***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. See the Shell that is running

$ 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. \*\*