Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Lab report: 1

[Code no: COMP 307]

Submitted by:

Pratistha Thapa

Roll No: 57

Level: UNG CS (III/II)

Submitted to:

Rabina Shrestha

Department of Computer Science and Engineering

Submission Date: 10/12/2025

1.INTRODUCTION

1.1 What is Linux?

Linux is an operating system that is free to use and open-source, meaning anyone can view or modify its code. It is built from the Unix system and is known for being stable, secure, and flexible. Because of these features, Linux is widely used in servers, networking devices, supercomputers, and many everyday systems like phones and smart appliances.

1.2 The Linux Hierarchical File System

Linux organizes all files and folders in a tree-like structure that begins at the **root directory (/)**. Every item on the system branches out from this root.

Common directories include:

- **/home** – stores personal files for each user
- **/bin** – contains basic commands
- **/etc** – holds system configuration files
- **/var** – keeps logs and files that change often
- **/usr** – includes applications and libraries
- **/tmp** – used for temporary files

This structure helps keep the system organized and separates user files from system files for security and management.

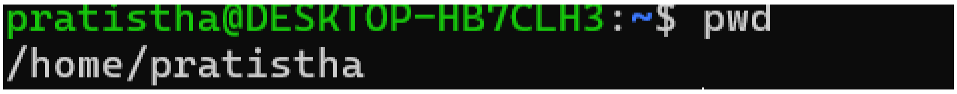
1.3 Importance of Linux Commands in Operating Systems

Linux commands provide a direct way to control the operating system through the terminal. They allow users to move through folders, manage files, view system information, monitor running programs, and perform administrative tasks. Using commands is often faster and more powerful than using a graphical interface, and it allows automation and better control over the system.

2. Linux Commands

1. pwd — Print Working Directory

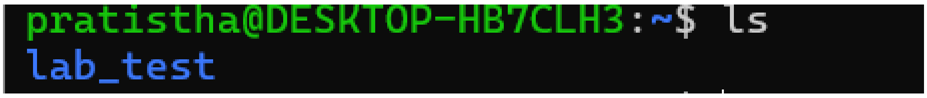
The pwd command displays the absolute path of the current working directory you are in. Linux uses a hierarchical file system, so knowing your exact location is important when navigating, creating files, or running scripts. The output always starts from the root directory `/`, showing the full path that leads to your current folder.



```
pratistha@DESKTOP-HB7CLH3:~$ pwd
/home/pratistha
```

2. ls — List Directory Contents

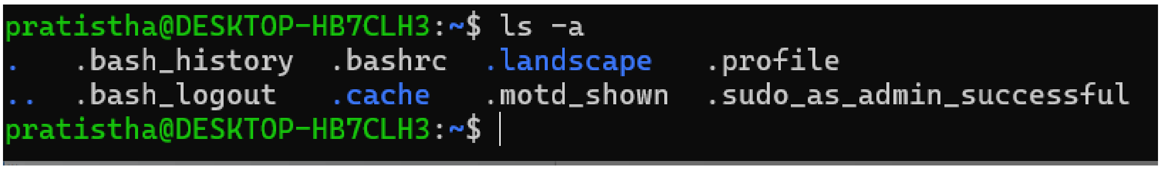
The `ls` command lists all files and directories inside the current location. It helps users quickly see what items exist in a directory. By default, `ls` only shows visible files, arranged in alphabetical order. It is one of the most frequently used commands when working in the terminal.



```
pratistha@DESKTOP-HB7CLH3:~$ ls
lab_test
```

3. ls -a — List All Files Including Hidden Ones

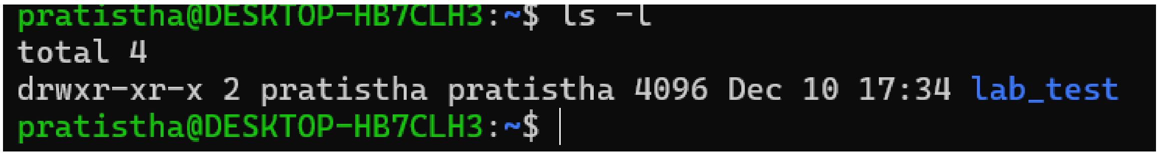
Linux uses hidden files (those beginning with a dot `.`) to store configuration settings. The `ls -a` command shows **all** files, including hidden system files and folders. This is useful when troubleshooting configurations, viewing application settings, or working with files that do not appear in a normal directory listing.



```
pratistha@DESKTOP-HB7CLH3:~$ ls -a
.    .bash_history  .bashrc   .landscape   .profile
..   .bash_logout   .cache    .motd_shown  .sudo_as_admin_successful
pratistha@DESKTOP-HB7CLH3:~$
```

4. ls -l — Long Listing Format

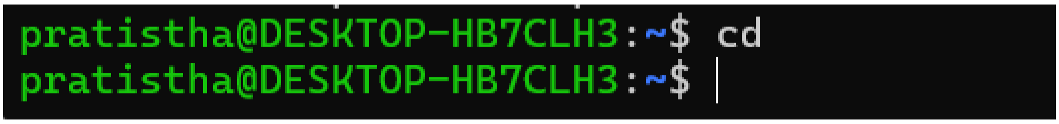
The `ls -l` command provides a detailed listing of directory contents. It displays file permissions, number of links, owner and group name, file size, and date of last modification. This format helps users understand access rights, track recent changes, and manage files more precisely.



```
pratistha@DESKTOP-HB7CLH3:~$ ls -l
total 4
drwxr-xr-x 2 pratistha pratistha 4096 Dec 10 17:34 lab_test
pratistha@DESKTOP-HB7CLH3:~$
```

5. cd — Change Directory

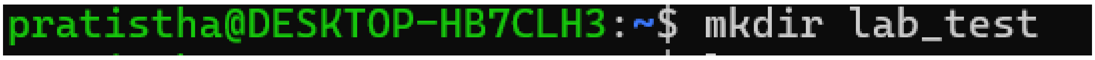
The `cd` command is used to move between directories in Linux. It allows users to navigate the file system easily. Learning to use `cd` efficiently makes file navigation much faster.



```
pratistha@DESKTOP-HB7CLH3:~$ cd
pratistha@DESKTOP-HB7CLH3:~$
```

6. mkdir — Make Directory

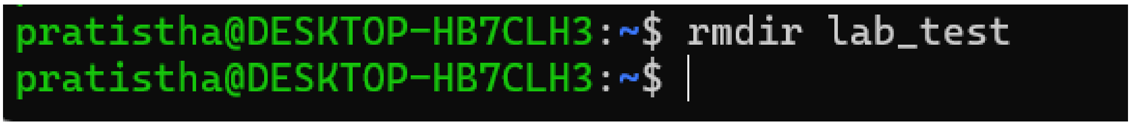
`mkdir` creates a new directory (folder) in the current location. It is commonly used to organize files into separate categories or project folders. You can create one folder or multiple folders at a time. If the directory already exists, Linux will show an error unless additional flags are used.



```
pratistha@DESKTOP-HB7CLH3:~$ mkdir lab_test
```

7. rmdir — Remove Empty Directory

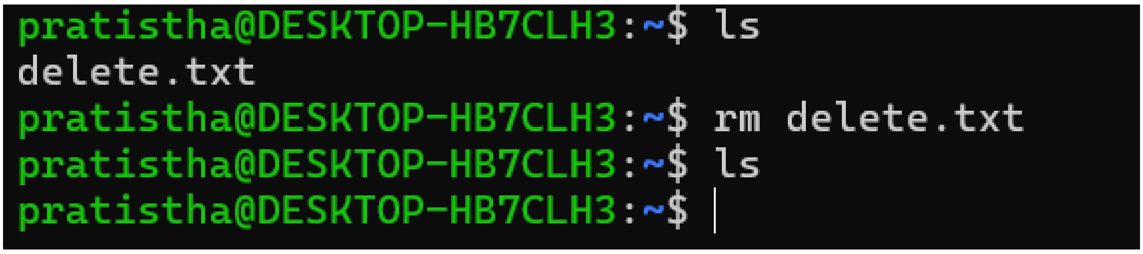
The `rmdir` command deletes directories that are completely empty. It will not work if the folder contains any files or other subdirectories. This prevents accidental loss of data. If you want to delete a directory with contents, you must use `rm -r` instead.



```
pratistha@DESKTOP-HB7CLH3:~$ rmdir lab_test
pratistha@DESKTOP-HB7CLH3:~$
```

8. rm — Remove Files

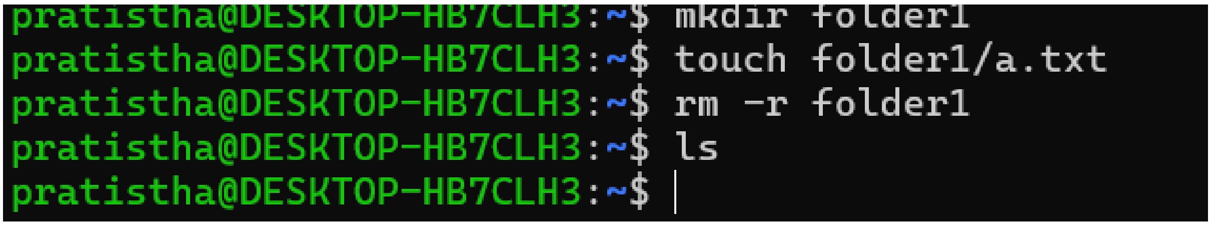
The `rm` command permanently deletes files from the system. Once removed with `rm`, files cannot be recovered using normal tools. It is used for removing unwanted data or clearing space. Care must be taken because Linux does not send deleted files to a recycle bin.



```
pratistha@DESKTOP-HB7CLH3:~$ ls
delete.txt
pratistha@DESKTOP-HB7CLH3:~$ rm delete.txt
pratistha@DESKTOP-HB7CLH3:~$ ls
pratistha@DESKTOP-HB7CLH3:~$
```

9. rm -r — Remove Directories Recursively

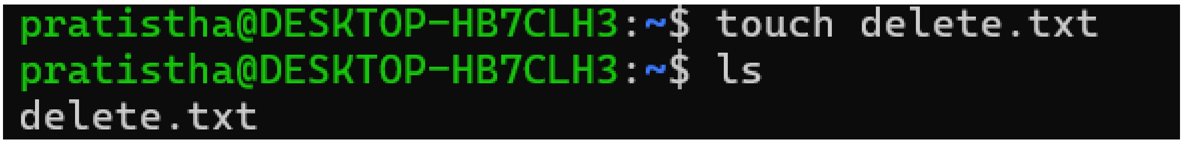
The `rm -r` command deletes a directory **along with all of its files and subfolders**. It performs a recursive deletion, meaning it goes inside the folder and removes everything inside it. This is a powerful command and should be used carefully to avoid accidental deletion of important files.



```
pratistha@DESKTOP-HB7CLH3:~$ mkdir folder1
pratistha@DESKTOP-HB7CLH3:~$ touch folder1/a.txt
pratistha@DESKTOP-HB7CLH3:~$ rm -r folder1
pratistha@DESKTOP-HB7CLH3:~$ ls
pratistha@DESKTOP-HB7CLH3:~$
```

10. touch — Create an Empty File

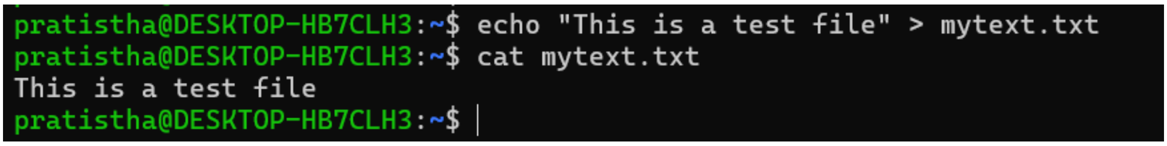
The `touch` command creates a new empty file if it does not exist. If the file already exists, `touch` updates the file's timestamp without changing its contents. It is useful for quickly creating placeholder files, logs, or testing file operations.



```
pratistha@DESKTOP-HB7CLH3:~$ touch delete.txt
pratistha@DESKTOP-HB7CLH3:~$ ls
delete.txt
```

11. cat — View File Content

The `cat` command displays the contents of a file in the terminal. It is often used for reading small files, combining multiple files, or checking data inside text documents. `cat` can also be used to create files using redirection operators (> or >>).



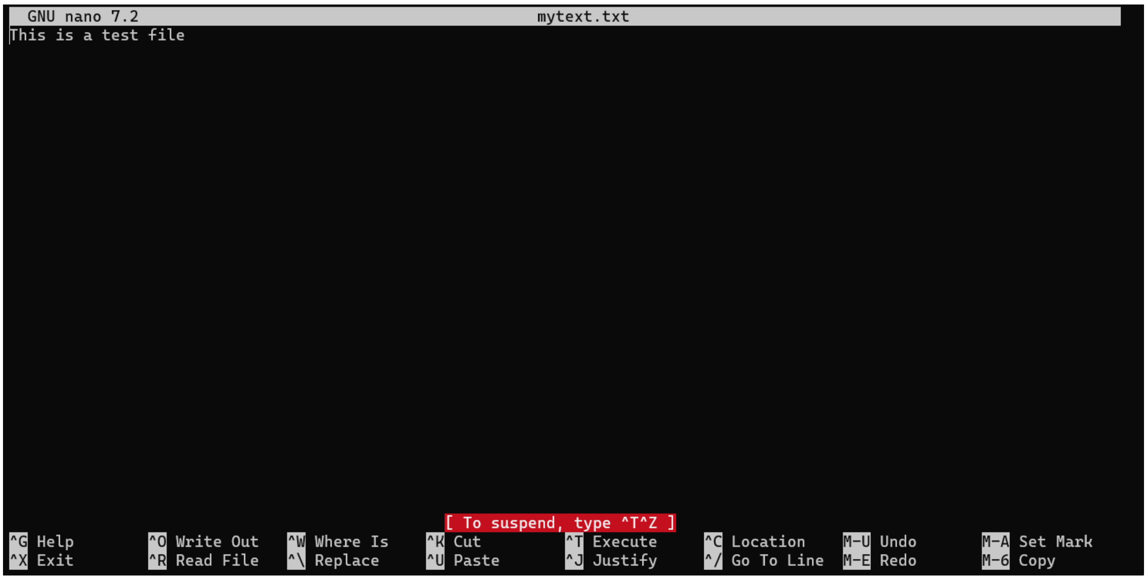
```
pratistha@DESKTOP-HB7CLH3:~$ echo "This is a test file" > mytext.txt
pratistha@DESKTOP-HB7CLH3:~$ cat mytext.txt
This is a test file
pratistha@DESKTOP-HB7CLH3:~$ |
```

12. nano, vi, jed — Terminal Text Editors

These commands open text editors inside the terminal.

- `nano` — a beginner-friendly editor with on-screen instructions.
- `vi` / `vim` — a powerful, mode-based editor used by professionals.
- `jed` — a lightweight editor available on some systems.

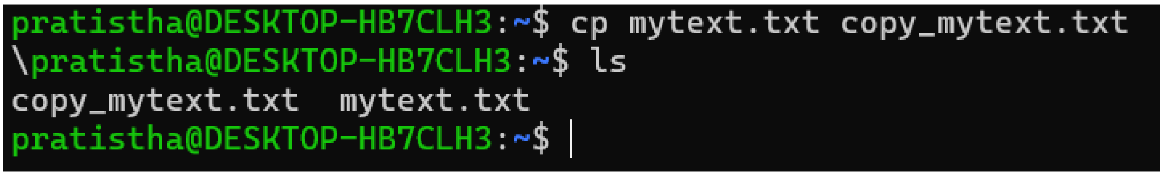
They allow users to write, edit, and modify files directly from the command line, which is essential for system configuration and scripting.



```
  GNU nano 7.2                                   mytext.txt
This is a test file










                                     [ To suspend, type ^T^Z ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

13. cp — Copy Files or Directories

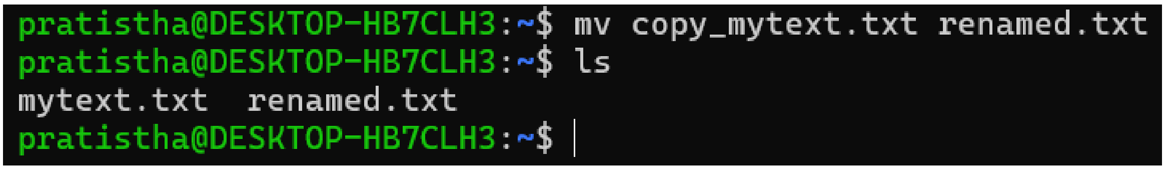
The `cp` command copies a file or folder from one place to another. It is used to duplicate important files, create backups, or organize data.
Example: `cp file1.txt file2.txt` creates a copy of file1 as file2.
You can also copy entire directories using the `-r` option.



```
pratistha@DESKTOP-HB7CLH3:~$ cp mytext.txt copy_mytext.txt
\pratistha@DESKTOP-HB7CLH3:~$ ls
copy_mytext.txt  mytext.txt
pratistha@DESKTOP-HB7CLH3:~$
```

14. mv — Move or Rename Files

The `mv` command either moves a file to another directory or renames it. Unlike copying, it removes the file from the original location.It is commonly used when organizing files or updating their names.



```
pratistha@DESKTOP-HB7CLH3:~$ mv copy_mytext.txt renamed.txt
pratistha@DESKTOP-HB7CLH3:~$ ls
mytext.txt  renamed.txt
pratistha@DESKTOP-HB7CLH3:~$
```

15. locate — Search for Files

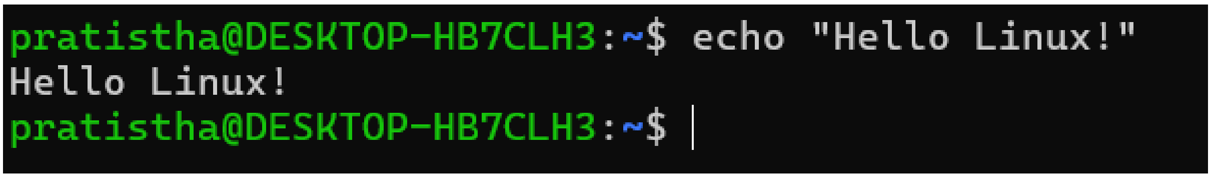
`locate` quickly finds files by name using a prebuilt database of file locations. It is much faster than searching manually.

Before using it, the database must be updated once using `sudo updatedb`.

Example: `locate sample.txt` lists all paths where a file with that name exists.

16. echo — Display Text or Write to Files

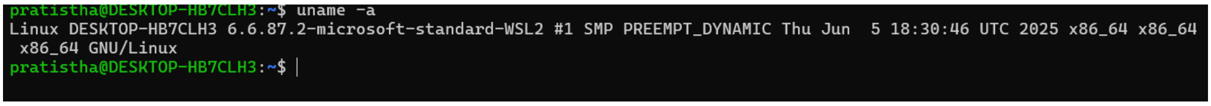
The `echo` command prints text to the terminal. It can also be used with redirection (`>` or `>>`) to write data into files. It is useful for scripting, printing messages, and generating quick text files.



```
pratistha@DESKTOP-HB7CLH3:~$ echo "Hello Linux!"
Hello Linux!
pratistha@DESKTOP-HB7CLH3:~$
```

17. uname -a — System Information

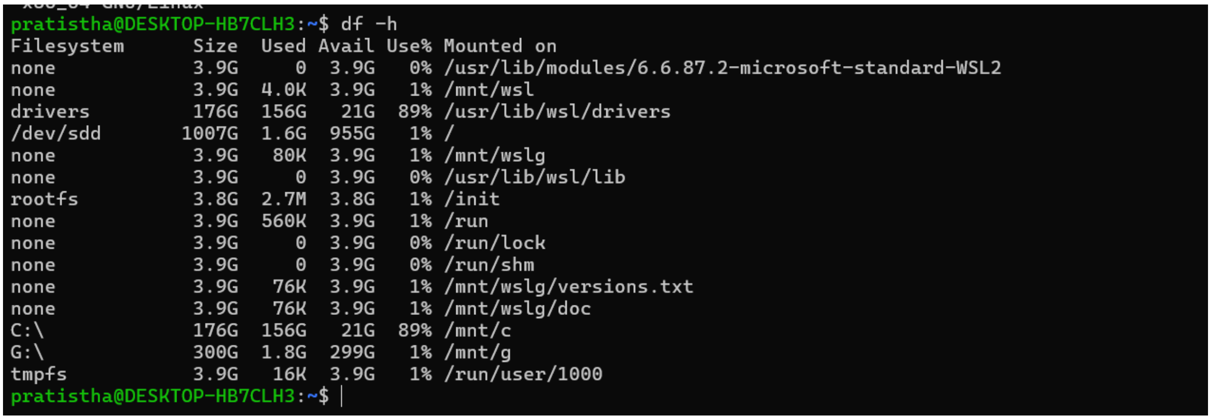
This command displays comprehensive system details, including the kernel version, OS type, architecture, and hostname. It is useful for identifying the system specifications and diagnosing compatibility issues.



```
pratistha@DESKTOP-HB7CLH3:~$ uname -a
Linux DESKTOP-HB7CLH3 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 2025 x86_64 x86_64
 x86_64 GNU/Linux
pratistha@DESKTOP-HB7CLH3:~$ 
```

18. df -h — Disk Space Usage

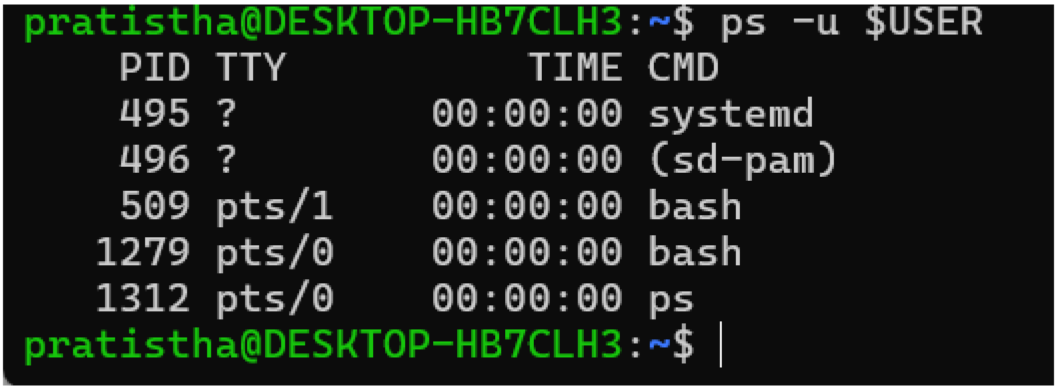
The `df -h` command shows disk space usage in a human-readable (GB/MB) format. It lets users see how much storage is available, how much is used, and where different partitions are mounted. System administrators use it to monitor and manage system storage.



```
 x86_64 GNU/Linux
pratistha@DESKTOP-HB7CLH3:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.9G     0  3.9G   0% /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2
none            3.9G  4.0K  3.9G   1% /mnt/wsl
drivers         176G  156G   21G  89% /usr/lib/wsl/drivers
/dev/sdd       1007G  1.6G  955G   1% /
none            3.9G   80K  3.9G   1% /mnt/wslg
none            3.9G     0  3.9G   0% /usr/lib/wsl/lib
rootfs          3.8G  2.7M  3.8G   1% /init
none            3.9G  560K  3.9G   1% /run
none            3.9G     0  3.9G   0% /run/lock
none            3.9G     0  3.9G   0% /run/shm
none            3.9G   76K  3.9G   1% /mnt/wslg/versions.txt
none            3.9G   76K  3.9G   1% /mnt/wslg/doc
C:\             176G  156G   21G  89% /mnt/c
G:\             300G  1.8G  299G   1% /mnt/g
tmpfs           3.9G   16K  3.9G   1% /run/user/1000
pratistha@DESKTOP-HB7CLH3:~$ 
```

19. ps -u $USER — Processes of Current User

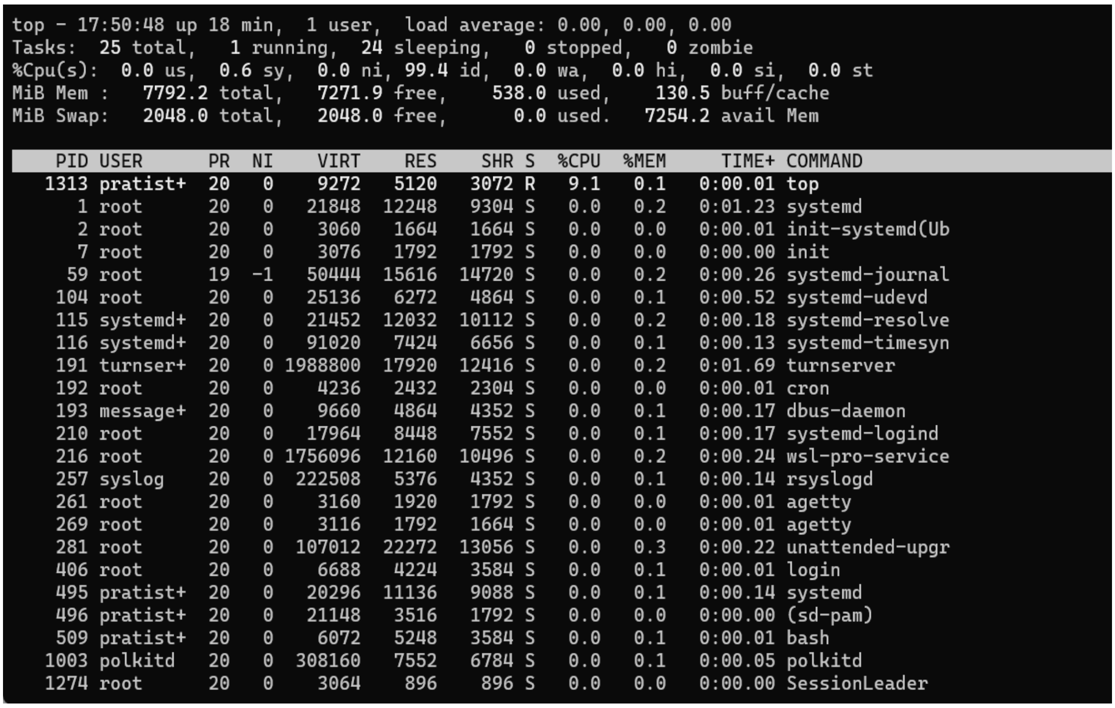
This command displays all active processes running under the current user account. It shows details like process ID, CPU usage, and the command used to start the process.
It is useful for monitoring tasks and identifying programs consuming system resources.



```
pratistha@DESKTOP-HB7CLH3:~$ ps -u $USER
    PID TTY          TIME CMD
    495 ?        00:00:00 systemd
    496 ?        00:00:00 (sd-pam)
    509 pts/1    00:00:00 bash
   1279 pts/0    00:00:00 bash
   1312 pts/0    00:00:00 ps
pratistha@DESKTOP-HB7CLH3:~$ 
```

20. top — Real-Time Process Viewer

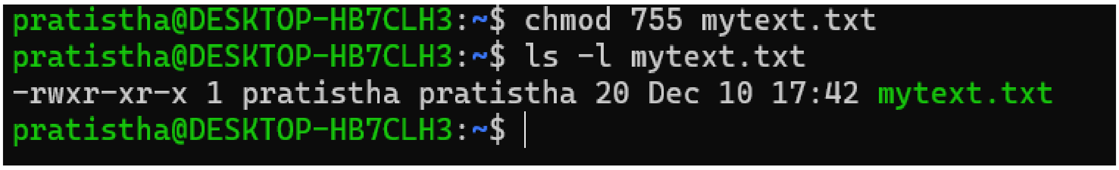
The `top` command shows real-time system activity, similar to Task Manager on Windows. It displays CPU usage, memory usage, running processes, system load, and more.
It refreshes automatically every few seconds and helps users understand system performance.



```
top - 17:50:48 up 18 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  25 total,   1 running,  24 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.6 sy,  0.0 ni, 99.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7792.2 total,   7271.9 free,    538.0 used,    130.5 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   7254.2 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1313 pratist+  20   0    9272   5120   3072 R   9.1   0.1   0:00.01 top
      1 root      20   0   21848  12248   9304 S   0.0   0.2   0:01.23 systemd
      2 root      20   0    3060   1664   1664 S   0.0   0.0   0:00.01 init-systemd(Ub
      7 root      20   0    3076   1792   1792 S   0.0   0.0   0:00.00 init
     59 root      19  -1   50444  15616  14720 S   0.0   0.2   0:00.26 systemd-journal
    104 root      20   0   25136   6272   4864 S   0.0   0.1   0:00.52 systemd-udevd
    115 systemd+  20   0   21452  12032  10112 S   0.0   0.2   0:00.18 systemd-resolve
    116 systemd+  20   0   91020   7424   6656 S   0.0   0.1   0:00.13 systemd-timesyn
    191 turnser+  20   0 1988800  17920  12416 S   0.0   0.2   0:01.69 turnserver
    192 root      20   0    4236   2432   2304 S   0.0   0.0   0:00.01 cron
    193 message+  20   0    9660   4864   4352 S   0.0   0.1   0:00.17 dbus-daemon
    210 root      20   0   17964   8448   7552 S   0.0   0.1   0:00.17 systemd-logind
    216 root      20   0 1756096  12160  10496 S   0.0   0.2   0:00.24 wsl-pro-service
    257 syslog    20   0  222508   5376   4352 S   0.0   0.1   0:00.14 rsyslogd
    261 root      20   0    3160   1920   1792 S   0.0   0.0   0:00.01 agetty
    269 root      20   0    3116   1792   1664 S   0.0   0.0   0:00.01 agetty
    281 root      20   0  107012  22272  13056 S   0.0   0.3   0:00.22 unattended-upgr
    406 root      20   0    6688   4224   3584 S   0.0   0.1   0:00.01 login
    495 pratist+  20   0   20296  11136   9088 S   0.0   0.1   0:00.14 systemd
    496 pratist+  20   0   21148   3516   1792 S   0.0   0.0   0:00.00 (sd-pam)
    509 pratist+  20   0    6072   5248   3584 S   0.0   0.1   0:00.01 bash
   1003 polkitd   20   0  308160   7552   6784 S   0.0   0.1   0:00.05 polkitd
   1274 root      20   0    3064    896    896 S   0.0   0.0   0:00.00 SessionLeader
```

21. chmod — Change File Permissions

The `chmod` command modifies the read, write, and execute permissions of files and directories. This grants full permission to the owner and read+execute permissions to others. It is an essential command for managing security and access control in Linux.



```
pratistha@DESKTOP-HB7CLH3:~$ chmod 755 mytext.txt
pratistha@DESKTOP-HB7CLH3:~$ ls -l mytext.txt
-rwxr-xr-x 1 pratistha pratistha 20 Dec 10 17:42 mytext.txt
pratistha@DESKTOP-HB7CLH3:~$ 
```