Linux Basic Commands For Beginner

man command_name: to know more about the command in terminal

1) **pwd command**: present working directory

2) **cd command**: change directory
- **cd directory_name**: goes inside the directory
- **cd ..** : get outside the directory
- **cd /** : root directory
- **cd ~** : home directory
- **cd *My Book*** : to enter inside the directory which have space

3) **ls command**: list out the directory and files
[ls option file/directory]
- **ls /** : root directory
- **ls ..** : list out from the one step back from present
- **ls ../..** : two step back

in option
-l : in long format
-a : show hidden files
-al : hidden in long format
-lS : short by size
/*.html : file which have have only html extension
/*.* : all file with all extension
-ls > output_file : save the output in file
-d */ : only directory

4) **cp command**: to copy the files and directory
[cp options sources destination]

in option
-i : prompts you before overwriting.
-n : avoids overwriting without prompting.
-R : recursive copy(copying the directory)

5) **cat command**: related to text file (display, combining, creating)
[cat option files_name]
- **cat** : echo the input
- **cat file_name**: to print the content of file in terminal
- **cat -E file_name** : add $ at each of the end of line
- **ctrl + D** : to get out of the cat command

6) **I/O redirection**: capturing output from the file, command or program and sending it to another file, command or the program as input
[output > file_name]
- **cat > file_name.txt** : (creating the file) convert enter text from terminal into text file
- **cat >> file_name.txt**: to append
- **cat file1 file2 > combined_file**: combined_file = file1 + file2(>> → for appending)
- **cat file1>> file2** : file2 = file1 + file2

7) **mkdir command**: making directory

- **mkdir directory_name:** creating the directory
- **mkdir directory_name/ sub_directory_name** : create sub directory inside the existing dir
- **mkdir -p directory/sub_directory_name** : even though directory doesn't exist
- **mkdir -p directory/ {sub_directory_lists}**: to create multiple sub directory

8) **rm and rmdir command**: remove the file and directory

- **-r** : recursive remove

9) **mv command** : to move files from one location to another

- **[mv options source destinations]**
  in options
  -i : for overwritten or not

10) **less command** : to read the file

- **less file_name.txt**
- **q** : get back to the terminal

11) **touch command** : to create new empty file and change the time stamping in existing file

- **touch file_name**
  1) **if not exist** : create the new empty file
  2) **if already exist** : change the time stamp

12) **nano command** : text editor for search, replace,……

13) **sudo command** :

- **sudo passwd** : to set password in the user

14) **top command** : dynamic and the real time view of the system

15) **kill command** : to kill the current process

- **kill -flags pid**

to know about the pid

- **ps -ux**
- **ps -aux**
- **ps -U user_name**
- **ps -C**

16) **echo command** :

- **var_name=" value "** : for creating the variable
- **echo $var_name** : print the variable
- **echo -e** : for escape sequence
- **echo -e \t** : horizontal tab
- **echo -e \n** : new line

17) **file permission** : control the access that users have to files and directories. Each file and directory has three types of permissions for three different categories of users.

chmod [who][+/-][permission] [file_or_directory]

In who:

- **-u (user or owner), -g(group), -o(others), a(all)**

+ for add and – for remove the permission

In permission

- **-r (read), -w(write), -x(execute)**

18) **directory permissions** : it is same as but for the directory

19) **octal and numerical permission** : giving the permission using the number

r	w	x	
1	1	1	(1→ yes)
4	2	1	(0 → no)

4+2+1 = 7 (for rwx permission)

20) **Bash scripting** : Script is the a test file that contains sequence of command. Its is better to give .sh extension for scripting file. Ex: my_script.sh . Inside the scripting file first line should be **#! location of the bash** then interpreter notice that is a scripting file. To find the location of bash use **which bash** inside the other new terminal.

- **./my_script.sh** : to execute the bash script
- **to execute the bash script you must have the execute permission**

Note: after script execute, the working directory return to where the script was initially called

21) **which and whatis command** :

- **which [command/file_name]** : return the path of a file_name or command and it is better to use the full location of a command
- **whatis** : return the short description of command

22) **useradd commands** : Is used to create a new user account on the system. It allows you to specify various options for the new user, such as the home directory, shell, and user ID.

- **sudo useradd name_user flags** :
  in flags
- **-m** : create default home directory
- **-s** : default shell
- **-g** : default user
- **-c** : to add comment in the user

23) **userdel commands**: to delete the user

- **sudo userdel user_name**:
- **sudo userdel -r user_name**: also delete home directory
- **sudo rm -r /home/user_name**

24) **Basic group management**: to create, modify, and delete groups of users.

- **groups** : gives groups added with user

- **cat /etc/group** : all the group in system
- **sudo groupadd group_name** : to add group
- **sudo groupdel group_name** :
- **sudo gpasswd -a user_name group_name** : to assign a user to a particular group
  -a → adding group
  -d → removing group

25) **.bashrc file**: bash script runs whenever a user opens a new interactive non-login shell.

- **nano .bashrc** : to see/edit the contain inside the bashrc file