Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Lab Report

on

"Introduction to Linux"

Submitted by

Roshis Sainju (45)

Submitted to

Ms. Rabina Shrestha

Department of Computer Science and Engineering

Submission Date: December 10 , 2025

Introduction

Linux

Linux is a family of operating systems built around the principles of openness and community-driven development. It functions as the core software layer that coordinates hardware activity and provides a foundation for applications to run. Since its initial release by Linus Torvalds in the early 1990s, Linux has grown into an ecosystem used by organizations and developers who prioritize flexibility, transparency, and performance. It supports a wide variety of environments, ranging from large-scale data centres to everyday computing devices, and serves as a preferred platform in technical fields where control, scripting, and automation are essential.

Linux Hierarchical File System

The Linux file system is arranged as a single connected structure rather than separate drives or volumes. At its highest level is the root directory, written as /. Every directory and file that exists on the system stems from this point, forming a branching layout. Key areas of the system are arranged according to function. User accounts are placed under /home, critical configuration details are stored within /etc, executable programs are located in directories such as /bin, and operational data, such as logs, appears under /var. A path like /home/student/projects illustrates how locations are nested, making it easier to navigate and manage resources in a consistent way.

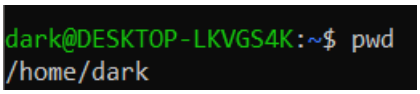
Importance of Linux

• Users can access, study, and modify the system without licensing barriers.
• The permission structure helps prevent unauthorized activity and protects system resources.
• Its reliability makes it suitable for continuous operation in enterprise and cloud settings.
• Administrators benefit from a broad collection of command-line utilities that simplify automation.
• It plays a major role in technical disciplines where scripting, networking, and system-level control are important.
• Linux distributions operate on diverse hardware, from small embedded chips to high-performance computing clusters.
• A large global community contributes updates, improvements, and long-term support.

Commands

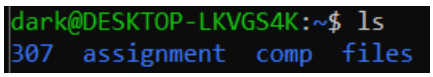
1. PWD :

The pwd command allows users to check the exact directory they are currently operating in by showing the complete path from the root of the file system. This is especially helpful when working deep inside multiple folders or when writing scripts that depend on accurate directory references. It acts as a quick way to confirm your position before performing file operations.



2. ls:

The ls command provides an overview of what is stored inside a directory. Running it gives a simple listing of available files and folders, and adding optional flags can reveal extra details such as size information, time of last change, and permission settings.

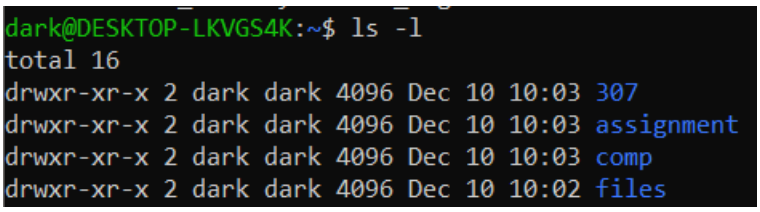


3. ls -a:

When ls -a is executed, Linux displays every item inside the directory, including files that are normally hidden. These hidden items begin with a dot and are often used for storing configuration data. This option is useful when troubleshooting user settings or inspecting files that would not appear in a standard directory listing.

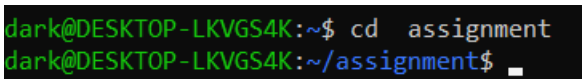
4. ls -l:

Using ls -l produces a structured, line-by-line view of directory contents. This format includes ownership data, access rules, file size, and timestamps. Because of this additional detail, it is commonly used when verifying who controls a file, how it can be accessed, or whether its attributes need modification.



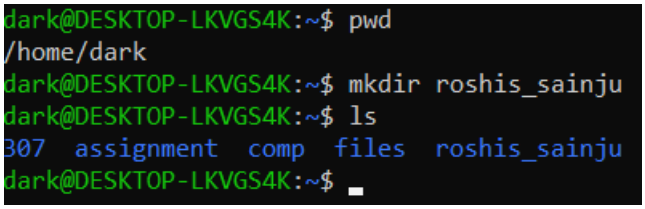
5. cd:

The cd command allows you to move from one directory to another by specifying the desired location. It supports relative and absolute paths, making navigation flexible whether you are exploring nearby folders or jumping across the system. If no location is given, the command returns the user to their home directory.



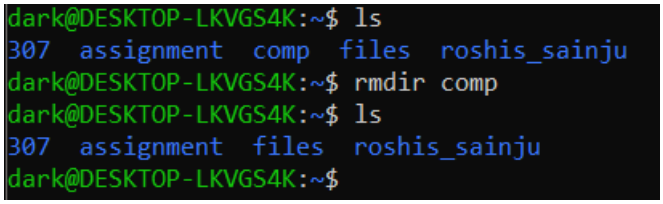
6. mkdir:

The mkdir command is used whenever you need to create a new folder within the file system. It helps keep information organized by allowing users to set up dedicated directories for different projects, documents, or data sets.



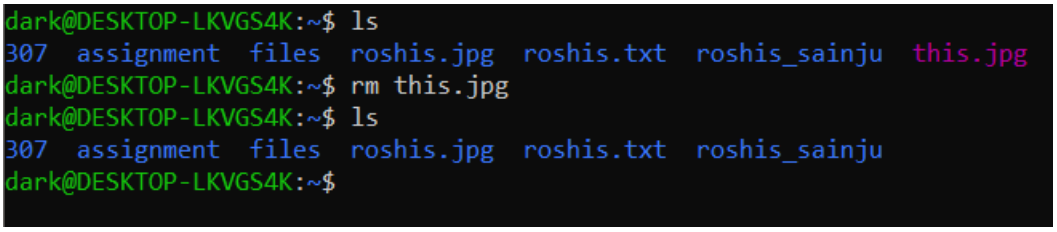
7. rmdir:

The rmdir command removes a directory, but only if it contains nothing inside it. If the folder still holds files or additional subfolders, Linux will prevent the action and report an error, which protects users from accidental data loss.



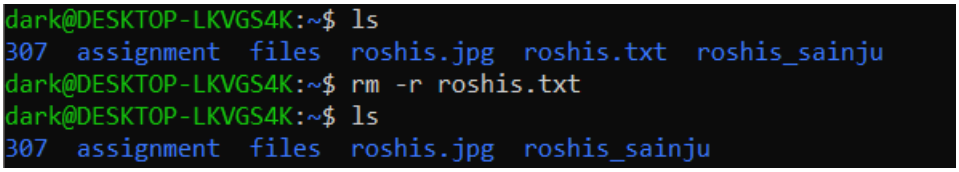
8. rm:

The rm command deletes files from the system. Once a file is removed with this command, normal tools cannot bring it back easily. This is why it is often recommended to double check the file name before using it.



9. rm -r:

The rm -r option allows users to delete a directory along with everything it contains. This includes all files and any nested directories. Because it clears entire structures in one action, it is a powerful command that needs to be handled carefully.



10. touch:

The touch command has two common uses. It can create a brand new, empty file with whatever name you specify, or it can update the time metadata of an existing file. Developers often use it to prepare placeholder files.

```
dark@DESKTOP-LKVGS4K:~$ touch listeningto_trishnagurung
dark@DESKTOP-LKVGS4K:~$ ls
307  assignment  files  listeningto_trishnagurung  roshis.jpg  roshis_sainju
dark@DESKTOP-LKVGS4K:~$
```

11. cat:

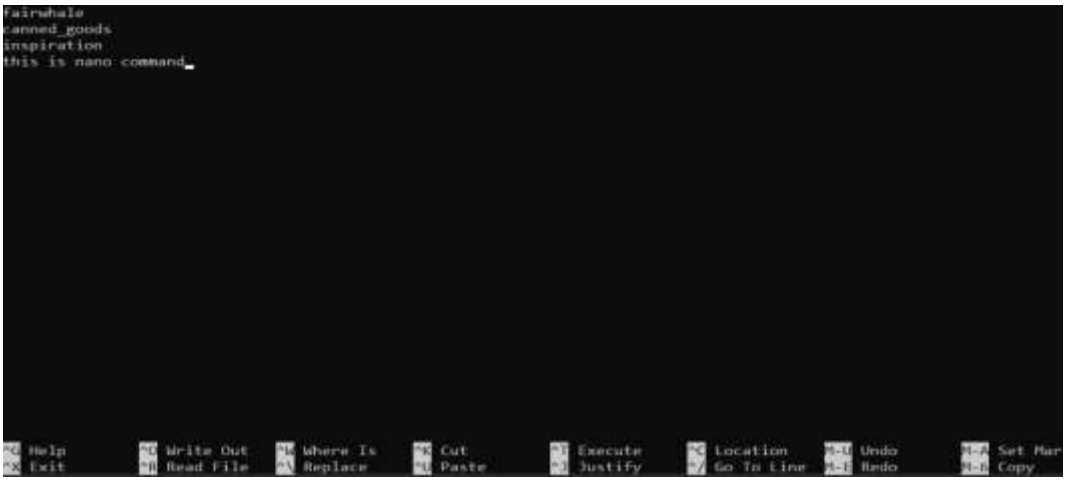
The cat command prints a file's contents directly onto the terminal screen, which makes it useful for quick reviews. It can also merge several files together or send their output into another file, depending on how it is combined with redirection.

```
dark@DESKTOP-LKVGS4K:~/roshis_sainju$ cat designs.txt
fairwhale
canned_goods
inspiration
```

12. nano, vi, jed:

These programs are terminal-based editors that let users write or adjust text files. nano is simple and easy for beginners to learn. vi offers extensive editing capabilities and is often preferred by experienced users. jed is a lighter editor that provides a straightforward environment for quick edits.

```
fairwhale
canned_goods
inspiration
~
~
~
~
~
~
~
~
```



13. cp:

The cp command is used whenever you want to make a duplicate of a file or folder and place it somewhere else in the system. Additional options can expand its functionality. For example, using the r option allows an entire directory and its internal structure to be copied at once.

```
dark@DESKTOP-LKVGS4K:~$ ls
307  assignment  designs  files  listeningto_trishnagurung  roshis.jpg  roshis_sainju
dark@DESKTOP-LKVGS4K:~$ cp designs
cp: missing destination file operand after 'designs'
Try 'cp --help' for more information.
dark@DESKTOP-LKVGS4K:~$ cp designs files
dark@DESKTOP-LKVGS4K:~$ cd files
dark@DESKTOP-LKVGS4K:~/files$ ls
designs
dark@DESKTOP-LKVGS4K:~/files$
```

14. mv:

The mv command handles both moving and renaming. It can shift a file to a different directory or update its name in the same location. Because it replaces existing names without confirmation, it is used carefully when working with important data.

```
dark@DESKTOP-LKVGS4K:~$ ls
307  assignment  designs  files  listeningto_trishnagurung  roshis.jpg  roshis_sainju
dark@DESKTOP-LKVGS4K:~$ mv listeningto_trishnagurung files
dark@DESKTOP-LKVGS4K:~$ cd files
dark@DESKTOP-LKVGS4K:~/files$ ls
designs  listeningto_trishnagurung
dark@DESKTOP-LKVGS4K:~/files$
```

15. locate:

The locate command helps you find files by searching through an indexed database that the system maintains. Since it refers to this prepared index rather than scanning the system live, it produces results very quickly.

16. echo:

The echo command displays a line of text or the value stored in a variable. It is frequently used inside scripts to show progress messages or return specific information to the terminal.

```
dark@DESKTOP-LKVGS4K:~$ echo "canned goods designs are really good"
canned goods designs are really good
```

17. uname -a:

Running uname -a prints a comprehensive summary of system-related information. It includes details such as the kernel version, system architecture, and other platform identifiers. It is often used when checking compatibility or troubleshooting.

```
dark@DESKTOP-LKVGS4K:~$ uname -a
Linux DESKTOP-LKVGS4K 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
```

18. df -h:

The df -h command reports how much disk space is in use and how much remains available on each mounted file system. Its output is presented in readable units like megabytes or gigabytes, which makes it easier to evaluate storage needs.

```
dark@DESKTOP-LKVGS4K:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            1.9G     0  1.9G   0% /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2
none            1.9G  4.0K  1.9G   1% /mnt/wsl
drivers         160G  116G   45G  73% /usr/lib/wsl/drivers
/dev/sdd       1007G  1.8G  954G   1% /
none            1.9G   72K  1.9G   1% /mnt/wslg
none            1.9G     0  1.9G   0% /usr/lib/wsl/lib
rootfs          1.9G  2.7M  1.9G   1% /init
none            1.9G  492K  1.9G   1% /run
none            1.9G     0  1.9G   0% /run/lock
none            1.9G     0  1.9G   0% /run/shm
none            1.9G   76K  1.9G   1% /mnt/wslg/versions.txt
none            1.9G   76K  1.9G   1% /mnt/wslg/doc
C:\             160G  116G   45G  73% /mnt/c
D:\              79G   14G   65G  18% /mnt/d
E:\             500M   37M  464M   8% /mnt/e
F:\             195G  281M  195G   1% /mnt/f
G:\             375G  1.6G  374G   1% /mnt/g
H:\             362G   57G  305G  16% /mnt/h
tmpfs           385M   20K  385M   1% /run/user/1000
```

19. ps -u $USER:

This command displays the processes currently running under the logged-in user's account. It shows identifiers such as process IDs, CPU activity, and the commands responsible for launching each task.

```
dark@DESKTOP-LKVGS4K:~$ ps -u $USER
    PID TTY          TIME CMD
    368 pts/0    00:00:00 bash
    428 ?        00:00:00 systemd
    431 ?        00:00:00 (sd-pam)
    455 pts/1    00:00:00 bash
   2279 pts/0    00:00:00 ps
dark@DESKTOP-LKVGS4K:~$ ps  -u $dark
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dark         368  0.0  0.1   6072  5248 pts/0    Ss   11:22   0:00 -bash
dark         455  0.0  0.1   6072  4864 pts/1    S+   11:22   0:00 -bash
dark        2444  0.0  0.1   8280  4096 pts/0    R+   11:28   0:00 ps -u
```

20. top:

The top command offers a live, continuously updating overview of system activity. It highlights memory consumption, processor load, and which tasks are using the most system resources. It is a valuable tool when diagnosing performance issues.

```
dark@DESKTOP-LKVGS4K:~$ top
top - 11:29:33 up  1:39,  1 user,  load average: 0.06, 0.49, 0.72
Tasks:  22 total,   1 running,  21 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  2.0 sy,  0.0 ni, 95.8 id,  1.3 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   3849.2 total,   2888.6 free,    474.8 used,    623.2 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used.   3374.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
     75 root      19  -1   66828  15380  14356 S   3.7   0.4   0:05.99 systemd-journal
      1 root      20   0   21704  12512   9184 S   1.0   0.3   0:06.44 systemd
    179 systemd+  20   0   21452  12416  10368 S   0.3   0.3   0:00.29 systemd-resolve
    193 message+  20   0    9524   4992   4480 S   0.3   0.1   0:01.17 dbus-daemon
      2 root      20   0    3120   2048   1920 S   0.0   0.1   0:00.03 init-systemd(Ub
     11 root      20   0    3136   1804   1792 S   0.0   0.0   0:00.00 init
    123 root      20   0   25164   6272   4864 S   0.0   0.2   0:00.33 systemd-udevd
    183 systemd+  20   0   91020   7680   6784 S   0.0   0.2   0:00.25 systemd-timesyn
    192 root      20   0    4236   2432   2304 S   0.0   0.1   0:00.01 cron
    200 root      20   0   17968   8320   7424 S   0.0   0.2   0:00.55 systemd-logind
    208 root      20   0    3160   1920   1792 S   0.0   0.0   0:00.03 agetty
    213 syslog    20   0  222508   4992   4352 S   0.0   0.1   0:00.48 rsyslogd
    227 root      20   0    3116   1792   1664 S   0.0   0.0   0:00.01 agetty
    229 root      20   0  107008  22144  12928 S   0.0   0.6   0:00.29 unattended-upgr
    366 root      20   0    3124    896    768 S   0.0   0.0   0:00.00 SessionLeader
    367 root      20   0    3140   1156   1024 S   0.0   0.0   0:00.14 Relay(368)
    368 dark      20   0    6072   5248   3584 S   0.0   0.1   0:00.22 bash
    369 root      20   0    6688   4480   3712 S   0.0   0.1   0:00.02 login
    428 dark      20   0   20120  10880   9088 S   0.0   0.3   0:00.26 systemd
    431 dark      20   0   21144   3516   1792 S   0.0   0.1   0:00.00 (sd-pam)
    455 dark      20   0    6072   4864   3456 S   0.0   0.1   0:00.04 bash
   2697 dark      20   0    9420   5504   3328 R   0.0   0.1   0:00.02 top
```

21. chmod:

The chmod command is used to modify access rights on files and directories. It lets the owner determine who can read, write, or execute a particular item. Permissions can be adjusted through numeric codes or symbolic notation depending on user preference.