

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



Lab Assignment 1
[Code No.: COMP 307]

Submitted by:

Name: Jebisha Bariya
Roll No: 06
Group: CS 60
Level: III/I

Submitted to:

Rabina Shrestha
Department of Computer
Science and Engineering

Introduction

1. What is Linux?

Linux is a family of open-source, Unix-like operating systems (OS) based on the Linux kernel. It acts as the core intermediary between a computer's hardware and its applications, managing resources like CPU, memory, and storage. Linux is known for its stability, security, and flexibility, making it the dominant OS for servers, supercomputers, and cloud infrastructure.

2. The Linux Hierarchical File System (FHS)

The Linux file system is organized in a single, unified, tree-like structure branching from the root directory (/). All files and directories, regardless of which device they reside on, are accessed through this one tree. Key standard directories include /home (user files), /etc (configuration), /bin (essential user binaries), and /var (variable data like logs).

3. Importance of Linux Commands in Operating Systems

Linux commands, executed through the Command Line Interface (CLI) or shell (like Bash), provide direct, precise, and efficient control over the operating system. Their importance is threefold:

1. System Administration: Commands are essential for managing user permissions, installing software, troubleshooting, and configuring core system services.
2. Automation: Commands can be combined into powerful shell scripts, allowing administrators to automate complex, repetitive tasks (e.g., backups, data processing).
3. Remote Management: Commands are the standard method for securely managing remote servers via protocols like SSH, which is fundamental to modern cloud computing.

Commands:

1. Pwd : The pwd command is mostly used to print the current working directory on your terminal. It is also one of the most commonly used commands. Another purpose for this command is when creating scripts because it can help us find the directory in which the script was saved.

Screenshot:

```
jbsha@J:~$ pwd  
/home/jbsha  
jbsha@J:~$ |
```

2. ls: The ls command is commonly used to identify the files and directories in the working directory. This command can be used by itself without any arguments and it will provide us the output with all the details about the files and the directories in the current working directory.

Screenshot:

```
jbsha@J:~$ ls  
newFolder photos work  
jbsha@J:~$ |
```

3. ls -a: Lists all contents, including hidden files and directories (those starting with a dot, .). The output always includes . (current directory) and .. (parent directory).

Screenshot:

```
jbsha@J:~$ ls -a  
. .bash_logout .cache .profile photos  
jbsha@J:~$ |
```

4. ls -l: Provides a long, detailed listing, including file permissions, number of links, owner, group, size, and last modification date.

Screenshot:

```
jbsha@J:~$ ls -l
total 12
drwxr-xr-x 2 jbsha jbsha 4096 Dec 10 05:12 newFolder
drwxr-xr-x 2 jbsha jbsha 4096 Dec 10 05:14 photos
drwxr-xr-x 2 jbsha jbsha 4096 Dec 10 05:14 work
jbsha@J:~$ |
```

5. Cd: The cd command is used to navigate between directories. It requires either the full path or the directory name, depending on your current working directory.

Screenshot:

```
jbsha@J:~$ cd work
jbsha@J:~/work$ |
```

6. Mkdir : This mkdir command allows to create fresh directories in the terminal itself. The default syntax is mkdir <directory name> and the new directory will be created.

Screenshot:

```
jbsha@J:~$ mkdir newFolder
jbsha@J:~$ ls
newFolder
```

7. Rmdir: The rmdir command is used to delete permanently an empty directory. To perform this command the user running this command must be having sudo privileges in the parent directory.

Screenshot:

```
jbsha@J:~$ rmdir newFolder
jbsha@J:~$ ls
photos work
jbsha@J:~$
```

8. Rm: rm command in Linux is generally used to delete the files created in the directory.

Screenshot:

```
jbsha@J:~/work$ rm journal.txt
jbsha@J:~/work$ ls
jbsha@J:~/work$
```

9. Touch & Rm -r: Touch creates an empty file named new.txt if it does not exist. If the file already exists, it updates the file's access and modification timestamps. Rm -r Removes a directory and its contents recursively. This is used to delete non-empty folders.

Screenshot:

```
jbsha@J:~/work$ touch new.txt
jbsha@J:~/work$ touch assignment.txt
jbsha@J:~/work$ cd ..
jbsha@J:~$ rm -r work
jbsha@J:~$ ls
photos
jbsha@J:~$
```

10. Echo & Cat: Echo prints the specified string ("Hello") to standard output (the terminal). Often used with > or >> to write text to files. Cat concatenates files and prints their contents to the standard output (terminal screen). It is commonly used to quickly view the content of a text file.

Screenshot:

```
jbsha@J:~/photos$ echo "Linux is a family of a open-source. Unix-like operating system (OS) based on the linux kernel." >> assignment.txt
jbsha@J:~/photos$ ls
assignment.txt lab.txt
jbsha@J:~/photos$ cat assignment.txt
Linux is a family of a open-source. Unix-like operating system (OS) based
on the linux kernel.
jbsha@J:~/photos$
```

11. Nano: A simple, command-line text editor ideal for beginners.

Screenshot:

The screenshot shows a terminal window titled "GNU nano 7.2" with the file "assignment.txt" open. The content of the file is:

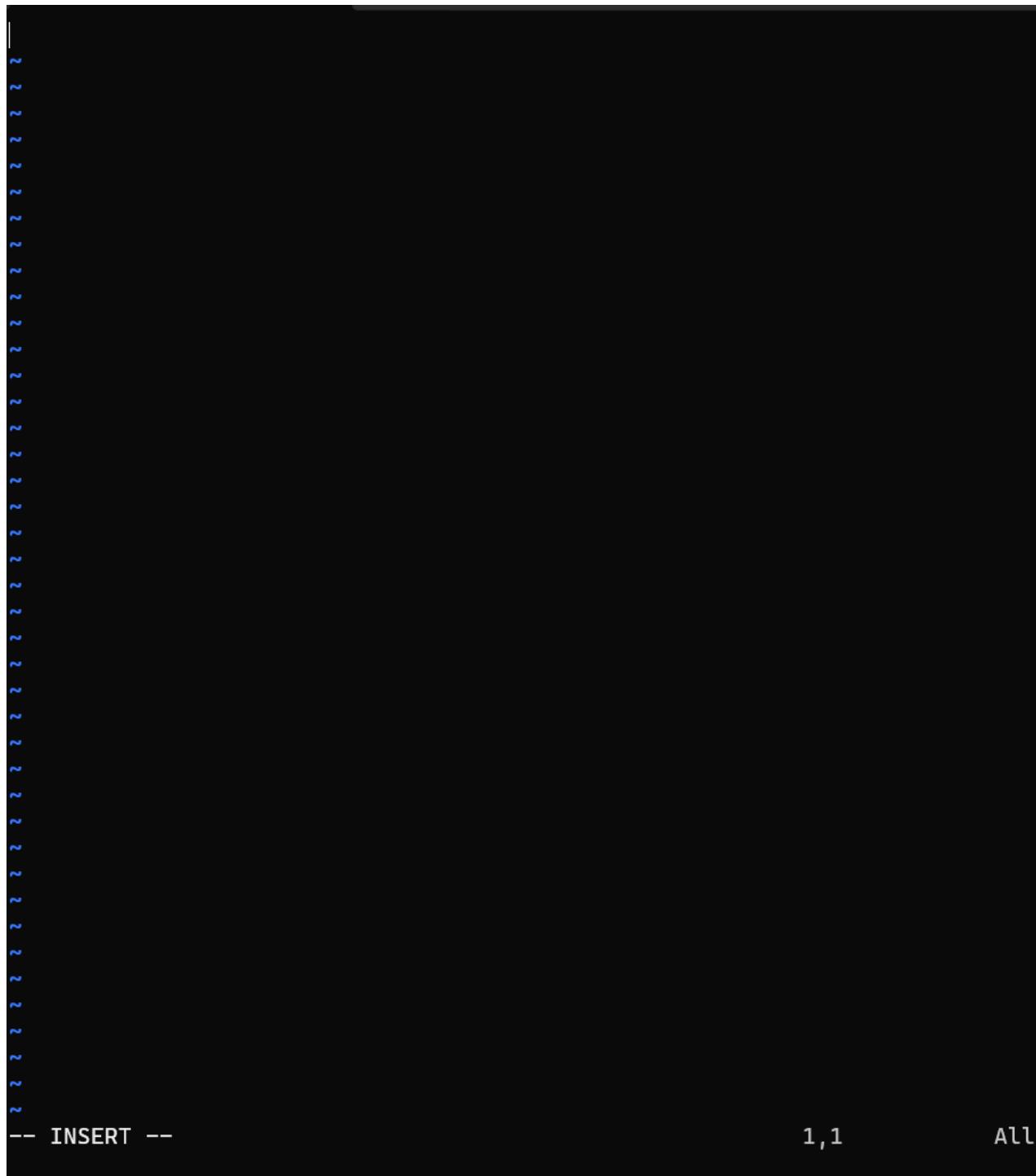
```
GNU nano 7.2           assignment.txt *
Linux is a family of open-source Unix-like operating systems (OS) based on the Linux kernel.
```

At the bottom of the screen, there is a menu of keyboard shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\\ Replace	^U Paste	^J Justify

12. Vi/Vim: A powerful, highly configurable, modal text editor.

Screenshot:



13. Cp: The cp command of Linux is equivalent to copy-paste and cut-paste in Windows.

Screenshot:

```
jbsha@J:~/photos$ cp assignment.txt lab.txt
jbsha@J:~/photos$ cat lab.txt
Linux is a family of open-source. Unix-like operating system (OS) base
n the linux kernel.
jbsha@J:~/photos$ |
```

14. Mv: The mv command is generally used for renaming the files in Linux.

Screenshot:

```
jbsha@J:~/photos$ mv assignment.txt new.txt
jbsha@J:~/photos$ ls
lab.txt  new.txt
jbsha@J:~/photos$ |
```

15. Locate: The locate command is generally used to locate the files in the database. Use an asterisk (*) to search for content that contains two or more words. As an example: locate first*file. This command will search the database for the files that contain these two names first and file.

Screenshot:

```
jbsha@J:~$ locate bashrc
/etc/bash.bashrc
/etc/skel/.bashrc
/home/jbsha/.bashrc
/mnt/c/Program Files/Git/etc/bash.bashrc
/mnt/c/msys64/etc/bash.bashrc
/mnt/c/msys64/etc/skel/.bashrc
/mnt/c/msys64/home/jebi3/.bashrc
/usr/share/dot.bashrc
/usr/share/base-files/dot.bashrc
/usr/share/byobu/profiles/bashrc
/usr/share/doc/adduser/examples/adduser.local.conf.examples/bash.bashrc
/usr/share/doc/adduser/examples/adduser.local.conf.examples/skel/dot.bashr
jbsha@J:~$ |
```

16. Uname -a: Shows complete system information (unix name), including kernel name, network hostname, kernel release, version, and architecture.

Screenshot:

```
jbsha@J:~$ uname -a
Linux J 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5
8:30:46 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
jbsha@J:~$ |
```

17. Df -h: Disk free command. Displays the amount of available and used disk space on mounted filesystems in human-readable format (e.g., KB, MB, GB).

Screenshot:

```
jbsha@J:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            1.5G    0   1.5G   0% /usr/lib/modules/6.6.87.2-microsoft-s
andard-WSL2
none            1.5G  4.0K  1.5G   1% /mnt/wsl
drivers         195G  185G   11G  95% /usr/lib/wsl/drivers
/dev/sdd        1007G  1.6G  955G   1% /
none            1.5G   80K  1.4G   1% /mnt/wslg
none            1.5G    0   1.5G   0% /usr/lib/wsl/lib
rootfs          1.4G  2.7M  1.4G   1% /init
none            1.5G  504K  1.4G   1% /run
none            1.5G    0   1.5G   0% /run/lock
none            1.5G    0   1.5G   0% /run/shm
none            1.5G   96K  1.4G   1% /mnt/wslg/versions.txt
none            1.5G   96K  1.4G   1% /mnt/wslg/doc
C:\             195G  185G   11G  95% /mnt/c
D:\             9.8G  9.1G  720M  93% /mnt/d
E:\             1020M  59M  962M   6% /mnt/e
F:\             31G   9.1G   22G  30% /mnt/f
tmpfs           287M   20K  287M   1% /run/user/1000
jbsha@J:~$ |
```

18. Ps -u \$USER: Process status. Lists all currently running processes associated with the current user (\$USER).

Screenshot:

```
jbsha@J:~$ ps -u jbsha
  PID TTY          TIME CMD
 387 pts/0        00:00:01 bash
 440 ?            00:00:01 systemd
 441 ?            00:00:00 (sd-pam)
 495 pts/1        00:00:00 bash
 6360 pts/0       00:00:00 ps
jbsha@J:~$ |
```

19. Top: Displays a real-time, dynamic view of the operating system's process activity, including CPU usage, memory usage, and running processes.

Screenshot:

```
jbsha@J:~$ top
top - 07:15:04 up 1:55, 1 user, load average: 0.02, 0.01, 0.00
Tasks: 24 total, 1 running, 23 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.1 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si,
MiB Mem : 2867.2 total, 2339.9 free, 438.9 used, 176.9 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 2428.3 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+
            1 root      20   0  22112  13056  9472 S  0.0  0.4  0:17.00
            2 root      20   0   3120   2048  1920 S  0.0  0.1  0:00.25
            9 root      20   0   3136   1800  1792 S  0.0  0.1  0:00.00
           182 root     20   0   4236   2560  2304 S  0.0  0.1  0:00.91
           183 message+ 20   0   9788   5120  4352 S  0.0  0.2  0:02.86
           203 root     20   0  17964   8320  7552 S  0.0  0.3  0:01.42
           224 root     20   0 1829572  13440 11008 S  0.0  0.5  0:03.12
           267 root     20   0   3160   1920  1792 S  0.0  0.1  0:00.03
           273 root     20   0   3116   1792  1664 S  0.0  0.1  0:00.02
           282 root     20   0 107012  22144 12928 S  0.0  0.8  0:00.35
           385 root     20   0   3124    900   768 S  0.0  0.0  0:00.00
           386 root     20   0   3140   1160  1024 S  0.0  0.0  0:00.37
           387 jbsha    20   0   6072   4992  3456 S  0.0  0.2  0:01.36
           388 root     20   0   6692   4224  3584 S  0.0  0.1  0:00.04
           440 jbsha    20   0  20004  11008  9216 S  0.0  0.4  0:01.04
           441 jbsha    20   0  21144   3516  1792 S  0.0  0.1  0:00.00
           495 jbsha    20   0   6056   4736  3328 S  0.0  0.2  0:00.10
           951 polkitd   20   0 308160   7680  6912 S  0.0  0.3  0:00.62
          3273 root     19  -1  33912  11776 10880 S  0.0  0.4  0:02.07
          3326 systemd+ 20   0  91020   7808  6912 S  0.0  0.3  0:00.49
          3378 root     20   0  24976   6144  4992 S  0.0  0.2  0:00.45
          3429 systemd+ 20   0  21452  13056 10880 S  0.0  0.4  0:00.43
          6148 syslog    20   0  222508   5248  4608 S  0.0  0.2  0:00.25
          6361 jbsha    20   0   9296   5632  3456 R  0.0  0.2  0:00.07
```

20. Chmod: Change mode (permissions). This example sets permissions (rwx for owner, r-x for group/others) for script.sh using octal notation.

Screenshot:

```
jbsha@J:~/photos$ chmod g+w new.txt
jbsha@J:~/photos$ ls -l
total 8
-rw-r--r-- 1 jbsha jbsha 97 Dec 10 05:31 lab.txt
-rw-rw-r-- 1 jbsha jbsha 97 Dec 10 05:28 new.txt
jbsha@J:~/photos$ |
```

