CSE1303 Part B: Introduction to Computer Systems

Practical Session CS1P
Introduction to Linux

10 marks

Aim of this laboratory

After this session you should:

- have a Windows NT account for the Part A section of the subject
- have a Linux account for the Part B section of the subject
- know how to start X Windows
- have a basic knowledge of Unix commands
- know how to use a text editor
- be able to compile and run C programs under Unix
- be able to transfer files between an MS-DOS floppy disk and your Linux account
- be able to transfer files between Unix machines

Preparation

Although there is no explicit preparation you need to do for this prac, you should:

- read this entire lab beforehand
- make sure your Windows NT account is working
- bring a 3.5" floppy disk

All remaining labs for CSE1303 require a significant amount of preparation. If you do not prepare you will not gain the preparation marks for the lab; more importantly you will be struggling to finish the work in the allotted time.

Session overview

- Windows NT accounts (used for Part A: Data Structures and Algorithms)
- Linux (Unix) accounts (used for Part B: Introduction to Computer Systems)
- Booting (starting up) Linux
- Logging in and out of Linux
    - Logging in and out of Linux
    - Command line prompt
- Starting X Windows

Introduction

CSE1303 Computer Science is made up of two parts:

- Part A, Data Structures and Algorithms (DS)
- Part B, Introduction to Computer Systems (CS)

In Part A you will write C programs on the computer in Borland C running under MS-DOS from your Windows NT (`MFS` or equivalent) account. Part B involves writing programs in the MIPS assembly language in a simulator called `SPIM`. `SPIM` runs in X Windows under Linux.

As a result, you will need two accounts this semester:

- A Windows NT account for doing Part A
- A Linux account for doing Part B

You are welcome to do your Part A pracs in Linux too, if you wish.

What's Linux?

You are probably familiar with MS-DOS and Windows 95. These are two different environments for you to work in, with different commands for doing things and different internal construction. They are two different *operating systems*. **Unix** is another operating system which is widely used in business and education. There are many flavours of Unix, each of which is very slightly different. **Linux** is one such flavour of Unix. It is freely available and runs on most machines that can run MS-DOS.

Windows NT accounts

Every Monash student can get an account on the Windows NT network. At Clayton the NT servers begin with `MFS`. If you already have one then you simply use it. If you don't have one or if you have forgotten your password you will need to take your ID card to IT Services Helpdesk where you can be issued with your username and password.

Linux accounts

1 mark

To do your Part B pracs you will need to obtain a Linux account. You will need to register to activate your Linux account, even if you had one last year. As with your Windows NT account, you need to get both a username and a password. Your username is the same as on Windows NT account, except that it is in lower case. To get your password, follow these steps:

- Have a pen and paper handy
- Log into your Windows NT account
- Do only **one** of the following:
  - type at the prompt: `TELNET ALPHA1.CC` then log in as `REGISTER` (*not* your own username), *or*
  - open Netscape in Windows and point the browser at `http://alpha1.cc.monash.edu.au/htbin/getpass`

Both methods ask for your username, ID number and date of birth. The password disappears after 40 seconds so make sure you write it down. You can only register once so if you lose your password you will need to go to IT Services Helpdesk to have your password changed.

Booting Linux

Once you have obtained your Linux password you can log into your account. To do this, you must first start up Linux on the computer in front of you. To do this, follow these steps:

- Turn on the computer, or if it is already on, press `CTRL+ALT+DELETE`
- The computer will prompt you for an operating system to load as it is booting. Simply select Linux and press `ENTER`. You may be prompted to do other things along the way.

Do I log into Windows to boot Linux?

You don't need to log into your Windows NT account to boot Linux, but it doesn't do any harm if you are logged in already. You are automatically logged out of your NT account when Linux boots.

Since Linux is a complex operating system, it takes a while to load. When it is ready, it displays a prompt like this:

```
clay-b19g13-10 login:
```

Every computer has a different name; this prompt tells you the name of the computer in front of you.

Logging into Linux

1 mark

Linux is a multi-user operating system. Many people can use the same machine at the same time (although usually only one user will sit at a console, like you are doing now). As a result, you need to *log in* before you can do anything.

To log in

At the prompt, type your username and password:

```
clay-b19g13-10 login: jrstu8
Password: 12MyPasswd34  (Your typing will not appear on screen)
```

! **Login pitfalls**

You must type your issued password in lower case, even if the registration program displayed it in capitals. You may be asked to type your password a second time when you log in.

To log out

When you're finished, type `logout` or `exit` to close your session. Try logging in and out a couple of times to see how it works.

The command line prompt

Linux, like MS-DOS, is mostly a command-line-based operating system. You type your commands at a prompt and the computer performs them one at a time. The prompt usually looks something like this:

```
clay-b19g13-10> _
```

This is called the *command prompt*. The flashing cursor shows where text you type will appear.

Typing a command

Try typing a command at the prompt:

```
clay-b19g13-10> uname -rs
```

`uname` is a Unix command to display the name and version number of the type of Unix your PC is running.

Careful of Capitals!

Linux, as all forms of Unix, is case-sensitive. This means that you have to type capital letters as capital letters and lower case as lower case. Almost all Unix commands are in lower case. Try typing **`UNAME`** in capitals and you'll see that Unix can't find a command called `UNAME`.

Changing your password

Since the Linux account password you were given by the registration program is difficult to remember, it's probably wise to change your password right away to something that you can remember more easily. To change your password, type:

```
clay-b19g13-10> passwd
```

You will be prompted for your old password, then you should type a new password and confirm it by typing it again. This will be your password until you change it again.

Password suggestions

- If Linux thinks your password is too simple it will tell you so and you will have to think of a better password. Mixing lower and upper case along with numbers and punctuation is a good way of making your password harder to guess.
- If you have trouble remembering passwords, it's probably a good idea to keep your passwords on all your accounts the same. Changing your Linux account password does not change your Windows NT account password, and vice versa. To keep all your passwords the same you will have to change each one individually.

X Windows

1 mark

By default, Linux has a text-only command-line interface. But just like Windows 3.1 runs over the top of MS-DOS, there is a graphical environment you can run in Linux. It is called the *X Window System*, or usually just "X Windows".

Starting up X Windows

You can start from the command prompt by typing:

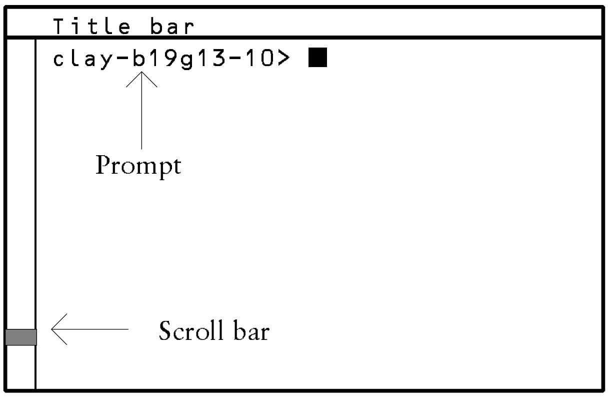
```
clay-b19g13-10> startx
```

Subject to change

The setup used from year to year on the computers in the labs is not always the same. The following is how things were done in Semester 1, 2000, but this may change. Your demonstrator will alert you to any major changes that you need to be aware of.

About X Windows

When X Windows starts, you will see one window on the screen. This is a terminal window, just like the white-on-black screen you were using before you started X Windows. You can open more of these windows by clicking on the `Start` button in the lower left corner of the screen, and choosing any entry that looks like "X terminal". Other applications are available this way too.



There are usually icons along the top edge of each window. Click on them and see what they do.

Each terminal window has a scroll bar. This scroll bar allows you to go back and see text which was lost beyond the top of the screen. To do this, put the pointer on the scroll bar and drag with the middle mouse button.

Middle button on a two-button mouse?

If your mouse has only two buttons you can press the imaginary "middle" button by pressing both buttons at once.

Exiting X Windows

To exit from X Windows, you should close all the applications that you are running in all virtual screens. (If you don't then any unsaved work you have in them is lost.) To close a terminal window, type **`exit`** or **`logout`**; other applications will likely have a "Quit" control.

Click on the Start menu button. Pick `exit fvwm` at the bottom of the menu. This will return you to the text-mode screen which you will also need to log out of in the usual way (i.e., by typing **`logout`** or **`exit`**).

Directory commands

1 mark

To keep files organised on your home computer you are probably used to placing files inside directories (or folders as some operating systems call them). Unix does the same thing, and like those other operating systems there are ways of moving about and managing directories.

Your home directory

As Linux is a multi-user system, there needs to be some way of keeping all of your files separate from everyone else's. Linux does this by assigning you a *home directory* which belongs to you alone. You typically do all your work in your home directory. As you have just begun to use this account, it will appear to be empty. However, it has some special hidden files (which you will see shortly).

You can put whatever files you like in your home directory and no one else can access them or delete them. There is a limit of 20 MB; if you exceed this you will receive an email asking you to delete files. If you ignore this message your account will be disabled.

Listing the contents of a directory: `ls`

`ls` stands for *list*. Try typing the following command:

```
ctl-106-010> ls
```

No file names appear because there are no user files in your home directory yet. However, there are some normally hidden files which you can list by giving `ls` the `-a` option:

```
clay-b19g13-10> ls -a
```

All hidden files begin with the "`.`" character. As you become more familiar with Linux you will discover the purpose of some of these hidden files.
Now try these two options:

```
clay-b19g13-10> ls -l

clay-b19g13-10> ls -F
```

Options can be mixed; try the combinations `ls -al`, `ls -lF`, `ls -alF`. What do you think the `-l` and `-F` options do?

Showing the path of your current directory: `pwd`

It is easy to get lost among all the directories in Linux. If ever you need to know what your current directory is, just type:

```
clay-b19g13-10> pwd
```

`pwd` stands for *print working directory*. The directory `pwd` just displayed is the path to your home directory. No matter what your current directory is, this will always refer to your home directory.

Changing the current directory: `cd`

`cd` stands for *change directory*. Try typing the following:

```
clay-b19g13-10> cd /etc
```

Now use the `ls` command to list the files in the directory `/etc`. Try and guess what `pwd` will print when you run it in this directory. Were you right?
Now change to the root directory and list the files there:

```
clay-b19g13-10> cd /
clay-b19g13-10> ls -F
```

Try to change into some of the directories listed, such as `/tmp` or `/bin` or `/dev`. What sort of files do these directories contain? What do you think the purpose of these directories is?

Special directory names

To change up a level use the special filename "`..`":

```
clay-b19g13-10> pwd
/usr/local/lib
clay-b19g13-10> cd ..
clay-b19g13-10> pwd
/usr/local
```

The special filename name "`.`" always means the current directory, and the filename "`~`" always refers to your home directory.

Creating a new directory: `mkdir`

`mkdir` means *make directory*. This command creates a new, empty directory inside the current directory. Go to your home directory and create a directory called `cs1p`:

```
clay-b19g13-10> cd ~
clay-b19g13-10> mkdir cs1p
clay-b19g13-10> cd cs1p
```

?

Did it work?

After you typed the three lines above, all you got was another prompt each time. How do you know that the commands worked? Unix is not a very verbose operating system and most commands only output text if something went wrong. If you don't trust Linux (or yourself), have Linux show you the results manually using commands you already know:

```
clay-b19g13-10> cd ~
clay-b19g13-10> pwd
/ccst1/j/jrstu8
clay-b19g13-10> mkdir cs1p
clay-b19g13-10> ls -F
csp1/
clay-b19g13-10> cd cs1p
clay-b19g13-10> pwd
/ccst1/j/jrstu8/cs1p
clay-b19g13-10> ls -F
clay-b19g13-10>
```

The last `ls` printed nothing because the directory was just created and is therefore empty.

As you gain more confidence you will get used to Linux's quiet nature and need to check yourself less and less.

Removing a directory: `rmdir`

`rmdir` stands for *remove directory*. To remove a directory you must be a level above it, in its *parent directory*:

```
clay-b19g13-10> cd ..
clay-b19g13-10> rmdir cs1p
```

To confirm that the directory has indeed been deleted, list the files in the current directory and you will see that `cs1p` is no longer there.

Removing a non-empty directory

Before you can remove a directory with `rmdir` the directory must be completely empty, even of hidden files that only `ls -a` lists. See the `rm` command later for how to delete files.

Working with directories

It's a good idea to put all your files for one prac into one directory, so that your home directory doesn't become too cluttered. At the end of this lab, re-create the `cs1p` directory and put all the files you created into that directory.

Online documentation

1 mark

Linux has an extensive collection of online documentation for most commands, including all the commands covered in this prac. The command you use to read these *manual pages* is `man`:

```
clay-b19g13-10> man ls
```

Use `ENTER` or the space bar to scroll through the documentation and press `q` to quit. Find the `-a`, `-l` and `-F` options and write down their meanings. You may find other useful options too which you might wish to try out.

Try some other manual pages for commands you have covered so far:

```
man mkdir
man rmdir
```

Examine the content and layout of the manual pages. Is there a common layout? You should become familiar with manual pages; you will be using them often.

Why is there no entry for `cd`?

Some commands, such as `cd`, don't have an entry in the manual pages. Many of these commands are not actually separate utilities such as `mkdir` and `ls`. `cd` is built into the command interpreter (or *shell* as it is called in Unix). Your default shell is called `csh`; for information on commands built into the shell, try `man csh`.

Some other commands without manual pages, such as `vi` (see the section on text editors later), were not supplied with documentation. Thankfully there are not many of these.

How does the `man` command work? Try `man man`.

File commands

1 mark

Linux provides a selection of commands for working on individual files rather than directories. These commands allow you to view, copy, move, rename or delete files.

Viewing the contents of a file: `more` and `less`

`more` and `less` are two programs which allow you to view any text file a screenful at a time.
Change to the directory `/etc` and view a file there:

```
clay-b19g13-10> cd /etc
clay-b19g13-10> more printcap
```

You can move about the file using the same keys as for reading manual pages. What do you think the file `/etc/printcap` contains?
Now try using `less`:

```
clay-b19g13-10> less printcap
```

What is the difference between `more` and `less`? (Hint: **`man more`**, **`man less`**.)

Copying a file: `cp`

`cp` stands for *copy*. Quite often you will want to make a copy of a file, so that you can modify it or place it somewhere more convenient than someone else's home directory. There are basically two ways of doing this:

- Change to the directory of the original version and copy it from there to the destination.
  - For example, to copy `/etc/printcap` to your home directory:

```
clay-b19g13-10> cd /etc
clay-b19g13-10> cp printcap ~
```

  - If you want to rename the file as you copy it, do this:

```
clay-b19g13-10> cd /etc
clay-b19g13-10> cp printcap ~/newprintcap
```

- Change to the destination directory and copy from the original version to the current directory.
  - Using the same example:

```
clay-b19g13-10> cd ~
clay-b19g13-10> cp /etc/printcap .
```

- Or to change the filename as you copy:

```
clay-b19g13-10> cd ~
clay-b19g13-10> cp /etc/printcap newprintcap
```

Notice the special directory names `.` and `~` from the section on the `cd` command.
Of course, if you are simply copying a file within the same directory, the command becomes:

```
clay-b19g13-10> cp printcap copyofprintcap
```

Moving or renaming a file: `mv`

`mv` stands for *move*. `mv` can move files across directories or within the same directory, in which case it becomes a renaming command.

```
clay-b19g13-10> mv printcap renamedprintcap
```

As always, you can do a directory listing with `ls` to confirm that `printcap` has been renamed to `renamedprintcap`.

Deleting a file: `rm`

`rm` stands for *remove*. To delete `renamedprintcap` from your home directory:

```
clay-b19g13-10> rm -i renamedprintcap
```

Examine the `-i` option by typing **`man rm`**.

! **Oops!**
Unlike some operating systems, Linux provides no "undelete" option. If you delete a file, it is gone forever. Keep backups of all your important files and stay in the habit of typing `rm -i` instead of just `rm`.

Printing a file: `lpr`

`lpr` stands for *line printer*. Laser printers didn't exist when this Unix command was named. Despite the name, `lpr` will print to any printer. Before you print you need to select your printer. The simplest way to do this is to type:

```
clay-b19g13-10> setprint
```

and follow the instructions. Ask your demonstrator if you need help.

!

Advice on printing from Linux

Printing on Linux has historically been difficult and not without pain. You may find it easier to simply transfer the file to a floppy disk and print it from your Windows NT account.

File extensions

Operating systems based on MS-DOS (including Windows) place great weight on file extensions (the part of a filename after the "`.`"). Unix is not quite so forceful about it but there are still some occasions when knowing file extensions will be helpful.

Match up the file extension with the type it indicates:

`.c`	PostScript printer file
`.ps`	Compressed file using Unix `gzip` program
`.txt`	Word processor document
`.Z`	Plain text file
`.gz`	Assembly language source code file
`.doc`	C source code file
`.s`	Compressed file using Unix `compress` program

Text editing

1 mark

There are several text editing programs available on Linux. The simplest to use is probably `nedit`. To learn how to use it read its manual page or run the program and select its Help feature.

Using `nedit`, type in the following file and save it as `sumFor.c`:

```
#include <stdio.h>
main()
{
  const int n=5;
  int sum, num, i;

  sum=0;
  for (i=0; i<n; i++)
  {
    printf("Please enter number %d:", i+1);
    scanf("%d", &num);
    sum = sum+num;
  }
  printf("sum = %d\n", sum);
}
```

Exit the editor and use the `cp` program to copy `sumFor.c` to `sumWhile.c`.

Now edit the file `sumWhile.c` so that the `for` loop is changed to a `while` loop. Save the changes you made to `sumWhile.c`.

Other text editors

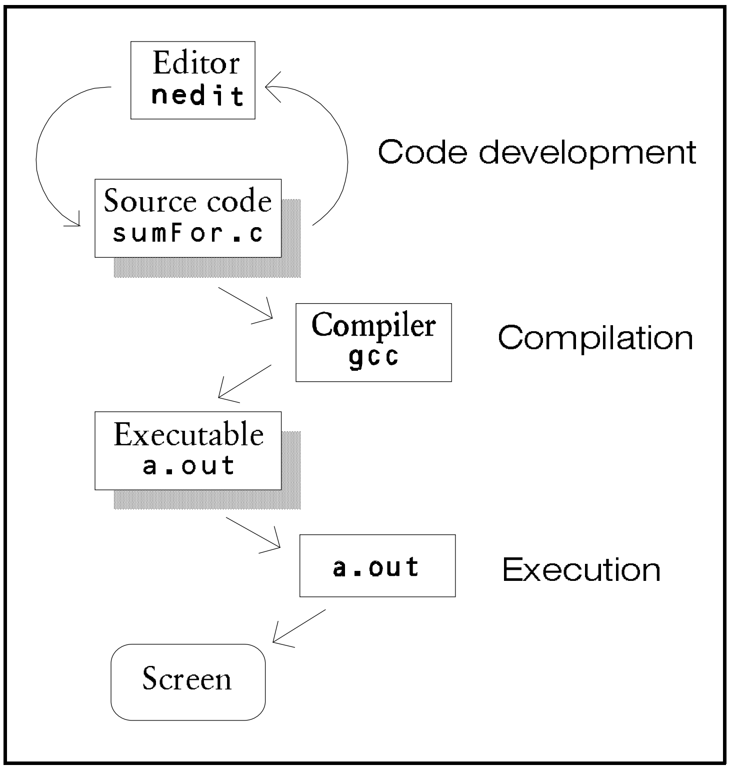
There are many other text editors available in Linux, and many of them are better suited to developing computer code than `nedit`. You may wish to try `vi`, `emacs` or `jove`. If you get stuck in an editor and need help exiting it, ask your demonstrator.

Compiling C programs

1 mark

Under DOS/Windows you used Borland C with an interactive development environment, so you could type and run your C programs in the same application. Linux does not have an interactive development environment, so you must edit, compile and run your programs using different commands. You have already seen how to edit programs using a text editor; here you will learn how to compile them and run them.

Once you have finished editing a program (the *code development* phase) you need to turn your source code into something that the computer can interpret directly. This process of converting source code into executable code is called *compilation*, and there is a compiler called `gcc` (which stands for the *GNU C Compiler*) which does this. When the program has been successfully compiled it may be run, or *executed*. Any output from the program then appears on the screen.



Compiling your program

Compile `sumFor.c` with the following command:

```
clay-b19g13-10> gcc sumFor.c
```

If there are errors, the compiler will tell you which line the error was found on. If the program was successfully compiled, a new file called `a.out` will be in the current directory. Run the program by simply typing:

```
clay-b19g13-10> a.out
```

Now compile and run `sumWhile.c`.

Compiler options

The `gcc` program has many varied options. Read the manual page for `gcc` and in particular take note of these options: `-o`, `-Wall`, `-g`.

Using floppy disks

1 mark

Quite probably you will want to do some work at home and bring your programs to your labs already typed in. Most programs that run under Linux can't easily access the floppy disk drive, but there are some utilities which allow you to transfer files back and forth between your home directory and an MS-DOS floppy disk.

Formatting a disk

You will need to format your disk using the MS-DOS `FORMAT` command. To do this, log into your Windows NT account and type at a DOS prompt:

```
L:\> FORMAT A:
```

> **!** **Don't format under Linux**
>
> The online documentation will have you believe that there is a program under Linux called `mformat` which will format an MS-DOS floppy disk for you. This program hasn't worked well in the past and probably shouldn't be trusted.

Listing the contents of a floppy disk

To show the contents of an MS-DOS floppy disk, insert it in the drive and type:

```
clay-b19g13-10> mdir a:
```

> **!** **Problems with `mdir`?**
>
> If `mdir` fails to locate your files on the floppy disk there is a chance that your disk is corrupted. Try putting it in another computer and seeing if the problem persists. If the problem is with your computer, ask your demonstrator to report the fault to Helpdesk.

Copying files to and from a floppy disk

The `mcopy` command works very much like the Unix `cp` command. Simply prefix "`a:`" to any file that is on the floppy disk. This command copies `myfile.c` from the floppy disk to the current directory:

```
clay-b19g13-10> mcopy a:myfile.c .
```

Look at the manual pages for these commands for more information.

> **!**
> ### Attention MS-DOS and Windows users!
> The text file formats for Linux and MS-DOS/Windows are very slightly different. To properly transfer text files between your home directory and a floppy disk you have used or intend to use with MS-DOS or Windows, you will need to include the `-t` option in your `mcopy` command. See the manual page for more information.

Transferring files with `ftp`

1 mark

Often you will need to transfer a file between one Unix machine and another. For instance, source code for this subject can be downloaded from the Computer Science FTP server, saving you much typing. `ftp` stands for *file transfer protocol*, the language with which you communicate with the other machine. *Anonymous FTP* is a way of getting files from a remote machine even if you don't have an account on that machine.

Connect to Computer Science's FTP server by typing the following:

```
clay-b19g13-10> ftp ftp.csse.monash.edu.au
(some text is omitted here)
Name: anonymous
Password: jrstu8@student.monash.edu.au (your full email address)
ftp> cd /subjects/cse1303/CS1P
ftp> bin
ftp> get hithere.c.gz
ftp> bye
```

Now decompress the file:

```
clay-b19g13-10> gunzip hithere.c.gz
```

Compile and run the `hithere.c` program that is in your current directory.

Try to copy this file onto your floppy disk.

> ### About `ftp`
> `ftp` is an important program to learn. You will be using it throughout your course. For more information, ask your demonstrator or see the document *Transferring Files using FTP* available from the IT Services Helpdesk.

Electronic mail *(optional)*

There are several mail-reading programs available on Linux. The simplest to use is `pine`. To learn how to use `pine` read its manual page. To run `pine` type:

```
clay-b19g13-10> pine
```

Try sending mail to your demonstrator saying you have managed to use `pine` successfully.

Using email on Linux

The laboratory Linux machines are not usually configured to receive mail. Thus you probably shouldn't use your Linux account as your main email account. Better alternatives are your Windows NT account, or the `yoyo` or `ra-clay` remote-access Unix computers.

Working at home *(optional)*

Since you have only a limited time in laboratories each week you will doubtless need to prepare in advance. It is helpful to be able to work at home on your pracs. Typing up the programs at home is not difficult, but in order to test them you will need the same working environment as Monash.

Installing Linux at home

Linux is a freely available operating system and if you are running MS-DOS or Windows at home you can also run Linux. The simplest way of obtaining Linux is to buy it on CD-ROM. If you want to install Linux at home you should read the installation guide, which is available via FTP:

```
clay-b19g13-10> ftp ftp.monash.edu.au
Name: anonymous
Password: jrstu8@student.cc.monash.edu.au (your email address)
ftp> cd /pub/linux/docs/linux-doc-project/install-guide
ftp> bin
ftp> get install-guide-2.3.ps.gz
ftp> bye
```

This is a compressed PostScript file. You can use `gunzip` to decompress this file and then either type `ghostview install-guide-2.3.ps` to view it on the screen, or print it to a PostScript printer.

Before you think of installing Linux

Installing an operating system is not for the faint-hearted. At the very least you will need to understand the hardware on your home computer and tasks such as how to back up and repartition your hard disk. Your demonstrator probably has Linux or another version of Unix on their computer. Talk to your demonstrator if you are not sure you have the expertise.

Installing SPIM at home

For you to be able to do your Part B pracs at home you will need to obtain a copy of the simulator, `SPIM`. `SPIM` runs under Linux, so you will need to install Linux on your computer if you want to use `SPIM`. You can get `SPIM` via FTP:

```
clay-b19g13-10> ftp ftp.cs.monash.edu.au
ftp.cs.monash.edu.au login: anonymous
Password: jrstu8@student.monash.edu.au (your email address)
ftp> cd /subjects/cse1303
ftp> bin
ftp> get README
ftp> bye
```

Read the README file for information on installing `SPIM`

!

Beware of the Windows version of SPIM!

There is a version of `SPIM` that runs on Windows. It contains a number of incompatibilities compared with the version for Linux that you use at Monash. You may choose to use it, but you should expect problems when transferring the work you did at home to Monash.

Modems

There is a computer that is set up exactly the same as the Linux machines that you use in labs. It is called `ra-clay` (the `ra` stands for *remote access*) and you can connect to it using `telnet` from any other computer on the Internet, including from home over your modem:

```
your-computer> telnet ra-clay.cc.monash.edu.au
```

✓

After this laboratory

You should:

- be able to log into and use your Linux account and X Windows
- be able to edit and compile C programs with Linux
- read and prepare for the next practical class