



Faculty of Engineering and Information Technology

# Student Computing Environment

## User Guide

2006

© The Australian National University 1997–2006

Department of Computer Science  
Faculty of Engineering and Information Technology  
The Australian National University  
CANBERRA ACT 0200

Phone (02) 6125 4043

Fax (02) 6125 0010

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Computing facilities</b>	<b>3</b>
2.1	ANU student computing environment . . . . .	3
2.2	Rules and regulations . . . . .	4
2.3	DCS student computing environment . . . . .	4
2.4	Computer system overview . . . . .	4
2.5	Password-based access . . . . .	4
2.6	Troubleshooting the computing environment . . . . .	5
2.6.1	Access to computer laboratories . . . . .	5
2.6.2	Access to your computer account . . . . .	5
2.6.3	Hardware problems . . . . .	6
2.6.4	Software problems . . . . .	6
2.6.5	Quota problems . . . . .	6
2.6.6	Out-of-hours problems . . . . .	6
<b>3</b>	<b>K Desktop Environment</b>	<b>7</b>
3.1	Login and logout . . . . .	7
3.2	Workspaces . . . . .	8
3.3	Help system . . . . .	8
3.4	Front panel and workspace menu . . . . .	9
3.5	KDE Problems . . . . .	9
<b>4</b>	<b>File manager</b>	<b>13</b>
4.1	File system concepts . . . . .	13
4.2	The KONQUEROR file manager . . . . .	13
4.3	Quotas and Trash . . . . .	14
4.4	Elements of File System Structure . . . . .	15
4.5	Configuration files . . . . .	15
4.6	Browsing and ethics . . . . .	16
<b>5</b>	<b>Mozilla browser</b>	<b>17</b>
5.1	World-Wide Web . . . . .	17
5.2	URLs . . . . .	17
5.3	Driving Mozilla Navigator . . . . .	19
5.4	Finding things on the Internet . . . . .	20

5.5	Downloading things from the Internet . . . . .	20
5.6	Resetting Mozilla Configuration . . . . .	20
<b>6</b>	<b>News and Mail</b>	<b>21</b>
6.1	Introduction . . . . .	21
6.2	Phorum . . . . .	21
6.3	Mozilla Mail & Newsgroups window . . . . .	22
6.4	Reading e-mail . . . . .	24
6.5	Sending e-mail . . . . .	24
<b>7</b>	<b>Editing text files</b>	<b>27</b>
7.1	Introduction . . . . .	27
7.2	Editors and their Pros and Cons . . . . .	27
7.3	The load-edit-save cycle . . . . .	28
7.4	KWrite . . . . .	28
7.5	Emacs . . . . .	28
7.5.1	Starting Emacs . . . . .	29
7.5.2	Emacs modes . . . . .	30
7.5.3	Invoking and aborting Emacs commands . . . . .	31
7.5.4	How to learn more about Emacs . . . . .	31
<b>8</b>	<b>Printing</b>	<b>33</b>
8.1	Introduction . . . . .	33
8.2	Submitting a print job . . . . .	33
8.2.1	Choosing another printer . . . . .	33
8.2.2	Other print dialogue box options . . . . .	35
8.3	Checking the status of the printers . . . . .	35
8.3.1	Cancelling a print job . . . . .	36
8.4	On-line help and further information . . . . .	36
8.5	Printing from within Applications . . . . .	36
8.6	Sensible use of the printing resources . . . . .	37
<b>9</b>	<b>Submitting Files and the Unix Command Line</b>	<b>39</b>
9.1	Submitting files for assessment . . . . .	39
9.2	Finding out your marks . . . . .	40
9.3	Using the Command Line (Shell) in a konsole Window . . . . .	40

# Chapter 1

## Introduction

This is an introductory manual for the LINUX-based student computing environment provided in the Faculty of Engineering and Information Technology. The computing environment is primarily used to support units taught by the Department of Computer Science and the Department of Engineering. Access to and use of the computing environment is subject to University rules and regulations. These are provided in the Department's *Student Handbook*.

The computing environment itself provides a great deal of information and help documentation, and there is no point in duplicating that material here. Rather, the point of this document is to get you started to the point where you can read and productively use the on-line documentation. It does this by providing a discussion of the general aspects of the computing environment, in order to give you a structured way of thinking about the system and its main components. Once you understand this structure, you can make sense out of the details provided in the on-line documentation.



# Chapter 2

## Computing facilities

### 2.1 ANU student computing environment

The Information Infrastructure Services program within the Division of Information (DOI) has the responsibility of providing and maintaining the *Information Commons* which includes the undergraduate computing environment at the ANU.

In particular they provide

- Unix-based laboratories, using Linux-based PCs,
- Macintosh-based laboratories, and
- Windows-based PC laboratories.

Details of the facilities provided, and the regulations that apply to their use, are available via tabs and links on the *infocommons* web page:

<http://infocommons.anu.edu.au/>

The page includes links to the online DOI helpdesk, InfoPlace information and the Student IT Guide. The Student IT Guide provides access to extensive information on all campus information services for students.

Laboratories are heavily used during ‘normal teaching hours’ for supervised laboratory sessions in particular units. The relevant bookings are posted. Outside these times the laboratories are available for general use associated with undergraduate studies. Access to laboratories is controlled by your student card, and you may have access to some facilities and not others, based on your enrolment. Most laboratories have extended availability, and some laboratories may be available on a 24-hour basis (at least for part of the academic year).

The residential halls and colleges at the ANU provide extensive facilities to access the undergraduate computing environment. Details are provided by each residence. If you have your own personal computer you may thus be able to do some of your work from your desk (either at home or at your hall or college) over the Internet.

## 2.2 Rules and regulations

Use of the computing environment is subject to formal ANU Rules. Copies of the rules are posted in each of the infocommons laboratories. You should read these rules carefully. Indeed, you will be explicitly asked to acknowledge that you have read them before you are given access to the undergraduate computing environment.

The rules go to a lot of trouble to identify what you may and may not do within this environment, and to identify the penalties for unacceptable use. Put informally, the power of the computing environment allows many opportunities for thoroughly unacceptable behaviour. While this is generally only perpetrated by a small number of students, it can degrade the amenity of the environment for all users. Dealing with such breaches of behaviour can often take a considerable amount of our scarce time. Offenders often find us very short of understanding and kindness.

## 2.3 DCS student computing environment

There are six infocommons laboratories in the CSIT building, in rooms N111, N112, N113, N114, N115 and N116. Although known informally as ‘DCS laboratories’ they are in fact part of the integrated ANU information commons computing environment and are subject to the relevant ANU Rules. A copy of these rules are provided in the Department of Computer Science *Student Handbook*.

Much of the day-to-day management and local IT support of the computer systems in these laboratories is carried out by the DCS Technical Support Group, located on the second floor of the CSIT building. TSG maintain their own web page at:

<http://tsg.anu.edu.au/>

## 2.4 Computer system overview

Each of the rooms N111, N112, N113, N114, N115, and N116 each contain 20 Linux workstations and a PostScript laser printer. The workstations are Intel based PCs running the Knoppix distribution of the Linux Operating System. The workstations allow the work of editing, graphic display and computation to be performed locally on the workstation. Authentication of users (see section 2.5) and the maintenance of the users files are managed by a set of central UNIX-based servers. A brief description of the server systems is provided on the TSG web pages.

## 2.5 Password-based access

Many of the facilities of the student computing environment require you to identify yourself. An obvious case in point is access to your electronic mail. To this end you are given a ‘user-name’ and an associated password.

Your user-name has to be regarded, in the end, as publicly known. It may used by the computer system to label printouts that you collect from a printer in a computing laboratory.

In some units you will submit assignments electronically and your assignment will be labelled with your user-name along with your full name.

In contrast, your password must be regarded as secret and you must keep it secret. Just as you are required by credit-card agreements not to divulge your PIN to anyone else, equally you are required by ANU Rules not to divulge your password to anyone else.

To use a workstation in one of the laboratories you need to login to that workstation. An available workstation will display a login screen in which you may enter you user-name and password, see Figure 2.1. You user-name will be readable, however, each character in your password will be displayed as an asterisk to prevent others from viewing your password.



Figure 2.1: Login Screen

## 2.6 Troubleshooting the computing environment

### 2.6.1 Access to computer laboratories

If your security card will not let you into a laboratory (allow three working days after being issued for it to be activated), go to the Security Office in building 45, the John Yencken building (on Sullivans Creek Road across the creek from Fellows Oval).

### 2.6.2 Access to your computer account

If you have your user-name and your password, and you cannot successfully log into your LINUX account on the CSIT systems then you should see the consultant programmer in the Department of Computer Science, in room N202.

If you have forgotten your password, go to InfoPlace for help. InfoPlace is located on level 3 of the Chifley library building. For more information about the services offered by InfoPlace, see

<http://ilp.anu.edu.au/infoplace>

To avoid the inconvenience of needing a new password, the message is

**DON'T FORGET YOUR PASSWORD!**

### 2.6.3 Hardware problems

Problems with the computing equipment which are clearly of a hardware nature, such as the mouse not functioning or the printer not printing, should be reported to Division of Information Help Desk. There are notices in each laboratory giving the contact phone number (currently 59666). Alternatively, report the problem (using a working terminal) via e-mail to `consultant@cs.anu.edu.au`. Please accurately gather and report as much information as possible.

### 2.6.4 Software problems

Difficulties with particular programs, applications, and so on, which are not obviously hardware problems should be reported to the Department of Computer Science. Reports may be made via e-mail to `consultant@cs.anu.edu.au`. Please accurately gather and report as much information as possible.

Difficulties with the subject matter of your course or with removing errors from your programs should be referred to your tutor or lecturer or by e-mail to the help service for your course, for example `comp1100@cs.anu.edu.au`, `comp1200@cs.anu.edu.au`, `comp2100@cs.anu.edu.au`, etc.

### 2.6.5 Quota problems

Disk quotas are intended to be adequate to complete all assignments. If you exceed your quota, you must remove excess files. Core files (files named `core`) are produced when a program crashes and are often very large. They are prime candidates for removal. See section 4.3 for further information.

If you are unable to remove files, e-mail the consultant programmer (`consultant@cs.anu.edu.au`).

### 2.6.6 Out-of-hours problems

See the notices in the laboratories.

# Chapter 3

## K Desktop Environment

The K Desktop Environment (hereafter referred to as KDE) is an example of a modern Graphical User Interface (GUI). By modern we mean that it is based on mouse and keyboard interaction with visual objects on a colour screen. A GUI provides a user environment that enables the user to perform multiple activities with a consistent look and feel across the utilities, tools and applications.

For IBM PC-compatible systems, you may well be familiar with the Windows 95/98/NT/ME/XP user environments. For Apple Macintosh systems, you may well be familiar with the MacOS/MacOS-X user environment. The GUI available for UNIX and LINUX are built on top of X-Window graphic system (X-Window provides the low level protocols and standards for graphic system interaction). Initially each major UNIX vendor produced their own proprietary user environment and in 1993 a consortium was established to produce a standard GUI for UNIX-based systems, called CDE. Similarly the groups within the Linux community initiated projects to develop GUIs for Linux, of which the two most significant and well developed are KDE and later GNOME. Only KDE is supported in the DCS laboratories.

### 3.1 Login and logout

KDE manages all aspects of a computing session. Initially you are presented with a ‘login window’, and are prompted to type your user-name and password (see section 2.5). Make sure to press the **Enter** key after your user-name and after your password. After a little time KDE will initiate a session. The screen will have a *front panel* along the bottom, as shown in Figure 3.1. Clicking the **Logout** button (see the right hand end of the front panel) will cause KDE to exit

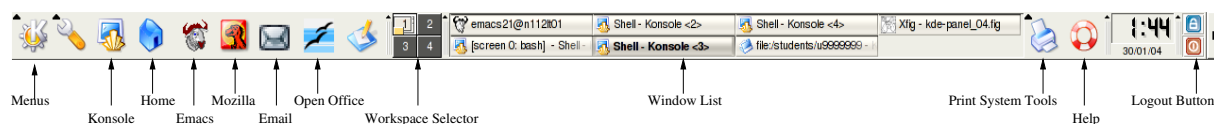


Figure 3.1: Front panel

the session. The next time you login, your layout of windows and icons will be restored<sup>1</sup>.

<sup>1</sup>Applications or tools that are not KDE compliant may not have their position and size restored correctly.

## 3.2 Workspaces

In a graphics-based user environment the screen can get very cluttered. KDE provides the notion of *workspaces*. The normal setup provides four. You can associate each workspace with a different set of tasks.

You can select which workspace you want displayed on the screen, and you can switch from one workspace to another by clicking on the particular workspace within the Workspace Selector applet on the front panel (see Figure 3.1). This represents the fact that in some circumstances computing resources are cheaper than your time. Rather than continually rearrange the items displayed on a single screen, you simply arrange the contents of workspaces and then let KDE keep track of the bookkeeping.

In a single workspace you will typically have a number of overlapping windows. One window will be ‘on top of the stack’ and will be *in focus*. All keystrokes from the keyboard will be directed to this window. Mouse clicks will be directed to the window that the cursor currently sits over.

If you finish up with too many windows in a workspace, you can close (that is, delete) ones that you are not using, and start them up again when you need them. An alternative to this strategy is to reduce a window to a small *icon*. In this state they don’t take up much real estate (but they still take up computing resources on the server). They can be restored to the window state very simply and quickly.

## 3.3 Help system

Rather than telling you any more about KDE behaviour we can now tell you about the *help system* that is provided in KDE.

At the right of the front panel (left of the clock) is an icon depicting a life saver ring. Clicking on this icon will open the window shown in Figure 3.2. You are now looking at an on-line user manual for KDE, and we need not continue in full detail in this manual. Rather, this manual can leave you in the competent hands of KDE itself to explain what is going on.

The following general ideas are of great advantage before you wander off into the details. The help information is collected into a number of documents. The left tab-panel has two tabs, *Contents* and *Glossary*. The Contents tab describes a tree structure organization of the online help documents that enable quick navigation through the documents. Each document can be considered to have a number of sub-documents, where each sub-document is considered to be a document. Of immediate interest are the *KDE user’s manual* and the *Tutorials*. The Glossary tab categorizes the help information alphabetically or by topic. Neither the path depicted in the Contents tab nor the Glossary tab reflect the path through the documents that you have taken, but rather provide a way to jump to specific points in the collection of help documents (see example in Figure 3.3).

In addition there is the notion of the path that you have traversed through the documents. This is recorded for you. You can go back one step (and another, and another) simply by clicking on the **Backtrack** button (the left arrow above the navigation tab panel in Figure 3.3). Alternatively you can look at a list of all the paths you have visited and jump back to any earlier document, by pressing and holding the left mouse button on the Backtrack button.



Figure 3.2: Help viewer

## 3.4 Front panel and workspace menu

The KDE philosophy is that the front panel provides the standard access path to the resources of the computing environment. You will find buttons on the front panel for KDE Menu, File Manager (KONQUEROR), Emacs, Web Browser (MOZILLA), Open Office, Printer Manager and KDE Help. Resting the mouse pointer above a button for a couple of seconds will cause a small tooltip description of that button's purpose to pop up. Pressing mouse button 1 (the left button) while the pointer is over an on screen button which has an upward pointing arrow will display that button's menu. Either releasing mouse button 1, pressing escape or pressing mouse button 1 on some other part of the screen will hide the menu again. The A Quick Start Guide to the Desktop document in the Welcome to KDE help page should be consulted for details on using the front panel.

A range of actions are also available via a **Workspace Menu**, which is obtained by clicking mouse button 3 (the right button) when the cursor is over the screen backdrop (and not over any window or icon).

## 3.5 KDE Problems

When your Linux based machine does not function properly (which is very rare), that is, it does not respond to mouse and keyboard inputs, then the server system has probably malfunctioned (crashed). You may use the **Ctrl + Alt + Backspace** keys to kill the graphic display server, after a few seconds the display should restart (note that if you have any unsaved changes to your documents those changes may be lost) after which you can repeat the login procedure. If

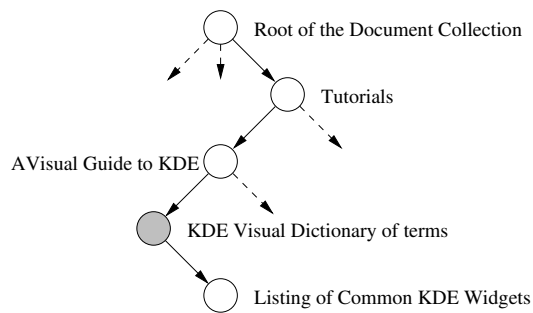
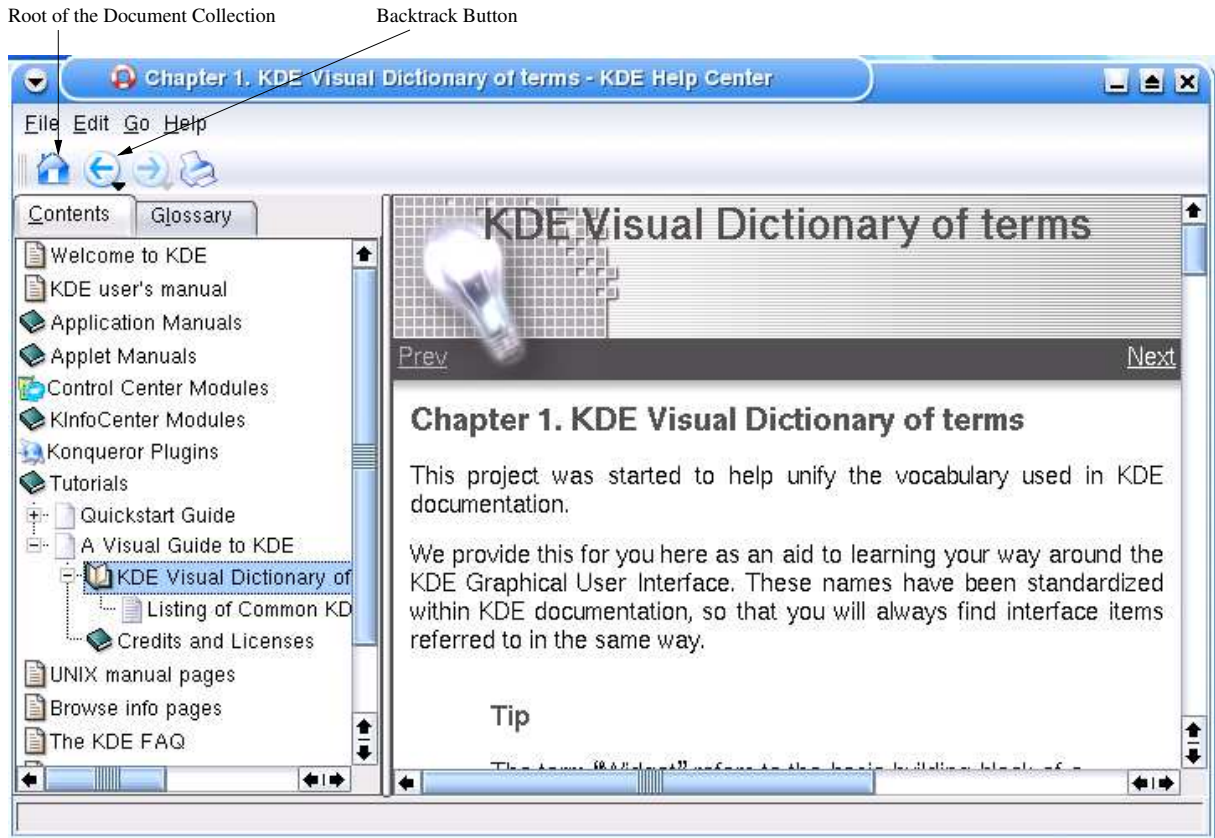


Figure 3.3: Document tree

it still doesn't work, you need to get some help from your tutor, or the consultant programmer in the department.



# Chapter 4

## File manager

### 4.1 File system concepts

The LINUX servers in the DCS student computing environment have a file system that you, as a user, need to understand and interact with.

One fundamental property of the file system is that it is *stable*. If you create a file and write data to it during one computing session, then it will still be there (with the same contents) when you next log in.

Another fundamental property of the file system is that it is hierarchical, or tree-structured. Basic data files may be organised by storing them in a file called a *folder* or *directory*. This folder can itself be organised by storing it in a higher-level folder, and so on. Associated with your computer account is a *home directory*. You may create data files and folders here, within certain limits (see section 4.3).

Finally the file system has an associated notion of protections. You may both read and change files in your home directory and its hierarchy of folders. You may read the files in large sections of the file system, but you may not change them. Finally there are some areas of the file system that you may not even read. For example, you cannot read files in the home directory of another student.

You will spend a considerable amount of your time as a computer user interacting with the file system. KDE provides a file manager tool called Konqueror for this purpose. The associated help system provides information both about file system concepts, and about using the tool itself.

### 4.2 The KONQUEROR file manager

You can start the file manager by clicking *once* on the icon in the front panel which looks like a house (home).

There are two ways of accessing the associated help system.

1. Start the Help viewer (see section 3.3) and select in turn (via the Contents tab pane) Application Manuals, System, More Applications and lastly File Manager - Super User Mode (this is not particularly well labelled).

2. At the far right end of the file manager menu bar there is a menu labelled Help. Select the Konqueror Handbook option from this menu.

The Konqueror file manager tool allows you to perform the following range of tasks.

- Basic File Management Skills:
  - Select a Single File or Folder.
  - Select Multiple Files or Folders.
  - Drag and Drop a File or Folder.
- Working with Files and Folders:
  - Create a New File or Folder.
  - Rename a File or Folder.
  - Open a File or Folder.
  - Move a File or Folder.
  - Copy a File or Folder.
  - Execute an Action for a File or Folder.
- Navigating to other Folders:
  - View a Folder.
  - Go to Your Home Folder.
  - Change to Parent Folder.
- Deleting Files and Folders:
  - Delete a File or Folder (to the Trash Can).
  - View the Desktop folder and the Trash Can on the desktop.
  - Move a File or Folder from the Trash Can Back into File Manager.
  - Shred a File or Folder Permanently.
- Changing Permissions:
  - Change the Permissions on a File or Folder.
- Using Different File Manager Views:
  - Show Hidden Objects.

## 4.3 Quotas and Trash

Depending on which units you are registered in, you will be allocated an appropriate *disk quota*. A disk quota limits the amount of disk space you are allowed to use. If you experience difficulties when trying to save or copy a file, it may be that you have exceeded your disk quota. Certain programs will inform you when you try to exceed your quota.

You can check your quota by entering the command `quota -v` on a command line (see section 9.3). You will probably not exceed your *files* quota, which limits the number of files



Figure 4.1: Trash Can

you can have at anytime. Sometimes you will exceed your *usage* quota, which limits the number of disk blocks you can use. If you have used an amount of space which is between your *quota* and your *limit* then you will have a certain *timeleft* to remedy the situation before drastic action occurs. Drastic action usually means disabling your account.

To reduce the amount of disk space you are using you need to remove files. This means putting them into the trash and emptying the trash can (permanently removing them). If you put files into the trash and forget to empty the trash can then they will still consume some of your disk quota. You can empty the trash can by pressing mouse button 3 (right mouse button) on the trash can icon on your desktop and selecting the Empty Trash menu item (see Figure 4.1).

Which files should you remove? Files which are unrelated to your work are an obvious choice! Compiled programs from previous laboratory sessions and assignments are another candidate for deletion. As long as you keep source files you will be able to compile them again to produce the compiled versions.

## 4.4 Elements of File System Structure

Some important system directories, and their allocated use, are given in the following table.

```
/dept/dcs/comp1100/public
```

```
/dept/dcs/comp1200/public
```

```
/dept/dcs/comp2100/public
```

...

All teaching support material for units taught by the Department of Computer Science can be found from the above directories.

## 4.5 Configuration files

You may exercise your preferences about many details of the computing environment. For example, you may want to change the colours used in your KDE display, or you may want to change the background image. These, and *many* other details are stored in configuration files. Most of these are stored in your home directory and are given a name starting with the period character (‘.’) and are normally classified as hidden files. Some of these files are

```
.Xdefaults
.cshrc
.history
.maildelivery
```

As a first approximation you should *not* directly modify these files yourself. You shouldn't do it as a second approximation, either. There are tools for under the Configuration sub-menu for altering the desktop configuration which should be used instead, although you should exercise care.

## 4.6 Browsing and ethics

One important thing about the LINUX file system is that it handles access control on a per-user basis. This means that, in theory, you can browse the file system without being able to read, modify or run anything which you're not supposed to. Browsing the file system can be a valuable learning experience! You might find things which you didn't know existed. Some of these things might even be useful!

While you are encouraged to browse the file system, you should still be careful. Consider these examples:

- If you discover, for whatever reason, that you can see another student's files, and you think that those files should be private, then you should probably inform them of the situation (probably using e-mail). Before you decide that it will save you a lot of work if you can find files containing other student's assignments, you should read the section in the *Student Handbook* about 'Misconduct in Examinations'.
- If you find information in the file system which you believe compromises the security of the system, then, rather than taking advantage of the information and becoming a (very) short-term local hero by cutting short your academic career, you should send e-mail to `consultant@cs.anu.edu.au` detailing your finding. If you have any doubts about this suggestion please read the 'Use of university computing resources' chapter in the *Student Handbook*.

# Chapter 5

## Mozilla browser

### 5.1 World-Wide Web

The World-Wide Web, or more simply, the ‘web’, is an elaborate collection of servers linked via a world-wide network called the Internet. Most of these servers are quite happy to interact for free with a user who is somewhere else on the Internet. The Internet runs over commercial communication lines run by companies such as Telstra and Optus, and the bills have to be paid for this use (but that is another story).

The DCS computing environment provides *Mozilla* as the browser program which provides access to the resources of the web. The major components of the Mozilla window are shown in Figure 5.1.

### 5.2 URLs

Everything publicly accessible on the web has a *Uniform Resource Locator*, or URL for short. The URL is an address that uniquely identifies an object across the whole web. Most businesses now have a web page, and the corresponding URL is quoted in general advertising. An example of a URL is

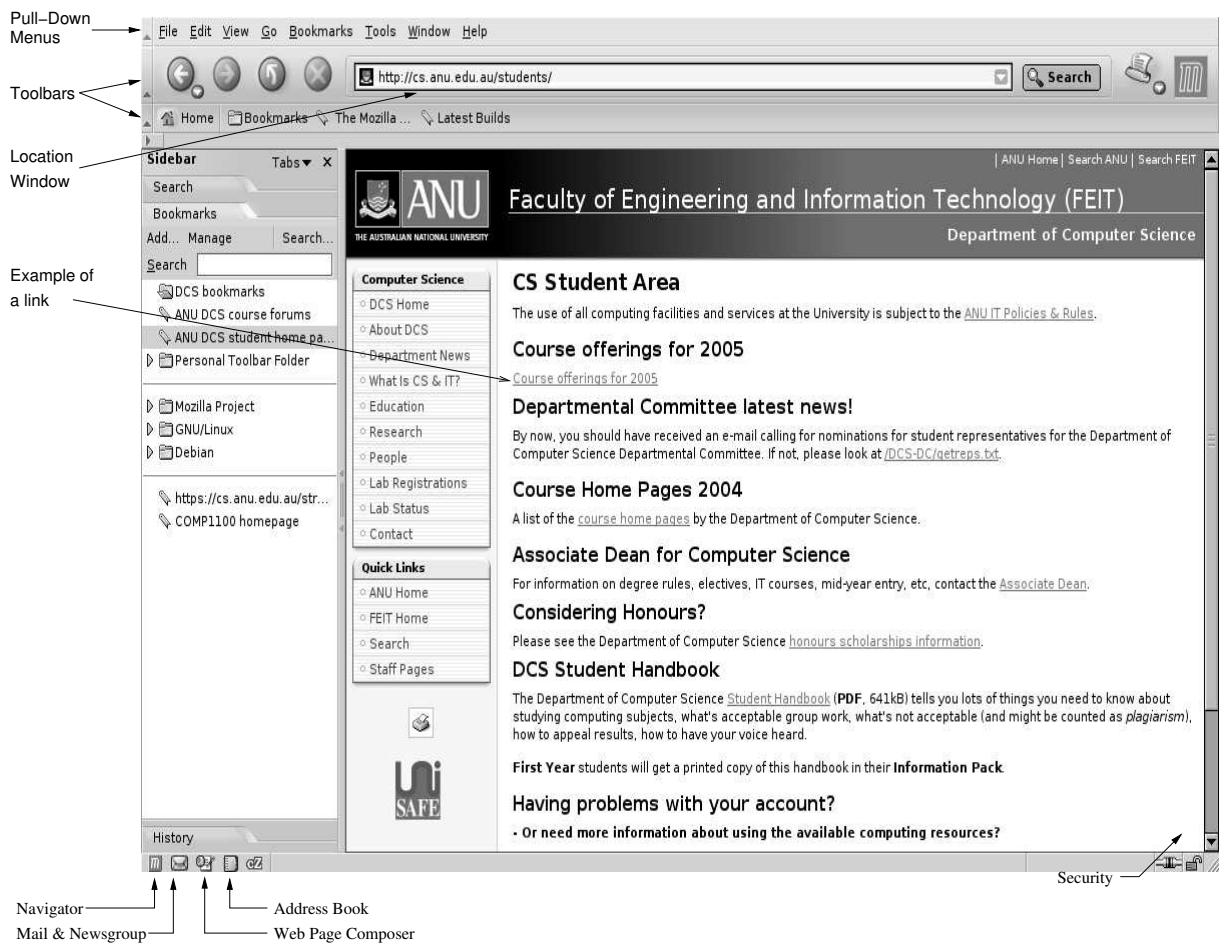
```
http://theage.com.au/technology/itnews/index.html
```

which refers to the IT News page of the Technology section of the Internet edition of *The Age*.

It is not really important that you understand the structure of a URL (that is what we have computers for), but in brief, `http` is the delivery protocol (the language in which Mozilla must speak to the computer holding the data), `theage.com.au` is the name of the computer which holds the data, and `technology/itnews/index.html` is the name of some data (probably a file) residing on that computer.

Often you can abbreviate URLs, especially for data on local computers. For example, `http://cs/` is an abbreviation for `http://cs.anu.edu.au/index.html`.

Figure 5.1: The Mozilla Window



## 5.3 Driving Mozilla Navigator

The display region occupying most of the window is where the data is shown. In Figure 5.1 Navigator is displaying the Students page from server `cs.anu.edu.au`. This should be the first page you see each time you start Mozilla and is called the *home page*.

Some of the items you see in the display region are *links* which point to different pages. Usually (but not always) you can identify links because they are underlined and/or have a different colour. If you point the mouse cursor at them, the arrow changes into a small hand. Clicking with the left mouse button will retrieve the web page to which the link points, or intuitively, take you to page that the link points at.

The words File etc. along the top of the Mozilla window hide menus that drop down when you click on them with the left mouse button. Menu items marked with a small right triangle (▷) lead to further cascading menus when the mouse is moved over them. Many of the menu items have a keystroke equivalent which is indicated on the right of the menu. Below the menu bar are a set of buttons called the *Navigation Tool bar*. You can find out what most of these do by experiment, so we will just mention a few important ones.

- File** The important items in this menu that are not more easily accessed via the Tool Bar are **New**, which leads to a sub menu from which you can open an additional Mozilla window or create a new Email message, **Save As**, which saves the current page to disk, and **Quit**, which closes Mozilla, etc.
- Go** This menu holds the history of recent pages you have visited without backing up. It provides an easy way to go back to a previous page. There is a small down arrow next to the **Back** button which also holds the backward history. To go back to the most recent page, clicking the **Back** button is easier.
- Bookmarks** Mozilla lets you keep a list of URLs that you might wish to visit again. This is your *bookmark list*. To add a URL to the list, go to that place then select **Bookmark This Page** from the **Bookmarks** menu.
- Back** This button takes you back to the place you were last.
- Forward** This button undoes the action of **Back**.
- Reload** Sometimes Mozilla will show you an old version of a page even after it has been changed. Clicking this button asks Mozilla to ensure that you are seeing the most recent version.
- Stop** If a page is taking too long to fetch, or you change your mind about wanting to see it, **Stop** will abort your request.
- Print** This button opens a control panel that lets you print the page you are viewing.
- Search** This button calls up the Google search engine web page for searching on the Internet.
- Home** This takes you to your home URL (the one that first displays when you start Mozilla).

## 5.4 Finding things on the Internet

The most common way to search the Internet for something is to use a *search engine*. The Search button defaults to our recommended search engine Google, which is located at

`http://www.google.com.au/`

To use the search engine, type in the subject you are interested in the box on the web page and click on the Google Search button. You will be returned a list of links that Google knows about that refer to your subject. To compose efficient queries you should click on Advanced Search and read the help pages.

## 5.5 Downloading things from the Internet

There are several ways you can take local copies of things you find on the Internet.

- To copy the whole page you are viewing, use **Save Page As...** from either the File menu or the pop up menu obtained by clicking the right mouse button anywhere on the page.
- To copy the page that a link refers to, point your cursor at the link then click the right mouse button. A little menu will pop up; select **Save Link Target As...**
- To take a copy of an image (preferably not a rude one), point your cursor at the image then click the right mouse button and select **Save Image As...**

## 5.6 Resetting Mozilla Configuration

There are many features and settings in Mozilla that may be changed to suit individual users. The default configuration for each student user includes the correct settings to access the Internet through the proxy server and correct ANU Email and Newshost accounts setup. You may reset your Mozilla configuration to the initial default settings by using the command:

```
reset-my-mozilla-prefs
```

on a command line in an xterm or terminal. Resetting the preferences will not affect your bookmarks or the mail that is stored in your Mozilla mail folders.

# Chapter 6

## News and Mail

### 6.1 Introduction

The use of the Internet to support the web is a relatively recent phenomenon. The Internet has had a much longer existence, supporting services such as electronic mail and electronic news.

The basic use of electronic mail, known more simply as e-mail, is to send a text message from user A to user B. User B may have an account on the same server system, or may have an account on an appropriate server on the other side of the world. All user A needs to know is the e-mail address of user B. Electronic news can be thought of as a variant of e-mail. User A sends a text message to an electronic bulletin board. It is posted to the bulletin board, and it may eventually expire by being removed from the board. User B may read the posting on the bulletin board at some later stage. DCS uses a web based variant of the bulletin board concept called *Phorum* which is used for course announcements and course based discussions (this has superseded the use of Internet News). Links to specific course forums or course newsgroups are included on DCS course home pages, and you will be expected to read the messages on a regular basis. You will also be expected to use e-mail to contact your tutor and your lecturer.

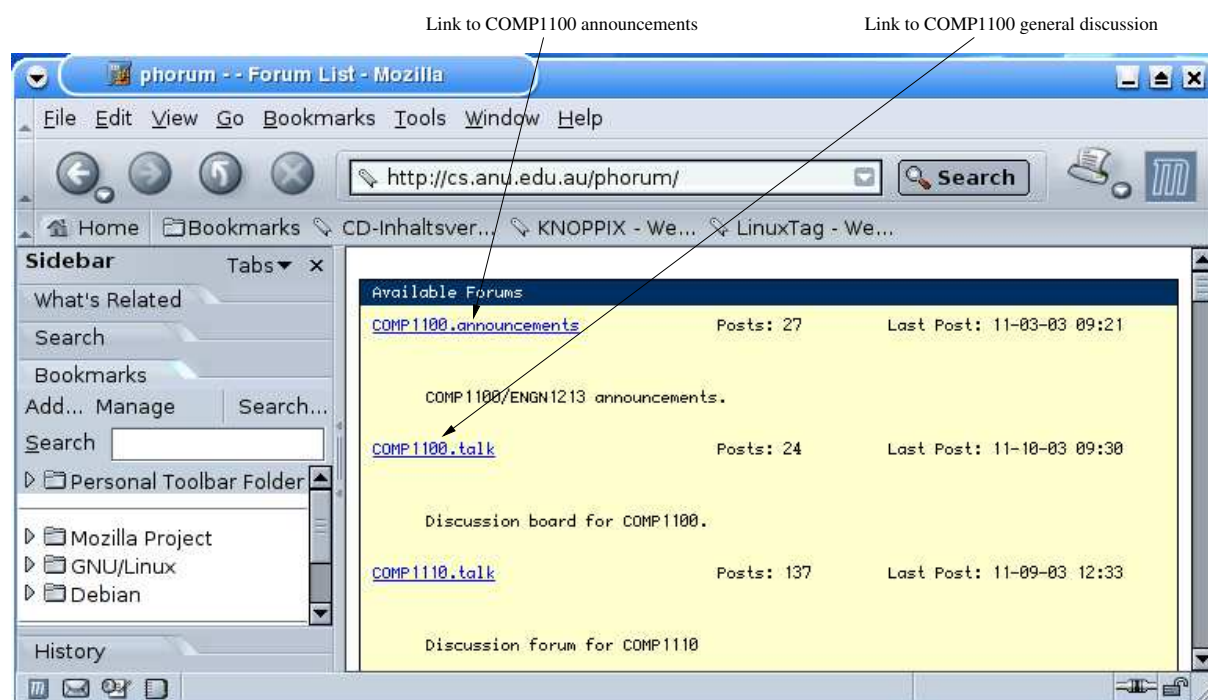
The Mozilla web browser is used to read and post to the *Phorum* bulletin boards and Mozilla has been extended to support Internet news and e-mail services, and so is the preferred utility for all these tasks.

### 6.2 Phorum

DCS announcements and discussion groups are maintained using a web based bulletin board. Each course would normally have a general discussion board (e.g. `COMP1100.talk`) and/or an announcement board (e.g. `COMP1100.announcements`). You can access the discussion groups for your subjects using Mozilla through the DCS Phorum site with the URL <http://cs.anu.edu.au/phorum/> (see Figure 6.1). Course announcement groups provide a central place for lecturers and tutors to provide important information about your course and should be read frequently. The discussion groups are an open forum for students to post topics of interest or concern and to reply to those posts. Note: to access the discussion groups you will be required to enter your user-name (student id) and password.

Each discussion group lists a number of topics. Under each topic may be one or more postings (see the example in Figure 6.2). You will normally be able to start a topic or reply

Figure 6.1: Phorum discussion groups.



to an existing topic through the particular discussion group. This is a useful open discussion system for course related issues and should be used for the purpose of helpful course based information (note these are moderated).

### 6.3 Mozilla Mail & Newsgroups window

Clicking the Mail & Newsgroups button (second from left) on the bottom line of the Mozilla navigator window will cause a new mail and newsgroups window to open. Alternatively you may use the **Window** menu. An example of what the window might look like is shown in Figure 6.3.

The default size for the Mozilla mail and newsgroups window may be smaller than desired and somewhat cluttered. You should start by dragging a corner to make it as large as you can. Mozilla will remember its window sizes from the last time it was used.

The left hand window pane shows the organisation of your e-mail in folders (messages sent to you turn up in the **Inbox**) and of the news<sup>1</sup> server (it shows the newsgroups that you have subscribed to). One of the folders will be *selected*, generally the **Inbox**. Any folders which have unread messages in them will be shown in bold with the number of unread messages in brackets after them, as does **Inbox (1)** in figure 6.3.

The right hand top window pane shows the messages that have been stored in the *selected* folder, which is the **Drafts** folder in figure 6.3. A message may be selected in the folder by

<sup>1</sup>The news system is no longer used or supported by DCS for course announcements and discussion and its use is left as an exercise for those who are interested.

Figure 6.2: Phorum COMP1100 general discussion group.

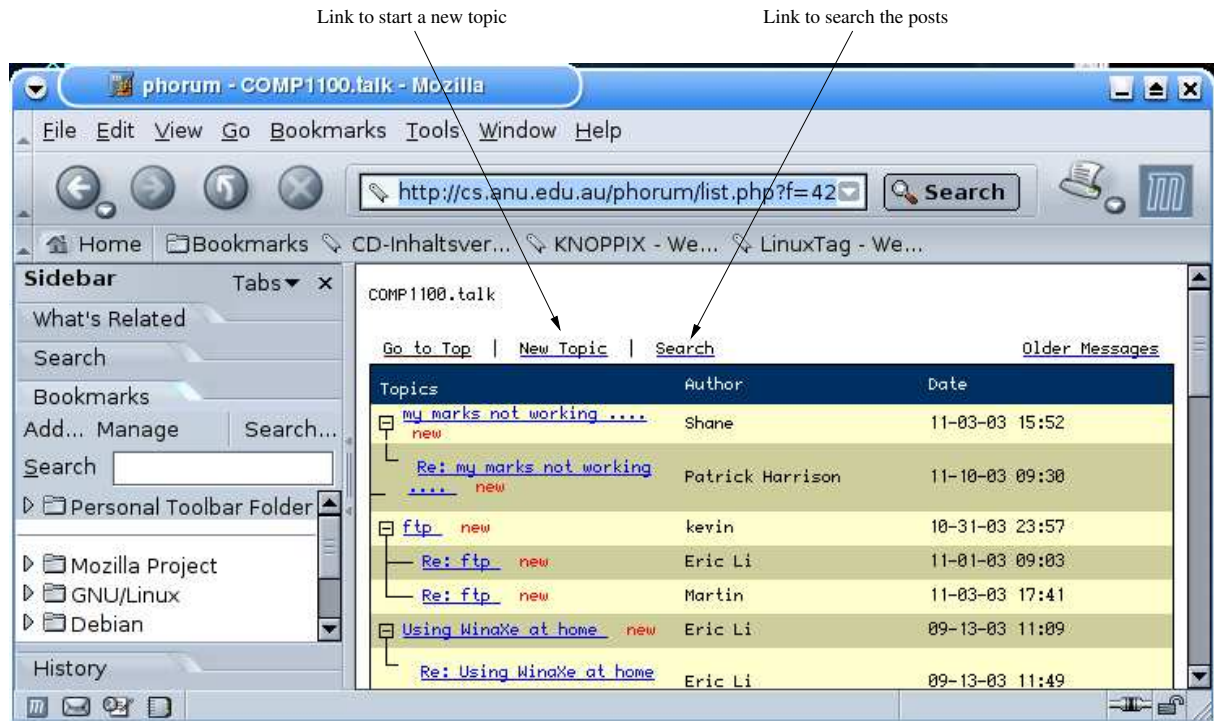
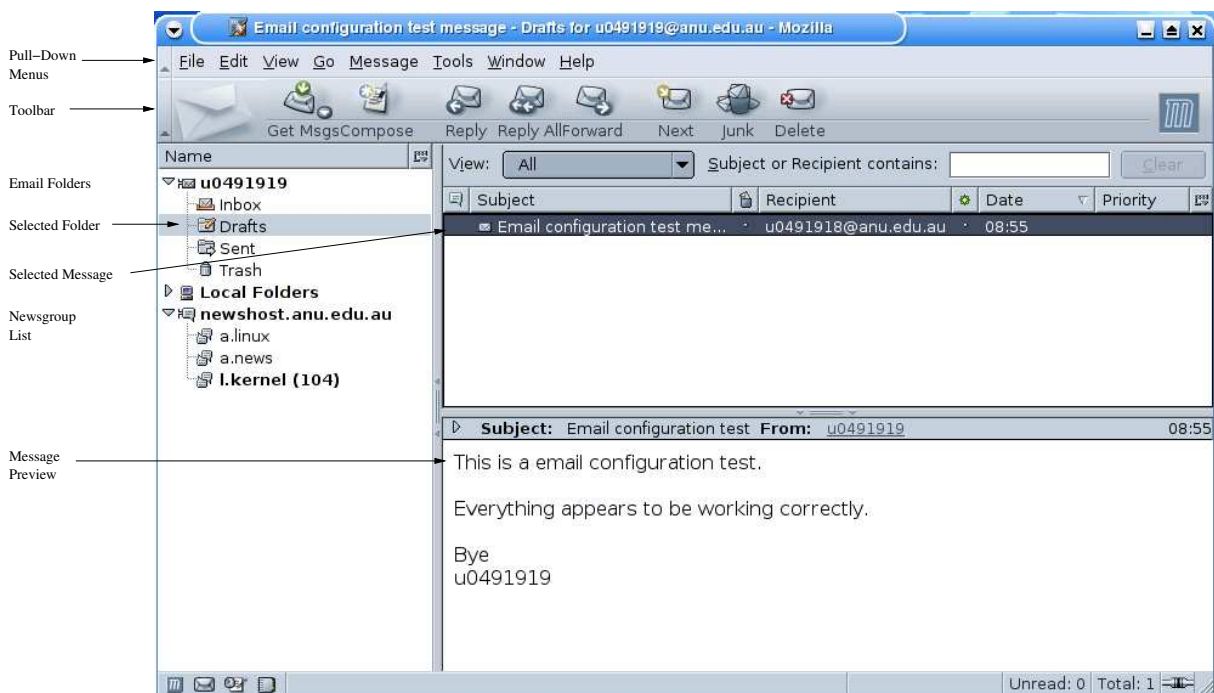


Figure 6.3: Mozilla Mail & Newsgroups Window



clicking on it with the left mouse button.

The message list may be sorted by clicking on the column header by which you wish to sort: **Subject**, **Date** etc. The list may also be sorted by messages on the same topic, called *threads* by using the menu sequence **View->Sort->Threaded**. They are then collected together under the first message in the thread and there will be a small plus sign + which you use to open up the thread to view the individual message titles. Sorting by an individual column heading opens up all threads.

The right hand bottom window pane previews the contents of the selected message and will show a scroll bar on the right hand side if the message is longer than the size of the window pane. The basic function of choosing a message to view is the same for e-mail and newsgroups.

## 6.4 Reading e-mail

What makes it important that you are comfortable with Mozilla Mail, is that it will sometimes be *necessary* for you to send and receive e-mails. Typical uses of the mail system follow, but you will also find it quite useful to send messages:

- To your tutor, to tender an excuse for missing your tutorial or not doing an assignment. (Some common ones are that you were temporarily kidnapped by aliens or that your dog ate your assignment.)
- To your lecturer, to tell them that the textbook is priceless/worthless.
- To the consultant programmer (`consultant@cs.anu.edu.au`), to advise that there is smoke coming out of the printer.

Not only will you miss out on gossip from friends if you ignore mail, but you will miss out on crucial messages from lecturers and tutors. The department deems that you have been adequately notified when you have been sent e-mail on any matter of mutual concern.

You can view your incoming mail by clicking on the **Inbox** folder (as described in section 6.3) and clicking the **Get Msgs** button to ensure you pick up all the latest mail. Selecting a message with a single click will display the message in the preview pane underneath the message list. Selecting a message by double clicking a message in the list will open it in a separate mail reading window which shows just the selected message and has an appropriate set of buttons. An example of the separate mail window is shown at figure 6.4

In the picture above, the **Next** button allows you to scan the message which is the one after the message you are currently reading. The buttons **Delete**, **Reply** and **Print** and other buttons do just what they claim for the message displayed.

Mozilla should check the main ANU e-mail server for new mail approximately every 10 minutes, however you may tell it to get any new mail *now* by clicking on the **Get Msgs** button wherever it appears.

## 6.5 Sending e-mail

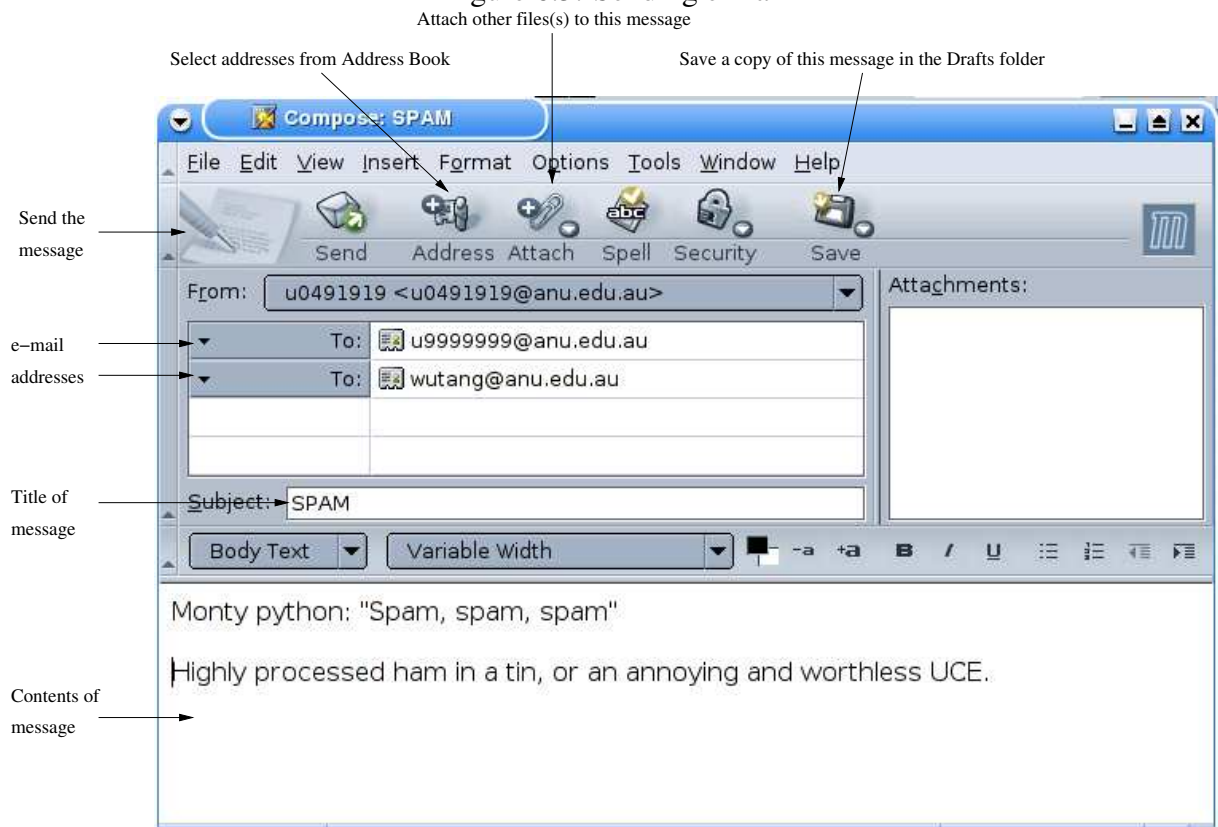
To create a new e-mail message you need to click on the **Compose** button in any of the mail windows. When you click this button, it will open a new mail composition window like the one

Figure 6.4: Reading e-mail



below (see Figure 6.5) You must fill in at least one e-mail address and the subject line, then you can write anything you want (subject to the guidelines). You can use the menus and buttons on the tool bar to take the actions indicated. You can send your message to the person(s) whose e-mail addresses appear in **To:**, **CC:**, and **Bcc:** lines by clicking the button **Send**. Clicking the button **Save** places a copy of the message in the **Drafts** folder and the compose window may then be closed. Double clicking on a draft message opens it up so that you may continue editing and/or send it.

Figure 6.5: Sending e-mail



# Chapter 7

## Editing text files

### 7.1 Introduction

A major part of the time you will spend using a computer is going to be devoted to creating files of text. These files might contain computer programs, or they might contain assignment reports. The process of creating and altering text files is called *editing*, and the programs used to help you do it are called *editors*.

This chapter will help you get started using an editor; the two likely suspects are called **KWrite** and **Emacs**.

### 7.2 Editors and their Pros and Cons

The following editors are available on our system:

- **vi** (or **vim**): this is a simple, terminal-based editor, widely available on Unix systems. It must be invoked from the Unix command line (see section 9.3), using a command like:

```
vim myfile.txt
```

For more information, use the Unix command `man vi`, and type `:help` when you start the editor (type `:q` to exit it).

- **KWrite**: this is the standard, GUI-based editor for KDE. It is relatively easy to learn from using its menus, and has keystroke shortcuts compatible with many non-Unix editors. It has some support for the main programming languages (including Haskell and Java), such as syntax highlighting (i.e. different parts of the file are displayed in different colours, according to the syntax of the programming language).
- **Emacs**: this is a more sophisticated GUI-based editor, and is freely distributable (under GNU Free Software) and available on a variety of computing platforms. It is also reasonably easy to start using, but its keystroke shortcuts are different from KWrite; they are however the same in the Mozilla editor, which you will need to use to compose for mail messages.

Emacs has better and wider support for programming languages, including finding where a variable or method is defined (useful in large programs over several files), help in matching parentheses and braces etc, and a commands to properly indent sections of code. It may be required for later CS courses.

It also has a terminal-based version, which is useful if you are connecting to our system remotely.

Emacs will also give you more protection against losing your work: it will maintain a backup copy of the file (from the last save), it will periodically save your file automatically, and (in some circumstances) it will attempt to recover the file if the system crashes while you are editing.

You will need to make a choice which editor you will use for the moment (perhaps after trying them); to help you, more information on the likely choices is given later in this chapter.

### 7.3 The load-edit-save cycle

It is important to understand the relationship of a text editor with the file system (see chapter 4). Files are not modified while they are stored in the file system. Rather, a text file is *copied* (also called *loaded*) into an internal buffer of a text editor, where it is modified as required. At this stage there are two copies of the file, namely, the original one in the file system (generally referred to as being ‘on disk’) and the modified one in the editor buffer (generally referred to as being ‘in memory’). The modified file needs to be *saved* from the editor buffer to the file system.

If the editor program is closed for some reason, or if the underlying computer system crashes for some reason, before the modified file is saved, then the modifications may well be lost. You have no doubt heard of computer users who have ‘lost a whole afternoon’s typing’ this way. Perhaps it has even happened to you.

### 7.4 KWrite

KWrite may be started from your main KDE menu under the **Editors** section. It has a slightly more advanced variant called **Kate**.

It may also be started from **Konqueror** by double-clicking on a text file’s icon. When the user double-clicks on an icon in the Konqueror’s window that represents a file, a default action is executed. In the case of a text file, it is to open it using KWrite.

It can be also be started from the Unix command line, using:

```
kwrite myfile.txt &
```

It is reasonably easy to learn how to use it from its menus. Further help can be obtained via the KDE help system (see section 3.3), under the **Application Manual** -> **Editors** section.

### 7.5 Emacs

As Emacs is somewhat more sophisticated, we will describe it in more detail below.

### 7.5.1 Starting Emacs

Emacs may be started by selecting it from the KDE Front Panel (middle button on left). The user is presented with the window of figure 7.1. Emacs supports a number of simultaneous buffers. As just indicated, a buffer can contain a copy of a text file. Emacs also uses buffers to display messages of various types. Figure 7.1 shows one such message buffer.

Figure 7.1: Initial Emacs window



It can be invoked from the Unix command line. For the file `HelloWorld.java` of figure 7.2, the command would be (from the directory containing the file, see section 9.3):

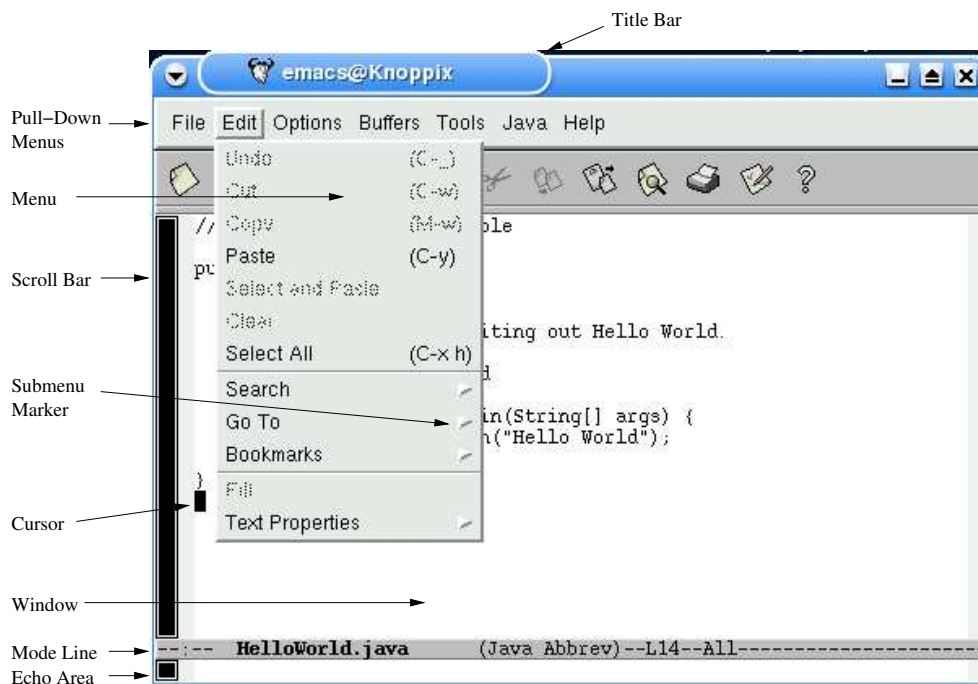
```
emacs HelloWorld.java &
```

Once Emacs is started, the file loaded into a buffer. This is indicated in figure 7.2.

Near the top of the frame, just below the title bar, you will find the menu bar, which has six menu titles: **F**iles, **E**dit, **S**earch, **B**uffers, **T**ools and **H**elp. You can see the items available under a menu by pointing at the title with the mouse, and holding down button 1 (the left button). To select an item from the menu you simply let go of the mouse button while pointing to it. To close a menu without selecting an item, just move the mouse off the menu before letting go of the button. The names of some menu items appear in grey rather than black. Such menu items can not be selected at this time. Some menu items end in a right-facing black triangle. These items contain sub-menus, and moving the mouse over them will reveal the sub-menu. You can select items from a sub-menu by sliding the mouse across to it, and then down to the item you wish to select.

You can also explore the menu structure by clicking rather than dragging. If you click mouse button 1 on a menu (or sub-menu) title, the menu (or sub-menu) will pop up and stay up, without your having to hold the button down. The menu will remain up until you click

Figure 7.2: An example Emacs frame



the mouse button again. Clicking on a menu item selects that item and closes the menu, while clicking somewhere off the menu just closes the menu without selecting any item. Try using this alternative method to explore Emacs’s menus, but again don’t select any items just yet.

To invoke Emacs as a terminal-based editor (this must be done from the Unix command line), simply use the `-nw` option. In the example above, you would type the command `emacs -nw HelloWorld.java` (no ‘&’ in this case). The editor will then open in the terminal window. Note that you will not be able to use menus anymore; you must use keystrokes for all commands; to exit, use the keystroke sequence `C-X C-C` (‘C-X’ means pressing the ‘Ctrl’ and ‘X’ keys simultaneously).

## 7.5.2 Emacs modes

In many cases a text file represents a highly structured document. The name of the file is usually chosen to denote this structure. If the contents of a file is a Java program then the file is given a name like `Wibble.java`, where the `java` suffix denotes an Java program (or class). Similarly, C programs are given the `c` suffix and Eiffel programs are given the `e` suffix. This document is written in a markup language called  $\text{\LaTeX}$ , and is represented in a text file with the suffix `tex`.

Emacs can be configured to respond to this naming convention, and to display the text file in an appropriate fashion (syntax highlighting). Emacs is correspondingly said to be in ‘Java mode’ or ‘LaTeX mode’, and so on. If the file contents have no particular structure, then Emacs operates in a basic ‘Text mode’.

### 7.5.3 Invoking and aborting Emacs commands

One simple way to use Emacs is to use keystrokes on the keyboard to enter text into the buffer, and to use the mouse to select the editing commands such as moving the cursor, saving files, and so on. The buttons on the menu bar provide access to all the Emacs commands.

Emacs predates the modern window-based user interface, and all Emacs commands can be specified using the keyboard alone. In fact, experienced users use Emacs in this way, even in a window-based environment, as it is faster to leave the hands on the keyboard than to continually move a hand from the keyboard to the mouse and back again. To this end several special characters are used to tell Emacs to interpret a command, rather than to put the keystrokes into the buffer.

For example, saving your buffer is something you should do quite regularly, and the process of selecting the `Save Buffer` item from the `Files` menu can become a little bit tiresome. You can do it quicker with the following keyboard short cut: hold down the `Control` key, then press and release the `x` key, then press and release the `s` key, then release the `Control` key. Try it now. In Emacs speak this sequence of key presses is abbreviated to `C-x C-s`. The ‘`C-`’ part means hold down the *Control* key. (The key might actually be labelled `Ctrl`.) Other common Emacs keyboard short cuts contain the abbreviation ‘`M-`’, requiring you to hold down another key that Emacs calls *Meta*, which is usually labelled `Alt` on your keyboard. One way to learn the most common keyboard short cuts is to look at Emacs’s menus. The keyboard short cuts for many commands that can be invoked from the menus are written in parentheses to the right of the menu items that invoke them.

If you get part way through invoking an Emacs command and decide you don’t want to go through with it, you can abort the command by typing `C-g`. This can be particularly useful later as some Emacs commands (such as searching a large file) take rather a long time to complete. Try using `C-g` to abort a `Save Buffer As . . .` command after you have selected it from the file menu. Hitting `C-g` a couple of times never hurts, and might help a lot if you somehow get stuck with Emacs and you don’t know what is going on.

### 7.5.4 How to learn more about Emacs

The `Help` button on the Emacs window provides access to a very large amount of information about Emacs. You should note that when help information is displayed, it is simply loaded into an appropriate Emacs buffer.

The Emacs Tutorial provides an introduction to Emacs basics, for first-time users. There is also a full Emacs manual. Select `Browse Manuals` from the `Help` menu and follow the instructions to select the Emacs Manual.

The Emacs Reference card, which contains many of the common keystroke commands, may be viewed or printed from the file `/usr/share/emacs/21.3/etc/refcard.ps`.



# Chapter 8

## Printing

### 8.1 Introduction

*Note: Printing arrangements in the Computer Science laboratories are expected to change (for the better) sometime this semester. The following information describes the current situation.*

KDE, together with the associated KJOBVIEWER and KPRINTER applications, can be used to print PDF (portable document format), PostScript and text files. They can also be used to get information about printers and to cancel previously submitted print jobs.

*Caution: only PDF (.pdf), PostScript (.ps) and text files are to be printed. Do not attempt to print compiled programs (as this may result in wasting vast amounts of paper)!*

### 8.2 Submitting a print job

There are several ways to submit a print job. Commonly the application you use to edit or view a file will have a *print* menu item under the *File* menu (see section 8.5) and selecting that menu option will usually bring up the KPRINTER tool (see Figure 8.1). Alternatively you can use mouse button 3 (right button) on the icon representing the file in the file manager (or on the desktop) to bring up a menu from which you can select the *print* menu item (see Figure 8.2). If the KPRINTER tool is open you may drag and drop a file icon into the KPRINTER tool.

This will indicate the name of the *default printer* in your laboratory room (e.g. `csitn112` if you are in Room CSIT N112). Clicking on the Print button will submit your job to the indicated printer.

#### 8.2.1 Choosing another printer

If the printer in your room is very busy, or is currently not working, you can send your job to a different printer by selecting the appropriate printer from a drop down list box of printers before clicking the Print button (see Selected Printer in above figure).

In the CSIT Labs, the printers are named after the rooms they are in, so for example, if you are in CSIT N115, and wish to send a job to the printer in CSIT N114, simply alter the name in the Printer field to `csitn114`.

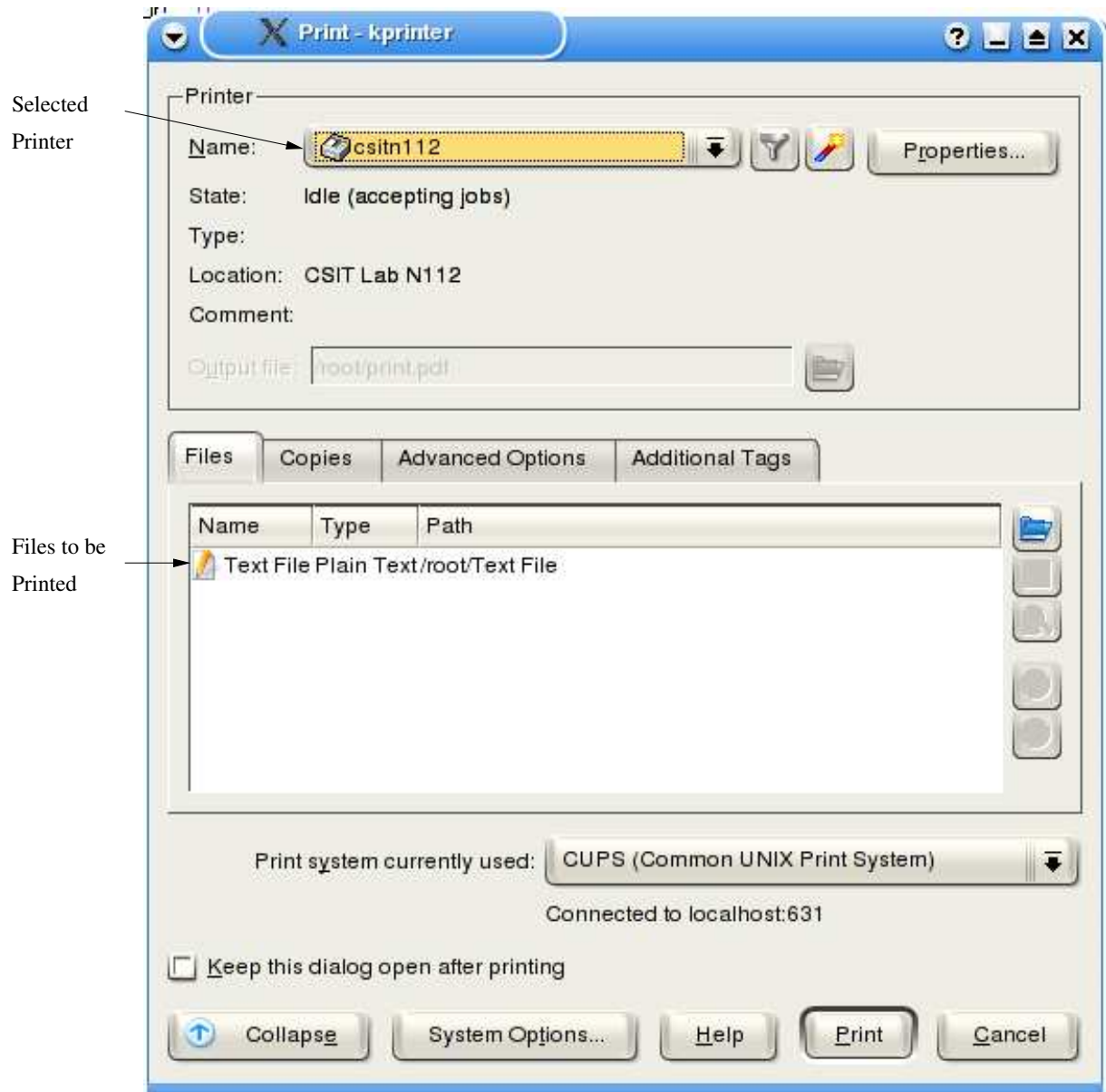


Figure 8.1: KPRINTER tool



Figure 8.2: File manager print menu item

Each printer has a set of properties that can be adjusted. The range of properties is somewhat dependent upon the particular printer. Two options that help reduce paper usage are duplex printing (i.e. both sides of the paper) and multiple pages per sheet of paper. Both these options can be selected via the properties options by clicking on the Properties button next to the printer selection box.

### 8.2.2 Other print dialogue box options

To print multiple copies of a file, select the *Copies* window on the KPRINTER tool dialog box. This will allow you enter the number of copies and the range of pages that you want to print.

## 8.3 Checking the status of the printers

In multi-user computer environments, jobs submitted to a printer will be queued.

It is often useful to check the *status* of a printer (i.e. what jobs are currently in the printer's queue) before submitting a print job to it. Also, if the printer is busy with other jobs, it is useful to check the progress of your job on the print queue, in order to know when it will be ready for collection. For the (default) printer in your room, this can be done by using the Printer Jobs application.

The KJOBSVIEWER window (see Figure 8.3) can be opened simply by clicking mouse button 1 on the printer icon on the KDE Front Panel, and selecting the desired printer on the menu. This window will display a queue of the current jobs on the available printers, including the user-names and the names of the files for each job. You may wish to view only the jobs queued on a

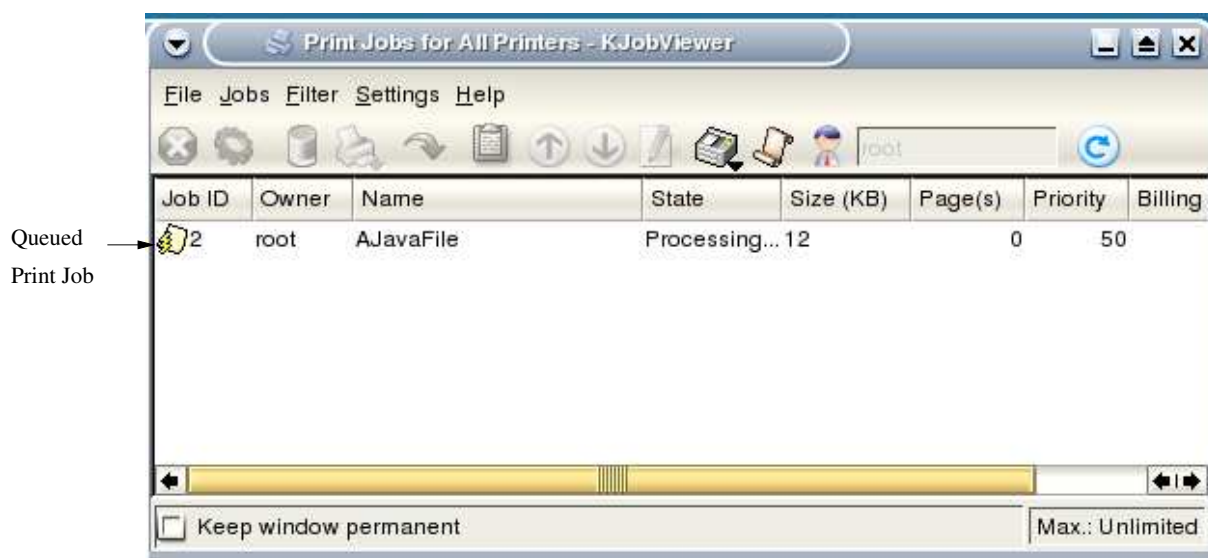


Figure 8.3: KJOBSVIEWER tool

particular printer. Selecting the particular printer (e.g. csitn112) menu item from the submenu *Select Printer* in the menu *Filter* will limit the display of jobs to those queued for that particular printer.

### 8.3.1 Cancelling a print job

You may sometimes find that the printer you submitted your job to has a long queue (perhaps it is even stalled, which can happen if it has run out of paper). In such situations, it is wise to remove your job from that printer's queue, especially if the job is part of a current assignment, before submitting your file to another printer.

Once the print queue for your printer is displayed on the *kjobsviewer* window, position the mouse on your print job's entry in the list of jobs, and click mouse button 3. This will bring up a pop-up menu; choose *Remove*.

## 8.4 On-line help and further information

The tasks described above will probably be adequate for most of your day-to-day usage of the printing facilities. To get help on these tasks, or to find information on the advanced use of the printing applications, you can access the *Help* menus of *KPRINTER* and/or *KJOBSVIEWER* windows.

## 8.5 Printing from within Applications

Some applications have a print capability from within the program window which may be used to print to your default printer. For example the web browser you may use the mouse to select the menu sequence *File*→*Print* and in Emacs you can use the *File*→*Print Buffer* or

File->Postscript Print Buffer menu selections. The latter is generally a more attractive and condensed printout, particularly for program code text files.

## **8.6 Sensible use of the printing resources**

The supply of paper for the printers is not unlimited, and involves significant costs to maintain. For this reasons, your tutor will bring paper to your formal laboratory sessions; at other times, you will need to bring your own A4-sized paper.

You should always immediately collect all printouts you initiate, particularly when the printout is assignment material that you have worked on and which must not be disclosed to other students.

Also, for work related to your assignments (e.g. a computer program), it is generally not necessary to make a fresh printout after every single session of work. In most circumstances, three printouts (e.g. in the case of a computer program, one soon after the program is typed in, one at an intermediate stage, and one for more or less the final version) should be adequate.

Your cooperation on this point will be greatly appreciated.



# Chapter 9

## Submitting Files and the Unix Command Line

This chapter describes how to submit files via the DCS submit system. The commands described below must be entered on a shell command line in a ‘konsole’ window. See section 9.3 for a brief description on using the shell and command line.

### 9.1 Submitting files for assessment

Many of your assignments in the Department of Computer Science will be required to be submitted for marking via the DCS computer system, rather than handed in on paper. It is *your responsibility* to ensure that you have submitted the *correct* files before the deadline. There are three simple commands which allow you to electronically submit files, to check that the files have been accepted and to get copies of files you have submitted so you can check their contents. The three relevant commands are:

- `submit <coursecode> <assignmentname> <filenames>`  
to electronically submit one or more of the files required for a particular assignment in a course. Files can be resubmitted up to the deadline specified for the assignment. Assignment sheets will specify the precise assignment name and filenames which must be submitted, and generally only those files will be accepted. For example an assignment sheet for COMP1100 may specify that you must submit files `Test.java` and `reporta1.txt` for assignment a1.
- `retrieve <coursecode> <assignmentname> <filename>`  
to retrieve a copy of the actually submitted file from the submission system. The copy of the file is deposited in your current directory and renamed by appending ‘`_MARKER`’ to the filename.

Use `STREAMS` to check the dates when you last submitted files for a particular course and assignment.

The way to use these commands is as follows:

1. Save your work and close any editor windows which have files which you are going to submit.

2. Bring up a `konsole` or `xterm` (terminal) window.
3. Change to the directory which has the files you wish to submit, for example:  
`cd /comp1100/assigns/a1`. You should list the directory to ensure the files you are about to submit are there and you may use the `more` command to quickly display a file to check it is the right one.
4. Use the `submit` command to submit your files, for example:  
`submit comp1100 a1 reporta1.txt`.
5. connect your web browser to `STREAMS` to see a list of the files you have submitted for the assignment specified.
6. Use the `retrieve` command to get a copy of the submitted file. You should closely check that this is really what you wanted to submit. If it is not what you expected, you will have to find and resubmit the correct file. For example, `retrieve comp1100 a1 reporta1.txt` will deposit a file named `reporta1.txt_MARKER` in the current directory. Use command `more reporta1.txt_MARKER` in the terminal window to display the file or use the GUI file-manager.

**WARNING:** Do not underestimate the importance of checking that you have submitted the correct file. Particularly with new users to the system, it is a relatively easy matter to submit the wrong copy of a file with the same filename. In the past this has often resulted in students submitting the original blank template of an assignment file instead of the copy with all their work in it.

After you have completed an assignment and submitted all files *use the retrieve command to check – look through the copy retrieved and at least check to see if your last amendments are there. You will get marked on the actual files you submit and as stated earlier: it is your responsibility to submit the correct files.*

## 9.2 Finding out your marks

This can be done via `STREAMS`. For further information on `STREAMS` check with TSG or your tutor.

## 9.3 Using the Command Line (Shell) in a `konsole` Window

This section is a mini-tutorial covering some *very basic* material to enable new users to access and use the command line for submission of assignment files and checking course marks as described in the sections 9.1 and 9.2.

For those familiar with MS Windows systems, this is similar to using a Command window or MS-DOS Prompt window. In a separate window a ‘Command Line Interpreter’ prompts you for a command then reads the next line you type. When you press the `Enter` key it attempts to execute the command using the command line arguments you type on the same line after

the command. Text output from the command is displayed in the window and a new prompt is issued.

In LINUX and KDE the window is called a `konsole` window (a popular alternative is called an `xterm`) and the command line interpreter is called a *shell*.

**To start a `konsole` window** select the `konsole` icon (depicting a monitor with a sea shell in front of it on the front panel) by pressing mouse button 1 (left mouse button) while the pointer is over the icon. Alternatively, you may select from the KDE menu the sub-menu `System` and menu item `Konsole (Terminal Program)`. When the shell is started up in the `konsole` window it operates in a 'Working Directory' which is by default your HOME directory. If you use the command to list files (`ls`) the shell will list the files in the working directory. If you enter a command to do something with a file (e.g. `more Test.java`) the shell will launch the program and send it the filename, and the program will look in the working directory for the file.

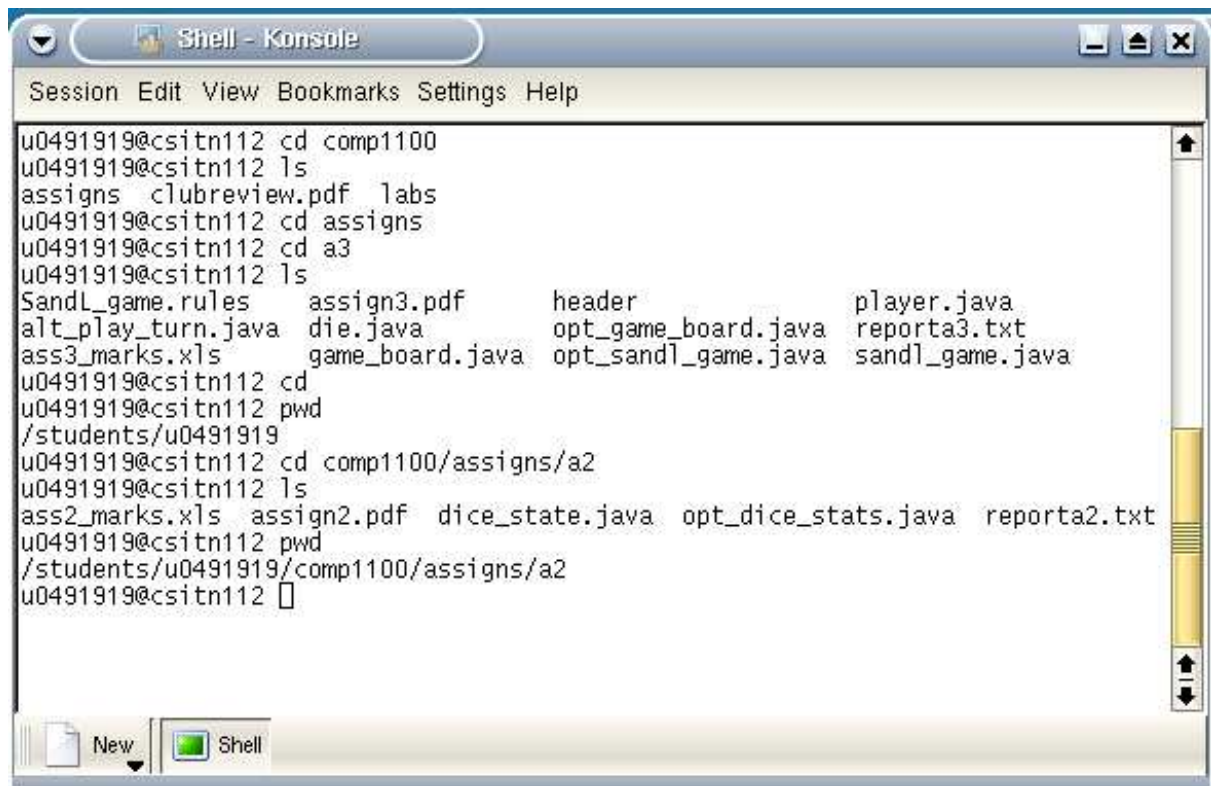
**To list files** in the Working Directory, use the command `ls`. To list more detail including read/write status and file date, tell `ls` to give a long listing by including command switch `-l`, ie: enter `ls -l`. Note the space between the command and the dash and no spaces between the dash and `ell`.

**To change and display the Working Directory** you use the `cd` command. `cd` by itself returns you to your home directory (NB: different from MS-DOS). To change to a subdirectory you simply specify that directory name, e.g. `cd assignments`. You may jump down more than one level at time using the *forward slash* (NB: different from MS-DOS backslash), e.g. `cd assignments/a3`. You can go one step up the directory tree using `cd ..` If a directory name has a space in it (*strongly discouraged*) you have to surround the directory name in double quotes (e.g. `cd "my projects"`). To display the Working Directory at any time use the Print Working Directory command `pwd`.

**To view a text file in a `konsole` window** use the `more` command, e.g. `more game.java`. The `more` program scrolls the text in the document one `konsole` window at a time. Press space-bar for the next screen full, `b` to go back a screen full, `q` to quit back to shell prompt or `?` for help on other commands while `more` is running.

**To edit a text file** if using Emacs simply enter the command `emacs` followed by a space, then the filename and then an ampersand (`&`). This will bring up a normal Emacs window with the file from the Working Directory displayed for editing. For example `emacs Test.java &`. If the file does not exist in the Working Directory Emacs will start with an empty buffer which may be saved to a new file as the name given. (The ampersand at the end of the command tells shell to run it in background and not to wait for the command to finish before re-prompting for your next command; so you can continue to use the shell without closing the Emacs window). If using KWrite, use the command `kwrite Test.java &` instead.

Figure 9.1: Sample konsole window with simple commands



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
u0491919@csitn112 cd comp1100
u0491919@csitn112 ls
assigns clubreview.pdf labs
u0491919@csitn112 cd assigns
u0491919@csitn112 cd a3
u0491919@csitn112 ls
SandL_game.rules    assign3.pdf         header              player.java
alt_play_turn.java  die.java           opt_game_board.java reporta3.txt
ass3_marks.xls      game_board.java    opt_sandl_game.java sandl_game.java
u0491919@csitn112 cd
u0491919@csitn112 pwd
/students/u0491919
u0491919@csitn112 cd comp1100/assigns/a2
u0491919@csitn112 ls
ass2_marks.xls  assign2.pdf  dice_state.java  opt_dice_stats.java  reporta2.txt
u0491919@csitn112 pwd
/students/u0491919/comp1100/assigns/a2
u0491919@csitn112
```

This is a VERY MINIMAL description which might help new command line users get to the desired directory, and enter simple commands to submit and check files and to check your marks. See figure 9.1 for an example konsole session with some simple commands.