Worksheet 1: Using UNIX/Linux

Updated: 24th February, 2020

Note: You are encouraged to keep notes on your observations and answers to the worksheet exercises and lecture examples.

Assistance is only given when students can demonstrate their own attempt of resolving any problems.

Make sure you include things that went wrong. Your notes will serve as evidence of attempt, and often you can learn a lot analysing things that did not work.

Your notes should include answers to the following questions:

- What programs or websites did you use in the practical today?
- What do you need to remember about your practical work today?
- How does your program work?

1. Basic Commands

Use the **man** page to find information about the following commands.

Note: Yes, you can execute:

```
[user@pc]$ man man
```

```
man
apropos
which
whereis
whatis
who
```

You are looking for the c programming function time, so you execute `man time`, only to find that this provides information about the unix command `time`. Use apropos to locate the c function, then use man to read the details. What header file do you need to `#include`?

2. Prompt

Edit your **`.bash_profile`** to modify your prompt, then **`source`** it.

Change the `umask`, open another shell to see if it has stuck

3. Expand on the lecture

- Explore all of the commands from the lecture notes
- Execute history. Look for a command that you previously executed, and use `!n` to re-execute it, then use `!text` to re-execute it.
- Create a small text file, then execute:

  ```
  grep hello somefile.txt
  grep -i Hello somefile.txt
  ```

- Find a large text file (any `.html` will do) and save it in your home directory. Use the man pages to learn how to use `more`, `less`, `head` and `tail`. Use command switches to modify the default behaviour of each
- Use `ls` to find the modified time of a file. touch the file and use `ls` to see what happened. Use touch to create an empty file.
- Use `od`, `hexdump` and `xxd` to open a binary file.

4. The `find` Command

Explore the following commands:

```
find . -name "myProg.c" -print
find . -name "myProg.c" -exec chmod o+r '{}' \;
find . -size +500000 -print
find . -exec grep "www.curtin" '{}' \; -print
find / -mmin -10 -print
find . -perm 664
find . -perm -664
find . -mtime -1 -exec rm '{}' \;
    // be careful with what you are about to delete
```

5. Redirection

Write two c programs to demonstrate redirection. The programs will need to read from `stdin`, and writeto both `stdout` and `stderr`. Investigate how to set `noclobber`, then use the examples in the lecture notes to see the effects.

End of Worksheet