CITS4211: Artificial Intelligence

Up Close and Personal with MacOSX

Aim

This exercise sheet introduces the MacOSX environment we will be using for this unit, including both the GUI environment and the underlying Unix operating system.

Getting Started

Start by logging in with your assigned password. Before getting too far you’ll want to access the on-line version of this exercise sheet so that you can follow the dynamic links. On the Dock (the application bar at the bottom of the screen) you’ll find the icon for Safari - MacOSX’s native web browser. Click on this to start Safari.

Notice that the menu bar at the top of the screen changes to the menus for Safari. This menu bar is context sensitive, and reflects whatever application is currently active. We’ll come back to this later.

At some stage Safari will ask you for your username and password for proxy and RSS servers. The former is needed to access sites off campus. The latter is for RSS feeds and not important for this lab - you can press cancel.

The proxy you need to use is:

proxy.uwa.edu.au

(which should appear by default when needed) and your central University username and password (the one you use to access your University email account, not your CSSE one). Before you can use this you will need to have upgraded your University account to a full service account. You should not need this for local addresses (ending in .uwa.edu.au) and can press “Cancel” if asked for one.

Your home page has been set by default to the CSSE Welcome page, which you should see in the Safari browser. Navigate to the Current Students page. You may want to set this as your home page. You can do this by clicking on the Safari menu in the top left corner and choosing Preferences. We will write this in shorthand as Safari | Preferences. Choose the General tab, and under Home Page click on Set to Current Page. Then click the red button to close the preferences window.

Unit Home Page

You can find this exercise sheet (and all subsequent ones) on the unit homepage. Scroll down to Artificial Intelligence and click on “Unit page”. This should take you to the AI Home Page. Whenever you visit the site you should check for any announcements by
clicking on discussion server under “Keeping in Touch”. Click this now to see what announcements have been posted.

If you haven’t done so before, it is important that you read the Unit Outline. This contains important information on matters such as lateness penalties, plagiarism and assessment.

After exploring the home page, go to the Lab Sheets. You should find an electronic copy of this document. Open the lab sheet and start working from the electronic version.

**MacOSX Basics**

A good starting point for learning your way around MacOSX is the [Mac 101](#) tutorial. Don’t do the whole thing now (I’m sure you’re too keen to get started), however there are four pages that are worth reading. These are Lessons 1, 3, 4, and 5 of the [Grand Tour](#). You could ignore Lesson 2, as you shouldn’t (ab)use the desktop by filling it up with junk as you do with some windowing environments - there are better ways of quickly accessing everything you need in MacOSX. In particular, the Dock provides a neat way of quickly accessing the most used applications, and the Finder sidebar provides a quick way of accessing parts of the filesystem hierarchy that you use most often.

Note that you will see some differences in the above lessons from what you will see on your screen. This is because your setup has been optimised for programming work. We will cover some of the differences below. The other applications mentioned in the lessons can still be accessed (via the Applications directory).

**The Finder**

Probably the most important application and the one you'll use most often is the Finder. This can be thought of as the Mac's user friendly interface to the unix operating system underneath. It allows you to navigate through the filesystem, navigate through the network that the computer is mounted on, open files and launch applications.

Open the Finder from the “happy face” in the Dock. At the top of the Finder you will see a toggle switch that looks like this . Click on the right hand “column view” icon. This is the recommended view for browsing as it shows you where you are in the hierarchical filesystem (or “tree”). It is recommended that you stick with this rather than the icon view that you may be used to from other operating systems. Once you are used to it you will find it more efficient and informative.

Create a new subdirectory (folder) in your home directory now, using the “Action” menu referred to in Lesson 1, called CITS4211. You can use this for your lab work for this unit. Now drag it to the sidebar for quick reference. (You can always remove it later by dragging it onto the background.)

**The Menus**

It is worth mentioning a few things about the menus at the top of the screen as this works differently to other desktop environments.

The apple on the top left is a system menu that is available in all applications. Some important things under this menu include:
• System Preferences. Used to set general or system preferences. As an example go to System Preferences | Keyboard & Mouse | Mouse. Here you can change what the different buttons on the “Mighty Mouse” do. Check that the right hand side of the mouse is set to Secondary Button.

You might also like to reduce the size and amount of magnification for the Dock as the default is rather large.

• Force Quit. Useful for killing an application that gets stuck or “hung” (as indicated by a “spinning beachball” that never stops).

• Log Out. This is where you log out when you finish your session (remember this for later).

The remainder of the menu is different depending on what application is active. However there are common conventions for all (good) applications.

The first drop down menu will be named after the active application itself. For example, click on the Safari window through which you’re reading this page (if its not already active). You will see a menu labeled “Safari” at the top. This menu is where you will find the preferences panel specific to the application, as well as the “Quit” button to quit the application. (Note: Many applications allow multiple windows. Closing a window by clicking the red button in the top left of the window therefore does not close the application, just the window. To close the application you need to go to “Quit” in the menu. This can be important if you are running out of memory because applications you have used but not quit are still resident in memory.)

As an example open Safari Preferences now and go to the Tabs pane. Check that all three boxes are ticked to enable tabbed browsing. (You may also wish to go to the RSS pane and change “Check for updates” to “Every Day” or “Never” if you don’t want to set up RSS now and you’re getting messages about proxy passwords.) Close the preferences window, go to View | Show Tab Bar, and check that right-clicking on the Safari tabs bar opens a new tab.

Before we leave Safari, go to View | Customize Toolbar and drag the Home button onto the Address Bar. This is useful for quickly finding course materials. Also select View | Show Status Bar.

While we’re on the topic of application preferences, open the Finder preferences and go to the Advanced page, and check that “Show all file extensions” is ticked. The hiding of file extensions is another “feature” popularised by another well-known operating system to “protect” the user from useful information. When we come to Unix later you will see that filename extensions are always shown - they are important.

The second menu is usually labeled File and contains options for opening, saving and printing files. The drop down menu also shows shortcut keys for commonly used options. For example Command-s is a useful one to remember for saving your work (which you should do often!)

The third menu typically contains editing commands.

At the right hand end will be the Help menu.
Other menus depend on the application, although there will typically be menus for managing windows and views (for example zooming in and out).

**The Dock**

The Dock has been mentioned earlier as a quick way of accessing regularly used applications. The Dock is also useful for showing you what applications are currently open (with the small light). If you have a lot of windows open, clicking on an open application in the Dock is useful for bringing its windows to the top. The right hand side of the Dock is where any minimised windows (from clicking the orange circle) sit.

Another useful tool for finding windows if you have many open is the F9, or Exposé, key.

**Developer Applications**

The applications used by most of the general public are contained in the Applications directory which for convenience appears in the Finder sidebar. Some of the applications we have installed for you for lab work also install here by default. Have a look in this directory and you will find, for example, the general purpose editor Emacs (which also appears on the Dock).

Another useful application is Preview (if this is not on the Dock drag it there now) which allows you to view a range of print and image file types very quickly. The pdf file for this exercise sheet that you are viewing in Safari is actually handled by a Preview plugin. You can swap it into a Preview window to give you more control over viewing, printing, etc - right click on the exercise sheet and select “Open with Preview”. This also frees your browser window for other things while you still view the Lab Sheet. Note: If you find at any stage that the Adobe Reader plugin is being used, you might like to ask Support to remove it, as it is significantly slower to start up. (Adobe plugins are like the Borg.) Normally when viewing files you will not need to explicitly open Preview, as it will open automatically if you double click on a file type that it handles.

As developers, you will also be interested in some other applications. There is a subdirectory of Applications called Utilities. This contains for example the Terminal program that we will use shortly (which also appears on the Dock) as well as other useful applications. If this has not been added to the Finder sidebar you should drag it there now for convenience.

As an example, lets say you want to see how hard your Mac is working. Open the Activity Monitor. This allows you to view CPU usage, memory usage, network traffic and other system functions, both overall and by application. If your system is slowing, perhaps due to excessive memory usage, you can look here to see which application might be the culprit.

Another directory you may be interested in is **Macintosh HD > Developer > Applications**. Here you will find Xcode (again also on the Dock) and, for example, Apple’s Performance Tools for optimising code. Developer also contains the Developer Documentation and Examples, tho these can usually be accessed directly through Xcode.

**Looking under the Hood: Accessing Unix**
Now that we’ve found our way around the MacOSX GUI, lets have a look at what’s going on underneath the bonnet.

Underlying MacOSX’s graphical interface is an open source Unix operating system (based on FreeBSD) called Darwin. Also included is the X11 windowing system used by the Unix community. This provides you with a lot of extra power and access to a huge range of open source projects. In fact many of the applications shipped in MacOSX use open source projects, or are provided by Apple as open source for use by the rest of the community.

A good example is the Apache web server included in MacOSX. If you go to the Apache Software Foundation website, you will see that the Apache HTTP server has been the leading web server since 1996. You will also see other projects that you may bump into during the course of your studies or working life. The Ant build tool, for example, is available through Xcode and will make an appearance in the next Lab Sheet.

Unix is also the platform of choice for some commercial applications. When we come to use Matlab, for example, you will see that it opens X11 to run under.

There are times when you want more control than the GUI applications provide, or when it is simply faster or move convenient to access the Unix operating system directly. In this section we will introduce some basic Unix commands to give a flavour of using Unix directly. For more detail you can access many Unix tutorials available on the web.

If you have used a Unix variant before, for example the linux system in our other labs, then you should be able to skip through this section quite quickly - it will look very familiar. If you haven’t, then you will also find this introduction useful when you come to use the linux machines.

Unix Basics

To access the operating system through a command line interface, click the Terminal program on the dock. This will open up a terminal window containing an operating system “shell”. It should say something like

```
Last login: Tue Mar 6 12:41:10 on ttypl
Welcome to Darwin!
csse2123%
```

The line ending in ‘%’ (it may alternatively use ‘$’ depending on the setup) is the prompt supplied by the shell. Anything you type into the terminal after the prompt will be interpreted as a unix command. In the following examples we will write ‘%’ as the prompt - you don’t type this, just what follows. Try:

```
% whoami
```

This will return your user name.

In the following try all the commands yourself as you read through the sheet.
Filesystem Commands

Just as the Finder is a core component of the gui environment, many of the commands you use most often in unix are related to navigating and modifying the filesystem. Try the following:

% pwd

short for “print working directory” will tell you where you currently are in the filesystem hierarchy or tree. Initially this should be your home directory.

% ls

short for something like “list” will list the contents of the directory. You should see the same subdirectories you saw in the Finder. Change into your CITS4211 directory (“change directory”):

% cd CITS4211

and check that worked (% pwd). Now create a new directory (“make directory”) for your Lab 0 work:

% mkdir Lab00

and check that it has appeared (ls). Now open the Finder and check that you can see the same directory there.

Rather than typing the full directory name, the unix shell will complete filenames where it can. Type the following without pressing return:

% cd La

then press the ‘tab’ key and ‘return’. The shell should complete the name. This is called ‘filename completion’. Check (using pwd) you are now in Lab00.

To move up the directory hierarchy you can use “..” to refer to the current directory’s parent. Try:

% cd ..
% pwd

You can also use absolute path names to jump straight to any point in the directory hierarchy (practice using filename completion in subsequent examples):

% cd /Applications
% ls

Finally,

%cd

with no arguments returns you to your home directory.
Running Programs

To run unix commands you type their name. Try:

```bash
% emacs
```

This starts an emacs editor in terminal mode, and can be very useful if you have to use a terminal, for example logging in from home. Normally however we will use the GUI version in the Dock. Press C-x C-c (control-x control-c) to close the emacs editor.

To find information on a command line application you can look at its on-line “manual” using the ‘man’ command:

```bash
% man emacs
```

This is a very useful command, and will also tell you the meaning of command line switches, or arguments you can pass to the program. (Use the arrows to scroll and ‘q’ to quit.) For example,

```bash
% emacs -version
```

will tell you the version you are using.

When using man pages it is usually more convenient to have at least two terminal windows (or “shells”) open - one to view the man page and the other to type the commands. Open a new shell using Terminal | Shell | New Window, look up the manual, and try running emacs in reverse video. (Note that some of the other options listed will not be available in terminal mode.)

Many programs also have a short help option, such as:

```bash
% emacs -help
```

You can also find out where in the filesystem the program you are running is stored:

```bash
% which emacs
```

Try all of the above options with the java command.

Shortcut Keys

There are many shortcut keys besides the tab key already mentioned for file completion. Luckily for us, the popularity of the Emacs editor has led to a de facto standard that is used in many places. In particular it is used in the shell that you are using (Z shell, or zsh) as well as other shells and, although it is not advertised, for the MacOSX applications themselves!

To demonstrate, open Carbon Emacs now from the Dock. Type a few lines of any old text. Now try hitting C-p (‘control-p’) which stands for “previous”. Notice the cursor moves up a line. Also try C-n (next), C-f (forwards) and C-b (backwards). Once you learn these they are much quicker than moving off the alphanumeric keypad to access the arrows.
Now move part way along a line of text and press C-k (kill). Notice the rest of the line disappear (it is actually “cut” and kept in the buffer). Then move to another position in the text and press C-y (yank), which pastes the text from the buffer.

Just learning these six keys will save you a lot of time. To demonstrate, leave Emacs and return to your terminal or shell window. Press C-p a few times. Notice this cycles through your previous commands. Similarly C-n takes you to the next command. Enter the following incorrect command:

```
% java -hilp
```

The java program will complain. Now rather than retyping the command, press C-p, then C-b until the cursor reaches the ‘i’. Remove the ‘i’ using C-d (delete), type ‘e’ and press return. Now press C-p again, C-b to the ‘-’, then press C-k to delete the rest of the line, and replace it with “version”.

Now lets try an application. Bring up a Safari window, and in the Google search field in the top right corner enter “emacs tutorial”. Lets say you decide after viewing the results that what you really want is a quick reference. Click somewhere in the Google search field, use C-f and C-b to position the cursor before the word “tutorial”, press C-k, and replace it with “quickref”. Now open the Calendar application from the Dock. Double click the name for that event, then use C-b and then C-k to rub out part of what you have typed. As you can see, nearly all MacOSX applications respond to these emacs commands. By learning the basic emacs commands you kill at least 3 birds (the editor, the shell, and the applications) with the same stone! These features are not advertised as novice users may be put off by non-gui commands.

Further information on these commands can be found in the Emacs and Java Editing and Emacs Quick Reference exercise sheets. These sheets are not tailored for MacOSX (they were written for a variant called Xemacs in the linux environment), but apply to using Emacs under all the operating systems in the labs (MacOSX, linux and Windows).

Cara MacNish
CSSE, UWA