Basic Linux Terminal Commands

S.No.	Linux Commands	Functions
1	**Is**	Displays information about files in the current directory.
2	**pwd**	Displays the current working directory.
3	**mkdir**	Creates a directory.
4	**cd**	Changes the working directory
5	**rmdir**	Removes empty directories from the directory lists.
6	**cp**	Moves files from one directory to another.
7	**mv**	Rename and Replace the files
8	**rm**	Delete files
9	**uname**	Command to get basic information about the OS
10	**locate**	Find a file in the database.
11	**touch**	Create empty files
12	**ln**	Create shortcuts to other files
13	**cat**	Display file contents on terminal
14	**clear**	Clear terminal
15	**ps**	Display the processes in terminal
16	**man**	Access manual for all Linux commands
17	**grep**	Search for a specific string in an output
18	**echo**	Display active processes on the terminal
19	**wget**	download files from the internet.
20	**whoami**	Create or update passwords for existing users
21	**sort**	sort the file content
22	**cal**	View Calendar in terminal
23	**whereis**	View the exact location of any command typed after this command
24	**df**	Check the details of the file system
25	**wc**	Check the lines, word count, and characters in a file using different options

Lab (To be implemented in C)

1.Basic Commands in Linux Operating System

2.Write a Shell program to check the given number is even or odd

3.Write a Shell program to swap the two integers

4.Write a Shell program to find the factorial of a number

5.write a shell program to generate fibonacci series

6.Write a C program using the following system calls (fork, exec).

7.Write a C program using the following system calls (get_pid, exit).

8.Write a C program using the I/O system calls (open, read, write, etc)

9.Write a C program to simulate CPU scheduling algorithms: FCFS, SJF, and Round Robin

10.Write a C programs to simulate Page Replacement Algorithms a) FIFO b) LRU

11. Implementation of memory allocation algorithms: a)First Fit b) Best Fit c)Worst Fit)

12.Implement the Producer – Consumer problem using semaphores

13.Implement the Dining Philospher problem using semaphores

14.Write a C programs to simulate Intra & Inter – Process Communication (IPC) techniques.

15. Write a C program to simulate Bankers Algorithm for Deadlock Detection

16. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance

17. Simulate all File Organization Techniques a) Single level directory b) Two level

18. Simulate all file allocation strategies a) Sequential b) Indexed c) Linked.

19. Write a C programs to simulate implementation of FCFS Disk Scheduling Algorithms.

20.Write a C programs to simulate implementation SSTF Disk Scheduling Algorithm.

The *head* Command

head [OPTIONS] FILES

$ head filename.txt

It will give first ten lines of the input file

Output a Specific Number of Lines

For example, if we want to have the first five lines printed to standard out, we'd use *-n 5*:

$ head -n 5 filename.txt

4. The tail Command

tail [OPTIONS] FILES

The tail command will by default write the last ten lines of the input file

example shows how to get the **last seven lines from the input file**:

$ tail -n 7 numbers_en.txtS