# Appendix B

Using the Suns at Lamont

B.1 Getting started: Logging in

Find any vacant (ha!) Sun Workstation that you are allowed to log into for free, and type your username at the login prompt. Your username for this course will be of the form `mm_??_00` the question marks relate to your name (my userid is `mm_ms`). The machine will then prompt for your password (but not reflect it). At the moment you can choose one of two windowing systems. I prefer CDE (Common Desktop Environment) which is becoming standard(ish) among Unix vendors. If you've chosen this and are logging in for the first time, your screen should look something like.
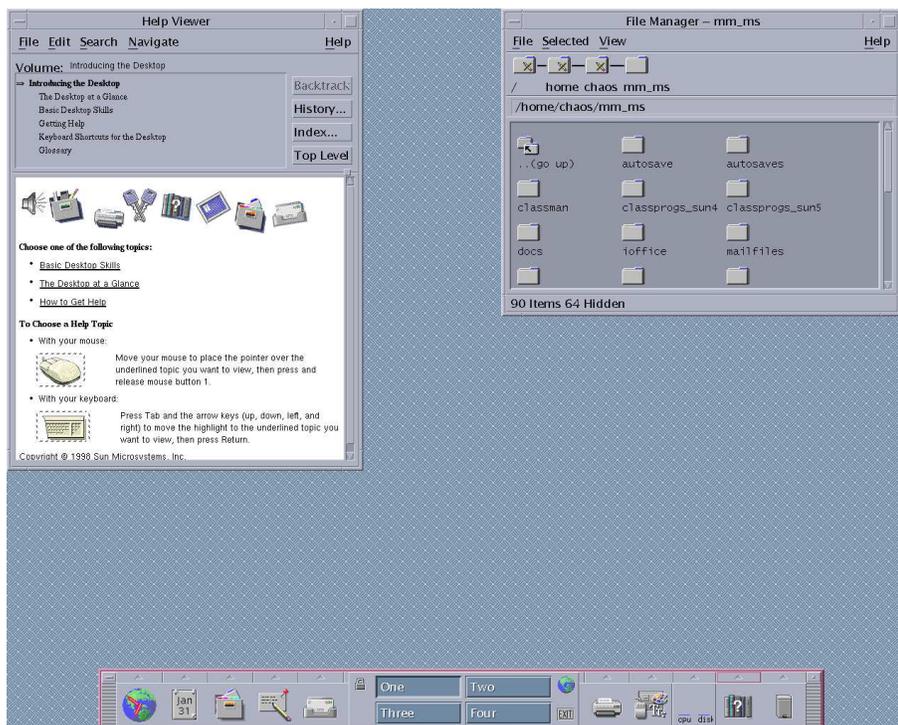
Changing your password   If you are a first time user for this account you will have been issued a temporary password. You should change this pronto. First you will need to pop up a command window. To do this, move the cursor over the background and hold down the right button. This will produce the *root menu*. Go to programs and choose terminal (and while you're at it, pop up a Console). Now go to the terminal window type `passwd`. You will be prompted for your current password, then asked for your new passwd twice just to make sure. None of the typing will be echoed to the screen. For security passwords should be at least 6 letters long and contain some funky characters like !#$% or numbers. Do not use proper names, common words, birthdays, etc. If all goes well, your should see the following

```
2-mingus% passwd
Changing NIS password for mm_ms on chaos.
Old password:
New password:
Retype new password:
NIS entry changed on chaos
```

Getting around OpenWindows/CDE   The CCE user-interface is a typical move-click-and-drag mouse driven system. It is similar to the macs with the principal distinction of having a 3 button mouse and no obvious menu bar. The initial help screen has 45 tedious pages of how to get around CDE most of which is pretty obvious. The most important thing to know is that the right button of the mouse will pop-up any appropriate menus for whatever application the mouse is pointing at. The most important menu is found on the root window (the background). This *Root Menu* allows you access to commonly used programs, let you set and save the properties of your workspace, calls up the help viewer and most importantly lets you exit the windowing system. In addition, CDE also has a bottom toolbar with many of the options of the rootmenu on it and it allows you to change your workspace. The left mouse button is for selecting things, double clicking and highlighting text. The best way to get a feel for all these widgets and buttons is to sit down and play and use the annoying helpviewer if you need to. **Remember:** The best way to learn this stuff is to poke and play at it. There is nothing you can break so go wild. For practice, a good thing to do at this point is to use the *Desktop Style* option on the toolbar to configure your workspace to your liking.

Getting around the filesystems   Once you're comfortable mousing around, the next important task is to maneuver around the file system, change directories and deal with files. For this you have two options

1. Use the file manager. The file manager is a Mac-like interface with folders, files etc. It can be useful and is a good way to get a feel for the dimensions of the lamont filesystem but it is not a substitute for learning Unix.

2. Learn Unix. Go to the cmdtool and start hacking, or better yet, fire up XEmacs (type `xe`) and start the shell from there.

B.2   Basic Unix Concepts and Commands

Filenames and directories The basic working unit in Unix are files and directories (actually directories are just special files). Files and directories all have names which can be of any length but should have *no spaces* and should avoid the special characters !$*. Some common filename conventions and examples are

```
Articles        News            classprogs_sun4 intro.tex
Handouts        class.cshrc     dead.letter     picture_0.ps
```

Note: Unix file names are case sensitive so `news` and `News` are two different directories. The directory structure under Unix is a simple inverted tree starting with the root directory and continuing down different branch directories (e.g. my home directory is `//Users/mspieg/mspieg/mm_ms`) with the individual files at the bottom. A few special directories are

. the current directory

.. the parent directory (one directory up closer to the root)

~`userid` the home directory of a user called userid (e.g.`˜mspieg`)

~ your home directory

Commands and the Manual pages The following sections will outline most of the common commands for maneuvering and manipulating files and directories and doing just about everything. An example session of usage will be included at the end.

Commands are typed at the unix prompt usually followed by arguments and files to be operated on. The following list is organized by concepts. This list is not complete and does not supply all the possible thousand permutations of command line arguments. To get more information about any command, the most important command is `man` which will list the manuals page describing everything you can do with a command. Examples include

`man ls` Give us information about the command ls (list files)

`man -k gzip` Give a short description of any command that deals with file compression (actually, any command whose description contains the string compress)

man in Xemacs in XEmacs, a more convenient way to access the man pages is under the Help menu under manuals (or just type `M-x manual-entry`).

Another important source of online help is the AnswerBook, which contains the entire Sun documentation set online (with pictures and everything). Answerbook can be searched for any topic (not always successfully) and has gobs of information on Openwindows, basic UNIX, Fortran (f77 and f90), configuring your environment etc. You can start the answerbook from the toolbar library menu. When in doubt

1. RTFM

2. Ask somebody who might know (other students are particularly good resources).

Self Awareness Commands

whoami returns your userid (useful when your confused)

pwd returns the current directory you are in

hostname returns the name of the machine you are on

Moving around Directories

cd change directory, some examples

```
cd
cd ..
cd ~mm_ms
cd /chaos/chaos/magma/mspieg
```

ls list the contents of a directory, thousands of options

lf list the contents of a directory with file information (alias of ls -F)

llf long listing with file information (alias of ls -lF)

mkdir make a subdirectory in the current directory (`mkdir crud`)

rmdir remove an empty directory (`rmdir crud`)

Doing (unspeakable) things to any file

cp copy files to other files or files to directories, examples

```
cp file1 file2
cp file1 directory
cp file1 file2... filen directory
```

mv move or rename files. essentially a copy then delete. Same syntax as cp

rm REMOVE files. Careful with this one, rm is forever.

chmod change the read-write-execute permission of a file

Doing (unspeakable) things to text files

cat spit out a file to standard out, can also be used to concatenate several files into one file (see the section on pipes)

more see the contents of a file, one page at a time `more file.txt`.

textedit open up a simple mouse driven texteditor (aliased to te) `textedit file.txt`

vi native Unix texteditor, not mouse driven. Useful for dumb terminals but otherwise a steep learning curve

emacs another editor. A favourite among real Unix hacks but not immediately intuitive.

xemacs A much more beautiful version of Emacs for X-Windows. Lots of lovely buttons and menus for keystroke-impaired (highly recommended)

lpr send a file to the printer (usually a laserwriter)

enscript format a beautiful file and send to the printer

grep Very powerful tool for finding strings in textfiles and printing out matching lines (see examples)

Programming Tools

f77/f90 the fortran compiler

cc/gcc the C compiler

make a very useful tool for building multi-file programs

Workshop An interactive debugging and programming environment that can be run from within XEmacs.

X and remote machines in Unix and OpenWindows you can compute on another machine across the planet or network just as easily (but not as quickly) as the one your sitting in front of. To access a remote machine use

xhost on your local machine to allow remote X access (e.g. `xhost +`)

rlogin/telnet to remote login to another machine `rlogin ouzel`

xsetd set your X display to your local machine `xsetd host` (Not a real unix command just my alias for `setenv DISPLAY host:0`.

File transfer and the Internet There is a whole planet of machines and programs living out on the net. Some useful commands for connecting to outside machines and transferring files are

telnet connect to a remote machine on the internet (or locally) `telnet columbianet` or `telnet phx.cam.ac.uk`

ssh If available, better than telnet. starts the secure shell which will allow encrypted telnet and Xsessions. It will automagically set up the proper displays. ssh can be slow but it protects you against password sniffing etc.

ftp file transfer protocol, get and send files directly to other machines. Most useful as *anonymous ftp* (see xarchie) for searching for freely available software and information.

Netscape Okay, if you don't know what this is by now I have a brooklyn.bridge.com to sell to you....

12

Panic Buttons and Bailing out Quite often things will go wrong, programs hang up or you just want to get out of something. There are a few Control sequences which you should know. These are often abbreviated with a caret or uparrow character for the control key. Thus Control-c means hold down the control key and type C. emacs shorthand for Control-c is C-c. Useful panic buttons and commands are

> **Control-c** Exit the current program (within emacs shell use C-c C-c)
>
> **Control-z** Stop the current program (C-c C-z)
>
> **Control-d** Leave a cmdtool or shell (C-c C-d)
>
> **logout** Leave a cmdtool or shell
>
> **kill** kill a specified process (you need to find out the process id)
>
> **killproc** kill a process by name `killproc text` will kill any process that contains the string text. **This is not a unix command but a local program I wrote** (do a `more   ˜mm_ms/classprogs/sun4/killproc` to see some real ugly shell programming).

If your whole workstation hangs up, don't panic just find another workstation, rlogin to the hung one and start killing processes selectively. If you're really panicked `killproc csh` Will blow everything back to the login prompt.

Spiegelman Specials many commands can be abbreviated, aliased, or included in *scripts* which are just files of commands. A few of these combined commands that I find particularly useful have been included for the class through the class.cshrc file. Just for reference a few of these are

> **lf** alias of ls-F
>
> **pun** find a process by name. E.g. `pun csh` will find all processed with the string csh. `pun mspieg` will find all process owned by mspieg.
>
> **te** run texteditor in the background
>
> **xe** run xemacs in the background
>
> **xcnet** pop an X-term running columbianet
>
> **gv** run the other postscript viewer, ghostview in the background
>
> **tidy** removes the texteditor and emacs backup files (they end with a % or ˜)
>
> **wpr** find out what the current default printer is (alias printenv PRINTER)
>
> **spr** set the default printer `spr cookie_lw`
>
> **xsetd** set the X display
>
> **killproc** kill processes by name
>
> **crexp** do a global search and replace for a single pair of strings. E.g. `crexp string1 string2 file1 file2...filen` will replace every occurence of string1 with string2 in files 1 to n
>
> **getnr** get numerical recipes (fortran) routines.

B.3   Additional commands and programs

This section outlines just some of the programs available at Lamont. They have been organized into three areas.

CDE Tools These are window based tools that come as part of CDE. Most of these are accessible from the root menu under Programs or the bottom toolbar.

> **Command Tool** a scrollable command line interface
>
> **Text Editor** a simple mouse driven text editor (not a word processor)
>
> **File Manager** a mac-like file system browser and controller
>
> **Mail Tool** an intuitive interface for e-mail (but also see VM in xemacs or the netscape mailers).
>
> **Calculator** a reasonable scientific calculator
>
> **Perfmeter** a graphical tool to show how much CPU you're burning
>
> **audiotool** listen to sound files on cheesy speakers
>
> **imagetool** a postscript and image previewer (I prefer gv and xv).
>
> **Games, Demos, and other things** Lousy games for such an expensive machine but check out the root menu to see everything else. Spider is a good but hard solitaire game. (XEmacs has tetris though).

Freeware at Lamont These are general utilities and programs that have been installed at lamont. These programs are of variable quality and reliability but they can be very useful.

> **xemacs** The kitchen sink of free-ware... Have I said how much I like this program? Reads your news, writes and debugs your code, washes the dishes and gives you a healthy coat and shiny teeth.
>
> **StarOffice** A rather remarkable, *free* MicroSt office clone that runs under Unix/Linux/you-name-it. Has a particularly nice spreadsheet, much like Excel. To install type soffice at the command line and follow directions. Once installed it will show up in the toolbar.
>
> **acroread** Free Adobe Acrobat reader for PDF's
>
> **rn, xvnews, vnews** Various news readers (xvnews is X-window based, see also Gnus in xemacs and Netscape News)
>
> **latex, tex** Donald Knuth's super duper technical typesetting program (also runs nicely under emacs)
>
> **Netscape** Web-o-rama
>
> **xmgrace** a free WYSIWYG xy plotting program
>
> **xfig** a free drawing program, also allows you to annotate postscript. A little clunky but useful for quick figures and posters.
>
> **gmtsystem** a script driven postscript and mapmaking toolkit
>
> **XGB** Geobase – Bill Menke's graphical database browser (X-Version)

ahsystem Lamonts time-series filters (probably obsolete by now)

pbm toolkit A suite of simple *filters* for image processing

xv an interactive image viewer

xanim A useful animation viewer (gifs, mpgs, quicktime)

gimp Gnus Image Processing system, free photoshop...way cool.

emacs GNU (Gnu's Not Unix) editor environment

gcc/g77/g++ GNU's free c, fortran and C++ compilers

gzip, gunzip, gzcat GNU's free file compression/ uncompression tools

and much much more... look in `/usr/local/bin` for just a sampling.

Licensed Software at Lamont This software is commercial, slick and expensive. It also comes with the nasty baggage of floating licenses where anybody at lamont can run the programs from any SunWorkstation; however, if there are only two licenses, only two people can run it simultaneously. **SO DON'T LEAVE THEM OPEN IF YOU'RE NOT USING THEM**. Some of the more useful software is

matlab MathWorks interactive system, particularly good for signal processing and numerical linear algebra and practically everything. Extremely useful for quick prototyping and integrating calculations with graphics. Many licenses I believe.

mathematica Symbolic math and graphics package. Way cool if you can figure out the peculiar syntax.

avs Advanced Visual Systems program for way-cool 3-D visualization and computation. Very flexible and extendible. **4 licenses** needs at least a color monitor and is very cpu and memory intensive

dx IBM's version of avs. Available on the SP2

acrobat/distill Unix version of acrobat for Suns. **2 licenses**. `acroexch` allows manipulation of pdf files. `distill` is very useful for making excellent pdf files from postscript (and excellent postscript from pdf files).

xframe Frame Technology's Framemaker. WYSIWYG textprocessor and formatter.

ermapper Real slick, image processing environment. Only has a student license (talk to Jeff Weissel or Chris Small if you need more info).

envi More image processing goodies in the Visualization lab.

B.4   Shortcuts and the csh

Unix is more than a collection of commands it's also a philosophy (ooooooh, deeeeeeeeep) that by combining lots of simple tools that do just one thing, you can build custom tools that do anything. There are therefore, some shortcuts and

useful structures you can use to do all kinds of things. The `csh` (pronounced C-Shell - or seashell), sits between you and the computer and interprets commands. Most of these commands and shortcuts use certain special characters. For a full understanding of the power of the csh, look at `man csh` or answerbook. On a day to day basis however there are a few csh shortcuts that will make your typing faster and your commands immensely cryptic. (Alternatively you can run the shell from within xemacs which has some handy filename completion and history functions that are great).

Filename completion and wildcards Any filename can be extended using the two wildcard characters `*` and `?`. The csh will expand `string*` to any filename that starts with string and has any number of characters that follow. `*string*` matches any file that contains string. `*` matches all files. `?` works the same way but only expands to single characters. Examples

```
lf *.tex
grep howdy *.tex
rm * (DON'T EVER DO THIS WITHOUT A GOOD REASON)
more file_??.ps
```

history substitution the csh keeps a record of your last 25 commands (this number can be changed). Typing h (alias to `history`) gives you a list of these commands and a number. Using the `!` character you can rerun these commands quickly. Important examples

!! Run the previous command

!21 run command number 21

!more run the most recent command that contained the string more

If you are running the shell under xemacs, it has a much nicer history function which you can access from either the *Complete menu* or by using Control and arrow keys.

pipes and redirecting input and output A standard unix command reads a file from its standard input and writes it to it's standard output (usually the terminal). This *data stream* can also be redirected into files or other programs so that the output of one program becomes the input to another. Thus very powerful processing filters can be built out of simple tools. The principal symbols for redirection are the pipe symbol $|$ and the to/from file symbols $<, >$ for directing output from and to a file and $>>$ to append to a file. Some examples

`gzcat file.gz|more` uncompress the compressed file `file.gz` and pipe the output through more just to see the thing one page at a time

`gzcat file.ps.gz |lpr` send the postscript file to a printer

`gzcat problems.tar.gz |tar -xvf -` uncompress and unpack a tar (tape archive) file. We will use this a lot.

> myprogram < input > output run a program and take its standard in-
> put from file `input` and write it's standard output to file `output`.

quotes single and double quotes in unix have different meanings and can be used
to lump strings together and protect commands from file expansion by the
shell. loads of fun

B.5   A typical unix session: sort of old now

```
1-mingus% whoami
mm_ms
2-mingus% cd
3-mingus% pwd
/tmp_mnt/data/sarah/mm_ms
4-mingus% lf
Articles  class.cshrc  mout
Handouts/  classprogs_sun4/ mylog.uu
News/   dead.letter  unix/
5-mingus% more class.cshrc
##################################################################
#   Greetings:  This is the class .cshrc file for Myth's and
# Methods in Modeling (mandms)
# This file will help get you all set up to do the homeworks and
# use some of the available tools.  If this file is included in
# your personal .cshrc file,  any changes and additions for the class
# will automatically be added to your environment.  Have fun
##################################################################

######################
#  set up TeX Environment for interactive shells
######################
if (  { tty -s } ) then
source /lamont/scratch/mspieg/tex/tex.cshrc2
endif

######################
#  set up GMT Environment
######################
gmtenv

######################
#  set up  path for shared course programs
######################
setenv CLASSBIN /data/sarah/mm_ms/classprogs_sun4
setenv MYBIN ${HOME}/${ARCH}
set path= ( $path $MYBIN $CLASSBIN )


######################
#  set up  default permissions (user+group=read,write,execute others read+x)
######################
umask 002


######################
#  Useful (?) aliases
######################
alias cp cp -i         #  check before overwriting a file
alias mv mv -i         #  check before overwriting a file
alias hg 'history | grep \!* | grep -v hg' #  check history by name
alias lf 'ls -F'       #  show filetype in short listing
alias llf 'ls -lF'       #  show filetype in long listing
alias pun  'ps -aux | grep \!* | grep -v grep'  # see processes by name
```

```
alias pv 'pageview \!* &'      #  run pageview in the background
alias te 'textedit \!* &'      #  run textedit in the background
alias tidy 'rm *%'             #  clean up texteditor backup files
alias spr 'setenv PRINTER \!*'    #  set a printer by name
alias wpr 'printenv PRINTER '     #  see what printer is set
alias xsetd 'setenv DISPLAY {\!$}:0'    #  set your X display by hostname

#######################
#   configure opewindows default layout
#######################

if ( ! -e $HOME/.openwin-init ) then
echo sorting out your windows
cp ~mm_ms/.setup/.??* $HOME
endif
6-mingus% lf
Articles  class.cshrc  mout
Handouts/  classprogs_sun4/ mylog.uu
News/   dead.letter  unix/
7-mingus% cd Handouts
8-mingus% lf
Conserv1.ps.Z Conserveq2.ps.Z Intro.ps.Z Vectorcalc.ps.Z problems1.ps.Z
9-mingus% zcat problems1.ps | head
%!PS-Adobe-2.0
%%Creator: dvips 5.47 Copyright 1986-91 Radical Eye Software
%%Title: problems.dvi
%%Pages: 2 1
%%BoundingBox: 0 0 612 792
%%EndComments
%%BeginProcSet: tex.pro
/TeXDict 200 dict def TeXDict begin /N /def load def /B{bind def}N /S /exch
load def /X{S N}B /TR /translate load N /isls false N /vsize 10 N /@rigin{
isls[[0 1 -1 0 0 0]concat}if 72 Resolution div 72 VResolution div neg scale
10-mingus% zcat problems1.ps | pv -
[1] 12660 12661
11-mingus% cd ..
12-mingus% lf
Articles  class.cshrc  mout
Handouts/  classprogs_sun4/ mylog.uu
News/   dead.letter  unix/
13-mingus% te dead.letter
[1] 12664
14-mingus% mkdir crud
15-mingus% lf
Articles  classprogs_sun4/ mylog.uu
Handouts/  crud/   unix/
News/   dead.letter
class.cshrc  mout
16-mingus% cp dead.letter crud
17-mingus% cd crud
18-mingus% lf
dead.letter
19-mingus% cp * junk.letter
20-mingus% lf
dead.letter junk.letter
21-mingus% rm junk*
22-mingus% h
     1 whoami
     2 cd
     3 pwd
     4 lf
     5 more class.cshrc
     6 lf
     7 cd Handouts
     8 lf
     9 zcat problems1.ps | head
    10 zcat problems1.ps | pv -
    11 cd ..
    12 lf
    13 te dead.letter
```

```
      14 mkdir crud
      15 lf
      16 cp dead.letter crud
      17 cd crud
      18 lf
      19 cp * junk.letter
      20 lf
      21 rm junk*
      22 h
23-mingus% hg junk
      19 cp * junk.letter
      21 rm junk*
24-mingus% cp ~mspieg/course/Unix.tex .
```