The Hitchhiker's Guide to FSU Mathematics Computing

by Mickey Boyd

August 15, 2014

Contents

Chapter 1

Introduction for New Grads

Welcome to the Florida State University Department of Mathematics. This document describes the various technology resources that are available to Mathematics department computer account holders, along with the policies associated with them.

Here are some important things to do soon after receiving your FSU Mathematics computer account:

- Check the spelling of your name on your password slip. Email `root@math.fsu.edu` if corrections are needed. Do this first, as an error here might change your email address and login info. Also, send yourself an email and make sure your full name appears as desired in the `From:` field.
- Consider your email strategy. You will have at least two different FSU email accounts to keep track of during your time in our department. Read the section in this document concerning your email options, and choose wisely.
- Establish your official Math department webpage. This is easy to do, by using the Graduate Student Webpage editor found in the Graduate Student Resources section of our website (click "Student Resources" on the main Math webpage). This will be your official student web presence within our department, so please spend some time making it informative. You can be as extensive as you like with your web entry. Also, if you are teaching you can publish important course info for your students using the editor.
- Explore the FSU Mathematics website. Note the "Graduate" and "Tech Support" sections of the webpage under the "Resources" dropdown menu. You will find the answers to many questions commonly posed by new users.

Chapter 2

Email Options

It is important that you regularly monitor your Math department email account, `user@math.fsu.edu`. There are many different approaches to this. Here are some suggestions:

2.1 General Email Forwarding

You can forward your Math department email to any other valid email address or webmail system. It is important that you read email sent to both your official FSU email address (either `user@fsu.edu` or `user@my.fsu.edu`), and your Mathematics email account. Forwarding one to the other will allow you to read all of your FSU related messages in one place. Instructions for creating an email forward are in the Tech Support section of the Math website. It only takes a minute via a simple web tool. Alternately, you can just email `root@math.fsu.edu` with a note saying "Please forward my Math email to some@email.address", and we will take care of it for you.

2.2 Gmail and Other External Webmail

Gmail can be used to manage your Math department email, and is the most popular choice among our grads. Gmail has nice features for managing multiple email accounts, making it easy to keep email from different accounts as separate as desired. The search capabilities are excellent. Gmail also preserves your ability to originate emails from your Math department email address `user@math.fsu.edu`. In other words, you can select which address appears in the `From:` field when you send an email, or have it select for you automatically. This can be very handy if you are managing several email accounts from one gmail account.

To set up gmail to manage your Math department email, go to the Tech Support section of the Math webpage and click the Email Settings link at the top of the page, then click on the large Gmail icon and follow the screenshots.

There are other webmail systems on the Internet that work well with multiple accounts. If you want to use such a system, see their help section for instructions on how to manage external accounts. The email forwarding instructions in the Tech Support section of the Math website will probably be useful in some part of the process.

2.3 External Email Clients

If you wish to keep your Math email completely distinct from any other email accounts, an external email client might be your best choice. Our email servers use the POP3 protocol with encryption.

Within your client settings, for incoming mail select POP and set server to `mail.math.fsu.edu` as well. If you are asked, set encryption (SSL preferred), and set port to 995.

Similarly, for your outgoing mail (SMTP) settings, set server to mail.math.fsu.edu and select password authentication (often with a checkbox). If you are asked, set encryption (TLS preferred) and set port to 587.

We have detailed instructions for several popular email clients in the Tech Support section of our website. The Math Email Settings link is at the very top of the page. The clients are: Apple Mail, Thunderbird, Eudora PC, and Gmail.

2.4 UNIX Command Line Email Clients

If you prefer a text-only approach to email, or if you want to do a quick check while logged in remotely, then try a UNIX command line email client. We have several from which to choose: `mutt`, `pine`, and even the original `mail`. You can see online documentation for each of these programs using the UNIX `man` command.

Chapter 3

Generalities

3.1 Computer Labs

Mathematics has two computer labs. They are primarily intended to provide access to specialized mathematical applications that cannot be found in the various general-use labs located on campus.

- Grad Lab 1 (004D Love): Main graduate lab.
- Grad Lab 2 (406 MCH): Mini lab, 4th floor of Milton Carothers Hall.

All grad students are issued a key that give 24 hour access to these rooms. These labs offer a UNIX workstation environment, and access to many specialized mathematical applications (see the Tech Support section of the website for details). There are also several FSU managed general-use computer labs on campus (offering Mac and Windows computer with standard office software), including a large one on the 3rd floor of Milton Carothers Hall open to all students.

3.2 Lab Conduct

Our computer labs are intended to provide a good work environment for students engaged in research. Please remember this, and conduct yourself accordingly. No food or beverages are allowed in the labs. The lab should be kept neat in appearance. If you use any of the manuals we stock in the labs, please return them to the bookshelves. Finally, please keep the noise level down.

3.3 Contacting the System Group

The system support staff provides needed system services such as software installation, hardware repair, file system backups, and system administration. In addition, they are available to answer general system questions[1].

In case of minor problems like a workstation lockup, printer problems, etc., send email to `root@math.fsu.edu`. Do not send email to a particular individual (e.g. boyd), as that party may

[1] Please attempt to resolve your questions using the various methods outlined in this document before contacting the system staff. Most of the questions we are asked can easily be answered via our this document or the Tech Support section of our webpage

be out of town or unavailable. There will always be someone responsible for resolving email sent to root@math.fsu.edu.

It is common for the system group to receive problem reports that are inadequate in terms of information. For any problem report, please include:

- The name and location of the machine upon which it occurred.
- A step by step description of how to invoke the problem.
- The approximate time the problem occurred.

In case of major emergencies like server breakdown, fire, etc., call the emergency pager. To use the pager, first dial 850-657-6425, listen to the instructions, type in the phone number at which you can be called back using the telephone touch pad, then hang up. Normally, someone will attempt to call you at the number you entered within 15 minutes. If you think you may have made a mistake when entering the phone number, simply call the pager number and try again. Multiple attempts are OK.

If you are unsure whether the problem you wish to report is major or minor, please go ahead and call the emergency pager. We would rather receive a few extra pages than be uninformed about something important.

3.4 General Policies

The Mathematics department supports hundreds of computer accounts at any given time. To do so, we must supply a reasonable level of security and privacy to all users. Remember, you are one of many users on our system. We also have an obligation to provide certain services and resources. This section outlines some of our general policies. Please read it carefully. If anything is confusing or unclear, please ask an appropriate staff member for clarification.

3.4.1 User Accounts

In this department, there is a one-to-one mapping of user accounts to individuals. There is absolutely no legitimate reason to give your password to anyone else. Doing so will result in the termination of your account, at minimum. It is possible that you will have more than one account on our system during some semesters, but they should be YOUR accounts. There are no group accounts.

We are very serious about this, as you should be. If you feel that someone has accessed your account without you present, inform the system group immediately (and change your password).

Mathematics accounts are intended to provide resources for Mathematics classes and research. They are not intended to provide resources for non-mathematics courses. They are not intended to provide resources for outside employment.

3.4.2 Electronic Mail Privacy

Occasionally, the system group may have to examine files containing electronic mail, to debug system problems and possible security intrusions. Other than that, the privacy of your electronic mail is entirely determined by the permissions of the files into which it is saved. The system mailbox (where you receive mail) is private by design. However, saving your email into other files could make it readable by other users on the system, unless file permissions are set correctly (see below for details). Thus, if you are concerned about email privacy, make sure you fully understand the section of this manual relating to file permissions.

3.4.3 Computer Security

There are those who feel that attempting to crack a security system is OK, given that there is no malicious intent. **We are not among them**. Do not attempt to penetrate any of our security systems. Doing so constitutes an illegal use of computer resources, and will be dealt with accordingly. On the other hand, if you notice something that may allow a malicious user to do damage please let us know. Your files, perhaps your dissertation, important class work, or private email, are among those things threatened by any illegal intrusion into our systems.

3.4.4 Internet Access in Math Grad offices

Internet access for student-owned computers is provided in the form of the FSUWIN campus wireless network. The FSUWIN campus wireless network provides good signal strength in all Math graduate student office areas (4th floor and 2nd floor Milton Carothers Hall). Visit this website for full details on FSUWIN: `http://helpdesk.fsu.edu/Network-Help/Wireless-FSUWIN`

3.4.5 File Backups

Currently, we back up all user file systems a minimum of once a week. We keep a history of weekly backups for one semester. In addition, all files are backed up once per semester, and preserved for a minimum of one year. You are expected to make your own backups of important files (like your thesis or dissertation). Note that system backups are for disaster recovery. We will restore files from backup that were lost due to user error, but please do not abuse this privilege.

3.4.6 Printer Quota

Graduate students have a print quota of 200 laser printed pages per semester. If you run out, please follow the directions in the email you will receive from the printer subsystem. In short, you will ask your advisor to request an increase of your quota. It is no problem as long as you are using your quota for research, coursework, or teaching. Please do not print quizzes or tests in bulk on the lab printers. It is cheaper for the department when the photocopiers are used for bulk documents, rather than the lab printers. To see the number of pages you have printed during the current accounting period, visit the FSU Mathematics webpage Tech Support section, and click the Graduate Student Printer Quota link.

3.4.7 File Quota

All student accounts come with a modest file space quota. If more space is needed to support Math coursework or class teaching, it will be given. To request more file quota, email `root@math.fsu.edu` with an explanation of why you need it.

Large temporary filespace is available on our compute servers. Please contact `root@math.fsu.edu` if you need large amounts of scratch space for research. Note that scratch space is never backed up, as it is intended for large temporary datasets and intermediate files.

Chapter 4

Getting Started With UNIX

4.1 UNIX Wonderland

This is a brief introduction to UNIX. It is intended only to get you started. There are many books on UNIX available, and it is an excellent idea to get one[1].

UNIX is a huge operating system, but it has a fairly small number of unifying concepts which tie it all together. To be precise, UNIX is a multi-user, multi-tasking operating system. This means that multiple people can be using a particular machine, and that each of these people could be running multiple programs, all at the same time. It is not uncommon for one user to be simultaneously logged into more than one UNIX machine in a typical work session.

In our department, every staff and faculty member has a UNIX account, as do most graduate students. It is important to learn at least enough about UNIX to be comfortable with the command line. In addition, UNIX familiarity is a valuable job skill (and resumé entry).

4.2 What's In A Name

UNIX machines are almost always connected to other computers via a network. Thus, when working on a UNIX machine, you could potentially be using resources from computers all over the world (generally, you only use resources local to the department, but there are plenty of exceptions). To distinguish the various computers on the network, they are given names. The Mathematics department machines are generally named after deceased mathematicians (e.g. *hilbert*). While working within this department, only these short names are needed. If you wish to explicitly make use of a machine outside of our local networks, you would have to give a longer, more informative name. For example, to contact a machine named *prep* located at MIT's Artificial Intelligence lab, you would have to use its full Internet name, which happens to be *prep.ai.mit.edu*. The full Internet name of any FSU Mathematics machine will have the ending *.math.fsu.edu* (e.g. *hilbert.math.fsu.edu*).

[1] Contact the system staff for recommendations.

4.3 Logging in

UNIX is case sensitive. If you have a problem with logging in, make sure you have entered the lower and upper case characters of your username and password correctly, and that the `CAPS-LOCK` key has not been pressed (there is usually a little light on the keyboard itself that indicates that `CAPS-LOCK` mode is engaged, often near the upper right corner of the unit).

Remember, your password is also case sensitive. In Mathematics, all password will have a mixture of upper and lower case characters, so be sure that you pay attention to this when you are typing your password. If you make a mistake, simply hit the `RETURN` key until you are given the login prompt, and try again.

When you first receive your account, you will be issued your password. However, you can change your password at any time. There are numerous reasons why you should change your password occasionally. For example, if you keep your password written down on a piece of paper and then manage to lose it, immediately change your password (or contact the system staff if you cannot remember it at all). If you make a mistake and your password appears on the screen in the clear, you should change it if it was seen by anyone else. If you have the slightest suspicion that someone else knows your password, you should change it (and contact the system staff). To change your password, type `passwd` from any Sun in the grad lab, and follow the instructions on the screen.

4.4 Logging Out

To log out of a workstation, go to an area of the screen with no windows and holding down one of the mouse buttons until a menu with `LOGOUT` or `EXIT` appears. After doing so, take a moment to be sure that you have logged completely out of the system (you should wait to see the login prompt on a workstation or X terminal, just as before you logged in). All the computer security in the world is worthless if you leave yourself logged into a publicly accessible machine. If the machine seems to be stuck, and you are not sure if you are really logged out, get some help (this would be a reasonable use of the system pager number). If you are feeling a bit more advanced, you can log into another computer, then remotely log into the stuck machine (use `ssh`) and kill processes until it becomes unstuck.

It bears repeating: remember to log out! This is the most common mistake people make in our labs. At best, forgetting to log out ties up a workstation that someone else might want to use. At worst, your personal files and email are available for anyone to see.

4.5 Shells

In UNIX, most interaction with the system will be through a *command line interpreter* (CLI). In UNIX, CLI's are called *shells*. The basic function of a shell is to provide a *prompt* at which the user can type commands. The shell then executes the command or commands. After execution is complete the user is returned to the prompt, so more commands can be typed.

In Math, the default command prompt generally looks something like "`[17] hilbert:/opt/doc>`", where "hilbert" is the name of the machine upon which that shell is running, 17 indicates the number of commands executed in that particular shell since login, and "/opt/doc" is the current directory.

There are several shells to choose from in UNIX, but the default shell we give to our users is the *tcsh* (pronounced *tee-sea-shell*). This is a more feature-rich version of the much older and more ubiquitous *csh*. After logging in, you will want to open a terminal window (use the one labeled `xterm`). With such a window, you can issue UNIX commands. The *tcsh* gets its name from the ability to interpret *shell scripts* written in a language that closely resembles the C programming language. Shell scripts can perform complex or repetitious tasks in batch (or non-interactive) mode. Also, the *tcsh* gives the user *job control* capabilities, such as being able to execute several commands simultaneously. Learning a bit about whatever UNIX shell you use can save one a great deal of work. See the *csh* and *tcsh* manual pages for details (this is described fully in the manual page section), or contact the system staff for a recommendation on a book that will help you.

4.6 UNIX Files

One of the previously mentioned unifying concepts in UNIX is that of a *file*. On many computer systems, a file is something that contains some data, and exists on some sort of storage device (like a hard drive, or floppy disk). UNIX extends the meaning of a file to include just about everything. Physical devices (disks, printers, screens, keyboards, etc.) are all represented and manipulated through the use of files. There is even a file which represents nothing (*/dev/null*)! This, combined with the fact that almost all UNIX commands can use any file as input or output, provides a consistent way of working with virtually any device connected to our computer systems. This also has the side effect of reducing the number of command options one has to learn.

With that said, it is likely to be that case that most files that you will consciously work with will really be conventional files. For example, files that contain programming language source code (FORTRAN or C code, usually), files that contain documents that you are preparing for research purposes (like a thesis or dissertation), files that contain email, etc. It is important that you understand how to locate and refer to such files.

In UNIX, the location of all files and directories are in relation to the topmost directory, called the *root* (/)[2]. Thus, a *full pathname* (also called an *absolute pathname*) is one that begins with a "/", and indicates precisely where a file or directory is located.

In every directory, there are two files named "`.`" and "`..`". They can be thought of as "pointers" that point to the current directory, and the directory above (or *parent directory*), respectively [3]. Thus, a pathname like `../../notes/automata.stuff` would indicate moving up two directories, then going down into the directory `notes`, then referencing the file `automata.stuff`. Note that this pathname depends upon the current directory being two levels below the directory `notes`. This is called a *relative pathname*, because its meaning is relative to the current directory. If you are not in the correct place, a relative pathname will not work.

Note that in UNIX, one never needs to specify a drive letter, as in MSDOS. Instead, UNIX has one file tree (starting from "/"). Parts of this one file tree might be on different disks (and even different computers), but the user never has to be aware of this. Everything is referenced from the root directory.

[2]Notice that UNIX uses a forward slash, usually colocated with the "?" character on the keyboard, to separate the components of a pathname. MSDOS uses a backward slash. This can be a cause of frustration for MSDOS users that are learning UNIX.

[3]The root directory does not have a "`..`", for obvious reasons.

4.7 UNIX Directories

When you login, UNIX places you in your *home directory*. This is a place in the file system that is designated specifically for you. The actual name of this directory will be your username, and it will be located in a directory designated for home directories (in Math, these directories are named something like *home/m1*, *home/m2*, etc.).

To change into a directory, use the command *cd*. It works as one would expect, with one special characteristic. If you type `cd` with no arguments, you will be placed in your home directory. Also, the command *pwd* will display the name of the current directory. These two commands are crucial, as UNIX filesystems can contain hundreds of thousands of directories, and millions of distinct files. Thus, if you ever get lost, you can see where you are by typing *pwd*, and you can "go home" by typing *cd*.

4.8 UNIX Commands

UNIX commands can be very confusing at first. Most of them seem to perform some trivial task which in itself may not seem very useful. However, UNIX commands are designed to be used together. Seemingly trivial commands can be linked together to provide powerful functionality. This is a basic UNIX philosophy, that of providing simple tools from which more complex tools can be built. The way this is accomplished relates back to the idea of a UNIX file. Most UNIX commands take their input from the file *stdin* (which usually defaults to the keyboard), and will produce output to the file *stdout* (which usually defaults to the screen). By using a *pipe*, commands can be linked together so that the output of one command becomes the input to the next. We will look at an example of this later.

Note that most UNIX commands have a large number of options. These options control the way the command performs its function. Nobody can remember all of these options for all UNIX commands, which is why UNIX systems have extensive online manuals so such information can be looked up quickly (more on this later). The point to remember is that if you know of a command that *almost* does what you want (but not quite), it would be wise to look up the options for that command to see if there is one that will provide the functionality you desire. Even if such an option does not exist, the manual page might list some other commands that will do what you want. So, read the manual pages!

4.8.1 Listing Files and Directories

When you login, you will be automatically be placed in your home directory. To get a listing of the files and directories contained within, just type

```
ls
```

To list the contents of another directory, you can either `cd` to it and type `ls`, or you can simply supply the directory name as an argument. For example:

```
ls /etc
```

Would list the contents of the directory `/etc`. Similarly,

```
ls ..
```

Would list the contents of the directory above the one in which you currently reside. You can list the contents of any directory for which you have read permission.

A useful option for `ls` is `-l`. It produces a long report which gives more information about the individual files. By default, `ls` does not list files whose names begin with a ".". Files whose names begin with a dot are usually configuration files that are not frequently changed, so they are not normally listed to reduce screen clutter. The option `-a` overrides this behavior, and will produce a listing of every file in the directory. As with most commands, more than one option can be present. Thus, the command `ls -la` will produce a long listing of every file in the directory.

You will note that UNIX allows significant flexibility in naming files. Most characters can be used in filenames, and they can be quite long. You can even use spaces in a file name, although this will often require you to surround the name in quotation marks when you refer to it (e.g. `rm "this is one file"`). If you accidentally create a filename beginning with a minus sign, see the man page for `rm` for instructions on how to remove it.

4.8.2 Erasing files

The command *rm* is used to erase files and directories. It is an extremely powerful command that can be unforgiving to the novice user. Be very leery of using the *rm* command with any wildcards[4] For example, this command would erase all files ending in ".o", and ".out":

```
rm *.o *.out
```

But the command:

```
rm *.o * .out
```

Will cheerfully wipe out all the files in the directory (notice the extra space)! When used with the `-r` option, it can do this *recursively* to all files and directories in the current directory, so be careful. This recursive option can also be very useful in deleting entire subtrees consisting of both files and other directories.

By default, new users to our system will have `rm` set up in such a way as to force interactive mode. This means that `rm` will ask if you really wish to erase each file. Users will often remove this restriction after gaining a bit of experience. If you decide to do this, be warned that `rm` erases the files you tell it to erase, which are not always the ones you really wanted erased.

You can remove a directory that you own with the `rmdir` command. It must be empty, so you must first go in and delete its contents with the `rm` command (and possibly `rmdir` if it has sub-directories). Use the `ls -a` command to make sure you have removed any hidden files.

[4]Wildcards are characters that can match more than one thing. For example, `ans*` would match all things beginning with the string `ans`.

4.8.3 Creating a directory

If you are in a directory for which you have write permission, you can create one or more subdirectories using the `mkdir` command. Examples:

```
mkdir thesis
mkdir private-stuff Letters milkdud
```

4.9 File manipulation

Most of the operating system commands in UNIX manipulate files; some of the most common ones are:

`more`	Display text file(s) on the screen, pausing for each page
`lpr`	Print file(s) to a printer
`cp`	Copy file(s)
`mv`	Move file(s)
`chmod`	Change permissions of file(s)

4.9.1 Viewing files

The command *more* will display text files or output one video page at a time. Use the space bar to move on to the next page; use the return key to scroll one line at a time. There are a bunch of other useful functions built into *more*, such as searching for the next occurrence of a string, and scrolling backward. These other functions are listed in the manual page for *more*[5]. Note that many programs will automatically use *more* to display files to the user (for example, email is usually displayed to the user with *more*). You can use *more* to conveniently display the textual output of just about any command or string of commands (see the last example below).

Examples:

```
more /etc/motd
more classnotes-spring01.txt
ls -l | more
```

4.9.2 Printing files

The *lpr* command is used to print a file or files on a printer. Like UNIX machines, UNIX printers have names. The laser printer named *lw* is intended for use by graduate students needing high quality printouts, and is located in Grad Lab 2. Examples:

```
lpr -Plw dissertation.ps
lpr -Plw slimemold.c foxtrot.h
```

[5]Manual pages will be discussed in depth later in the document.

Various options can be set to change the way the files are printed, or to enable the printing of special files. See the manual page **lpr** for a list of print options. Also, see the web page entry about printers in the Mathematics department. Read these before attempting to print anything!

If you ever wish to cancel a print job, use the *lprm* command. It will allow you to remove any jobs you have submitted to a printer. To use it, first run *lpq* to see a listing of jobs. Find yours, and note its job number. Then, use the *lprm* command to remove that job number. Example:

`lprm -Plw` *yourjobnumber*

4.9.3 Copying, Moving, and Renaming Files

You can copy a file with the *cp* command. You can move a file or a directory with the *mv* command. You can also use these commands recursively, to copy and move directories (see their man pages for details). Note that the *mv* command is also used to rename files (by "moving" them to their new name). Examples:

```
cp .login .login.backup
mv first-try.c bad-idea.c
mv current-version/* ./old-stuff
```

4.9.4 Changing the permissions of files

All UNIX files have *permissions* associated with them. These permissions define who can do what with any particular file or directory. You can see these permissions by typing `ls -l` *filename*. By default, these permissions are set so the owner of the file can do anything, and any other user can look at (and perhaps execute) the file, but cannot modify or delete it. This is fairly open, so if you have files that you wish to be private (files containing class quizzes and tests are really good candidates for this) you will have to change permissions. Also, if you give execute permissions to a text file, it will act as a *script* (or *batch*) file. Here are some common examples (read the *chmod* manual page for a complete explanation):

`chmod 700 mydir` : Make `mydir` private, only the owner can enter it (so all files inside are also protected).
`chmod 755 mydir` : Make `mydir` readable by anyone.
`chmod 600 test3-fall95.tex` : Make `test3-fall95.tex` private, only the owner can read it.
`chmod 644 group-tasks` : Make `group-tasks` readable by anyone.

Note that if you make a directory unreadable by others, then any files or directories contained within are also protected. This is a good way to organize your home directory. It should consist mainly of other directories, some of which are private, others not. Whenever you start a new project, you should create a new directory for it (perhaps in some other directory, to further organize things). Then, if you decide that the files should be private, you can do so easily.

4.10 Online Documentation

As stated previously, UNIX depends heavily on online manual pages. The use of these manuals (henceforth referred to as *man pages*) will enable you to use UNIX more effectively.

The command `man` *commandname* will display the man page for that command. By default, *more* will be used to display it, so you can use any of the movement and searching options it offers. Man pages are organized in a standard format. In the "NAME" section, the name of the command and a one line description is displayed. The "SYNOPSIS" section shows all of the possible options, and exactly how the command expects to be invoked (order of arguments, etc.). The "DESCRIPTION" is a verbose description of the command. The "OPTIONS" section is a verbose description of all command line options (probably the most consulted part of the man page). There are several other sections describing what files the command uses, known bugs, and other information. Perhaps the most useful is the "SEE ALSO" section. This section lists the names of commands that are related to this one. Using this information, you can often "chain" through several man pages to get to the one you want. Thus, if you can remember a command that is similar to what you want, look at the "SEE ALSO" section in its man page[6].

A good place to start with man pages is to type `man man`, which will display the man page for *man* itself. Man pages are written using a lot of UNIX terminology, and are sometimes deemed useless by novice users because of this. Stick with them and one day the proverbial light bulb will ignite and they will actually make sense. This large amount of specific terminology is probably responsible for UNIX's reputation of being "confusing" or "complicated". UNIX is really quite elegant after the basics are learned and one understands a few UNIX terms.

4.11 Changing your password

Use the command *passwd* from any Math Sun computer to change your password. Just type `passwd`, and follow the directions on the screen. If you fail to enter a valid password, the program will allow you to try again. If you change your mind, you can hit `CONTROL-c` (the program will abort and will not change your password). Please maintain a strong password. Strong passwords are the best way you can protect yourself and the other users of our systems from malicious activities.

[6] Remember, you can use the search feature of `more` to quickly leap to different sections of a man page.

Chapter 5

Text Editors

There are three full screen textually based editors available on our machines: *vi* (pronounced vee-eye), *GNU Emacs*, and *pico*. *Vi* is included with all UNIXs. It is a full screen version of a very old line editor. It has an extremely weird interface. It is not user friendly. On the other hand, it is always there no matter what UNIX machine you are using. It is also a pure text editor, with no bells and whistles. As a result, it is small and fast. *Emacs* is a much more than just a text editor. It has more functionality than *vi*, and is more polished. *Emacs* is not a standard part of UNIX, but most sites have it due to its enormous popularity. The majority of users at this department use *vi*[1]. Both of these editors have versions available for just about every kind of computer on the planet, personal computers included[2]. The last editor, *pico* is specifically designed to be easy to use. It is not as powerful as the other editors, but it takes almost no time to be able to use it. It is the default editor for new accounts.

5.1 vi

Vi is a member of the family of text editors originating from the ancient line editor *ex*. The interface is quite strange, but an experienced *vi* user can perform complicated editing commands with a small number of keystrokes. *Vi* commands demand less hand movement than *emacs* commands, and much less modifier use (i.e. holding down the Control key).

5.1.1 Running *vi*

Vi is invoked by typing `vi` *filename(s)*. This will clear your screen and display the first file if it exists, or will create the file if it does not. If you make a mistake when typing a filename (and will therefore be creating an unwanted file), simply exit without saving and *vi* will remove all traces of the file. When creating a file, you will see a column of tilde characters at the left of the screen. These represent lines of "nothing" (as opposed to blank lines). They can also be seen when viewing the bottom of an existing file.

[1] Text editor choice can be a touchy subject with UNIX people. This is strange.

[2] Contact the system group if you want to get a version for your home PC or Mac.

5.1.2 Vi tutorial

There is a printed tutorial for *vi* in manual set within the grad lab (there are other copies available to staff and faculty members). The best way to learn vi is to go through a good tutorial. There are many available via the web, so do a search and pick one you like.

5.1.3 Aborting a *vi* session

If you enter *vi* by mistake, you can get out by pressing the ESCAPE key several times, then typing "`:q!`". This will exit without saving any changes.

5.2 GNU Emacs (emacs)

Emacs (Editing MACroS) is essentially a complete interpreted LISP environment which happens to be running a text editor program written in LISP. Thus, the full GNU Emacs implementation is much more than a text editor. There are many "programs" available for this LISP environment which extend the functionality of EMACS. You can write such programs yourself (in fact, advanced EMACS users can do amazingly complicated edit operations with just a line or two of LISP). It also provides multiple text-based windows in a single session. There are several special modes in emacs, many of which are entered automatically based on the file name. For example, when editing a C file (ending with ".c"), there are commands available that will properly indent the lines to show code blocks (this feature is very useful when debugging control flow).

5.2.1 Starting Emacs

You invoke **Emacs** on Sun workstations by typing `emacs`. Typically, you will want to edit just one file in a session. For this, you can type `emacs` *filename*, and the file chosen will be loaded into the editor.

If you ever have difficulties starting Emacs, try using the command `emacs -nw` *filename*. This instructs Emacs to not attempt to use features specific to X Windows. The origin of this occasional problem has to do with the existence of the DISPLAY environmental variable. Thus, if you have been experimenting with X Windows, and *emacs* does not seem to work correctly, use the command above.

5.2.2 Emacs Tutorial

Emacs has a built-in interactive tutorial for new users. To invoke it, run *emacs* and type `CONTROL-h t` (that is, `CONTROL-h` followed by a plain "t"). This tutorial will show you how to move the cursor around, scroll through the file, insert and delete text, and much more. It is the best way to learn *emacs*. Note that Emacs has a printed manual that is over two inches thick, but the text of that entire manual is available online via a hypertext information reader within Emacs. To use this, type `CONTROL-h i` (do the tutorial first). Also, there is a reference card available in TeX format located at `/usr/local/emacs/etc/refcard.tex`. It also exists in PostScript format in `/usr/local/doc/refcard.ps`. It is only 2 pages long, and is very handy for *emacs* use. Print out a copy and keep it close!

5.2.3 Getting yourself out of weird EMACS modes

In *emacs*, is very simple to enter extraordinary states that bear no relation to common realities. You can back out of most of them by typing `CONTROL-g` several times. Use this whenever you have entered some strange command that you did not mean to type.

5.2.4 Writing your work to a file

Press `CONTROL-x`, then `CONTROL-s`. This writes out the buffer (what you are working on) to the file. On some terminals, this will also freeze the cursor. If everything seems to stop working after a `CONTROL-x CONTROL-s`, type `CONTROL-q`, which will reactivate the cursor.

5.2.5 Getting Help

In *emacs*, you can get help with `CONTROL-h`. You can repeat `CONTROL-h` several times, with increasing amounts of help with each repetition.

5.2.6 Ending a session

Press `CONTROL-x`, then `CONTROL-c`. This will cause *emacs* to exit. It will ask if you want to save any modified buffers.

5.3 Pico

Pico is almost completely self explanatory. Run it by typing `pico` *filename*, as you would any text editor. The bottom of the screen lists common commands. A help page of all commands available can be seen by pressing `CONTROL-g`. Exit by pressing `CONTROL-x`.

Chapter 6

Document Preparation

Many users in the Mathematics department use TEX to produce publication quality documents[1]. TEX is a programming language designed to specify printed output. It is particularly suited for scientific papers, as it supports most types of mathematical notation. It is NOT a WYSIWYG package. You literally write a program to describe what you want, then "compile" it with TEX to produce a *dvi* file. This file can be printed by using *lpr* with the `-d` option. It can also be previewed graphically with several tools[2].

There are many tools available to create graphs, figures (for papers), and other graphical drawings. See the software descriptions located in the Tech Support section of the Math website for a listing of these utilities.

[1] This document is written in LATEX, which is a set of TEX macros.
[2] Use a previewer like `xdvi`! It will save both time and your printer quota.

Chapter 7

Available Software

The Mathematics system group installs new software on a continuous basis. Some of these are upgrades to current packages, others are new packages that seem interesting and useful. To see a brief description of most of this software, visit the FSU Mathematics webpage Tech Support section, and click on the UNIX Software By Category link. Note that the software listed there does not represent everything we have[1]. However, it will give you a good overview. If you have a need or desire for some particular kind of software that we don't have, let us know!

Of course, the best way to see what we have is to explore. Most of our software is located in a fairly small number of directories, and virtually everything has man pages. Have fun!

[1] We don't know everything we have, and we put it there!