***Basic Linux Commands for Beginners***  
  
  
  
What Is Linux?  
  
Linux is an operating system's kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called “distros”.  
  
Ubuntu Linux  
  
Red Hat Enterprise Linux  
  
Linux Mint  
  
Debian  
  
Fedora

* Linux 𝐃𝐢rectories

/ (𝐑𝐨𝐨𝐭 𝐃𝐢𝐫𝐞𝐜𝐭𝐨𝐫𝐲): At the helm, the / directory is the cornerstone, housing critical files needed to kick start the Linux file system. It takes precedence before any other file system can be mounted, laying the groundwork for system operations.

/𝐛𝐨𝐨𝐭: Strategically positioned, /boot is a vital repository for static kernels, bootloader configurations, and essential executable files. It serves as the launch pad, propelling Linux systems into action during start up.

/𝐛𝐢𝐧: The /bin directory hosts fundamental user-executable files, crucial for everyday commands and operations. It ensures the smooth functionality of essential user interactions with the system.

/𝐝𝐞𝐯: Central to hardware integration, /dev houses device files—distinct from device drivers—serving as virtual gateways to all hardware devices connected to the Linux system.

/𝐞𝐭𝐜: A pivotal repository for local system configuration files, /etc plays a key role in customizing and fine-tuning system settings to meet the unique needs of the host system.

/𝐥𝐢𝐛: Nestled at the core of system startup, /lib is a reservoir of shared library files critical for initializing the Linux system. These shared libraries are the building blocks for a seamless system initiation.

/𝐡𝐨𝐦𝐞: User-centricity shines in the /home directory, offering personalized space for users. Each user has a designated directory within /home to securely store files and data.

/𝐫𝐨𝐨𝐭: Exclusively reserved for the root user, /root serves as the home directory for elevated system access and management, embodying a principle of heightened responsibility.

/𝐦𝐧𝐭: Flexibility takes centre stage in the /mnt directory, a temporary space for mounting basic file systems during system repairs or maintenance. It adapts dynamically to the evolving needs of administrators.

/𝐦𝐞𝐝𝐢𝐚: Bridging physical and digital realms, /media is the rendezvous for external removable media devices like USB thumb drives. It facilitates seamless data exchange between the Linux system and external devices.

* **Package 𝐌𝐚𝐧agement**

1️⃣ 𝐚𝐩𝐭-𝐠𝐞𝐭 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

APT stands for Advance Packaging Tool. Apt-get is one on the package management tool in Linux. Its main task is to retrieve the information and packages from authenticated source for installation and upgrade and removal of package along with their dependencies. It is mainly used in debian based Systems.

✏ Syntax: 𝐚𝐩𝐭-𝐠𝐞𝐭 [𝐨𝐩𝐭𝐢𝐨𝐧𝐬] [𝐜𝐨𝐦𝐦𝐚𝐧𝐝𝐬] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞]

2️⃣ 𝐀𝐩𝐭 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

Apt command is also a package management tool like apt-get command. Apt command provides a higher level interface and is intended as an interface for end user which enables some options better suited for interactive usage by default as compared to other command.

✏ Syntax: 𝐚𝐩𝐭 [𝐜𝐨𝐦𝐦𝐚𝐧𝐝] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞𝐬]

3️⃣ 𝐘𝐮𝐦 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

Yum command is a powerful package management tool used for installing, updating and managing package in red-hat based Linux distribution like centos, fedora.

✏ Syntax: 𝐲𝐮𝐦 [𝐨𝐩𝐭𝐢𝐨𝐧𝐬] [𝐜𝐨𝐦𝐦𝐚𝐧𝐝] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞𝐬]

4️⃣ 𝐝𝐧𝐟 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

dnf is also a package management tool commonly used in red-hat based Linux distribution like fedora and centos. It stands for ‘dandified YUM”.

✏ Syntax: 𝐝𝐧𝐟 [𝐜𝐨𝐦𝐦𝐚𝐧𝐝𝐬] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞𝐬]

5️⃣ 𝐑𝐩𝐦 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

RPM is both a file format and a package management system. It is designed to simplify software installation and maintenance by packaging software applications and their dependencies into a single file, allowing for easy distribution and installation. RPM packages typically have a .rpm file extension.

✏ Syntax: 𝐫𝐩𝐦 [𝐨𝐩𝐭𝐢𝐨𝐧𝐬] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞\_𝐧𝐚𝐦𝐞]

6️⃣ 𝐃𝐩𝐤𝐠 𝐜𝐨𝐦𝐦𝐚𝐧𝐝:

Just like rpm, dpkg is a package management system commonly used in debian based Linux distribution to install and download software packages. Debian files end with .deb extension

✏ Syntax: 𝐝𝐩𝐤𝐠 [𝐨𝐩𝐭𝐢𝐨𝐧𝐬] [𝐩𝐚𝐜𝐤𝐚𝐠𝐞\_𝐧𝐚𝐦𝐞]

* **Basic Commands**  
    
  1. pwd — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd”command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash( / ). The user directory is usually something like "/home/username".  
    
    
  2. ls — Use the "ls" command to know what files are in the directory you are in. You can see all the hidden files by using the command “ls -a”.  
    
    
    
  3. cd — Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in“cd Downloads”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine you have a folder named “Raspberry Pi”. In this case, when you type in “cd Raspberry Pi”, the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use a backward slash. That is, you can use “cd Raspberry\ Pi” in this case. Spaces are denoted like this: If you just type “cd” and press enter, it takes you to the home directory. To go back from a folder to the folder before that, you can type “cd ..” . The two dots represent back.  
    
    
    
  4. mkdir & rmdir — Use the mkdircommand when you need to create a folder or a directory. For example, if you want to make a directory called “DIY”, then you can type “mkdir DIY”. Remember, as told before, if you want to create a directory named “DIY Hacking”, then you can type “mkdir DIY\ Hacking”. Use rmdir to delete a directory. But rmdir can only be used to delete an empty directory. To delete a directory containing files, use rm.  
    
    
    
  5. rm - Use the rm command to delete files and directories.  Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.  
    
    
    
  6. touch — The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, “touch new.txt”.  
    
    
  7. man & --help — To know more about a command and how to use it, use the man command. It shows the manual pages of the command. For example, “man cd” shows the manual pages of the cd command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., cd –help).  
    
    
  8. cp — Use the cp command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.  
    
    
  9. mv — Use the mv command to move files through the command line. We can also use the mv command to rename a file. For example, if we want to rename the file “text” to “new”, we can use “mv text new”. It takes the two arguments, just like the cp command.  
    
    
  10. locate — The locate command is used to locate a file in a Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file. Using the -i argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word “hello”, it gives the list of all the files in your Linux system containing the word "hello" when you type in “locate -i hello”. If you remember two words, you can separate them using an asterisk (\*). For example, to locate a file containing the words "hello" and "this", you can use the command “locate -i \*hello\*this”.  
    
    
    
  **Intermediate Commands**  
  1. echo — The "echo" command helps us move some data, usually text into a file. For example, if you want to create a new text file or add to an already made text file, you just need to type in, “echo hello, my name is alok >> new.txt”. You do not need to separate the spaces by using the backward slash here, because we put in two triangular brackets when we finish what we need to write.  
    
  2. cat — Use the cat command to display the contents of a file. It is usually used to easily view programs.  
    
  ￼  
    
  3. nano, vi, jed — nano and vi are already installed text editors in the Linux command line. The nanocommand is a good text editor that denotes keywords with color and can recognize most languages. And vi is simpler than nano. You can create a new file or modify a file using this editor. For example, if you need to make a new file named "check.txt", you can create it by using the command “nano check.txt”. You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no). In my experience, using nano for HTML editing doesn't seem as good, because of its color, so I recommend jed text editor. We will come to installing packages soon.  
    
  ￼  
    
  4. sudo — A widely used command in the Linux command line, sudo stands for "SuperUser Do". So, if you want any command to be done with administrative or root privileges, you can use the sudo command. For example, if you want to edit a file like viz. alsa-base.conf, which needs root permissions, you can use the command – sudo nano alsa-base.conf. You can enter the root command line using the command “sudo bash”, then type in your user password. You can also use the command “su” to do this, but you need to set a root password before that. For that, you can use the command “sudo passwd”(not misspelled, it is passwd). Then type in the new root password.  
    
    
    
  5. df — Use the df command to see the available disk space in each of the partitions in your system. You can just type in df in the command line and you can see each mounted partition and their used/available space in % and in KBs. If you want it shown in megabytes, you can use the command “df -m”.  
    
    
    
  6. du — Use du to know the disk usage of a file in your system. If you want to know the disk usage for a particular folder or file in Linux, you can type in the command df and the name of the folder or file. For example, if you want to know the disk space used by the documents folder in Linux, you can use the command “du Documents”. You can also use the command “ls -lah” to view the file sizes of all the files in a folder.  
    
  ￼  
    
  7. tar — Use tar to work with tarballs (or files compressed in a tarball archive) in the Linux command line. It has a long list of uses. It can be used to compress and uncompress different types of tar archives like .tar, .tar.gz, .tar.bz2,etc. It works on the basis of the arguments given to it. For example, "tar -cvf" for creating a .tar archive, -xvf to untar a tar archive, -tvf to list the contents of the archive, etc. Since it is a wide topic, here are some examples of tar commands.  
    
  8. zip, unzip — Use zip to compress files into a zip archive, and unzip to extract files from a zip archive.  
    
  9. uname — Use uname to show the information about the system your Linux distro is running. Using the command “uname -a” prints most of the information about the system. This prints the kernel release date, version, processor type, etc.  
    
  ￼  
    
  10. apt-get — Use apt to work with packages in the Linux command line. Use apt-get to install packages. This requires root privileges, so use the sudocommand with it. For example, if you want to install the text editor jed (as I mentioned earlier), we can type in the command “sudo apt-get install jed”. Similarly, any packages can be installed like this. It is good to update your repository each time you try to install a new package. You can do that by typing “sudo apt-get update”. You can upgrade the system by typing “sudo apt-get upgrade”. We can also upgrade the distro by typing “sudo apt-get dist-upgrade”. The command “apt-cache search” is used to search for a package. If you want to search for one, you can type in “apt-cache search jed”(this doesn't require root).  
    
  ￼  
    
  11. chmod — Use chmod to make a file executable and to change the permissions granted to it in Linux. Imagine you have a python code namednumbers.py in your computer. You'll need to run “python [numbers.py](http://numbers.py/)” every time you need to run it. Instead of that, when you make it executable, you'll just need to run “[numbers.py](http://numbers.py/)” in the terminal to run the file. To make a file executable, you can use the command “chmod +x [numbers.py](http://numbers.py/)” in this case. You can use “chmod 755 [numbers.py](http://numbers.py/)” to give it root permissions or “sudo chmod +x [numbers.py](http://numbers.py/)” for root executable. Here is some moreinformation about the chmod command.  
    
  ￼  
    
  12. hostname — Use hostname to know your name in your host or network. Basically, it displays your hostname and IP address. Just typing “hostname” gives the output. Typing in “hostname -I” gives you your IP address in your network.  
    
  ￼  
    
  13. ping — Use ping to check your connection to a server. Wikipedia says, "Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network". Simply, when you type in, for example, “ping [google.com](http://google.com/)”, it checks if it can connect to the server and come back. It measures this round-trip time and gives you the details about it. The use of this command for simple users like us is to check your internet connection. If it pings the Google server (in this case), you can confirm that your internet connection is active!  
    
  ￼
* **RESTART SERVICES**

Enabling, disabling, and restarting services in Linux can be done using the systemctl command. Here are the steps for each:

* To enable a service: **sudo systemctl enable service\_name**
* To disable a service: **sudo systemctl disable service\_name**
* To start a service: **sudo systemctl start service\_name**
* To stop a service: su**do systemctl stop service\_name**
* To restart a service: **sudo systemctl restart service\_name**
* **Tips and Tricks for Using Linux Command Line**  
    
  You can use the clear command to clear the terminal if it gets filled up with too many commands.  
    
  TAB can be used to fill up in terminal. For example, You just need to type “cd Doc” and then TAB and the terminal fills the rest up and makes it “cd Documents”.  
    
  Ctrl+C can be used to stop any command in terminal safely. If it doesn't stop with that, then Ctrl+Z can be used to force stop it.  
    
  You can exit from the terminal by using the exit command.  
    
  You can power off or reboot the computer by using the command sudo halt and sudo reboot.
* **TOP 50 Linux commands**

1. SSH Key Authentication:

ssh-keygen -t rsa

ssh-copy-id user@hostname

2. File Transfer:

scp localfile.txt user@remote:/path

3. Text Search:

grep -r "pattern" /path/to/search

4. Process Management:

ps aux | grep process\_name

kill -9 process\_id

5. System Information:

uname -a

cat /etc/os-release

6. Disk Usage:

df -h

apt-get update

apt-get install package\_name

7. Network:

ifconfig

netstat -tulpn

8. User add:

useradd username

passwd username

9. L permission:

chmod +x filename

chown user:group filename

11. Cron Jobs:

crontab -e

12. System Logs:

tail -f /var/log/syslog

13. SSH Tunneling:

ssh -L local\_port:remote\_host:remote\_port

user@hostname

14. Firewall Configuration:

ufw allow 80

15. Check Service Status:

systemctl status service\_name

16. Create a RAM Disk:

mount -t tmpfs -o size=512M tmpfs

/mnt/ramdisk

17. Environment Variables:

export VARIABLE=value

18. Disk Encryption:

cryptsetup luksFormat /dev/sdX

19. Docker Commands:

docker ps

docker exec -it container\_id /bin/bash

20. Check System Load:

Top

21. System Upgrades:

apt-get upgrade

22. Run a Command in the Background:

command &

23. List Open Ports:

lsof -i

24. Find and Replace in Files:

sed -i 's/old\_text/new\_text/g' filename

25. Check Available Memory:

free -m

26. Monitor Network Traffic:

tcpdump -i eth0

27. Install Nginx:

apt-get install nginx

28. SSH Configurations:

nano ~/.ssh/config

29. Generate Random Password:

openssl rand -base64 12

30. Archive and Compress:

tar -czvf archive.tar.gz /path/to/directory

31. Check System Uptime:

Uptime

32. Run a Command on Multiple Servers:

parallel-ssh -h hosts.txt -l username -i

"command"

33. Monitor Disk I/O:

iostat -d 5

34. Check Kernel Version:

uname -r

35. Find Large Files:

find / -type f -size +100M

36. Install Node.js:

curl -sL

<https://lnkd.in/d3JaFZ_u> | bash

-

apt-get install -y nodejs

37. Check File System Type:

df -Th

38. Run a Command at Regular Intervals:

watch -n 1 command

39. Limit CPU Usage:

cpulimit -e process\_name -l 50

40. Install Git:

apt-get install git

41. Check System Architecture:

Arch

42. List Installed Packages:

dpkg –list

43. Create Symbolic Link:

ln -s /path/to/source /path/to/link

44. List USB Devices:

Lsusb

45. List Open Files by User:

lsof -u username

46. Check SELinux Status:

Sestatus

47. Install Python Pip:

apt-get install python3-pip

48. Check RAID Status:

cat /proc/mdstat

49. Check OpenVPN Status:

systemctl status openvpn

50. Check Failed Login Attempts:

cat /var/log/auth.log | grep "Failed password