

Introduction to the School of Earth Sciences UNIX computer network

Pandora Hope

February 2000

Revised and extended by:

Kevin Keay
February 2006

Contents

1	What is UNIX?	1
1.1	The School of Earth Sciences UNIX network	1
1.2	Logging on to the network	2
2	Some UNIX Commands	3
2.1	Manual pages	3
2.2	Navigation	3
2.3	Moving files	4
2.4	Permissions	5
2.5	Viewing files	5
2.6	A few short cuts	5
2.7	Finding things and directing output	6
2.8	Changing your password	7
2.9	Time and date	7
2.10	Checking up on people	7
2.11	Compiling	7
3	Directory Structure	8
4	Editing and Printing	8
5	Running Jobs	10
6	Data and Graphics formats	11
7	Scripts	12
8	External Stuff	13
9	Dot files - controlling your own set-up	14
10	Miscellaneous	15
	Appendix A Accessing UNIX from a Windows PC	17
	Appendix B Accessing UNIX from an iMac	22
	Appendix C An Introductory Tutorial on UNIX	24

1 What is UNIX?

UNIX is an operating system that drives the hardware of the computer. Each computer company that makes UNIX machines has their own 'flavour'. Historically we have relied on the Solaris variant of SUN Microsystems but in recent years our network has expanded to encompass PCs running Linux, iMacs running X Darwin and Windows PCs with a UNIX emulation known as Cygwin. On top of the operating system are extra commands and shells to help you communicate with it. You are using the TC shell (`tcsh`) at the command line. This is an extension of the older C shell (`csh`) and includes such features as tab completion of filenames. There are also graphical windows utilities that are broadly referred to as X Windows e.g. `display`, `nedit`.

For additional help, please ask Brett Holman or Doug Morrison (support@earthsci.unimelb.edu.au), Kevin Keay (keay@unimelb.edu.au) or any of the Meteorology postgraduate students. A very useful and concise reference book covering UNIX and the C shell is:

Unix in a Nutshell, O'Reilly and Associates (Engineering Library: 005.43 GILL).

A useful online resource from the University of Surrey, *UNIX Tutorial for Beginners*, is available at:

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

The above webpage includes a link to download a zip file of the tutorial to your PC.

1.1 The School of Earth Sciences UNIX network

Currently the School of Earth Sciences has about a dozen machines running either UNIX or Linux. These all share the common domain name: `earthsci.unimelb.edu.au`. Most of these machines are in the UNIX Laboratory, Rooms 347-348 (this is usually referred to as the 'UNIX Lab'). They may be accessed directly by sitting at the console of a particular machine. However most users find it more convenient to log on from a remote machine e.g. via a PC in the Computer Laboratory, Room 313.

Historically our machines were SUN Microsystems workstations (SUNs) running a 'flavour' of UNIX called Solaris. The machines that use Solaris UNIX are: `atlas`, `mawson`, `orthus` and `rossby`. In recent years, due to the greatly improved in the performance of personal computers, the SUNs have been augmented by PCs running Linux, which comes in several flavours. The machines

that use Red Hat Linux are: cove, gulf, gyre, tide and wave. The latest additions to our UNIX network are PCs that run Debian Linux: vislab01, vislab02, vislab03 and vislab04. Finally there is a DEC Alpha machine called zeus that runs a version of UNIX called OSF1. This particular machine is not generally used except for certain implementations of the Melbourne University General Circulation Model (MUGCM).

In addition to the machines that run a UNIX or Linux based operating system we also have a network of Windows PCs in the Computer Laboratory, Room 313 (this is usually known as the 'PC Lab'). These have a public domain UNIX emulation for PCs called Cygwin that enables UNIX-based software to be compiled and run under the Windows operating system. However the main attraction of Cygwin is that it allows connection to a UNIX machine including the use of X Windows i.e. as if you are sitting at a console in the UNIX Lab. For Honours students this laboratory represents the most convenient point of entry to the UNIX network since there is access to both UNIX and Windows applications from a single machine. A guide to accessing UNIX via a Windows PC is given in Appendix A.

For additional flexibility there is a network of Apple iMacs that run a variant of UNIX called X Darwin. These are located in the Geospatial Analysis Laboratory, Room 315 (this is usually known as the 'iMac Lab'). Some brief notes for accessing UNIX via an iMac are given in Appendix B.

1.2 Logging on to the network

To access the UNIX computer system you need to obtain a username and password by completing the computer account form provided by your supervisor, then forwarding it to Brett Holman or Doug Morrison (support@earthsci.unimelb.edu.au). These login details will enable you to access all of the computers on the network including those in the UNIX, PC and iMac Labs. As part of this tutorial you may access UNIX via a PC (see Appendix A) but you may log on through any machine e.g. the consoles in the UNIX Lab.

2 Some UNIX Commands

'Commands' can be UNIX commands, some are aliases and some are home-grown software, or your own programs.

2.1 Manual pages

```
man alias which whereis
```

You can find information on most UNIX commands and home-grown software using the `man` command.

Sometimes the command that you know is not what it seems - if you type: `alias command`, it should tell you if it has been aliased to something else. To remove an alias, use a `\` in front of the command.

`which` and `whereis` will tell you where you can find the command executable and the man page.

2.2 Navigation

```
cd ls pwd pushd popd
```

The UNIX system is arranged like the root system of a tree, starting at the top with `'/'`. You can look at all directories if the permissions let you (generally other people's mail folders are off-limits). To move around type `cd` (change directory) and the destination.

To view the contents of the directory where you are type `ls` (list).

To check where you are type `pwd` (present working directory), (or '*where the hell am I*').

Try:

```
cd ..      Up one directory
cd .       Stay in current directory - useful in other commands
cd         Back to your home directory
```

`cd /` To the top of the UNIX tree
`cd /home/phope` or `cd ~` Into my home directory
Type `ls` and `pwd` at each new location.

With the `ls` command you can specify options. Here are some examples:

`ls -l` will give you a full listing
`ls -t` listing ordered by time.
`ls -a` will list all files, including those starting with ‘.’

`ls` can be used to list certain files using multiple filenames or wildcards. A ‘?’ is one character, and a ‘*’ is any number of characters.

If you want to remember where you have been and go back there, you can use `pushd` instead of `cd` to move to a location, and then `popd` to get back to the original place.

2.3 Moving files

`mkdir mv rm cp`

`mkdir` (make directory) is used to make a directory in your area. Make one called ‘tute.dir’:

```
mkdir tute.dir
```

Move into this directory:

```
cd tute.dir
```

Copy the contents of the directory `/home/kevin/tute.dir` into your new directory:

```
cp /home/kevin/tute.dir/* .            (note the period).
```

To move files permanently use `mv`.

To remove files **permanently** use `rm`. Most people have `rm` aliased to `rm -i` for safety (you are prompted to carry out the removal).

To remove an empty director- use `rmdir`

2.4 Permissions

`chmod`

Using `ls -l` you will notice a string of letters that look like `drwxr-xr-x` - these are the permissions on that file or directory. The first tells you whether it is a directory or link or not. The next nine are in lots of three, telling you the read, write and executable permissions. The first three are yours, the user's, the next are the group's and the next is others'. You can change the permissions on those files that you own with the `chmod` (change mode) command.

Refer to notes on 'Common UNIX commands' for examples.

2.5 Viewing files

`more less cat m head tail`

Text files can be viewed with the `cat`, `more` and `less` commands. To exit from `less` type `q`. `m` is an alias for `more` or sometimes `less`.

2.6 A few short cuts

`history !n !$!! file extension ^D ^L`

Note: `^` represents the **Ctrl** key.

If you are typing a file or directory name, and you have already typed enough to make it distinct from any other name in that directory (e.g. `cd /home/phope/unix`) you can press the **Tab** key to end the name for you. If you press **Tab** and it does not finish the name, then it is not specific, or you have mistyped some of the first bit. A way to check all the names starting with what you have typed, try `^D`.

If you want to repeat a command, `history` (shortened to `h`) will give you a list of the last few commands. You can use one of these by typing `!number`.

If you want to repeat the last command type `!!`. To use the last word in the last command in Your present command type `!$`.

To repeat a recent command, type ! and the first few letters of the previous command.

To clear the command line type ^U.

To refresh the window you are working in type ^L

To abort from a command that you are entering type ^C

Drag and drop - you can highlight some text and drop it at the cursor in any one of your windows. Use your left mouse button to select, by holding the button down, then click middle mouse button to drop at the cursor in whatever window the mouse is over. Note: This procedure may vary depending on the type of computer you are using (SUN workstation, PC or iMac).

2.7 Finding things and directing output

```
grep find | < >
```

grep looks for strings in files or lists and prints them out. For example, in the directory where you have copied the files from my directory, type: `grep font *.tex` The wildcard with grep is a '.' for a single letter, and '*' for any number of letters – note the forward slash.

grep can also be used with a pipe. A pipe, '|', strings commands together by doing the subsequent command on the result of the first. For example: `ls -l | grep tex` will give you all files with 'tex' in their name. Another excellent use is to pipe fast moving results through less (or m) e.g.

```
ls | m
```

One I find really useful: `ls | grep d` which tells me all the directories at my present location.

Some other ways of 'controlling' commands are:

- ; allows you to write more than one command on a line
- > directs output, from a command somewhere, usually a file
- < directs the contents of a file into a command

If you want to find a file, you can use the `find` command, which recursively goes down the, directories from where you are, looking for a match. I like to use these options:

`find . -name *tex` This will give the full path from where you are of files ending in `tex`.
`find` sometimes takes a long time.

2.8 Changing your password

`passwd`

Type: `passwd` and follow the instructions.

New passwords have to be at least 6 letters long and have a non-letter in them e.g. `AwxyQ7`

2.9 Time and date

`date cal`

`date` tells you the day time and year, `date -u` tells you Greenwich Mean Time.

`cal year` will give you a yearly calendar, `cal month year` (e.g. `cal 7 2004`) will give you a month's calendar.

2.10 Checking up on people

`finger whoami who rwho hostname`

`finger` will tell you a bit about the person (see 'dot files').

`whoami` tells you who you are.

`who` tells you who is logged onto the machine you are working on.

`rwho` tells you who is logged onto all machines.

`hostname` tells you the machine you are on.

2.11 Compiling

`f77 g77 gcc`

These will be discussed in more detail during the *Introduction to Fortran* course later in the semester.

Example: `f77 program.f`

This produces a file `a.out`, which is then run by typing: `a.out`

3 Directory Structure

You will have a directory on the home disk (`/home/username`). It is recommended that you only use that for your dot files, and directories that are put there by default. For example, when StarOffice sets up, it puts a directory in your home directory. Your work should all be done in a `/work*` directory. Each user will have different disk usage requirements depending on their project e.g. running a GCM will require a lot of space compared to processing data from a few climate stations. See Kevin Keay (*not* Brett Holman) to obtain a work directory appropriate for your project.

4 Editing and Printing

`vi nedit lpr lprm lpq a2ps LaTeX glw g2ps`

`nedit` is an editor similar to those found on PCs. Type: `nedit` or `nedit filename` to activate the program. There is online help within the program – also see: `man nedit`. The editor is useful for writing Fortran and C programs since it has syntax highlighting.

`vi` is a line-based editor that is very useful for writing programs. Try: `man vi` and perhaps check out the reference book in Section 1. On the Linux PCs `vi` is an alias for `vim`, which has more features.

To produce ‘pretty’ type-setting, as you’ll need for your thesis, you have a few options. One is to use a PC package such as Microsoft Word, the other is to use LaTeX on the UNIX system. LaTeX (pronounced *lay-tek*) is very powerful, will automatically number your pages and labels, and can

create listings of figures and tables of contents, as well as bibliographies. LaTeX files end in '.tex'. Have a look at: `tute.unix.tex`.

To print use the `lpr` command e.g. `lpr output.eps`

The default printer is `unix` in the UNIX Lab. To use another printer add the `-P` option:

`lpr -Ppclab output.ps` Prints to the PC Lab printer.

`lpr -Pcopy output.ps` Prints to the Copy Room printer on the 4th Floor.

`lpr -Phons output.ps` Prints to the Honours Room printer (Room 447).

Note: To print, files MUST be in ASCII or Postscript. Never use: `lpr gmeta`

If a file is in a binary format, it will produce many many pages with a few characters per page, wasting heaps of paper. Turn off the printer! If in doubt, contact Brett Holman or Doug Morrison.

`lprm` will stop your last job if it has not started printing. `lpq` will give you a listing of the print queue.

Generally you should only print files on a UNIX or Linux machine from an appropriate application e.g. open PDF files with `acroread`, unless they are text only (also known as ASCII) or Postscript. If you are unsure whether a file is in the appropriate format, have a look at it. Postscript filenames generally end in 'ps' (though they don't have to). Depending on the machine, the `file` command may shed more light e.g. `file gmeta` should tell you that it is a data (binary) file.

To print a postscript file type: `lpr filename`

To print a text file with fancy borders type: `a2ps filename`

To print an NCAR Graphics output file (`gmeta`) type: `glw gmeta`.

You may also use the `g2ps` command by Kevin Keay. Type: `g2ps` for help.

For additional information about network printers see:

<http://www.earthsci.unimelb.edu.au/itsupport/internal/printerHT.php#queues>

5 Running Jobs

```
^C ^Z ps jobs renice bg fg top df quota du
```

As stated previously, the definition of a command is any executable, whether that is a shell built-in command, or the executable of a program or script you have written.

To run a command, just, type it at the command line.

Some commands, such as the executable for a climate model take a long time to run, sometimes several weeks, and thus there is a need for some form of job control.

To kill a command type: `^C`

To suspend a command type: `^Z`

When you suspend a command, it is still active, but not running.

To then bring it back to the command line, type `fg` (foreground)

To make the command continue to run in the background, type `bg` (background)

Your command can continue to run in the background while you work in the foreground. It will continue to run after you log out.

We like to give jobs in the background lower priority than those interactive processes so type:

```
renice +15 PID
```

where PID is the process ID.

To kill a job: `kill -9 PID`

To find the PID, you need to view the jobs running. One way is with `top`. One way to see only your own jobs in that window is `jobs -l`. Another way is with: `ps -efl | m`

which gives you information about every little process on the machine you are logged into. One way to filter that is to look only for processes that are 'running' is with:

```
ps -efl | grep " R "
```

If you are going to run a big job, you need to find a machine with the fewest people already running jobs (usually models). To do that you need to check each machine. I like to do it by running the script: `topcheck.csh`. When you have found the right machine, get a window for that machine, and go to your work directory, (not `/home`) and if there is enough space there, start your job. To check how much space is left on each disk, type: `df -k` To check how much space your account has type: `quota -v`

To start your job automatically in the background, type `&` at the end e.g. `run-tracks &`

`du -sk` will tell you the space taken up by the directories below at\below your present location in kilobytes. (A `*` will give information on each directory individually).

6 Data and Graphics formats

`compress` `uncompress` `gzip` `gunzip` `conmap` `graphz` `convert` `gmt` `ferret`
`grads` `matlab` `xv` `display`

In meteorology, much of the work is done with data. This data must be in some kind of format. Some is in ASCII - that you can view with `m`, but most is in some kind of binary format that needs programs to unravel. Here we use a format called ‘conmap’, for global data that is a format curious to Melbourne Uni, and places where people have taken it like CSIRO. We also commonly use data in NetCDF format, some others are: GRIB and HDF.

There are many commands that can be used to work with data in the conmap format. The most obvious one is to plot it to a graphics file: `conmap -G conmapfile` which produces a file called `gmeta`.

`mm` will give you information about the conmap file or modify it.

`graphz` (and `scatter`, `gzhist`) are programs written here that can plot x-y plots. output of this is also a `gmeta` file.

gmt (generic mapping tools) are a suite of commands that can be used to build up Postscript files of graphics. This is useful for pretty maps. Other data processing and graphics tools are `ferret`, `grads` and `matlab`.

Graphics files also come in many different formats, which can be viewed using different software. `gmeta` files can be viewed using: `ctrans gmeta`, or: `idt gmeta`.

Postscript files can be viewed using: `ghostview filename` or: `gv filename`

Many other formats, such as `gif`, `jpeg` and bitmaps can be viewed on a SUN with `xv` (also accessible from the right mouse button menu). For the Linux PCs (and SUNS) use `display`.

`soffice` is a UNIX based office package that is largely compatible with Microsoft Office. It takes a little while to load, and often interprets layouts and fonts differently from PCs. It will set up a directory in your home directory. The Linux PCs have the Open Office software. However, if you are dealing with Microsoft Office documents then it is preferable to use the Office package rather than the emulation software.

`display` can be used to play `mpeg` movies, animated GIF files and the common graphics formats.

`convert` can convert between many graphics formats to others.

For example: `convert file.gif file.bmp`

7 Scripts

The commands you type at the command line that are system commands are actually a part of the TC shell. If you are doing repetitive tasks you can build them up within one file and then run only that one file as a new command. This is called a *script*. There will be more discussion of these in conjunction with the Fortran course.

On UNIX there are other shells that you can work in like the Bourne shell.

8 External Stuff

To transfer files from our system to another one or vice-versa, historically we used `ftp` (file transfer protocol). This is particularly useful for getting data from other institutions or machines, (often with an ‘anonymous’ login), and transferring data between the PCs and UNIX machines. Make sure you use binary transfer. If using a command line form of `ftp`, type `bin`. Today it is preferable to use a secure file transfer method like `sftp` or `scp`.

You can transfer files between the iMacs and UNIX machines via `atlas`. This enables a drag-and-drop approach to transfer files between an iMac and `atlas`. Otherwise you can use `scp`. You need to know the name of the machine you are logged onto. For instance, assume you are on `imac22`. To transfer a file from your home directory on `atlas` to this machine:

```
scp atlas:/home/username/file .    (the full-stop represents imac22)
```

To log onto another system that you have an account on, use `ssh` e.g. `ssh -X cove` (the `-X` option allows X Windows; omit for just text screen use). In some cases, `telnet` will work e.g. `telnet cove`, but its use is discouraged and this utility will eventually be disabled.

Devices

The connection of some of these devices is under review so the information may be out of date. Ask Kevin Keay, Brett Holman or Doug Morrison if you are in doubt.

Floppy drives: On `mawson` and `rossby` there is a floppy drive. It reads PC formatted disks. To use: put floppy disk in drive, type `volcheck`, then `df`, where you will see a directory called something like `/floppy/unnamed-floppy` transfer to there, and your files will be there!

CD-RW drives: The Linux PCs have CD-RW drives. See the program: `X-Toast`. The Windows PCs also have CD-RW drives – look for the appropriate software e.g. `Roxio`.

Tape drives: There is a DLT tape drive connected to `rossby`. This is used to permanently archive files. Since the network is effectively backed up then you shouldn’t normally need to use it. However you may need to restore some data files created by a past user for your own work. Make

sure you are logged into a rossby window. You need to correctly set the TAPE environment variable:

```
setenv TAPE /dev/rmt/2hbn
```

The commands we use at present are: `archive`, `archive_buster` and `dearchive.test`. These scripts will ask you how many blocks you want to skip, and, due to a label added to each block, to dearchive the first block, skip 0, but then to move forward by a block, skip 1. (Therefore to dearchive the 3rd block, skip 2).

In addition there is an Exabyte tape drive connected to rossby. This is only used for *reading* the older Exabyte format tapes (same size as Video 8 used in camcorders). Make sure you are logged into a rossby window. You need to correctly set the TAPE environment variable:

```
setenv TAPE /dev/rmt/3hbn
```

The commands we use at present are: `archive` and `dearchive`.

Scanner: Connected to the iMac server known as terra in the iMac Lab.

Colour Printers: AO and A2(or A3, A4) in the Front Office on the 4th Floor. The easiest way to print to AO at present is to transfer pictures to a Windows PC or iMac, open them in Adobe Illustrator (making sure you have loaded the appropriate printer), size them to the page, and print. For A4 colour prints use: `lpr -Pa4 filename`. Note: **There is a charge** for using this printer (about **\$2 per sheet**). Check with the Front Office staff and with your supervisor to determine *who* will be paying.

9 Dot files - controlling your own set-up

Files that begin with a dot in your home directory control aspects of your set-up. `.cshrc` and `.login` are the major controllers - they run every time you log-in. They contain your paths, aliases and any other fancy extras you want set up when you log in.

When you modify such dot files; be very, very careful - I usually keep a copy of the old version if I am modifying mine.

When you run X Windows, by default you are using a certain type of window manager, called `twm`, the dot file that controls that is called `.twmrc` There are other managers e.g. `fvwm` (or

`fvwm2`) which give your X Windows layout a different look – some are designed to resemble Microsoft Windows!

`.xinitrc` controls the layout of your desktop - window sizes and positions.

`vi` can be set up exactly how you like it by modifying `.exrc`

You will have a *directory* called `.netscape` with info about your past netscape sessions (you can empty the cache every now and again).

`.plan` and `.project` come up if someone enquires about you with `finger`.

`.signature` comes up on the bottom of your mail messages if you set 'signature at bottom of message' in `pine`

`calendar` is an ideal way to remind you to do something - have a look at it.

There are various other dot files that modify other bits of your set up.

10 Miscellaneous

Setting your DISPLAY variable

Sometimes you cannot display graphics from a terminal window that is connected to a machine other than the, one you are sitting at. To remedy this type:

```
setenv DISPLAY 128.250.120.137:0.0
```

(the long number - IP address - is the address of the machine you are sitting at).

Accounts at home

You can log onto the UNIX system from home through your ISP and `ssh`. There are free `ssh` programs for Microsoft Windows. A useful version is at: www.ssh.com. This will enable text-based work. To use X Windows at home you will need an X server. A good commercial one is X-Win32

from: www.starnet.com – in *demo* (free) mode it will run for 15 minutes only. Alternatively you can install Cygwin for Windows. Talk to Kevin Key for more information.

Backups

All disk drives are currently backed up each night. However if you want to *permanently* delete files from the system then you need to archive them to some medium e.g. CD, tape. Talk to Kevin Key if you have a large amount of disk space to archive since you may need to use a DLT tape.

Web Pages

If you want to make your own web page, make a directory called `public_html` in your home directory i.e. `/home/username/public_html.`, then add a file called `index.html` in that directory. You will then have a page called `www.earthsci.unimelb.edu.au/~username`

Note: You *may* have to ask Brett Holman or Doug Morrison to change some administration settings for allowing file access via your page.

multimet

A handy tool for placing a few Postscript files on the same A4 page. Look at the file `do4plot`.

Changing machines

To change from machine to machine you can use the `ssh` command. If you are using X Windows, open a new terminal window and type: `ssh newmachine`.

Spell checker

`ispell` can be used to check the spelling in files.

Reading email

`mail`, `elm` or `pine` can be used to read and send email. However you may find it more convenient to use a web-based mail program via Netscape or Mozilla.

Appendix A

Accessing UNIX from a Windows PC

Kevin Keay

February 2006

For general use it is most convenient to access the UNIX network from the Windows PCs in the Computing Laboratory, Room 313, generally referred to as the 'PC Lab'. It is assumed that you have an account to access the UNIX network. This is normally a part of the administrative aspect of your Honours induction.

1. Sit yourself down in front of a Windows PC in the PC Lab.
2. Log on to the computer with your username/password.
3. You *may* get some Windows-related prompts – click OK.
4. Look for the Cygwin icon on the desktop – it should look like this:



Double-click on this icon.

5. You should see a Cygwin *shell* window appear. It will have a *prompt* constructed from your UNIX username and the machine name (see the sticker on the left hand top of the computer case) followed by a dollar sign e.g. kevin@pclab08\$. By default, the properties of the window are determined by the *Bourne* shell. The *C* shell tends to have more 'friendly' features and so we will activate it.

At the \$ prompt type: csh

You should now see a new prompt similar to this one: [kevin@pclab08 ~]\$.

The tilde (~) is a shortcut for the path:

/cygdrive/d/Documents and Settings/user (in my case, user would be kevin) and represents your *home* directory. It is equivalent to the environment variable \$HOME (note that such variables are case-sensitive). Any subdirectories (folders) that you create are referenced relative to this directory.

6. If you wish, you can try out UNIX commands within the Cygwin C shell. For convenience we will represent commands to be typed by:

% command

where the symbol % represents the prompt.

e.g. % pwd will print the current (working) directory on the screen

```
% cd /cygdrive/d/Documents\ and\ Settings/kevin/work
    go to a folder called work
% cd ~/work    an easier way to do the above
```

Note: Normally you will need to insert a backslash (\) before a blank space so that it will be treated as a character.

A useful online resource from the University of Surrey, *UNIX Tutorial for Beginners*, is available at: <http://www.ee.surrey.ac.uk/Teaching/Unix/>

The above webpage includes a link to download a zip file of the tutorial to your PC. However, our main reason for using Cygwin is so we can connect to the UNIX network. Hence we will defer the main UNIX tutorial until we are accessing the network.

7. In order to access the network ‘nicely’ you will need to alter the behaviour of the C shell. Firstly, you need to download some files to facilitate this process. Make sure you are in your home directory in the Cygwin C shell. If in doubt, type:

```
% cd ~
```

We will now transfer a compressed package (zip) file to your current folder using the text-based web browser lynx:

```
%lynx www.earthsci.unimelb.edu.au/~kevin/UNIX_Course/zcygwin_setup.zip
```

You will be prompted to download the file – type d. The action ‘Save to disk’ will be highlighted in red – press Enter. Then press Enter again to save the file with the same name. When the ‘Saving ...’ message has disappeared, type q to quit, then type y (yes ... you want to quit).

To see a listing of the contents of the zip file type:

```
% unzip -v zcygwin_setup.zip
```

Unzip the contents of the zip file into this folder:

```
% unzip zcygwin_setup.zip
```

If you type:

```
% ls -lat
```

or

```
% ls -lat | more    (pauses the listing after the screen is full)
```

you will see the new files added to your home directory (there will probably be some earlier system-related files here too). The one called `.cshrc` contains some customised settings for the C shell. There are other *dot* files (starting with a dot or full-stop) e.g. `.twmrc`, that are used by other applications. The files `*.sh` (Bourne shell scripts) are different ways of accessing X Windows under Cygwin. Note: Although you are using the C shell you can run scripts that work with the Bourne shell *within* the script. To reset the C shell settings type:

```
% exit
```

```
% csh
```

You should see the following message on the screen:

```
csh setup complete
```

This message will appear each time you activate the C shell unless you remove it from the file `.cshrc`.

8. The scripts `*.sh` in your home directory allow you to access X Windows under Cygwin. Each script starts the X Windows server (`XWin`) but then differs in the windows presentation. You will choose *one* of these. There is a shorthand name or *alias* for each script specified in the file `.cshrc`:


```
% X                opens a single C shell X window or xterm
```

```
% fvwm2          opens two C shell windows on a blank background
                  using the window manager fvwm2
% twm            as with fvwm2 but with the window manager twm
```

The first is convenient if you are using Microsoft Windows applications e.g. Word, as well as UNIX-based applications from a PC i.e. it is an *integrated* environment. The other two approaches give you an X Windows *desktop* similar to one you would see if you logged on to the console of a machine in the UNIX Lab. `fvwm2` behaves slightly differently to `twm`.

For simplicity, type:

```
% X &
```

The ampersand (&) means that the script will run in the background, allowing you to use the Cygwin window for other tasks. You may see some system diagnostic messages and the X server icon  will appear in the System Tray. Then a single Cygwin C shell X window (`xterm`) will appear at the upper left of the screen.

From the current `xterm` you can try out various UNIX commands and software that has been compiled to run on a PC running Cygwin. In most cases the behaviour is identical to that of a true UNIX operating system – the emulation is very good. For instance, there is a JPEG graphics file included in the zip file (see step 7). You can view this image with `display`:

```
% display vance_track.jpg
```

Left-click to get a menu; choose File, then Quit to exit the program. This same program is available on the machines of our UNIX network.

9. As mentioned in step 8 you can try the window managers, `twm` and `fvwm2`. In both cases you will get a screen-sized background with two `xterms` on the left hand side. Remember: you can only have one instance of X Windows running i.e. one X server icon in the System Tray. So pick one of the three options in step 8 e.g.

```
% fvwm2 &
```


When activating a new application e.g. an `xterm`, `twm` shows them in outline form until you left-click while `fvwm2` brings up a window in the same way as Windows i.e. no clicking required. With `twm`, to close a window you left-click on the *background* and select Kill. Then you move the skull-and-crossbones cursor over the desired window and release the mouse – the window should disappear (quite a dramatic process!) With `fvwm2`, the procedure is similar to Windows: left-click at the upper left corner of the window and select Close. With `fvwm2`, if you left-click on the background a menu will appear. Not all of the items may be installed under Cygwin. If you select Terminal, an `xterm` will appear; select Editor, a `nedit` window will appear. Select Programs – Applications – GV (Postscript): a `gv` window will appear. Alternatively, these programs can be activated from an open `xterm` e.g.:

```
% nedit &
```

The behaviour of `twm` can be customised by modifying the dot file `.twmrc` in your *home* directory. Similarly, modify `.fvwm2rc` to change the properties of `fvwm2`. More information on these window managers is given by:

```
% man twm
% man fvwm2
```

10. When you have finished with X Windows under Cygwin it is **important to close the X server**. The safest procedure is:

- Close all `xterm` windows connected to a UNIX machine e.g. atlas.
- **Right-click on the X server icon**  and click **Exit**; then click **Exit** in the dialog box which appears.

If you don't do this then next time a new user logs on to the current PC they will be unable to activate the X server. The error message relates to being unable to connect to 127.0.0.1:0.0 (which is equivalent to localhost:0.0). There will be a temporary file called X0 that is owned by the previous user. You can check for this by:

```
% ls -l /tmp/.X11-unix/X0
```

If you see the file X0 present then you may be able to determine the owner from the username and ask them to delete it. Otherwise contact Brett Holman or Doug Morrison to delete the file for you.

11. However, our main motivation for using the X Windows capabilities of Cygwin is so we can connect to a UNIX machine. The machines that are currently accessible are:

- | | |
|-----------------------|--|
| (1) Solaris OS: | atlas, mawson, orthus, rossby |
| (2) Red Hat Linux OS: | cove, gulf, gyre, tide, wave |
| (3) Debian Linux OS: | vislab01, vislab02, vislab03, vislab04 |
| (4) OSF1 OS: | zeus |

(OS means operating system). Group 3 are the newest and fastest machines. Group 2 are recommended for most uses. Apart from orthus, the group 1 machines are relatively slow. Group 4 is for specialised uses e.g GCM work. So, normally you would select a machine in group 2 or 3 unless you need to use the Solaris OS. Also, see section 1.1 of the notes *Introduction to the School of Earth Sciences UNIX computer network*.

Let us assume that you have started X Windows with X, `twm` or `fvwm2` (see steps 8-9). In an `xterm` window type:

```
% ssh -X atlas
```

You can replace atlas with any valid UNIX machine name e.g. cove. The connection is controlled by the secure shell client program (`ssh`). The `-X` option allows X Windows under Cygwin to display windows created by the remote machine. The first time you ever connect to a particular machine you will get a message asking for confirmation of your desire to connect – type **yes** not **y**. This message will not appear again for you and this *pair* of machines. Note that your PC has a name e.g. pclab11. You are then prompted for your password (your username is carried through from your PC login). Some messages will appear and then a C shell prompt. You are now logged on to a remote UNIX machine!

12. Since you may use a machine running Solaris (what we normally call UNIX) or a version of Linux it is necessary to customize your log on procedure. Let us assume you are logged on to a UNIX machine e.g. atlas. Ensure that you are in your home directory – type:

```
% cd ~
```

You don't type % - it just represents your prompt. Then:

```
% cp /home/kevin/UNIX_Course/zunix_setup.zip .
```

Note the dot at the end of the command – this means use the same filename in your account.

Now:

```
% unzip zunix_setup.zip
```

You may get a message to overwrite files (possibly `.cshrc`) – respond with **y** (yes). To list the new files type:

```
% ls -a .cshrc*
```

There is a master C shell customisation file (.cshrc) and specific files for each operating system: .cshrc.solaris (group 1), .cshrc.linux (groups 2 and 3) and .cshrc.osf (group 4) See step 11 for the group details. These files may be customised further – see: `man csh`. To activate the settings, log out i.e. type:

```
% exit
```

Then log on again with `ssh` as in step 11 e.g.:

```
% ssh -X atlas
```

In future your customisation should be suitable for whatever machine you connect to.

13. It is useful to be able to transfer files between a PC and a UNIX machine. You can use the



Windows Secure File Transfer Client

or you can use secure copy (`scp`).

From a Cygwin C shell window (you don't need an X server for this task) move to the desired location (folder). Then to transfer files from a PC *to* a UNIX machine:

```
% scp files unixmachine:
```

Assume you have one file to transfer from the PC to (say) atlas.

```
% scp myfile.dat atlas:
```

You will be prompted for a password (you may also get a connection message to which you should respond 'yes') and your file will be transferred to `/home/user` on the UNIX machine.

If you want to transfer the file to another location then you can give the folder name.

```
% scp files unixmachine:/folder
```

For instance:

```
% scp anotherfile.txt cove:/home/user/mydata.dir
```

For multiple files, you can use something like:

```
% scp \*.dat atlas:
```

The slash allows the `*` to be interpreted correctly. Note that the `*` allows any file ending with `.dat` to be transferred.

If you want to transfer a file *from* a UNIX machine to your current PC:

```
% scp unixmachine:/folder/files .
```

For instance:

```
% scp atlas:/home/user/data.dir/test.jpg .
```

will transfer the file `test.jpg` to your current folder under Cygwin. This file might be inserted into a Word document on the PC.

Note: Each time you use `scp` you will be asked for your password.

14. You can get a text-only connection by using the Windows Secure Shell Client:



Cygwin is not required.

15. You are now ready to try the introductory UNIX tutorial in Appendix C.

Appendix B

Accessing UNIX on an iMac

Kevin Keay

February 2005

Firstly, I am *not* an expert on the use of an iMac! I chose this platform for the introduction to UNIX since the main features are available via X Darwin

1. Log on to an iMac using your system username and password (you should have obtained one from Brett Holman after sending him the relevant form).

2. X Darwin is the Apple version of UNIX. To access it:

Go -> Applications

 X Darwin

Drag the icon or name to your menu bar (dock) at the bottom of the screen.

3. You might also want to use the terminal program called Terminal (what else!)

Go -> Applications

 -> Utilities

 Terminal

Again, drag this icon to the menu bar.

4. Click on the X Darwin icon in the menu bar. Select the Full Screen option. You may also see a dialog box asking about X11 – click yes.

Some X windows will pop up. Each of the terminal windows is a Unix `tcsh` running on your own iMac. The ‘look’ of the windows is that associated with `twm` (default).

5. Normally you will connect to one of the Meteorology machines or the School server atlas. From a terminal window type: `ssh -X atlas` You will probably receive an `ssh` connection message for the first time you ever connect to a specific machine. Just answer: yes. Then you will be asked for your password (`ssh` assumes that the username is that of the person logged on to the iMac). If all goes well, then a prompt will appear in the window. You can now run software on atlas as if you were physically sitting at its console. You may wish to try `orthus` instead of `atlas`. Note: If you receive an `ssh` connection prompt during another session just answer: yes.

6. Note some special iMac keys at the bottom left of the keyboard.

 Alt – Apple –A Toggle between X Darwin and the iMac

 Ctrl – Alt Brings up a font menu

To copy and paste text: Select text with the mouse. Then move the cursor to where you want to paste and simultaneously press the Apple key while clicking the mouse.

7. A note on resizing windows: If you click and hold on the top right-hand box in the window title bar, you will be able to resize the window – an outline appears. A tip: To reduce the vertical size of the window you need to drag the outline *up* then down.

8. If you want to copy a file from atlas to your iMac then use scp:

```
scp atlas:/home/username/file .      (the period is your iMac)
```

```
or scp atlas:/home/username/file imac22:  (assuming your iMac is imac22)
```

The file will be transferred to the home directory of the specific iMac you are on.

For access from other iMacs you should move the file to the pgrad area. You should be able to do this from the Folders icon.

Note: If you want to transfer in the other direction then you could log on to atlas via X Darwin and use:

```
scp file imac22:      (assuming your iMac is imac22).
```

With a SMB account on atlas the procedure is a little easier.

9 When you have finished your work, log out from the remote machine e.g. log out from atlas in all of the windows that you ran `ssh`. Press Alt – Apple –A and then quit X Darwin from the top menu.

Appendix C

An Introductory Tutorial on UNIX

Kevin Keay

February 2005

The purpose of this tutorial is to provide a *basic* introduction to UNIX. It is assumed that the user has read Appendix A or B i.e. you are able to correctly access a UNIX machine from a PC running X Windows under Cygwin or an iMac running X Darwin. In this document, % represents your prompt and user is your username e.g. kevin.

1. Log on to atlas.

```
% ssh -X atlas
```

2. Create a folder (also called a directory) in your home directory /home/user where user is your username – by default, this is your current location. If you are unsure, go to your home directory with

```
% cd ~
```

In this example our new folder is called Tute – note: UNIX is *case-sensitive*.

```
% mkdir Tute
```

Go into this folder.

```
% cd Tute
```

You can check your location with pwd.

```
% pwd
```

Your location should be: /home/user/Tute

3. Copy the following file from Kevin Keay's account.

```
% cp ~kevin/UNIX_Course/test.f .
```

The dot at the end of the command means that the copied file will have the same name as the original. You can specify your own name if you wish.

```
% cp ~kevin/UNIX_Course/test.f mytest.f
```

To see a listing of the files in the folder Tute use the ls command.

```
% ls
```

```
% ls -l
```

Gives a long listing – more details

```
% ls -l *
```

All files but not those starting with a dot (dot files)

```
% ls -la
```

Includes dot files

```
% ls -l t*
```

Those starting with t

```
% ls -l *.f
```

Those ending with .f

```
% ls -l *.*
```

Those with a second-last dot, then a single character

4. The file test.f is a text file. If you use the file command, you are told it is ASCII text (normal text).

```
% file test.f
```

The extension `.f` indicates that this is a *Fortran source file*. This particular program prints a name (some text) and the output of a simple calculation on the screen. To convert this code into an *executable* (program or software) you need to *compile* it with `g77`.

```
% g77 -o test test.f
```

The argument after the `-o` option is the name of the executable, in this case `test`. After compilation you can run the program by typing its name.

```
% test or perhaps ./test if the current folder is not in your path (check the PATH environment variable i.e. % echo $PATH)
```

You should see the following text printed on the screen.

```
My name is: Kevin
x= 2.000000 y= 9.200000
```

The name is 'hardwired' into the program. The value of $y = x + 7.2$ where $x=2.0$.

5. You can edit the Fortran source code to give a different response. The standard UNIX text editor is called `vi` (visual editor but some say 'virtually impossible'!) However there is an easier text editor called `nedit`, which is similar to Windows programs like notepad. Let us edit the file `test.f`.

```
% nedit test.f &
```

The ampersand (&) allows the current window to be used for other tasks i.e. `nedit` is run in the background. You can change the name Kevin to another one. In addition you could change the value of `x` or the number added to it. Save the file using the top menu: File – Save. Then redo step 4 and see how the output has changed. You could alter the program so that it prompts the user for a name or number. You will learn how to do this and many other tasks during the Fortran course later in the semester.

6. If you list the files in the directory you should see something like this (there *might* be other files too).

```
% ls -l

total 82
-rwxr-xr-x  1 kevin  wheel      81595 Feb 17 14:34 test*
-rw-r--r--  1 kevin  wheel       206 Feb 17 13:44 test.f
```

The total value of 82 means that the files occupy 82 KB (kilobytes) of disk space. The owner is kevin and the group is called wheel . The group name is not usually important. However the file system is normally set up such that your fellow students and perhaps staff have read access to your files (except mail folders).

The file `test` is 81595 bytes in size while `test.f` is much smaller at just 206 bytes. Also `test.f` was created (or last modified) at 13:44 on Feb 17 of the current year (2006). The asterisk (*) next to the filename `test` indicates it is an executable file i.e. it can be run as software.

The first column in the listing gives the permissions for the files. The first character (-) means it is not a directory (you will see `d` here for a directory or folder) followed by three sets of three characters. These sets represent the permissions for the user (you!), the group and other (anyone else). The codes are read (r), write (w) and execute (x). The symbol – means no access. They also have the numeric codes 4, 2 and 1 respectively, with 0 denoting no access.

For the text file `test.f`, the user can read and write to it and all others can just read it. For the executable file `test`, the user can read, write and execute it and all others can read and execute it.

You can change the access permissions with `chmod`. For instance, to prohibit anyone but you from accessing your Fortran source code files you can use the command:

```
% chmod 600 test.f or:
% chmod 600 *.f
```

where the user permission is `rw-` ($6 = 4 + 2 + 0$) and the group and other permissions are both 0 ($0 = 0 + 0 + 0$).

7. To move back to the home directory (`/home/user`) type:

```
% cd ~
```

To see a directory listing of your home directory type:

```
% ls -l
```

or:

```
% ls -la
```

The `-a` option shows the dot files (those starting with a dot or full-stop) e.g. `.cshrc`

You will also see the directory `Tute` (with a `d` as the first permission code):

```
drwxr-xr-x ... Tute
```

Depending on whether you use mail programs on a UNIX system you might see a `Mail` or `mail` folder.

```
drwx----- ... Mail
```

Note that only the user can look at files in this directory (the group and other permissions are both `--`).

If you create a folder within a folder then you can use the `/.` notation to move around the directory structure or tree. Assume that you have created two folders called `examples` and `code` in the folder called `Tute`.

```
%cd ~
% cd Tute
% mkdir examples
% mkdir code
% cd examples
```

To go up one level type:

```
% cd ..
```

A directory listing with `ls` would show that there are two folders, `examples` and `code`, in this current location (`Tute`). Repeating `cd ..` would take you up one level to your home directory. Assume you are in the folder `examples`. Then to go to `code` you would type:

```
% cd ../code
```

You can always type the complete path to a folder if it gets complicated.

```
% cd /home/user/Tute/code
```

Later in the semester you will be assigned a *work disk area*. The work areas have names like `/work18/user`. Hence you would issue commands like:

```
% cd /work18/user/data.dir
```

The folder `data.dir` (file and folder names may contain dots) might contain some pressure data files.

8. As a final exercise you can perform some typical UNIX tasks related to your Honours work. Assume that you need to produce a contour map of a pressure data file for June 1 1996 600 UTC. There is an appropriate file from the NCEP Reanalysis in Kevin Keay's area.

```
% cp ~kevin/UNIX_Course/pmsl.ncep.1996060106.cmp .
```

To see the contents of the file:

```
% mapx -f pmsl.ncep.1996060106.cmp | more
```

```
% mapx -f pmsl.ncep.1996060106.cmp >! pmsl.txt
```

In the first form the output of the `mapx` command is piped to the `more` command so you can view it one page at a time. Without `| more` the output will quickly rush by on the screen. In the second form the output of the `mapx` command is directed (`>`) to a file called `pmsl.txt`. The `!` means to overwrite `pmsl.txt` if it already exists. You can view this file with:

```
% more pmsl.txt
```

```
% nedit pmsl.txt (followed by & to run in the background if you wish)
```

In this ‘dump’ of the contents of the pressure file you can see information about the latitudes and longitudes as well as the actual values at each (longitude,latitude) grid point.

To produce an image of the pressure data you can use a NCAR Graphics program called `conmap`.

```
% conmap -G pmsl.ncep.1996060106.cmp
```

Note: On the Linux machines `conmap` is actually `conmap_kk`. This version of `conmap` is also available on the Solaris machines like `atlas` – type: `conmap_kk`.

If you type:

```
% ls -l g*
```

You will see a file called `gmeta`. This is a graphics metafile. To view this file:

```
% ctrans gmeta &
```

A graphics window will appear with a pressure map. With the mouse over the window, type `q` to close the window.

To convert the file to Postscript for printing:

```
% g2ps gmeta
```

List the files starting with `g` with:

```
% ls -l g*
```

or Postscript files ending with `.ps`

```
% ls -l *.ps
```

and you will see a file called `g.ps`

You can view this file with `gv`:

```
% gv g.ps
```

The top menu allows you to reduce or enlarge the image on the screen. With the mouse over the window, type `q` to close the window.

You may print this file to the printer in the PC Lab:

```
% lpr -Ppclab g.ps
```

9. This concludes the introduction. Later in the semester you will learn about scripts and more advanced features of UNIX. For now, you should have enough information to use the UNIX network for most common tasks. Please review the material in the main part of the notes notes *Introduction to the School of Earth Sciences UNIX computer network*. In addition, some help on UNIX commands is given by:

```
% man unixcommand
```

e.g.

```
% man nedit
```