Linux Notes

* / represents the root of the system. Absolute path starts with /
* ~represents user home directory. For example user/johndoe/documents can be represented as ~/documents
* For file or folder names with a space use escape character (\) before the space. For example for folder name new folder : ~/documents/New\ Folder
* ls -R directory\_name lists all files and folders recursively inside the directory
* Cd – takes your to the previous directory and cd alone takes you to the home directory
* Ll lists all documents including hidden ls -l lists the same without hidden files
* To create multiple directories with mkdir use space between folder names. For example
* mkdir Linux/folders/new\_folder1 Linux/folders/new\_folder2 Linux/folders/new\_folder3
* To create a directory with sub-directories use -p For example
* mkdir -p Linux/folders/new\_folder1/sub\_folder1/sub\_sub\_folder
* If you want to copy a file to the same directory just give it another name and write it after space. For example cp text text1
* Wildcard \* character can be used to move or copy all files notwithstanding the file name or extension. For example mv Linux/folders/\* . will move all files from folders to our current location.
* Su command means substitute user.
* Sudo changes to super user with root privileges, sudo -s gives you privileges, sudo -k takes back root privileges. If just for a short time just sudo command like sudo ls /root will work. For longer times use sudo -s and the username will change to root.
* For file permissions if we use = sign for permissions, like a = wr, = resets the permissions to the new specified value, to keep the previous and add or remove some permissions, use + and – plus wrx. Using numbers is expressed as using octal value.
* Stat myfile.txt gives you information regarding the file including permissions by octal value. Stat command gives information such as the size of the file, access permissions and the user ID and group ID, birth time access time of the file.
* Pipes | connect several commands and create a pipeline
* Grep command helps find strings of text. For example, grep “the” poems.txt finds all instances of the in a file named poems.txt and display them in the terminal. Grep -n also gives line numbers for the results. Use grep -i to disable case sensitivity in the search. (grep -in for adding line count as well). Grep -v “the” to drop the lines that include “the” from the results
* Using sed we can replace words in a text file without using another editor. For example sed s/Orange/Red/ sample.txt will change all occurrences of Orange to Red. S stands for substitute and do not forget to end the word with a /
* Rev command reverses all the text in a text file. For example, rev data.txt
* Tac writes the text upside down. First line becomes last line.
* In Vi command mode “o” takes you to a new line. If you just opened vi without a file, after you write some text press esc :w file.txt to save it
* For compression of files and folders tar command is used. You can not compress the folder that you are in. You need to cd .. back. Then write tar -cvf filename.tar Foldername. C is for compress, v is for verbose and f is for write to a file. Then to extract them, use the command tar -xf extractedfilename to extract it to current working directory
* Zip command also compresses files and fodlers.zip -R newarchive.zip foldername. To extract mkdir a folder and write unzip newarchive.zip
* > is called redirection. Use > to overwrite a file and >> to append a file. Also if you use no text before the > sign and write a filename after it, you simply replace the contents of the file with an empty string so you clean it.
* Which command helps us to find where a specific command is located. For example which ls
* Uname -a helps us get information regarding our system