. Subsequently, you can configure your computer to boot into either DOS or Linux.

When you set up your DOS boot disk for booting other computers that may not have DOS installed on them, you should make sure that it contains the following files (in addition, of course, to `COMMAND.COM` and the hidden files):

```
FORMAT.COM
DISKCOPY.COM
SYS.COM
CHKDSK.EXE
SCANDISK.EXE
FDISK.EXE
```

The first of these files or programs is used for formatting floppy disks or partitions on a hard disk. The second is used for copying the of contents of one floppy disks to another. As noted above `SYS.COM` is used to copy the operating system from your boot floppy to a newly created C: Drive on another computer to enable that computer to boot into DOS off that drive. Both `CHKDSK.EXE` and `SCANDISK.EXE` are programs for checking and correcting errors on floppy disks and hard disks. We now turn to a discussion of how to use the program `FDISK.EXE`.

2.1 Using DOS FDISK

FDISK is a program for setting up and removing partitions on the hard disk on your computer. Every computer has at least one hard (or fixed) disk and many have two or more. Before using disk formatting programs you must always back up any files that you want to keep—when you remove existing partitions, any data on them will be lost! To use FDISK, make sure that you have the file `FDISK.EXE` on your boot floppy disk and then use that disk to boot the computer into DOS from Drive A. Then execute the command

```
A:\>fdisk
```

and the program will load. If you are booting with an MS-Windows boot disk you may be asked whether or not you want to enable large disk support. Answer no if you only want to create a small DOS partition and want it to be accessible by older versions of DOS and Windows95 boot disks. You will then be presented with a menu from which you can choose options.

From this point it will be obvious how to proceed, although there are a few things you should know. First, the active partition is the one you want the computer to boot off. Second, if you have more than one disk drive installed on the computer there will be a menu item called 'Change the current fixed disk drive' which you can choose to move to the disk you want to format (if it is not disk 1). Third, there are two types of partitions, *primary* and *extended*. You are allowed to have no more than four primary partitions. Alternatively, you can choose to set up as many as three primary partitions and one extended partition. Within that extended partition you can create a number of *logical* partitions. If you are going to set up Linux, you will need at most one primary partition. Leave the remainder of disk 1, and all of any additional disks unpartitioned. You will partition those with the Linux fdisk program.

3 Acquiring and Setting Up BasicLinux

An up-to-date version of BasicLinux can be obtained from the web-site

`http://distro.ibiblio.org/pub/linux/distributions/baslinux/`

You can download the zip file containing the distribution to the hard drive on any computer on which MS-Windows 95 or later is loaded and unzip it. You can run BasicLinux from there on that computer without doing any damage—though you should ask the computer's owner first! To set up the floppy disk version you need two clean formatted floppy disks. Just follow the instructions on the BasicLinux web-site.

If you are setting up your BasicLinux disks from a machine whose owner doesn't want you unzipping files and running programs, you can obtain a somewhat older BasicLinux distribution off my web-site. You should also setup this older version off my web-site if you are going to be dialing in to the University of Toronto system to access the Web—the newer versions of BasicLinux cannot be configured easily for dialing into that system. To setup the older version, put the first of your two floppy disks in the A-Drive: of any DOS or MS-Windows machine and make it into a boot disk by executing the command

```
format a:/s
```

You then need to find the file `HIMEM.SYS` on the same computer and copy it to the boot disk in Drive A. Next, use the text editor to make a `config.sys` file on the disk in Drive A containing the following single line of text

```
DEVICE=HIMEM.SYS
```

Make sure that your second floppy disk is cleanly formatted—it should not be a boot disk.

Now point the browser at my ftp site

```
ftp://islamey.economics.utoronto.ca/pub/jfloyd/linux/
```

and download `baslin1.zip` and `baslin2.zip` and unzip them onto your first and second floppies, respectively. You might want to rename the file `boot.bat` on the boot disk as `autoexec.bat` so that BasicLinux will boot up automatically when you start a computer with your new BasicLinux boot floppy in Drive A.

4 Acquiring and Setting Up Tom's Root Boot Diskette

To obtain Tom's Root Boot Diskette use a public or a friend's computer with Internet access to go to the following Web location:

```
http://www.toms.net/rb/
```

Click on the download URL and copy the file `tomsrtbt-2.0.103.dos...` to a temporary directory of the hard disk of the computer you are using. Then, after obtaining a fresh new floppy disk that must contain no bad sectors (check it with scandisk!), execute the command `install` on that computer. You will be prompted to insert the floppy disk in Drive A and Tom's mini-distribution will be automatically written to that disk. The disk is then ready for use!

5 Getting Around in The Linux File System

The Linux file system extends outward—I visualize it as downward—from the root directory, commonly denoted as `/`. When you boot up with Tom's Diskette you will be presented with a prompt `/#` from which you can enter a 'list files' command `ls`, which is equivalent to `dir` or `DIR` in DOS. In contrast to DOS, upper-case and lower-case letters are perceived as separate letters in Linux. If you enter `LS` on the command line you will receive an error message. When you enter the command `ls` the screen will appear as follows:

```
/# ls
bin         etc         mnt             sbin            usr
cdrom       fl          not             tmp             var
dev         lib         proc            tomsrtbt.FAQ
/#
```

You are in the root partition as indicated by the `/` character that precedes the `#` in the command prompt. All the above names except `sbin` and `tomsrtbt.FAQ` should appear on the screen in blue, with `sbin` appearing in turquoise and `tomsrtbt.FAQ` appearing in green. All of the blue words signify directories. `tomsrtbt.FAQ` is a text file that explains Tom's distribution and `sbin` is simply a link to the directory `bin`. Links are files that are nothing but referrals to other files. A more detailed listing can be obtained by adding switches to the `ls` command as follows:

```
/# ls -l
/# ls -a
/# ls -l -a
```

Use of the `-l` switch will cause the `ls` command to present details about the files and directories and the `-a` switch will cause a listing of hidden files and directories, which start with a period (the only one in Tom's distribution is the file `.exrc`).

If you list the files in a directory like `bin` or `dev` using the `-l` switch you will get a list that will take up more than one screen. In this case you can send the list to a text file using the command

```
/# ls -l > list.txt
```

and then view that text file with the text editor. Alternatively, you can *pipe* the output from the list command to the program `more` which will present the list one screen at a time. The command to use is

```
/# ls -l | more
```

where the character that appears as a vertical line redirects (or pipes) the output of `ls` to the program `more`. In typical Linux systems you can also scroll through text files by using the command `less`. This commands differs from the `more` in that it allows you to scroll up through the file as well as down using the arrow keys. You could also learn about the details of Tom's distribution by entering the command

```
/# more tomsrtbt.FAQ
```

If you boot up BasicLinux you will be presented with the prompt `~<#>` and upon executing the list files command your screen will look like this:

```
~<#>ls
hotlist.html  mail      netsetup
~<#>
```

where `mail` is a directory which appears in blue and `netsetup` is a file that appears in green. The item `hotlist.html` is a file that appears in the normal white print. It turns out that since you logged in as 'root' user you are immediately placed in the root user's directory. You also log in as root in Tom's Linux distribution but that distribution contains no directory for the root user. To get to the root directory (as distinct from the root user's directory) in BasicLinux you issue the change directory command

```
~<#>cd ..
```

and you will then presented with the prompt

```
/<#>
```

from which, if you list files, you will see

```
# ls
bin         hd          mnt          sbin         var
dev         lib         proc         tmp
etc         lost+found  root         usr
/#
```

with all the items being directories and appearing in blue. Note the directory `root`. You can move to that directory using the command

```
/#cd root
```

and you will find yourself right back in the root user's directory where you started.

Complete distributions of Linux, unlike Tom's, also have 'root' directories that are the home directory of the root user. They also have a directory called `home`, which contains the home directories of all users on the system other than the root user. You will probably want to set your Linux system up having two users, a root user and another regular user representing yourself, called `john` in my case. You will have separate passwords for each user and will always log in as yourself for doing your regular work. Only when you need to make changes in your Linux system or perform special

operations will you log in as root. As yourself you can only operate on files belonging to yourself as regular user—you cannot accidentally screw up the system files. As root user you can do a lot of damage if you are not careful.

In all distributions you will find a `/bin` directory which will include executable files that all users can use. The slash `/` in front of the word `bin` specifies that the `bin` directory is in the root directory `/`. You could have other `bin` directories such as, for example, `/usr/local/bin` which is three branches below the root directory on the directory tree. The directory `/sbin` usually contains executable files used by the Linux system itself. Linux systems have programs called *daemons* which lurk in the background and spring into action whenever appropriate conditions occur. For example, if a user issues a print command, the line-printer daemon goes into action and deals with it. The directory `/etc` contains configuration files for the whole system—nobody but the root user can write to or create files in this directory. The directory `/dev` contains files that activate all devices (printers, modems, disk partitions, etc.) that are present on the computer—in Linux, these devices are treated the same way as ordinary files except that usually only the root user is allowed to read from and write to (i.e., use) most of them. The `/lib` directory houses 'library' files containing code that many different programs will need to access. The directory `/var` is where system log files are kept. These files record everything that happens when you boot your computer or when anyone uses it—this is unlike MS-Windows where the comparable files are hidden from view. You can read these log files (as root user only) to see who is trying to break into your system or figure out why, for example, your system rebooted on its own in the middle of the previous night. The `/tmp` directory houses, as you might expect, 'temporary' files and `/usr` is the directory, which typically contains subdirectories called `local/lib` and `local/bin`, where the programs created and set up to suit the particular tastes of ordinary users are normally housed. The other directories, such as `/mnt` (meaning mount), `/hd` (probably meaning hard disk), and `/cd` or `/cdrom` are usually empty and available to mount on hard disk partitions, floppy drives, or CD-ROM's. In Linux the file system is not organized according to Drives such as C: or D: as in DOS or Windows, but can be 'mounted' in pieces and distributed across various hard disk partitions in many different ways. In linux, the first hard disk is known as `/dev/hda` and the second as `/dev/hdb`. The first partition on `/dev/hda` is known as `/dev/hda1`, the second partition as `/dev/hda2`, and so forth. On some Linux systems the CD-ROM drive is known as `/dev/cdrom`. On others, it is denoted simply as an additional hard disk—for example, `/dev/hdc` on a computer that contain two hard disks.

As previously noted, you can switch focus between directories using the `cd` command. In Linux you copy files using the `cp` command as follows:

```
/<#>cp root/hotlist.html .
/<#>cp -i hotlist.html usr/bin/
/# cp tomsrtbt.FAQ tmp
```

The first line copies, in BasicLinux, the file `hotlist.html` from the root user's directory to the current working directory, denoted by the period at the end of the command line. The second line, also in BasicLinux, copies that same file from the root directory of the system to the subdirectory `bin` in the directory `usr`. The parameter `-i` specifies that a prompt be given if the file is going to be copied on top of another file of the same name. The third line, in Tom's distribution, copies `tomsrtbt.FAQ` from the root directory `/` to the directory `tmp`. If you are copying files to some subdirectory far down the tree from the root directory, the easiest thing to do is to `cd` yourself to the directory in which you want to place the file and then issue the copy command from that directory as follows:

```
/home/john<#>cp -i /usr/local/lib/libfile.so .
```

Here you specify that the file to be copied is `libfile.so` in the directory `/usr/local/lib`. The `/` in front of the word `usr` is necessary to specify that the path to the file starts from the root directory of the system (when working from the root directory you do not need to include this), and the `.` at the end of the line separated by one space from `so` specifies that the file is to be copied to the current directory. And we can see from the command line prompt `/home/john<#>` that the current working directory is the directory of user `john` which is a subdirectory of the directory `home` that is a subdirectory of the root directory `/`.

To remove files we use the command `rm` as follows

```
/<#>rm -i root/hotlist.html
/<#>rm -i *
/<#>rm -i .*
```

where the switch `-i` specifies that you be asked for confirmation before that file is deleted. The first command, when executed from the root directory `/`, removes the file `hotlist.html` from the root user's directory `root`. If the working directory is not the root directory, a `/` would have to be placed at the beginning of the word `root`. Of course, the user issuing the command will have to be the root user. Nobody else can delete the root user's files.

The second command removes all files in the current or working directory while the third removes all hidden files. Always use the `-i` switch because it forces you to be stupid twice in a row in order to delete a file you really wish to keep.

To create and remove directories, do the following:

```
/<#>mkdir junk
/<#>rmdir junk
```

6 Making and Using Shell Scripts

All of the commands above are called *shell* commands. The shell on a Linux computer is a program that takes commands from users and passes them along to the core of the operating system, called the *kernel*, which then gives appropriate directions to the computer's hardware. There are many different alternative shells that can be used in Linux—the default one in most Linux distributions, including Debian, is the *bash* shell. This is in contrast to DOS where the shell and the operating system are completely integrated.

In DOS we had `.bat` or batch files that executed a batch of commands in sequence. In Linux we do the same thing with shell scripts. A shell script is simply a text file containing shell commands that we want the computer to execute (or, more correctly, the shell to tell the computer to execute) when the name of the shell script is entered on the command line. Suppose that we want to construct a silly message that will appear on the screen in BasicLinux or Tom's Linux whenever the user types the letter `a` by itself on the command line. First we construct a text file called, say, `message.txt` containing the message we want to deliver. Then we need to construct a program to deliver the message. In BasicLinux we create a text file using the default editor on the system by executing the command

```
/<#>edit message.txt
```

Then we type on the screen the line

```
Love Linux!! Hate Microsoft!!
```

and save the file and exit the editor by pressing simultaneously the `Ctrl` key and the letter `x` as indicated by the help lines along the bottom of the screen. In Tom's Linux we have a problem. If we type the above edit command we will get an error message. Tom requires that we edit files using the venerable old UNIX editor `vi` which some people think one has to be a masochist to

use. You will have to consult Appendix A to this document to learn the minimal things you need to know about `vi`. Actually, it is useful to learn the basics of how to use this editor because it is found on every Linux and UNIX-type system in existence (except BasicLinux!). If you have an account on any of your University's computer systems you will be able to use that editor to do anything you might need to do.

Since we don't want our user to see the file `message.txt` on boot-up, we move it to the directory `lib` using the command (from within the root directory)

```
/<#>mv -i message.txt lib
```

The `mv` command has another use that we should mention—to rename a file by 'moving' it to a different name.

Now we need to make the program that will run when the user presses the letter `a` by itself on the command line. Obviously, we will have to call this program `a`. So enter the command

```
/<#>edit a
```

to begin constructing a text file with this name. We essentially want the shell to type the contents of the file `message.txt` on the screen when this program is called (i.e., when the letter `a` is typed on the command line). In Linux, the command that types the contents of a text file on the screen is called `cat` (as opposed to `type` in DOS). So our program file should contain one line

```
cat /lib/message.txt
```

This tells the shell to write the contents of the text file `message.txt`, to be found in the directory `lib`, to the screen.

In order for this file to be executed when the user types its name on the command line two things need to be done. First, the file must be copied to a directory on the PATH along which the shell will look to find executable files. To find out what directories are in the PATH we execute the command

```
/<#>echo $PATH
```

to which we will receive the reply in BasicLinux

```
/sbin:/usr/sbin:/bin:usr/bin:
```

The reply in Tom's will be similar but shorter. Obviously, a suitable place to put our executable file is in the directory `/bin`. We do that using the command

```
/<#>mv a bin
```

The second thing we have to do is make the file `a` executable. We do this (always as root user in a 'real' Linux distribution) by executing the command

```
/<#>chmod +x a
```

We can tell that the file `a` is executable by looking at the information presented when we apply the list files command to it with the `-l` switch. The screen will look as follows:

```
/<#>ls -l a
-rwxr-xr-x  1  root      root  21  May 24 21:52   a
```

Reading from the extreme left along the second line, the one representing the shell's response to the `ls` command, the `-` character signifies that `a` is a file. If it were a directory there would be a `d` in this position instead of a `-`. The next three characters `rwx` tell us that the owner of the file can read, write to it, and execute it. The next three characters `r-x` tells us that the group of users that has access to this file can read and execute it but not write to it. The final `r-x` tells us that everyone else (i.e., all other users) can read and execute the file but not write to it. The number 1 tells us that there is one occurrence of this file (i.e., no links to it), the left-most word `root` tells us that the 'owner' of the file is the root user, and the right-most word `root` tells us that the group of users having access to the file consists only of the root user. The number `21` tells us that the size of the file is 21 bytes. This is followed by the date and time the file was created, and by the name of the file.

Now let us see what happens when we go to the root directory on the file system and type the letter `a` on the command line.

```
/<#>a
Love Linux!! Hate Microsoft!!
```

The program works!

You can change the ownership of a file, say to user `john`, by executing the following command the shell prompt

```
chown john filename
```

and the group allowed to access the file can be changed using the command

```
chgrp john filename
```

assuming that, as in the case of most personal computer systems, there is only one user `john` and therefore only one relevant group `john` to which that single user belongs. In Linux systems containing only one human user there will typically be additional virtual users such as `staff` and `mail` as well as, of course, `root`.

To give write permission on a file to those having group access, enter the command

```
chmod g+w filename
```

and to give read or execute permission substitute the letters `r` and `x`, respectively for `w`. To give a permission to everyone, change the letter `g` in the above command to an `o`. To withdraw permission, change the symbol `+` to the symbol `-`.

7 Getting Connected to the Internet

If you are thinking about setting up a Linux distribution for yourself, the most important hurdle will be setting up a dial-in connection to access the Internet. The connection will use the Point-to-Point Protocol, otherwise known as PPP. If your computer is hardwired directly to a network, you should have been using Tom's distribution thus far if it automatically detects your ethernet card and establishes a connection. In Debian Linux you will be prompted to establish this connection in the process of setting up the system. If you have to work through a modem, setting up an Internet connection is less transparent and can be quite difficult if you are dealing with a university network whose service may be free and not-for-profit, with the result that university personnel have no incentive to support anything but MS-Windows operating systems.

It turns out that an Internet connection can be established rather easily in BasicLinux. After the system loads, execute the command

```
/<#>pppsetup
```

and you will be presented with a screen giving a number of options. Choose '`2. Edit your PPP settings`'. You will be prompted to modify a number of files. The first one will appear on the screen as follows:

```
#!/bin/sh
#
# substitute your ISP's telephone number for 087304484
#
# if your ISP uses PAP/CHAP then remove the last two lines;
#  otherwise, substitute your username/password for ichi/xxx

exec chat                               \
      TIMEOUT      3                    \
      ABORT        '\nBUSY\r'           \
      ABORT        '\nNO ANSWER\r'      \
      ''           \rAT                 \
      TIMEOUT      30                   \
      OK           ATS11=55             \
      OK           ATDT087304484        \
      CONNECT      ''                   \
      ogin         ichi                 \
      word         xxx
```

If your Internet Service Provide uses either the PAP or the CHAP protocol your task is dead-easy. Simply do as instructed and press the `Ctrl` and `x` keys simultaneously to save your changes and exit the screen. If your ISP requires a text dialogue, as do the University of Toronto Systems, things can get quite complicated. To proceed, you must be using the version of BasicLinux obtained from my website.

On the above screen you will notice the line `exec chat`. This tells us that the shell is being asked to execute the program chat, which will in turn execute the *chat script* represented by the lines that follow. Through this, your computer is going to have a 'chat' with the computer that is being dialled into! The chat program reads in the first three lines and stores them in memory for future use. Henceforth, if the other computer is busy or gives no answer, it will abort the session. Chat then sends the modem a hard return `\r` and then, on the same line, asks for it's attention with the command `AT`. The `TIMEOUT` command tells the chat program to abort the session if nothing happens for 30 seconds. After the Modem answers `OK`, chat sends it a setup string. Then after the modem answers `OK` on the next line chat sends to the modem the telephone number that is to be dialled. The letters `DT` tell the modem to use tone dialling. You will have to change the telephone number to correspond with the one used by your ISP. After a connection has occurred, chat waits until the other computer sends a word

ending in the four letters `ogin`. This is consistent with a login prompt where the word login begins with either an upper case or lower case letter. When the chat program receives that prompt it relays back to the other computer your username which, of course, you will have replaced the word ichi with. Then chat waits to receive from the other computer a word ending with the four letters `word`. It then passes to the other computer your password, which you must replace the `xxx` in the script with. Within 30 seconds the PPP connection will be made. It turns out that, as noted at the top of the script, you can delete the login and password lines if your ISP uses PAP or CHAP. This information will be sent to the ISP in subsequent files.

Unfortunately, some ISP's (like the University of Toronto Systems) require not only these two lines, but a more complicated dialog between the two computers. If you are a University of Toronto student (or faculty member) and have a CHASS (Computing for the Humanities and Social Sciences) account you have to edit the above script as follows:

```
#!/bin/sh
#
# substitute your ISP's telephone number for 087304484
#
# if your ISP uses PAP/CHAP then remove the last two lines;
#  otherwise, substitute your username/password for ichi/xxx

exec chat                               \
      TIMEOUT      3                    \
      ABORT        '\nBUSY\r'           \
      ABORT        '\nNO ANSWER\r'      \
      ''           \rAT                 \
      TIMEOUT      30                   \
      OK           ATS11=55             \
      OK           ATDT4169463393       \
      CONNECT      ''                   \
      ogin         ppp                  \
      serid        ichi                 \
      word         xxx PPP
```

where you again replace the word `ichi` with your user ID and `xxx` with your password.

If you are a student or faculty member at the University of Toronto with a UTORID account (obtained through the Information Commons),

the required script is even more complicated. You should edit the file as follows:

```
#!/bin/sh
#
# substitute your ISP's telephone number for 087304484
#
# if your ISP uses PAP/CHAP then remove the last two lines;
#  otherwise, substitute your username/password for ichi/xxx

exec chat                               \
      TIMEOUT      3                    \
      ABORT        '\nBUSY\r'           \
      ABORT        '\nNO ANSWER\r'      \
      ''           \rAT                 \
      TIMEOUT      30                   \
      OK           ATS11=55             \
      OK           ATDT4169780449       \
      CONNECT      ''                   \
      name         01394417             \
      word         xxx                  \
      dialin       'ppp default'
```

Here, your username, represented above by `01394417`, must be replaced by appropriate digits from your library card number and `xxx` must again be replaced by your password.

When you are finished editing this file and save it by pressing `Ctrl` key and the `x` key simultaneously, and follow the resulting instructions, the file will be saved as `/etc/ppp/ppp-on-dialer` and you will then be presented with a second file to edit. This file is

```
# edit /dev/ttyS2 (if necessary) to match your modem's comport
# note:  com1= /dev/ttyS0   com2= /dev/ttyS1   com3= /dev/ttyS2
#
# edit 'user' line only if your ISP uses PAP/CHAP authentication

/dev/ttyS2
57600
lock
crtscts
modem
0.0.0.0:
defaultroute
# user [your_username]
connect /etc/ppp/ppp-on-dialer
```

If your ISP supports PAP or CHAP you must replace the word in rectangular brackets, and the brackets as well, with your username and remove the # from the beginning of the line. (That character, when appearing in a program rather than in a command prompt, tells the program not to read the rest of the line that follows it.) Otherwise, you just have to change the last character in the line `/dev/ttys2` to correspond to the Com (or communications) port to which your modem is attached. In Linux the Com ports are called `ttyS0`, `ttyS1`, and so forth, rather than `Com1` and `Com2`, etc. When you save and exit this file in the usual fashion the program will save it as the file `/etc/ppp/options`.

Next you will be presented with a file that you will need to edit only if your ISP uses either PAP or CHAP authentication.

```
# edit this file only if your ISP uses PAP/CHAP authentication
# name in 1st field, asterisk in 2nd field, password in 3rd field

[your_username]    *       [your_password]
```

If you need to edit this file you should insert your username and password appropriately and remove the brackets. Upon saving/exiting this file, (it will be saved to `/etc/ppp/pap-secrets`) you will then be presented with the file

```
nameserver 203.29.160.4
nameserver 203.29.160.2
```

Replace the numbers in this file with the ones given you by your ISP and save the file in the usual fashion. The program will save this file as `/etc/resolv.conf`.

After you have finished editing these files the same menu that appeared on your screen when you started pppsetup will reappear. Choose the option '`3. Save your PPP settings to floppy disk.`' Then choose the option '`4. Exit.`'

To connect to the Internet, enter the command

```
/<#>ppp-on
```

and to end the connection, enter the command

```
/<#>ppp-off
```

You will find the shell scripts `ppp-on` and `ppp-off` in the `/usr/bin` directory. As a matter of interest, have a look at them using the text editor (but don't change anything!).

8 Telnet: Reading Your Mail

When you are connected to the Web and have the appropriate privileges from your ISP you can use the `telnet` command log on to your ISP and work as an ordinary user on that machine. If the ISP has the program `pine` installed you can use it to check your mail. Suppose, for example, that you have a CHASS account. Log into CHASS using the command

```
/# telnet chass.utoronto.ca
```

and reply to the prompts with your username and password. To read your mail, enter at the `%` command prompt the word `pine`:

```
/homes/username % pine
```

It will be obvious how to proceed from the options on the screen and the help menu along the bottom of it.

If you are a University of Toronto student or faculty member and have a UTORDIAL account you can read mail you receive at that address by executing the command

```
/# telnet log.agent.utoronto.ca
```

and responding to the login prompt with the relevant digits of your library card and typing your password when asked. Then select the option '`1. UTORmail via Pine`' by pressing the '1' key but do not press `ENTER`. You will be asked to enter your password a second time. After that you use pine in the same way as in the CHASS account.

9 Using Linux fdisk

To find out what partitions exist on a computer and to modify those partitions you need to use the Linux fdisk program. To start fdisk enter the command (I again ignore the command prompt)

```
fdisk /dev/hda
```

where `hda` refers to the first hard disk drive on the computer. To work on the second disk drive, change this to `hdb`. You will receive the reply

```
Command (m for help):
```

and after typing `m` your screen will appear as follows:

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

To find out what the existing partitions are on the hard disk, press `p`. Your screen will appear as follows.

```
Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 789 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1       382   3068383+   b  Win95 FAT32
/dev/hda2           383       550   1349460   83  Linux
/dev/hda3           551       567    136552+  82  Linux swap
/dev/hda4           568       789   1783215   83  Linux

Command (m for help):
```

You should not run this program on anyone else's computer and you should use use it only to determine the layout of the drives on your own computer unless you have backed up all your files! To exit fdisk without doing damage, enter the command `q`. The discussion below assumes that you want to modify the partitions or create new ones on a computer on which you plan to install a 'real' version of Linux.

Before proceeding further, you should copy down on a piece of paper the above information not only for the disk you are planning to restructure but for all disks on the computer. You can find the total size of the disk in bytes by multiplying 16065 by 512, in the above case, to obtain the number of bytes per unit (or cylinder), and then multiply that number by the number of cylinders. To find the size of any partition, multiply the number of cylinders in the partition by the number of bytes per cylinder calculated above.

To delete a partition press `d` and then, when prompted, the number of the partition, where `/dev/hda1` is the first partition and `/dev/hda2` is the second partition, and so forth. Deletion of any partition marked 'extended' in the right-most column of the partition table will automatically delete any partitions associated with the blocks that are in that extended partition. If you are going to set up Linux on the disk you should delete all partitions unless you are doing the initial install from a DOS partition. In that event you should leave as is an appropriate DOS partition that you or someone else should have previously set up using DOS FDISK. Indeed, you can even have on that partition MS-Windows 95/98 which you can install Linux from

and then keep and use on occasion for browsing the Web, etc.[4]

To add a partition press `n`. You will be asked whether you want to add a primary or an extended partition. Press `p` to create a primary partition. You will then be asked the number to give the partition. Start with `1`, assuming that your previously created DOS partition is not partition 1. You will be asked to specify the first cylinder of the new partition—you can usually use the default here and press `ENTER`. You will then be asked to specify the last cylinder or the size in either cylinders, kilobytes, or megabytes. To pick the last cylinder, just enter the number of the cylinder. Alternatively, to set the size in cylinders and let the program find the last cylinder, enter `+nnn` where `nnn` is the number of cylinders. To set the size in megabytes, enter `+nnnM` where `nnn` is the number of megabytes.

Set the partition on which the root system will be mounted first. If you are going to use my Debian 2.2 setup you should set its size at 300 megs, more if you have lots of disk space. If your are installing Debian 3.1 from `http://www.debian.org` over the web, this partition should be set at 500 megs or more. Next you will need to create a swap partition. For Debian 2.2, I would recommend giving this partition at least 32 megs of space, or 64 megs if you have room, and certainly no less than 16 megs. Set it at 128 megs for a Debian 3.1 setup. Make both it and the next (and final) partition, which you will use for your research files, primary partitions. Then create the third partition, using the remainder of the space on the disk, by selecting the defaults at the prompts. Next you want to change the ID of the second, middle, partition to specify that it is to be a swap partition. To do this press `t`, choose partition 2 at the prompt, and enter the code 82. You should now have three partitions—a Linux swap (82) partition with Linux (83) partitions on each side of it. If everything is the way you want it, press `w` to write the revised partition table to the disk. If you have screwed something up you can exit `fdisk` without any changes taking effect by pressing `q`.

You should use the partition `/dev/hda3` (assuming that you are setting up Linux on the first hard disk on your machine) to house the files you will need to set up your Linux system. Before you can copy files to it, however, you must format it to house a Linux file system. You must first reboot the

[4]Linux can also share a computer with Windows XP. In this case, however, you will have to set Linux up from a Linux partition. If Windows is initially occupying the entire hard disk, you can downsize the Windows partition using the program fips, which you can obtain from my ftp site. Download `fips20.zip` and `fips.txt` from the place where you obtained FreeDos. The latter file is a manual giving intructions on how to use the fips program.

computer for the partition changes you made to take effect. Simply turn the computer off and then back on if you are using BasicLinux or Tom's Diskette. If you happen to be operating on a real Linux system installed on the computer, and are modifying partitions on that computer other than the one containing your operating system, press the Ctrl., Alt. and Del. keys simultaneously. When the computer has rebooted you can format partitions by following the instructions set out below.

10 Formatting a Partition

In BasicLinux, use the following command to place a Linux file system a partition /dev/hda3 and determine if the partition has any bad sectors.

```
mke2fs -t /dev/hda3
```

In Debian Linux 2.2 and 3.1 the equivalent command would be

```
mkfs -t ext2 /dev/hda3
```

The BasicLinux command `mke2fs` automatically makes a Linux file system while it is necessary to tell `mkfs` in Debian that we want a Linux (i.e., ext2) file system.

In BasicLinux you can make and mount a second 4 meg ramdisk (again, solely in the computer's memory) using the commands

```
mke2fs /dev/ram2 4096
mount /dev/ram2 /mnt
```

This will give you practice without damaging the file systems on the computer's actual hard disk(s).

11 Mounting Partitions

You can use either BasicLinux or Tom's Linux to read from and write to the hard disks on any computer you boot with them. This can be useful when you need to make repairs to get the computer to boot on its own. It is also useful for setting up a computer on which you wish to install Linux.

To mount a new directory called, say, `newdir` on the partition `/dev/hdb3`, we need to first create the directory with the command (henceforth I will ignore the command prompt)

```
mkdir newdir
```

and then mount it with the command

```
mount -t ext2 /dev/hda3 /newdir
```

The command line parameters `-t ext2` tell the computer that the partition is formatted for Linux. In fact, if the partition `/dev/hda3` is an `ext2` Linux partition with files on it you should be able to mount any empty directory on the partition using the command

```
mount /dev/hda3 /dirname
```

and the operating system will automatically detect that the file system on the partition is `ext2`. The files on the newly mounted partition can be accessed with the standard Linux shell commands. Simply `cd` to `/dirname` to access the files on that partition.

To mount a MS-Windows 95/98 partition, create a directory called `win` using the command

```
mkdir win
```

and then mount it with the command

```
mount -t vfat /dev/hdb1 /win
```

where `/dev/hdb1` is the partition you are mounting the directory `win` on. You can then use the standard Linux shell commands to access the files on that partition.

If you want to mount a standard DOS partition (pre-Windows 98) you would use the word `msdos` instead of `vfat` in the above command.

Being able to mount DOS-Windows partitions will be essential if you wish to install the base of a Linux system from a DOS partition to avoid having to use floppy disks in the installation process.

To mount the directory `/cdrom` on the CD-ROM drive of a computer with two hard disks and the Linux operating system Debian 2.2 or 3.1, execute the command

```
mount /dev/hdc /cdrom
```

You can determine what partitions have file systems mounted on them by executing the command `df`.

12 Retrieving Debian Linux Base System Disks

If you are going to install Debian Linux 3.1 you might want to use either BasicLinux or Tom's Diskette to move the necessary files from the Internet to the computer on which you want to make the installation. For Debian 3.1 that computer should have at least 24 megs of ram and 1 gigabyte of hard disk space. For a Debian 2.2 installation the target computer should have at least 16 megs of ram and 500 megs of hard disk space. To run a purely text-based system, with no graphic user interface, you would need only 12 megs of ram for either Debian 2.2 or 3.1. If you are going to use graphics, I would not recommend Debian 3.1 for computers older than a Pentium II—you should use Debian 2.2 in that case.

12.1 Using BasicLinux

After running pppsetup in BasicLinux, mount `mnt` (or any other empty directory) on the appropriate Linux or DOS partition on the hard drive of the computer you are going to set up Linux on. You should have already created the appropriate partition with DOS FDISK or with the Linux fdisk program. Then make that directory the working directory. To retrieve files to that directory with the web browser, load the Links browser by entering `links` on the command line. Press the key `G` (meaning 'go to') to obtain a 'request-for-URL'. In the text box enter the URL

`ftp://islamey.economics.utoronto.ca/pub/jfloyd/linux/debian30/`

if you are going to set up Debian 3.1. Actually, you will be setting up the Debian 3.0 base and then upgrading to Debian 3.1 over the Internet. Move the cursor to `basdebs.tar` and press `D` to download that file to your working directory. You can now take a snooze for a few hours, as the size of this file is 28 megs! This is the kind of download you want to do overnight! After this download is complete move the cursor to the file `drivers.tgz`, press `D` again and relax again while this file downloaded—although it is only 5.4 megs in size, it will still take quite a while for the download to complete. Then select and download the two files `rescue.bin` and `root.bin`. These two files will download quickly as compared to the earlier ones. Now download the two manuals `install.en.txt` and `fdisk.tex`.

If you are going to install the Debian 3.0 base from a Linux partition, these are all the files you will need. If you are going to boot the installation from DOS, you will need to download three additional files, `linux.bin`, `install.bat` and `loadln.exe`.

Should you choose to set up the older and less resource hungry Debian 2.2 distribution, you can obtain all the files for the simple version I use on my computers from my ftp site using the same URL as above but leaving off the part `debian30/`. The required files have the same names as the Debian 3.0 files above with one exception. The exception is that the Debian 2.2 base file is called `base2_2.tgz` instead of `basedebs.tar`. If you are going to do the installation from a DOS partition, you can obtain all the required files in the self-extracting zip file `debbasuz.exe` You will also have to obtain from this same place all the additional files required for the distribution I use for my research. These are

```
basutils.tar.gz
textprgs.tar.gz
X11prgs.tar.gz
netsc476.tar.gz
extras.tar.gz
uslocbin.tar.gz
usloclib.tar.gz
```

The latter two files contain some useful shell scripts and programs that I use on my system together with a group of template files to use with various programs. You should also obtain these latter two files if you are going to set up Debian 3.1. If your system has no more than 16 megs of RAM, and you are therefore going to set up Debian 2.2, you might want to use an older, less resource hungry version of Netscape. In that case replace `netsc476.tar.gz` with the two files `oldnets.tar.gz` and `netsc304.tar.gz`. Of course, it will be possible to add a few programs later by downloading selected files from the Debian archives.

12.2 Using Tom's Diskette

If you are using Tom's Diskette you will not have a browser to work with. Nevertheless, Tom's has a program called `wget` which you can use to download files. Boot your system with Tom's Diskette, mount the directory on your hard disk to which you are going to download the files, and make that directory your working directory. Then, to obtain the files from my ftp site, simply enter the command

```
wget ftp://islamey.economics.utoronto.ca/pub/jfloyd/linux/debian30/filename
```

for each file you want to obtain for the Debian 3.0 base setup or

```
wget ftp://islamey.economics.utoronto.ca/pub/jfloyd/linux/filename
```

for each file required for the Debian 2.2 setup.

12.3 Further Considerations

After you have installed the base system and established an Internet connection, you will have to set up the rest of a Debian 3.1 distribution over the Internet from the Debian website using the `apt-get` program. Further details on how to do this are contained in Appendix B to this document. As noted above, to complete your setup of Debian 2.2, you will only need `tar.gz` files noted above, although you may want to add some additional programs later. For details, see Appendix C below.

If you are operating in a country where you pay by the minute for usage of your telephone line or ethernet connection, or if you have an ethernet card but no modem and Tom's Diskette will not recognize your card and get you connected to the Web, you will have a serious problem. The cost of downloading even the smaller Debian 2.2 files over a telephone line will be very substantial and the cost of setting up Debian 3.1 over the Internet will clearly be prohibitive. In the case of Debian 3.1 you have two options if Tom's Diskette fails to access the Web. You can go to a friend's place or an Internet cafe and download the base files to a computer there and burn them to a CD. Then you can copy those files, using BasicLinux or Tom's Diskette, from the CD to the directory from which you are installing the base system. Alternatively, you can pay a small amount (possibly as little as $5 U.S.) to order a single CD from a site recommended by Debian. Point your browser at

```
http://www.debian.org/CD/vendors/
```

to find an appropriate vendor in your country. If you are operating over a phone line for which you are paying by the minute of usage, this is really the only option open to you. In this case, you can set up the whole system from the CD. In the case where you want to set up Debian 2.2, your only option is that of obtaining the relevant files from my website using another computer and burning them to a CD. Then you would copy all files to the partition from which you are setting up Debian 2.2 and then proceed in the same way as if you had downloaded them directly to your computer from the Web. If your computer has no CD-ROM drive, it will probably be cheaper to upgrade it by installing one than to pay for a costly download of the whole system from the Web over a telephone line!

13 Writing Boot Disk Images to Floppy Disks

To set up either the Debian Linux 2.2 or 3.0 base system you will need 'rescue' and 'root' floppy disks to initiate the installation unless you are doing a diskless install from a DOS partition. To set up these disks, use the commands

```
dd if=rescue.bin of=/dev/fd0 bs=1024
dd if=root.bin of=/dev/fd0 bs=1024
```

in BasicLinux or Tom's distribution to write the images to fresh new disks in Drive A.

14 Appendix A: Using the vi Editor

To start vi to edit a particular file, enter the command

```
vi filename
```

The vi editor has two modes—command mode and insert mode. You get into command mode by pressing the `Esc` key until the machine beeps. In command mode you can use the following keystroke commands:

`l` — moves the cursor one character to the right

`h` — moves the cursor one character to the left

`j` — moves the cursor down one line

`k` — moves the cursor up one line

`x` — deletes the character under the cursor

`r` — replaces the character under the cursor by the next character typed

`dd` — deletes the line the cursor is on

`yy` — copies (i.e., yanks) into memory the line the cursor is on

`u` — undoes the last deletion

`p` — inserts (puts) the last copied line on the screen

`i` — switches to insert mode—anything subsequently typed is inserted starting at the cursor position

`a` — switches to insert mode—anything subsequently typed is inserted starting to the right of the character the cursor is on

`:` — shifts command focus to a command line along the bottom of the screen following which

`:w` — saves (i.e, writes the current file)

`:q` — quits the current session

`:wq` — saves and quits

`:q!` — quits without saving

To delete or copy N lines including the line the cursor is on and N - 1 lines below it enter `Ndd` or `Nyy`. To search for a phrase, enter the command

```
:/phrase
```

To copy a file into the current file at the cursor position, enter the command

```
:r filename
```

To insert the line numbers along the left margin, enter the command

```
:set nu
```

To copy, say, lines 20 to 35 to a temporary file, enter the command

```
:20,35 w tempfile
```

Use `w!` in the above command to overwrite an existing file with the same name.

15 Appendix B: Setting Up Debian 3.1 Linux

Before attempting an installation on any computer you should determine the type of monitor and video card installed on it, most particularly the horizontal and vertical refresh rates of the montor, and make sure that your ethernet card is compatible with Linux. You can do the latter by checking the compatibility requirements from the Debian site or by insisting that the company selling you the computer install an appropriate card. You should know the type of mouse you have and the port to which it is connected. If you are using a modem, you should know what port it is connected to

and make sure that it is not a so-called win-modem that only works under MS-Windows. If you have a sound card you should also make note of the driver for it. Finally, you should have a printer that can use a driver listed in the group below:

```
x11 x11alpha x11cmyk x11mono x11cmyk2 x11cmyk4 x11cmyk8
x11rg16x x11rg32x x11gray2 x11gray4 bbox bff bit bitcmyk
bitrgb bmp16 bmp16m bmp256 bmp32b bmpa16 bmpa16m bmpa256
bmpa32b bmpamono bmpasep1 bmpasep8 bmpgray bmpmono bmpsep1
bmpsep8 ccr cfax cgm24 cgm8 cgmmono cif dfaxhigh dfaxlow
epswrite faxg3 faxg32d faxg4 inferno jpeg jpeggray mag16
mag256 mgr4 mgr8 mgrgray2 mgrgray4 mgrgray8 mgrmono miff24
pbm pbmraw pcx16 pcx24b pcx256 pcx2up pcxcmyk pcxgray
pcxmono pdfwrite pgm pgmraw pgnm pgnmraw pkm pkmraw pksm
pksmraw plan9bm png16 png16m png256 pnggray pngmono pnm
pnmraw ppm ppmraw psgray psmono psrgb pswrite sgirgb
sunhmono sxlcrt tiff12nc tiff24nc tiffcrle tiffg3 tiffg32d
tiffg4 tifflzw tiffpack alc2000 alc4000 alc8500 alc8600
ap3250 appledmp atx23 atx24 atx38 bj10e bj10v bj10vh bj200
bjc600 bjc800 bjc880j bjccmyk bjccolor bjcgray bjcmono
cdeskjet cdj1600 cdj500 cdj550 cdj670 cdj850 cdj880 cdj890
cdj970 cdjcolor cdjmono chp2200 cljet5 cljet5c cljet5pr
coslw2p coslwxl cp50 declj250 deskjet dj505j djet500
djet500c djet820c dl2100 dmprt dnj650c epl2050 epl2050p
epl2120 epl5800 epl5900 epl6100 eps9high eps9mid epson
epsonc escp escpage escpc fmlbp fmpr fs600 gdi hl1240
hl1250 hl7x0 hpdj1120c hpdj310 hpdj320 hpdj340 hpdj400
hpdj500 hpdj500c hpdj510 hpdj520 hpdj540 hpdj550c hpdj560c
hpdj600 hpdj660c hpdj670c hpdj680c hpdj690c hpdj850c
hpdj855c hpdj870c hpdj890c hpdjplus hpdjportable ibmpro
imagen iwhi iwlo iwlq jetp3852 jj100 la50 la70 la75 la75plus
laserjet lbp310 lbp320 lbp8 lex2050 lex3200 lex5700 lex7000
lips2p lips3 lips4 lips4v lj250 lj3100sw lj4dith lj4dithp
lj5gray lj5mono ljet2p ljet3 ljet3d ljet4 ljet4d ljet4pjl
ljetplus ln03 lp1800 lp1900 lp2000 lp2200 lp2563 lp3000c
lp7700 lp8000 lp8000c lp8100 lp8200c lp8300c lp8300f
lp8400f lp8500c lp8600 lp8600f lp8700 lp8800c lp8900
lp9300 lp9400 lp9600 lp9600s lq850 lx5000 lxm3200
lxm5700m m8510 md1xMono md2k md50Eco md50Mono md5k
mj500c mj6000c mj700v2c mj8000c ml600 necp6 npdl
```

```
oce9050 oki182 oki4w okiibm paintjet pcl3 photoex
picty180 pj pjetxl pjxl pjxl300 pr1000 pr1000_4 pr150
pr201 pxlcolor pxlmono r4081 rpdl samsunggdi sj48 st800
stcolor t4693d2 t4693d4 t4693d8 tek4696 uniprint xes
cups ijs omni stp nullpage
```

After you have downloaded the base files to a Linux partition on the computer on which you plan to install Debian 3.1 and created the rescue and the root floppy disks, insert the rescue disk in Drive A and turn on the computer. Alternatively, if you are booting the installation from DOS, execute the command `install` from the DOS directory to which you copied the base files files. In either case, simply follow the instructions on the screen, choosing the defaults unless it is clear that you need to do something else. Use the Arrow and Tab keys to move among menu items. After indicating the partition from which you are installing the base files, let the installation program 'Choose from a list of all likely directories'. When configuring device-driver modules, choose 'ide-floppy' in the 'block' devices section, and under 'net' select the correct driver for your ethernet card. If your computer has a sound card and speaker, go to the 'misc' devices and install the driver for that card. These are the only drivers you should need to install. Configure the network using the option DHCP if you are connected through Rogers—check with your systems administrator if you are operating from a university backbone and with the cable company if you are using one other than Rogers. Then proceed to install the base system. As the installation proceeds you should definitely 'Make a Boot Floppy' when asked and if you are not also using MS-Windows you should also 'Make Linux Bootable Directly from the Hard Disk'. You should definitly install 'shadow passwords' when asked, remove the PCMCIA packages (assuming you are installing the system on a desktop), and choose not to configure PPP at this time. You will be asked to select the methods by which the program `apt-get` will retrieve software packages for installation. Choose 'http' and at the subsequent prompt pick the mirror site closest to your location. Include all types of packages—non-US, non-free, and contrib—when prompted. If the installation program has trouble connecting to a particular site and starts timing out, press the keys `Ctrl+C` (the keys `Ctrl` and `C` simultaniously) to abort the choice and then pick an alternative site. When prompted, choose a second site, following the same procedure as in the case of the first.[5] Answer 'Yes' to the inquiry as to whether you want

[5]Note that if you choose the 'http' method you may nevertheless obtain your packages by http from 'ftp' sites that are included among the indicated mirror sites.

to install security updates. Choose not to run `tasksel` and also choose not to run `deselect`. Then you can press `Y` to proceed with the updating of packages. When asked to configure your mail system, select option (5). If you cannot do your mail using `pine` through an account with your University (or ISP) you can do it using Mozilla, which you will set up in due course. You will eventually be asked to set a root password and set yourself up as a user. When prompted about the key-map, select the option 'don't touch keymap' to simply keep the one set up when you installed the system. Also, choose not to have 'man' and 'manb' installed setuid man. To save space, choose to erase any previously downloaded deb files.

You will then have a functioning Debian system on which you can use either `vi` or `nano` to edit files. The latter editor is easy to use because the functions performed by various key combinations are shown along the bottom of the screen. If you have an ethernet card on your machine you will already be connected to the Internet. Otherwise, you will have to set up your modem to access the Internet over the telephone line. To do this, follow the instructions in Appendix D.

You should now upgrade that your Debian 3.0 system to Debian 3.1 over the Net by logging on as root and, using the root directory `/` as the working directory, entering the command

```
apt-get upgrade
```

At the prompts, simply select the defaults.

Before installing the packages for the programs you will need for doing your research you should, as originally planned, move your `\home` directory to the partition on which you copied your base files for installation. If you have installed from a DOS partition you should have left a separate partition at the end of your hard drive to house your personal and research files. Rename the directory `/home` to `/home.bak` using the `mv` (move) command and then make a new directory called `/home` and mount it on the partition `/dev/hda3` (or `/dev/hda4` if you have kept a DOS partition). The relevant commands are

```
mv /home /home.bak
mkdir /home
mount /dev/hda3 /home
```

Then copy all data from your old home directory `/home.bak` to the new directory `/home`. There should only be a few hidden files and/or directories to copy. To identify these use the command (from within that directory)

```
ls -a *
```

Finally, you need to ensure that your new home partition is correctly mounted when the computer is rebooted. To do this you need to edit the file `/etc/fstab` (after making a backup copy!), adding to it the following lines (the lines beginning with `#`, which instructs the computer not to read them, are comments you might put in to remind yourself later of why and how you set up the uncommented line).

```
# Addition to /etc/fstab
# Mount /dev/hda3 as /home into the file system
/dev/hda3      /home     ext2       defaults    0   2
```

Now you are ready to install the packages for your system. Log in as root. You can do this in either of two ways. First, you can simply type `root` at the login prompt and enter your root password when asked. Alternatively, if you are logged in as yourself you can become root by entering the letters `su` at the command prompt and then following with your root password when asked. You can also run two or three separate sessions at one time either as root or as yourself. After you have logged in, you can press `Alt+F2` to activate a second terminal to which you can log in. A third terminal can be activated by pressing `Alt+F3`. To move back and forth between the three terminals, simply press `Alt+Fx` where `x` is either `1`, `2`, or `3` as desired.

Install your packages over the Internet using the program `apt-get`. Additional packages required by any package you select will be installed automatically. To set yourself up for printing documents excecute the following four commands to install the programs `lpr`, `gs`, `magicfilter`, and `a2ps`:

```
apt-get install lpr
apt-get install gs
apt-get install magicfilter
apt-get install a2ps
```

The program `lpr` is the line-printer daemon that lurks in the background and sends files to the printer when the appropriate command is issued. Ghostscript—denoted by `gs`—converts postscript and other files to a form that your printer can print. The `magicfilter` program translates material sent to the printer using the command line instruction

```
lpr filename
```

into language that your printer understands and `a2ps` (anything to postscript) converts ordinary text files to postscript form for saving to file or automatic translation into machine language and printing using the program

`magicfilter`. You will be asked to configure `magicfilter` when you set it up. If your configuration turns out not to work you can redo it with the command

```
magicfilterconfig --force
```

Some instructions on how to use `a2ps` are included in Appendix E.

Now proceed to install the other text-based programs you will need by substituting the program names in the above `apt-get` commands with the ones below. Install the text browser `lynx` (you could alternatively choose `links` which is the browser used in BasicLinux), the editor `joe`, which can handle easily more sophisiticated tasks than either `vi` or `nano`, the program `mtools` that enables you to read files from and write them to floppy disks in much the same manner as you can do with DOS, and the two programs `ispell` and `ibritish` for spell checking documents. Instructions on how to use `joe` and `mtools` are included in Appendix E. Next, you will need the venerable TEX/LATEX typesetting program, `tetex-bin`, and a companion program `dvipdfm` which converts files generated by TEX and LATEX into PDF files. The program `octave` is a useful MatLab-like program for manipulating matrices and doing matrix calculations. You will also need a statistical program. For simple tasks and for learning basic statistics `xlispstat` will prove very useful. I have prepared *A Short Manual for Xlispstat*, `minmanls.ps` (`minmanls.pdf`), which can be obtained from my ftp site, where you previously obtained all the files referred to above. If you are going to do heavy-duty statistical work you should obtain Ox. The basic Debian package file for this program, `ox3.30-2.i586.deb`, can be downloaded from my ftp site and installed using `dpkg` with the command

```
dpkg --install ox3.30-2.i586.deb
```

Once you have installed the program, you will find a manual `oxintro.pdf` in the directory `/usr/local/lib/ox-3.30/doc/` and, in the same directory, a help link that can be accessed with either the lynx or links browser using one of the commands

```
lynx -nocolor /usr/local/lib/ox-3.30/doc/oxfnmenu.html
links /usr/local/lib/ox-3.30/doc/oxfnmenu.html
```

To avoid insulting your University or your ISP by logging onto it with `telnet`, a program that allows your password to pass over the lines unencrypted, you should set up the secure shell program `ssh` using the standard `apt-get` command.

```
apt-get install ssh
```

Next you need to install and set up your graphic user interface—commonly known as the X-Window System or, simply, X. To do this, execute the commands:

```
apt-get install icewm
apt-get install x-window-system
```

The first command installs the Ice Window Manager, which is much easier to use than the default one that will automatically be installed. (The window manager takes commands from the mouse and the keyboard and draws the appropriate windows on the screen.) Once you get your system installed you can choose from a wide variety of different window managers. The second command retrieves and installs the X-Window system which, since `icewm` is already installed, will use the Ice Window Manager as the default. As the X-Window system is being set up you will be asked about your mouse and about some of the features of your monitor. Choose the 'simple' method of configuration if possible. Use the default suggestions if they do not seem inappropriate. If you happen to make some wrong choices you could end up with a mess on the screen, or a complete failure of the graphic installation. If this happens you should reboot the system and reconfigure it using the command

```
dpkg-reconfigure xserver-xfree86
```

This will install a new `XF86Config-4` file in the directory `/etc/X11`. You can also edit that file using your text editor (after backing it up under a new name). Running the command

```
XFree86 -configure
```

will create a version of this file called `XF86Config.new` in the directory `/root` which will contain what the configuration program thinks the specifications of your mouse, video card, monitor, etc. are. This information might be useful in replying to the questions asked when you run the `dpkg-reconfigure xserver-xfree86` command.

To reboot the computer when you have a graphics screen in place, press `Alt+Ctrl+F1` to access a text terminal and reboot from there by pressing `Alt+Ctrl+Del`. To return to the graphics screen after switching to a text screen, press `Alt+Ctrl+F7`.

When the X-Window system is properly set up, a log-in screen with a small text terminal in the bottom right corner should appear on boot-up.

Simply type your username, press `Enter`, and then enter your password and the basic Ice Window Manager screen will appear. Click on the icon in the bottom left corner or press `Ctrl+Esc` and a menu of the available programs and options will appear. Click on the computer-terminal icon, or use the arrow keys and press `Enter`, to activate a terminal window in which you can enter commands just as you would on a text terminal. The little squares numbered 1 through 4 represent alternative 'workareas' to which you can move by clicking on the relevant square with the mouse. Upon boot-up you are initially in the first of these four 'work areas'.[6]

You can pretty-up the background color on your window-manager screen as follows. Go, as root, to `/etc/X11/icewm` and load the file `preferences` into the text editor (after backing up the file under a different name). Go to the line that contains the word `DesktopBackgroundColor` and change the code fragment `/00/20/40` to `/00/20/900`. Then remove the `#` character from the beginning of that line so that the program will read and process it rather than use the default. You can then change the toolbar along the bottom of the screen by pressing simultaneously the `Ctrl` and `Esc` keys to bring up a menu in the bottom left corner of the screen and then scrolling down to `Themes` and then choosing the theme `"nice"`. Next time you boot up the window manager you should be presented with a beautiful blue screen with a much nicer toolbar along the bottom and should find the text terminals in which you will be working much more pleasing to the eye. If, by chance the toolbar does not change, become root user again and go to `/etc/X11/icewm/` and load the file `toolbar` (after backing it up under a different name). Go to the line that starts with `Theme=` and change the code fragment `"Infadel2/default.theme"` to `"nice"`.

Also, you may find the log-in screen presented on boot-up to be unattractive, and you may wish on occasion to boot directly to a text terminal rather than have `xdm` (the X-Display-Manager) run automatically. To accomplish this, go to the directory `/etc/init.d` (which contains links to the programs that boot automatically) and (as root user) make the file `xdm` non-executable using the command

```
chmod -x xdm
```

[6]It is common among Linux users to use sophisticated X11-based graphical user interfaces such as KDE or GNOME, which are built on top of window managers and have many MS-Windows-like drag-and-drop features that minimize the need to use keystroke commands. I find that I can get things done faster by simply using the window manager alone—on those of my machines in which KDE and/or GNOME are installed, I find that I rarely use the extra features that these constructs provide.

Thereafter, when you boot the computer you will be presented with a text screen in which to log in. You then have the option of working in the text screen, either as yourself or as root user if some administrative tasks have to be performed, or loading X. To load the X-Window System, enter the command

```
startx
```

You can tweak your system to list directories in blue, executable files in yellow, links in turquoise, and zipped archives in red by adding a line to the `/etc/profile` file (after backing it up under a new name). Between the lines

```
export PATH PS1
```

and

```
umask 022
```

add the line

```
alias ls='ls --color=auto'
```

Thereafter, when you log in as yourself and execute the command `ls` directories and file names will appear in appropriate colors.

Now you can install the additional graphics programs you will need using the `apt-get` command as above. To view postscript and PDF files created by TEX and LATEX you will need the program `gv`. A useful alternative program for viewing PDF files is `xpdf`. And you will need `gnuplot`, a useful program for viewing and creating printable charts. My tiny manual for this program, `minmangp.ps` (or `minmangp.pdf`) can also be obtained from my ftp site. Also, you will need the program `xfig` for drawing graphs. All of these programs work off a command line in a text terminal—simply type the program name (`gv`, `xpdf`, etc.) followed by the name of the file on which the program is to operate.

Finally, you will need a good browser for surfing the Web. I recommend Mozilla, which you can set up using the commands

```
apt-get install mozilla-browser
apt-get install mozilla-psm
```

As things stand you can only access Mozilla by opening up a text window and typing the command `mozilla` in it. To add access to Mozilla to the toolbar along the bottom of the screen, edit the file (as root) `/etc/X11/icewm/toolbar` (after backing it up under a new name). Replace the line referring to Netscape with the line

```
prog MOZILLA mozilla mozilla
```

The next time you boot the computer you will observe a Mozilla button on the toolbar. Click on it to load the browser.

If you want to bring your mail to your home computer to read and manage instead of leaving it on your University or ISP's server to be accessed by webmail or the program 'pine' on that server, you will need to also obtain and install `mozilla-mailnews` using the `apt-get` command. You can configure this program easily to enable you to read, save, and send mail. Those who are paying by the minute for their Internet connection can then dial in, retrieve their mail, then hang up and reply to their mail, then dial up again and send their outgoing mail—money is not dribbling away while they puzzle about what to say when writing replies.

Install `gnumeric` to obtain a graphic spreadsheet program which which you can view (at least some) Lotus and Excel files. I find the venerable old UNIX spreadsheet program Sc much better than modern graphic spreadsheets for the work I do. You can obtain this by downloading the file `sc` from my web-site and copying it to the directory `/usr/bin`. As is, this version of `sc` will refuse to run on the grounds that it cannot find the library file `libncurses.so.4`. That library has been superceded by `libncurses.so.5.2` which remains compatible with older versions. So you will have to create a link called `libncurses.so.4` to the actual file `libncurses.so.5.2` in the directory `/lib` using the command

```
ln -s libncurses.so.5.2 libncurses.so.4
```

where the first filename is the actual file being linked to and the second file name is the name assigned to the file that is merely a link. You can download my little manual for this program, `minmansc.ps` (or `minmansc.pdf`), from my web-site. Alternatively, you can obtain a slightly more modern version of Sc in the usual way using `apt-get`, but I find the older program (which is the one covered in my manual) easier to use.

You might also want to have on your computer a more conventional word processor for writing memos and letters and other simple documents. For this purpose you should obtain the program Abiword using the commands

```
apt-get install abiword
apt-get install abiword-doc
```

When you fire up this program (by simply typing `abiword` on the command line) you might get a warning about the program's inability to install certain fonts. You can usually ignore this warning. To suppress it, load `abiword`

and click on 'Tools', then 'Preferences' and then 'Preference Schemes' and deactivate the option 'Show font warning at start-up'.

For viewing graphics files you will need to add the program Gqview. And if you have a sound card and want to listen to CD's while you work you can install the program Workman. Use the commands

```
apt-get install gqview
apt-get install workman
```

There are many programs you can add to your system should you choose. You can browse through the alternatives on the `debian` site. Point your browser at `http://www.debian.org` and click on the link 'Debian Packages'.

16 Appendix C: Setting Up Debian 2.2 Linux

Before proceeding you must know the name of the video card on your system and your monitor's horizontal and vertical refresh rates, whether your monitor supports SVGA, rather than just VGA, the name of the driver for your ethernet card (if you have one),[7] the type of mouse you will be using and the ports to which your mouse and modem (if you have one) are connected.[8] Your printer must connect to the parallel port on your computer and it will have to support one of the following drivers:

```
t4693d2 t4693d4 t4693d8 tek4696 appledmp ccr lp2563 lbp8 lips3
m8510 necp6 cp50 oce9050 oki182 okiibm r4081 sj48 xes ln03 la50
la70 la75 la75plus sxlcrt deskjet djet500 laserjet ljetplus
ljet2p ljet3 ljet4 declj250 paintjet pjetxl cdeskjet cdjcolor
cdjmono cdj550 cdj500 djet500c dnj650c pj pjxl pjxl300 hpdj
uniprint epson eps9mid eps9high epsonc lq850 lp8000 st800
stcolor ap3250 ibmpro bj10e bj200 bjc600 bjc800 ljet3d lj4dith
lj5mono lj5gray lj250
```

[7] If you are purchasing an Ethernet card, make sure you obtain one for which a driver is available in Debian. A list of suitable cards and their drivers can be found in Appendix D of Bill McCarty's book, *Learning Debian Gnu/Linux*, O'Reilly and Associates, Inc., 1999. You can read (and buy) this book on the Web. Point the browser at `http://www.oreilly.com/catalog/debian/chapter/book/appd_06.html`
and click on the section entitled 'Using Loadable Ethernet Drivers'.

[8] In selecting a modem for your computer you should choose a fast one (this is no place to scrimp) and an external one that can subsequently be used on another computer should you choose later to upgrade. Make sure you don't get stuck with one designed only for MS-Windows, a so-called win-modem!

Start by setting up the base system. In what follows I will assume that a Linux swap partition `/dev/hda2` has been created, that the files `base2_2.tgz` and `drivers.tgz` are in `/dev/hda3` and that Debian 2.2 is to be installed on `/dev/hda1`. The procedure for an install from a DOS partition is the same as outlined below after the system is booted from the DOS partition by entering the command `install`. In that case the DOS partition would be `/dev/hda1`, the swap partition `/dev/hda3`, the partition containing the installation files `/dev/hda4`, and you will be setting up Debian 2.2 in `/dev/hda2`.

Boot the machine from a cold start with the Debian 2.2 rescue disk. Remove that disk and insert the Debian 2.2 root disk when prompted. Press `ENTER` after the first screen and then again on the subsequent screen to configure the keyboard. Select 'Qwerty U.S.', provided of course that you are in an English-speaking country.

Press `ENTER` again to configure and initialise a swap partition. You will be prompted with the name of partition `/dev/hda2` which you should already have given type 82 (linux swap) designation. Select that partition, pressing `ENTER` as required and then choose to either scan or not scan the partition on the next screen. If you are unfamiliar with the hard disk on your computer you should probably choose the scan option. Next, initialise a linux partition, choosing `/dev/hda1` as planned when you set up the partitions. Press `ENTER` when prompted to retain Linux kernel 2.0 (an earlier version) compatibility. Then choose to scan or not scan the partition and press `ENTER` when asked to confirm your choice. Then press `ENTER` at the prompt to mount `/dev/hda1` as the root filesystem.

Now proceed to install operating system kernel and modules. You will be asked to select the medium you will use to install the system. Use the down-arrow-key to move the cursor to the 'harddisk' entry and press `ENTER`. You will then be asked to select the partition where your Debian archive resides. Choose `/dev/hda3` as planned.[9] When you are asked to choose the directory on which your installation files reside select 'Choose from a list of all likely directories'. You will be given the directory `/instmnt` which you should select by pressing `ENTER`.

Now proceed to configure device driver modules. At the prompt, skip the step of loading device drivers from a floppy disk. (assuming that you are using standard equipment). Next you will be asked to select the modules (loadable device drivers) you want supported by the kernel. Start by selecting 'block'. On the screen that pops up when you select block devices,

[9] Or the DOS partition if you are installing the base using DOS.

select 'ide-floppy', press `ENTER` and at the prompt choose not to enter a command line (again, assuming that you are using standard equipment). When a black screen appears press `ENTER` to signify that you are ready to continue and then select the item 'Finished with these modules'.

As long as the CD-ROM on your machine is standard you will not need to make any selection with respect to the 'cdrom' option. Skip the 'fs' option. If you have an Ethernet card connected to your machine, scroll down and choose 'net' next. Use the down arrow to scroll down to the driver for your card (mine happens to be `rtl8139`) and press `ENTER`. Again, do not enter a command line.

Now exit the module selection screen entirely and select the 'Configure the Network' option. You are required to give your machine a host name. If you have an Ethernet card you might need to follow instructions of your network administrator concerning the host name you can select. You will then be asked if you want DHCP to configure the interface? Select yes if your network allows it. Otherwise you will have to input some numbers provided by your network administrator.

Now select 'Install the Base System'. Select the 'harddisk' option and then select 'Choose from a list of all likely directories' which will again call forward the suggestion `/instmnt`. Press `ENTER` to confirm that choice. Now proceed to 'Configure the Base System'. You will be asked to select a time zone. Choose your country and your time zone—for Toronto that will be Canada and either EST or EDT, depending upon the time of the year. In making your selections you can use the TAB key to switch between the two menus.

Next you are asked whether you want to 'Make Linux Bootable Directly from the Hard Disk' or to 'Make a Boot Floppy'. Choose the latter. Insert a fresh new blank floppy disk in the floppy disk drive and press `ENTER`. If you are using only Debian 2.2 Linux on your computer you can then choose to make Linux bootable directly from the hard disk by scrolling down the menu to the option 'Make Linux Bootable Directly from the Hard Disk' and selecting it. You should always make a boot floppy.

Now select 'Reboot the System' at the prompt. When the system reboots, press `ENTER` to not enable md5 passwords, and then `ENTER` again to install shadow passwords. Next you are asked to enter your root password and then enter it again to verify. And then you are asked to create an account for yourself, entering a username of your choice, your full name and your password (the latter twice to verify).

Finally, press `ENTER` to remove the PCMCIA packages. Also, you should choose not to configure PPP at this time. You will then be asked what

medium you will use to install packages. Use the TAB and ENTER keys to sellect 'cancel'. When you are asked about the installation method, choose 'Simple'. The installation will crap out at this point (You don't care because you are going to install your packages later). Choose 'n' to 'not continue' and then press ENTER when 'abort' appears on the black screen that comes up. Press ENTER to bring up the login prompt and you are ready to log into your new system.

At this point you have a functioning Debian system. You will have two text editors, ae and vi, to choose from with which to edit files and make executable scripts. The ae editor is easiest to learn, since it has a help display that always appears across the top of the screen, but every person who interacts with UNIX systems, of which Linux is a representative, should know how to do a few basic things with vi—all UNIX systems have the vi editor installed. Appendix A of this document will provide you with the minimum you will need to know about vi.

You can now install the additional software that will make your system useful in your research. I recommend that you simply install the packages in the tar.gz files you should have already copied to /dev/hda3 from my ftp site.

The basic package installation tool is dpkg. After unzipping the tar.gz files, a simple but time consuming way to proceed is to install the packages one-by-one using dpkg. Give yourself root-user privileges by executing the command

```
su
```

and typing your root password at the resulting prompt. Then create a new directory off the root directory using the command

```
mkdir debs
```

and mount that directory on /dev/hda3 by executing from the root directory the command

```
mount /dev/hda3 /debs
```

Move to /debs and list the files there using the commands

```
cd debs
ls -l *
```

and you will see a listing of the zip-files you previously copied to /dev/hda3 (on which the directory /debs is now mounted) using BasicLinux or Tom's Root Boot distribution.

When installing packages from this directory using dpkg you have to install them in a particular order—dpkg won't install a package unless those packages it depends on are already installed. To obtain the packages containing basic utilities, unzip the file `basutils.tar.gz` to disgorge the appropriate package or 'deb' files using the command

```
tar xvzf baseutils.tar.gz
```

Then you could enter, one by one in order, the following commands to install these packages:

```
dpkg --install lpr_0.48-1.deb
dpkg --install groff_1.15.2-1.deb
dpkg --install man-db_2.3.16-1.1.deb
dpkg --install magicfilter_1.2-39.deb
dpkg --install libpaperg_1.0.3-13.deb
dpkg --install a2ps_4.13-2.deb
dpkg --install xfree86-common_3.3.6-11potato32.deb
dpkg --install xlib6g_3.3.6-11potato32.deb
dpkg --install mtools_3.9.6-4.deb
dpkg --install joe_2.8-15.2.deb
dpkg --install libncurses4_4.2-9.deb
dpkg --install sc_6.21-10.deb
dpkg --install zlib1g_1.1.3-5.deb
dpkg --install lynx_2.8.3-1.deb
dpkg --install freetype2_1.3.1-1.deb
dpkg --install libpng2_1.0.5-1.deb
dpkg --install svgalibg1_1.4.1-2.deb
dpkg --install vflib2_2.25.1-4.deb
dpkg --install gs_5.10-10.1.deb
dpkg --install gsfonts_5.10a-2.deb
dpkg --install ispell_3.1.20-12.deb
dpkg --install ibritish_3.1.20-12.deb
```

Somewhere along the line you will be asked to choose a default paper name. In North America that will be [letter].

A number of the above packages are simply library and supporting software for other programs. But several are worth noting as we proceed—this will make you aware of what is in fact installed on your computer. The `lpr` package sets up the line printer daemon, which listens for instructions from the shell and other programs to send files to the printer and then responds appropriately. The `man-db` package sets up a data-base framework

in which the manual pages of all other installed programs will be contained. In addition, it includes man-pages for programs already installed.

The package `magicfilter` sets up a filtering system in which all files sent to the printer are converted from their current form, be it text, postscript or pdf, into language the printer understands. You will be asked to configure this program when it is set up. You will need to know the driver for your printer as well as the port to which is connected. This will normally be the first parallel port, in Linux-speak called `/dev/lp0` (the second parallel port is called `/dev/lp1`). The magicfilter setup installs a file in the `/etc` directory called `printcap`. The printer daemon reads this file for instructions before sending files to the printer. Either a text, postscript or PDF file can then be printed by simply entering on the command line the word `lpr` followed by the file name. If you happen to screw up in setting up magicfilter you can have another go at it later by entering the command

```
/usr/sbin/magicfilterconfig --force
```

The package `a2ps` (anything to postscript) provides a neat program for converting text files into postscript files and printing them. The `mtools` package provides the ability to copy files to and from a floppy disk in Drive A: using commands similar to well-known DOS commands and in a form that can be stored on DOS formatted disks. A brief discussion of how to use the program can be found in the Appendix E. The package `joe` contains my favourite editor, which was written many years ago by Joe Allen. It is simple to learn and can do everything I need to do with an editor—again, see Appendix E.

The `sc` package contains the spreadsheet calculator program Sc, which is a simple text-based program sufficient for most research tasks unless you are planning to store large quantities of data in multi-paged spreadsheet files. I have written a mini-manual, `minmansc.ps`, which is available from my ftp site.

You can find additional information about all these programs by entering `man` followed by the name of the program on the command line.

The package `lynx` contains a web browser that does not require a graphics interface. It is much faster than Netscape for retrieving files from the Internet, especially when one knows ahead of time where those files are located. You can figure out how to use lynx from its manual page, which you can obtain by entering `man lynx` on the command line. Alternatively, to print out the manual page you can execute the two commands

```
man lynx > lynxman.ftxt
```

```
lpr lynxman.ftxt
```

The file `lynxman.ftxt` will contain formatting codes that make it difficult to read using a text editor.

The `svgalibg1` package, which is required because other packages depend on it, can be used to view postscript files from a text terminal (without using X) although it does a rather crude job. The `gs` package contains the Ghostscript program for printing postscript files—it can be used with `svgalib` to view postscript files. `ispell` is a text-based spell-checking program which can be used by simply typing on the command line the word `ispell` followed by the name of the file to be spell-checked. An appropriate dictionary for Canadians is contained in the package `ibritish`.

Returning to the procedure for installing these packages, I will now outline a much faster method than running dpkg on the packages one by one. You can simply copy the script `basutilss`, which should been disgorged when you unzipped `basutils.tar.gz` and therefore be present in the `/debs` directory, to `/usr/local/bin` and make it executable using the command

```
chmod +x basutilss
```

Then type from within the directory `/debs` the word `basutilss` on the command line and the packages listed above will be automatically installed one by one. The order in which the `dpkg` commands are listed in the script is such that the packages that other packages depend on get installed first.

To install the remaining programs sufficient for a text-based system, you can unzip the file `textprgs.tar.gz` and copy the file `txtprgsu` to `/usr/local/bin` and make it executable. Then type the word `txtprgsu` on the command line and the following commands will be executed automatically.

```
dpkg --install gnuplot_3.7.1p1-4.deb
dpkg --install xlispstat_3.52.14-1.deb
dpkg --install ed_0.2-18.1.deb
dpkg --install perl-5.005_5.005.03-7.1.deb
dpkg --install libnet-perl_1.0703-3.deb
dpkg --install dpkg-perl_0.1-3.0.deb
dpkg --install tetex-base_1.0-10.deb
dpkg --install tetex-lib_1.0.6-7.deb
dpkg --install tetex-bin_1.0.6-7.deb
```

The `gnuplot` package contains the plotting program Gnuplot which you will use to plot charts and incorporate them in your LaTeX documents. The

basic things you will need to know about this program is contained in my tiny manual `minmangp.ps` which can be obtained from my ftp site. XlispStat, an excellent free lisp-based statistical program that is relatively easy to learn, is contained in the package of the same name. Get my *A Short Manual for Xlispstat*, `minmanls.ps`, from my ftp site for the basics on how to use the program. The remaining commands install the TEX/LATEX document preparation system.

Next you have to install the X-Windows graphics system. Unzip the file `X11prgs.tar.gz`, copy the file `X11prgss` to `/usr/local/bin` and make it executable. Then type `X11prgss` on the command line and press `ENTER`. The following commands will be executed automatically.

```
dpkg --install libace5.0_5.0.7-4.deb
dpkg --install libjpeg62_6b-1.2.deb
dpkg --install libtiff3g_3.5.4-5.deb
dpkg --install t1lib1_1.0-2.deb
dpkg --install menu_2.1.5-3.deb
dpkg --install cpp_2.95.2-13.deb
dpkg --install xaw3dg_1.3-6.9potato1.deb
dpkg --install xbase-clients_3.3.6-11potato32.deb
dpkg --install xfonts-100dpi_3.3.6-2.deb
dpkg --install xfonts-75dpi_3.3.6-2.deb
dpkg --install xfonts-base_3.3.6-2.deb
dpkg --install xfonts-scalable_3.3.6-2.deb
dpkg --install xserver-common_3.3.6-11potato32.deb
dpkg --install xserver-vga16_3.3.6-11potato32.deb
dpkg --install xserver-svga_3.3.6-11potato32.deb
dpkg --install xterm_3.3.6-11potato32.deb
dpkg --install xv_3.10a-25.deb
dpkg --install gv3.5.8-17.deb
dpkg --install ivtools-interviews_0.7.9-6.deb
dpkg --install ivtools-unidraw_0.7.9-6.deb
dpkg --install ivtools-bin_0.7.9-6.deb
dpkg --install tcl8.2_8.2.3-2.deb
dpkg --install tk8.2_8.2.3-2.deb
dpkg --install ical-2.2-6.deb
dpkg --install addressbook_0.7-13.deb
dpkg --install libc5_5.4.46-3.deb
dpkg --install xpm4g_3.4k-5.deb
dpkg --install icewm_1.0.1-1.deb
```

Along the way you will be asked whether to make the VGA server the default and then whether you want to make the SVGA server the default. If you have decent equipment, you should be able to make the SVGA server the default.

Most of these packages are required to satisfy dependencies of the ones we really want to use. The postscript viewer `gv`, contained in the package of the same name, is indispensable for previewing documents in preparation using LaTeX and various manuals on the system as well as other postscript files obtained over the Internet. Through it, postscript documents can be printed all or in part via Ghostscript. You can use it by simply typing `gv` on the command line in a terminal window followed by the name of the file to be viewed. The `xv` package contains the program xv, which is used to view gif, jpeg, and other graphics files and convert them to different formats. The program can be run by entering the command `xv` in a terminal window. A manual can be found at `/usr/share/doc/xvdocs.ps.gz`. It will need to be unzipped using the command

```
gunzip /usr/share/doc/xvdocs.ps.gz
```

to obtain the postscript file `xvdocs.ps` which can be viewed and printed using the postscript viewer. The `ivtools` package contains the drawing program Idraw which you can use to make supply and demand and other drawings. A minimal manual is available off the web to show you how to use it. Go to

```
http://cns-web.bu.edu/pub/paolo/arvo-titles/idraw-help.html
```

The `ical` package contains a useful program for keeping track of your appointments and `addressbook` contains a program whose purpose is self-evident. A comparable program is not available for Debian 3.0. To use addressbook you will have to give yourself (rather than the root user) ownership of and write access to the directory in which the program resides, `/etc/addressbook/`, and all files in it. To do this, become root user and, after making `/etc` the working directory, issue the commands (substituting, of course, your username for `john`)

```
chown john addressbook
chgrp john addressbook
```

Then `cd` to the directory `addressbook` as follows and enter the further commands

```
cd addressbook
chown john *
chgrp john *
```

The `icewm` package contains the Ice Window Manager. A window manager interprets the mouse clicks and key strokes of the various graphics programs and passes along to the X-Server (which is the core of the X11 graphics interface) appropriate instructions to cause it to draw windows and other objects on the screen. Each of the many available window managers will present any given program with its own unique (and configurable) look and feel. The Ice Window Manager will end up being the default (and only) window manager on your system.

You will need Netscape Navigator to surf the Web, obtain data over the Internet, and handle your mail through external web-mail hosts such as hotmail, yahoo, etc., or ones provided by your college or university. Unzip the file `netsc476.tar.gz`, copy the file `nets4su` to `/usr/local/bin` and make it executable and then enter `nets4su` on the command line. The following commands will be executed automatically, one by one:

```
dpkg --install netscape-base-4_4.76-1.deb
dpkg --install netscape-base-476_4.76-1.deb
dpkg --install netscape-java-476_4.76-1.deb
dpkg --install navigator-base-476_4.76-1.deb
dpkg --install libstdc++2.9-glibc2.1_2.91.66-4.deb
dpkg --install xpm4g_3.4k-5.deb
dpkg --install navigator-smotif-476_4.76-1.deb
dpkg --install navigator-nethelp-476_4.76-1.deb
```

Alternatively, if you only have 16 megs of RAM you may want to use an older less resource demanding version of Netscape. In that case, unzip the files `netsc304.tar.gz` and `oldnets.tar.gz`, copy to the directory `/tmp` the netscape 3.04 archive

```
cp netscape-v304-export.x86-unknown-linux-elf.tar.gz /tmp
```

and install the following packages in order.

```
dpkg --install libc5_5.4.46-3.deb
dpkg --install xlib6_3.3.6-11potato32.deb
dpkg --install xpm4.7_3.4k-5.deb
dpkg --install motifnls_2.1-6.deb
dpkg --install netscape3_3.04-8.deb
```

If you want to switch from one netscape to the other, you can purge the currently installed netscape using dpkg with the command

```
dpkg --purge netscape3
```

or

```
dpkg --purge netscape4
```

and then, after erasing all files in `/home/yourusername/.netscape/` and removing the directory itself, install the other version as above.

You will need the Acrobat Reader to be able to view and print PDF files obtained over the Web. Unzip the file `extras.tar.gz` (which contains a number of useful packages in addition to the acrobat reader). You can install the acrobat reader using the command

```
dpkg --install acroread_4.05-3.deb
```

After unzipping the above zip file you will also find the Debian package `ox-3.00-2.i386.deb` containing the high-powered statistical program Ox. It can be installed using the command

```
dpkg --install ox-3.00-2.i386.deb
```

The manual for Ox can now be found on your system at
`/usr/local/lib/ox-3.00/doc/`
—you will need both `OxIntro.pdf` and `OxAppendix.pdf`. To read the help file on line, point your browser to
`/usr/local/lib/ox-3.00/doc/index.html`.
Add this URL to your Bookmarks.

As an alternative to using the program Idraw (`idraw`) to create graphs, you can use the program XFig (`xfig`). To set this program up, obtain the two files `xfig_3.2.3.a-6.deb` and `xfig-doc_3.2.3.a-6.deb` from my ftp site and install them using `dpkg`.

Additional packages from the Debian 2.2 distribution can be obtained by pointing your browser at the locations

```
http://http.us.debian.org/debian/dists/potato/main/binary-i386/
http://non-us.debian.org/dists/potato/non-US/
```

Next you need to move the `/home` partition to the partition `/dev/hda3`, replacing the files there that you used to set up the system. Give yourself root privileges by entering `su` on the command line and then typing in the root password when asked. Then make a new directory off the root directory—call it `/archives`—as follows:

```
mkdir archives
```

Now copy any files that you want to keep from the `/debs` directory to this new directory

```
cd /archive
cp /debs/filetobecopied .
```

The '`.`' in the last command is important—it tells the bash shell to put the file in the directory you are executing the command from. After copying to `/archives` any files from `/dev/hda3` that you want to keep, you need to empty the `/debs` directory by executing the command

```
rm -i *
```

from within that directory. The `-i` switch on the command tells the shell that you want to be prompted separately for each file you remove. This will take a while but it protects you from screwing up and erasing something you want to keep.

Now that all files have been removed from the `debs` directory and the partition `/dev/hda3`, go to the root directory and remove the `debs` directory as follows

```
cd /
umount debs
rmdir debs
```

Next, rename the directory `/home` to `/home.bak` using the `mv` (move) command and then make a new directory called `/home` and mount it on the partition `/dev/hda3`. The relevant commands are

```
mv /home /home.bak
mkdir /home
mount /dev/hda3 /home
```

Then copy all data from your old home directory `/home.bak` to the new directory `/home`.

```
cp -dpR /home.bak/* /home
```

This will also copy the files and subdirectories involved to `/dev/hda3` on which your new `/home` directory is now mounted. Actually, if you do this right after setting up your system there will only be a few hidden files and no subdirectories in your `home.bak` directory. To view these hidden files, use the command (from within that directory)

```
ls -a *
```

You can thus copy them to `/home` using a much simpler command than the one above.

Finally, you need to have your new home partition mounted when the computer is rebooted. To do this you need to edit the file `/etc/fstab`, adding to it the following lines (the lines beginning with `#` are comments you might put in to remind yourself later of what you did with the uncommented line).

```
# Addition to /etc/fstab
# Mount /dev/hda3 as /home into the file system
/dev/hda3      /home     ext2       defaults    0   2
```

Finally, you have to configure the X-Window System. Before doing so you should know the correct horizontal sync and vertical refresh rates for your monitor and the name and model of the video card that will be driving the display. You will be asked to choose among the following monitor setups.

```
 hsync in kHz; monitor type with characteristic modes
 1  31.5; Standard VGA, 640x480 @ 60 Hz
 2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
 3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced
                                                (no 800x600)
 4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced,
                                                 800x600 @ 56Hz
 5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz,
                                                640x480 @ 72 Hz
 6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz,
                                                800x600 @ 72 Hz
 7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
 8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
 9  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
10  31.5 - 95.0; Monitor that can do 1280x1024 @ 85 Hz
11  Enter your own horizontal sync range
```

You can choose one of the generic setups or enter your own horizontal sync range directly. You will then be asked to enter the vertical refresh rate of your monitor from the following list, with the latter option being to enter your monitor's specifications directly.

```
 1  50-70
```

```
2  50-90
3  50-100
4  40-150
5  Enter your own vertical sync range
```

In setting up your video card you will be given the opportunity to select from a large database of cards. If you choose one from the list it MUST BE AN EXACT MATCH for your card. If your card is not on the list you need not make any selection from this list—a generic installation will result.

You will also have to pick a server from the list below. Only options 2 and 3 are installed in the setup outlined here. You should pick option 3 if your equipment supports it.

```
1  The XF86_Mono server.  This a monochrome server that should
   work on any VGA-compatible card, in 640x480 (more on some
   SVGA chipsets).
2  The XF86_VGA16 server.  This is a 16-color VGA server that
   should work on any VGA-compatible card.
3  The XF86_SVGA server.  This is a 256 color SVGA server that
   supports a number of SVGA chipsets.  On some chipsets it is
   accelerated or supports higher color depths.
4  The accelerated servers. These include XF86_S3, XF86_Mach32,
   XF86_Mach8, XF86_8514, XF86_P9000, XF86_AGX, XF86_W32,
   XF86_Mach64, XF86_I128, XF86_S3V, and XF86_3DLabs.
```

It is worthwhile to know exactly what your choices here will be when you buy your equipment—this will enable you to seek advice from the technicians at the shop of sale so that you will know ahead of time exactly what your choices will be in the setup. If you already have a computer with a video-card in it and an old monitor, it might be useful to consult a local Linux guru. If issues like these are resolved beforehand, the installation will be a breeze—otherwise you can waste a lot of time fiddling around and wringing your hands!

The X-Window system for Linux consists of an X-Server that talks to the computer's hardware and X-clients on whose behalf the X-Server talks to that hardware. The most important and absolutely necessary X-client is the window manager whose job it is to take directives from other clients (programs) and arrange for the X-server to construct the necessary windows and manipulate them in response to the mouse and keyboard input of users of those programs.

As noted above, a number of pieces of information are necessary for setting up the X-Window system. Almost every graphics card and color

monitor will operate with the VGA-16 X-Server although the presentation will be rather crude, using only 16 colors. And most of the standard cards and monitors will operate with the SVGA X-Server, using 256 colors. Both these servers were installed above. More up-to-date fancy cards will use fancier X-servers which can also be installed, but most will work also with the above servers, albeit at below capacity. It is very important to have selected a card and monitor that will work with this Debian 2.2 setup when you purchased your computer.

Begin by configuring the X-Server. This is done by running, as root user, the program

```
/usr/sbin/xserver-configure
```

Before doing so read the manual page for xserver-configure which can be viewed by typing `man xserver-configure` on the command line.

Follow the prompts as you work through the configuration program. MAKE ABSOLUTELY SURE THAT YOUR HORIZONTAL AND VERTICAL REFRESH RATES DO NOT EXCEED THE RECOMMENDED RATES FOR YOUR MONITOR OR YOU MAY DESTROY IT! Run the test at the end to see if the configuration you have made is acceptable. You can look at and edit the configuration file if necessary. This file is `/etc/X11/XF86Config`.

Now start the X server, as yourself, not root, by typing `startx` on the command line. The Ice Window Manager should load automatically presenting you with a task bar along the bottom of the screen which you can click on with the mouse. Click on the various items on this task bar to figure out how to use this window manager. You will notice the squares numbered 1, 2, 3 and 4. These are alternative 'workareas' which really represent separate screens on which programs can be left open but out of view. When you use Netscape, for example, you might want to click on workarea 2 and open it there. You can then move between using Netscape and using the programs you were previously working with on workarea 1 by moving back and forth between workareas 1 and 2. This avoids having a muddle of windows plastered all over the screen.

The Ice Window Manager can also be operated using keystrokes. Enter `Ctrl+Esc` to open the menu on the bottom left corner of the screen. You can move around that menu using the arrow keys and press `ENTER` to activate choices. `Alt+F5` rotates the focus among the windows and `Alt+F4` deletes the currently focused window. `Shift+Esc` opens the window menu for the window currently in focus. Individual programs will also typically have key-

stroke commands for various functions. You will find the manual for the Ice Window Manager at `/usr/doc/icewm/`.

You can change the background color and other preferences by editing the file `/etc/X11/icewm/preferences`. To change the background color to a nice blue, edit the line (leaving, in this case, the `#` character in place)

```
# DesktopBackgroundColor="rgb:00/50/60"
```

to read

```
# DesktopBackgroundColor="rgb:46/82/B4"
```

For more information on colors, point your browser to

`http://eies.njit.edu/~walsh/rgb.txt.html`

17 Appendix D: Getting on the Internet Using a Modem

Your most important task is to get your new Debian system connected to the Internet. If you have established a network connection using an Ethernet card, this is already accomplished.

If you are working through a modem, you have already established a dial-up PPP connection using BasicLinux. The task now is to do the same thing in your newly set up system. This should be easy if your ISP uses the PAP protocol. In the case of the University of Toronto systems, where chat scripts are used, it is quite difficult—thus far impossible for a UTORDIAL connection.

The easiest thing to do is to copy your BasicLinux files to an appropriate place in your new Debian Linux system. Boot your computer into BasicLinux and mount the partition on which your Debian system is installed. Then, after making back-up copies of the `pap-secrets` and `options` files in the `/etc/ppp` directory of your Debian system, copy the `ppp-on-dialer`, `pap-secrets` and `options` files from your BasicLinux RAM disk to the same directory on your Debian root partition.. Then, go to the `/etc` directory of your Debian system and make a back-up copy of any file called `resolv.conf` that happens to reside there. Now copy the `resolv.conf` file from your BasicLinux system to the `/etc` directory of the root file system on your Debian system. Finally, copy the `ppp-on` and `ppp-off` files from the `/usr/sbin` directory of the BasicLinux system to `/usr/local/bin` on your Debian system.

Now boot into your Debian system again and give yourself ownership of all of the above files by executing the commands (as root user)

```
chown john filename
chgrp john filename
```

substituting, of course, your username on your system for `john` and inserting the appropriate file names. Also, make sure that the files `ppp-on`, `ppp-off` and `ppp-on-dialer` are executable.

You should now be able to dial in to your ISP. If things don't happen as they did in BasicLinux, edit the `ppp-on-dialer` file and change the line `exec chat` to read `exec chat -v`. Then dial in again and, after things crap out, read (as root user) the bottom lines of the file `/var/log/syslog` to find out what happened when you dialled in. You probably have made some mistake in placing the files or assigning permissions or ensuring that appropriate files are executable.

As noted, I was successful in establishing a PPP connection to CHASS on my Debian system by this method. I could also establish, according to what the `syslog` file tells me, a PPP connection to UTORDIAL. Unfortunately, however, that connection did not work—no data could pass through it. And little can be done because the staff at Information Commons refuse to discuss Linux!

The above method of transfer between BasicLinux and Debian should also work for simpler types of connection that use either the PAP or the CHAP protocol. I have not confirmed this because I use the University of Toronto system as my ISP. If your ISP uses one of the above protocols and you cannot convert your BasicLinux dial-up to Debian, you can use the `pppconfig` program, which should be installed as part of the base system, to try to set up a connection. Log on as root and type `pppconfig` on the command line. Using what you already know, you should be able to respond appropriately to the prompts. If you have trouble, I suggest that you read online a few more sections of McCarty's book. You will find the section on PPP configuration at

`http://www.oreilly.com/catalog/debian/chapter/book/ch11_04.html`

If you want to learn more about configuring your modem, you can download the minicom package (and the libncurses package it depends on) from my ftp site and install them using the commands

```
dpkg --install libncurses4_4.2-9.deb
dpkg --install minicom_1.82.1-1.deb
```

and read `ch11_02.html` and `ch11_07` on the O'Reilly web-site above. Minicom is a program that you can use to dial into your ISP and appropriately configure your modem. Using minicom, you can also log directly onto your account there without establishing a PPP connection, provided that your ISP supports *shell* accounts (as CHASS does). If you approach the Web by this route, however, you will only be able to browse the Web using the Lynx text browser from your ISP account, and then only if your ISP has Lynx installed.

18 Appendix E: Using Your Debian System

You should now copy the file `uslocbin.tar.gz` obtained from my ftp site to the directory `/usr/local/bin/` and unzip it, making sure that all the files disgorged are executable. These consist of a number of programs that I have obtained from the Web or written myself to facilitate my research work. For example, included is a script `oxmanual` that accesses the on-line manual to the Ox statistical program, the installation of which is discussed above. Next, get and move the zip-file `usloclib.tar.gz` to the `/usr/local/lib` directory and unzip it. These files will also be usfull in your research as noted below.

18.1 Some Tweaks and Tips

In the course of managing your computer and doing your work you will need to collect files together in compressed archive files and to unzip archive files that others have created. The standard way to do this is using the `tar` command. To create an archive, use the command

```
tar cvzf name.tar.gz file1 file2 file3 *.txt
```

where `name` is the root name of the file you are creating and `file1`, `file2`, `file3` and `*.txt` are the files to be compressed into the archive. You can include as many files as you wish in any archive. To unzip an archive with the suffix `.tar.gz` or `.tgz`, execute the command

```
tar xvzf filename
```

Sometimes you will encounter an archive of the form `filename.gz`. In this case, unzip the archive with the command

```
gunzip filename
```

You will also occasionally run across zip files with the suffix .zip that have been created on MS-Windows systems. To unzip these in Debian 3.1 you can install the program `unzip` in the usual way using `apt-get`. Use this program by entering the command `unzip` followed by the name of the file.

All of the programs installed here have manual pages that you can access from the command line by entering

```
man programname
```

There are a few additional tweaks to your system that you should know. To reset the time, execute the command

```
date MMDDHHMMCCYY
```

where `MM` is a two-digit representation of the month (06 for June), `DD` is a two-digit representation of the day, `HH` is a two digit representation of the hour, and CCYY is a four-digit representation of the century and year (2003 for the current year). Entering `date` by itself will cause the shell to print on the screen the date and time. To change the time and date you must operate as root user.

To configure your system to set the local time in relation to Greenwich Universal Time enter (as root user) `tzconfig` on the command line and follow the prompts. For Toronto and Canada you would choose '4' in response to the country prompt and 'Eastern' ('Central' for daylight saving time) in response to the region prompt.

To check what day of the week a particular day falls on in the current month, use the `cal` program by entering the word `cal` on the command line. To access the dates in a month other than the current one, enter `cal mm`, where mm is a two digit place-holder for the month. To obtain a full calendar for a particular year, enter `cal yyyy` where `yyyy` is a four-digit place-holder for the year. To obtain a calendar for the month in which King Henry VIII of England had his fifth wife Catherine Howard be-headed for the crime of adultery, enter `cal 03 1543`.

To find files on your system, execute, as root user, a command of the following form

```
find -name xxxx*
```

from the root directory after substituting the appropriate filename for `xxxx*` or filename segment for `xxxx`.

If you boot your system from the boot disk you created upon installation the boot process will take a long time. As an alternative, you can easily make

a different type of boot disk for your Linux system. Make a DOS boot disk and copy to it the program `loadlin.exe` which you can obtain from my website if you have not already obtained it to make a DOS install of the base system. Then seek out the linux kernel which you will find on your system in the directory `/boot`. This file will have a name like `vmlinuz-2.2.20`. Copy this file to your boot disk, renaming it as simply `vmlinuz`. Then create an `autoexec.bat` file containing the single line

```
LOADLIN vmlinuz root=/dev/hda1 ro
```

substituting your root partition for `/dev/hda1` if it happens to be different. Now copy this file to your boot disk. That disk should boot up your Linux system. You can alter this disk to boot any linux system by simply copying the appropriate kernel to it and changing the root partition if required.

18.2 Using the Editor joe

To start joe to edit a particular file or files, make the following command line entry as appropriate

```
joe filename
joe file1 file2 file3
```

Most of the commands in joe involve holding down the `Ctrl` key while pressing `K` and some other specific key. To move between file windows when more than one file is loaded onto the screen, use the commands `Ctrl+K,N`, (i.e., hold down `Ctrl` and press `K` and then `N`) to go down to the next file and `Ctrl+K,P` to go up to the previous one.

To save the current file, enter `Ctrl+K,D`. To exit enter `Ctrl+C` and to save and exit enter `Ctrl+K,X`. To block text for copying or deleting, enter `Ctrl+K,B` at the beginning of the block and `Ctrl+K,K` at the end of the block. Copy the block to a new cursor position by using `Ctrl+K,C`, move it to the new cursor position using `Ctrl+K,M` and delete it using `Ctrl+K,Y`. Undo the effects of previous commands by using `Ctrl+_`.

To move a rectangular area from one place in a file to another, enter `Ctrl+T` and then the letter `X` to enter rectangular mode. Then block the area to be copied, moved (or deleted) by entering `Ctrl+K,B` at the upper left corner and `Ctrl+K,K` at the bottom right corner. Move the cursor to the place where you want the upper left corner of the material to be inserted and execute either `Ctrl+K,C` or `Ctrl+K,M` as appropriate. Delete the rectangular area of material by entering `Ctrl+K+Y`. Get back into insert mode by entering the command `Ctrl+T` and then pressing `ENTER`.

Write blocked material to file using `Ctrl+K,W` and insert a file at the cursor position by entering `Ctrl+K,R`.

All these commands and more are outlined in the help file which can be brought to the top of the screen using the command `Ctrl+H` and removed using the same command.

18.3 Using mtools

As it is set up, `mtools` can only be used by the root user. For ordinary users, namely you as yourself, to use the program, read and write priviledges to the floppy drive `/dev/fd0` have to be extended using the commands (as root user)

```
chmod o+r /dev/fd0
chmod o+w /dev/fd0
```

To copy files from your hard drive to a floppy disk in Drive A, use the command

```
mcopy filename a:
```

if the file is a binary file or

```
mcopy -t filename a:
```

if it is a text file. To copy a file from Drive A: to the directory you are currently in, use the commands

```
mcopy a:filename
mcopy -t a:filename
```

depending upon whether the file is a binary file or a text file. To delete a file on Drive A: enter the command

```
mdel a:filename
```

and to make a list of the files on a disk in Drive A: enter the command

```
mdir a:filename
```

To format a disk for use with DOS enter the command

```
mformat a:
```

18.4 Logging Into and Obtaining Files from Other Computers

To log into another computer—say, your university system or Internet provider, enter the command

```
ssh -l username othersystem
```

where `othersystem` will be a name like `credit.utm.utoronto.ca`, and `username` is your username on that system. Respond with your password when prompted. To copy a file from the other computer to your computer, use the command

```
scp username@othersystem:~/filename .
```

where one or more directory names can be inserted before the filename. To copy a file from your computer to the other computer, use the command

```
scp filename username@othersystem:~/
```

Again, directory names can be added.

18.5 Number Calculations, Spreadsheets and Graphs

Use my simple command-line calculator by entering `calc` on the command line. Type `help` or any other four letters to obtain an on-line manual.

To load the spreadsheet program `sc` enter `sc filename.sc` on the command line. If no filename is entered a new spreadsheet will be started and you will be prompted to supply a name the first time you save it. I have written four useful utilities for this program. The utility `mat2sc` converts a text file containing a matrix of numbers into an `sc` file called `tempmat.sc` with the upper corner of the matrix fixed to a particular cell. Use this utility with the command

```
mat2sc filename.mat C 12
```

where `C 12` is the column and row of the cell in which the upper left corner of the matrix will reside. The resulting file `tempmat.sc` can be either renamed or merged into a pre-existing spreadsheet file. A second utility, `text2sc`, converts a column of lines of text into a single-column worksheet file called `temptext.sc` with leftmost character of the upper line tied to any chosen cell. This file too can be copied under a new name or merged with a pre-existing spreadsheet. A third utility program, `cdsst2sc`, converts a

comma delineated text file, identical with the spreadsheet files produced by the Chass Data Centre, into an `sc` file called `tempwks.sc` to be subsequently renamed. In addition to the name of the source file, a token to occupy empty cells must also be specified on the command line. The choices are `NA`, `-1`, `-999` and `0` (to leave the cell empty). The source file must contain comma's separating the cells, with quotation marks around the contents of those cells consisting of strings (words) rather than numbers. A fourth utility program `cdtxt2sc` also converts a comma delimited spreadsheet to an `c` worksheet using the same command line as the previous program. The difference is that strings in the source file must not have quotation marks around them—the program infers which cells contain strings and which contain numbers. Typing the name of any of the above utility programs by itself on the command line will produce an explanation of how to use it. The maximum permissible width of the worksheet files for the three utilities that process numbers and not just text is 26 columns.

A Gnuplot manual will be present on your system in the form of a zip file called `gnuplot.ps.gz` in the directory `/usr/share/doc/gnuplot`. Unzip it with the command

```
gunzip /usr/share/doc/gnuplot/gnuplot.ps.gz
```

to convert it to an ordinary postscript file and then view it using the command

```
gv /usr/share/doc/gnuplot/gnuplot.ps  &
```

The `&` on the end of the command line tells the shell to spawn a completely separate window for the viewing program, `gv`, so that the window in which the command is executed can be used for other things while the postscript file is being viewed. This program operates through menus that are self-explanatory.

The program `pl` uses Gnuplot to plot text matrix files created with spreadsheet programs. Before using it you should create in your home directory (as yourself, not as root) a resource file called `.plrc` containing the lines

```
set data style points
set data style lines
```

As configured, the points will be connected by lines, as required for time series plots. To leave the points unconnected and designated by diamond-shaped symbols, comment out the second line by placing the character `#` in

front of it. For time series plots, the left-most column of the data matrix should be a vector of dates. If the data are not annual, the dates must be written as, say, 1981.0, 1981.083333, 1981.166667, 1981.25, ... etc., for monthly data, and 1981.0, 1981.25, 1981.5, 1981.75, 1982.0, ... etc., for quarterly data. To plot all columns of the matrix, simply enter the command

```
pl filename
```

To plot column 2 against column 1 (which will be the date vector in the case of time series), enter the command

```
pl -u 1:2 filename
```

To plot both columns 2 and 3 against column 1, enter the command

```
pl -u 1:2 -u 1:3 filename
```

To make a scatter plot of columns 2 and 4 of the matrix, comment out the appropriate line in `.plrc` and enter the command

```
pl -u 2:4 filename
```

Before I was aware of the `pl` program, I constructed some scripts to make quick plots using Gnuplot. These are still useful on occasion. To make time series plots of one, two, three, or four time series, respectively, each series should be copied, using the spreadsheet or a statistical program, to a separate two-column file of real numbers containing the dates of the observations in the left column and the observations themselves in the right column. These data files should be named `tsdata1.tmp`, `tsdata2.tmp`, `tsdata3.tmp`, and `tsdata4.tmp`. Then enter either `plotts1`, `plotts2`, `plotts3`, or `plotts4` on the command line, depending on the number of series you wish to plot. The four scripts with the above names, which must be resident and executable in the directory `/usr/local/bin` (and are contained in the zip file `uslocbin.tar.gz`), instruct the shell to copy the respective template files `plotts1.gpt`, `plotts2.gpt`, `plotts3.gpt` and `plotts4.gpt`, which must be resident in `/usr/local/lib`, as files of the same root name but with suffix `.tmp` in the directory from which the command is issued and in which the data files above must reside. The shell then executes the respective commands `gnuplot plotts1.tmp`, `gnuplot plotts2.tmp`, etc., to plot the relevant set of time series to the screen. The files `plotts?.gpt`, which will be placed in the `/usr/local/lib` directory when you unzip `usloclib.tar.gz` there, can be used at as templates with which to create charts for inclusion in LaTeX documents. These `.gpt` files can be copied

under new names and edited in obvious ways, and corresponding data files renamed with, say, `.gd` instead of `.tmp` suffixes, to produce the type of chart desired. To produce a scatter plot, copy the data for the two variables side by side in a text file called `xydata.tmp` with the variable to be plotted on the horizontal axis in the left column. Then execute the command `plotxy` on the command line. The gnuplot command file `plotxy.gpt` will be copied from `/usr/local/lib` to the working directory as `plotxy.tmp` and the chart constructed. The file `plotxy.gpt` will also be a useful template file for making scatter plots for inclusion in documents. After doing quick and dirty plots of the kind above, you will find it useful to clean up after yourself by executing the command

```
rm -i *.tmp
```

in the working directory.

Probably the easiest practical way to obtain a frequency distribution is to write the numbers to a single-column text file, sort the file using the command

```
sort < file > sortedfile
```

and then, after setting up appropriate classes, simply count the number of observations in each class. If you have a large data set you can read the list of numbers to be sorted into XlispStat and use that program to create a histogram from which can be extracted the frequency distribution. The proper commands to do this are explained in my short manual for XlispStat, which can be obtained from my ftp site. Neither Gnuplot nor XlispStat plot histograms in a form sufficiently pleasing to the eye to merit inclusion in publications. The best way to incorporate a histogram in a LaTeX document is to create it in the form of a bar chart within the document itself. See below for further discussion.

For drawing simple charts, use the program XFig (the command is `xfig`). A manual for this program can be

```
http://www.xfig.org/userman
```

Alternatively, if you are working in Debian 2.2 you can use the program Idraw. A manual for this program can be found at

```
http://cns-web.bu.edu/pub/paolo/arvo-titles/idraw-help.html
```

18.6 Printing Text and Spreadsheet Files

I have written a number of shell scripts for using the program `a2ps` to convert text files to postscript and print them. To print a text file in standard `a2ps` form, enter `prntxt filename` on the command line.[10] This prints the file, two pages side-by-side on each sheet of paper in landscape mode, with the pages numbered and the filename written across the top of each page. To print the pages one to a sheet in portrait mode with normal-sized print, use the command `prntxtb`. To print a file having lines a bit too long to fit on the page in normal print, use the command `prntxtw` followed by the file name. To print a spreadsheet file, saved as a text file by `sc` or Gnumeric, use either `prntxtw` above if there are a small number of columns, or `prntxtxw` if there are many columns—the limit is 204 text characters of width (e.g., 10 columns, each 20 characters wide). The first of these utilities prints the file in portrait mode with small print (the size used in `prntxt`) while the second prints the file in landscape mode with the same small print. For each of these four utilities there is a corresponding one for printing to a postscript file instead of directly to the printer. For printing to file, simply add the character `f` to the beginning of the script name—for example `fprntxtw` instead of `prntxtw`. All these eight scripts will be in `/usr/local/bin` if you unzipped `uslocbin.tar.gz` there. They must, of course, be executable.

18.7 LaTeX Tips

LaTeX is a document preparation program used by a wide variety of academicians in many fields. You can learn how to use LaTeX by reading *The Not So Short Introduction to LaTeX* 2_ϵ by Tobias Oetiker et. al.[11] It is also worthwhile to obtain Leslie Lamport's little book, LaTeX *A Document Preparations System: User's Guide and Reference Manual*, Addison-Wesley Publishing Company, 1994. One prepares a formatted text file (these always have the suffix `.tex`) for the document and then processes it by entering on the command line

```
latex filename
```

without the suffix (LaTeX automatically adds the suffix `.tex` if it is not specified on the command line). This will result in the creation of a device

[10] a2ps will refuse to print files with the suffix `.tex`, cleverly thinking that the file should be processed first! To trick it, make a copy of the file called `filename.texcode` and print the copy.

[11] Go to `http://people.ee.ethz.ch/oetiker/short/short.pdf`.

independent or `.dvi` file with the same root name that can be processed by a wide range of printing devices. Here we process it to create a postscript file using the program `dvips` by entering

```
pps filename
```

again omitting the suffix, which will be `dvi`. Actually the required command line is really

```
dvips -t letter filename.dvi -o filename.ps
```

but I have created the shell script `pps` which instructs the shell to execute the command

```
dvips -t letter $1".dvi" -o $1".ps"
```

where `$1` is the root filename entered on the command line. If you only want to process particular pages of the document you enter the command

```
ppps filename p1 p2
```

where `p1` and `p2` are numbers indicating the first and last pages to be processed. The shell script `ppps` contains the single line of code

```
dvips -t letter -p$2 -l$3 $1".dvi" -o $1".ps"
```

where `$1`, `$2`, and `$3` are place-holders for the file name (minus the suffix) and the two page numbers. These two scripts are among those zipped up in the file `uslocbin.tar.gz`. If you are working in Debian 3.1, you can create a PDF file by entering the command

```
dvipdfm filename
```

again leaving off the suffix.

To simplify writing letters and memos with LaTeX I created two template LaTeX code files called `lettemp.tex` and `memotemp.tex`. You will find them in the directory `/usr/local/lib` if you unzipped `usloclib.tar.gz` there. To create a letter or a memo, simply copy the respective template file under an appropriate new name and modify it according the instructions contained in the file itself. You can process the template files and any files created from them by the usual method.

18.7.1 Including Graphics in LaTeX Documents

To include figures and charts in LaTeX documents, first save them as postscript files. For charts containing time-series and scatter plots, simply uncomment the following lines in the Gnuplot command file (which you should have obtained by modifying the appropriate template)

```
#set terminal postscript eps
#set output 'filename.eps'
```

by removing the `#` characters from the beginning of the lines and appropriately changing the filename. When you process the file with Gnuplot it will write the chart to an encapsulated postscript file rather than put it on the screen. Figures and drawings made with XFig and Idraw can also be incorporated into LaTeX documents rather easily by saving them as `.eps` files.

To incorporate postscript graphic files into your LaTeX document you first need to add a line

```
\input epsf
```

in the preamble of the document. Then create a separate `.tex` file for each figure or chart or cluster of charts and insert it at the appropriate place in your document with the command

```
\input figfile.tex
```

where the suffix `.tex` can be left off the command (but not the file). I have created a template `figtemp.tex` that will also be in `/usr/local/lib` if you unzipped `usloclib.tar.gz` there. If you want to include three figures on the page, all you need to do is copy `figtemp.tex` under an appropriate name and change the names of the postscript `.eps` files to match the ones produced by your Gnuplot commands. And, of course, you will also have to alter the figure number and the descriptive comment at the bottom of the file. The code line

```
\baselineskip 12 points
```

in `figtemp.tex` makes the space between the lines of comment at the bottom of the page smaller than between regular lines of text in the document. You can change the number 12 to something else, or delete the line entirely, if you wish. If you want to change the size of a chart, modify the relevant line

```
\epsfxsize=4.7in
```

to change the width from 4.7 inches to an appropriate alternative number. A corresponding change in the height of the graphic will occur automatically. If you want to include only two charts in the figure, or one, simply delete the lines of code pertaining to the charts you do not want to include. You can include a fourth chart by creating an additional line, but this will make the figure longer than a page. You will then have to make the charts smaller in the vertical dimension. You can do this by altering the line

```
set size 1.0,0.7
```

in the Gnuplot command files `name.gpt` to read, say,

```
set size 1.0,0.55
```

to reduce the vertical height of the chart to 55% instead of 75% of the default.

You can also adjust the `\vglue = .8` command at the top of figures spawned from `figtemp.tex`, or add a similar command elsewhere in the file, to move graphics further up towards the top of the page, or down towards the bottom, or to change the space between the chart or figure and any text that may be above or below it.

Sometimes you will be trying to include a postscript drawing or chart in your document and it behaves in an unruly fashion, taking up much more space than is covered by the actual drawing. This is an indication that the drawing or chart is not encapsulated by a bounding box. We usually denote encapsulated postscript files made with Gnuplot or `xfig` by giving them the suffix `.eps`. This errant figure will probably have the suffix `.ps` rather than `.eps`. To fix the problem you have to specify a bounding box for the chart or drawing. Print the graphic on a page by itself by entering the command `lpr` followed by the name of the graphic. Then take a ruler and measure the distance from the left edge of the paper to the left edge of the printed graphic in inches. Multiply the number of inches by 72 to convert the distance into postscript units and call this distance `llx`. Now measure the distance from the bottom of the page to the bottom of the printed graphic and convert that distance to postscript units, calling it `lly`. Next measure the distance from the leftmost edge of the paper to the rightmost edge of the printed graphic in postscript units and call it `urx`. Finally, measure the distance from the bottom of the page to the uppermost edge of the printed graphic and call that `ury`. A bounding box can then be added to the graphic by changing the line

```
\centerline{\epsffile{name.eps}}
```

in the `.tex` file for the figure to

```
\centerline{\epsffile[llx lly urx ury]{name.eps}}
```

so that the line will look something like

```
\centerline{\epsffile[124 145 486 502]{name.eps}}
```

When you print out a graphic on a sheet by itself and it is encapsulated by a bounding box, it should appear on the bottom left corner of the page.

To learn more about these issues you can obtain a copy of the `dvips` manual from the web at

```
http://www.ctan.org/tex-archive/dviware/dvips/dvips_man.ps.gz
```

Download and unzip the graphic (use `gunzip dvips_man.ps.gz`) and print the resulting postscript file. Be prepared for a heavy read!

If you want to incorporate histograms in your document you will probably obtain the best results by actually creating a bar chart from the numbers in the frequency distribution directly in LaTeX using the Barchart package. To do this you will need to consult a book called *The LaTeX Companion*, written by Michael Goossens, Frank Mittelbach and Alexander Samarin, and published by Addison-Wesley (1994). Work through pages 283–289. If you want to try and wing it, go to the following Web location

```
http://www.ichimusai.org/latex/#barchart
```

You may have to obtain the style file `bar.sty` from the Web. You will find it on my ftp site or at

```
http://www.ctan.org/tex-archive/macros/latex209/contrib/misc/bar.sty
```

The file should be copied to the directory

```
./usr/share/texmf/tex/latex/tools/
```

on your system where the other `.sty` files reside.

One problem you will occasionally have is getting your 'figures', defined in LaTeX code by the `\begin{figure}` and `\end{figure}` commands in your modified versions of `figtemp.tex`, to appear where you want them in the document. You may have to move the `\input figfilename` commands ahead or back in the document to tweak LaTeX into doing what you want. If a group of charts slightly overfill a page (you won't notice this from viewing the document) LaTeX will put that page and all subsequent graphics at the end of the document. See page 197 of Leslie Lamport's book for instructions on how to tweak LaTeXinto putting the graphics where you want them in the document.

18.8 Using TEX Instead of LATEX

If you are really strapped for cash you may want to avoid having to buy Leslie Lamport's book by using TEX instead of LATEX so that you can use the free manual "A Gentle Introduction to TEX" by Michael Doob. You can obtain this manual in postcript form, `gentle.ps`, from my ftp site. TEX is the programming language for document preparation on which LATEX is built. When you use TEX you are using that programming language in the raw. The commands are somewhat different than the ones you would use in LATEX and, of course, it is much harder to do fancy things because you have to program the fancy stuff yourself. Nevertheless, you can do basic document preparation quite easily in TEX. Simply follow the instructions in the manual. The inclusion of graphics in your document will use the same approach as outlined above for LATEX (`dvips` does not care whether you are using LATEX or TEX), with a few minor but crucial adjustments. First, when you are using the template `figtemp.tex` you will have to delete the `\begin{figure}` and `\end{figure}` commands on the first and last lines of the file. At the end of the code you will have to restore `baselineskip` to the value you are using for the regular text in the document. Moreover, you will save yourself a lot of trouble by putting all drawings, whether they form a group that fit on a single page or are individual stand-alone graphics, at the end of the document rather than trying to insert the code for these graphics within the document. This will avoid pagination hassles. If you want to have your graphics distributed throughout the document, rather than at the end, the best way to proceed is to insert an `\advancepageno` code on a line by itself wherever a figure or chart or group of related figures or charts that fit on a single page will appear. This will cause TEX to skip that page number in numbering the pages of the document. Then insert a new page command `\vfill\eject` at the end of the document where the first figure or chart, or page of charts, is to be printed and add a command like `\pageno=13` or whatever to reset the page number to the number of the page on which the graphic will appear. Do this for each page of graphics, some of which will only have a single chart or drawing on them, and then after printing the document move these pages to the place where they belong in the document.

18.9 Spell-Checking Your Documents

To spell-check a document, use the `ispell` program by entering the command

```
ispell docname.tex
```

Words that `ispell` does not recognise will be highlighted and you must respond according to the menu along the bottom of the page. The option `A)` means that you accept what is there regardless of what `ispell` thinks. If `ispell` has in mind a different possible spelling for the highlighted word it will number and list it on the screen. You can choose to replace the word with one suggested by `ispell` by simply pressing the appropriate number.

After you have spell-checked the document you should check it for double occurrences of words—these happen when you accidentally type a word, usually to, the, or a, twice in succession. You can do this with my little program `dubword` by entering

```
dubword docname.tex
```

An output file called `dubword.out` will be created, containing the line numbers of all incidents of double typing of words and the word involved in each instance.