CS&IT
INDUCTION MANUAL
2013

including

UNIX
SURVIVAL
GUIDE

RMIT School of Computer Science and Information Technology
Introduction

Welcome to the School of Computer Science and Information Technology (CS&IT) at RMIT!

As a CS&IT student at RMIT you have access to a wide range of facilities that you will find useful whilst completing your program. Most of these resources are computer-based.

This handbook and the induction sessions introduce you to some of the facilities you have access to as a student, and have been designed to equip you with the necessary skills to utilise them properly. You will also become familiar with the terminology that is used here in CS&IT at RMIT.

The Quick Start Guide for new students at the beginning of this manual is to answer your basic questions about what you will be doing now that you are at University. Please take the time to read it and look at the services mentioned.

Your orientation begins with the Lab and Unix Induction which is a formal class, much like a tutorial session, giving an overview of the CS&IT network and how you will be using it. You will also be working with essential Unix skills, followed by some more advanced material. As CS&IT students you are expected to be able to use the Unix operating system for your coursework.

If you have any questions, make sure you have read the section carefully, you might find the answer yourself! However please ask if you would like some further explanation.

Complete all the Test Yourself tasks and you will be well equipped to study in CS&IT.

We hope that you find these sessions both useful and enjoyable.

Keep this manual as a resource to come back to during your program.

February 2013

Special thanks to the Duty Programmers for advice on the multiplatform iMacs and other academic and professional staff members for their earlier work on this manual.
Quick Start Guide for new students ..................................................... 1
Useful Contacts and Information Sources ................................................. 1
Your Program
Your Courses
Keeping in touch ....................................................................................... 2
Email
Discussion Board
Face to Face Communication .................................................................. 4
Course Consultation
Meeting the Academic Advisor
Administrative Questions
Student Staff Consultative Committees
Group Meetings
Learning Resources ..................................................................................... 5
Time
Classes
Study Skills

Computing Resources .............................................................................. 6
RMIT Information Technology Services (ITS)
NDS Username and Password
RMIT ITS Labs
After hours access ..................................................................................... 7
ITS Help
Wireless Networking
CS&IT Computer Resources ..................................................................... 8
Unix
Windows Terminal Server
Printing
Duty Programmers Office .......................................................................... 9
CS&IT Computer Labs ............................................................................... 10
Rules of Use ............................................................................................... 11

Accessing the Network ............................................................................ 12

RMIT Online Resources .......................................................................... 13
Student Email

Storage ........................................................................................................ 14
Accessing Unix Storage
Accessing Windows Storage ................................................................. 15

Using your laptop at RMIT ....................................................................... 17
Classroom laptop use
Quick Start Guide for new students

Now that you have enrolled in your degree program in Computer Science and Information Technology what will you need to be thinking about? The next few pages are to point you in the right direction and to help you feel like you know what is going on. There is much information on the web and in your enrolment pack about courses, administration and support services. Become familiar with where to find things now, then when you need them you will be ready.

The words in bold are terms used by the University if you do not understand them, please ask someone. The page references are for this manual.

Useful Contacts and Information Sources

**School office email enquiries**

csit@mrit.edu.au

**Academic Advisor**

csit.advisors@rmit.edu.au

The CSIT academic advisor provides program and course related advice for students enrolled in CSIT programs.

**RMIT website**

Please use the search function to locate information about RMIT processes and policies such as Special Consideration.

**CSIT pages on the RMIT website**

The easiest path into the CSIT student pages is through rmit.edu.au/compsci.
The tab for ‘Current Students’ will then take you to many useful information links.

**Duty Programmers**

CSIT.Helpdesk@rmit.edu.au

The Duty Programmers can assist with school-related systems.

**ITS**

itservicedesk@rmit.edu.au

ITS is the contact for RMIT systems such as password difficulties, and other RMIT systems.

**Study and Learning Centre (SLC)**

Search this name on the RMIT website or go directly to rmit.edu.au/studyandlearningcentre.

The SLC runs many workshops and also has helpful study and learning tips. Follow links on the site. You can also drop into the SLC, located just near the Hub in the Swanston St building.
Your Program

At RMIT, we use the term ‘program’ to refer the degree you are enrolled in. To complete your program in the expected duration, you would complete 8 ‘courses’ each year and normally enrol in 4 courses (or 48 credit points) in each semester.

Your Program Structure has the rules you need to follow to complete your program and graduate. It is used with Enrolment Online to select courses each year, and to make any changes at the beginning of the semester. You must complete the specified core courses (these are compulsory), courses from your specialisation or major study and electives. The order you do things is determined by the availability of courses and the prerequisites you have already completed. If you have advanced standing you will need to know the courses from which you have been exempted. Your Academic Advisor has this information.

Any questions you have about what you should do as part of your CSIT program are usually best directed to the CSIT Academic Advisor.

Your Courses

Each semester you will typically study four courses. They must be ones that contribute to your program and you choose them after you have had advice.

A course has timetabled teaching activities and learning resources both online and in print. The Course Guide details everything that is available to you and all that is expected from you, including how you will be assessed. You will find the course guide in the MyRMIT: Studies, or by searching for it.

The Course Guide has two parts, the Course Overview (part A) helps you select the course, and the Course Detail (part B) which gives specific information for this semester including the lecturer, textbook, topics and assessment due dates. You must read these.

Keeping in touch

Communication is the key to university study, you will do lots of listening and reading, but it is also important for you to talk and write.

As a student you are expected to be active in your learning and also to take responsibility for administrative matters. You will therefore need to be aware of deadlines, changes and opportunities. It is assumed by the School and the University that you are attending to your study and keeping in touch by accessing myRMIT regularly.
Email

The primary communication tool used by staff and students at RMIT and in CS&IT is **student email**. You will receive lots of messages; read them and archive them as you may need the information later.

There are different types of messages:

*Broadcast messages from RMIT are sent to all students in the form of announcements and bulletins. You may not be expecting anything as you have not asked a question, but it is important you scan through each message, some of the information will be relevant to you.*

*Broadcast messages are also sent from CS&IT to students enrolled in our programs. Again you didn't ask for anything but the information will be relevant to you because you are a student in the program. Read, make note of deadlines and activities, and act on the information.*

*Email from your lecturer or head tutor may be sent to everyone enrolled in the course. This is usually when something urgent needs to be said to all students.*

*Personal email will be sent to you as replies to questions or comments that you have sent to an administrator or lecturer. If you do not get a reply to an email, think of another way of communicating, perhaps talking after a lecture or attending consultation time. You might see the Academic Advisor if you have problems like this.*

*Personal email between students is an important learning tool, especially if you are working on a group assignment. If you do not hear anything from a group member you should try another way of getting in touch. If this fails alert your lecturer or head tutor.*

*It is your responsibility to check your email, and manage your mailbox.*

Discussion Board

Each course will have at least one area (usually in the Blackboard area of MyRMIT: Studies) where students can post and read messages about course work. All other students can then read and learn from the discussion. Many students are a bit shy about using this tool and only read. This is ok, but it is much better to ask a question early rather than wait for someone else. The discussion area is usually managed by the Head Tutor, but other tutors and the lecturer will add comments. You will need to learn their names so you know who the answer is from.
Face-to-Face Communication

While you are on campus talking with someone will often be necessary, there are a number of ways to do this.

Course Consultation

Time is set aside by each lecturer or head tutor to meet with students outside class time. It can be that you need to clarify part of a topic that you don't understand, or you want to check up on something related to assessment. The time and location for consultation is given in the first lecture and will be in Blackboard in the staff information area.

Contacting The Academic Advisor

The Academic Advisor is available for help with study skills, assignment and examination strategies, understanding RMIT University rules, personal worries, or if you just don't know where else to go. The email address is given on the ‘Useful Contacts and Information Sources’ List earlier in this booklet.

Administrative Questions

The Main Office for CS&IT is 14.08.03 and has an enquiry counter. Staff here will either answer your question or refer you to the Hub other relevant personnel. The CSIT office is open 9am-5pm on RMIT operating days unless advised in advance by office notice and/or student email notice.

Duty Programmers at 14.09.02 will answer technical questions and school-related systems. Contact details are given earlier in this booklet.

The Hub located on Level 4 of Building 12 is the place for all questions about being a student. The staff will either give you information or refer you to the correct service. The Hub also has lots of information about Student Services offered by RMIT.

Student Staff Consultative Committees

You are invited to become a part of the process for improving the learning experience for students in CS&IT. These SSCC meetings provide direct communication between students and staff of the school. All programs and year levels need representatives for these committees, the meetings are friendly and assistance and training for representatives is available. http://www.rmit.edu.au/seh/students/sscc. Students are recommended to participate training through RMIT LEAD (www.rmit.edu.au/lead) as part of involvement in the SSCC.

Group Meetings

Students like to work together and finding a space can be a challenge. The Swanston Library has eight discussion rooms equipped with computer and white board which can be booked for up to two hours per day for group assignment work or informal study groups. www.rmit.edu.au/library.

The Student Lounge on Level 9 of Building 14 has tables and chairs and is useful for informal discussion.
Learning Resources

Here is a quick look at the tools and activities you will engage with while you are studying in CS&IT.

Time

The **Important Dates** has the semester and exam dates and deadlines for making changes to your enrolment. These deadlines are final.

http://www.rmit.edu.au/browse/Current%20students%2FImportant%20dates/

Your **Timetable** is your personal schedule of classes during the week. see pages 26, 30

Once you have filled in lectures, labs and tutorials you will need to schedule your personal study and coding practice time. Allow a minimum of **12 hours for every course** you are doing. Full time study is a commitment to 48 hours per week; it is recommended that you make this your priority. Paid work can then fit around your study, perhaps a day on the weekend or two afternoons during the week. The semester break is a good time to do extra paid work and get money in the bank.

A **Study Planner** is the whole semester's work at a glance. You fill in the weeks with course topics, assessment starting and due dates as well as progressive submission dates. The Student Union produces a free planner for members or you can draw up your own. Some lectures make a planner for their course with week by week links to lecture, tutorial and lab tasks. Your job is to combine them all so you know what you are doing each week, and also planning for busy times when a lot of assignments are due.

Classes

You are encouraged to attend all the classes for each course. Your lecturer assumes that this is what you are doing, and that you come to class prepared for that topic. This means reading the **lecture notes** and text book chapters before the lecture. The tutorial will challenge you with questions related to the topic from the previous week. This gives you time to prepare your answers, to think about the ideas, and to try the programming examples. Lab classes often relate to assignment work. Key concepts and skills that you need for the assignment are presented in exercises. Once you have completed them successfully you should be able to work on that part of the assignment.

If you miss classes because of something unavoidable such as ill health, you should meet your lecturer or head tutor during consultation time to get advice on how to quickly catch up. Even if you are ill, you can often keep in touch with the online elements of your course. However serious illness requires advice regarding **Special Consideration**. The Academic Advisor and Student Services can help.

If you get behind with topics and assessment, tell your lecturer, tutor or the Academic Advisor as soon as you can. Don't just try and catch up weeks and weeks of work without advice.

Study Skills

The RMIT Study and Learning Centre at the Hub provides free assistance in English language, study skills, academic writing, and Maths. There are online study skills tutorials called the Learning Lab. International students will find interesting interviews with students discussing the differences they found at RMIT.

http://www.rmit.edu.au/studyandlearningcentre
Computing Resources

RMIT Information Technology Services

A summary of ITS services can be found in the brochure IT@RMIT distributed at enrolment. The ITS website has full and up to date information.

NDS Username and Password

All RMIT students are assigned an RMIT-wide username and password that can be used to access protected parts of the RMIT network, such as the student email system.

Your NDS username is the letter 's' followed by the numeric portion of your student number. For example, if you had a student number of 3012345, your NDS username would be s3012345.

Your initial password is the letter p followed by your date of birth in the format YYYYMMDD. For example, if you were born on the 21st of March 1980, your password would be 'p19800321'. You should change this as soon as possible after first logging in. Your password must meet RMIT criteria.

• Be at least eight (8) characters in length
• Be a maximum of 25 characters in length
• Contain at least one lower case letter (a,b,c,….,)
• Contain at least one numeral (0,1,2,3,…,9)
• Has not been one of your previous 15 passwords

You are also advised to set a challenge question so you can retrieve a password if you forget yours or if it expires.

RMIT ITS Labs

Building 8 Level 3
Hours: Mon to Fri 8.00am to 6.00pm

The Hub Building 12 Level 4
20 Pentium PCs
Hours: Mon to Fri 9.00am to 6.00pm

Learning Centre 14.04.12
This lab is open 24 hours a day seven days a week.
General use Monday to Friday from 8am to 6pm.
At all other times, each user must have an authorised 24 hour access card to enter and use this facility

PostGraduate Lab Building 28 Level 4
Post Graduate students only
20 Pentium PCs
Access with RMIT security perimeter access card,
Lab available 24 hours seven days a week

Library Computer Facilities
PCs and Macintosh workstations
Printing and Scanning during Library hours
Powerpoints available for personal laptops
Some PCs may be booked in advance using online booking
http://www.rmit.edu.au/library
Gaining entry to labs after hours

All students wishing to use the 24 hour labs, must obtain an RMIT security perimeter access card. The request form is available at the school office 14.08.03 Post Graduate and Honours students can also request 24 hour access to selected CS&IT Labs.

ITS Help

Staff in the Building 8 Level 3 lab can assist with resetting NDS logins, software assistance and Wireless for laptops.

www.rmit.edu.au/its/ithelp

Email: helpdesk@rmit.edu.au Ph: 9925 8888

Wireless Networking

The wireless network has become an integral part of the RMIT IT network, offering RMIT students and staff the freedom to work from a variety of indoor and outdoor locations. RMIT staff and students can now access RMIT IT services including the internet, RMIT intranet, RMIT University Library online content, email, their RMIT home drives, myRMIT, Enrolment Online, and the Student Timetabling System (STS).

RMIT will continue to develop and expand the wireless precinct.

For further information see: www.rmit.edu.au/its/wireless

For information and advice about using your own laptop, scroll down to the “Using your own laptop section.”
CS&IT Computer Resources

As a student in the School of Computer Science and Information Technology (CS&IT) at RMIT you are granted access to computer resources by means of individual student accounts.

You have been provided with accounts on two different types of network server, each with a different operating system:

### Unix

Your Unix accounts provide you with access to web browsing, e-mail, Internet news and a host of other facilities which you will find useful whilst you are completing your course.

A large part of the work you will be required to do in the course can be done using your Unix account, especially for programming subjects.

Students in CS&IT have accounts on 2 Unix servers: Yallara and Numbat

Students may also have accounts on other Unix machines specific to certain courses. Such accounts will be detailed by the course lecturer.

### Windows Terminal Server

These machines allow students to use Microsoft Windows and standard Microsoft applications such as Word and Excel.

They also provide indirect access to your Unix Accounts as well as other useful applications such as Mozilla Firefox, which allows you to access the World Wide Web (WWW).

Windows Terminal Servers are named Blowfly.

### Printing

There may come a time in the course of your study that you will need to print documents course work or other material related to your study. Each student is allocated a print quota at the beginning of the year, additional credit can be purchased at the Hub.

To print, first create a print job (normally file → print). This will send your document to the RMIT print cloud. The next step is to locate a printer.

14.09 & 14.10 outside the labs.

When you have successfully located a printer, you can use your student card to bring up available jobs.

*If you are experiencing any issues printing, please contact the duty programmers.*
Duty Programmers Office

Location: Room 14.09.02  
Email: csit.helpdesk@rmit.edu.au  
Phone: 9925 1994  

Hours: During the semester.  

Monday – Friday  
9.00 am – 11.00 am  
1.00 pm – 3.00 pm  
5.30 pm – 8.00 pm  

Saturday / Sunday  
Closed  

Opening hours are reduced in vacation periods.

The Duty Programmers are the student’s first point of contact for any technical problems or support for the resources in CS&IT.

Typical problems that the duty programmers tackle daily are password changes, new account set-up, software issues, and printer servicing and fixing jams etc. If a printer is not working, report it to the Duty Programmers and they will attend to it immediately. Duty Programmers are also able to answer questions regarding programming, they are not there (of course) to complete your assignments. From the website, students are also able to access the online help desk documentation.

You can also borrow CDs of software relevant to your course from the help desk. If you provide a blank CD, the duty programmers can copy free software or your coursework files onto it for you. The available software includes Linux Distributions, a Windows Software CD (Freeware Applications) including Eclipse, PuTTY. If you do not have a broadband connection this is the best way to get the resources you need.


The Duty Programmers cannot help with Blackboard or Weblearn issues, you should direct any of these to your course lecturer, particularly if you cannot access your course material.

The provision of this service may change during 2013
CS&IT Computer Labs

There are two types of general-purpose machines in the labs in CS&IT. These are PCs and Multiplatform iMacs. The PCs are generally a standard Pentium-based computer with a CD-rom. A Multiplatform iMac is a dual boot computer that can run either a Microsoft Windows XP or a Mac OSX session. These will be configured to be triple boot during 2012.

Lab opening times are weekdays 8:30am to 9:30pm. Opening times will change during holidays as announced on CS newsgroups. Postgraduate students may obtain 24 hour access to CSIT labs and Undergraduate students can use the ITS Learning Centre. Forms to apply for the RMIT security perimeter access card are available at the school office 14.08.03.

CSIT Labs

PC Labs:
- Knuth 14.10.31
- Oasis 14.10.31a *(Mentoring, during scheduled hours)*
- Lions 14.10.30

Multiplatform iMac labs
- Babbage 14.09.23
- Bolam 14.09.15

Specialist labs
- Beard *(Linux Lab)* 14.10.32
- Games Studio *(Games Degree)* 14.11.37
- Alice *(Games Degree)* 14.11.36
- Sutherland *(Graphics lab)* 14.11.38
- YourSoftware *(PG Industry Projects only)* 14.10.35

Except where specified, and provided that a class is not scheduled, all labs are available for free access. You may also be able to use a terminal in a lab while a scheduled class is present, provided that the supervising lab assistant permits it, however you will be expected to work quietly so that the lab class is not disturbed.

CS Tutors and Lab Assistants are authorised to request that students not allocated to a lab class remove themselves from the lab even if the lab is not full to capacity. Please be accommodating if requested to vacate the lab.
**Rules of Use**

You will have signed and agreed to the RMIT CS&IT terms of use upon signing up for your computer account at enrolment.

Specific rules for the labs include:

- Access to CS&IT labs is restricted to only those who are enrolled in CS&IT courses. Duty Programmers, security or any staff member of RMIT University may ask for identification.

- No food or drinks are to be consumed or brought into the labs. This rule is enforced vigorously by staff to avoid damage to the machines, the fixtures in the labs, and to reduce the amount of cleaning required.

- CS&IT labs are for students to complete their work and for research purposes. They are not to be used as general gathering areas. Because labs are spaces shared by many people, those using them should be courteous to others and refrain from excessive noise.

- Terminals and other equipment are to be used for course related work only. Chat, games and accessing inappropriate websites are forbidden.

- Students are not permitted to access websites that involve downloading extremely large files except where this is directly related to course work. YouTube and Google Video are examples of such sites.

- You may not connect other equipment including laptops to power or networks in labs. This is strictly enforced to protect resources for all students.

Make sure you know these rules well, and observe them carefully. Failure to observe the rules can result in your account being suspended.

Online copies of the rules and policies are available at www.rmit.edu.au/compsci/policies/rulesofuse

This page is only a brief summary and is not a substitute for the full documents. A list of activities that will result in account suspension is found at the back of this manual.
Accessing the Network

There are two types of computers available in the labs. The multi-platform iMac machines found in 14.09 labs and the Windows machines found in the other labs.

The multi platform iMacs run both Windows 7 and OSX.

*When using the machines provided in the labs please observe the rules of use.*

**Logging In (Multi-platform iMac)**

The terminal you are sitting at is a multi-platform iMac which allows you to access your Windows account, as well as accessing your UNIX account on Numbat and Yallara.

If your screen is blacked out then your terminal is probably in power save mode. Move the mouse around to power it up, or, if that fails, you will need to turn the computer on. The power switch is to the bottom left behind the screen.

The iMac machines run both Windows 7 and OSX. To change operating systems simply restart the computer, or if off turn it on and select the desired operating system at the chooser screen.

**Logging In (Windows)**

In order to use the Windows based facilities here in Computer Science you must first log into your Windows account.

To login press ctrl + alt + delete, then need to enter your NDS user name and password, the same one you use for myRMIT.

**Logging In (OSX)**

In order to use the OSX based facilities here in Computer Science you must first log into your OSX account.

To login need to enter your NDS user name and password, the same one you use for myRMIT.
RMIT Online Resources

myRMIT  
www.rmit.edu.au/myrmit

This online student portal gives you a personalised snapshot of your information across a number of RMIT services including student email, library loans, RMIT announcements, your online content each course, and access to all RMIT student systems including Enrolment Online.

With a single signin myRMIT allows you to quickly check all important RMIT information and avoid multiple logins. You must remember to log out of any windows to secure areas after have finished as well as logging out of myRMIT.

Student Electives  
http://www.rmit.edu.au/students/studentelectives

Student Timetabling System (STS)  
http://www.rmit.edu.au/students/sts

RMIT Libraries  
www.rmit.edu.au/library

Student Email

Every student enrolled at RMIT has an RMIT student e-mail account. This is used for official RMIT correspondence; you must check this email account regularly. RMIT student email service is provided and maintained by Google. To access the system, visit the URL: https://mail.google.com

Your Google username is your student number followed by @student.rmit.edu.au  
e.g. s1234567@student.rmit.edu.au

Your default Google password is the letter ‘p’ followed by your date of birth  
e.g. 28 April 1995 is p19950428

You must use your Student Email account for all communication with RMIT staff - you may not receive a response otherwise.

You may already have another personal gmail account. Be careful: you must log into gmail using above mentioned student email address when you communicate with RMIT staff. It is not recommended to forward emails back and forth between your personal account and RMIT student account.

In addition to email, Google Apps provide a set of communication tools that makes communicating and collaborating easier and more efficient.

For frequently asked questions on student email and Google Apps, visit:  
http://www.rmit.edu.au/students/services/googlefaqs
Storage

RMIT provides two places to store your work in addition to your student email.

The first option is the UNIX servers named Yallara and Numbat which can be accessed via the SFTP protocol.

The second option is the windows server named Blowfly that is automatically mounted to H: when logging in.

Accessing UNIX Storage

An RMITCS student will, as a general rule, have 50MB of initial storage quota. In addition, programs in the SET portfolio receive an extra 12MB of quota, and a student will receive a further 12MB of quota for each course they study in RMITCS.

This means that a typical (full-time) computer-science or IT student will have (50 + 12 + 4*(12)) = 110MB quota.

Certain courses receive a quota bonus if the course content necessitates it.

The servers Yallara and Numbat are mirrored, this means that if you store a file on one it will also be on the other. Depending on how many students are using each server, it is recommended to try either Yallara or Numbat as performance may differ depending on load.

In addition to storing files on these servers you can also remotely login and execute code and edit text files as well as performing other tasks. This will be explained below in the UNIX induction guide.

To transfer files from the desktop to the UNIX servers we use a program called FileZilla. This is an FTP/SFTP program.

To start FileZilla press the start button located in the bottom left of your screen type “filezilla” and click on the short cut to launch the application.

If you are an advanced user you may use any SFTP program.
Once the application loads you will be presented with a screen.

The local computer’s files are listed on the left, while the remote server’s files are listed on the right.

To connect to **yallara** or **numbat** servers you must first login.

To login simply fill in the information at the top of the application and press **Quickconnect**.

| Host:       | yallara.cs.rmit.edu.au  |
|            | or                      |
|            | numbat.cs.rmit.edu.au   |

| Username:  | NDS Username            |
| Password:  | NDS password            |

| Port:      | 22                      |

Once connected you can simply drag files from the left (local computer) to the right (remote server) to copy them to the remote server.

**You can also access the servers from home** if you have an active internet connection. FileZilla is a free application and can be downloaded from [http://filezilla-project.org](http://filezilla-project.org)
Accessing Windows Storage

After logging in to your Windows account, your windows storage folder is automatically mounted to H:/ drive. This provides a simple easy to use storage location for your course work.

You can access your windows storage from home by the web interface.

For more information on remote access
http://rmit.com.au/browse;ID=tjda4cll2fmy

As most courses require your work to run on yallara or numbat, it is recommended to use the Unix storage instead of the Windows storage.
Using your own laptop at RMIT

A wireless enabled Macintosh, Windows or Linux-based laptop computer or wireless-enabled mobile device such as an iPhone, iPod Touch or Pocket PC is needed to access the RMIT wireless network. RMIT Wireless access is available at many inside and outside locations around RMIT, at the State Library of Victoria and the level 2 food court at Melbourne Central.

Refer to the ITS wireless website for information on how to connect for the first time. www.rmit.edu.au/its/wireless

You may not plug into power while you are in the lab. Plugging in a device is a violation of the Acceptable Use Policy and will result in your CS Account being suspended.

Your access to Yallara or Numbat will be the same as if you were logging in from home.

Wireless at RMIT is not related to CS&IT and does not have access to CS resources. This means that you cannot print directly from your laptop or mobile device. The only way to print from your laptop is to SCP the files to your Yallara account, SSH into Yallara, and print them from the Yallara command line.

Any wireless network problems need to be referred to ITS.

Classroom Laptop use

If you are experienced with using a laptop in class, you may find it useful to take yours to lectures and tutorials. Remember to have a fully charged battery as there is no facility for recharging or power connection, and not all lecture theatres have wireless access.

A skilled user can try code examples as the lecturer demonstrates, or look up a reference, but it is also easy to become distracted. Be aware of those around you so you do not disturb other students or your teacher. Viewing material not related to the lecture is very disturbing to students around you even when there is no sound.

You will find it more effective to use the terminals in the labs for lab or lab/tute classes, as you will need specific software and full access to the network and your yallara directories.

If you use a laptop for assignment work, remember to make regular backups to yallara, not just when you have finished. Losing your work because your laptop or usb stick is lost, broken or not working properly is not an acceptable reason for an extension of time.

A usb memory stick, cd or even email can be effective ways to transfer work from your personal computer to your directory on yallara or numbat.

Remember your home directory on the CS&IT network is regularly backed up. It is the safest way to protect your work.

You should also pay attention to assignment specification advice about making sure your code works on the CS&IT network. Upload your work well before the due date and test that it runs. Your teacher will heavily penalise any work that does not run, or has to be modified before it can be marked.
UNIX
SURVIVAL
GUIDE
Unix

While the Windows PCs and terminal servers are available for you to use, most Computer Science assignments need to be completed on the Unix servers (Yallara and Numbat).

What is Unix?

There are actually many operating systems that are “Unix-like”. Both Yallara and Numbat run Sun Solaris. You may have heard of some others: Linux, FreeBSD, Mac OS X, SCO, and so on. While internally these operating systems are very different, they all share a common interface, meaning the skills you learn here are generally applicable to all Unix-like operating systems.

If you have used Linux or another Unix-like operating system before you may already be familiar with many of the commands introduced here; be aware, however, that there are some commands that are either Solaris or RMIT specific.
Logging In (from a Windows Operating System)

If you are connecting from a Windows operating system we must use a terminal emulator. The most popular program and the one installed at RMIT is called putty.

To launch putty, click on the start button located on the bottom left of the screen. Type putty and press enter.

Putty will now launch. To connect to the UNIX servers, in the field “Hostname or IP address” type the address of the server you wish to connect to.

Yallara:

  yallara.cs.rmit.edu.au

Numbat:

  numbat.cs.rmit.edu.au

Then press enter or click Open. If this is the first time logging in with PuTTY you will be presented with a security alert. This alert lets you know that it is the first time connecting from this computer to the server. To continue click “Yes” and then you will be prompted to login.

The credentials used to login are your NDS username and password.

*If you are experiencing issues logging in please go to the duty programmers desk or ask your tutor.*

Once logged in, you can scroll down to using the terminal and continue with the UNIX induction.

If you would like to connect from home you may download PuTTY from the following link.

Logging In (from a “Unix-Like” Operating System)

If you are connecting from a “unix-like” or “unix” operating system, such as OSX you simply launch a terminal. It is built into the operating system.

**OSX:** Top right corner click on spotlight, type “terminal”

**Linux (gnome):** Press alt + f2 and type “gnome-terminal”

This will launch a terminal. A terminal is a command line interface that is capable of executing local system commands and connecting to remote servers.

The command used to connect to yallara or numbat is known as SSH. It requires two pieces of information to connect to the server. The hostname or IP address of the server you wish to connect to and the username of the user you wish to login as (nds username).

**Hostnames:**

**Yallara:**

\[\text{yallara.cs.rmit.edu.au}\]

**Numbat:**

\[\text{numbat.cs.rmit.edu.au}\]

An example of the command used to connect to Yallara.

\texttt{ssh s1234567@yallara.cs.rmit.edu.au}

An example of the command used to connect to Numbat.

\texttt{ssh s1234567@numbat.cs.rmit.edu.au}

Make sure to substitute your username with the example username s1234567.

If you are experiencing issues logging in please go to the duty programmers desk or ask your tutor.

Once logged in, you can scroll down to using the terminal and continue with the UNIX induction.
Using the terminal

When you first log into Yallara you are presented with two xterms. Remember you can always open another terminal from the start menu if you need to; it is often convenient to have a few open when doing several tasks at once.

After the welcome message is the prompt, which lets you know that the terminal is ready for input: yallara.cs.rmit.edu.au%

In this manual we usually don’t show the prompt; just the commands you need to type. For example, try typing “date” and pressing enter:

    date

This runs the program “date” and displays the output to the terminal.

Most commands accept arguments, or parameters, which modify their behaviour. For example, if you type:

    date -u

    Sat Feb 11 14:17:57 GMT 2006

The “date” program is run with the argument “-u”, which then displays the time at GMT (a different timezone). This time we have also shown the result, in a smaller, italicised font. When you need to repeat a command with the same or similar arguments, you can save typing by pressing the up-arrow on the keyboard. This shows the last command you entered. You can continue pressing the up and down keys to move back and forth through the history of commands you have typed in the current session.

Getting help

When you know the name of a command, but can’t remember what arguments it takes, you can look it up in the man pages (short for manual pages). For example, to find out all the options for the date program:

    man date

You can scroll through the page with the spacebar. To quit viewing the manual page and return to the prompt, press q.

If you don’t remember the name of the command, man won’t help. You will probably need to ask a friend, lab assistant, use this manual or Google.
Files and Directories

Most people are familiar with Windows and DOS drive letters where disks are accessed as C:, D:, and so on. In Unix there is no such distinction. All files, regardless of where they are stored, are accessible through the root directory or “/”. Whereas on Windows directories (sometimes called folders) are separated in a path with a backslash (“\”), on Unix you must use a forward slash (“/”).

Here are some example directory paths:

/  
/home  
/home/stud  
/home/stud/s1234567  
/usr/local/bin

Paths are read from left to right. The first slash (“/”) means to start at the root directory (you can start in some other places as well, as we shall see shortly).

You can use the ls command to list the contents of a directory:

```
ls /
```

This will print out all files and directories directly below the root directory.

Some of these directories and their purpose are explained below:

```
/bin    Standard Unix programs  
/etc     Configuration files  
/home    Students' home directories  
/public   Academic material for specific courses  
/tmp     Temporary files
```

Look inside the /home directory

```
ls /home
```

This directory contains every student's personal home directory organised by the first letter of their username. For example, if your username is s1234567, your home directory is /home/stud/s1234567. Try to list the contents of your own home directory now. Your home directory is the place where you should store any assignments you are working on, email, and is where applications can store your preferences.

As a shortcut to writing out your whole home directory path, you can simply write ~ (tilde
character, in the top-left corner of the keyboard). Assuming your username is **s1234567**, the following two commands are equivalent:

```
ls /home/stud/s1234567
ls ~
```

What happens when you try to list another student’s directory?

```
ls /home/stud/s3239139
```

You are sharing the computer Yallara with all other Computer Science staff and students, but there are systems in place that prevent you from reading or modifying other people’s data, as well as data that could interfere with the upkeep of the system. Permissions are discussed in detail later.

---

**Test Yourself**

What is the path to your home directory?

What files and directories exist in your home directory? Can you guess what they are?

Log into **Numbat** now. Can you see any differences in either your home directory or the root directory? Why do you think that might be? Return to **Yallara** when you are done.

---

**Working directory**

Every terminal window you have open has a *current working directory*. This is the directory that applications will load and save their data to or from by default. You can print the entire path to the working directory with the **pwd** command:

```
pwd
```

If the directory or file in which you are interested is in the current working directory, you don’t need to specify a complete path to it. For example, you can list the contents of the current working directory by typing **ls** without any arguments:

```
ls
```
Creating directories

There is not much interesting in your home directory yet, so let’s create some directories. To create a directory, use the `mkdir` command. Let’s say you want to create a directory called “courses” within your home directory:

```
mkdir courses
```

Remember that since your current working directory is your home directory you didn’t need to type in the whole path. Of course, you could have if you wanted to; the above is equivalent to:

```
mkdir /home/stud/s1234567/courses
```

List the contents of your home directory now and make sure you can see `courses` as one of the items.

Let’s assume you are taking 3 courses: “maths”, “programming” and “databases”, and create a directory under `courses` for each one:

```
mkdir courses/math
mkdir courses/programming
mkdir courses/databases
```

Now list the contents of `courses` and make sure you can see all of these directories.

```
ls courses
```

Changing the working directory

Earlier we saw that the current working directory was your home directory. Let’s now change directory to the “courses” directory:

```
cd courses
```

Now that the working directory has changed, what do `pwd` and `ls` do?

Create one more directory under `courses` named “induction”:

```
mkdir induction
```

Note: that we didn’t write `courses/induction` this time, as we are creating the directory directly within the current working directory. List the contents of the current directory and make sure you now see all 4 directories.

```
ls
```

The layout of directories is often referred to as a directory tree, and is displayed like this:
In this diagram you can see that the s1234567 directory is the parent of courses, which in turn is the parent of the four subject directories you created. We changed directory from s1234567 to courses by typing

```
cd courses
```

You can't, however, change back to s1234567 from courses by typing cd s1234567; this is because s1234567 is not visible from courses. You can change back to s1234567 by typing its full path:

```
cd /home/stud/s1234567
```

Alternatively you can change to the parent directory with “..” (two full-stops, commonly pronounced "dot-dot"):  

```
cd ..
```

The “dot-dot” can appear anywhere in a regular path to signify “the parent”:

```
cd /home/stud/..
pwd
cd /home/../../public
pwd
```

```
Test Yourself

Write down the absolute path of the following paths
(an absolute path is one that begins at the root directory):

```
courses/
courses/..
../
../..
courses/../.courses/../.courses
```

Create two further directories under courses/maths named “calculus” and “algebra”.

Change directory to algebra, from there list the contents of courses.

---

Tab completion

When typing the names of files or directories on the command-line, you can often type just the first few characters, then press the tab key to fill in the rest automatically. For example, you can save a lot of typing when changing to the courses/maths directory by just typing:

```
cd c(tab)m(tab) becomes cd courses/maths/
```

---

Hidden files and directories

Change to your home directory and list all the files. Now add the -a option to ls:

```
ls -a
```

You should see about 20 extra files and directories that weren’t in the standard listing. These are hidden files, and have a full-stop (”.”) as the first character in their name. Typically they are used to store application preferences and caches.

Create a hidden directory in courses named “.hidden”:

```
mkdir courses/.hidden
```

Make sure you can see it only when you use the –a option with ls.

At the beginning of the listing of hidden files in each directory are the two special directories “.” and “..”. We already know that “..” refers to the parent directory. The “.” (“dot”) directory refers to the current directory. Ordinarily this is not needed on the command line, but you may find you need it at some stage.

Check now that the following three commands are equivalent:

```
ls courses ls
./courses ls
./courses/.
```
Checkpoint directory

There is one more special directory that exists on the file server used by Yallara and Numbat. It is named "\texttt{.ckpt}" and does not even show in a \texttt{ls -a} listing. You can, however, see inside it:

\texttt{ls .ckpt}

The "checkpoint" directory contains backups of the current directory that are made automatically at regular intervals. For example, you might see a directory called \texttt{.ckpt/2012_02_05_13.15.01\_GMT}, which is the backup that was made on the 5th of February at 1:15 PM, GMT. If you list the contents of that directory you will see the files that existed in your home directory at that exact time.

This automatic backup is a feature of the RMIT Computer Science servers and is generally not available on standard Unix systems. Backups are not kept forever, so you should make a habit of continuing to backup your work both at home and university onto CDs and storing them away from your computer.

\textit{Note that you cannot modify files or directories in the checkpoint directory, but you can view them and make copies back into your regular home directory.}

Creating and editing text files

Many files you work on in your Computer Science subjects will be plain text files (they have no formatting or fonts like a Microsoft Word file, for example). There are many programs you can use to edit these files. One very powerful and flexible editor is \texttt{vim}, which is introduced in the next section of this manual. Regular practise is recommended so that you become proficient editing using vim.

For now, however, we will use a much simpler editor called \texttt{pico} which only runs in xterm. Another editor you might use is pico.

Change into the \texttt{courses/maths} directory and start editing a new file called "assignment1":

\begin{verbatim}
  cd courses/maths
  pico assignment1
\end{verbatim}

Pico behaves much like any Windows text editor, though very basic. The commands for pico are listed at the bottom of the terminal window. The ^ represents control, for example to save a file press CTRL + O (write out). Pico will prompt you to confirm the name of the file. Press enter to confirm. A file is now created with the name assignment1, to exit Pico press CTRL + X.

To check if the file was successfully created use the list command.

\texttt{ls}

To display the contents of the file in terminal use the command \texttt{cat}.

\begin{verbatim}
  cat assignment1
\end{verbatim}
Renaming, moving, copying and deleting files

You should now have a file assignment1 in the maths directory. Let’s say you wanted to rename it to “assignment1.txt”:

```
 mv assignment1 assignment1.txt
```

The first argument to mv is the original file name, the second is the name you would like it renamed to. mv won’t actually output anything; you will need to list the current directory contents with ls to check that the result is what you expected.

You can use the same command to move a file to another directory:

```
 mv assignment1.txt ../induction
```

This moves the file assignment1.txt to the directory ../induction. Remember that the double-dots ("..") indicate to start from the parent directory, which in this case is courses. Similarly, you can move entire directories with the same command:

```
 mv ../induction ~/
```

This moves the induction directory to your home directory (remember that the tilde ~ is short-hand for your home directory).

The cp command works similarly, except that instead of moving or renaming a file it makes a copy:

```
 cp ~/induction/assignment1.txt ./
```

This makes a copy of assignment1.txt and places it in the current working directory (remember that "." means to start from the current directory), which if you have not changed it is the maths directory.

If you want to copy an entire directory you need to specify the –r argument to cp:

```
 cp -r ./ ../maths-backup
```

This copies the current working directory (maths) to the directory ~/courses/mathsbackup.

To delete a file, use the rm command:

```
 rm ~/induction/assignment1.txt
```

To delete an entire directory, add the -r argument to rm:

```
 rm -r ../databases
```

**Careful!** The automatic backups only take place every few hours; if you delete something you need there may be no way to get it back if it’s too recent. The mv and cp commands won’t give you any warning if you overwrite another existing file.
Test Yourself

Rename the maths-backup directory to make it hidden. Hint: how does the name need to change to make it not show up in a normal directory listing? Check with \texttt{ls} that it doesn't show up, then show that is there when you supply the appropriate argument to \texttt{ls}.

Create a text file in the induction directory listing all the commands you have learnt so far and call it \texttt{notes.txt}.

Make a copy of this file (\texttt{notes.txt}) in the same directory and call it \texttt{commands.txt}.

Find a file in the checkpoint directory (the automatic backup directory) from two days ago and copy it into the \texttt{induction} directory.

Delete the original \texttt{notes.txt} file.

In the space below, write out the whole directory tree starting from your home directory and including all the files and directories you have added.

---

A note about Unix file name conventions

You will have noticed that most of the directory and file names given in this manual are composed entirely of lower-case letters. This is entirely optional: Unix systems allow filenames to have any form of letter, including most punctuation marks and characters from non-English character sets. Unlike Windows and DOS, however, Unix treats upper- and lower-case letters as different. In other words, you can have a directory containing the files \texttt{test.txt}, \texttt{TEST.TXT} and \texttt{Test.TXT}, and they would all represent different files. As you can imagine, this can get quite confusing.

For the sake of simplicity and clarity, most users elect to name their files entirely in lowercase English letters, with the addition of the hyphen ("-"), full-stop ("."), and underscore ("_") punctuation marks. Some programs may not work correctly with other punctuation marks in the filename, as they have special meaning to the command environment, as you will see.

You will have also noticed that unlike Windows, many files do not have an extension (like \texttt{.txt}). Again, these are optional, however they can help you to organise your files and let programs know what kind of data to expect.
Using Windows files on Unix

Often you will want to work on an assignment at home on Windows, then copy it back to Yallara for submission (see page 17 for tips on copying files from home). There is a sample Windows file at /public/induction/windows.txt. Copy this file to your induction directory, and change to that directory:

    cp /public/induction/windows.txt ~/induction
    cd ~/induction

If you display the contents with cat it appears to be fine.

    cat windows.txt

Now try using cat with -v argument.

    cat -v windows.txt

What’s wrong with the text? Can you see the extra ^M characters at the end of each line? In Windows and DOS, each line of a text file is terminated by two characters: a carriage return (CR, code 13) followed by a line-feed (LF, code 10).

In Unix, lines in a text file are terminated with just a line-feed. Some Unix programs (like nedit) can handle both types of file, but most will not work correctly with the Windows style line-endings.

In cat -v, the carriage returns appear as ^M characters, which is harmless but annoying.

In Perl (a programming language), however, a program written with these characters will just fail to work. This can be quite a shock to a student who has worked all weekend at home on an assignment only to find it does not work at all at uni.

Luckily, the solution is simple: simply remove the carriage returns from the file. You can use the dos2unix program to do this:

    dos2unix windows.txt unix.txt

This converts the file windows.txt and saves it as unix.txt (It will print out a 2-line warning about an unknown keyboard layout which you can safely ignore). Open this file in gvim and check that the lines now look correct.
Filename globbing

List the contents of the /public/induction directory. There are a series of files ending in .ant and .syn. These are lists of words that are antonyms or synonyms of the filename, respectively.

If you were to copy all of the synonym files (those ending in .syn) into your current directory, you would need to type the cp command 8 times. Actually, there is an easier way:

    cp /public/induction/*.syn ./

The "*" character (an asterisk, commonly called a "star") is a placeholder, or "globbing operator" for any sequence of characters. You can use it in place of part of a filename where you want to list all the files that match the pattern.

It works on all commands, not just cp:

    ls /public/induction/*.ant
    ls ./d*

Warning: using a glob is equivalent to typing out all the filenames it matches in sequence. This is not always intuitively what you want. Consider what happens when you type:

    dos2unix *.txt

This is equivalent to typing (assuming there are two files in the directory):

    dos2unix file1.txt file2.txt

Instead of converting all the files in the directory, you have overwritten file2.txt with file1.txt! Later in this manual we will write a script that can convert all the files in a directory.
Finding text within files

You should now be in a directory with several .syn synonym files. If you look at these files with nedit you can see that they are just simple text files with one word per line. You can search a file for a word or phrase with the `grep` command:

```
grep perplexed *.syn
```

This searches all files matching the glob `*.syn` for the word “perplexed”. If any are found they will be printed to the terminal. Specifying the `-n` option also causes the line number that the word was found on to be printed:

```
grep -n perplexed *.syn
```

This is a particularly useful feature when you start dealing with large amounts of source code, and need to find where a particular variable or routine is being used.

Finding files within directories

`grep` is very good for searching within files, but it doesn’t help when you know the name of a file but can’t remember which directory it is in. Do you remember where the `assignment1.txt` file is? You can use the `find` command here:

```
find ~/ -name assignment1.txt
```

Note that the arguments are quite different from `grep`. First, you specify a directory where the search will start. You could specify the root directory (“/”), but that would take a long time; here we start from the home directory. The `-name` option instructs `find` to show only files with the given filename.

You can use a glob with `find` to locate files with just part of the filename, but you must then surround it in quotation marks:

```
find ~/ -name "*.txt"
```

Test yourself

Use the `unix2dos` program to add carriage-returns to `assignment1.txt`. Check that you can see the `^M` symbols with `cat`. Now convert it back to Unix format and check the output.

Copy all the antonym files (those ending in `.ant`) from `/public/induction` into your `~/induction` directory.

What files contain the word “satisfied”?

What is “addlepated” a synonym for?

Where under the `/usr/include` directory is there a file named `scf.h`?
Disk usage

You have a limited amount of disk space on the file servers. You can check your current usage with the `quota` command:

```
quota -v
```

The “usage” column shows the amount of disk space you are currently using, in kilobytes. The “limit” column shows how much disk space you are permitted before limits are enforced (also in kilobytes). In this case, the user has used 5.9 megabytes of their 20 megabyte limit. The amount of space you are allocated is dependent on how many courses you are taking.

When you are approaching your limit (or are over it!) you will need to delete files so you can continue working. It can be helpful to see which files or directories are taking up the most room. You can use the `du` command for this:

```
du -ks ~/*
```

This shows the size of each file and directory (directories recursively include the size of all files and directories within them) in kilobytes in your home directory. Don’t forget to check for hidden files:

```
du -ks ~/.*
```

Note that this also lists the total size of the parent directory (“..”), which you can of course ignore. You may find that the “.mozilla” directory is using a lot of space; this is due to Firefox's browser cache. You can easily clear this from within Firefox.

Compressing, archiving and extracting files

When you are running out of disk space, an easy way to reclaim some space is to compress files you don’t use on a day-to-day basis, but don’t want to delete (such as old assignments).

The most common way for compressing files on Unix is using the programs `tar` and `gzip`. `tar` creates one file that contains many files, and `gzip`, reduces the file size of that file. Thankfully `gtar` can do this all in one step:

```
gtar -czf backup.tar.gz *.syn
```

We pass three options to `gtar`: `-c` means to create a new archive, `-z` means to compress it with `gzip`, and `-f` is used directly before the name of the file to create. Finally, `*.syn` is the list of files to backup. `.tar.gz` is the standard extension for files made this way, though you may also see `.tgz` sometimes.

We can see a list of the files in an archive with the `-t` option:

```
gtar -tzf backup.tar.gz
```
To extract the contents of the archive into the current directory, use the `-x` option:

```
gtar -xzf backup.tar.gz
```

Besides gzip, some people are starting to use bzip2 compression on their files instead; this almost always makes files smaller. Archives created this way typically have the extension `.tar.bz2`, and you can work with them by using the `-j` option instead of `-z`:

```
gtar -cjf backup.tar.bz2 *.syn
  gtar -tfj backup.tar.bz2
  gtar -xjf backup.tar.bz2
```

Note that there is also a `tar` command which is commonly used on Linux and other operating systems, however the version installed on Yallara and Numbat is different and generally not useful; you should always use `gtar` instead.

On Windows it is more common to use `zip` files. You can work with them on Unix as well with the `zip` and `unzip` commands:

```
zip backup.zip *.syn
unzip backup.zip
```

*Note that none of these commands deletes the original file(s); typically if you are trying to save space you would delete the files after creating the backup and checking that its contents are correct.*

### File permissions

Earlier we saw that certain files and directories (such as those belonging to other students) cannot be read and gave the error message "Permission denied". Unix file permissions are quite complicated but it is essential you understand the basics of them so you know how to protect and share your files appropriately.

First, let's look in more detail at the files in your home directory, by adding the `-l` (lowercase “L”) flag to `ls`:

```
ls -l ~
```

```
drwx------ 2 s1234567 students 80 Jan 24 10:59 Mail
drwx------ 2 s1234567 students 80 Jan 24 10:59 News
drwxr-xr-x 4 s1234567 students 1024 Feb 2 14:05 WINDOWS
drwxr-xr-x 5 s1234567 students 1024 Feb 10 18:05 courses
drwxr-xr-x 5 s1234567 students 1024 Feb 10 19:21 induction
-rw-r--r-- 1 s1234567 students 12 Feb 10 21:31 unix.txt
```
Instead of just listing the names of the files, we now have a detailed listing of the files.

The columns, from left to right, are:

- **drwx------**: The permission bits for this file or directory. These are explained in great detail below.
- **2**: The number of hard links to the file or directory. You can probably ignore this number for your entire career.
- **s1234567**: The owner of the file. That’s you.
- **students**: The group that the file belongs to. Groups are described below.
- **80**: The size of the file, in bytes. Note that for a directory this does not include the files within it, merely the amount of space the directory itself is taking.
- **Jan 24 10:59**: The date and time the file was last modified.
- **Mail**: The name of the file or directory.

When you access a file, you are classified into one of three categories with respect to the file:

- You are the owner of the file.
- You belong to the group that the file belongs to.
- You are someone else.

To see what groups you belong to, use the **groups** command:

```
groups
```

You may be added to more groups for certain courses, or to access a particular resource such as a CD burner. While you can belong to many groups, a file can only belong to one group. By default, all the files you create will belong to the **students** group.

Now look closely at the permission bits for the last file in the earlier list:

```
-rw-r--r--
```

Ignoring the first hyphen (it is a d for directories), you can divide the remaining characters into three sets of three characters:

```
  rw- r-- r--
```

Each of these sets corresponds to the rules to apply for a user falling into the respective category listed above. The first set is for the owner of the file, the second for a user belonging to the group that the file belongs to, and the third set is for everybody else.
Each set can have the letters r (read), w (write), or x (execute) set. If a letter is not set, a hyphen (“-”) is displayed in its place. The meaning of these letters depends on whether the file is a directory or a regular file:

<table>
<thead>
<tr>
<th></th>
<th>File</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>The contents of the file can be read</td>
<td>The contents of the directory can be listed.</td>
</tr>
<tr>
<td>w</td>
<td>The file can be written to or replaced.</td>
<td>Files can be added to and deleted from this directory.</td>
</tr>
<tr>
<td>x</td>
<td>The file is a script or program and can be executed.</td>
<td>The user can change to the directory and can access files within this directory.</td>
</tr>
</tbody>
</table>

So, for the permission bits:

```
-rw-r-- r--
```

The owner of the file can read and write it, members of its group can read it, and everyone else can also read it.

The three categories that the permission bits address are called user (for the owner), group (for members of the group) and world or other (for everyone else). So we would say the above file is world-readable and user-writeable.

You can use the `chmod` command to change the permissions of a file or directory that you own:

```
chmod g+w unix.txt
```

The g refers to the group category of users, + means to add permission, and w refers to the write permission bit. In other words, it gives write permission to members of the group. The resultant permission bits will be: `-rw-rw-r--`

```
chmod ug+x unix.txt
```

This adds the execute permission bit (x) for the owner (u) and group (g). You can remove permissions with a minus sign:

```
chmod ugo-x unix.txt
```

This removes the execute permission bit (x) for the owner (u), group (g) and others (o).

So `chmod` alters the permission bits for files and directories. The left-hand side selects which users to apply the changes to (u, g or o), the right hand side selects what permission bits to change (r, w or x), and they are separated by either a + sign, to add the permission, or a - sign, to remove the permission.

*Note that Windows does not have the same security model as Unix, so all permissions on files will be lost when you transfer them to a Windows computer. When copying files from a Windows computer to Unix, you will often find that all files have all the permission bits set for all users.*
Test yourself

How much disk space do you have left, in megabytes?

Which directory or file is taking up most of the space in your home directory? Compress that file or directory with tar/gzip, tar/bzip2 and zip. Which method gives the best result (Hint: you could use either `du` or `ls -l` to check the file sizes)?

Who owns your home directory? Does this mean you can change the permission bits on it?

Create a new text file in your home directory. What permission bits does it have? Can another student read it? Ask your friend to try and open the file. Can they? Why or why not? (Hint: you may need to look at the permission bits for the directory it is in, as well as the file itself).

Change the permission bits on the file so your friend cannot read it, if they could; or change them so they can, if they could not.

Set appropriate permissions on your `courses` directory so that other students cannot read the files within it, or even see what files are in it. Get your friend to check for you that they cannot access it.

Creating a personal homepage

You can use your account on Yallara to set up a small personal website, for showing off previous projects, keeping a weblog or just telling the world a little bit about yourself.

First of all, create a directory under your home directory named “.HTMLinfo”. Note that the capitalisation is important, as is the full stop (making it a hidden directory). Make this directory world-readable and executable (that is, set the `r` and `x` permission bits for all users `u`, `g` and `o`).

Now, make sure your home-directory is world-executable (it should not be readable, however, except for yourself).

Using `pico` or another editor, create a file in that directory called “index.html”. Make sure it’s world-readable. Write a couple of lines of text introducing yourself and save the file.

If all has gone well, you should now be able to access your home page at `http://yallara.cs.rmit.edu.au/~s1234567`

Please remember that your website is viewable by anyone, including people outside RMIT, and reread the appropriate sections of the Acceptable Usage Policy for guidelines on what you may and may not publish.
Introduction to Vim

Vim is a powerful and flexible text editor used by users of Unix around the world. It can be quite tricky to learn at first, but once mastered it is an invaluable tool when editing configuration files, programming and writing reports. There are so many features of Vim that there is probably not a single person alive who knows them all. Everybody has a set of commands that they use themselves though, and you will need to find out which commands suit you the best.

Vim is an enhanced version of an older text editor called Vi. Both Vim and Vi are textonly; they run directly in the terminal window (also making the suitable for use over an ssh connection).

We recommend using the graphical Vim, gvim, while in the labs. It is exactly the same program, except it runs in a separate window and has mouse support. You can follow this manual using either Vim, Vi, or gvim. Note that gvim is also available for Windows and Mac OS X (see http://www.vim.org), and Vim is already available on every installation of Mac OS X.

Command mode editing

Unlike other text editors you have used, Vim has two modes: command and create. You can only type text while in a create mode. While in command mode you can load and save files, do searches and replacements, import other text files, and so on.

To start Vim: 

```
vim file1.txt
```

When Vim starts you will be in command mode with a blank file. To start typing, press i (for "insert"). Type a few sentences of nonsense, then return to command mode by pressing ESC (the escape key). In summary:

Type i to enter insert mode, where you can type text.
Press ESC to leave any create mode and enter command mode.

Saving and exiting

To save the file you are working on, in command mode type :w (that’s a colon, followed by a lower-case w for “write”) and hit enter.

To quit, in command mode type :q. You can actually save and quit in one smooth move by typing :wq.

If you try quitting without first saving, Vim will show a warning that the file is not saved. You can override this with :q!, which means “quit, don’t save, I know what I’m doing.”
Help me, what’s going on?

Sometimes a simple typo can make you feel hopelessly lost in Vim as it activates a feature you have never heard of. In almost all cases, you can simply return to command mode by hitting ESC two or three times. Similarly, you can undo the last command or insertion by hitting u (for “undo”).

More creation modes

You have seen i, which starts insertion mode wherever the cursor is. There are a couple of shortcut keys for starting insertion in different places relative to the cursor:

- a begins inserting text just after the cursor.
- A (capital a) begins inserting text at the end of the line
- I (capital i) begins inserting text at the beginning of the current line.
- o “opens” a new paragraph below the current line.
- O (capital o) “opens” a new paragraph above the current line.

Try each of these now to get a feel for how they work; they can save a lot of time that in any other editor would be spent moving the cursor.

Moving the cursor

Speaking of moving the cursor, Vim has a hundred ways of letting you move the cursor around. Here are just a few (note that with the exception of the first two you need to be in command mode):

You can use the arrow keys as with any other text editor to move the cursor around.

The w and b keys move forward and back one word, respectively.

The ( and ) keys move to the previous or next sentence.

The { and } keys move to the previous or next paragraph.

The 0 (zero) and $ keys move the the start and end of the current line.

Press Control+U and Control+D to scroll up and down half a page at a time.

The Page-Up and Page-Down keys scroll up and down a whole page at a time.

You can move to a specific line-number by typing the number in and pressing G. For example, to go directly to line 48 you would type 4 8 G. This becomes very useful as you start programming as most errors are reported with the line number on which the error was found.

All of the above movement commands can be extended by specifying a number of times to repeat the command. For example, typing 5 w moves the cursor forward 5 words. Typing 3 and then pressing the down arrow moves down three lines.
Deleting text

While typing text in a create mode you can delete straight away as usual with the delete and backspace keys. In command mode you have a few more options:

Press \texttt{x} to delete the character the cursor is highlighting.

Press \texttt{d d} (you press it twice) to delete the current line.

Press \texttt{d} and one of the movement keys listed above to delete that amount. For example, typing \texttt{d w} deletes one word. Typing \texttt{d 3 )} deletes three sentences.

Make a selection with the mouse and press \texttt{d}.

Pasting text

Vim does not have separate “delete” and “cut” commands. After you delete something, it is immediately available to be pasted.

Press \texttt{p} to paste text just after the cursor.

Press \texttt{P} to paste text just before the cursor.

An easy way to fix those typos where you swap two letters around (e.g. in “teh”) is to position the cursor at the first of the pair of letters and press \texttt{x p} in succession.

An easy way to swap two lines of text is to press \texttt{d d p} in succession.

Copying text

Vim calls copying “yanking”. After you have “yanked” some text you can paste it with one of the commands above.

To yank (copy) the current line, press \texttt{y y} (press \texttt{y} twice).

As with the delete command, you can use \texttt{y} with a movement. For example, \texttt{y} and the down key copies two lines; \texttt{y 2 )} copies two paragraphs.

Make a selection with the mouse and press \texttt{y} to copy.
Replacing text

These are shortcuts to deleting text and then inserting new text.

Press r to replace a single character. For example, pressing r b replaces the character under the cursor with a “b”, and then returns to command mode.

Press R to replace a lot of text. This enters a create mode where every letter you type overwrites the existing text instead of inserting.

Press c and a movement to change some text. For example, c w deletes one word and then enters insert mode, perfect for changing just that word.

Make a selection with the mouse and press c to delete the whole selection and enter insert mode.

Indenting text

When you start programming you will find that having well-indented source code is invaluable (both for readability and getting reasonable marks on your assignments!).

To indent the current line, press >> (the right angle bracket twice).

To unindent the current line, press <<.

You can use > or < with a movement. For example <} will unindent the whole paragraph.

Make a selection with the mouse and press < or >.

Searching text

Check that you are in command mode first (press ESC).

Press / (forward-slash) and type the word or phrase to search for and press enter. For example to search for “checker” you would type /checker and hit enter.

To move to the next search result press n.

To move to the previous search result press N.

Note that searches can contain complex regular-expressions. If you don't know what a regular expression is yet, make some time to find a tutorial. You will love them!

You can search for occurrences of the word the cursor is in simply by pressing * (asterisk).


Replacing text

Usually you will want to do a text replacement over an entire file. In command mode, type:

\texttt{:%s/original/replacement/g}

and hit enter. An explanation of this command follows:

- The colon is used before most complex commands consisting of more than one or two characters.
- The percent sign ("%") signifies that the command is to act over the entire file. There are ways (not discussed here) of restricting it to just a section of the file.
- The \texttt{s} stands for "substitute" and is borrowed from \texttt{sed}'s language (see the references at the end of the Unix section of this manual).
- The \texttt{/} (forward slashes) separate the original text that you are searching for with the replacement text.
- The \texttt{g} option at the end stands for "global", and instructs Vim to make the replacement for every occurrence of the original text on each line. Without this flag, only the first occurrence on each line is be changed.
- The original text can be a regular expression, and the replacement text can contain back-references into that expression, making for a very flexible substitution scheme. See the Vim manual for examples on usage.

Getting help

Vim's online help is very comprehensive, sometimes a little too comprehensive. You can access the table of contents by typing:

\texttt{:help}

or search for a particular feature (say, the substitute command described above):

\texttt{:help :s}

Close the help window with:

\texttt{:q}
Vim configuration

You can create a Vim configuration file to store your preferences in your home directory. The file should be called .vimrc (note the full-stop at the start, making it a hidden file). Create this file now with the following contents:

```vim
syntax on
set autoindent
set ts=4 sw=4 et si
set whichwrap+=<,>,[,]
set backspace=indent,eol,start
set hlsearch
```

These options turn on some nice user-interface features, such as allowing the cursor to move anywhere, highlighting search results, replacing tabs with spaces, automatically indenting source code sensibly, and highlighting source code according to its programming language.

Congratulations!

If you have made it this far you are well on your way to becoming a Vim grand master. Remember that the commands introduced here represent less than 1% of Vim’s functionality. Browse through the tips at http://www.vim.org for ways other people are using Vim.
Where to go for more information

This manual represents just a sampler of the programs installed on Yallara and Numbat. Every single program presented has a myriad of options for customising how data is processed and formatted. Shell scripting in bash can be extremely flexible and goes far beyond the material presented here.

Any time you spend learning more about Unix will easily repay itself when you come to do assignments – especially using shell scripts to automate repetitive tasks. The following resources are good starting points for investigating the programs here that interest you.

Advanced Bash-Scripting Guide
http://www.tldp.org/LDP/abs/html/
Easily the best online tutorial and reference for doing anything and everything with Bash.

Dave Raggett’s Introduction to HTML
http://www.w3.org/MarkUp/Guide/
If you would like to continue writing your web page, you will need to learn a little HTML.

Getting Started with awk
http://doc.ddart.net/shell/awk/
Awk is a unique programming language for processing text files as collections of data records, making it easy to manipulate the data, create summaries and perform numerical computations on columns of text.

Sed – An Introduction and Tutorial
Sed is a program installed on all Unix systems. It is extremely useful for modifying text files with simple rules, for example, doing search-and-replace.

A Tutorial Introduction to GNU Emacs
http://www.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html
If you really don’t like Vim (introduced in the next section), you may find Emacs and xemacs a better choice for you. They have a similar feature-set to Vim and gvim, respectively, but a completely different interface. There are not many people who can comfortably edit a file in both Emacs and Vim!

Unix man pages
As mentioned in the first section, the manual pages (accessible by typing “man <command>” are the definitive reference for everything currently installed. Some man pages are particularly comprehensive and worth at least skimming through:

- tcsh: explains the syntax, configuration files and options.
- ls: ls has many options built-in for formatting and sorting directory listings.
- sort: sort can perform ascending and descending, alphanumeric and numeric, case sensitive or insensitive sorting on multiple columns of text.

Vim documentation
http://vimdoc.sourceforge.net/htmldoc/usr_toc.htm
http://www.vim.org/tips/index.php
Vim and gvim are extremely flexible text editors used by the majority of computer science students at RMIT. The next section of this manual will give a very brief introduction, but to really take advantage of Vim you need to find the features that work best for you. The first URL above is the complete Vim documentation; the second is a series of over 1000 tips submitted by users which are particularly useful for seeing just what Vim can do.
Unix Cheat Sheet

bc  A simple numeric calculator.
cat  Reads files and prints their contents to standard output.
cd  Change the current working directory.
chmod  Change the permission bits on a file or directory.
cp  Copy a file or directory.
cut  Filter out columns of a tabulated text file.
date  Display the current date and time.
dos2unix  Convert DOS line endings to Unix line endings in a text file.
du  Shows how much disk space a file or directory is using.
echo  Prints its arguments to standard output.
file  Identify the type of a file.
find  Search for files or folders matching a pattern.
grep  Search for text within one or more files, or within standard input.
groups  Shows what groups you belong to.
gtar  Create, extract and list tar and compressed tar archives.
head  Show just the specified number of lines from the start of a text file.
kill  Send a signal to a process; commonly used to terminate processes.
ln  Create symbolic and hard links.
ls  List the contents of a directory.
man  View a Unix manual page.
mkdir  Make a new directory.
mv  Move or rename a file or directory.
pico  A simple text editor.
ps  List currently running processes.
pwd Print the current working directory.
quota Shows how much of your allocated disk space you are using.
rm Remove (delete) a file or directory.
sed Stream editor, a program for manipulating lines in a text file.
set Set a variable in tcsh.
setenv Set an environment variable in tcsh.
sort Sort the lines in a text file.
tail Show just the specified number of lines from the end of a text file.
tar A program similar to gtar (always use gtar instead).
tcsh The default shell used by students.
top Shows and updates a list of processes that are the most active.
turnin An RMIT Computer Science program for submitting assignments.
uniq Remove duplicate adjacent lines from a text file.
unix2dos Convert Unix line endings to DOS line endings in a text file.
unzip Extract the contents of a zip archive file.
vim A powerful text editor.
w Flint Count the number of characters, words and lines in a text file.
which Determine which program will be run from the PATH.
xargs Executes a program with the standard input as its arguments
xterm A simple terminal application.
zip Create a zip archive file.

This is only a small selection of commands available in UNIX / Yallara