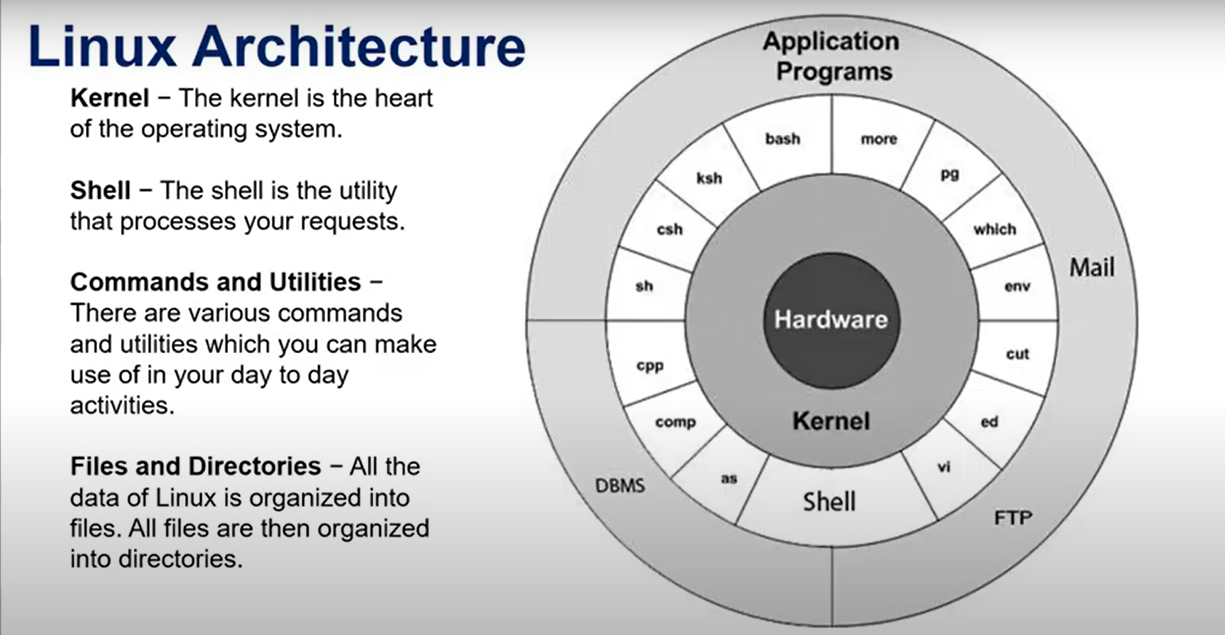
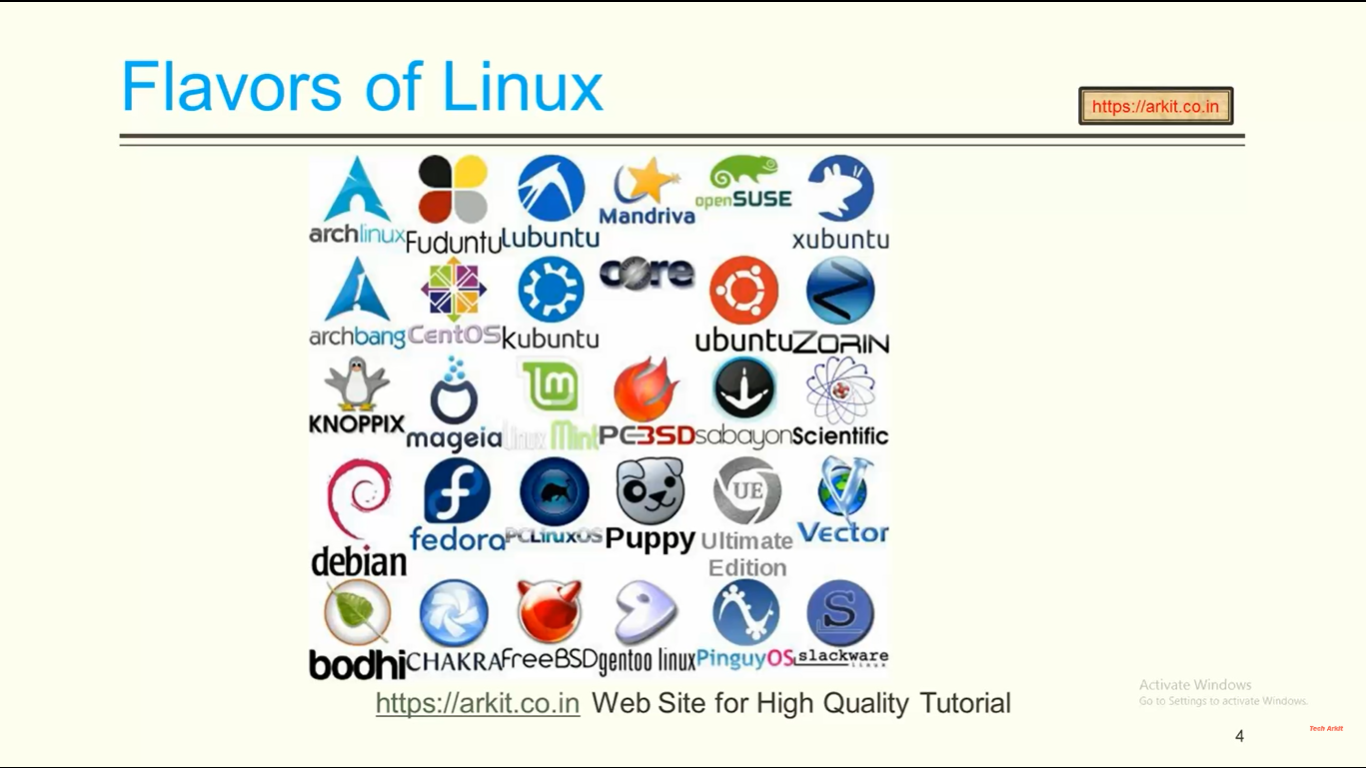
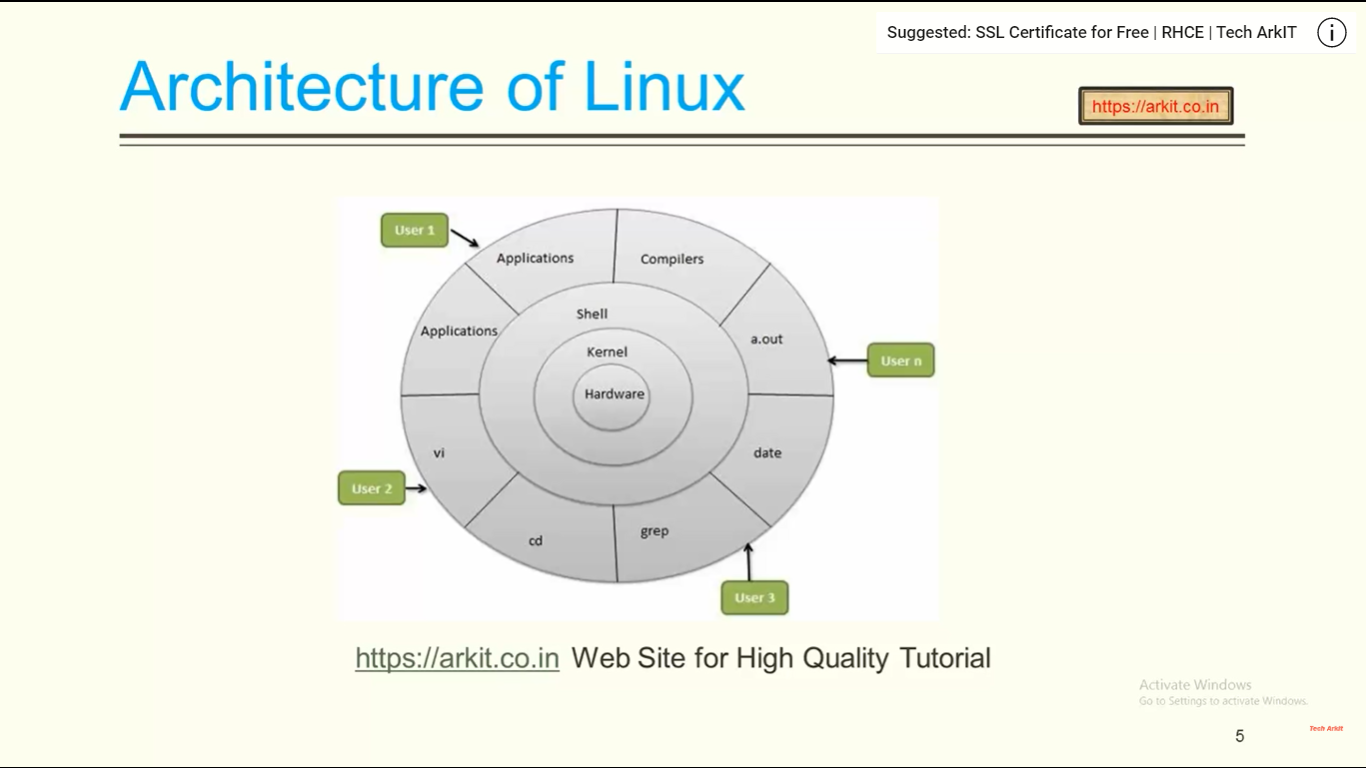
**What is Linux:**

Linux is one of the OS that is a set of programs acts as a link b/w the user and the computer.

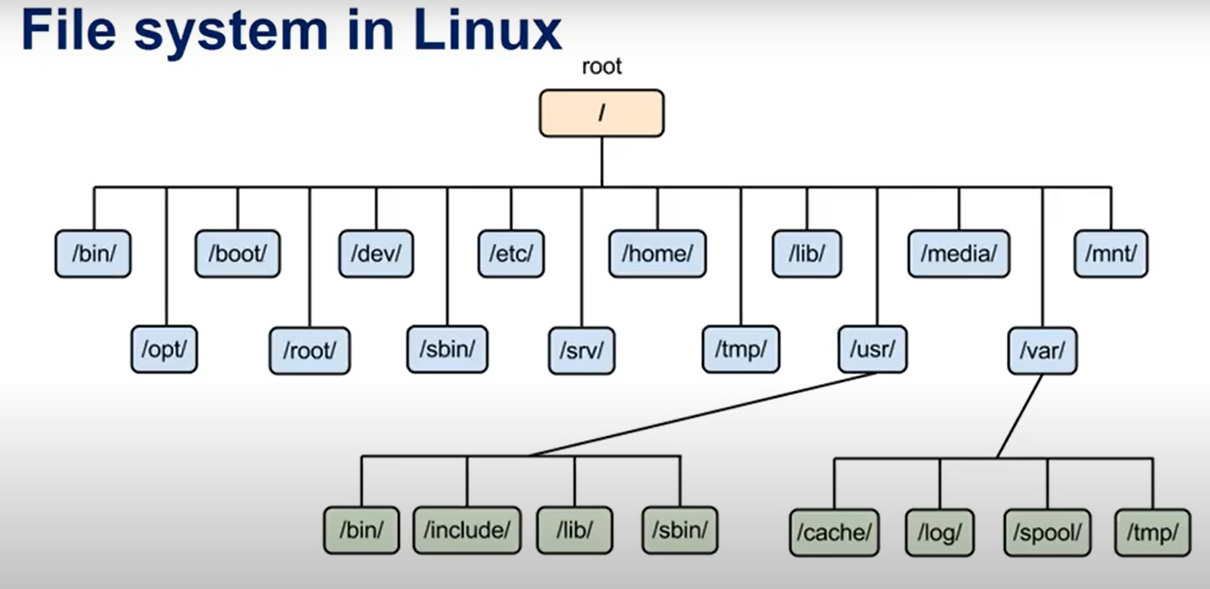






**~** ====================Home directory

**/** ==================== root



**How to see filesystem/directories: in linux?**

**ls /**

/ ---root (Top level directory)

/bin—all executable files

/root ---root directory for root users

/boot—boot configuarion files

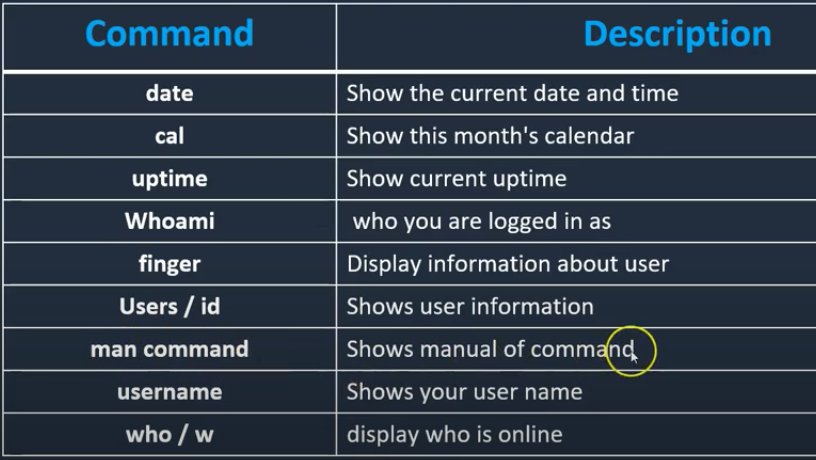
/dev---It has the info about the devicce which are attached.

/home—home directory for other users

/usr ---by default softwares are installed on this

/sbin—it contains commands used by only super user

/var---it contains variable data

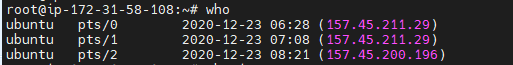


**Man** :: help provider command

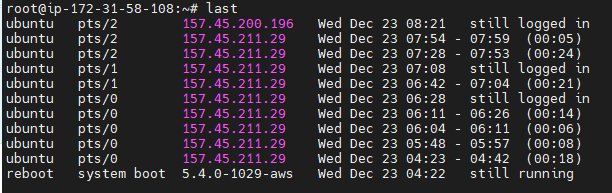
**pwd** - print name of current/working directory.

**whoami** - print effective userid

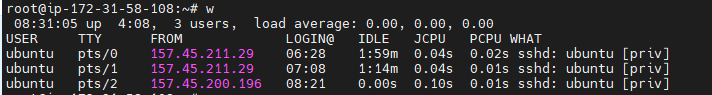
**cal** - display a calendar

**who** -  displays a list of users who are currently logged into the computer

**last**  - show a list of last logged in users



**w**---- - Show who is logged on and what they are doing.



----------------------------------------------------------------

**touch** ------to create empty files and also using which we can create duplicate files(it creates duplicate files based on the time)

**touch file1 file2 file3 ….**

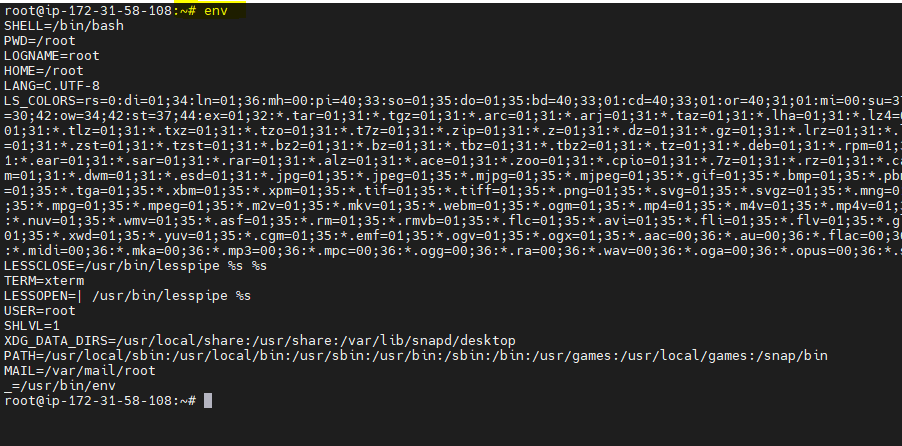
**touch file{1..10}.txt**

**touch –c file1** --------------It will not create any file

-c, --no-create do not create any files



**env**-----to see all environmental variables.



**How to mount the file systems temporarily or permanently?**

# mkdir /mnt/oracle

# mount /dev/sdc1 /mnt/oracle (temporary mount)

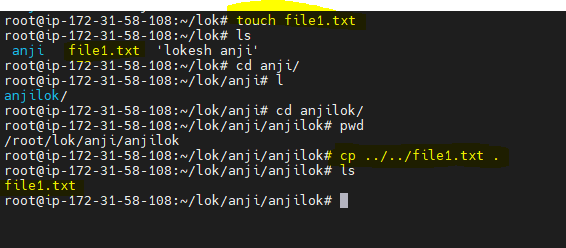
# vim /etc/fstab

/dev/sdc1 /mnt/oracle xfs defaults 0 0

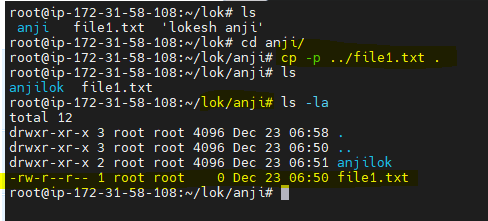
Esc+:+wq!

# mount -a (permanent mount)

**Cp --------- copy files and directories**



**-p ---to preserve the time stamp**



**-R ----to copy directory**

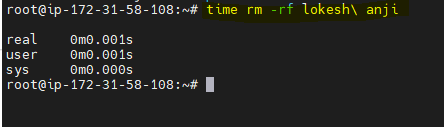
**cd - ================to go to previous directory**

**rm –d directoryname**

**rm –rf directoryname**

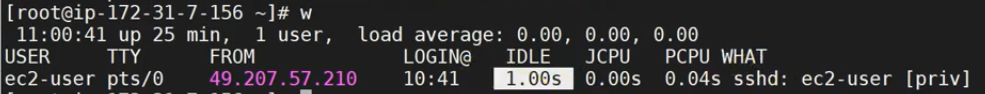
**rmdir ---to remove empty directory**

**time ---------time estimation to do work**





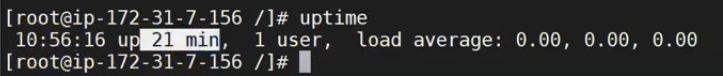
**w—**



**date** ---display date



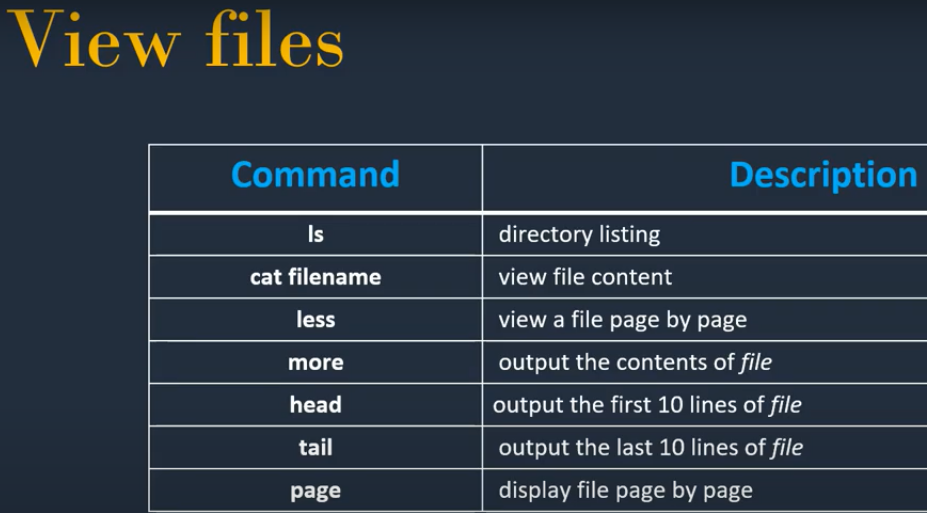
**Uptime:** shows the uptime



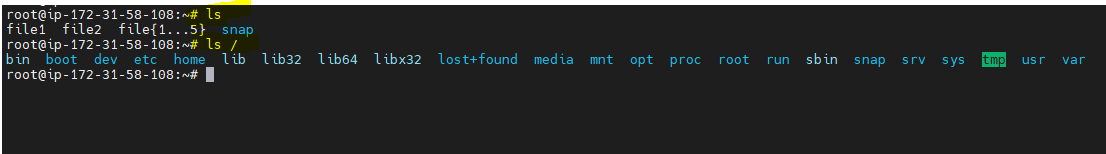
**Finger & Users—**displays info about the user

**Id –**shows the id





**ls –list files and directories**

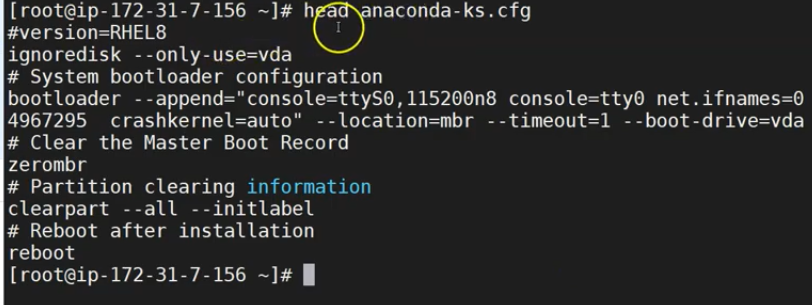


**less =--**shows the contents of a file page by page

**more =**When we have files with n no of lines this command is useful to see.

Cat file1 | more

**head ===**shows first 10 lines



**tail ===**shows last 10 lines

to see first or last 20 lines then

**head -20 filename**

**tail -20 filename**

**Sort ---** To sort the content in alphabetical order**. (sort file1)**

Sort –k1 file1

Sort –nk1 file1

Sort –rk1 file1

**df** - report file system disk space usage

Filesystem Size Used Avail Use% Mounted on

devtmpfs 474M 0 474M 0% /dev

tmpfs 492M 0 492M 0% /dev/shm

tmpfs 492M 400K 492M 1% /run

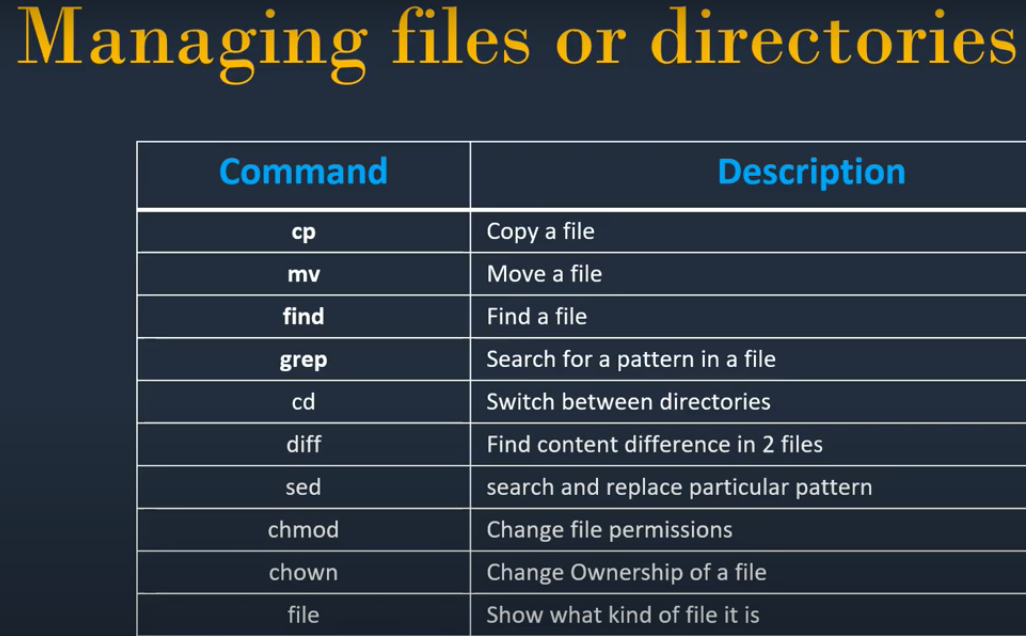
tmpfs 492M 0 492M 0% /sys/fs/cgroup

/dev/xvda1 8.0G 1.4G 6.7G 17% /

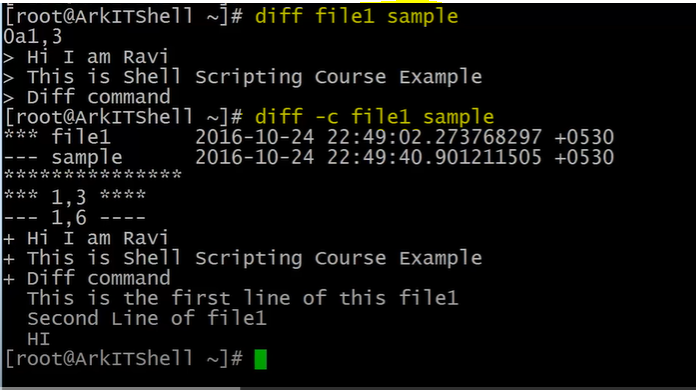
tmpfs 99M 0 99M 0% /run/user/1000

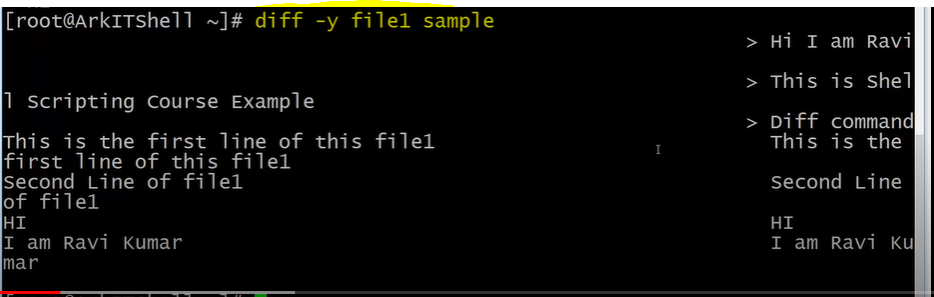
**du** - estimate file space usage





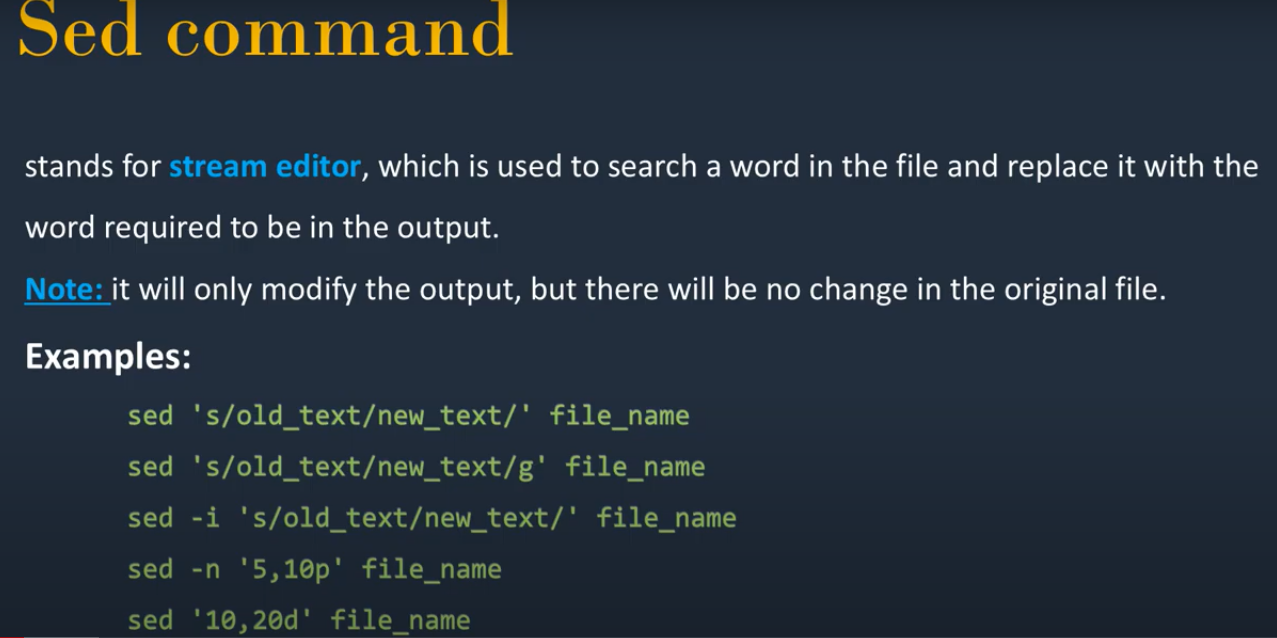
**Diff –To find the diff in two or more files**

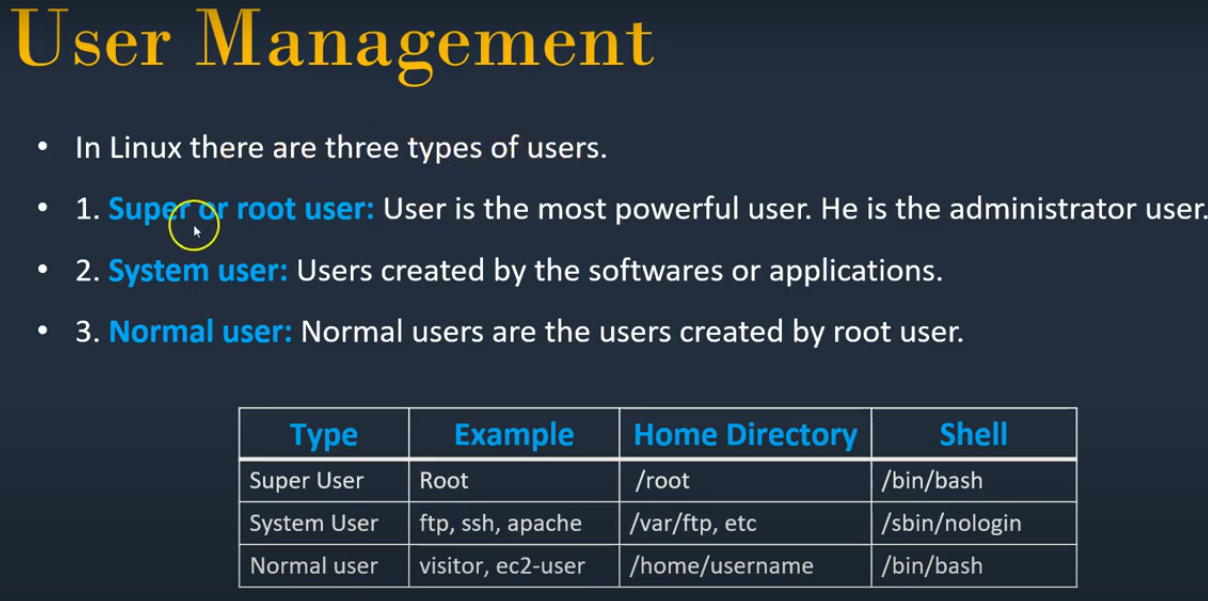




**Grep –to search for a contents**







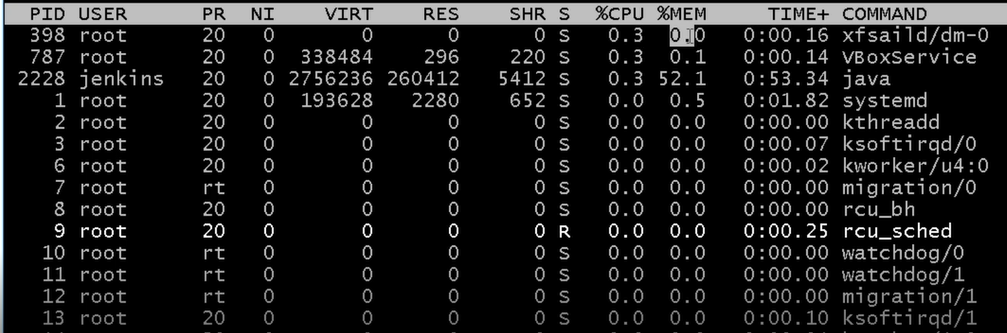
**rm –rf \***

**Pipe operator ( | ):: Used to give one command output as input to the other**

**netstat –na ====to list the ports**

**tracert ----used to track in real-time the pathway taken by a packet on an IP network from source to destination.**

**top ----to find cpu,memory usage (process id ,user, etc)**



**which bash------**

**wc –word count**

**wc –l line numbers**

**wc –c character numbers**

**wc –w number of words**

**Display a number b/w 40 – 50 =====head -50 | tail -10**

**grep –I =====for ignoring the case**

**grep –w ===search for a word**

**grep –n ===print line along with the search**

**tail –f ========shows the log file dynamically**

**grep –An ABC (grep –A2 ABC) ===print n no of lines after the occurrence**

**grep –Bn ABC ======print n no of lines before the occurrence**

**vi ,vim & nano ====vim is the improved version of vi ,,,nano is the recent its for the beginners to start**

**:w ---save**

**:q----quit without saving**

**:wq ---save & quit**

**:yy ---copy**

**nyy ----copy n lines ( 3yy ----copy 3 lines)**

**p ----paste**

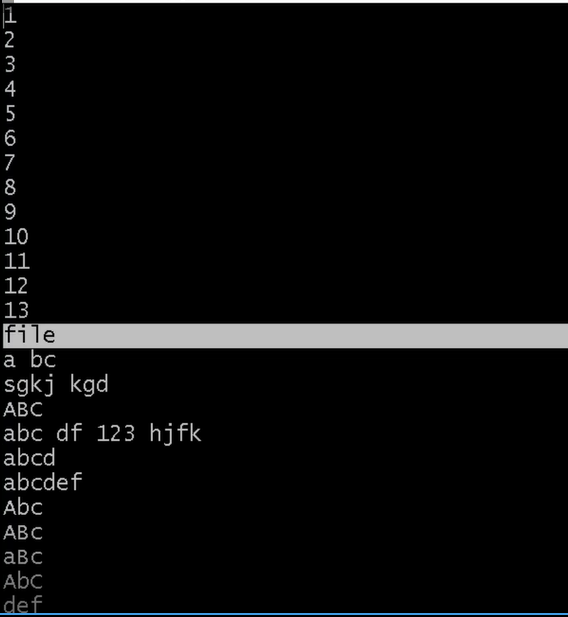
**x ----to delete single character**

**dd ----cut (ndd -----cut n lines)**

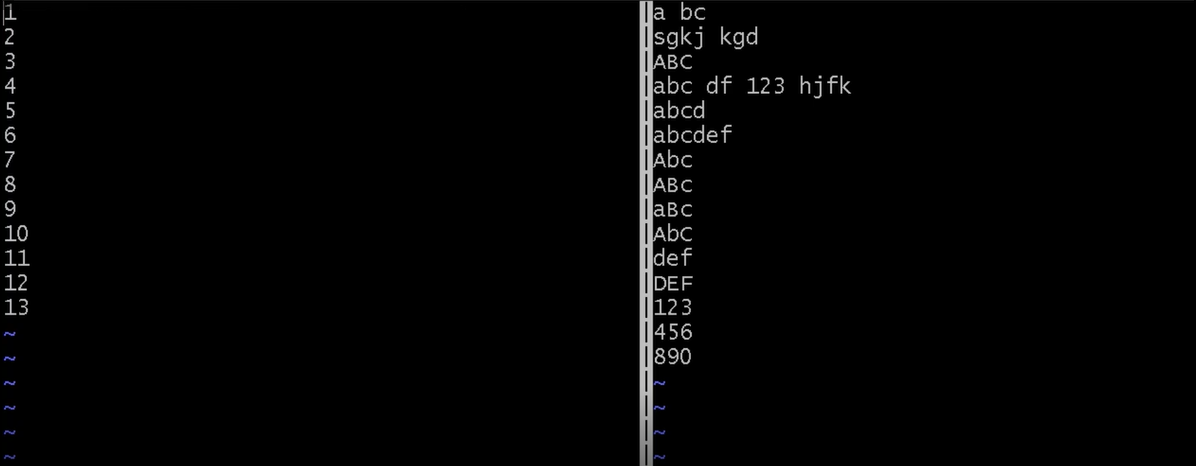
**/word ---to search for a particular word**

**Multiple files:**

**vim –o file1 file2 (opens the files horizontally) ctrl+w -----to goto second file**



**vim –O file1 file2 (Opens files vertically)**

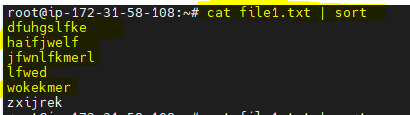


**wget =====to download the package to linux machine.**

**winscp ============linux to windows and viceversa**

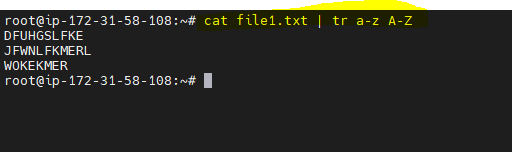
**iputils---**

**Sort**:: To sort in alphabetical order.



------------------------------------------------------------

**tr ====upper case to lower and vice-versa**



**free - Display amount of free and used memory in the system**

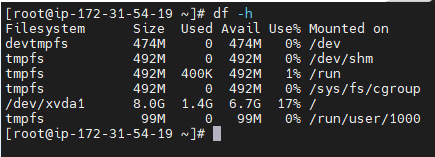
free –b ,,,free –m

free –s 2

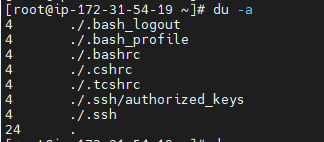
free –c 4

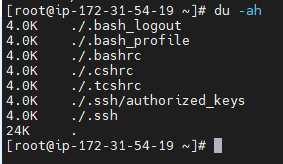
**df - report file system disk space usage**

**df –a –h**

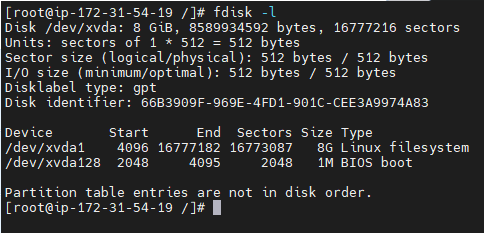


**du - estimate file space usage**





**fdisk - manipulate disk partition table**



**Parted –**To find free space after we created partition .( parted - a partition manipulation program)

**HELP commands::**

1. **Man --------** man command in Linux is used to display the user manual of any command

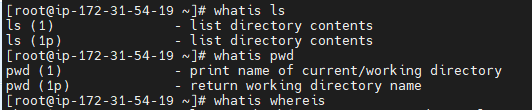
**Ex: man ls**

**man pwd**

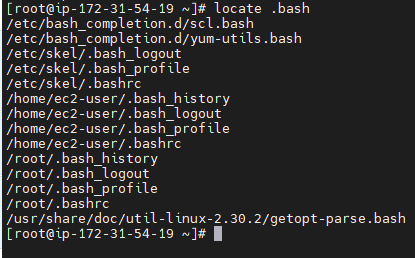
1. **whereis locates the binary, source and manual files for the specified commands located at**



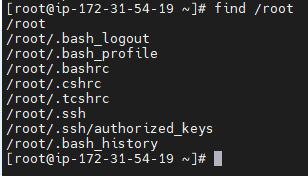
1. **whatis searches the manual page names and displays the manual page descriptions**



**locate – -** find files by name (update db) Maintain own database.

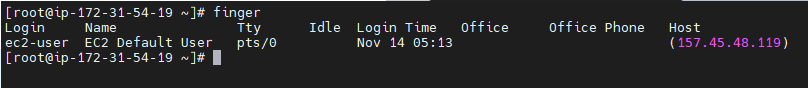


**find—**updated one (search for files in a directory hierarchy)



**last, lastb** - show listing of last logged in users





**uname -** print system information

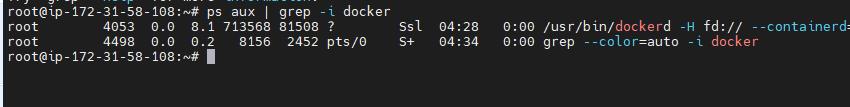
**top—Dynamic real time view of a running system.(process), uptime, how many users logged in**

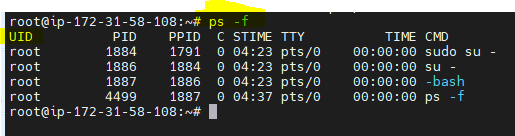
**top - display Linux processes(Detailed)**

**--------------------------------------------------------------------------------------------**

**ps - report process status or** information related with the processes

ps aux | grep –I docker





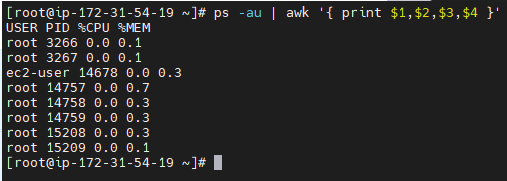
**grep - search a file for a pattern**

**sed -- search a file for a pattern and replace**

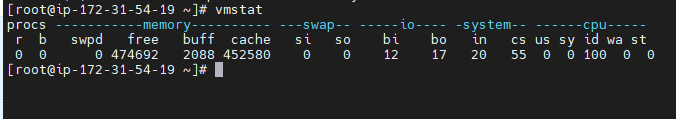
sed ‘s/Lokesh/Dharani/2’ filename

**awk---**used to print rows, colomns etc

(Awk is a scripting language used for manipulating data and generating reports,,based on the column,rows)



**vmstat—reports data about the machine (- Report virtual memory statistics)**



**wc –**it counts lines ,characters ,words in file.(**wc file1)**

**netstat -** Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

**Cut**—To cut the character or byte

Echo ‘Hello’ | cut –c 1

**scp** — secure copy (remote file copy program)

scp filename 192.168.2.34:/tmp

**chmod ::** change permissions of a file

In Unix-like operating systems, the chmod command is used to change the access mode of a file.  
The name is an abbreviation of change mode.

**Syntax :**

chmod [reference][operator][mode] file...

Reference Class Description

u owner file's owner

g group users who are members of

the file's group

o others users who are neither the

file's owner nor members of

the file's group

a all All three of the above, same as ugo

The operator is used to specify how the modes of a file should be adjusted. The following operators are accepted:

Operator Description

+ Adds the specified modes to the

specified classes

- Removes the specified modes from

the specified classes

= The modes specified are to be made

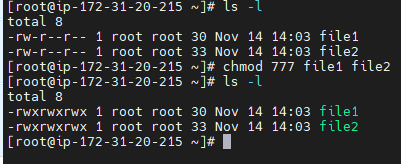
the exact modes for the specified

classes

**d U G O U G**

**d r-x r-x r-x 18 root root 257 Nov 14 13:46 ..(Directory)**

**- rw- --- --- 1 root root 200 Nov 14 14:16 .bash\_history (file)**



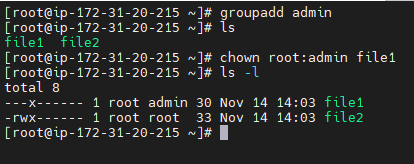
**Read=4 write==2 execute=1**

chmod u+rw file1 file1

chmod o-rwx file1 file2

chmod u-rw file1 file1

**Chown::** change owner and group of a file, ownership of a file

Different users in the operating system have ownership and permission to ensure that the files are secure and put restrictions on who can modify the contents of the files.

**chgrp command** in Linux it is used to change the group ownership of a file or directory. All files in Linux belong to an owner and a group. You can set the owner by using “[chown](https://www.geeksforgeeks.org/chown-command-in-linux-with-examples/)” command, and the group by the “chgrp” command.

**groupadd**command is used to create a new user group.

**useradd** is a command in Linux that is used to add user accounts to your system

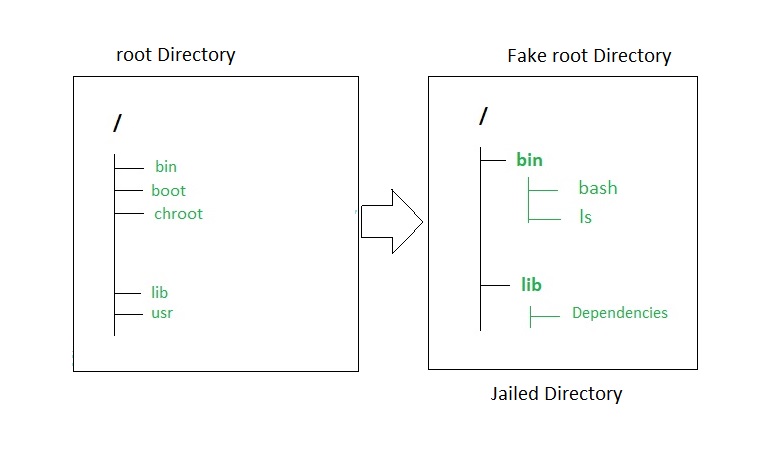
 It make changes to the following files:

* /etc/passwd
* /etc/shadow
* /etc/group
* /etc/gshadow
* creates a directory for new user in /home

# chroot command in Linux with examples

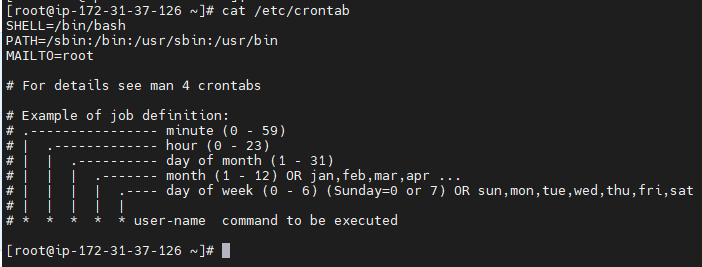
*chroot* command in Linux/Unix system is used to change the root directory. Every process/command in Linux/Unix like systems has a current working directory called root directory. It changes the root directory for currently running processes as well as its child processes.

A process/command that runs in such a modified environment cannot access files outside the root directory. This modified environment is known as “[**chroot jail**](https://www.geeksforgeeks.org/linux-virtualization-using-chroot-jail/)” or **“jailed directory”**. Some root user and privileged process are allowed to use chroot command.

[](https://media.geeksforgeeks.org/wp-content/uploads/chroot-command.jpg)

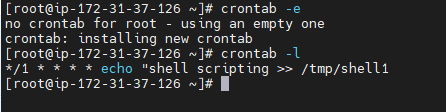
**userdel** command in Linux system is used to delete a user account and related files

**Crontab :: (M H DM M DW)**



**Crontab –e**

**Crontab –l**



**$PATH –variable( Executables)**



**HOSTAME=$(hostname)**

**DATE= $(date)**

**top -b -n 1 -d1 | grep "Cpu(s)" | awk '{print $2}' | awk -F. '{ print $1} ' ----CPU**

**free | grep Mem | awk '{ print $3/$2 \* 100.0 }' ----Mem**

**df -h | awk '{print $5 }' | sed 's/%//g'**

# Special Variables

|  |  |
| --- | --- |
| **Sr.No.** | **Variable & Description** |
| 1 | **$0**  The filename of the current script. |
| 2 | **$n**  These variables correspond to the arguments with which a script was invoked. Here **n** is a positive decimal number corresponding to the position of an argument (the first argument is $1, the second argument is $2, and so on). |
| 3 | **$#**  The number of arguments supplied to a script. |
| 4 | **$\***  All the arguments are double quoted. If a script receives two arguments, $\* is equivalent to $1 $2. |
| 5 | **$@**  All the arguments are individually double quoted. If a script receives two arguments, $@ is equivalent to $1 $2. |
| 6 | **$?**  The exit status of the last command executed. |
| 7 | **$$**  The process number of the current shell. For shell scripts, this is the process ID under which they are executing. |
| 8 | **$!**  The process number of the last background command. |

**Kill:::**

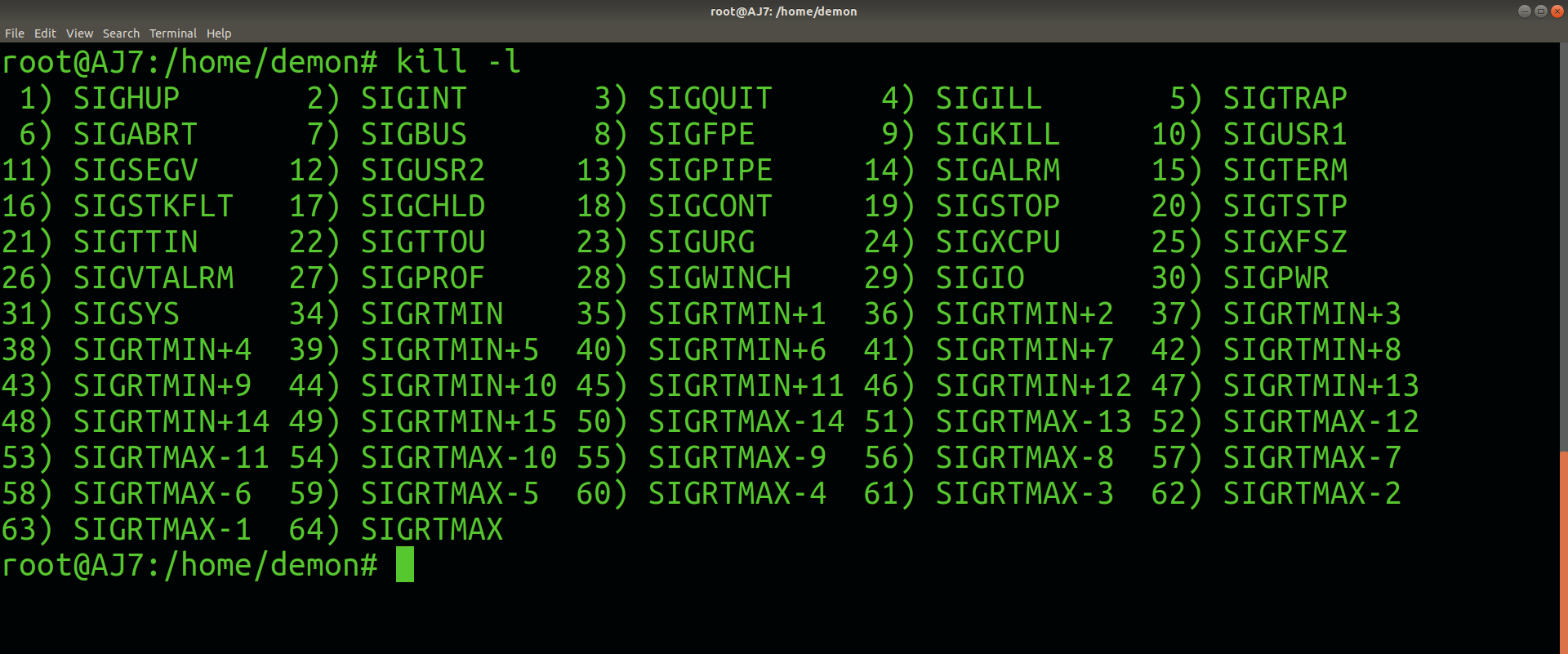
*kill* command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually. *kill* command sends a signal to a process which terminates the process. If the user doesn’t specify any signal which is to be sent along with kill command then default *TERM*signal is sent that terminates the process.

#### Options and examples

**1. kill -l :**To display all the available signals you can use below command option:

**Syntax:**

$kill -l



* **By number (e.g. -5)**
* **With SIG prefix (e.g. -SIGkill)**
* **Without SIG prefix (e.g. -kill)**

**2. kill pid :** To show how to use a PID with the kill command.

**Syntax:**

$kill pid

**Unary operators:**

**1)File 2)Test**

**-b**

**-c**

**-d =====whether it is a directory or file**

**-e=====file exist or not**

**-f ==regular or not**

**-----------------------------------------------**

#!/bin/bash

echo -e "Enter first value: "

read –r A

echo -e "Enter secod value: "

read –r B

echo "additon is `expr $A + $B` "

echo "substraction is `expr $A - $B` "

echo "multiplication is `expr $A \\* $B` "

**Relational operator:**

-lt ====less than <

-le==less than or equal to <=

-ge==greater than equal >=

-gt==greater than >

-eq==equal to =

-ne==not equal !=

**Logical operators:**

And (-a) &&

OR (-o) ||

Not (-n) !

**If –else-if statement:**

If [ ] ; then

Echo “ “

Elif [ ] ; then

Echo “ “

….

….

Else

…

Fi

**Nested if:**

If [ condition ]; then

If [ condition ]; then

………….

……

Else

…

Fi

**While loop:**

While [ condition ]

Do

………

………

done

**Case statement:**

Case value in

1. …
2. …
3. ….

\*) ….

Esac

**Continue statement:**

---------------------------

Set command : We can set