Linux Commands 🔥

By Garvit Singh, IT Sophomore

1. pwd - Prints the current working directory.
2. cd - Change working directory. Examples :
    - Change directory from Home directory(represented by ~) to Downloads. When you are somewhere inside your Home directory, the terminal uses ~ as an abbreviation.
        - cd Downloads
    - Go up to the parent directory. '..' means previous directory. '.' means current directory.
        - cd ..
    - Move up through mutiple levels of parent directories.
        - cd ../..
    - Switch to the root directory, then follow the route from there towards the directory 'etc'
        - cd /etc
    - cd command requires you to provide atmost one arguement or no arguement. Cannot be more than one.
3. whoami - Prints your username.
4. mkdir 'Directory Name' - Create a new directory. Examples :
    - Create a folder named 'abc' in the current directory.
        - mkdir abc

- Create a folder named 'gs' in the Downloads directory. It will first go to the Home directory(represented by ~), then go to the Downloads directory and create a directory named 'gs' in it.
    - mkdir ~/Downloads/gs
- Switch to root directory, then create tmp directory and src directory inside tmp.
    - mkdir /tmp/src
- Creates all folders in the current directory.
    - mkdir dir1 dir2 dir3
- Here, dir4 is created and dir5 is created inside dir4 and dir6 is further created in dir5. We use the -p switch which creates the parent directories as decribed in the arguement.
    - mkdir -p dir4/dir5/dir6
    - mkdir command requires you to provide atleast one arguement.

5. ls - list command. Displays files & directories in a directory.
    - Use the -a switch to list the hidden files like the .bashrc file or .zshrc file.
    - Capture the output of the command in a text file, we use the greater than(>) sign.
        - ls > result.txt
6. echo - Prints the provided arguement in the terminal.
    - echo "Hello"
7. cat - Displays the content of the file in the terminal. If you pass more than one filename, it will output each one of them, one after the other. 'cat' comes from the word concatenate which means to link together.
    - Display the content of sample.txt stored in dir1.

        - cat dir1/sample.txt
8. mv - Move command. mv command lets you move more than one file at a time. If you pass more than two arguements, the last one is taken to be the destination directory and the others are considered to be the files or directories to be moved.
    - Here, abc.txt and dir1 will be moved to dir2
        - mv abc.txt dir1 dir2
    - Here, abc.txt(located in dir1) will be moved to dir4 which is inside dir3 which is further inside dir2.
        - mv dir1/abc.txt dir2/dir3/dir4
    - To rename a file, use the mv command. Here, abc.txt is renamed to xyz.txt. Works on both files and directories.
        - mv abc.txt xyz.txt
9. cp - Copy command.
    - Here, sample.txt which is inside dir1 is copied to dir3.
        - cp dir1/sample.txt dir3
10. rm - Remove file command.
    - Remove sample.txt stored in dir2 which is further inside dir1/
        - rm dir1/dir2/sample.txt
    - Remove all files starting with 'h' in the current directory
        - rm h*
    - Remove everything in the current directory

    - rm *
  - Remove dir1 and delete everything in it even if there were files in it. Use -r switch for this.
    - rm -r dir1
  - Use -i switch with rm command which will prompt you to confirm the deletion of each file. Here, we are deleting all files(represented by *) in the dir3
    - rm -i dir3/*
11. rmdir - Remove directory command. If there are any files in the directories, then they will not be deleted.
  - Here, we are deleting multiple directories at the same time which are stored in a hierarchy. Use the -p switch with rmdir command for this purpose. It deletes dir3 first, then dir2 is deleted and finally dir1 is deleted. Files contained in these directories will not be deleted.
    - rmdir -p dir1/dir2/dir3
12. wc - word count command. Use the -l switch to know line count.
  - Here, ls ~ lists the content of the home directory and wc -l counts the lines. The vertical bar ' | ' is the piping operator.
    - ls ~ | wc -l
  - Count the lines in the /etc directory.
    - ls /etc | wc -l
13. less - Display file's content one screen at a time. This can be used if a file output is very large.
  - Display the content of sample.txt
    - less sample.txt

14. uniq - Output unique lines in that file.
    - Display the content of sample.txt and count the number of unique lines.
        - cat sample.txt | uniq | wc -l
15. man - Instruction Manual command.
    - Open instruction manual for rmdir command.
        - man rmdir
16. sort - Sort in different kinds of ways.
    - sort : Sort alphabetically
    - sort -r : Reverse alphabetical sorting
    - sort -f : Case insensitive sorting
    - sort -n : Sort numerically
17. su - Allows users to switch to the root account and perform administrative tasks when passed with no username as arguements. If given a username, then it switches to a specific user account.
18. logout - Logout from superuser or root account back to normal
19. sudo - It is used to run a command with superuser privileges. A configuration file is used to define which users can use sudo and which commands they can run. When running a command like this, user is prompted for their own password.
20. reset - Clear the terminal
21. where - To get the path or location of a file in the directories.
22. open - Open a directory or path

23. tr - Translate command to translate, squeeze and/or delete characters from standard input, writing to standard output.
24. touch - Create a file
    - Create a text file in dir2, which lies inside dir1.
        - touch dir1/dir2/sample.txt
25. df - Display disk space usage and similar information. Use -m switch to display sizes in MBs insteads of KBs.
26. du - Display disk usage statistics.
27. head - Displays text starting from above.
28. tail - Displays text starting from below.
29. diff - Compares content of two files and outputs every line that doesnt match.
30. locate - Locate files
    - Locate all files that end with .txt
        - locate *.txt
31. find - To find files & directories inside a directory. Shows hidden files as well. It is similar to ls but has got more functionality, which are shown in the examples.
    - Find all directories within the current directory. (.) the dot represents the current directory.
        - find . -type d
    - Find all files in current directory.
        - find . -type f
    - Find file named sample.txt in the current directory.

    - find . -type f -name sample.txt
- Find all files starting with abc in current directory.
    - find . -type f -name abc*
- Find all files ending with .txt in dir2, which lies inside dir1 which further lies inside Home directory.
    - find ~/dir1/dir2 -type f -name *.txt
- Do the same but now it is case insensitive search.
    - find ~/dir1/dir2 -type f -iname *.txt
- Find all files that were modified less than 20 minutes ago in the current directory.
    - find . -type f -mmin -20
- Find all files modified more than 15 minutes ago in current directory.
    - find . -type f -mmin +15
- Find all files modified less than 10 days ago in current directory.
    - find . -type f -mtime -10
- Find folders with 1 file depth in current directory.
    - find . -type f -maxdepth 1
- Find all files having size more than 1kB.
    - find . -size +1k
- Find all empty files and directories.
    - find .
- Find files which have 777 permission in current directory.

    - find . -perm 777
32. chmod command - File Permissions
    - 4 : Read
    - 2 : Write
    - 1 : Execute
    - 0 : No permissions
    - Ex - If you want to give write and execute permissions, then 2 + 1 = 3
    - Give Read, Write and Execute permissions to User, Group and other in file named sample.txt stored in dir2.
        - chmod 777 dir2/sample.txt
    - Give Read, Write and Execute permissions to User, Read and Execute permissions to Group and only Read permission for others in the file abc.txt stored in dir3. 'u' stands for user, 'g' stands for group, 'o' stands for others.
        - chmod u=rwx, g=rx, o=r dir3/abc.txt
33. exec - Executes a terminal command without creating a new process. Instead, it replaces the currently open Shell operation. Can be used to rename a large number of files at once.
    - Find and delete all files ending with .txt
        - find . -type f -name *.txt -exec rm -rf {} +
34. grep - Used to search for things written inside a file.
    - Search the word 'Garvit' inside names.txt
        - grep Garvit names.txt

- Display complete word
  - grep -w Garvit names.txt
- Case insensitive search
  - grep -i garvit names.txt
- Display the line number
  - grep -n Garvit names.txt
- Combine all three switches.
  - grep -win Garvit names.txt
- Display 3 lines that come before the searched word.
  - grep -B 3 Garvit names.txt
  - Search for all files having 'Garvit' and ending with .txt in current directory.
    - grep -win Garvit ./*.txt
  - List all files containing 'Garvit' in current directory.
    - grep -wirl Garvit .
  - Count how many files contain 'Garvit' in current directory.
    - grep -wirc Garvit .

35. history - Shows the history of all past commands executed.
    - Show history of all 'mkdir' commands using grep.
      - history | grep "mkdir"
36. clear - Clear the terminal screen.

37. jobs - Displays current jobs running in the shell.
38. ping - Displays connectivity status.
39. wget - Download files from internet.
40. top - Display what all processes are running in the CPU.
41. kill - Kills a running process.
42. zip - Compress files and add to zip folder.
43. hostname - Displays domain name, hostname, system name. Use -i switch to get IP Adrress.
44. useradd - To add a new user.
45. userdel - To remove an existing user.
46. lscpu - Get CPU details.
47. free - Check used and free memory.
48. vmstat - Virtual memory statistics.
49. getent - To check if a user exists in a group or not.
50. lsof - List all open files.
51. nslookup - Find IP Addresses of a particular domain.
52. netstat - List active ports
53. cut - Cut out some portion from a file.
54. Operators
    - & - Used to run commands in the background so that other commands can be run.
    - && (AND) - Execute a command only if the command preceding to it is successfully executed.

- | | (OR) - Execute only if the previous command fails
- | (Pipe) - Combine multiple commands
- '>>' (inside) - Transfer results inside the specified file without over-riding previously stored data in the file.
- '>' - Completely overides the content stored in the specified file.

55. Terminal Keyboard Shortcuts
    - Ctrl + A : Go to beginning of a line
    - Ctrl + E : Go to end of a line
    - Ctrl + K : Remove everything in line after the cursor
    - Ctrl + U : Deletes entire line.
    - Tab : Autocompletion
    - Ctrl + R : Search for commands in history.
    - Ctrl + L : Clear the terminal screen
    - Ctrl + Alt + D : Open Terminal
    - Ctrl + D : Close Terminal
    - Note that these shortcuts might vary depending upon which Linux distribution you are using, or how you have cutomised the settings of your terminal.

Thanks For Reading! 💙

By GARVIT SINGH

Information Technology Undergraduate