

**Ex No: 1b)**

**Date:** 23/1/2025

## BASIC LINUX COMMANDS

### 1.1 GENERAL PURPOSE COMMANDS

#### 1. The 'date' command:

The date command displays the current date with day of week, month, day, time (24 hours clock) and the year.

**SYNTAX:** \$ date

The date command can also be used with following format.

Format	Purpose	Example
+ %m	To display only month	\$ date + %m
+ %h	To display month name	\$ date + %h
+ %d	To display day of month	\$ date + %d
+ %y	To display last two digits of the year	\$ date + %y
+ %H	To display Hours	\$ date + %H
+ %M	To display Minutes	\$ date + %M
+ %S	To display Seconds	\$ date + %S

#### 2. The 'echo' command:

The echo command is used to print the message on the screen.

**SYNTAX:** \$ echo

**EXAMPLE:** \$ echo "God is Great"

#### 3. The 'cal' command:

The cal command displays the specified month or year calendar.

**SYNTAX:** \$ cal [month] [year]

**EXAMPLE:** \$ cal Jan 2012

#### 4. The 'bc' command:

⑤ \$ who

root pts/0 2025-01-23 8:14 (:0)

CSE368 pts/1 2015-01-23 8:23 (172.16.9.18)

CSE376 pts/0 2025-01-23 8:29 (172.16.9.22)

⑥ \$ who am I

CSE376 pts/1 2025-01-23 8:29 (172.16.9.22)

⑦ \$ id

uid=1377(CSE376) gid=1377(CSE376) group=1377(CSE376)

content = unconfined - u:unconfined or:unconfined - t so so:0 . class

⑧ \$ tty

1:den/pts/1u

⑨ \$ class

⑩ \$ man cat

Name:

CAT - concatenate focus and print on standard output

SYNOPSIS

CAT [option].... [FPU] ...

DESCRIPTION

- A, --show-cell : equivalent to -v  
- b, --number-non-blank : number non empty o/p lines, overrides -n

Unix

SYNTAX: \$

EXAMPLE:

164

5/2

5. The 'who'

The v

logged into th

SYNTAX: \$

6. The 'who a

The v

SYNTAX: \$

7. The 'id' co

The i

SYNTAX: \$

8. The 'tty' c

The t

SYNTAX: \$

9. The 'clear'

The

SYNTAX: \$

10. The 'ma

The

SYNTAX: \$

11. The 'ps'

The

command,

produces a

SYNTAX:

EXAMPLE:

\$ ps -e

\$ ps -aux

Unix offers an online calculator and can be invoked by the command bc.

**SYNTAX:** \$ bc

**EXAMPLE:** bc -l

16/4

5/2

#### 5. The 'who' command

The who command is used to display the data about all the users who are currently logged into the system.

**SYNTAX:** \$ who

#### 6. The 'who am i' command

The who am i command displays data about login details of the user.

**SYNTAX:** \$ who am i

#### 7. The 'id' command

The id command displays the numerical value corresponding to your login.

**SYNTAX:** \$ id

#### 8. The 'tty' command

The tty (teletype) command is used to know the terminal name that we are using.

**SYNTAX:** \$ tty

#### 9. The 'clear' command

The clear command is used to clear the screen of your terminal.

**SYNTAX:** \$ clear

#### 10. The 'man' command

The man command gives you complete access to the Unix commands.

**SYNTAX:** \$ man [command]

#### 11. The 'ps' command

The ps command is used to the process currently alive in the machine with the 'ps' (process status) command, which displays information about process that are alive when you run the command. 'ps;' produces a snapshot of machine activity.

**SYNTAX:** \$ ps

**EXAMPLE:** \$ ps

\$ ps -e

\$ps -aux

⑪ \$ ps

PID	TTY	Time	CMD
3085	pts/14	00:00:00	bash
3933	pts/14	00:00:00	ps

⑫ \$ uname - M

i1686

\$ uname - n

local host . local domain

\$ uname - sr

U.11.8-300-SL26:i1686+PAC

\$ uname - s

Linux

\$ uname - v

# i SMP Thursday Jan 29 20:38:21 UTC 2017

\$ uname - a

Linux localhost.localdomain 4.11.8-300-SL26:i1686+PAC

SMP Thursday Jan 29 20:38:21 UTC 2017 i1686 i1686

i386 GNU/Linux

12. The 'uname' command  
The uname command is used to display relevant details about the operating system on the standard output.

-m -> Displays the machine id (i.e., name of the system hardware)

-n -> Displays the name of the network node. (host name)

-r -> Displays the release number of the operating system

-s -> Displays the name of the operating system (i.e., system name)

-v -> Displays the version of the operating system.

-a -> Displays the details of all the above five options.

SYNTAX: \$ uname [option]

EXAMPLE: \$ uname -a

## 1.2 DIRECTORY COMMANDS

1. The 'pwd' command:

The pwd (print working directory) command displays the current working directory.

SYNTAX: \$ pwd

2. The 'mkdir' command:

The mkdir is used to create an empty directory in a disk.

SYNTAX: \$ mkdir dirname

EXAMPLE: \$ mkdir receive

3. The 'rmdir' command:

The rmdir is used to remove a directory from the disk. Before removing a directory, the directory must be empty (no files and directories).

SYNTAX: \$ rmdir dirname

EXAMPLE: \$ rmdir receive

4. The 'cd' command:

The cd command is used to move from one directory to another.

SYNTAX: \$ cd dirname

EXAMPLE: \$ cd receive

5. The 'ls' command:

## 1.2 : Directory commands

① \$ pwd  
/home/cse 376

② \$ mkdir sec cse 376

→ \$

③ \$ cd sec cse 376

→ \$

④ \$ ls

addition. C a.out multiplication. C

### 1.3 File handling commands

① \$ cat > sec

\$ cat > sec 376

→ enter contents separately for both files

② \$ cat sec

\$ cat sec 376

→ cse department → cse d section os.lab

③ \$ cat sec sec 376

→ \$

\$ cat sec sec 376

→ CSE department

④ \$ rm sec

→ \$ cat sec

cat : sec : No such directory or file

The ls command displays the list of files in the current working directory.

SYNTAX: \$ ls

EXAMPLE: \$ ls

\$ ls -l

\$ ls -a

### 1.3 FILE HANDLING COMMANDS

1. The 'cat' command:

The cat command is used to create a file.

SYNTAX: \$ cat > filename

EXAMPLE: \$ cat > rec

2. The 'Display contents of a file' command:

The cat command is also used to view the contents of a specified file.

SYNTAX: \$ cat filename

3. The 'cp' command:

The cp command is used to copy the contents of one file to another and copies the file from one place to another.

SYNTAX: \$ cp oldfile newfile

EXAMPLE: \$ cp cse ece

4. The 'rm' command:

The rm command is used to remove or erase an existing file

SYNTAX: \$ rm filename

EXAMPLE: \$ rm rec

\$ rm -f rec

Use option -fr to delete recursively the contents of the directory and its subdirectories.

5. The 'mv' command:

The mv command is used to move a file from one place to another. It removes a specified file from its original location and places it in specified location.

SYNTAX: \$ mv oldfile newfile

EXAMPLE: \$ mv cse eee

6. The 'file' command:

The file command is used to determine the type of file.

SYNTAX: \$ file filename

EXAMPLE: \$ file receee

⑤ \$ mv file1 file2

Output file1 is moved to file2

⑥ \$ wc 396

Output 1 2 12 396

⑦ \$ ls -P \*\*

Output

Pradeep praveen

### 1.4 piping commands

1. \$ who -d date

Output

student pts/0 2025-01-25 13:31 (:0)

student pts/1 2025-01-25 13:34 (:0)

saturday Jan 25 14:09:30 IST 2025

2. \$ who >> data

Output

student pts/0 2025-01-25 13:31 (:0)

student pts/1 2025-01-25 13:34 (:0)

saturday Jan 25

14:09:30 IST 2025

3. \$ who >> data

Output

student pts/0 2025-01-25 13:31 (:0)

student pts/1 2025-01-25 13:34 (:0)

7. The 'wc' command:  
The wc command is used to count the number of words, lines and characters in a file.

SYNTAX: \$ wc filename  
EXAMPLE: \$ wc receive

8. The 'Directing output to a file' command:  
The ls command lists the files on the terminal (screen). Using the redirection operator '>' we can send the output to file instead of showing it on the screen.

SYNTAX: \$ ls > filename  
EXAMPLE: \$ ls > csece

9. The "pipes" command:  
The Unix allows us to connect two commands together using these pipes. A pipe ( | ) is an mechanism by which the output of one command can be channeled into the input of another command.

SYNTAX: \$ command1 | command2  
EXAMPLE: \$ who | wc -l  
10. The 'tee' command:  
While using pipes, we have not seen any output from a command that gets piped into another command. To save the output, which is produced in the middle of a pipe, the tee command is very useful.

SYNTAX: \$ command | tee filename  
EXAMPLE: \$ who | tee sample | wc -l

11. The 'Metacharacters of unix' command:  
Metacharacters are special characters that are at higher and abstract level compared to most of other characters in Unix. The shell understands and interprets these metacharacters in a special way.  
\* - Specifies number of characters  
? - Specifies a single character  
[] - Used to match a whole set of file names at a command line.

EXAMPLE:  
\$ ls r\* - Displays all the files whose name begins with 'r'  
\$ ls Tkk - Displays the files which are having 'kkk', from the second characters irrespective of the first character.

- \$ ls [a-n] - Lists the files whose names begins alphabets from 'a' to 'm'  
\$ ls [a-m] - Lists all files other than files whose names begins alphabets from 'a' to 'm'.  
12.

## 1.5 Filters

1. \$ head student

output

c programming

python

java

html

css

java script

html

css

output

student

tail

java script

html

css

output

student

DBMS

operating system

java script

html

css

output

4. \$ sort students

output

Kishan CSE

Parman CSE

Suhbu CSE

Thouni CSE

The 'File permissions' command:

File permission is the way of controlling the accessibility of file for each of three users namely Users, Groups and Others.

There are three types of file permissions are available, they are

r-read	
w-write	
x-execute	

The permissions for each file can be divided into three parts of three bits each.

First three bits	Owner of the file
Next three bits	Group to which owner of the file belongs
Last three bits	Others

EXAMPLE: \$ ls college

-rw-r-x-- 1 Lak sid 1525 jan10 12:10 college

Where,

-rw- The file is readable, writable and executable by the owner of the file.

Lak Specifies Owner of the file.

r-x Indicates the absence of the write permission by the Group owner of the file. Sid is the Group Owner of the file.

r- Indicates read permissions for others.

13. The 'chmod' command:

The chmod command is used to set the read, write and execute permissions for all categories of users for file.

SYNTAX: \$ chmod category operation permission file

Category	Operation	permission
u-users	+ assign	r-read
g-group	- Remove	w-write
o-others	= assign absolutely	x-execute
a-all		

5 - \$ nl students

### Output:

1. thorun cse
2. hari cse
3. praveen cse
4. kishan ece
5. thorun cse
6. sulhu cse



#### EXAMPLE:

\$ chmod u-wx college  
Removes write & execute permission for users for 'college' file.

\$ chmod u+rw, g+rwx college  
Assigns read & write permission for users and groups for 'college' file.

\$ chmod g-wx college  
Assigns absolute permission for groups of all read, write and execute permissions for 'college' file.

14. The 'Octal Notations' command:

The file permissions can be changed using octal notations also. The octal notations for file permission are

Read permission	4
Write permission	2

EXAMPLE:  
\$ chmod 761 college

Execute permission	1
--------------------	---

Assigns all permission to the owner, read and write permissions to the group and only executable permission to the others for 'college' file.

### 1.4 GROUPING COMMANDS

#### 1. The 'semicolon' command:

The semicolon(;) command is used to separate multiple commands at the command line.

SYNTAX: \$ command1;command2;command3.....;commandn

EXAMPLE: \$ who;date

#### 2. The '&&' operator:

The '&&' operator signifies the logical AND operation in between two or more valid Unix commands. It means that only if the first command is successfully executed, then the next command will be executed.

SYNTAX: \$ command1 && command2 && command3.....&& commandn

EXAMPLE: \$ who && date

## 1.5 Other essential commands

1. \$ free  
output total used free shared buff/cache available

Mem	4062408	49336	264704	66392	921368	330088
Swap	3424252	0	0	3424252		

15 OTHER ESSENTIAL COMMANDS  
1. free

Display amount of free and used physical and swapped memory system.

2. \$ ps

<u>Output</u>	PID	TTY	TIME	CMD
	1597	pts/1	00:00:00	bash
	1777	pts/1	00:00:00	PS
				Total: 6665816 605464 4666516

3. \$ vmstat

Output

proc - - - - - memory - - - - - swap - - - - - system - - - - -  
or  
free swap free buff cache si so bi bo in  
1 0 0 2643360 63120 857804 0 0 80 19 21

4. \$ off  
output

file system

dev mounts

tmpfs

/dev/mapper/  
fedora /home

1K-blocks used available use% mounted on

2020216

0 2020216 0%

/dev

2031204 0 2031204 0% /dev/shm

18261268 310460 16970128 2% /home

fedora /home

synopsis- top [options]  
example

[root@localhost ~]# free -t

total used free shared buff/cache available

Mem: 4044380 605464 2045080

Swap: 2621436 0 2621436

Total: 6665816 605464 4666516

2. top

It provides a dynamic real-time view of processes in the system.

synopsis- ps [options]  
example

[root@localhost ~]# top

top - 08:07:28 up 24 min, 2 users, load average: 0.01, 0.06, 0.23

Tasks: 211 total, 1 running, 210 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.8 us, 0.3 sy, 0.0 id, 98.9 id, 0.0 wa, 0.0 hi, 0.0 st

KiB Mem: 4044380 total, 2052960 free, 600452 used, 1390968 buff/cache KiB Swap:

2621436 total, 2621436 free, 0 used, 3234820 avail Mem PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

1105 root 20 0 175008 75700 51264 S 1.7 1.9 0 20.46 Xorg 2579 root 20 0 80444 32640 24796 S 1.0 0.8 0 0 2.47 gnome-terminal 3 ps

It reports the snapshot of current processes

synopsis- ps [options]

example

[root@localhost ~]# ps -c

## 1.5 Other essential commands

1. `$ free` total used free shared buff/cache available

Mem	4062408	493136	267704	66392	921368	330088
Swap	3424252	0	0	3424252		

2. `$ ps`

<u>ps</u>		PID			TIME		CMD	
		1597	pts/1	00:00:00	bash			
		1777	pts/1	00:00:00	PS			

3. `$ vmstat`

<u>vmstat</u>		procs - - - - memory - - - swap - - - system - cpus - - - - -										
		1	0	0	2643360	68120	857804	0	0	80	19	21

### 1.5 OTHER ESSENTIAL COMMANDS

#### 1. free

Display amount of free and used physical and swapped memory system.

synopsis- free [options]

#### example

```
[root@localhost ~]# free -t
```

total used free shared buff/cache available Mem: 4044380 605464 2045080  
148820 1393836 3226708 Swap: 2621436 0 2621436

```
Total: 6665816 605464 4666516 0:00:00 1249 1111 1111 1111
```

#### 2. top

It provides a dynamic real-time view of processes in the system.

synopsis- top [options]

#### example

```
[root@localhost ~]# top
```

top - 08-07-28 up 24 min, 2 users, load average: 0.01, 0.06, 0.23

Tasks: 211 total, 1 running, 210 sleeping, 0 stopped, 0 zombie

%Cpu(s): 0.8 us, 0.3 sy, 0.0 ni, 98.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 4044380 total, 2052960 free, 600452 used, 1390968 buff/cache KiB Swap:

2621436 total, 2621436 free, 0 used, 3234820 avail Mem PID USER PR NI VIRT RES

SRH S %CPU %MEM TIME+ COMMAND

1105 root 20 0 175008 75700 51264 S 1.1 1.9 0:20:46 Xorg 2529 root 20 0 80444

32640 24796 S 1.0 0.3 0:02:47 gnome-terminal 3 ps

It reports the snapshot of current processes

synopsis- ps [options]

#### example

```
[root@localhost ~]# ps -c
```

### Other essential commands



- 4. \$ off  
output
- File system
- shortcuts
- trayfs
- anon / swapfs
- fdisk / hdd

```
root@system1K:~# blocks Used Available Use% Mounted on
/dev/sda1 1018000 0 1018000 0% /dev
tmpfs 20221188 148 2022040 1% /dev/shm
tmpfs 20221188 1404 2020784 1% /run
/dev/sda6 487652 168276 289680 37% /boot
```

11 REPORTS VIRTUAU MEMORY STATISTICS

PID	TTY	TIME	CMD
1	?	00:00:03	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	ksmfulld/0
4			vmstat

```
64 bytes from 172.16.4.1: icmp_seq=3 ttl=64 time=0.264 ms
64 bytes from 172.16.4.1: icmp_seq=4 ttl=64 time=0.312 ms
^C
-- 172.16.4.1 ping statistics --
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.228/0.283/0.328/0.039 ms
```

## 7. ifconfig

It is used to configure network interface.

**synopsis-** ifconfig [options]

**example**

```
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu
1500 net 172.16.6.102 netmask 255.255.252.0 broadcast 172.16.7.255 mac
fe80::4ab0:cf1ff:fe6d:6057 brd fe80::ff:fe6d:6057 pretxqueuelen 1000 (Ethernet)
      RX packets 23216 bytes 2483338 (2.3 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 107740 (105.2 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 8.
```

**traceroute**

It tracks the route the packet takes to reach the destination.

**synopsis-** traceroute [options]

**example**

```
[root@localhost ~]# traceroute www.rajalakshmi.org
traceroute to www.rajalakshmi.org (220.227.30.51), 30 hops max, 60 byte
packets
 1 gateway (172.16.4.1) 0.299 ms 0.297 ms 0.327 ms
 2 220.225.219.38 (220.225.219.38) 6.183 ms 6.203 ms 6.189 ms
```

**Result:**

The basic Linux commands like few general purpose directory, file handling - grouping, filters and other essential commands have been executed successfully.