**LINUX COMMANDS**

* **cd /filename/filename/ : change directory or move to the specified location**
* **cd ~ : go to home directory**
* **cd ../filename/ : go one step back and move to the specified filename**
* **cd filename : go to the specified filename in the current directory**
* **cd - : switch back to previous step**
* **cd .. : go to parent directory**
* **cd ~/filename && ls : change to specified filename and list all the files**
* **cd ../../../filename/filename : go three steps back and move to the specified locations**
* **ls : displays all the files in the present location**
* **ls ./foldername : displays the files in the specified directory or folder**
* **ls -l : shows file or directory, size, modified date and time, file or folder name and owner of file and its permission**
* **ls -a : view hidden files**
* **ls -r : shows list of files in alphabetic reverse order**
* **ls -ltr : shows list of commands in latest modification**
* **ls -lS : shows by file size**
* **ls –help : list of ls commands and their usage**
* **su username : switch to specified user**
* **nano ./filename : opens a file in texteditor in just readable format (cant be edited)**
* **sudo nano ./filename : opens a file in text editor in writable format (editable)**
* **ctrl + o : save file in text editor**
* **ctrl + X : exit from text editor**
* **sudo !! : run the previous command with sudo**
* **apt -get : to install applications. Sometimes this doesn’t work because all directories may not have permission for applications**
* **sudo apt-get install application\_name : install specified application**
* **sudo apt -get remove application\_name : unistall the specified application**
* **apt -cache search application\_name\* : searches for applications that have the specified name in it**
* **apt -cache policy application\_name : searches for applications that have the specified name in it installed in your computer**
* **dpkg --get -selections : shows all the applications installed on your pc**
* **sudo dpkg -i /(location of the file)/full filename : installs the specified file**
* **sudo apt -get upgrade : shows the applications that are upgradable**
* **sudo chown user:group filename : changes the specified file with the specified username to the specified group**

**chown-change ownership**

* **4 – readable, 6 – readable and writable**
* **sudo chmod 646 filename : usually the file will be in 644, means user is able to read and write but group and public are only able to read.**

**This command changes the file for the public to read and write format**

**Chmod- to change permissions**

* **sudo chmod 664 filename : this means user and group can read and write**
* **rm filename : delete a file**
* **mkdir filename : create a file with specified filename**
* **sudo mkdir filename : creating a file as a admin with specified filename.**

**If you create a file as a admin it will have rwx that means the file is readable, writable and executable for user, executable and readable (xr) for group and executable (x) for public. If you create a file as admin (use sudo) then the user and group by default are going to be root**

* **sudo chown -R username:username ./foldername : this means all the files in the specified folder name comes under the ownership of specified username. Sudo is used because if you create a file with sudo it will be under root admin and it cannot be written. So using sudo we will change the ownership to the specified username.**

**-R means recursive.**

* **sudo chown username:username ./foldername : this changes the ownership of the specified directory but not the files**
* **touch filenmae1 filename filename 3 : create files/file**
* **rm ./\*.txt : remove all files that have txt(any specified) extension**
* **rm foldername/\* : remove all the files in the specified folder(or directory)**
* **rm -rf foldername : removes specified folder (or directory)**
* **ipconfig : displays the ip address of your machine**
* **ifconfig : displays all the details of all the machine connected to your machine**
* **cp filename1 ./foldername/filename2 : copy the contents from filename1 and put it in the specified location(folder name) and name it as filename2**
* **rm filename : remove the specified filename**
* **rm dir/filename : remove the specified file from the specified directory**
* **mv filename1 directory/filename1 : takes the filename1 from present location goes into the directory and puts it as filename1**
* **mv filename1 filename2 : renames filename1 by filename2, leaving no copy**
* **find /directory -type f -name “\*.txt” : finds and brings all the files with .txt from the specified directory. Under name you can give filename. In the place of name if you put iname it becomes case sensitive**
* **find . -size +100k : finds and brings files in the current directory which have size greater than 100kb. If you put – then files lower that 100k are displayed**
* **grep “function” filename1 filename2 filename3 : searches for function inside the specified filename/filenames and displays it**
* **grep -i “term” ./\* : searches for the term specified in the current directory. -i means ignore case sensitivity**
* **grep -n -i “term” ./\* : n gives the line number from the file where the specified term is found**
* **find . -type f -iname “\*.php” : finds for all the php files in the current directory (file type only, . refers to current directory)**
* **find . -type f -iname “\*.php” -exec grep -i -n “function” {} + : what this does is, first executes find command and then executed grep command. {} + is used to close the flag**
* **top : displays all the running applications**
* **pgrep application(liri-browser) : gets you the process ids of an application. That means if three internet browsers were opened in your system. This command displays the users.**
* **kill -9 processid(s) : closes the window of browser of the respective processid/ids**
* **killall application : closes all the application with different process ids**
* **sudo apt-get install git git-extras : installation of git**
* **git init : initializing git repository.**

**Go into the file location ypu want to use for git and initialize**

* **git remote add origin linkofgithub : adding git origin**
* **git config --global user.name “Name” : giving username of git**
* **git config –global user.email email : giving email id to git**
* **git pull : pull the contents from github**
* **git pull origin master : the branch where you want to pull information from**
* **git rm -r .nameofdirectory : removes the specified directory**
* **git rm -r -f .nameofdirectory : removes the specified directory if the file is modified, -f should be used**
* **git -ignore .directory/\* : ignore the directory**
* **sudo adduser username : creates new user for the system**
* **sudo adduser username sudo : creates a user with username and adds it to the group sudo**
* **sudo touch filename : creates a file**
* **sudo rm filename : removes a file**
* **sudo deluser username : deletes a existing user**
* **sudo passwd userame : change password of the specified username**
* **ssh usernameonremotehost@remotehost/ipaddressofremotehost : helps to connect to remote server**
* **echo $applicationname : gives the environmental value assigned for the specified application**
* **whereis application : gives the location of the application**
* **sftp usernameonremotehost@remotehost/ipaddressofremotehost : helps to connect to remote server. Similar to ssh**
* **vi filename : takes you into the vi editor. To insert content you need to press i and then enter content. After pressing esc you go into command mode. To exit from command mode: (these commands are to be used in command mode)**

1. **:w : to save your file but not quit vi (this is good to do periodically in case of machine crash)**
2. **:q : to quit if you haven’t made any edits**
3. **:wq : to quit and save edits (basically the same as ZZ)**
4. **:q! : force quit**

* **Sometimes direction arrows doesn’t work. You can use letters:**

1. **h : left one space**
2. **i : right one space**
3. **j : down one space**
4. **k : up one space**

* **Some commands in command mode:**

1. **dd : to delete or cut the current line (3dd will cut 3 lines)**
2. **p : paste**
3. **dw : delete from the current cursor location to the next word**
4. **yy : copy the current line (3yy will copy 3 lines)**
5. **yw : copies a word (7yw will copy 7 words) copies from current cursor location to next word**
6. **to search for a word in command line type /wordyouwanttosearch**
7. **n will search in forward direction and N will search in backward direction**
8. **r : replace one character at a time (under cursor with the nect character typed)**
9. **R : keep replacing character until esc is hit**
10. **u : undoes the last change you have done**
11. **ctrl + r : redo**
12. **x : delete a character**
13. **if you don’t know in which mode you are then you can press esc twice and it will bring you back to command mode**
14. **if you want to complete a word, enter first word and ctrl+p, it will list out words that start with the letter**
15. **if you want to complete a line, ctrl+x and then ctrl+l, it will list out the lines**

**SUMMARY:**

**\* Directory commands – pwd,ls,cd,mkdir,rmdik**

**\* File commands- file, touch, rm, cp, mv, rename**

**\* Man commands—( man -aw), (man -a), (sman -k), -f, whatis, whereis**

**\* File Contents commands – head, tail, cat, tac, more, less**

**\* Linux Directories**

**\* Filter commands: cat,cut,grep,comm,sed,tee,tr,uniq,wc,od,sort,gzip**

**\* Unix tools : find, locate, date, date, cal, sleep, df**

**\* Users commands – whoami, who am i, who, w, id**

**\* Vi editor**

**Commands – :wq, :w, :q, :w!, :q!**

**\* Networking Commands—ping,host, nslookup, dig, whois, route, traceroute, tracepath, ip netstat**

**\* File Security:**

**\* File ownership: Users,Group,Other**

**Commands— cut | column**

**Chgrp – changes group**

**Chown – changes owner**

**o Permissions: Read, Write, Execute**

**Symbolic - chmod r+w**

**Assignment- chmod 777**