Unix commands: grep, chmod, wc, file

Have other laptop ready to run commands to test them.

File navigation:

Pwd – print working directory

Ls – list current directory

Cd – change directory

Relative path - Location of file or directory relative to our current location in the file system.

Absolute path – location of file or directory relative to the root of the file system (‘/’)

File manipulation:

Mkdir – make directory

Rmdir – remove directory

Touch – create an empty file

Cp – copies file or directory

Mv – moves a file or a directory (can also be used to rename file or directory)

Rm – remove a file

Displaying contents of files NOT EDIT IT

Cat – displays the contents of a file

Less – displays the file but allows forward and backward movement within it.

Head – list top 10 lines (by default) of a file (head -n3 to display the first 3 lines)

Tail – list bottom 10 lines (default) of a file (tail -n3 to display the last 3 lines)

Process management

CTRL – z – pauses the current foreground process

Bg – moves the process to the background

Fg – brings the process to the foreground

To continue running the process when logged off, you would need to use a window manager like **screen**.

Pipes and filters

Programs can output to other programs – piping.

Program\_1 | program\_2 – program\_1’s output becomes program\_2’s input

Program\_1 > file.txt – program\_1’s output and error logs are written to a file called “file.txt”

Program\_1 << input.txt – program\_1 gets its input from a file called “input.txt”

**Filters:**

A filter is a program which accepts textual input and transforms it in some way.

Filters can be connected together by pipes.

Filters can be thought of as building blocks to be easily put together to do what you want.

Eg: cat student\_attendance.txt | grep –e August | sort –> 3 filters

CAREFUL cat tomslee\*.csv | cut -f4 -d, | sort | uniq with this shit because

If you don’t sort it, the uniq function will not work because the uniq function is borderline special.

Wildcards to refer to multiple files

Wildcards allow you to operate on multiple files at a time.

If the command-line argument has a wildcard, your shell will replace it with a list of matching filenames.

**\* matches zero or more characters**

**? exactly one character**

**[abcde] exactly one character listed**

**[a-e] exactly one character in the given range**

**[!abcde] any character that is not listed**

**[!a-e] any character that is not in the given range**

**{debian,linux} exactly one entire word in the options given**

**\* -** any character *zero or more times* (eg. **rm \*.txt)**

**?** – any *single* character (eg. **rm a?.txt)**

**[]** – a range of characters (eg. **rm example\_[abc].txt** OR **rm example\_[a-c].txt)**

**Wildcards** can be combined.

(eg. **rm\*.???**)