**Commands that work on other commands**

man [command] – Manual for the command (it shows you how to use the command)

*man samtools*

This will display the manual page for the samtools program which contains all of the different ways to use the command. Great if you don’t know what you’re doing!

sudo [command] – SUDO stands for Super-User DO, it’s the linux equivalent of “Run as Administrator”

*sudo rm my\_file.txt*

Removes a file “my\_file.txt” even if it is protected for some reason. Don’t use it if you don’t need it. Installing software almost always requires sudo, and this it is the main purpose for the command.

**Navigation Commands**

pwd – Lists the Present Working Directory (PWD), this is also the text in red shown below:

kyle@KylesLaptop:Users/kyle/Desktop$

ls [directory] – Lists all of the files and folders in the directory, or if no directory is given, lists everything in the Present Working Directory (PWD)

*ls ~*

Lists every file in your home (~) directory (Home is the linux equivalent of a windows Desktop)

cd [directory] – Changes the Present Working Directory (PWD) to the directory specified

*cd my\_folder/bio496/*

Changes the Present Working Directory (PWD) to go into my\_folder, and within that folder, it will enter another folder called bio496. To leave directories:

*cd ../..*

This will undo the first command, by going up two folders.

NOTE: There is a difference between absolute and relative paths that we will cover in class. An absolute path will start with “/”. Absolute paths can be used from anywhere, regardless of your Present Working Directory (PWD) and will always point to the same file or folder. Relative paths don’t start with a “/” and will depend on your Present Working Directory (PWD) and may function differently depending on which folder you are in. The two examples above use relative paths.

**Creating, editing, and deleting files**

nano [file\_name] – Opens the file in a basic text editor. If it does not exist, a new one will be created.

*nano student\_list.txt*

This command will open the file student\_list.txt in the editor or create it if it doesn’t exist in the Present Working Directory (PWD). Once you are in the editor, you can save the file with Ctrl + O and exit the editor with Ctrl + X.

mkdir [folder\_name] – Creates a new folder of the specified name

*mkdir test\_folder*

Makes a new folder called test\_folder in your Present Working Directory (PWD).

mv [file\_name] [new\_directory] – moves the file into the new directory

*mv grades.txt ~/bio496/*

Moves the file grades.txt into the folder “bio496” within the home directory (~). If you want to move a whole folder, and not just one file, you need to use the “-r” argument, as shown below:

*mv -r student\_projects ~/bio496/*

This command moves the folder “student\_projects” and everything inside that folder into the bio496 folder within the home (~) directory.

cp [file\_name] [new\_directory] – Same as *mv*, but copies instead of moves.

NOTE: Both mv and cp are used to rename files. There is no command to rename a file. To rename a file, specify its new name in the [new\_directory] argument:

*mv grades.txt ~/bio496/bio\_grades.txt*

This is the same example as the mv command shown above, but instead of just copying grades.txt into the bio496 folder of the home (~) directory, the file is also renamed to “bio\_grades.txt”

rm [file\_name] – deletes/removes the file specified.

*rm -r junk\_folder*

Deletes the folder “junk\_folder” and everything inside of it. The “-r” flag is not needed with single files, only when deleting folders.

NOTE: **DO NOT EVER TYPE**: rm -rf /

This harmless looking command will literally force delete every file and folder on your computer. It sometimes gets posted on the internet masquerading as a solution to a problem, don’t fall for it!