Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavrepalanchowk

Lab-I

Submitted by:
Saksham Humagain (25)

Submitted to:
Mrs. Rabina Shrestha
Department of Computer Science and Engineering

Submission Date: December 09, 2025

INTRODUCTION

1. **What is Linux?**

Linux is an open-source, Unix-like operating system kernel that was created by Linus Torvalds in 1991. It forms the core of a wide range of operating systems, collectively called Linux distributions (like Ubuntu, Fedora, and CentOS). Linux is free to use, modify, and distribute, making it highly popular in servers, desktops, mobile devices, and embedded systems. It is known for its stability, security, multi-user capabilities, and flexibility

2. **The Linux Hierarchical File System.**

Linux uses a hierarchical file system, which means files are organized in a tree-like structure starting from a single root directory, represented by /. All files and directories in Linux are placed under this root, unlike Windows, which uses separate drives (C:, D:, etc.).

Key Features of Linux File System

1. Root Directory (/) – The topmost directory from which all other directories branch out.

2. Directories (Folders) – Containers for files and other directories.

3. Files – Can be regular files (text, data), directories, symbolic links, or device files.

4. Pathnames – The location of a file is specified by its pathname, which can be:

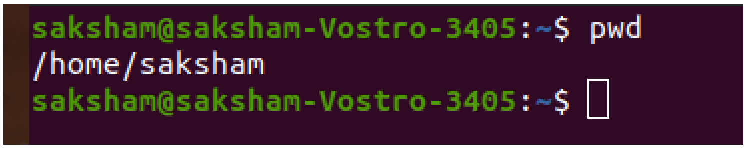
    ○ Absolute Path: Starts from /, e.g., /home/user/Documents/file.txt

  - Relative Path: Starts from the current working directory, e.g., Documents/file.txt

3. Importance of Linux commands in Operating Systems.
Linux commands are essential in operating systems because they allow users and administrators to efficiently interact with and manage the system through the command-line interface. They enable tasks such as navigating the file system, creating, copying, moving, and deleting files and directories, managing user permissions, monitoring system resources, installing and updating software, and troubleshooting network issues. Additionally, Linux commands can be combined in scripts to automate repetitive tasks, saving time and reducing errors. Mastery of these commands provides greater control, speed, and precision compared to graphical interfaces, making them crucial for effective system administration, software development, and overall Linux system management.

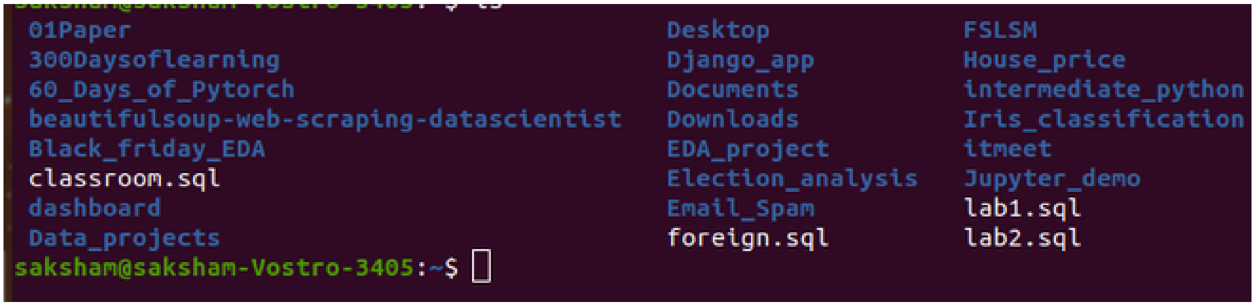
COMMANDS

1. pwd — When you open the terminal, you start in your user's home directory. The `pwd` command shows the absolute path of your current directory, starting from the root directory `/`. For example, your home directory might be `/home/username`.



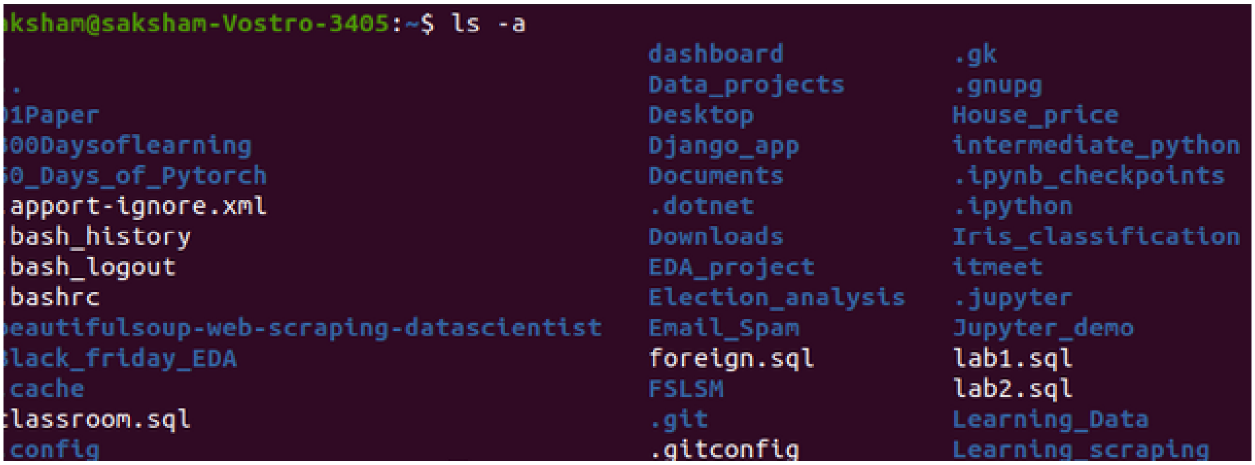
```
saksham@saksham-Vostro-3405:~$ pwd
/home/saksham
saksham@saksham-Vostro-3405:~$ 
```

2. ls – The `ls` command lists all the files and folders in your current directory. It helps you see what items are present in that location, and with options like `ls -a` or `ls -l`, you can view hidden files or detailed information about each item.



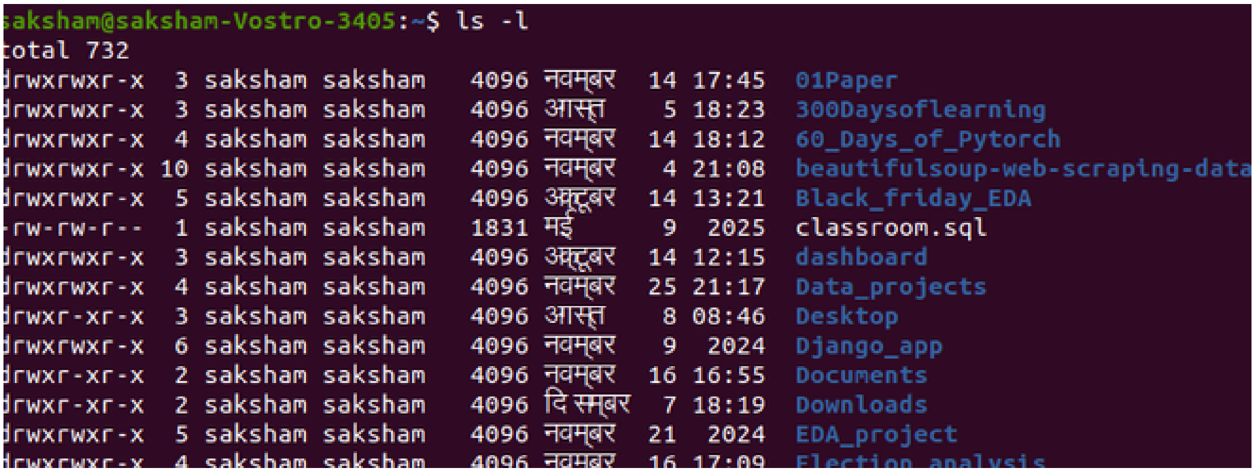
```
saksham@saksham-Vostro-3405:~$ ls
 01Paper                                   Desktop             FSLSM
 300Daysoflearning                         Django_app          House_price
 60_Days_of_Pytorch                        Documents           intermediate_python
 beautifulsoup-web-scraping-datascientist  Downloads           Iris_classification
 Black_friday_EDA                          EDA_project         itmeet
 classroom.sql                             Election_analysis   Jupyter_demo
 dashboard                                 Email_Spam          lab1.sql
 Data_projects                             foreign.sql         lab2.sql
saksham@saksham-Vostro-3405:~$ 
```

3. ls -a – The `ls -a` command displays all files and directories in the current location, including hidden files. Hidden files in Linux start with a dot (`.`), such as `.bashrc` or `.config`, and `ls` normally doesn't show them unless `-a` is used.



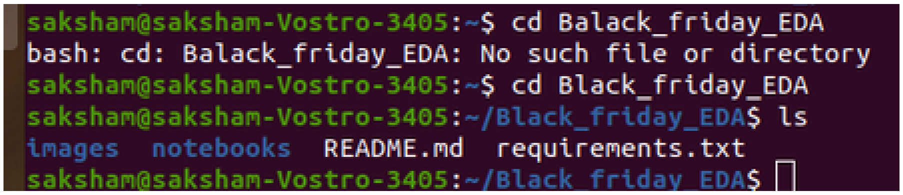
```
aksham@saksham-Vostro-3405:~$ ls -a
.                                         dashboard           .gk
..                                        Data_projects       .gnupg
01Paper                                   Desktop             House_price
300Daysoflearning                         Django_app          intermediate_python
60_Days_of_Pytorch                        Documents           .ipynb_checkpoints
.apport-ignore.xml                        .dotnet             .ipython
.bash_history                             Downloads           Iris_classification
.bash_logout                              EDA_project         itmeet
.bashrc                                   Election_analysis   .jupyter
beautifulsoup-web-scraping-datascientist  Email_Spam          Jupyter_demo
Black_friday_EDA                          foreign.sql         lab1.sql
.cache                                    FSLSM               lab2.sql
classroom.sql                             .git                Learning_Data
.config                                   .gitconfig          Learning_scraping
```

4. ls -l – The `ls -l` command shows files and directories in long listing format, displaying detailed information such as permissions, number of links, owner, group, file size, and the date and time they were last modified.



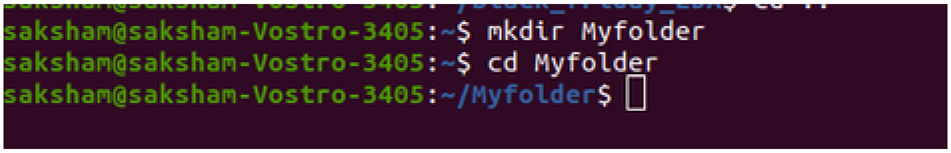
```
saksham@saksham-Vostro-3405:~$ ls -l
total 732
drwxrwxr-x  3 saksham saksham   4096 नवम्बर  14 17:45  01Paper
drwxrwxr-x  3 saksham saksham   4096 अगस्त    5 18:23  300Daysoflearning
drwxrwxr-x  4 saksham saksham   4096 नवम्बर  14 18:12  60_Days_of_Pytorch
drwxrwxr-x 10 saksham saksham   4096 नवम्बर   4 21:08  beautifulsoup-web-scraping-data
drwxrwxr-x  5 saksham saksham   4096 अक्टूबर  14 13:21  Black_friday_EDA
-rw-rw-r--  1 saksham saksham   1831 मई       9  2025  classroom.sql
drwxrwxr-x  3 saksham saksham   4096 अक्टूबर  14 12:15  dashboard
drwxrwxr-x  4 saksham saksham   4096 नवम्बर  25 21:17  Data_projects
drwxr-xr-x  3 saksham saksham   4096 अगस्त    8 08:46  Desktop
drwxrwxr-x  6 saksham saksham   4096 नवम्बर   9  2024  Django_app
drwxr-xr-x  2 saksham saksham   4096 नवम्बर  16 16:55  Documents
drwxr-xr-x  2 saksham saksham   4096 दिसम्बर   7 18:19  Downloads
drwxrwxr-x  5 saksham saksham   4096 नवम्बर  21  2024  EDA_project
drwxrwxr-x  4 saksham saksham   4096 नवम्बर  16 17:09  Election_analysis
```

5. cd – The `cd` command is used to change directories in Linux. For example, `cd Documents` moves you into the Documents folder, `cd ..` takes you one level up to the parent directory, and `cd /home/user` moves you directly to the specified absolute path.



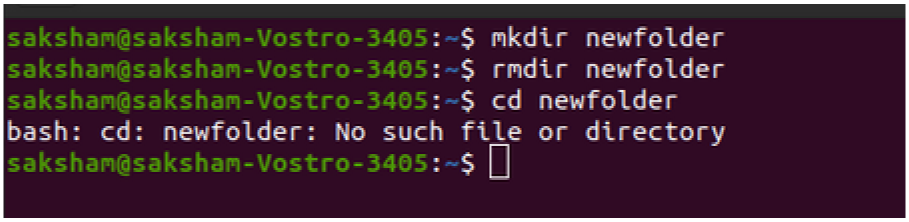
```
saksham@saksham-Vostro-3405:~$ cd Balack_friday_EDA
bash: cd: Balack_friday_EDA: No such file or directory
saksham@saksham-Vostro-3405:~$ cd Black_friday_EDA
saksham@saksham-Vostro-3405:~/Black_friday_EDA$ ls
images  notebooks  README.md  requirements.txt
saksham@saksham-Vostro-3405:~/Black_friday_EDA$
```

6. mkdir – The mkdir command is used to create a new directory. You can create any folder you want by giving it a name.
Example: mkdir myFolder will create a directory named *myFolder* in the current location.



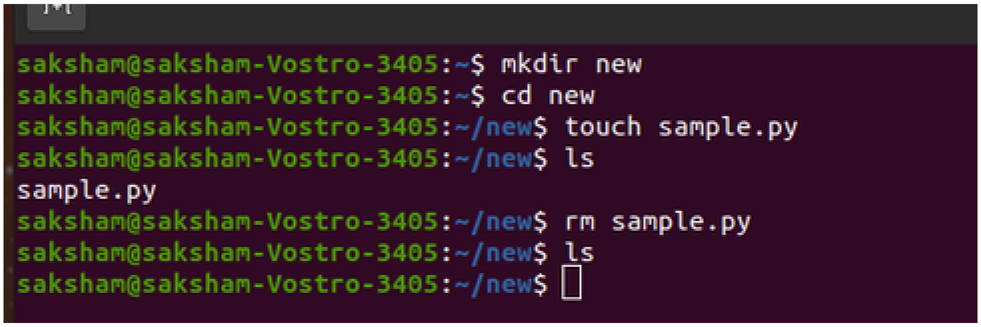
```
saksham@saksham-Vostro-3405:~/Black_friday_EDA$ cd ..
saksham@saksham-Vostro-3405:~$ mkdir Myfolder
saksham@saksham-Vostro-3405:~$ cd Myfolder
saksham@saksham-Vostro-3405:~/Myfolder$
```

7. rmdir – The `rmdir` command is used to delete empty directories. It only works if the folder has no files inside it; otherwise, it will show an error.



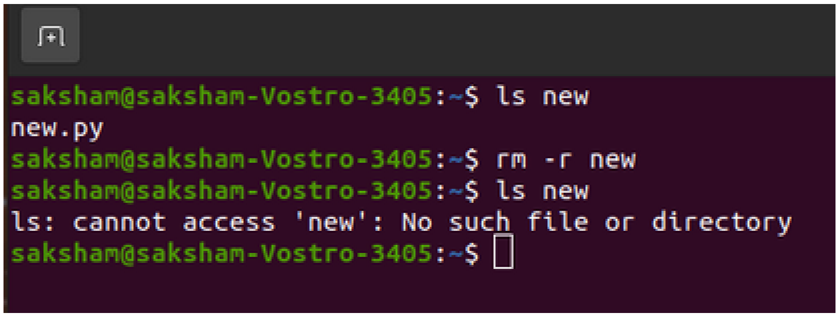
```
saksham@saksham-Vostro-3405:~$ mkdir newfolder
saksham@saksham-Vostro-3405:~$ rmdir newfolder
saksham@saksham-Vostro-3405:~$ cd newfolder
bash: cd: newfolder: No such file or directory
saksham@saksham-Vostro-3405:~$ 
```

8. rm – The `rm` command is used to delete files permanently from the system. For example, `rm file.txt` will remove the file named *file.txt*. Be careful—deleted files cannot be recovered.



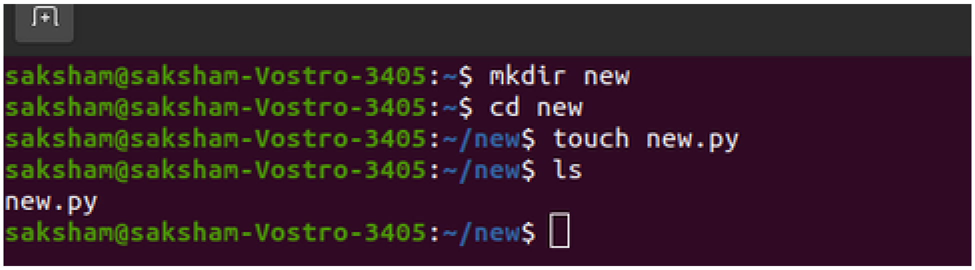
```
saksham@saksham-Vostro-3405:~$ mkdir new
saksham@saksham-Vostro-3405:~$ cd new
saksham@saksham-Vostro-3405:~/new$ touch sample.py
saksham@saksham-Vostro-3405:~/new$ ls
sample.py
saksham@saksham-Vostro-3405:~/new$ rm sample.py
saksham@saksham-Vostro-3405:~/new$ ls
saksham@saksham-Vostro-3405:~/new$ 
```

9. rm -r – The `rm -r` command is used to delete directories along with all the files and subdirectories inside them. It removes everything recursively, so use it carefully because the deletion is permanent.



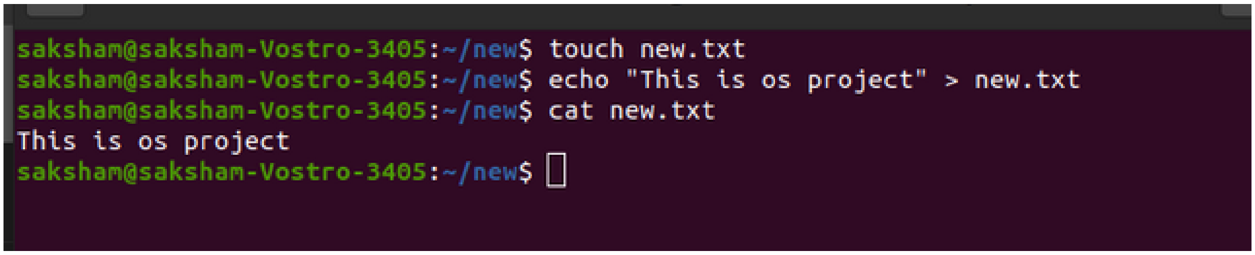
```
saksham@saksham-Vostro-3405:~$ ls new
new.py
saksham@saksham-Vostro-3405:~$ rm -r new
saksham@saksham-Vostro-3405:~$ ls new
ls: cannot access 'new': No such file or directory
saksham@saksham-Vostro-3405:~$ 
```

10. touch – The `touch` command is used to create an empty file or update the timestamp of an existing file. For example, `touch new.txt` creates a file named *new.txt* in the current directory.



```
saksham@saksham-Vostro-3405:~$ mkdir new
saksham@saksham-Vostro-3405:~$ cd new
saksham@saksham-Vostro-3405:~/new$ touch new.py
saksham@saksham-Vostro-3405:~/new$ ls
new.py
saksham@saksham-Vostro-3405:~/new$
```

11. cat – The `cat` command is used to display the contents of a file directly in the terminal. It can also be used to create or join multiple files, but its main use is to read what's inside a file.



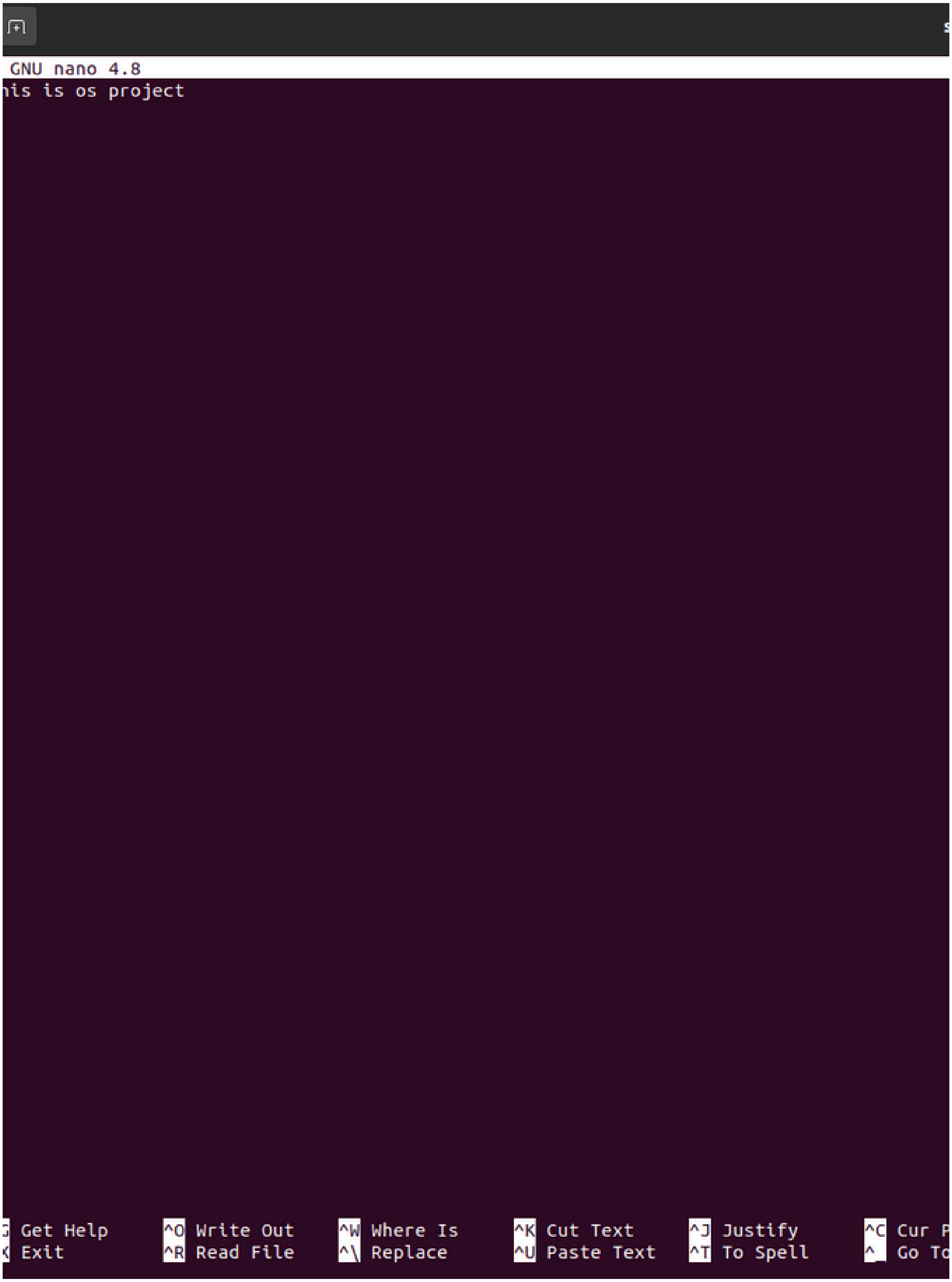
```
saksham@saksham-Vostro-3405:~/new$ touch new.txt
saksham@saksham-Vostro-3405:~/new$ echo "This is os project" > new.txt
saksham@saksham-Vostro-3405:~/new$ cat new.txt
This is os project
saksham@saksham-Vostro-3405:~/new$
```

12. nano, vi, jed – These are **text editors** in Linux used to create or edit files directly from the terminal.

- **nano** – A simple and beginner-friendly text editor.

- **vi** – A powerful editor with modes for inserting and editing text; widely used by advanced users.

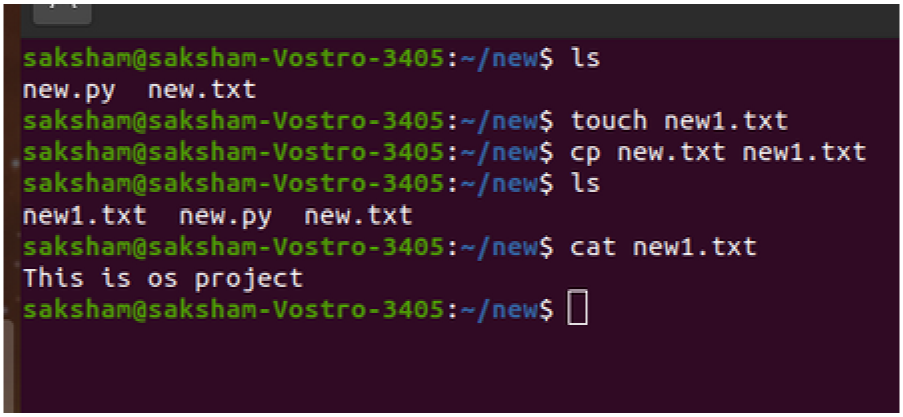
- **jed** – Lightweight editor, similar to Emacs, for editing files in the terminal.



```
 GNU nano 4.8
his is os project
```

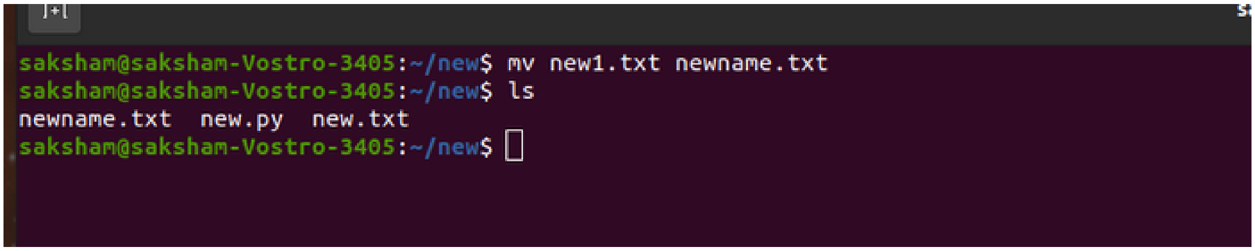
```
G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur P
X Exit        ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To
```

13. cp – The `cp` command is used to copy files or directories. For example, `cp a.txt b.txt` creates a copy of the file *a.txt* and names it *b.txt* in the same directory.



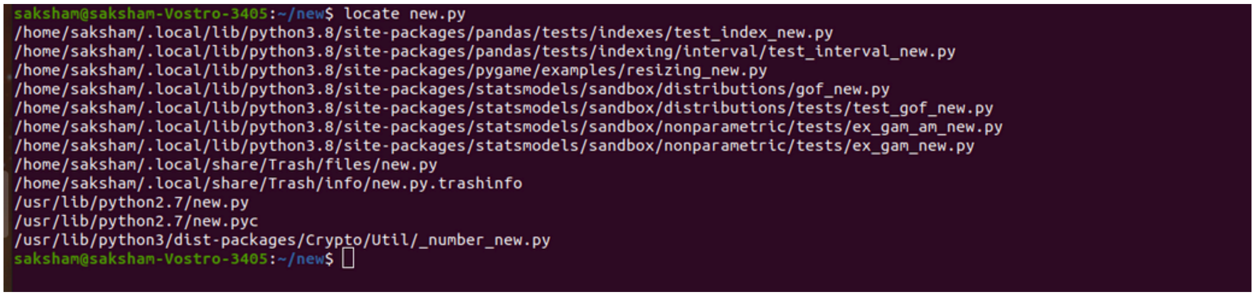
```
saksham@saksham-Vostro-3405:~/new$ ls
new.py  new.txt
saksham@saksham-Vostro-3405:~/new$ touch new1.txt
saksham@saksham-Vostro-3405:~/new$ cp new.txt new1.txt
saksham@saksham-Vostro-3405:~/new$ ls
new1.txt  new.py  new.txt
saksham@saksham-Vostro-3405:~/new$ cat new1.txt
This is os project
saksham@saksham-Vostro-3405:~/new$ 
```

14. mv – The `mv` command is used to move or rename files and directories. For example, `mv oldname.txt newname.txt` renames a file, and `mv file.txt /home/user/Documents/` moves a file to a different directory.



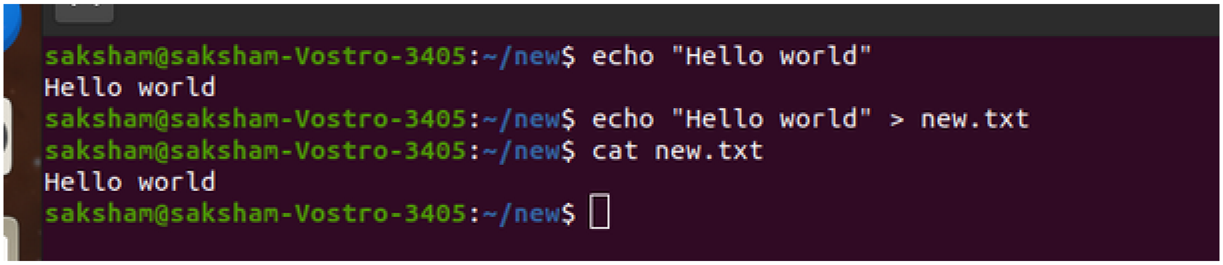
```
saksham@saksham-Vostro-3405:~/new$ mv new1.txt newname.txt
saksham@saksham-Vostro-3405:~/new$ ls
newname.txt  new.py  new.txt
saksham@saksham-Vostro-3405:~/new$ 
```

15. locate – The `locate` command is used to find the path of files and directories quickly in the system by searching a prebuilt database. It is faster than `find`, but the database may need to be updated using `updatedb`.



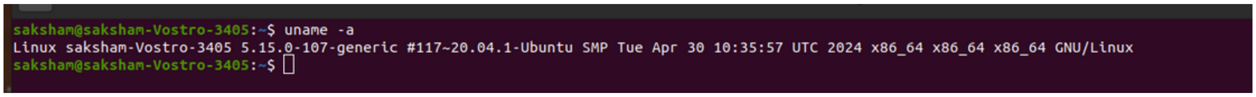
```
saksham@saksham-Vostro-3405:~/new$ locate new.py
/home/saksham/.local/lib/python3.8/site-packages/pandas/tests/indexes/test_index_new.py
/home/saksham/.local/lib/python3.8/site-packages/pandas/tests/indexing/interval/test_interval_new.py
/home/saksham/.local/lib/python3.8/site-packages/pygame/examples/resizing_new.py
/home/saksham/.local/lib/python3.8/site-packages/statsmodels/sandbox/distributions/gof_new.py
/home/saksham/.local/lib/python3.8/site-packages/statsmodels/sandbox/distributions/tests/test_gof_new.py
/home/saksham/.local/lib/python3.8/site-packages/statsmodels/sandbox/nonparametric/tests/ex_gam_am_new.py
/home/saksham/.local/lib/python3.8/site-packages/statsmodels/sandbox/nonparametric/tests/ex_gam_new.py
/home/saksham/.local/share/Trash/files/new.py
/home/saksham/.local/share/Trash/info/new.py.trashinfo
/usr/lib/python2.7/new.py
/usr/lib/python2.7/new.pyc
/usr/lib/python3/dist-packages/Crypto/Util/_number_new.py
saksham@saksham-Vostro-3405:~/new$ 
```

16. echo – The `echo` command is used to display text or variables in the terminal. It is often used in scripts to show messages or output the value of environment variables.



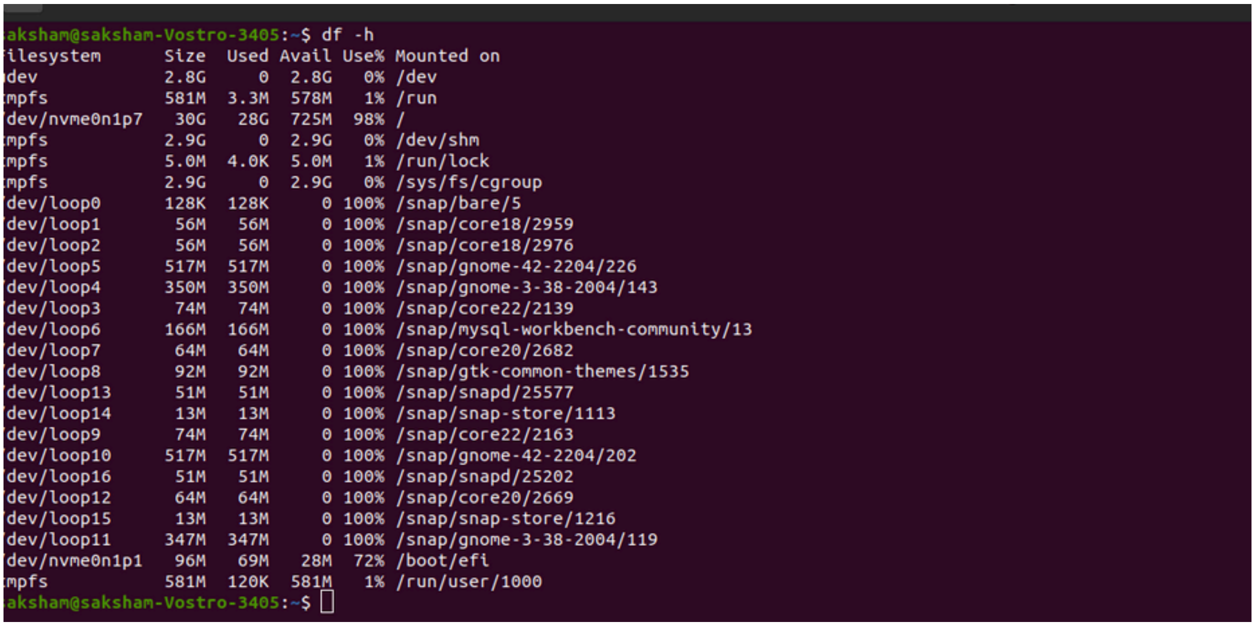
```
saksham@saksham-Vostro-3405:~/new$ echo "Hello world"
Hello world
saksham@saksham-Vostro-3405:~/new$ echo "Hello world" > new.txt
saksham@saksham-Vostro-3405:~/new$ cat new.txt
Hello world
saksham@saksham-Vostro-3405:~/new$ 
```

17. uname -a – The `uname -a` command displays detailed system information, including the kernel version, operating system type, system architecture, hostname, and other relevant details.



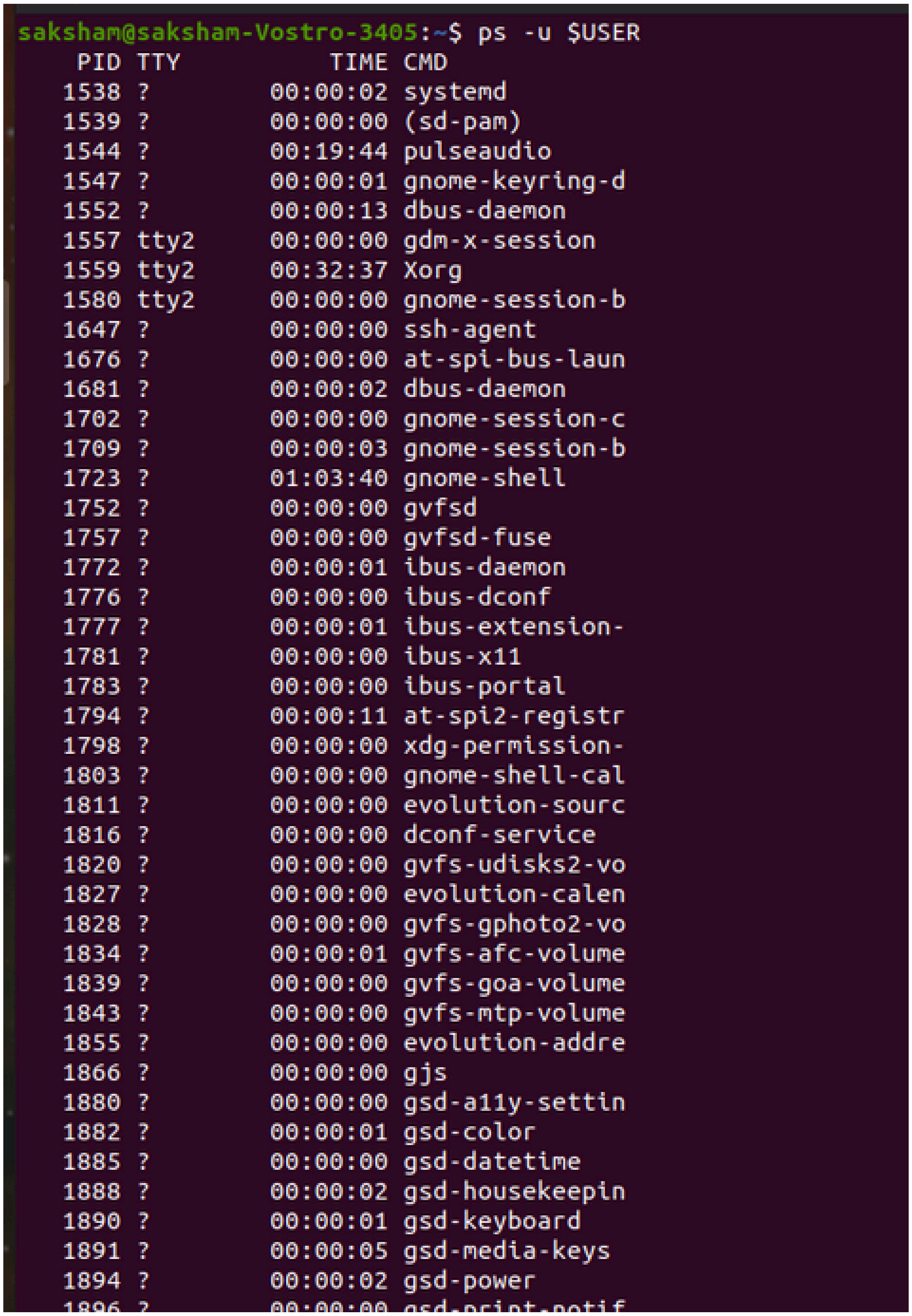
```
saksham@saksham-Vostro-3405:~$ uname -a
Linux saksham-Vostro-3405 5.15.0-107-generic #117~20.04.1-Ubuntu SMP Tue Apr 30 10:35:57 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
saksham@saksham-Vostro-3405:~$ 
```

18. df -h – The `df -h` command shows disk space usage of all mounted file systems in a human-readable format (sizes in KB, MB, or GB), making it easier to understand storage usage.



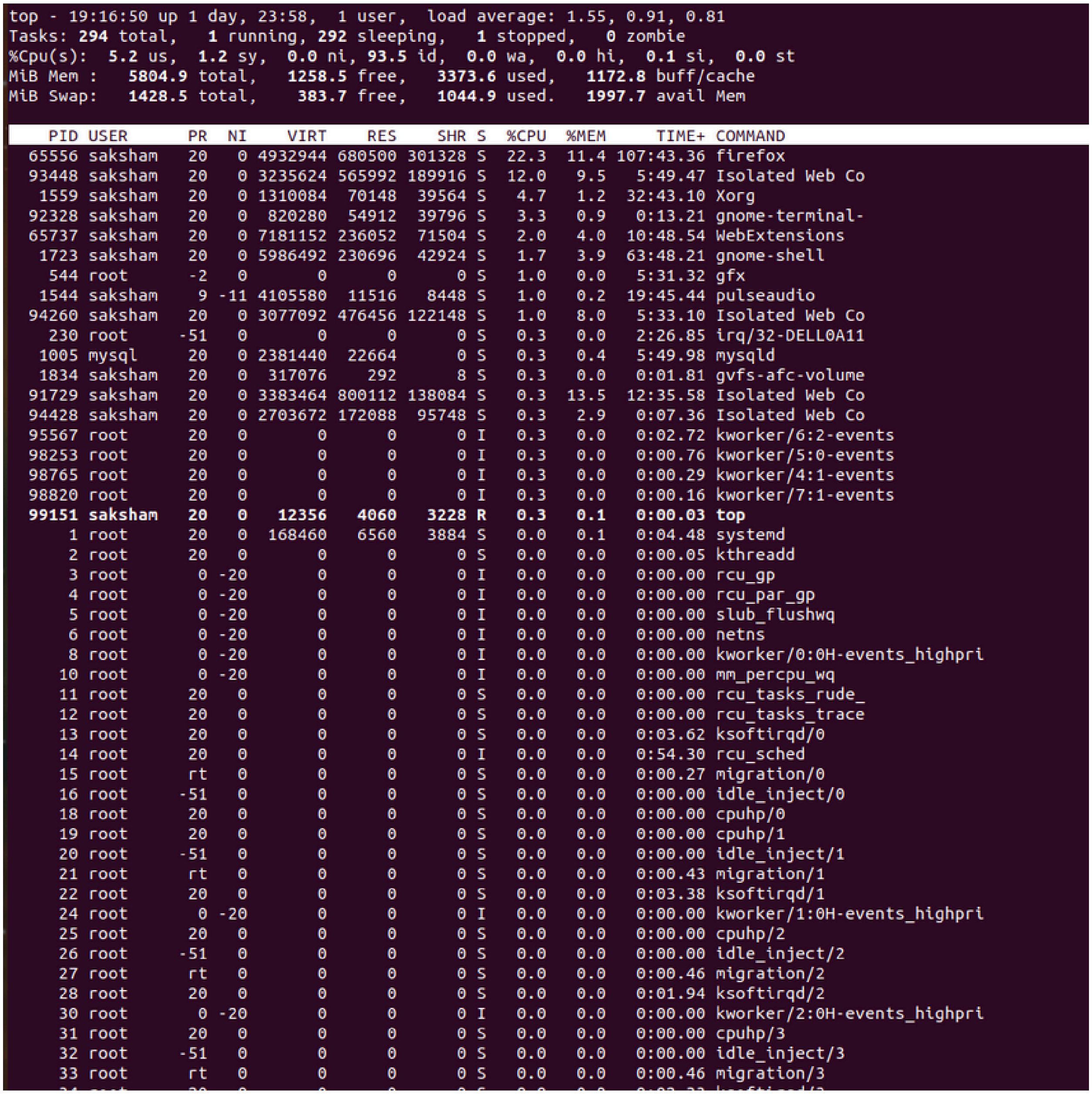
```
aksham@saksham-Vostro-3405:~$ df -h
ilesystem      Size  Used Avail Use% Mounted on
dev            2.8G     0  2.8G   0% /dev
mpfs           581M  3.3M  578M   1% /run
dev/nvme0n1p7   30G   28G  725M  98% /
mpfs           2.9G     0  2.9G   0% /dev/shm
mpfs           5.0M  4.0K  5.0M   1% /run/lock
mpfs           2.9G     0  2.9G   0% /sys/fs/cgroup
dev/loop0      128K  128K     0 100% /snap/bare/5
dev/loop1       56M   56M     0 100% /snap/core18/2959
dev/loop2       56M   56M     0 100% /snap/core18/2976
dev/loop5      517M  517M     0 100% /snap/gnome-42-2204/226
dev/loop4      350M  350M     0 100% /snap/gnome-3-38-2004/143
dev/loop3       74M   74M     0 100% /snap/core22/2139
dev/loop6      166M  166M     0 100% /snap/mysql-workbench-community/13
dev/loop7       64M   64M     0 100% /snap/core20/2682
dev/loop8       92M   92M     0 100% /snap/gtk-common-themes/1535
dev/loop13      51M   51M     0 100% /snap/snapd/25577
dev/loop14      13M   13M     0 100% /snap/snap-store/1113
dev/loop9       74M   74M     0 100% /snap/core22/2163
dev/loop10     517M  517M     0 100% /snap/gnome-42-2204/202
dev/loop16      51M   51M     0 100% /snap/snapd/25202
dev/loop12      64M   64M     0 100% /snap/core20/2669
dev/loop15      13M   13M     0 100% /snap/snap-store/1216
dev/loop11     347M  347M     0 100% /snap/gnome-3-38-2004/119
dev/nvme0n1p1   96M   69M   28M  72% /boot/efi
mpfs           581M  120K  581M   1% /run/user/1000
aksham@saksham-Vostro-3405:~$ 
```

19. ps -u $USER – The `ps -u $USER` command displays all processes currently running under your user account. It shows details like process ID (PID), terminal, CPU time, and the command that started each process.



```
saksham@saksham-Vostro-3405:~$ ps -u $USER
    PID TTY          TIME CMD
   1538 ?        00:00:02 systemd
   1539 ?        00:00:00 (sd-pam)
   1544 ?        00:19:44 pulseaudio
   1547 ?        00:00:01 gnome-keyring-d
   1552 ?        00:00:13 dbus-daemon
   1557 tty2     00:00:00 gdm-x-session
   1559 tty2     00:32:37 Xorg
   1580 tty2     00:00:00 gnome-session-b
   1647 ?        00:00:00 ssh-agent
   1676 ?        00:00:00 at-spi-bus-laun
   1681 ?        00:00:02 dbus-daemon
   1702 ?        00:00:00 gnome-session-c
   1709 ?        00:00:03 gnome-session-b
   1723 ?        01:03:40 gnome-shell
   1752 ?        00:00:00 gvfsd
   1757 ?        00:00:00 gvfsd-fuse
   1772 ?        00:00:01 ibus-daemon
   1776 ?        00:00:00 ibus-dconf
   1777 ?        00:00:01 ibus-extension-
   1781 ?        00:00:00 ibus-x11
   1783 ?        00:00:00 ibus-portal
   1794 ?        00:00:11 at-spi2-registr
   1798 ?        00:00:00 xdg-permission-
   1803 ?        00:00:00 gnome-shell-cal
   1811 ?        00:00:00 evolution-sourc
   1816 ?        00:00:00 dconf-service
   1820 ?        00:00:00 gvfs-udisks2-vo
   1827 ?        00:00:00 evolution-calen
   1828 ?        00:00:00 gvfs-gphoto2-vo
   1834 ?        00:00:01 gvfs-afc-volume
   1839 ?        00:00:00 gvfs-goa-volume
   1843 ?        00:00:00 gvfs-mtp-volume
   1855 ?        00:00:00 evolution-addre
   1866 ?        00:00:00 gjs
   1880 ?        00:00:00 gsd-a11y-settin
   1882 ?        00:00:01 gsd-color
   1885 ?        00:00:00 gsd-datetime
   1888 ?        00:00:02 gsd-housekeepin
   1890 ?        00:00:01 gsd-keyboard
   1891 ?        00:00:05 gsd-media-keys
   1894 ?        00:00:02 gsd-power
   1896 ?        00:00:00 gsd-print-notif
```

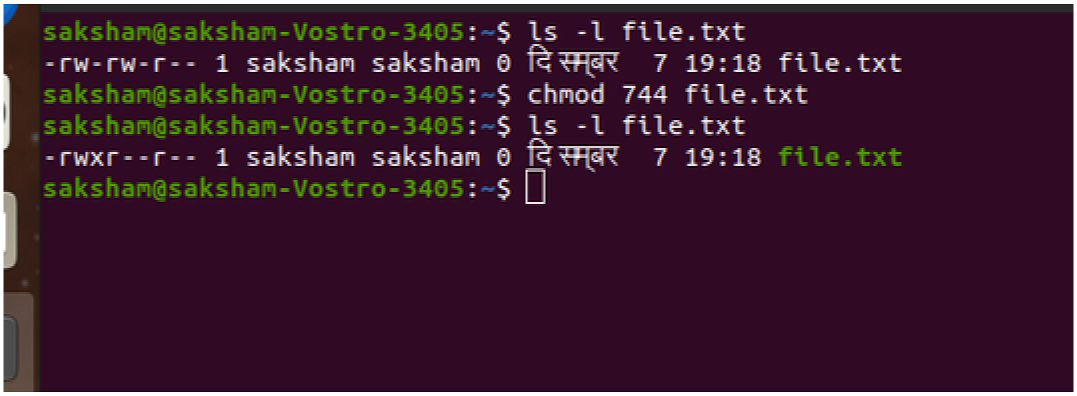
20. top – The `top` command displays a real-time view of system processes, including CPU and memory usage, process IDs, running time, and resource consumption. It is useful for monitoring system performance and identifying resource-heavy processes.



```
top - 19:16:50 up 1 day, 23:58,  1 user,  load average: 1.55, 0.91, 0.81
Tasks: 294 total,   1 running, 292 sleeping,   1 stopped,   0 zombie
%Cpu(s):  5.2 us,  1.2 sy,  0.0 ni, 93.5 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :   5804.9 total,   1258.5 free,   3373.6 used,   1172.8 buff/cache
MiB Swap:   1428.5 total,    383.7 free,   1044.9 used.   1997.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  65556 saksham   20   0 4932944 680500 301328 S  22.3  11.4 107:43.36 firefox
  93448 saksham   20   0 3235624 565992 189916 S  12.0   9.5   5:49.47 Isolated Web Co
   1559 saksham   20   0 1310084  70148  39564 S   4.7   1.2  32:43.10 Xorg
  92328 saksham   20   0  820280  54912  39796 S   3.3   0.9   0:13.21 gnome-terminal-
  65737 saksham   20   0 7181152 236052  71504 S   2.0   4.0  10:48.54 WebExtensions
   1723 saksham   20   0 5986492 230696  42924 S   1.7   3.9  63:48.21 gnome-shell
    544 root      -2   0       0      0      0 S   1.0   0.0   5:31.32 gfx
   1544 saksham    9 -11 4105580  11516   8448 S   1.0   0.2  19:45.44 pulseaudio
  94260 saksham   20   0 3077092 476456 122148 S   1.0   8.0   5:33.10 Isolated Web Co
    230 root     -51   0       0      0      0 S   0.3   0.0   2:26.85 irq/32-DELL0A11
   1005 mysql     20   0 2381440  22664      0 S   0.3   0.4   5:49.98 mysqld
   1834 saksham   20   0  317076    292      8 S   0.3   0.0   0:01.81 gvfs-afc-volume
  91729 saksham   20   0 3383464 800112 138084 S   0.3  13.5  12:35.58 Isolated Web Co
  94428 saksham   20   0 2703672 172088  95748 S   0.3   2.9   0:07.36 Isolated Web Co
  95567 root      20   0       0      0      0 I   0.3   0.0   0:02.72 kworker/6:2-events
  98253 root      20   0       0      0      0 I   0.3   0.0   0:00.76 kworker/5:0-events
  98765 root      20   0       0      0      0 I   0.3   0.0   0:00.29 kworker/4:1-events
  98820 root      20   0       0      0      0 I   0.3   0.0   0:00.16 kworker/7:1-events
  99151 saksham   20   0   12356   4060   3228 R   0.3   0.1   0:00.03 top
      1 root      20   0  168460   6560   3884 S   0.0   0.1   0:04.48 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:03.62 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:54.30 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.27 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     20 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     21 root      rt   0       0      0      0 S   0.0   0.0   0:00.43 migration/1
     22 root      20   0       0      0      0 S   0.0   0.0   0:03.38 ksoftirqd/1
     24 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     25 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
     26 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/2
     27 root      rt   0       0      0      0 S   0.0   0.0   0:00.46 migration/2
     28 root      20   0       0      0      0 S   0.0   0.0   0:01.94 ksoftirqd/2
     30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/2:0H-events_highpri
     31 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/3
     32 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/3
     33 root      rt   0       0      0      0 S   0.0   0.0   0:00.46 migration/3
```

21. chmod – The `chmod` command is used to change the permissions of files or directories in Linux. It controls read (r), write (w), and execute (x) permissions for the owner, group, and others.



```
saksham@saksham-Vostro-3405:~$ ls -l file.txt
-rw-rw-r-- 1 saksham saksham 0 दिसम्बर  7 19:18 file.txt
saksham@saksham-Vostro-3405:~$ chmod 744 file.txt
saksham@saksham-Vostro-3405:~$ ls -l file.txt
-rwxr--r-- 1 saksham saksham 0 दिसम्बर  7 19:18 file.txt
saksham@saksham-Vostro-3405:~$ 
```