***Linux Commands***

1. If 'mc' is not a typo you can use command-not-found to lookup the package that contains it, like this:

$ cnf mc

1. Display and Manage File System

$ ls (short for list)

$ tree

$ ranger

$ mc

1. SHELL (Bash): is a part of the operating system that defines how the terminal will behave and looks after running (or executing) commands for you.

See the Shell that is running (BASH is the most common Shell)

$ echo $SHELL

1. Print Working Directory

$ pwd

1. List Command

$ ls [options] [location]

-l (long listing option)

First character indicates whether it is a normal file ( - ) or directory ( d )

Next 9 characters are permissions for the file or directory (we'll learn more about them in section 6).

The next field is the number of blocks (don't worry too much about this).

The next field is the owner of the file or directory (ryan in this case).

The next field is the group the file or directory belongs to (users in this case).

Following this is the file size.

Next up is the file modification time.

Finally, we have the actual name of the file or directory.

/bin (location example)

1. The file system under linux is a hierarchical structure. At the very top of the structure is what's called the **root** directory. It is denoted by a single slash ( / ).
2. Absolute paths specify a location (file or directory) in relation to the root directory. You can identify them easily as they always begin with a forward slash ( **/** )
3. Relative paths specify a location (file or directory) in relation to where we currently are in the system. They will not begin with a slash.
4. ~ (tilde) - This is a shortcut for your home directory. eg, if your home directory is /home/ryan then you could refer to the directory Documents with the path /home/ryan/Documents or ~/Documents.
5. . (dot) - This is a reference to your current directory. eg in the example above we referred to Documents on line 4 with a relative path. It could also be written as ./Documents (Normally this extra bit is not required but in later sections we will see where it comes in handy).
6. .. (dotdot)- This is a reference to the parent directory. You can use this several times in a path to keep going up the hierarchy. eg if you were in the path /home/ryan you could run the command ls ../../ and this would do a listing of the root directory.
7. Change Directory

$ cd [location]

\*\*If you run the command cd without any arguments then it will always take you back to your home directory. \*\*

1. Linux is an Extensionless system.

$file [path]

In other systems such as Windows the extension is important, and the system uses it to determine what type of file it is. Under Linux the system ignores the extension and looks inside the file to determine what type of file it is. So, for instance I could have a file myself.png which is a picture of me. I could rename the file to myself.txt or just myself and Linux would still happily treat the file as an image file. As such it can sometimes be hard to know for certain what type of file a particular file is. Luckily there is a command called **file** which we can use to find this out.

\*\*\*whenever we specify a file or directory on the command line it is actually a path\*\*\*

1. Linux is Case Sensitive

\*\*\*Both with file names and command line options\*\*\*

1. Spaces in names

A space on the command line is how we separate items. They are how we know what the program name is and can identify each command line argument.

Quotes: Use either single or double quotes

$cd ‘Holiday Photos’

Escape Characters: Use what is called an escape character, which is a backslash ( \ ). What the backslash does is escape (or nullify) the special meaning of the next character.

$cd Holiday\ Photos

1. Hidden Files and Directories

$ls -a

1. The manual pages are a set of pages that explain every command available on your system including what they do, the specifics of how you run them and what command line arguments they accept. To exit the man pages press 'q' for quit.

$man <command to look up>

1. Searching

It is possible to do a keyword search on the Manual pages

$man -k <search term>

If you want to search within a manual page this is also possible. To do this, whilst you are in the particular manual page you would like to search press forward slash '/' followed by the term you would like to search for and hit 'enter' If the term appears multiple times you may cycle through them by pressing the 'n' button for next.

1. long hand command line options begin with two dashes ( -- ) and short hand options begin with a single dash ( - ). When we use a single dash we may invoke several options by placing all the letters representing those options together after the dash.