Linux Commands

Blue - Folder/Directory
White - File
Green - Program or binary
Saffron - Link

Or

drwxr-xr-x - starts with 'd' means **directory**
-rwxr-xr-x - it's a **file**
lrwxrwxrwx - Link to an another file
~ - Represent **Home**
.. - Represents previous directory

Important Directories

1. tmp
2. var
3. root

FileSystem:-

1. **ls** - list storage
    1. Which shows all the folders.
2. ***ls /***
    1. Which shows most beginning of the directories/folders
3. **clear or ctrl + L -** To clear the screen
4. ***ls -l / or ls -l***
    1. **-l** means **long-listing**
    2. Which give complete details of the folders/directories
5. ***ll -*** which gives all directories of current folder
6. ***ls -l /folder-name/folder-name***
    1. Gives us a list of directories—> ***ls -l /Desktop***
7. ***ls . -*** Show current dir files
8. **ls -l .. -** Show previous directories files

Consider now we are in Desktop type —> **ls**
It will all the directories/folders in Desktop folder like **Images Videos Downloads** etc.

***pwd* - Print Working Directory**
To check the current directory I.e where we are navigate to.
/Desktop

***cd file-name* - Change Directory**

Basic File Editing: -
touch fileName.txt

Which creates a new file in the current directory.

cat fileName.tsx
Which returns the content in the textFile. If there is no content return nothing.

nano - Provides a text editor with some commands in the bottom of the console.
Which is a buffer by using it we can also create a file **^o.**

nano fileName.tsx - directly create a file and shows an editor to add any text into the file.
vim fileName.txt - Opens the vim text editor

which nano - Shows where nano binary is installed or not.
which vim - Shows where **vim(which is also a text editor like nano)** is installed or not.

VIM: -

1. Which is mot widely used text editor similar to nano also lite more advances than nano.
2. :**q** - quit.
3. **:w** - write/save the text in the edit.
4. **:w** fileName.txt - To create the file and save the text that written in the editor.

Moving and Renaming Files: -

1. ***cp fileOne.txt newFile.txt***
    1. Copying fileOne text into fileTwo
2. ***diff newFile.txt fileOne.txt***
    1. There is no difference because we copied same text into newFile.txt.
3. **rm**
    1. To remove the file
4. ***mkdir folderName***
    1. To make new directory
5. **mv fileName.txt folderName**
    1. To move files from one folder to another
6. ***mv *.txt folderName***
    1. * represents any file with extension .txt will moved to folder that we mentioned.
7. ***mv fileName.txt ..***
    1. Which moves the file from current director to previous directory
    2. Suppose **Home > Notes,** If cmd given in nodes folder then it moves folder to **Home**(Say parent).
8. ***mv ../fileName.txt .***
    1. Suppose if we are in Notes directory, Now we need to move file

from Home dir without navigating to Home. And ../ represents that file was in previous dir(Parent/Home).
    2. . represents current dir, means moving fileName.txt to current dir(**Notes**).
9. ***mv fileName.txt abc.txt***
    1. To rename the files.
    2. Need to be very **careful in renaming files**. If we rename the file with already existed file it ask us mv: **overwrite fileName.txt** ? Y or N which replaces the files.

The Bash Configuration File

1. ***ls -a***
    1. Which gives hidden files.
    2. Which is very similar to ls -l.
2. ***c*** - To clear screen
3. ***nano -.baschrc***
    1. To edit bashrc file. Which is used to configure using aliases. Example **alias c = clear.** From now

Command Alias

1. ***alias***
    1. Which shows all aliases that we **configure** in .bashrc file.
    2. Where we have few insult aliases as well.
2. ***speedtest***
    1. It test our internet speed.
    2. If we try to alias speedtest it include some python script.
3. ***unalias speedtest***
    1. To delete aliases that we have.
    2. To add again type alias speedtest = ‘’

Understanding Permissions: -

When giving command **ls -l** which gives the directories and folder present in the current directory with complete details like name, date, time etc. Where we can see **drwxrwxr-x** this type of text for each dir/file.

drwxrwxr-x where

1. d -> directory
2. rwx(first) -> User(Owner file 0r directory)
3. rwx(second) -> Group
4. R-x -> aka World

1. r = read
2. w = write
3. x = execute

If anything show “-” that means we didn’t have a permission.

For example, **drw-rw-r-x** here x is replaced with " - " means that there is permission to execute that file/directory.

To change those modes

1. ***chmod +x fileName.txt***
    1. Change mode by adding x to the mentioned file.
2. ***chmod +r fileName.txt***
3. ***chmod +w fileName.txt***
4. ***chmod -r fileName.txt***
    1. ***Removes permission to read file for the user***
5. ***chmod u + x fileName.txt***
    1. To give permission for a specific one like user/group/global where others no access the file
6. ***chmod a + rwx***
    1. Which is open to every one to access(read, write, execute). **a** denotes **all**(global)/ **drwxrwxrwx.**
7. ***chmod g-rws fileName.txt***
    1. Removes only group access. **g** denotes group.
8. ***chmod o+rwx fileName.txt***
    1. **o** denotes others.

Resource Usage: -

1. ***free or free -m***
    1. To check the usage of the system.
2. ***df***
    1. Df denotes disk free, Which is used to check the disk space.
3. ***/bin/df***
    1. Which is same as the df command but the storage values are clear and exact numbers.
4. ***df -I***
    1. To check the **Inodes.** Which return the storage values 2M, 346K etc, which is not in detail.
5. ***/bin/df***
    1. Which is same as the df command but the storage values are clear and exact numbers.
6. ***df -h***
    1. Here, ***h*** refers to **'Human Readable'.**
7. ***htop***
    1. Gives all kind of info about the resources.
8. ***uptime***
    1. Gives performance of cores which shows the load average.

Package Management bases on Debian-based Distribution: -

1. ***sudo apt update***
    1. To update the repository.

2. ***apt search firefox***
3. ***sudo apt install vim-nox***
4. ***vpt remove pkgName***
    1. To delete the particular package.
5. ***sudo apt upgrde***
6. ***sudo apt dist-upgrade***
7. ***sudo reboot***

Managing System Units: -

1. ***systemctl status apache2***
    1. To check the status of apache2.
2. ***sudo systemctl status apache2***
    1. Some of the information was hidden by using sudo that hidden info was shown.
3. ***sudo systemctl disable apache2***
    1. Make disabled but active.
4. ***sudo systemctl stop apache2***
    1. Make inactive

Restart: -

1. ***sudo systemctl restart apache2***
    1. Which is used to restart the unit.

Logs: -

1. ***cat apache2/error.log***
    1. To check error logs
2. ***cat dmessage***
3. ***head /var/log/syslog***
    1. Which give first 10 lines of log file
4. ***tail /var/log/syslog***
    1. Which returns last 10 lines of the log file
5. ***tail -n 50 /var/log/syslog***
    1. ***Which returns 50 lines***
6. ***tail -f /var/log/syslog***
7. ***journalctl -u ssh***
8. ***journalistl -u apache2***
9. ***journalistl -fu apache2***
    1. Those are useful to follow the application logs. Whether is restarted or started.

Managing Users: -

—> Creating users
—> Deleting users
—> Creating groups
—> Add a user to a group

—> Delete a user from a group
—> Deleting groups

1. ***cat etc/passwd***
    1. It returns all the users on the system which are system users.
2. ***cat etc/shadow***
3. ***sudo !!***
    1. !! Is to repeat the most recent command i.e sudo cat etc/sahdow
4. ***cat etc/group***
    1. List of groups like scanner, vboxusers etc..
5. ***groups***
6. ***sudo adduser userName***
    1. Which is used to create new-user.
    2. Creates a group with same userName.
    3. Creates a home directory.
    4. Copying files from etc/skel.
    5. We create a new password.
    6. ***ls*** we got new directory.
7. ***sudo su -userName***
    1. To change the password
8. ***sudo user del -r userName***
    1. To delete the user.
9. ***sudo groupadd groupName***
    1. To create a new group.
10. ***groups***
    1. Displays all groups we have.
11. ***sudo user mod -aG groupName userName***
    1. To add a user to the group.
12. ***gpasswd -d userName***
    1. To delete the userName which is added in the group.
13. ***sudo groupdel heroes***
    1. To delete the group.

Bash History: -

1. To go to history of the bash just by clicking on the up and down arrow keys.
2. ***history***
    1. Shows all the list of commands with **serial number, date and timestamp.**
3. ***!567***
    1. Which is the command number to run that particular command using exclamatory.
    2.