K.R. MANGALAM UNIVERSITY

TITLE : Basics of Linux and Open-Source Tools

NAME : ADITYA JHA

COURSE : B.TECH CSE

ROLL NO : 2501010062

SEMESTER : 1st

Course Name : Computer Science Fundamentals and Carrer PATHWAYS

COURSE CODE : ETCCCP105

FACULTY : MR. RAJESH KUMAR .

2.1) LINUX INSTALLATION :

Introduction :

I installed **Ubuntu Linux** on my Windows system using **Windows Subsystem for Linux (WSL 2).**

WSL 2 allows us to run a **real Linux environment directly inside Windows** without using a separate virtual machine. This setup provides a safe and efficient environment to **practice Linux commands**

and work with real Linux tools, while still using Windows as the primary operating system.

Step 1: Enable WSL Feature

- Press **Windows + R**, type `powershell`, and right-click → **Run as Administrator**
- In PowerShell, type:
- This command installs **Windows Subsystem for Linux** and **Virtual Machine Platform** automatically.
- Restart your system once installation completes.

Step 2: Download Ubuntu

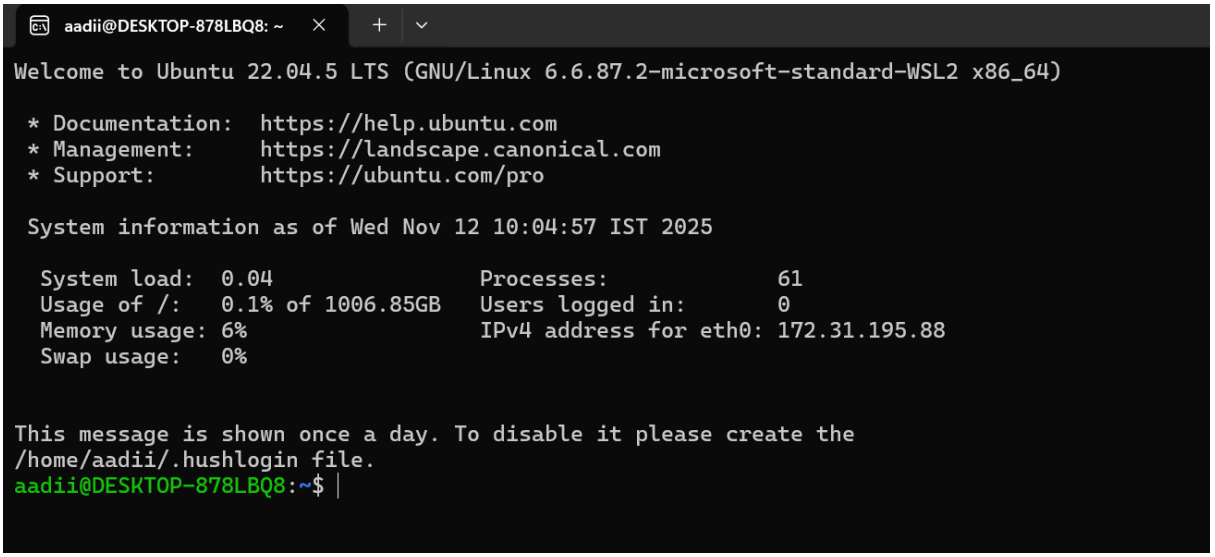
- Open **Microsoft Store**
- Search for **“Ubuntu 22.04 LTS”**
- Click **Get / Install**
- Wait for the download to complete
- Once installed, click **Open**

Step 3: Set Up Ubuntu for the First Time

- When you launch Ubuntu for the first time, it will show:

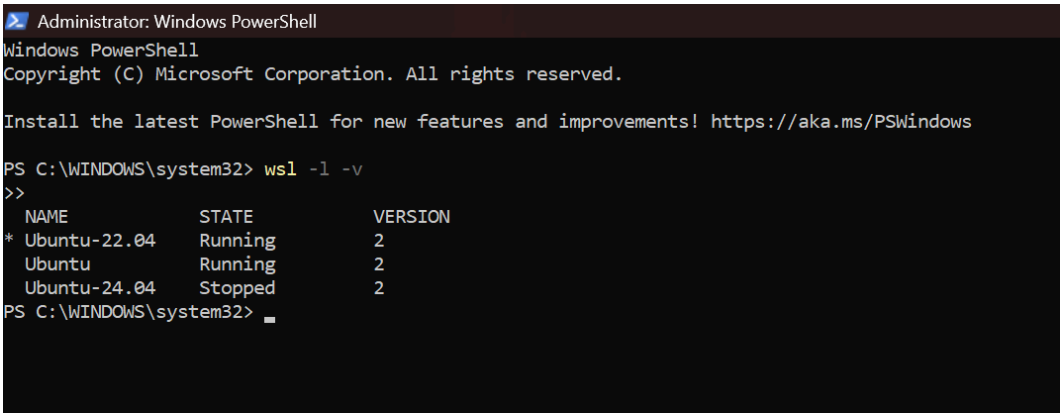
  "Installing, this may take a few minutes..."

- After setup, it asks to create:
  - **Username**
  - **Password**
- Once done, you'll see the Ubuntu terminal prompt like:



Step 4: Verify Installation

- To confirm Ubuntu is installed and running, type:



Step 5: Final Working Ubuntu Terminal

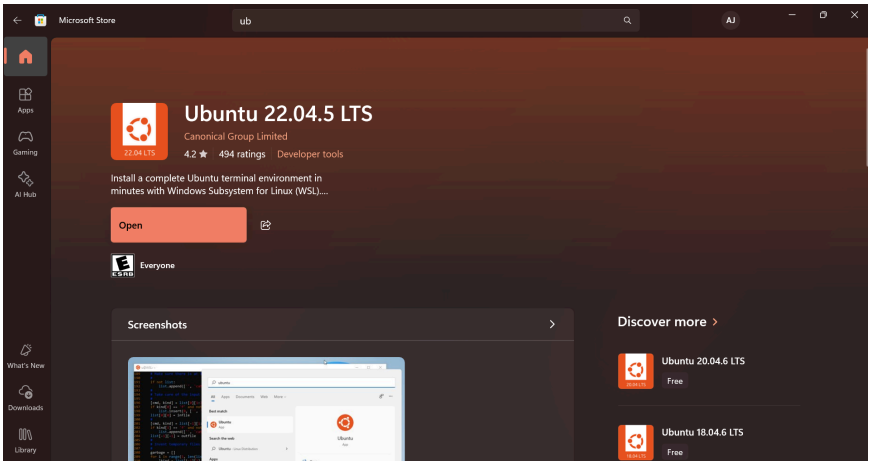
- You now have a fully functional Ubuntu command-line system running *inside Windows* — no separate virtual machine required.
- You can directly execute Linux commands, create shell scripts, and perform all tasks safely within WSL.

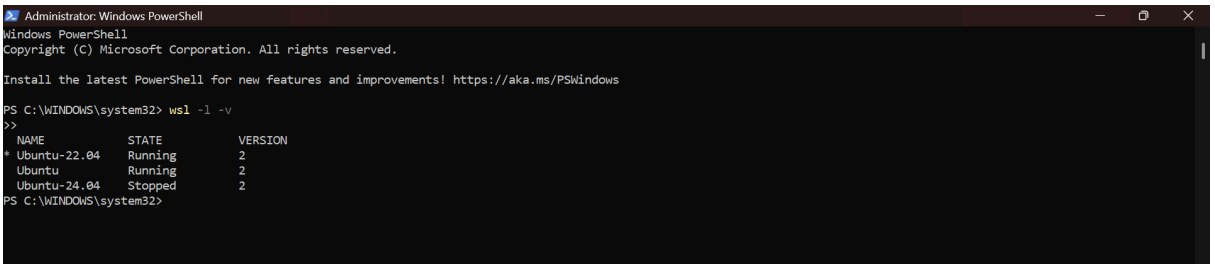
. Hardware Configuration Details:

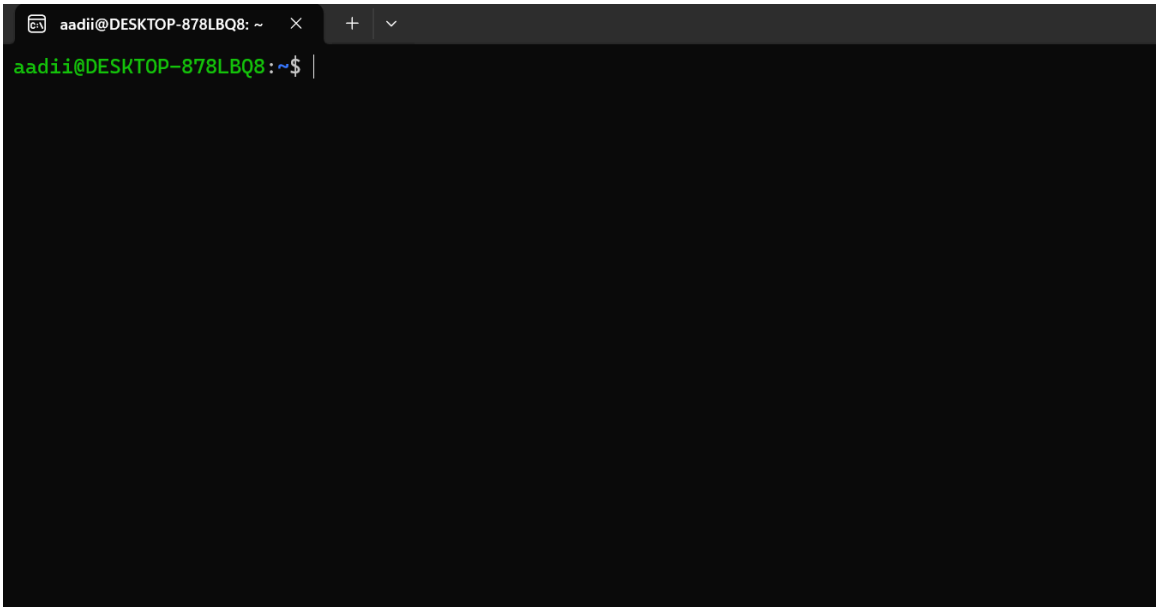
Component	Specification
System Manufacturer	Dell Inc.
System Model	Latitude 5420
Processor (CPU)	11th Gen Intel® Core™ i5-1145G7 @ 2.60GHz (8 CPUs)
Installed RAM	16.0 GB (15.4 GB usable)
System Type	64-bit Operating System, x64-based processor
Graphics Card	Intel® Iris® Xe Graphics (128 MB VRAM, 8008 MB shared memory)
Storage	238 GB SSD (87 GB used)
Disk Allocated for Ubuntu (WSL)	Automatically managed by Windows (Dynamic allocation)
Virtualization Type	Windows Subsystem for Linux (WSL 2) using lightweight Hyper-V backend

Final Working Ubuntu Environment

Now you have **Ubuntu 22.04 LTS** running **inside Windows using WSL 2** — a fully functional Linux terminal environment. It's ready to **execute shell commands, write and run shell scripts**, and perform all Linux operations seamlessly, **without needing a separate virtual machine**.







2.2) SHELL COMMANDS IMPLEMENTATION AND DOCUMENTATION :

1) pwd Command:

Description :

This command shows the current folder (directory) in which I am working.

It basically tells “where I am” in the Linux file system.

When I used it:

I used `pwd` to confirm that I was in my home directory `/home/aadii`.

2) ls Command:

Description:

This command lists all files and folders present in the current directory.

When I used it:

I used `ls` to check what items were present inside my home folder and inside my testfolder.

3) cd Command:

Description:

`cd` means “change directory.”

It is used to move from one folder to another.

When I used it:

I used `cd ~` to come back to the home directory, and `cd testfolder` to enter the folder I created.

```
aadii@DESKTOP-878LBQ8:~$ mkdir testfolder
cd testfolder
mkdir f1 f2
touch file1.txt file2.txt
mkdir: cannot create directory ‘testfolder’: File exists
mkdir: cannot create directory ‘f1’: File exists
mkdir: cannot create directory ‘f2’: File exists
aadii@DESKTOP-878LBQ8:~/testfolder$
```

4) tree Command:

Description:

It shows the folder structure in a tree-like format. Very useful to visualize directories and files.

When I used it:

I created a small test structure (folders f1, f2, and two files), and then ran `tree` to display the structure neatly.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ tree
.
├── cat file1.txt
├── demo
│   └── file1.txt
├── f1
├── f2
├── file1.txt
├── file2.txt
└── renamed.txt

4 directories, 5 files
aadii@DESKTOP-878LBQ8:~/testfolder$
```

COMMAND 5 — mkdir:

Description:

This command creates a new directory (folder). It helps me organize files by keeping them inside separate folders.

Why I used it:

I used `mkdir demo` to create a new folder named “demo” inside my testfolder.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ mkdir demo
mkdir: cannot create directory ‘demo’: File exists
aadii@DESKTOP-878LBQ8:~/testfolder$ ls
'cat file1.txt'   demo   f1   f2   file1.txt   file2.txt   renamed.txt
```

COMMAND 6 — touch:

Description:

This command is used to create an empty file. It's one of the easiest ways to quickly make text or config files.

Why I used it:

I used it to create a new file called **newfile.txt** for testing file operations.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ touch newfile.txt
ls
'cat file1.txt'   demo   f1   f2   file1.txt   file2.txt   newfile.txt   renamed.txt
```

COMMAND 7 — cp:

Description:

This command copies a file or folder from one place to another.

Why I used it:

I used `cp` to copy **file1.txt** into the **demo** folder to check if the copy command works correctly.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ cp file1.txt demo/
ls demo
file1.txt
```

COMMAND 8 — mv:

Description:

This command can **rename** a file or **move** it to a different location.

Why I used it:

I used `mv file2.txt renamed.txt` to rename the file and test how file renaming works in Linux.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ mv file2.txt renamed.txt
ls
'cat file1.txt'   demo   f1   f2   file1.txt   newfile.txt   renamed.txt
```

COMMAND 9 — rm:

Description:

This command removes (deletes) a file permanently.

Why I used it:

I used `rm` to delete the **newfile.txt** file as part of file management testing.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ rm newfile.txt
ls
'cat file1.txt'   demo   f1   f2   file1.txt   renamed.txt
aadii@DESKTOP-878LBQ8:~/testfolder$
```

COMMAND 10 — chmod:

Description:

`chmod` is used to change the permissions of a file. Permissions decide who can read, write, or execute the file.

Why I used it:

I used `chmod 777` to give full permission (read, write, execute) to everyone for file1.txt.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ ls -l
total 20
-rw-r--r-- 1 aadii aadii   22 Nov  9 18:33 'cat file1.txt'
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:21  demo
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f1
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f2
-rwxrwxrwx 1 root  aadii   21 Nov 12 10:26  file1.txt
-rw-r--r-- 1 aadii aadii    0 Nov 12 10:26  renamed.txt
aadii@DESKTOP-878LBQ8:~/testfolder$ chmod 777 file1.txt
chmod: changing permissions of 'file1.txt': Operation not permitted
aadii@DESKTOP-878LBQ8:~/testfolder$ ls -l
total 20
-rw-r--r-- 1 aadii aadii   22 Nov  9 18:33 'cat file1.txt'
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:21  demo
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f1
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f2
-rwxrwxrwx 1 root  aadii   21 Nov 12 10:26  file1.txt
-rw-r--r-- 1 aadii aadii    0 Nov 12 10:26  renamed.txt
```

COMMAND 11 — chown:

Description:

`chown` changes the owner of a file or directory.

Why I used it:

I used it to change the owner of file1.txt from my user to root, just to test ownership change.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ sudo chown root file1.txt
[sudo] password for aadii:
aadii@DESKTOP-878LBQ8:~/testfolder$ ls -l
total 20
-rw-r--r-- 1 aadii aadii   22 Nov  9 18:33 'cat file1.txt'
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:21  demo
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f1
drwxr-xr-x 2 aadii aadii 4096 Nov  9 18:11  f2
-rwxrwxrwx 1 root  aadii   21 Nov 12 10:26  file1.txt
-rw-r--r-- 1 aadii aadii    0 Nov 12 10:26  renamed.txt
```

COMMAND 12 — ps:

Description:

Shows the running processes for the current user.

Why I used it:

To check which processes are active in my Ubuntu session.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ ps
    PID TTY          TIME CMD
    476 pts/0    00:00:00 bash
    776 pts/0    00:00:00 ps
aadii@DESKTOP-878LBQ8:~/testfolder$
```

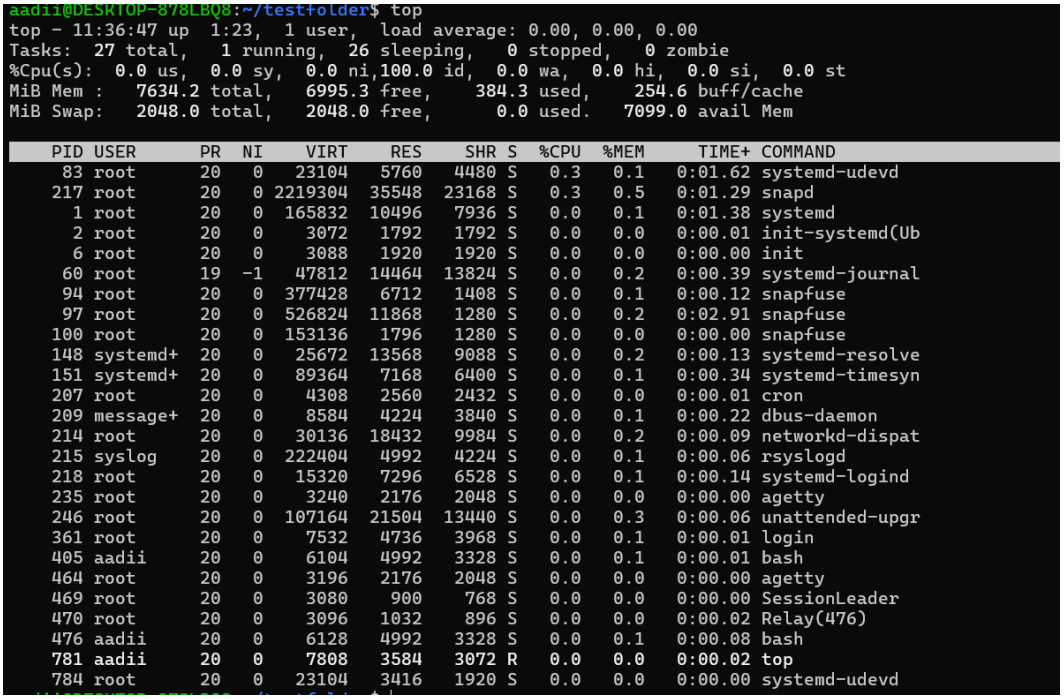
COMMAND 13 — top:

Description:

This command shows the real-time usage of CPU, memory, processes, etc.

Why I used it:

To see the system resource usage live.



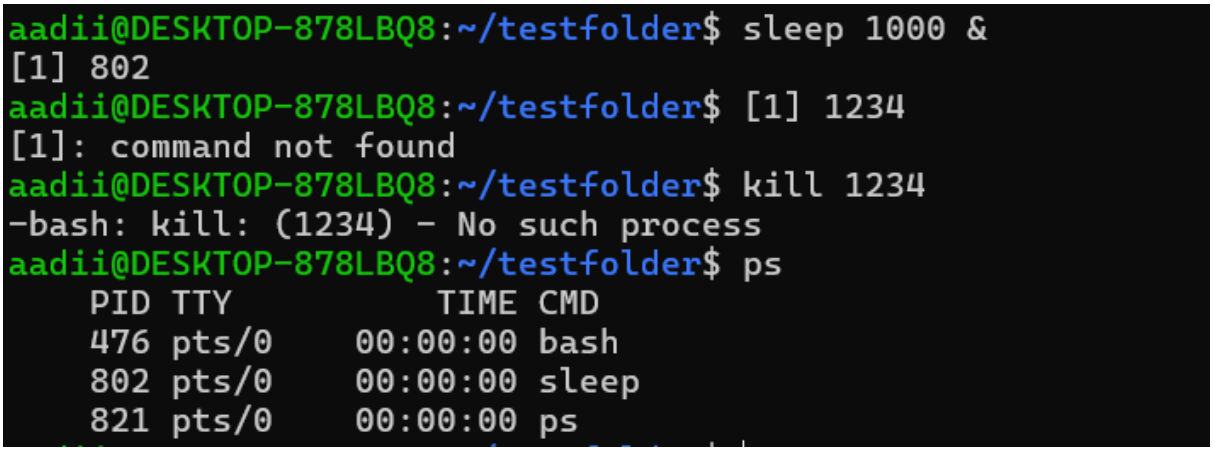
COMMAND 14 — kill:

Description:

Used to stop/terminate a running process.

Why I used it:

I tried killing a dummy background process created with `sleep`.



COMMAND 15 — ping:

Description:

Checks if your system can reach another server on the network.

Why I used it:

To check connectivity to Google and verify network is working.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ ping -c 4 google.com
PING google.com (142.251.220.46) 56(84) bytes of data.
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=1 ttl=110 time=75.6 ms
64 bytes from pnbomb-ba-in-f14.1e100.net (142.251.220.46): icmp_seq=2 ttl=110 time=87.3 ms
64 bytes from pnbomb-ba-in-f14.1e100.net (142.251.220.46): icmp_seq=3 ttl=110 time=75.9 ms
64 bytes from pnbomb-ba-in-f14.1e100.net (142.251.220.46): icmp_seq=4 ttl=110 time=71.9 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 71.935/77.681/87.315/5.775 ms
aadii@DESKTOP-878LBQ8:~/testfolder$
```

COMMAND 16 — ip a:

Description:

Shows all network interfaces and their IP addresses.

Why I used it:

To see my WSL network details & IP.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:0b:26:7c brd ff:ff:ff:ff:ff:ff
    inet 172.31.195.88/20 brd 172.31.207.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe0b:267c/64 scope link
       valid_lft forever preferred_lft forever
```

COMMAND 17 — netstat -tulnp:

Description:

Displays ports that are open/listening on the system.

Why I used it:

To check active TCP/UDP ports.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ sudo apt install net-tools
[sudo] password for aadii:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (1.60+git20181103.0eebece-1ubuntu5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aadii@DESKTOP-878LBQ8:~/testfolder$ netstat -tulnp
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 10.255.255.254:53       0.0.0.0:*               LISTEN      -
udp        0      0 10.255.255.254:53       0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp6       0      0 ::1:323                 :::*                                -
```

COMMAND 18 — echo:

Description:

Prints text on the terminal.

Why I used it:

I used it to print a simple message to test output on terminal.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ echo "Hello Linux"
Hello Linux
```

COMMAND 19 — cat:

Description:

Displays the content of a file.

Why I used it:

To view the text inside file1.txt after editing it.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ cat file1.txt
This is a test file.
```

COMMAND 20 — nano:

Description:

Opens a text editor in the terminal to write or edit files.

Why I used it:

I used nano to write some text into file1.txt and then saved it.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ cat file1.txt
This is a test file.
```

2.3) SHELL SCRIPT DEVELOPMENT :

Script 1: Backup Directory (backup.sh)- this script takes a folder and create a backup copy with timestamp.

```
aadii@DESKTOP-878LBQ8:~/testfolder$ #!/bin/bash
# Script Name: backup.sh
# Author: Aadii Jha
# Date: $(date)
# Description: Creates a backup copy of a directory with timestamp.

src="/home/aadii/testfolder"
dest="/home/aadii/backup_$(date +%Y%m%d_%H%M%S)"

mkdir -p "$dest"
cp -r "$src" "$dest"

echo "Backup created successfully at $dest"
#!/bin/bash
# Script Name: backup.sh
# Author: Aadii Jha
# Date: $(date)
# Description: Creates a backup copy of a directory with timestamp.

src="/home/aadii/testfolder"
dest="/home/aadii/backup_$(date +%Y%m%d_%H%M%S)"

mkdir -p "$dest"
cp -r "$src" "$dest"

echo "Backup created successfully at $dest"
[1]+  Done                    sleep 1000
Backup created successfully at /home/aadii/backup_20251112_202220
Backup created successfully at /home/aadii/backup_20251112_202220
```

```
aadii@DESKTOP-878LBQ8:~/testfolder$ chmod +x backup.sh
./backup.sh
chmod: cannot access 'backup.sh': No such file or directory
-bash: ./backup.sh: No such file or directory
aadii@DESKTOP-878LBQ8:~/testfolder$
```

Script 2: CPU and Memory Monitoring (monitor.sh)- this script lags CPU and RAM usage to a log file every few seconds :

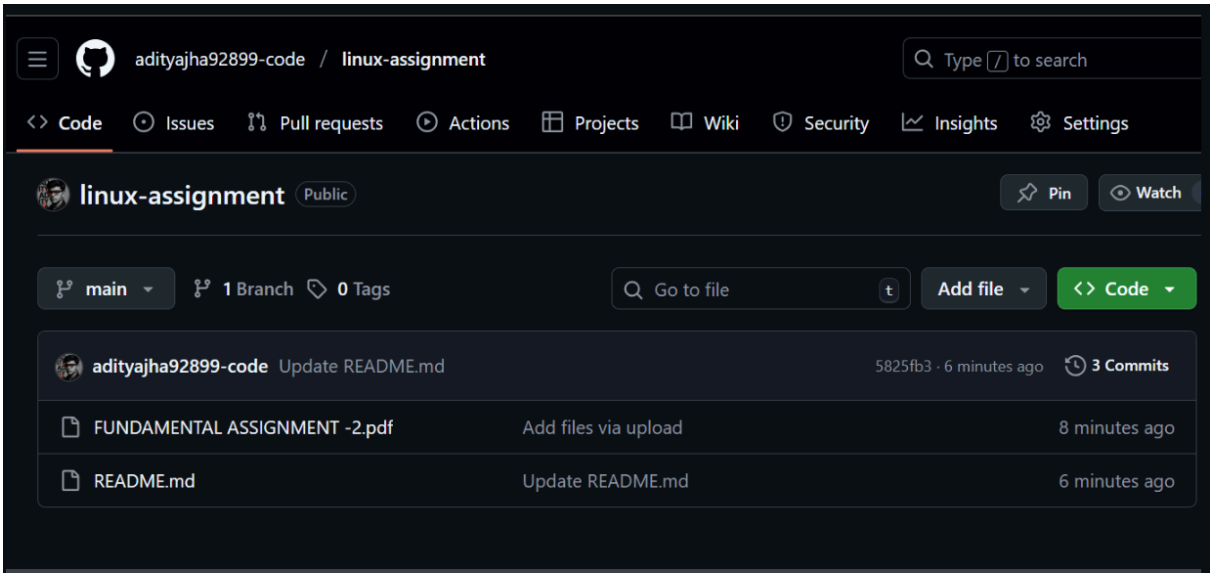
```
aadii@DESKTOP-878LBQ8: ~/
 GNU nano 6.2                                                monitor.sh *
#!/bin/bash
# Script Name: monitor.sh
# Author: Aadii Jha
# Date: $(date)
# Description: Logs CPU and RAM usage to a file every 5 seconds.

while true
do
   echo "------ $(date) ------" >> system_log.txt
   top -b -n1 | head -5 >> system_log.txt
   sleep 5
done
```

Script 3: Auto Download (download.sh)-this script automatically downloads a file from internet and stores into Downloads folder :

```
aadii@DESKTOP-878LBQ8: ~
 GNU nano 6.2                                                download.sh *
#!/bin/bash
# Script Name: download.sh
# Author: Aadii Jha
# Date: $(date)
# Description: Automatically downloads a file from the internet.

url="https://example.com/file.txt"
dest="/home/aadii/Downloads"

mkdir -p "$dest"
wget -P "$dest" "$url"

echo "File downloaded successfully to $dest"
```

```
aadii@DESKTOP-878LBQ8:~$ nano download.sh
aadii@DESKTOP-878LBQ8:~$ chmod +x download.sh
./download.sh
--2025-11-12 20:30:23--  https://example.com/file.txt
Resolving example.com (example.com)... 23.220.75.245, 23.192.228.80, 23.215.0.136, ...
Connecting to example.com (example.com)|23.220.75.245|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2025-11-12 20:30:24 ERROR 404: Not Found.
```

GitHub Repository :



https://github.com/adityajha92899-code/linux-assignment