

Ex No: 1b)**Date: 24.1.25****BASIC LINUX COMMANDS****1.1 GENERAL PURPOSE COMMANDS**

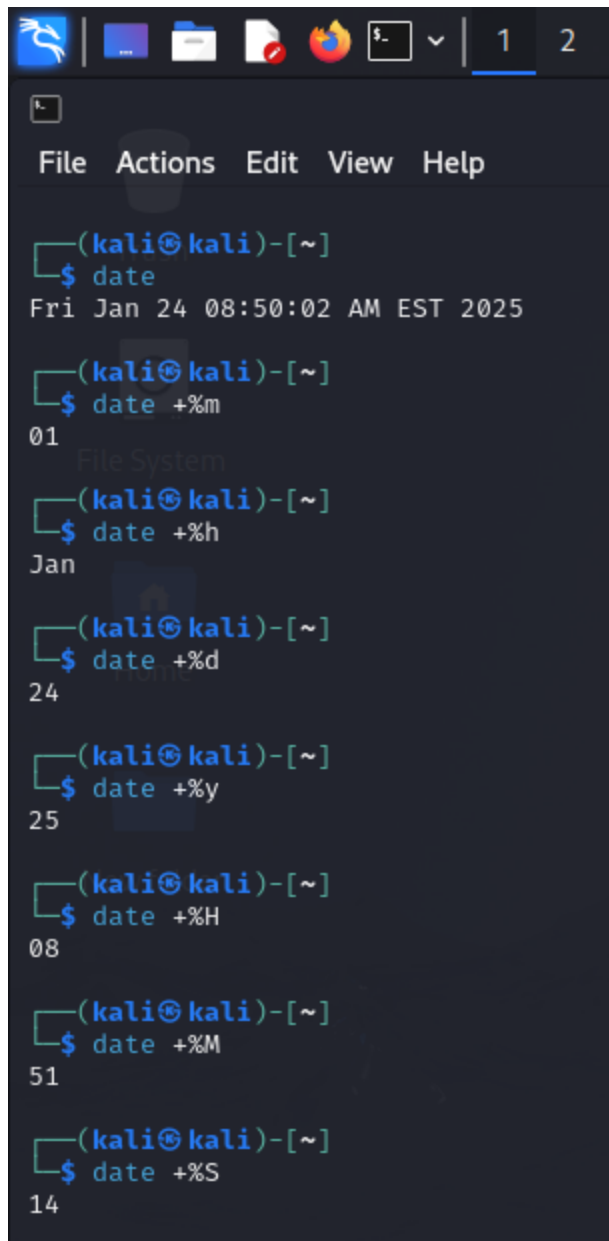
1. The 'date' command:

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



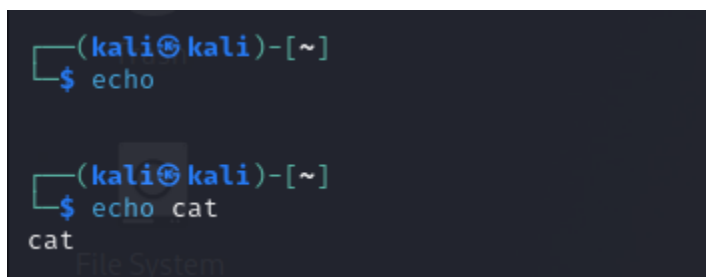
```
(kali㉿kali)-[~]  
$ date  
Fri Jan 24 08:50:02 AM EST 2025  
  
(kali㉿kali)-[~]  
$ date +%m  
01  
  
(kali㉿kali)-[~]  
$ date +%h  
Jan  
  
(kali㉿kali)-[~]  
$ date +%d  
24  
  
(kali㉿kali)-[~]  
$ date +%y  
25  
  
(kali㉿kali)-[~]  
$ date +%H  
08  
  
(kali㉿kali)-[~]  
$ date +%M  
51  
  
(kali㉿kali)-[~]  
$ date +%S  
14
```

2. The echo'command:

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

SYNTAX: \$ echo

EXAMPLE: \$ echo "God is Great"



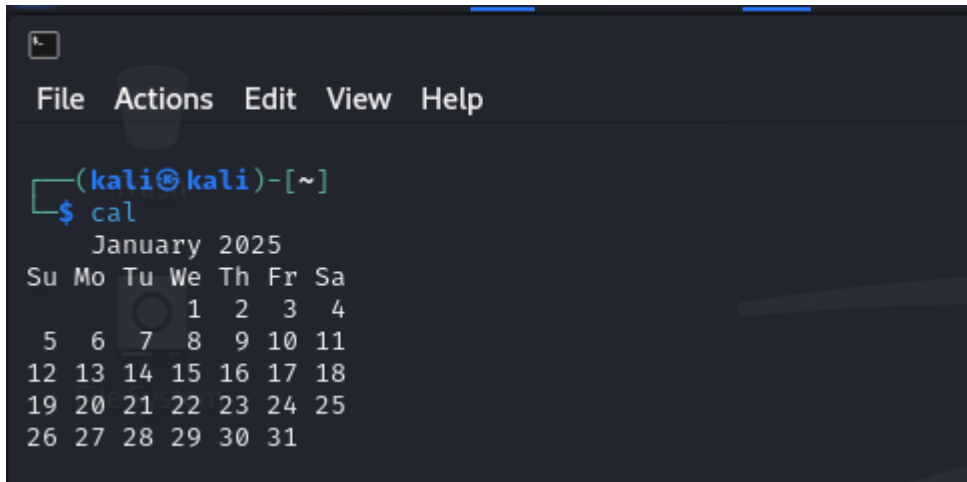
```
(kali㉿kali)-[~]  
$ echo  
  
(kali㉿kali)-[~]  
$ echo cat  
cat
```

3. The 'cal' command:

The cal command displays the specified month or year calendar.

SYNTAX: \$ cal [month] [year]

EXAMPLE: \$ cal Jan 2012



```
(kali@kali)-[~]  
$ cal  
January 2025  
Su Mo Tu We Th Fr Sa  
      1  2  3  4  
 5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30 31
```

4. The 'bc' command:

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

SYNTAX: \$ bc

EXAMPLE: bc -l

16/4

5/2

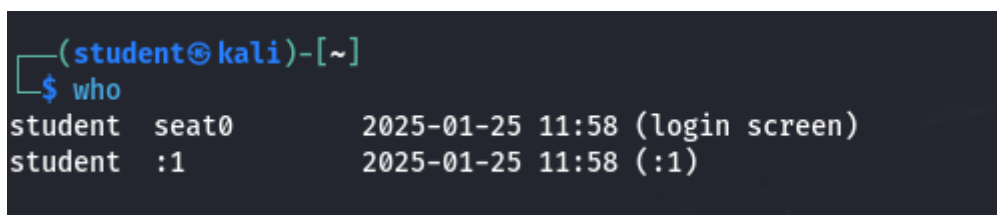


```
(kali@kali)-[~]  
$ bc -l  
bc 1.07.1  
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type `warranty'.  
5+3  
8  
File System  
5-6  
-1  
█  
Home
```

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

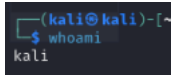


```
(student@kali)-[~]  
$ who  
student seat0 2025-01-25 11:58 (login screen)  
student :1 2025-01-25 11:58 (:1)
```

6. The 'who am i' command

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

SYNTAX: \$ whoami

A terminal window with a dark background. The prompt is '(kali@kali)-[~]'. The user has entered '\$ whoami' and the output is 'kali'.

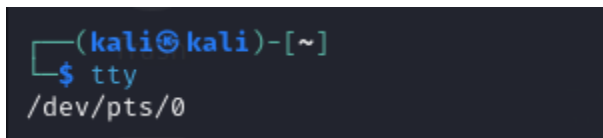
7. The 'id' command

The id command displays the numerical value corresponding to your login. SYNTAX: \$ id

A terminal window with a dark background. The prompt is '(kali@kali)-[~]'. The user has entered '\$ id' and the output is 'uid=1000(kali) gid=1000(kali) groups=1000(kali),4(ade),28(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),46(video),46(plugindev),100(users),101(netdev),106(bluetooth),113(scanner),136(wireshark),137(kahoxer),138(vboxsf)'.

8. The 'tty' command

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

A terminal window with a dark background. The prompt is '(kali@kali)-[~]'. The user has entered '\$ tty' and the output is '/dev/pts/0'.

9. The 'clear' command

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

SYNTAX: \$ clear

A terminal window with a dark background. The prompt is '(kali@kali)-[~]'. The user has entered '\$ clear' and the screen is cleared.

10. The 'man' command

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

SYNTAX: \$ man [command]

```

ls(1)                                User Commands
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
      with -l, print the author of each file
  -b, --escape
      print C-style escapes for nongraphic characters
  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups
      do not list implied entries ending with ~
  -C
      with -lt: sort by, and show, ctime (time of last change of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first
  -c
      list entries by columns
  --color[=WHEN]
      color the output WHEN; more info below
  -d, --directory
      list directories themselves, not their contents
  -D, --dired
      generate output designed for Emacs' dired mode
  -f
      list all entries in directory order
  -F, --classify[=WHEN]
      append indicator (one of */=>@) to entries WHEN
Manual page ls(1) line 1 (press h for help or q to quit)

```

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

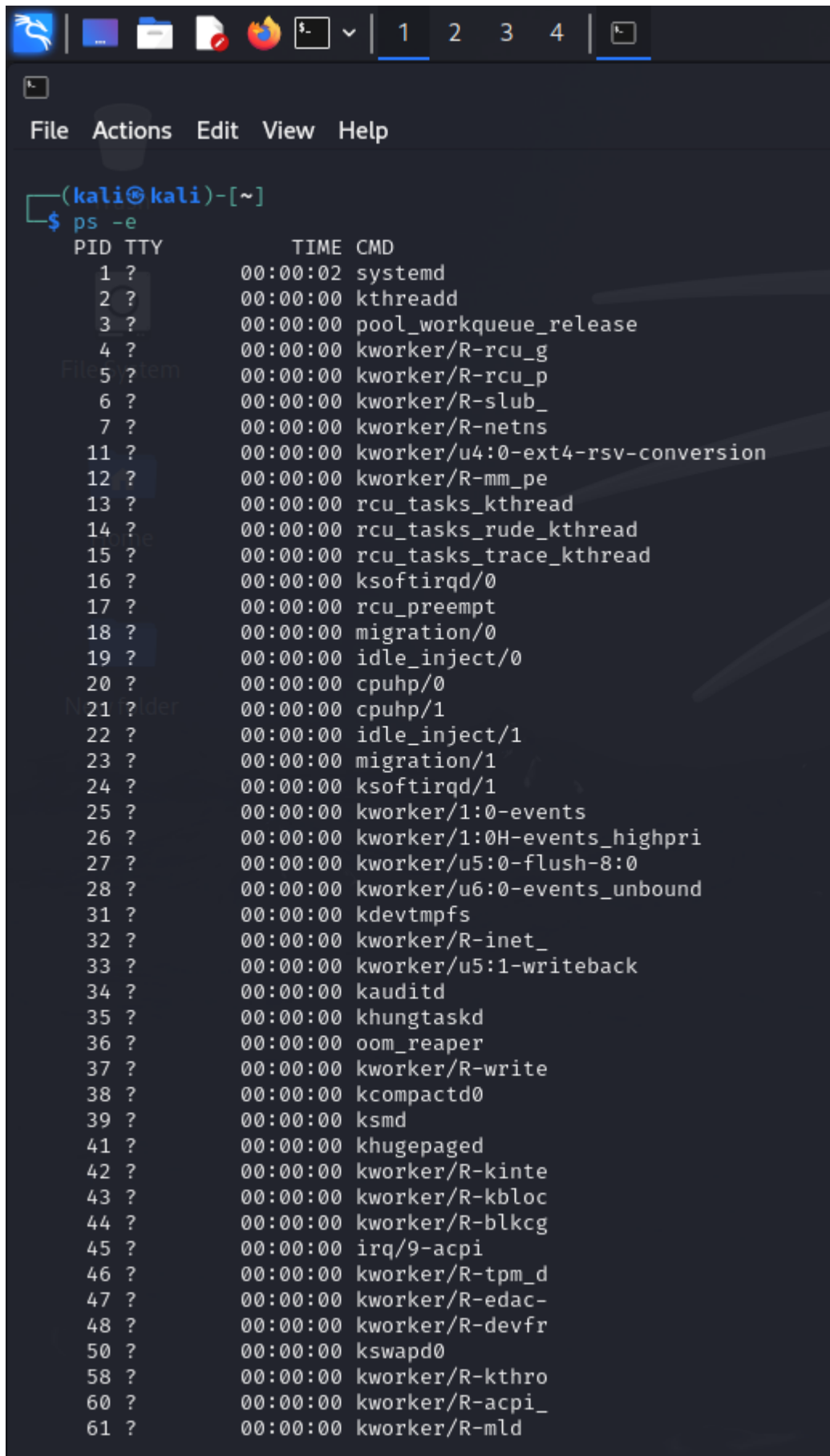
EXAMPLE: \$ ps

```

(kali㉿kali)-[~]
$ ps
  PID TTY          TIME CMD
 1544 pts/0    00:00:03 zsh
 5666 pts/0    00:00:00 bc
 6068 pts/0    00:00:00 bc
 6684 pts/0    00:00:00 bc
 6868 pts/0    00:00:00 bc
 9412 pts/0    00:00:00 ps

```

\$ ps -e



```
(kali㉿kali)-[~]  
$ ps -e  
  PID TTY          TIME CMD  
    1 ?        00:00:02 systemd  
    2 ?        00:00:00 kthreadd  
    3 ?        00:00:00 pool_workqueue_release  
    4 ?        00:00:00 kworker/R-rcu_g  
    5 ?        00:00:00 kworker/R-rcu_p  
    6 ?        00:00:00 kworker/R-slub_  
    7 ?        00:00:00 kworker/R-netns  
   11 ?        00:00:00 kworker/u4:0-ext4-rsv-conversion  
   12 ?        00:00:00 kworker/R-mm_pe  
   13 ?        00:00:00 rcu_tasks_kthread  
   14 ?        00:00:00 rcu_tasks_rude_kthread  
   15 ?        00:00:00 rcu_tasks_trace_kthread  
   16 ?        00:00:00 ksoftirqd/0  
   17 ?        00:00:00 rcu_preempt  
   18 ?        00:00:00 migration/0  
   19 ?        00:00:00 idle_inject/0  
   20 ?        00:00:00 cpuhp/0  
   21 ?        00:00:00 cpuhp/1  
   22 ?        00:00:00 idle_inject/1  
   23 ?        00:00:00 migration/1  
   24 ?        00:00:00 ksoftirqd/1  
   25 ?        00:00:00 kworker/1:0-events  
   26 ?        00:00:00 kworker/1:0H-events_highpri  
   27 ?        00:00:00 kworker/u5:0-flush-8:0  
   28 ?        00:00:00 kworker/u6:0-events_unbound  
   31 ?        00:00:00 kdevtmpfs  
   32 ?        00:00:00 kworker/R-inet_  
   33 ?        00:00:00 kworker/u5:1-writeback  
   34 ?        00:00:00 kauditd  
   35 ?        00:00:00 khungtaskd  
   36 ?        00:00:00 oom_reaper  
   37 ?        00:00:00 kworker/R-write  
   38 ?        00:00:00 kcompactd0  
   39 ?        00:00:00 ksmd  
   41 ?        00:00:00 khugepaged  
   42 ?        00:00:00 kworker/R-kinte  
   43 ?        00:00:00 kworker/R-kbloc  
   44 ?        00:00:00 kworker/R-blkcg  
   45 ?        00:00:00 irq/9-acpi  
   46 ?        00:00:00 kworker/R-tpm_d  
   47 ?        00:00:00 kworker/R-edac-  
   48 ?        00:00:00 kworker/R-devfr  
   50 ?        00:00:00 kswapd0  
   58 ?        00:00:00 kworker/R-kthro  
   60 ?        00:00:00 kworker/R-acpi_  
   61 ?        00:00:00 kworker/R-mld
```

\$ps -aux

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

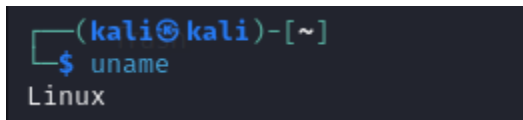
system. -s -> Displays the name of the operating

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

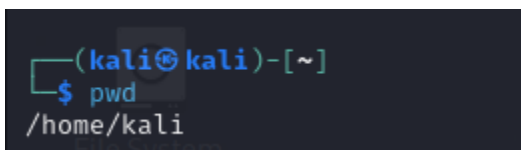


```
(kali@kali)-[~]  
$ uname  
Linux
```

1.2 DIRECTORY COMMANDS

1. The 'pwd' command:

The pwd (print working directory) command displays the current working directory. SYNTAX: \$ pwd



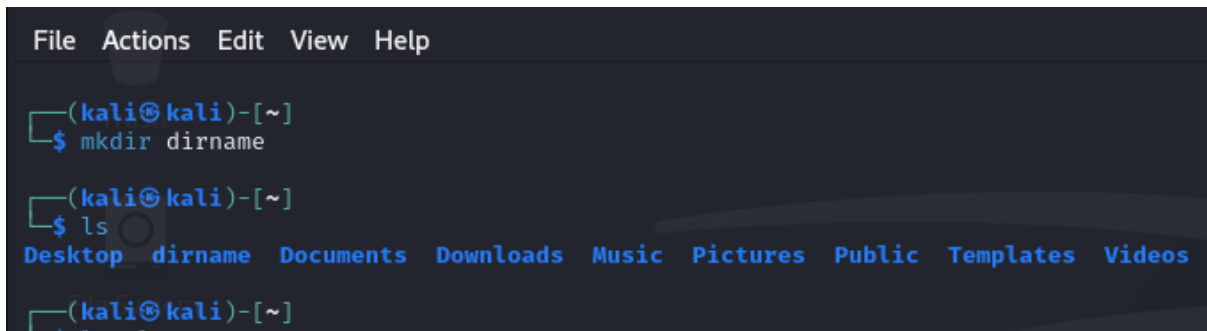
```
(kali@kali)-[~]  
$ pwd  
/home/kali
```

2. The 'mkdir' command:

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

SYNTAX: \$ mkdir dirname

EXAMPLE: \$ mkdir receee



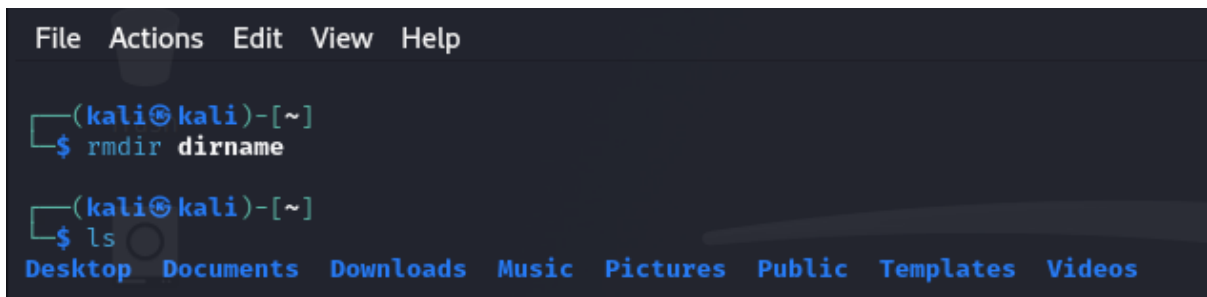
```
File Actions Edit View Help
(kali@kali)-[~]
$ mkdir dirname
(kali@kali)-[~]
$ ls
Desktop  dirname  Documents  Downloads  Music  Pictures  Public  Templates  Videos
(kali@kali)-[~]
```

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 receee`

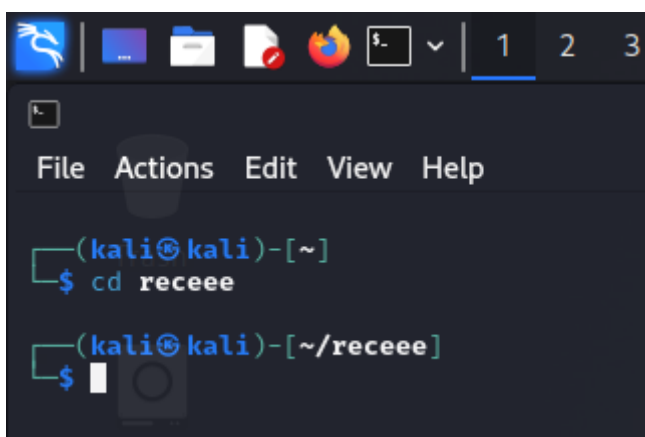


```
File Actions Edit View Help
(kali@kali)-[~]
$ rmdir dirname
(kali@kali)-[~]
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

4. The 'cd' command:

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

SYNTAX: `$ cd dirname` EXAMPLE: `$ cd receee`



```
File Actions Edit View Help
(kali@kali)-[~]
$ cd receee
(kali@kali)-[~/receee]
$
```

5. The 'ls' command:

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

SYNTAX: `$ ls`

EXAMPLE: \$ ls

\$ ls -l

\$ ls -a

```
(kali@kali)-[~]
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  reccee  Templates  Videos
(kali@kali)-[~]
$ ls -l
total 36
drwxr-xr-x 3 kali kali 4096 Jan 21 06:01 Desktop
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Documents
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Downloads
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Music
drwxr-xr-x 2 kali kali 4096 Jan 26 09:06 Pictures
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Public
drwxr-xr-x 2 kali kali 4096 Jan 24 09:04 reccee
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Templates
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Videos
(kali@kali)-[~]
$ ls -a
.          .cache      Downloads  .java      Public      .vboxclient-clipboard-tty7-service.pid  .vboxclient-hostversion-tty7-control.pid  .Xauthority
.bash_logout  .config     Desktop    .face      .local      .vboxclient-display-svgx-x11-tty7-control.pid  .vboxclient-seamless-tty7-control.pid      .xsession-errors
.bashrc       .dirc      .face.icon  Music      .sudo_as_admin_successful  .vboxclient-display-svgx-x11-tty7-service.pid  .vboxclient-seamless-tty7-service.pid      .xsession-errors.old
.bashrc.original  Documents  .ICAuthority  .profile  .vboxclient-clipboard-tty7-control.pid  .vboxclient-draganddrop-tty7-control.pid  .vboxclient-vmvga-session-tty7-control.pid  .zshrc
Videos
```

1.3 FILE HANDLING COMMANDS

1. The 'cat' command:

The cat command is used to create a file.

SYNTAX: \$ cat > filename

EXAMPLE: \$ cat > rec

```
(kali@kali)-[~]
$ cat >rec
Hi
```

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

```
(kali@kali)-[~]
$ cat rec
Hid
```

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

```
(kali@kali)-[~]
$ cp cse eee
```

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



```
(kali@kali)-[~]  
$ rm 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



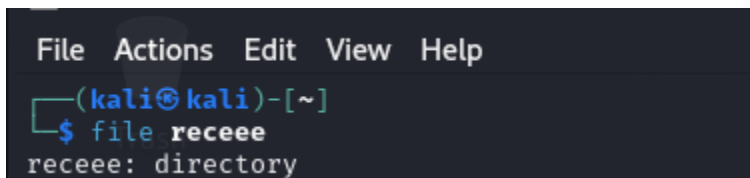
```
(kali@kali)-[~]  
$ cat > cse  
CSE  
IT  
CSE(CS)  
(kali@kali)-[~]  
$ cat > ece  
ECE  
EEE  
(kali@kali)-[~]  
$ mv cse ece
```

6. The 'file' command:

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

SYNTAX: \$ file filename

EXAMPLE: \$ file receee



```
File Actions Edit View Help  
(kali@kali)-[~]  
$ file receee  
receee: directory
```

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 receee




```
(kali@kali)-[~]
$ wc ece
3 3 15 ece
```

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

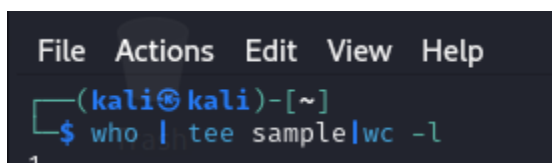
EXAMPLE: \$ ls > cseeee



```
(kali@kali)-[~]
$ ls > cse
```

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

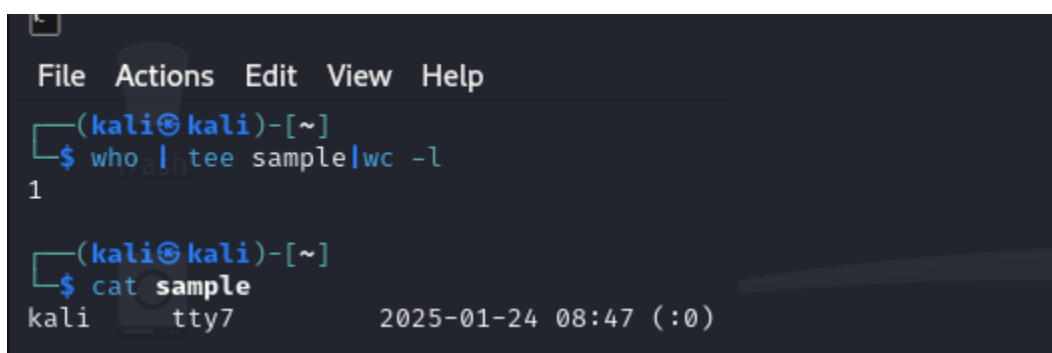


```
File Actions Edit View Help
(kali@kali)-[~]
$ who | tee sample | wc -l
1
```

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



```
File Actions Edit View Help
(kali@kali)-[~]
$ who | tee sample | wc -l
1

(kali@kali)-[~]
$ cat sample
kali tty7 2025-01-24 08:47 (:0)
```

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. ! – Used to Specify Not EXAMPLE:

\$ ls r** - Displays all the files whose name begins with 'r'

\$ ls ?kkk - Displays the files which are having 'kkk', from the second characters irrespective of the first character.

\$ ls [a-m] – 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'



```
(kali㉿kali)-[~]
└─$ ls r**
rit
recee:
```

12.

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

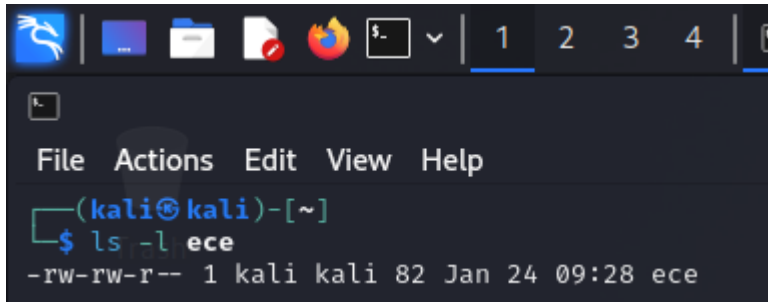
EXAMPLE: \$ ls college

-rwxr-xr-- 1 Lak std 1525 jan10 12:10 college

Where,

-rwx 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. Std Is the Group Owner of the file.



```

(kali@kali)-[~]
$ ls -l ece
-rw-rw-r-- 1 kali kali 82 Jan 24 09:28 ece

```

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		

EXAMPLE:

\$ chmod u -wx college

Removes write & execute permission for users for 'college' file.

\$ chmod u +rw, g+rw 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.



```

(kali@kali)-[~]
$ chmod u-wx ece

```

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.

```
(kali㉿kali)-[~]
$ chmod u-wx ece

(kali㉿kali)-[~]
$ chmod 761 receee

(kali㉿kali)-[~]
$ ls -l
total 48
drwxr-xr-x 3 kali kali 4096 Jan 21 06:01 Desktop
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Documents
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Downloads
-r--r--r-- 1 kali kali 82 Jan 24 09:28 ece
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Music
drwxr-xr-x 2 kali kali 4096 Jan 24 09:36 Pictures
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Public
drwxrw--- 2 kali kali 4096 Jan 24 09:04 receee
-rw-rw-r-- 1 kali kali 4 Jan 24 09:06 rit
-rw-rw-r-- 1 kali kali 44 Jan 24 09:29 sample
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Templates
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Videos
```

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

```
File Actions Edit View Help
(kali@kali)-[~]
$ who;date
kali      tty7          2025-01-24 08:47 (:0)
Fri Jan 24 09:37:30 AM EST 2025
(kali@kali)-[~]
```

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

```
File Actions Edit View Help
(kali@kali)-[~]
$ who&&date
kali      tty7          2025-01-24 08:47 (:0)
Fri Jan 24 09:37:56 AM EST 2025
```

3. The '||' operator:

The '||' operator signifies the logical OR operation in between two or more valid Unix commands. It means, that only if the first command will happen to be unsuccessful, it will continue to execute next commands.

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

EXAMPLE: \$ who || date

```
File Actions Edit View Help
(kali@kali)-[~]
$ who||date
kali      tty7          2025-01-24 08:47 (:0)
```

1.5 FILTERS

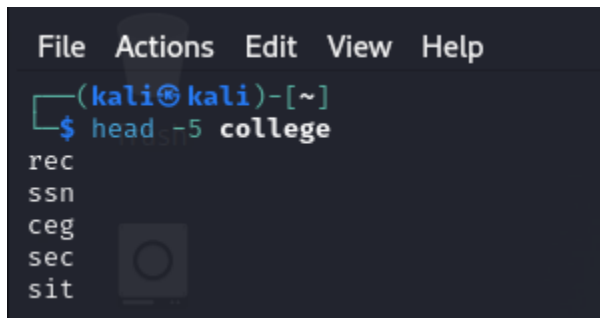
1. The head filter

It displays the first ten lines of a file.

SYNTAX: \$ head filename

EXAMPLE: `$ head college` Display the top ten lines.

`$ head -5 college` Display the top five lines.

A terminal window with a dark background and light-colored text. The menu bar at the top shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The command '\$ head -5 college' has been entered. The output shows the first five lines of the 'college' file: 'rec', 'ssn', 'ceg', 'sec', and 'sit'.

```
File Actions Edit View Help
(kali@kali)-[~]
$ head -5 college
rec
ssn
ceg
sec
sit
```

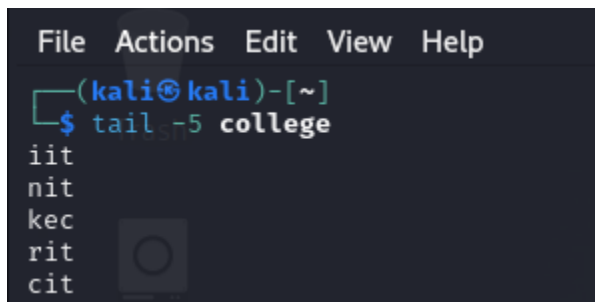
2. The tail filter

It displays ten lines of a file from the end of the file.

SYNTAX: `$ tail filename`

EXAMPLE: `$ tail college` Display the last ten lines.

`$tail -5 college` Display the last five lines.

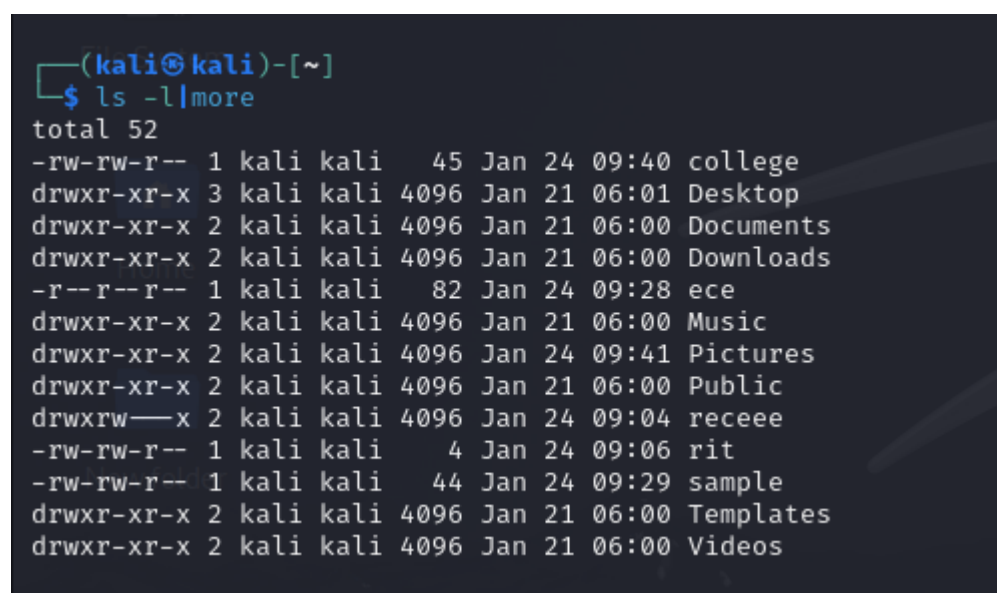
A terminal window with a dark background and light-colored text. The menu bar at the top shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The command '\$ tail -5 college' has been entered. The output shows the last five lines of the 'college' file: 'iit', 'nit', 'kec', 'rit', and 'cit'.

```
File Actions Edit View Help
(kali@kali)-[~]
$ tail -5 college
iit
nit
kec
rit
cit
```

3. The more filter:

The pg command shows the file page by page.

SYNTAX: `$ ls -l | more`

A terminal window with a dark background and light-colored text. The menu bar at the top shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The command '\$ ls -l | more' has been entered. The output shows the first page of the 'ls -l' command output, listing files and directories with their permissions, owner, group, size, date, and name. The first line is 'total 52'. The following lines are: '-rw-rw-r-- 1 kali kali 45 Jan 24 09:40 college', 'drwxr-xr-x 3 kali kali 4096 Jan 21 06:01 Desktop', 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Documents', 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Downloads', '-r--r--r-- 1 kali kali 82 Jan 24 09:28 ece', 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Music', 'drwxr-xr-x 2 kali kali 4096 Jan 24 09:41 Pictures', 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Public', 'drwxrw-x 2 kali kali 4096 Jan 24 09:04 receee', '-rw-rw-r-- 1 kali kali 4 Jan 24 09:06 rit', '-rw-rw-r-- 1 kali kali 44 Jan 24 09:29 sample', 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Templates', and 'drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Videos'.

```
(kali@kali)-[~]
$ ls -l | more
total 52
-rw-rw-r-- 1 kali kali 45 Jan 24 09:40 college
drwxr-xr-x 3 kali kali 4096 Jan 21 06:01 Desktop
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Documents
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Downloads
-r--r--r-- 1 kali kali 82 Jan 24 09:28 ece
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Music
drwxr-xr-x 2 kali kali 4096 Jan 24 09:41 Pictures
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Public
drwxrw-x 2 kali kali 4096 Jan 24 09:04 receee
-rw-rw-r-- 1 kali kali 4 Jan 24 09:06 rit
-rw-rw-r-- 1 kali kali 44 Jan 24 09:29 sample
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Templates
drwxr-xr-x 2 kali kali 4096 Jan 21 06:00 Videos
```

4. The 'grep' command:

This command is used to search for a particular pattern from a file or from the standard input and display those lines on the standard output. “Grep” stands for “global search for regular expression.”

SYNTAX: \$ grep [pattern] [file_name]

EXAMPLE: \$ cat> student

Arun cse

Ram ece

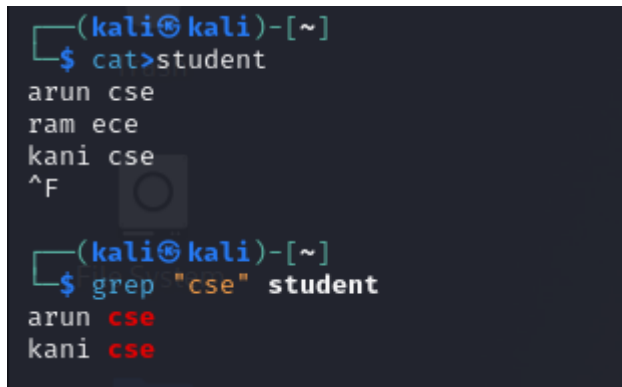
Kani cse

\$ grep “cse”

student

Arun cse

Kani cse



```
(kali㉿kali)-[~]  
$ cat>student  
arun cse  
ram ece  
kani cse  
^F  
  
(kali㉿kali)-[~]  
$ grep "cse" student  
arun cse  
kani cse
```

5. The ‘sort’ command:

The sort command is used to sort the contents of a file. The sort command reports only to the

screen, the actual file remains unchanged. SYNTAX: \$ sort

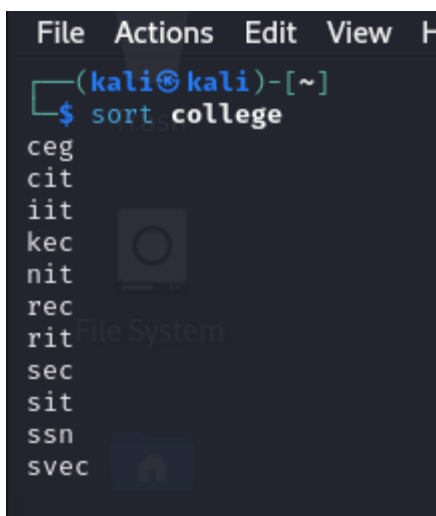
filename EXAMPLE: \$ sort

college OPTIONS:

Command	Purpose
Sort -r college	Sorts and displays the file contents in reverse order
Sort -c college	Check if the file is sorted
Sort -n college	Sorts numerically

Sort -m college	Sorts numerically in reverse order
--------------------	------------------------------------

Sort -u college	Remove duplicate records
Sort -l college	Skip the column with +1 (one) option.Sorts according to second column



```
File Actions Edit View H
(kali@kali)-[~]
$ sort college
ceg
cit
iit
kec
nit
rec
rit
sec
sit
ssn
svec
```

6. The 'nl' command:

The nl filter adds line numbers to a file and it displays the file and not provides access to edit but simply displays the contents on the screen.

SYNTAX: \$ nl filename

EXAMPLE: \$ nl college

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ nl college
 1 rec
 2 ssn
 3 ceg
 4 sec
 5 sit
 6 svec
 7 iit
 8 nit
 9 kec
10 rit
11 cit
```

7. The 'cut' command:

We can select specified fields from a line of text using cut command.

SYNTAX: \$ cut -c filename

EXAMPLE: \$ cut -c college

OPTION:

-c – Option cut on the specified character position from each line.

```
(kali㉿kali)-[~]
$ cut -c 1-2 college
re
ss
ce
se
si
sv
ii
ni
ke
ri
ci
```

1.5 OTHER ESSENTIAL COMMANDS

1. free

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

synopsis- free [options]

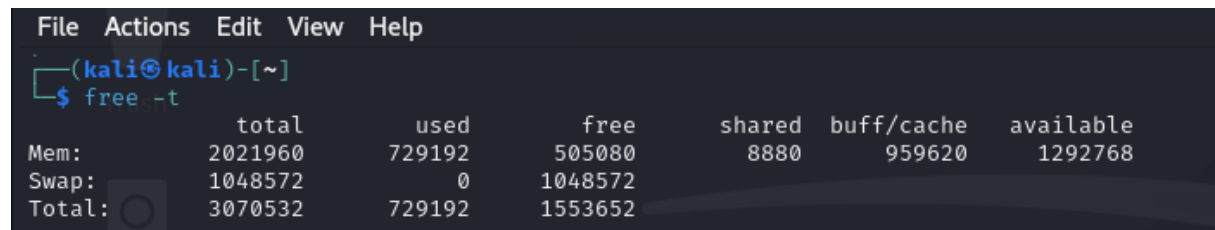
example

```
[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
```



	total	used	free	shared	buff/cache	available
Mem:	2021960	729192	505080	8880	959620	1292768
Swap:	1048572	0	1048572			
Total:	3070532	729192	1553652			

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
```

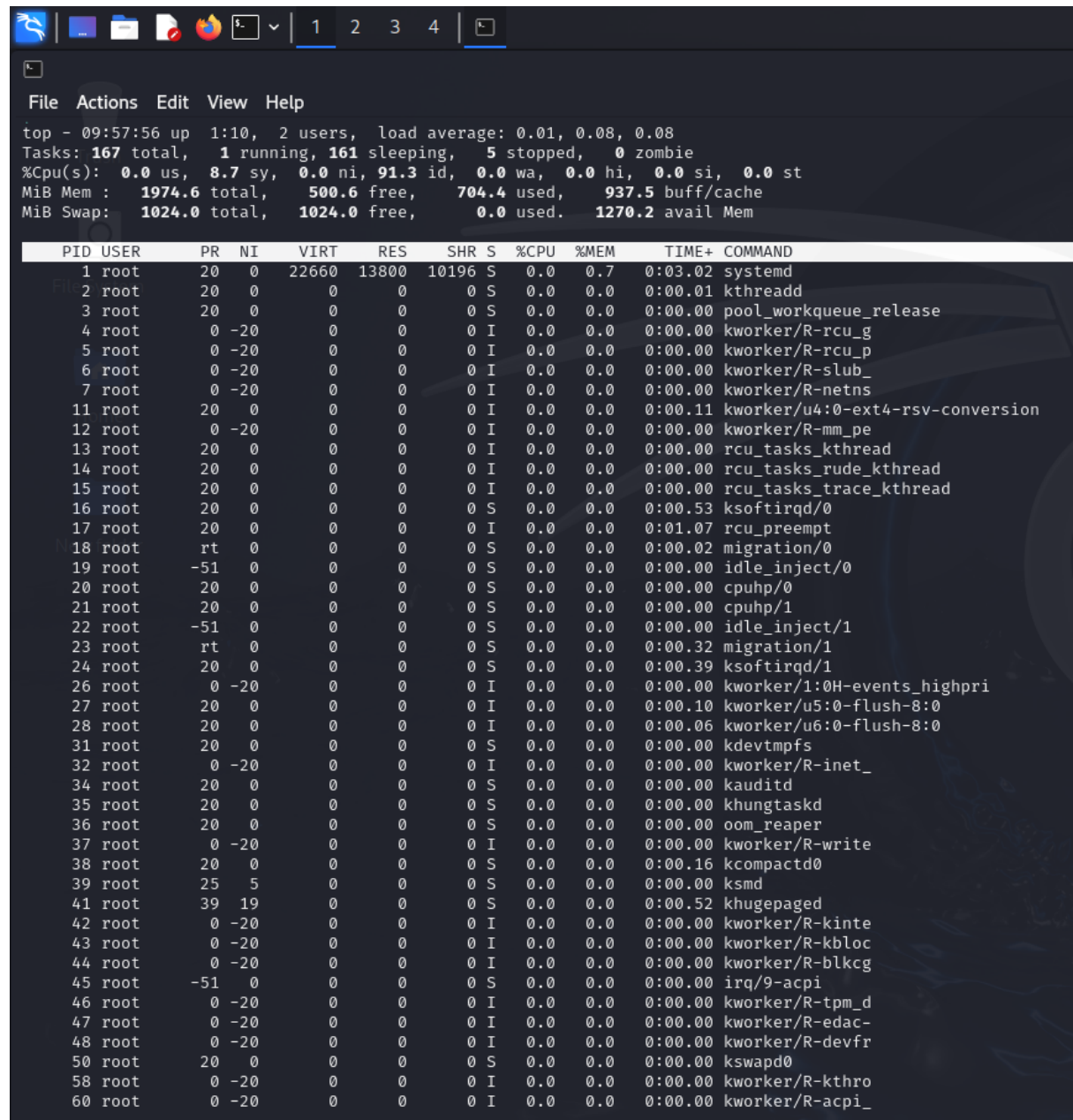
```
KiB Mem : 4044380 total, 2052960 free, 600452 used, 1390968 buff/cache KiB Swap:
```

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

```
SHR S %CPU %MEM TIME+ COMMAND
```

```
1105 root 20 0 175008 75700 51264 S 1.7 1.9 0:20.46 Xorg 2529 root 20 0
80444
```

```
32640 24796 S 1.0 0.8 0:02.47 gnome-term
```



```

top - 09:57:56 up 1:10, 2 users, load average: 0.01, 0.08, 0.08
Tasks: 167 total, 1 running, 161 sleeping, 5 stopped, 0 zombie
%Cpu(s): 0.0 us, 8.7 sy, 0.0 ni, 91.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1974.6 total, 500.6 free, 704.4 used, 937.5 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1270.2 avail Mem

  PID USER      PR  NI   VIRT    RES    SHR  S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   22660   13800   10196 S   0.0   0.7   0:03.02 systemd
    2 root        20   0     0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root        20   0     0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
   11 root        20   0     0      0      0 I   0.0   0.0   0:00.11 kworker/u4:0-ext4-rsv-conversion
   12 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
   13 root        20   0     0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   14 root        20   0     0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   15 root        20   0     0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   16 root        20   0     0      0      0 S   0.0   0.0   0:00.53 ksoftirqd/0
   17 root        20   0     0      0      0 I   0.0   0.0   0:01.07 rcu_preempt
   18 root        rt   0     0      0      0 S   0.0   0.0   0:00.02 migration/0
   19 root       -51   0     0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   20 root        20   0     0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   21 root        20   0     0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   22 root       -51   0     0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
   23 root        rt   0     0      0      0 S   0.0   0.0   0:00.32 migration/1
   24 root        20   0     0      0      0 S   0.0   0.0   0:00.39 ksoftirqd/1
   26 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
   27 root        20   0     0      0      0 I   0.0   0.0   0:00.10 kworker/u5:0-flush-8:0
   28 root        20   0     0      0      0 I   0.0   0.0   0:00.06 kworker/u6:0-flush-8:0
   31 root        20   0     0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   32 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-inet_
   34 root        20   0     0      0      0 S   0.0   0.0   0:00.00 kauditd
   35 root        20   0     0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   36 root        20   0     0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   37 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-write
   38 root        20   0     0      0      0 S   0.0   0.0   0:00.16 kcompactd0
   39 root        25   5     0      0      0 S   0.0   0.0   0:00.00 ksmd
   41 root        39  19     0      0      0 S   0.0   0.0   0:00.52 khugepaged
   42 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-kinte
   43 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-kbloc
   44 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-blkcg
   45 root       -51   0     0      0      0 S   0.0   0.0   0:00.00 irq/9-acpi
   46 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-tpm_d
   47 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-edac-
   48 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-devfr
   50 root        20   0     0      0      0 S   0.0   0.0   0:00.00 kswapd0
   58 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-kthro
   60 root         0 -20     0      0      0 I   0.0   0.0   0:00.00 kworker/R-acpi_

```

3. ps

It reports the snapshot of current processes

synopsis- ps [options]

example

```
[root@localhost ~]# ps -e
PID TTY TIME CMD
```

```
1 ? 00:00:03 systemd
```

```
2 ? 00:00:00 kthreadd
```

```
3 ? 00:00:00 ksoftirqd/0
```

```

File  Actions  Edit  View  Help

(kali㉿kali)-[~]
(kali㉿kali)-[~]
$ ps -e
  PID TTY          TIME CMD
    1 ?           00:00:03 systemd
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 pool_workqueue_release
    4 ?           00:00:00 kworker/R-rcu_g
    5 ?           00:00:00 kworker/R-rcu_p
    6 ?           00:00:00 kworker/R-slub_
    7 ?           00:00:00 kworker/R-netns
   11 ?           00:00:00 kworker/u4:0-ext4-rsv-conversion
   12 ?           00:00:00 kworker/R-mm_pe
   13 ?           00:00:00 rcu_tasks_kthread
   14 ?           00:00:00 rcu_tasks_rude_kthread
   15 ?           00:00:00 rcu_tasks_trace_kthread
   16 ?           00:00:00 ksoftirqd/0
   17 ?           00:00:01 rcu_preempt
   18 ?           00:00:00 migration/0
   19 ?           00:00:00 idle_inject/0
   20 ?           00:00:00 cpuhp/0
   21 ?           00:00:00 cpuhp/1
   22 ?           00:00:00 idle_inject/1
   23 ?           00:00:00 migration/1
   24 ?           00:00:00 ksoftirqd/1
   26 ?           00:00:00 kworker/1:0H-events_highpri
   27 ?           00:00:00 kworker/u5:0-flush-8:0
   31 ?           00:00:00 kdevtmpfs
   32 ?           00:00:00 kworker/R-inet_
   34 ?           00:00:00 kauditd
   35 ?           00:00:00 khungtaskd
   36 ?           00:00:00 oom_reaper
   37 ?           00:00:00 kworker/R-write
   38 ?           00:00:00 kcompactd0
   39 ?           00:00:00 ksmd
   41 ?           00:00:00 khugepaged
   42 ?           00:00:00 kworker/R-kinte
   43 ?           00:00:00 kworker/R-kbloc
   44 ?           00:00:00 kworker/R-blkcg
   45 ?           00:00:00 irq/9-acpi
   46 ?           00:00:00 kworker/R-tpm_d
   47 ?           00:00:00 kworker/R-edac-
   48 ?           00:00:00 kworker/R-devfr
   50 ?           00:00:00 kswapd0
   58 ?           00:00:00 kworker/R-kthro
   60 ?           00:00:00 kworker/R-acpi_
   61 ?           00:00:00 kworker/R-mld
   62 ?           00:00:00 kworker/R-ipv6_
   67 ?           00:00:00 kworker/R-kstrp

```

4. vmstat

It reports virtual memory statistics

synopsis- vmstat [options]

example

```
[root@localhost ~]# vmstat procs -----memory-----
---swap-- -----io----- -system-- -----cpu----- r b swpd free buff
cache si so bi bo in cs us sy id wa st 0 0 0 1879368 1604
1487116 0 0 64 7 72 140 1 0 97 1 0
```

5. df

It displays the amount of disk space available in file-system.

Synopsis- df [options]

example

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

Filesystem 1K-blocks Used Available Use% Mounted on

```
devtmpfs 2010800 0 2010800 0% /dev tmpfs 2022188 148 2022040 1%
/dev/shm tmpfs 2022188 1404 2020784 1% /run /dev/sda6 487652 168276
289680 37% /boot
```

6. ping

It is used verify that a device can communicate with another on network.

PING stands for Packet Internet Groper. synopsis- ping [options]

```
[root@localhost ~]# ping 172.16.4.1
```

```
PING 172.16.4.1 (172.16.4.1) 56(84) bytes of data.
```

```
64 bytes from 172.16.4.1: icmp_seq=1 ttl=64 time=0.328 ms
```

```
64 bytes from 172.16.4.1: icmp_seq=2 ttl=64 time=0.228 ms
```

```
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
```

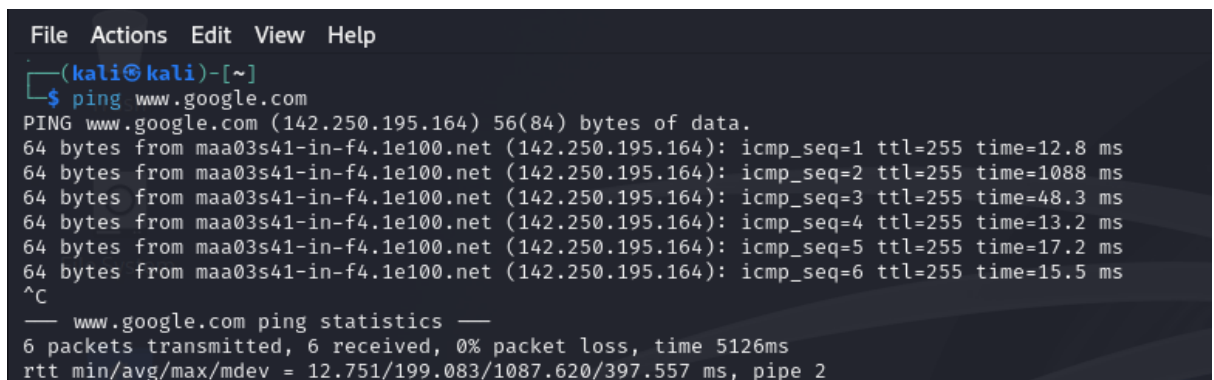
```
^C
```

```
--- 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 5
```



```
File Actions Edit View Help
(kali@kali)-[~]
$ ping www.google.com
PING www.google.com (142.250.195.164) 56(84) bytes of data.
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=1 ttl=255 time=12.8 ms
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=2 ttl=255 time=1088 ms
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=3 ttl=255 time=48.3 ms
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=4 ttl=255 time=13.2 ms
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=5 ttl=255 time=17.2 ms
64 bytes from maa03s41-in-f4.1e100.net (142.250.195.164): icmp_seq=6 ttl=255 time=15.5 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5126ms
rtt min/avg/max/mdev = 12.751/199.083/1087.620/397.557 ms, pipe 2
```

7. ifconfig

It is used configure network interface.

synopsis- ifconfig [options]

example

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

```
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu
1500  inet 172.16.6.102 netmask 255.255.252.0 broadcast
172.16.7.255  inet6 fe80::4a0f:cfff:fe6d:6057 prefixlen 64
scopeid 0x20<link>  ether 48:0f:cf:6d:60:57 txqueuelen 1000
(Ethernet)
```

```
RX packets 23216 bytes 2483338 (2.3 MiB)
```

```
RX errors 0 dropped 5 overruns 0 frame 0
```

```
TX packets 1077 bytes 107740 (105.2 KiB)
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



```

File Actions Edit View Help
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fd00::4b90:626b:31ca:e89f prefixlen 64 scopeid 0<global>
    inet6 fe80::2ad6:e904:a0c0:bb prefixlen 64 scopeid 0<link>
    ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
    RX packets 55431 bytes 78308301 (74.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4084 bytes 257404 (251.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    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) 6.185 ms
6.203 ms 6.189 ms

```

```

(student@kali)-[~]
$ traceroute www.google.com
traceroute to www.google.com (142.250.196.164), 30 hops max, 60 byte packets
1 _gateway (172.16.52.1) 0.226 ms 0.209 ms 0.201 ms
2 115.245.95.249 (115.245.95.249) 1.421 ms 1.407 ms 1.393 ms
3 * * *
4 172.16.12.64 (172.16.12.64) 5.088 ms 172.16.12.122 (172.16.12.122) 4.219 ms 172.16.12.64 (172.16.12.64) 5.413 ms
5 172.16.12.64 (172.16.12.64) 3.819 ms 3.290 ms 172.16.12.122 (172.16.12.122) 4.677 ms
6 72.14.217.252 (72.14.217.252) 3.969 ms 3.752 ms 4.231 ms
7 * * *
8 142.251.55.204 (142.251.55.204) 3.507 ms 142.250.224.6 (142.250.224.6) 3.441 ms 142.251.55.238 (142.251.55.238) 5.471 ms
9 172.253.75.14 (172.253.75.14) 7.735 ms 142.251.230.90 (142.251.230.90) 4.415 ms 108.170.231.129 (108.170.231.129) 3.342 ms
10 maa03s47-in-f4.1e100.net (142.250.196.164) 3.411 ms 142.250.208.153 (142.250.208.153) 4.849 ms 142.251.50.59 (142.251.50.59) 3.382 ms
(student@kali)-[~]

```

Result:

Thus,the basic linux command has been successfully verified.

NAME:PRIYANKA E

ROLL NO:231901038

CSE(CYBER SECURITY)