

K. R. MANGALAM UNIVERSITY, GURUGRAM

Assignment: Basics of Linux and Open-Source Tools

Name – Anshika Kalhans

Roll No. - 2501010060

Course – B.Tech CSE Core

Course Name – Computer Science Fundamentals and Career Pathways

Course Code – ETCCCP105

INTRODUCTION

In this assignment, I learned about the basics of Linux and how it is used in the field of computer science. Linux is an open-source operating system that is widely used because of its security, flexibility, and command-line features. It helps users perform different tasks using simple commands and scripts instead of only relying on graphical interfaces.

The main aim of this project is to install Linux, understand different shell commands, and create automation scripts to perform tasks like taking backups, checking CPU usage, and downloading files. While doing this project, I also learned how to upload my work to GitHub, which is useful for saving and sharing code.

This assignment helped me understand how Linux works in real-life situations and how shell scripting can make daily computer tasks easier and faster.

Step 1 – Linux Installation

Method Used:

I installed Ubuntu Linux using Windows Subsystem Linux (WSL) on my Windows 10/11 system.

Hardware Configuration:

Component	Specification
Processor (CPU)	Intel(R) Core (TM) Ultra 7 155H
RAM	32.0 GB
Disk Space Allocated for WSL	20 – 25 GB
Operating System (Host)	Windows 11
WSL Version	WSL 2
Distribution Used	Ubuntu 24.04.3 LTS (Noble Numbat)

Installation Steps:

Step 1: Enable WSL

- Open PowerShell as Administrator
- Run the following command to enable Windows Subsystem for Linux:
wsl --install ubuntu
- This installs all required components (WSL, Virtual Machine Platform, and Ubuntu by default.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
ws1 --install -d ubuntu
>> C:\WINDOWS\system32>
Downloading: Ubuntu
[===== 12.7% ]
```

Step 2: First Launch of Ubuntu

- Search Ubuntu in the search bar, then click open.

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
anshikakalhans@LAPTOP-LK2D74LF:~$ |
```

Step 3: Create a User Account

- Once setup completes, it'll ask to create:
 Username
 Password
- This will be a Linux user for the Ubuntu environment.

Step 2: Shell Command Implementation and Documentation

Section A: File Navigation Commands (pwd, ls, cd, tree)

1. **pwd**

Syntax: pwd

Description: Displays the current working directory.

When and Why: Used to confirm your current directory before performing other file operations.

Sample Output:

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
anshikakalhans@LAPTOP-LK2D74LF:~$ pwd  
/home/anshikakalhans  
anshikakalhans@LAPTOP-LK2D74LF:~$
```

2.ls

Syntax: ls

Description: Lists all files and directories in the current folder.

When and Why: Used to view what's inside the current directory.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls  
test  
anshikakalhans@LAPTOP-LK2D74LF:~$ |
```

3.cd

Syntax: cd directory_name

Description: Changes the current working directory.

When and Why: Used to move between folders.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls
myfile.txt myfolder test
anshikakalhans@LAPTOP-LK2D74LF:~$ cd myfolder
anshikakalhans@LAPTOP-LK2D74LF:~/myfolder$ pwd
/home/anshikakalhans/myfolder
anshikakalhans@LAPTOP-LK2D74LF:~/myfolder$ ls
anshikakalhans@LAPTOP-LK2D74LF:~/myfolder$ |
```

4.tree

Syntax: tree

Description: Displays files and directories in a tree structure.

When and Why: Helpful for visualizing folder structure.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ tree myfolder -L 2
myfolder
└── folder_name

2 directories, 0 files
anshikakalhans@LAPTOP-LK2D74LF:~$ |
```

Section B: File and Directory Management Commands (mkdir, touch, cp, mv, rm)

5.mkdir

Syntax: mkdir folder_name

Description: Creates a new directory.

When and Why: Used to create new folders for organizing files.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ mkdir this
anshikakalhans@LAPTOP-LK2D74LF:~$ pwd
/home/anshikakalhans
anshikakalhans@LAPTOP-LK2D74LF:~$
```

6.touch

Syntax: touch filename

Description: Creates a new empty file or updates the timestamp of an existing file.

When and Why: Used to quickly make files for testing or notes.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls
touch demo.txt
ls
backup backup.sh download.sh file1.txt monitor.sh myfile.txt myfolder test this world
backup backup.sh demo.txt download.sh file1.txt monitor.sh myfile.txt myfolder test this world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

7.cp

Syntax: cp source destination

Description: Copies files or folders.

When and Why: Used to duplicate files safely.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls
cp file1.txt copyfile.txt
ls
backup backup.sh demo.txt download.sh file1.txt monitor.sh myfile.txt myfolder test this world
backup backup.sh copyfile.txt demo.txt download.sh file1.txt monitor.sh myfile.txt myfolder test this world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

8.mv

Syntax: mv source destination

Description: Moves or renames files/directories.

When and Why: Used to move files or rename them.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls
mv file1.txt renamedfile.txt
ls
backup backup.sh copyfile.txt demo.txt download.sh file1.txt monitor.sh myfile.txt myfolder test this world
backup backup.sh copyfile.txt demo.txt download.sh monitor.sh myfile.txt myfolder renamedfile.txt test this world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

9.rm

Syntax: rm filename

Description: Removes(deletes) files or folders.

When and Why: Used to delete unwanted files.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls
rm sample.txt
ls
backup backup.sh copyfile.txt demo.txt download.sh monitor.sh myfile.txt myfolder renamedfile.txt test this world
rm: cannot remove 'sample.txt': No such file or directory
backup backup.sh copyfile.txt demo.txt download.sh monitor.sh myfile.txt myfolder renamedfile.txt test this world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

Section C: Permissions Management Commands (chmod, chown)

10.chmod

Syntax: chmod [permissions] filename

Description: Changes file permissions (read, write, execute).

When and Why: Used to control who can access or modify files.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls -l
chmod +x file.sh
ls -l
total 32
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 18:50 backup
-rw xr-xr-x 1 anshikakalhans anshikakalhans 837 Nov 13 18:48 backup.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 05:01 copyfile.txt
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:58 demo.txt
-rw-r--r-- 1 anshikakalhans anshikakalhans 866 Nov 13 18:59 download.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 1045 Nov 13 18:55 monitor.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 13 08:20 myfile.txt
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 09:56 myfolder
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:57 renamedfile.txt
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 13 04:22 test
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 this
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 world
chmod: cannot access 'file.sh': No such file or directory
total 32
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 18:50 backup
-rw xr-xr-x 1 anshikakalhans anshikakalhans 837 Nov 13 18:48 backup.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 05:01 copyfile.txt
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:58 demo.txt
-rw-r--r-- 1 anshikakalhans anshikakalhans 866 Nov 13 18:59 download.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 1045 Nov 13 18:55 monitor.sh
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 13 08:20 myfile.txt
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 09:56 myfolder
-rw-r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:57 renamedfile.txt
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 13 04:22 test
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 this
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

11.chown

Syntax: sudo chown user:group filename

Description: Changes ownership of a file.

When and Why: Used when a file needs to belong to another user.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls -l
sudo chown yourname file.txt
ls -l
total 32
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 18:50 backup
-rw xr-xr-x 1 anshikakalhans anshikakalhans 837 Nov 13 18:48 backup.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 05:01 copyfile.txt
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:58 demo.txt
-rw r--r-- 1 anshikakalhans anshikakalhans 866 Nov 13 18:59 download.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 1045 Nov 13 18:55 monitor.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 13 08:20 myfile.txt
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 09:56 myfolder
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:57 renamedfile.txt
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 13 04:22 test
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 this
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 world
[sudo] password for anshikakalhans:
chown: invalid user: 'yourname'
total 32
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 18:50 backup
-rw xr-xr-x 1 anshikakalhans anshikakalhans 837 Nov 13 18:48 backup.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 05:01 copyfile.txt
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:58 demo.txt
-rw r--r-- 1 anshikakalhans anshikakalhans 866 Nov 13 18:59 download.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 1045 Nov 13 18:55 monitor.sh
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 13 08:20 myfile.txt
drwxr-xr-x 3 anshikakalhans anshikakalhans 4096 Nov 13 09:56 myfolder
-rw r--r-- 1 anshikakalhans anshikakalhans 0 Nov 14 04:57 renamedfile.txt
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 13 04:22 test
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 this
drwxr-xr-x 2 anshikakalhans anshikakalhans 4096 Nov 14 04:54 world
anshikakalhans@LAPTOP-LK2D74LF:~$
```

Section D: Process Monitoring (ps, top, kill)

12.ps

Syntax: ps

Description: Displays a list of active processes.

When and Why: Used to check running programs or background processes.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ps
  PID TTY      TIME CMD
  683 pts/0    00:00:00 bash
  966 pts/0    00:00:00 ps
anshikakalhans@LAPTOP-LK2D74LF:~$
```

13.top

Syntax: top

Description: Displays real-time system processes (like Task Manager in Windows).

When and Why: Used to monitor CPU and memory usage live.

Sample Output:

System Resource Utilization Report											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	21848	12132	9140	S	0.0	0.1	0:00.77	systemd
2	root	20	0	3072	1584	1584	S	0.0	0.0	0:00.01	init-systemd(Ub)
8	root	20	0	3072	1760	1760	S	0.0	0.0	0:00.00	init
44	root	19	-1	66744	17036	16156	S	0.0	0.1	0:00.29	systemd-journal
91	root	20	0	25508	6864	4928	S	0.0	0.0	0:00.89	systemd-udevd
106	systemd+	20	0	21456	11968	10032	S	0.0	0.1	0:00.12	systemd-resolve
107	systemd+	20	0	91024	7216	6512	S	0.0	0.0	0:00.33	systemd-timesyn
187	root	20	0	4236	2464	2288	S	0.0	0.0	0:00.02	cron
188	message+	20	0	9636	4928	4400	S	0.0	0.0	0:00.11	dbus-daemon
195	root	20	0	17964	8448	7568	S	0.0	0.1	0:00.11	systemd-logind
200	root	20	0	3160	1760	1760	S	0.0	0.0	0:00.01	agetty
213	root	20	0	3116	1760	1760	S	0.0	0.0	0:00.00	agetty
221	syslog	20	0	222508	5456	4400	S	0.0	0.0	0:00.09	rsyslogd
236	root	20	0	107028	22528	13200	S	0.0	0.1	0:00.07	unattended-upgr
319	root	20	0	6696	4048	3520	S	0.0	0.0	0:00.00	login
362	anshika+	20	0	20320	11088	9152	S	0.0	0.1	0:00.10	systemd
363	anshika+	20	0	21148	3440	1760	S	0.0	0.0	0:00.00	(sd-pam)
378	anshika+	20	0	6072	5104	3520	S	0.0	0.0	0:00.02	bash
647	polkitd	20	0	308164	7568	6864	S	0.0	0.0	0:00.13	polkitd
678	root	20	0	3080	880	880	S	0.0	0.0	0:00.00	SessionLeader
679	root	20	0	3096	1056	1056	S	0.0	0.0	0:00.02	Relay(683)
683	anshika+	20	0	6072	5104	3520	S	0.0	0.0	0:00.06	bash
975	anshika+	20	0	9284	5456	3344	R	0.0	0.0	0:00.03	top

14.kill

Syntax: kill process_id

Description: Terminates a running process using its PID.

When and Why: Used to stop unresponsive or background processes.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ps
kill <PID>
ps
  PID TTY          TIME CMD
  994 pts/0    00:00:00 bash
  1007 pts/0    00:00:00 ps
-bash: syntax error near unexpected token `newline'
  PID TTY          TIME CMD
  994 pts/0    00:00:00 bash
  1008 pts/0    00:00:00 ps
anshikakalhans@LAPTOP-LK2D74LF:~$
```

Section E: Networking Tools (ping, ifconfig / ip, netstat)

15.ping

Syntax: ping website_or_ip

Description: Sends data packets to test network connectivity.

When and Why: Used to check if your internet or website is reachable.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ping -c 4 google.com
PING google.com (142.251.223.206) 56(84) bytes of data.
64 bytes from tzdela-as-in-f14.1e100.net (142.251.223.206): icmp_seq=1 ttl=116 time=5.78 ms
64 bytes from tzdela-as-in-f14.1e100.net (142.251.223.206): icmp_seq=2 ttl=116 time=13.7 ms
64 bytes from tzdela-as-in-f14.1e100.net (142.251.223.206): icmp_seq=3 ttl=116 time=8.42 ms
64 bytes from tzdela-as-in-f14.1e100.net (142.251.223.206): icmp_seq=4 ttl=116 time=17.9 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3229ms
rtt min/avg/max/mdev = 5.776/11.441/17.870/4.681 ms
anshikakalhans@LAPTOP-LK2D74LF:~$
```

16.ifconfig / ip

Syntax: ifconfig

or

ip a

Description: Displays network interface details (IP address, etc.).

When & Why: Used to check or troubleshoot your network connection.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ 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 1450 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:75:da:44 brd ff:ff:ff:ff:ff:ff
        inet 172.17.77.28/20 brd 172.17.79.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::215:5dff:fe75:da44/64 scope link
            valid_lft forever preferred_lft forever
anshikakalhans@LAPTOP-LK2D74LF:~$
```

17.netstat

Syntax: netstat

Description: Displays network connections and ports in use.

When & Why: Used for monitoring active network connections.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ netstat -tuln
Command 'netstat' not found, but can be installed with:
sudo apt install net-tools
anshikakalhans@LAPTOP-LK2D74LF:~$
```

Some Extra Commands

18.man

Syntax: man command_name

Description: Displays manual/help page for a command.

When and Why: Used to learn what a command does and its options.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ man ls | head
LS(1)                                         User Commands                                         LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by default). Sort entries alphabetically if none of
anshikakalhans@LAPTOP-LK2D74LF:~$ |
```

19.clear

Syntax: clear

Description: Clears the terminal screen.

When & Why: Used to keep your workspace tidy.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ ls  
clear
```

```
anshikakalhans@LAPTOP-LK2D74LF:~$
```

20.history

Syntax: history

Description: Shows a list of recently used commands.

When & Why: Used to repeat or check past commands.

Sample Output:

```
anshikakalhans@LAPTOP-LK2D74LF:~$ history 10  
101 ping -c 4 google.com  
102 ip a  
103 netstat -tuln  
104 man ls  
105 man ls | head  
106 ls  
107 clear  
108 ls  
109 ls clear  
110 history 10  
anshikakalhans@LAPTOP-LK2D74LF:~$
```

Step 3: Shell Script Development

Script 1: Backup a Directory (backup.sh)

```
#!/bin/bash
#
# -----
# Script Name : backup.sh
# Purpose     : To backup a specified directory to a backup folder with a timestamp.
# Author      : <Your Name>
# Date        : <Date>
# -----
#
# ----- Variables -----
SOURCE_DIR="$1"                                # Directory to back up (passed as argument)
BACKUP_DIR="$HOME/backup"                         # Backup destination directory
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")              # Current date and time
DEST_DIR="$BACKUP_DIR/backup_${TIMESTAMP}"         # Folder name with timestamp

# ----- Main Script -----
# Create backup directory if it doesn't exist
mkdir -p "$DEST_DIR"

# Copy files
cp -r "$SOURCE_DIR"/* "$DEST_DIR"

echo "Backup of '$SOURCE_DIR' completed successfully at $DEST_DIR"
|
```

Script 2: CPU/Memory Monitoring (monitor.sh)

```

#!/bin/bash
#
# Script Name : monitor.sh
# Purpose      : Logs CPU and memory usage to a file at regular intervals.
# Author       : <Your Name>
# Date        : <Date>
#
# -----
#
# ----- Variables -----
LOG_FILE="$HOME/system_usage.log"    # File where data will be saved
INTERVAL=5                           # Interval in seconds between readings

# ----- Main Script -----
echo "Logging CPU and Memory usage every $INTERVAL seconds..."
echo "Timestamp, CPU(%), Memory(%)" > "$LOG_FILE"

# Infinite loop that records data periodically
while true
do
    TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")           # Current time
    CPU=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}') # CPU usage %
    MEM=$(free | grep Mem | awk '{print $3/$2 * 100.0}')   # Memory usage %
    echo "$TIMESTAMP, $CPU, $MEM" >> "$LOG_FILE"          # Append to file
    sleep $INTERVAL                                     # Wait before next log
done
|

```

Script 3: Automated Download Task (download.sh)

```
#!/bin/bash
#
# -----
# Script Name : download.sh
# Purpose     : Downloads a file from the internet and stores it in a predefined directory.
# Author      : <Your Name>
# Date        : <Date>
# -----
#
# ----- Variables -----
URL="$1"                                # URL of the file to download (passed as argument)
DOWNLOAD_DIR="$HOME/downloads"           # Directory where downloaded files will be stored

# ----- Main Script -----
# Create the download directory if it doesn't exist
mkdir -p "$DOWNLOAD_DIR"

# Check if URL is provided
if [ -z "$URL" ]; then
    echo "Usage: ./download.sh <file_URL>"
    exit 1
fi

# Download the file using wget
wget -P "$DOWNLOAD_DIR" "$URL"

# Print confirmation message
echo "✓ File downloaded successfully to: $DOWNLOAD_DIR"
```