What To Do

Read Chapters 1–2 of Glass and Ables to get an overview of the Unix philosophy and be introduced to some of the basic utilities provided by this system. As you read Chapter 2, it might be helpful to be within arm's reach of a computer[1]. To experiment with the various Unix commands, first launch **iTerm** (found on the dock) to gain access to a terminal window.

The computers in the Old Main laboratories use the Apple Mac OS X operating system. Strictly speaking, this operating system is not Unix, but since it is POSIX compliant it shares a great deal of functionality with traditional Unix systems. Thus, much of what is described in the Glass and Ables book applies to our labs.

The following exercises ask you to explore some facets of Unix—its historical context as well as some aspects of its user interface. Complete this assignment by adding your solutions in the indicated places, processing with pdfLaTeX to generate `hw01.pdf`. When you have completed all exercises, submit your PDF file in the usual way, dropping a copy onto the EIU submit icon.

Some Unix Commands

In the process of becoming familiar with Unix commands, you should be aware of these:

```
apropos  cat    chmod  clear  cp     date
file     head   ls     man    mkdir  more
mv       nano   pwd    rm     rmdir  stty
tail     touch  vi     wc     which
```

Exercise 1. To learn about the historical roots of Unix, explore the following web sites:

```
http://www.bell-labs.com/history/unix
http://www.multicians.org
```

then answer the following questions:

(a) Before Unix, there was an operating system known as Multics. Which three organizations were involved in the development of this operating system?

(b) The first version of Unix was developed for the PDP-7, a machine that cost approximately $72,000 with a memory consisting of 4K words. Each word on this computer was 18 bits wide. How many *bytes* did this computer have? How many times larger is the memory of the computers in Old Main 3041? (Consult the **About This Mac** menu item, using the Apple icon at the upper left of the screen.)

(c) In what year was Unix ported from the PDP-7 to the PDP-11?

(d) Name the two computer scientists who developed the Unix system.

(e) What were their fields of study while in college?

(f) Who designed the C programming language? In what year? What language was the inspiration for C?

[1]This assignment (and many others for this class) is intended to be done with a computer running Mac OS X, as found in the Old Main computer labs. While you might be able to do a large portion of these assignments on some other type of Unix-like computer system (such as Linux), there may be subtle differences which could cause problems. Ultimately, it your responsibility to ensure that what you turn in is compatible with the computers in our labs.

(g) Who is credited with the concept of *pipes*?

(h) According to the paper *The Evolution of the Unix Time-sharing System*, how much money did it cost (for the CPU time) to play a game of **Space Travel**? (To put this in perspective, a 1969 Chevrolet Impala had a list price of $2999.)

Solution.

Exercise 2. Explore the Wikipedia entry for POSIX, then answer the following questions:

(a) What does the acronym POSIX stand for?

(b) Is Mac OS X *fully* POSIX compliant? What about Linux? Windows?

(c) Why do you suppose full compliance might matter to a programmer?

Solution.

Exercise 3. The Unix manual pages are organized into sections. Sometimes it is important to search a specific section of the manual, as illustrated by these examples:

```
man 1 ls
man 2 read
man 3 exit
```

By examining these manual pages, what can you say about sections 1, 2, and 3?

Solution.

Exercise 4. Experiment with the `date` command, then consult its manual page. What command will display the date as a sequence of numerical values? For example, on January 5, 2012 we want the command which will produce:

```
01/05/2012
```

Solution.

Exercise 5. Interacting with a Unix system at a command prompt requires a *command-line interpreter*, a program known as a *shell*. As noted in Chapter 2, popular choices include the Bourne shell, Korn shell, C shell, and Bourne-again shell (bash). Which shell is being for your account? In what directory is its executable file stored? How many bytes long is this executable? According to its manual page, who are the authors of this shell? (Hints: the `help` and `which` commands will be useful here.)

Solution.

Exercise 6. According to the manual page for the standard C library function `exit`, what `#include` statement is required?

Solution.

Exercise 7. What is the purpose of the `apropos` command? To find out, read its manual page, then try the following examples:

```
apropos java
apropos python
apropos printer
```

Solution.

Exercise 8. As explained in Chapter 2, Unix defines a number of special metacharacters. These control characters can be determined with the `stty` command. Complete the table below by filling in the incomplete column of control characters. (The notation used by `stty` may look peculiar: for example, `^?` indicates the key combination control-?)

Solution.

Option	Meaning	Control Character
erase	Backspace one character	control-?
susp	Suspend the process	
kill	Erase all of the current line	
intr	Terminate the foreground process	
eof	End of input	

Exercise 9. Explain the effect of performing the following sequence of commands:

```
cd
mkdir -p 4970/testing/data
cd 4970/testing/data
touch alpha beta gamma
```

Solution.

Exercise 10. What single command will create a `bin` directory within your `4970` directory? Use a *relative* pathname and assume your current directory is the one from Exercise 9.

Solution.

Exercise 11. Give a sequence of commands which will create a file named `sample`, intended to reside in your `bin` directory. One of these commands should add executable permission for you, the owner. The `sample` file should have 0 bytes.

Solution.

Exercise 12. Using the editor `vi`, enter the poem of Figure 2.50 on page 59, storing it in the file `alpha` you created earlier. Explore the discussion on pp. 57–69 which describes this editor. Your goal should be to become *minimally* competent with this editor. If you find yourself needing to edit a file on a Unix system, this editor will almost certainly be available — one reason to be familiar with it.

When you have finished your exploration of `vi`, a special sequence of keystrokes is needed to exit the editor. What is this sequence?

Solution.

Exercise 13. Another commonly available editor on Unix systems is `nano`. Read the manual page for `nano`, then enter the same poem as before, storing it in the file `beta`. Aim for minimal competence — you don't need to become an expert in either of these editors.

What sequence of keystrokes allows you to save your work and exit the `nano` editor?

Solution.

Exercise 14. Using a text editor of your choice, enter the following in your `sample` file:

```
#!/bin/sh
for i in `jot - 4 6`
do
  echo hello, world!
done
exit 0
```

You can execute this shell script by giving the following command at a shell prompt:

```
~/4970/bin/sample
```

What is the result?

Solution.

Exercise 15. A file of words is stored at `/usr/share/dict/words`. Read the manual pages for `more` (or `less`), then try it out with this file of words. What keystroke is used by `more` to quickly move to the last line? How can we quickly move to the first line?

Solution.

Exercise 16. We want to know more about `/usr/share/dict/words`. Give commands with the following effect:

(a) Display the first ten words of the words file.

(b) Display the last five words of the words file.

(c) Display the total number of words in this file.

Solution.

Exercise 17. Some utilities are stored in `/bin`. What command will display all the files in this directory, sorted by their size starting with the largest? How would you adjust this command to show the sizes with an appropriate unit suffix — kilobyte, megabyte, etc?

Solution.