Assignment 02 — Basics of Linux and Open-Source Tools

Course: ETCCCP105

— Computer Science Fundamentals & Career Pathways

Name: *Agrima Mishra*
Student ID: *2501010207*
Section : B
Semester: Odd
Date of submission: *18 – Nov – 2025*

Link: Git Hub link

Introduction

Linux is one of the most important operating systems used in the fields of programming, networking, cybersecurity, and servers. Unlike Windows, Linux works mainly through commands, which makes it faster and more flexible. In this assignment, I learned how to install Linux using Windows Subsystem for Linux (WSL), which allowed me to run Ubuntu directly on my Windows laptop without needing a separate computer or virtual machine. After installation, I explored different Linux commands such as creating folders, copying and moving files, checking system information, managing processes, and viewing network details.

I also learned how to write simple shell scripts. These scripts helped me automate tasks like taking backups, monitoring the system, and downloading files. Working with the Linux terminal gave me practical experience, and I got a clear idea of how real programmers and cybersecurity professionals use Linux every day. This introduction to Linux has helped me understand the basics in a simple and hands-on way.

COMMAND TABLE

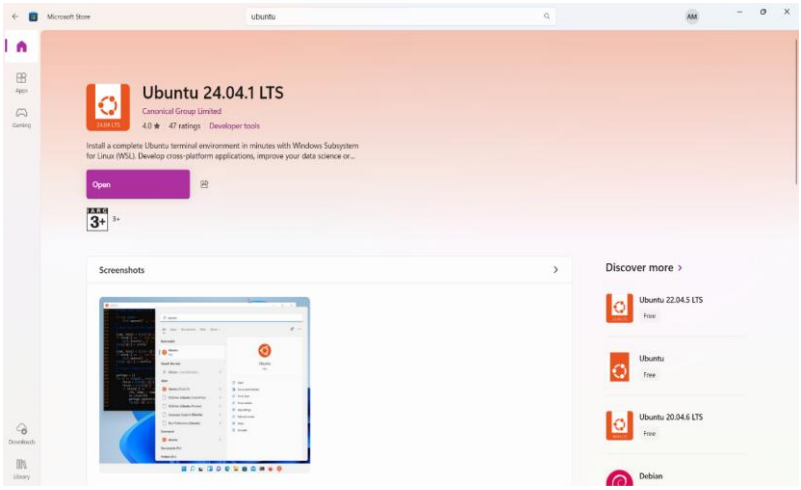
Command (syntax)	Short description	When / why used
pwd	Print Working Directory — shows full path of current directory.	Confirm where you are in the filesystem before running file operations or giving paths.
ls / ls -l / ls -a	List files and directories (long format, show hidden files).	Inspect directory contents, check file names, permissions, sizes and hidden files.
cd <dir> / cd // cd	Change directory (cd with no args → home).	Move around the filesystem to the directory where you want to work.
tree or ls -R	Show directory tree recursively.	Get a quick overview of folder structure (useful before backups or packaging).
mkdir <dir>	Create a new directory.	Create project folders, output directories, or structure for scripts.
touch <file>	Create an empty file (or update timestamp).	Quickly create placeholder files or update file timestamps.
cp <src> <dest>	Copy files or directories (cp -r for directories).	Make backups or duplicate files before editing.
mv <old> <new>	Move or rename files/directories.	Rename files or move them into other directories for organization.
rm <file> / rm -r <dir>	Remove file or directory (recursive).	Delete unwanted files/folders — use carefully (no recycle bin).
chmod <mode> <file>	Change file permissions (e.g., chmod 755 file).	Make scripts executable, restrict or allow read/write/execute access.

Command (syntax)	Short description	When / why used
ps / ps aux	List processes running in current shell or system.	Find PIDs to manage or investigate running processes.
top	Real-time interactive process monitor (CPU/memory).	Troubleshoot high CPU/memory usage and see active processes live.
sleep <seconds> &	Run sleep in background (use & to background).	Create a simple background job for testing background/kill behavior.
kill <PID> / kill -9 <PID>	Send signal to process (terminate).	Stop misbehaving or test processes by PID; -9 forces kill if needed.
ping <host> -c <n>	Send ICMP echo requests (e.g., ping google.com -c 4).	Check network reachability and round-trip latency to a host.
ip a (or ip addr show)	Show network interfaces and IP addresses.	Verify IP addresses, interfaces status, and troubleshoot networking.
netstat -tulpn (or ss -tulpn)	Show listening ports and active network sockets.	Find which services are bound to which ports for troubleshooting.
date	Display current date and time (formatable).	Timestamp logs, name backup files, or verify system clock.
df -h	Show disk free/used space (human-readable).	Check available storage before copying, backing up, or installing packages.
uname -a	Show kernel and system information (OS, kernel version, arch).	Document system details for reports, debugging or compatibility checks.
history	Show shell command history.	Provide an audit trail of commands you ran for documentation or grading.

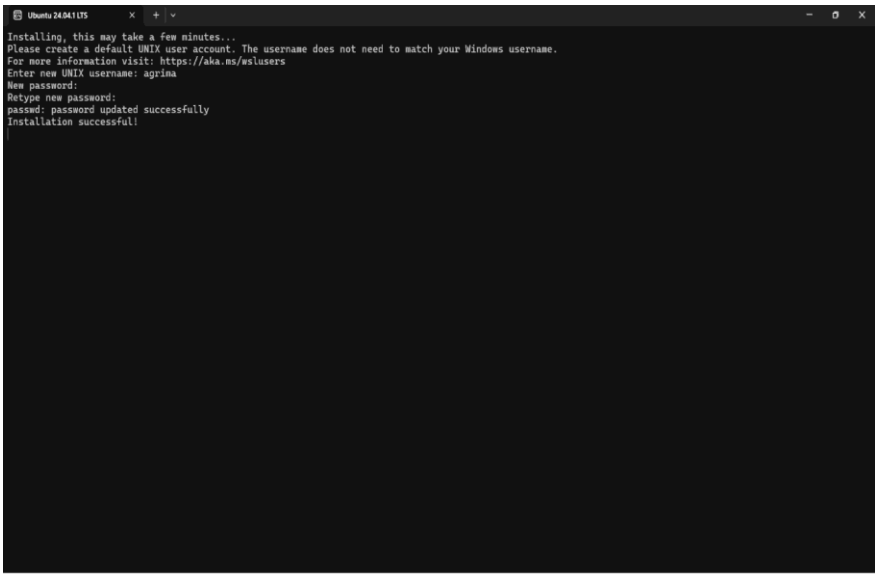
LINUX INSTALLATION DOCUMENTATION

Platform used: Windows Subsystem for Linux (WSL) — Ubuntu 24.04 LTS

Step 1- Installed Ubuntu 24.04 from Microsoft Store.



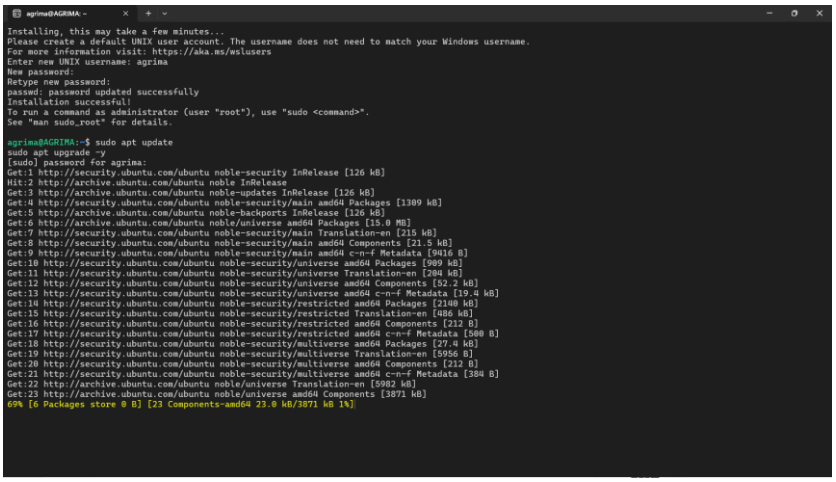
Step 2- Opening Ubuntu installing and creating the username for the first terminal login.

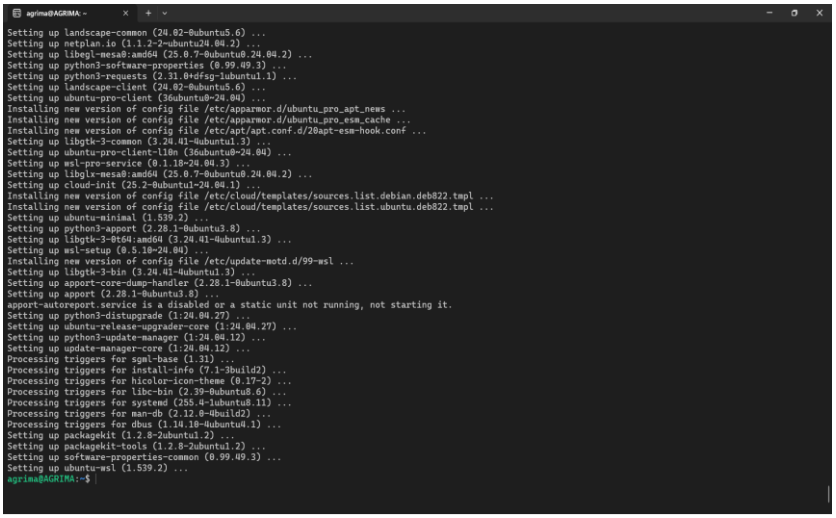


Step 3- Updating the Ubuntu by using the following commands

sudo apt update

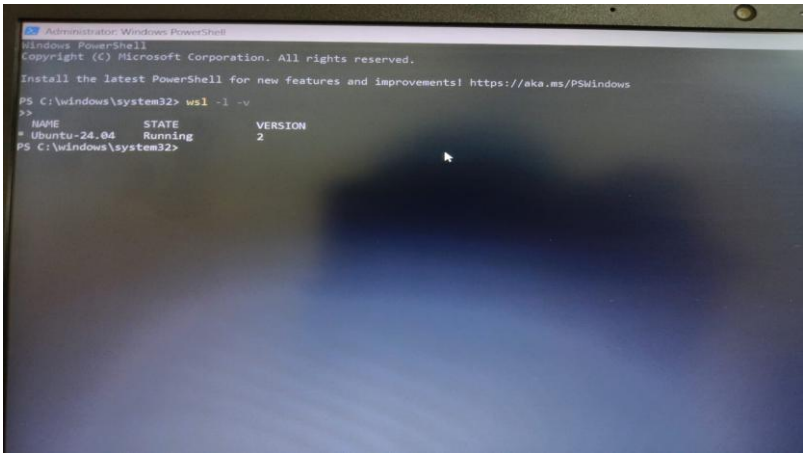
sudo apt upgrade -y





Step 4- Checking WSL version using the command

wsl -l -v



LINUX COMMANDS

1. File Navigation:

1). pwd- Print Working Directory — shows current directory path.

Used - Use to confirm your current location in filesystem.

2). ls - List files in the directory. examples: ls -l, ls -a

Used - Inspect files and directories.

Commented [AM1]: ls shows files except hidden ones while ls -a command shows all the files including the hidden ones

3). cd/ - Change directory. cd / goes to root.

Used - Navigate the filesystem.

4). cd is used to **change directory**, return to home directory.

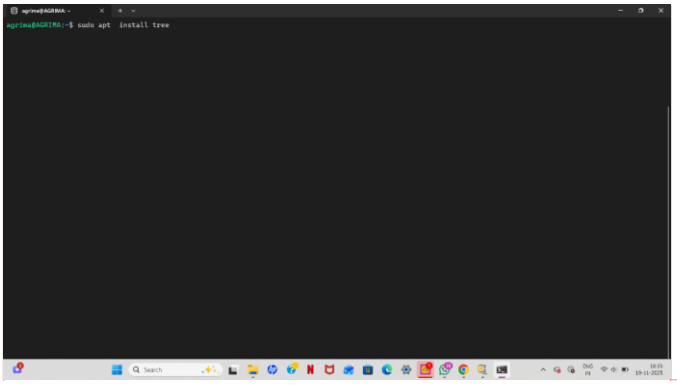
Used – Navigate filesystem with ease accessing directory and exiting back to home directory through this command.

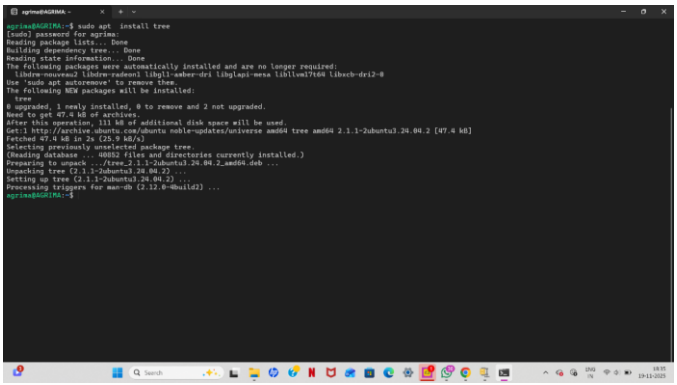
Commented [AM2]: Cd / is root directory while cd is home directory

5). tree –

Installing the demo tree consisting sample directories and files in tree form.

-- By **sudo apt install tree** command





Commented [AM3]: Used a tree installing command to show tree which are files and folders with the root directory or parent they are in

File & Directory Management Commands:

1). mkdir- make directory.

mkdir testfolder created a folder.

Syntax- mkdir <folder name>

Used- Create project or storage directories.

2). touch - Create an empty file.

Syntax- touch <file>

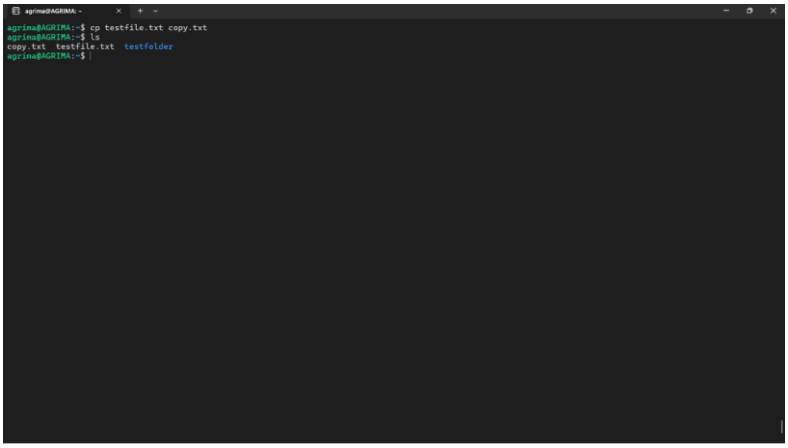
--touch testfile.txt.

Used- Quickly create files to test or use as placeholders.

3). cp – Copy files

Syntax- cp source dest

Used- Duplicate files for backup or editing safely.

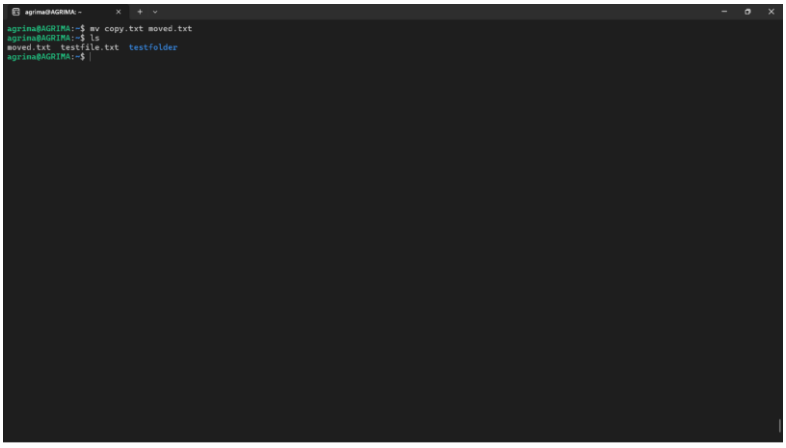


4). mv - Move or rename files.

Rename files or relocate them.

Syntax- mv old.txt new.txt

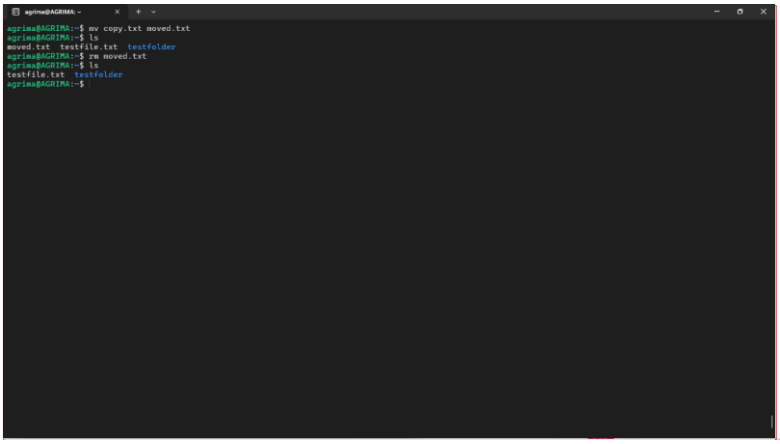
--- mv copy.txt moved.txt



5). rm – remove files

Syntax- rm <file>

Delete unwanted files



Commented [AM4]: rm is used to remove a file but the individual has to be careful cause the deleted file won't be retrieved back again through any means

PERMISSIONS COMMANDS:

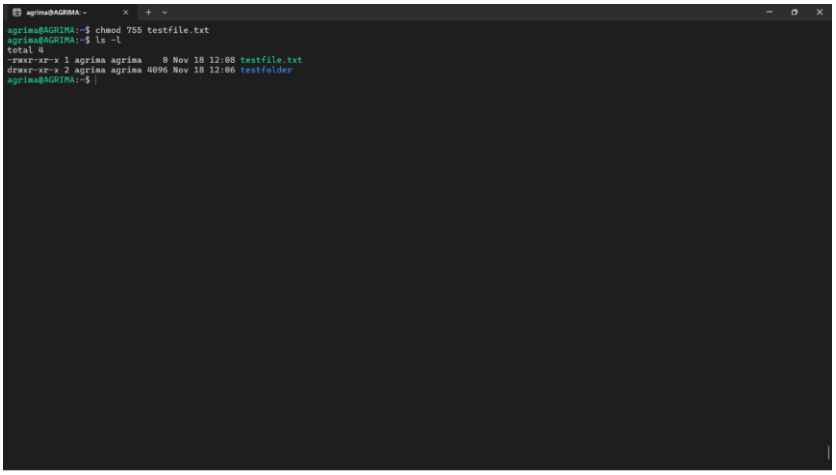
Commented [AM5]: These commands are very sensitive for security and privacy related task as it gives access to ones system and the charge which might create troubles and serious consequences if not handled sensitively and carefulluy hence one mustnt share or allow any of these without proper thinking or knowledge

1). chmod- change permissions.

chmod 755 testfile.txt sets rwx r-x r-x.

Syntax- chmod 755 file

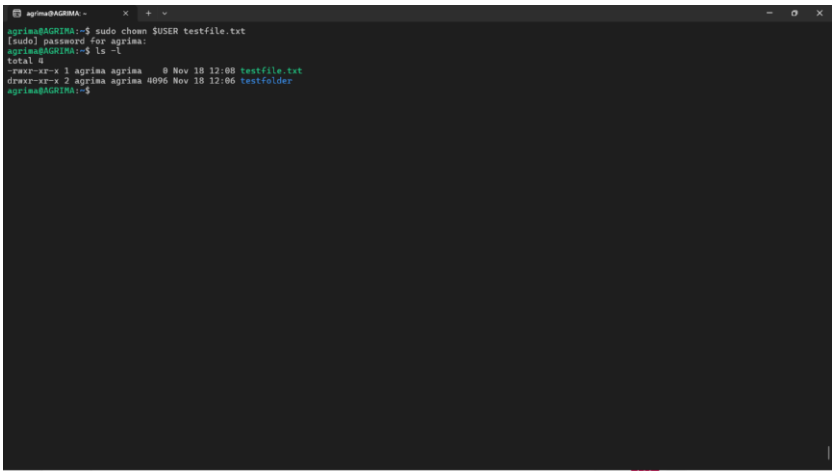
Controls read (r), write (w), execute (e) permissions



2). chown- change the ownership of the file

Syntax- sudo chown $USER file

Fix ownership issues or set for deployment.



PROCESS MONITORING COMMANDS

1). ps - Show current shell processes.

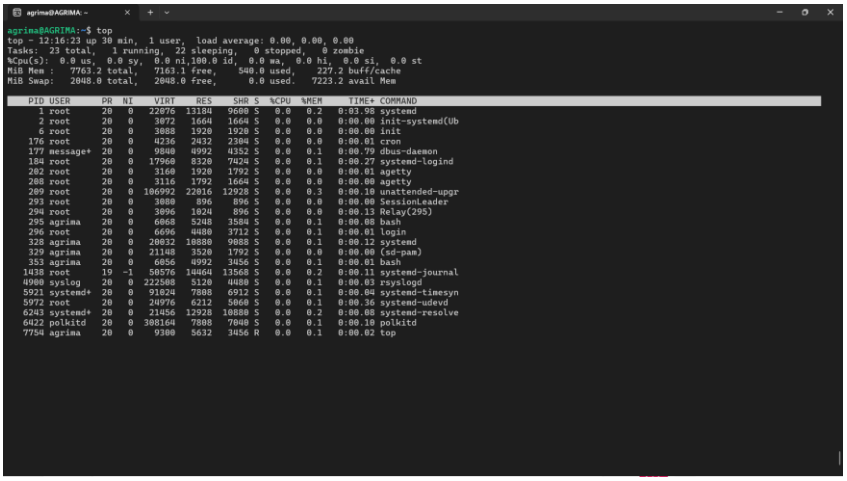
ps (or ps aux)

Check running processes and their PIDs.

2). top- Real-time process monitor (or htop if installed).

top

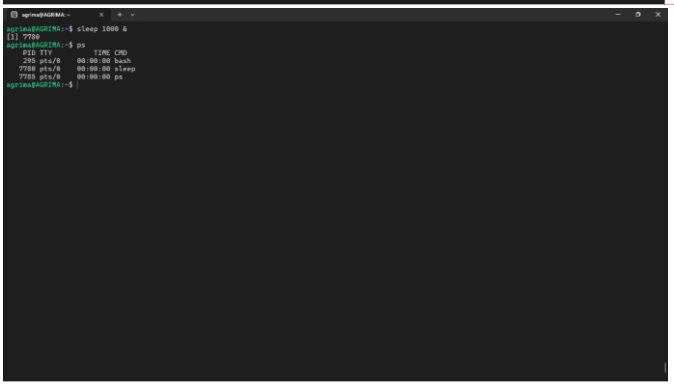
Monitor CPU, memory and running processes**.**



3). Kill – starts a background job and kills it

Syntax- sleep 1000 & + kill <PID>

Demonstrates starting background tasks and stopping them safely.



Commented [AM6]: A fake background task is started through sleep command to perform the kill command

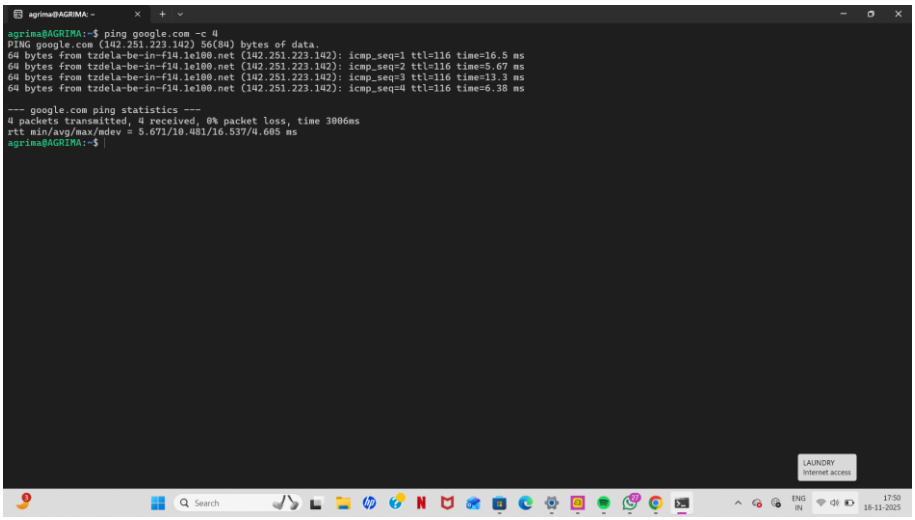
Commented [AM7]: Start a long-running background process

NETWORKING COMMANDS:

1). ping- Send ICMP echo requests to check connectivity**.**

ping google.com -c 4

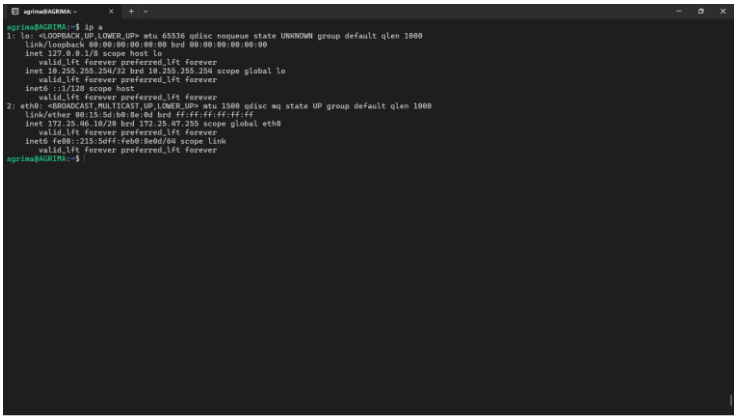
Test network reachability and latency.



2). ip - show network interfaces & IP addresses (modern replacement for ifconfig).

Syntax- ip a

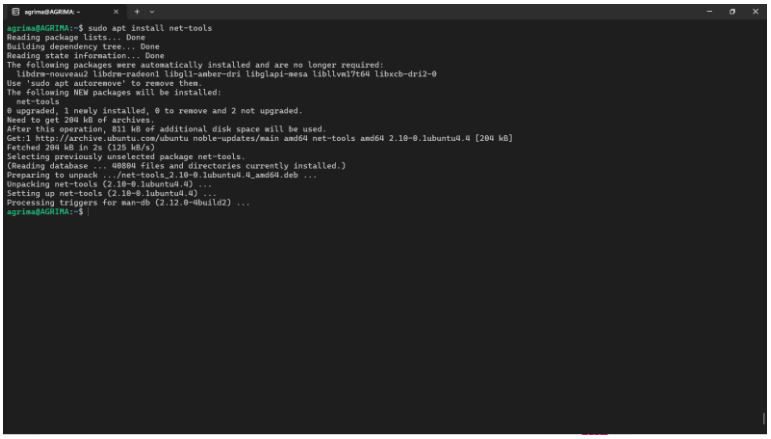
Check IP, interface status.

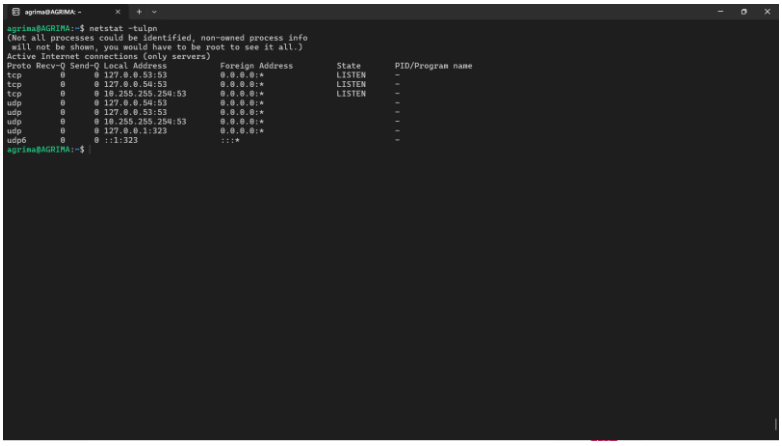


3). netstat -tulpn - Show active listening ports (install net-tools if needed).

Syntax- netstat -tulpn

find services bound to ports and troubleshoot network services.





4). date – Show current date/time**.**

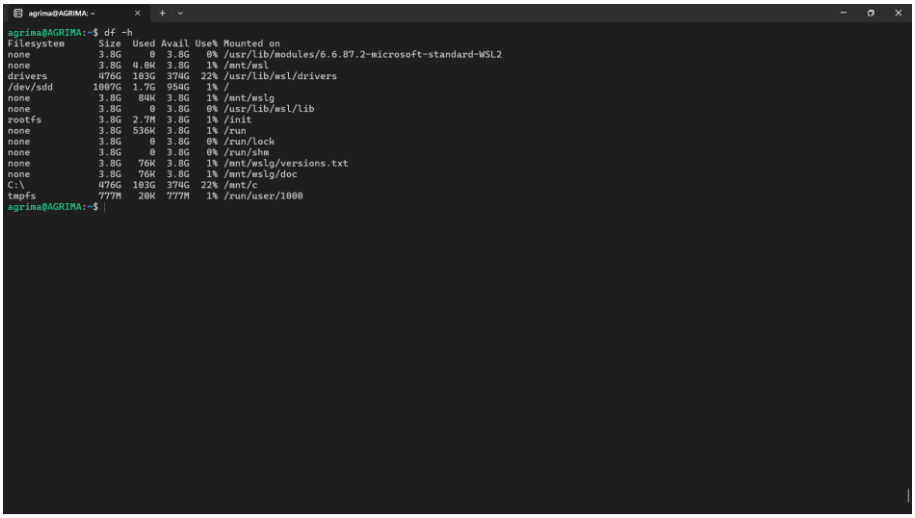
Syntax- date

Log timestamps; used in scripts for naming backups.

5). df -h - disk usage (human readable)

Syntax- df -h

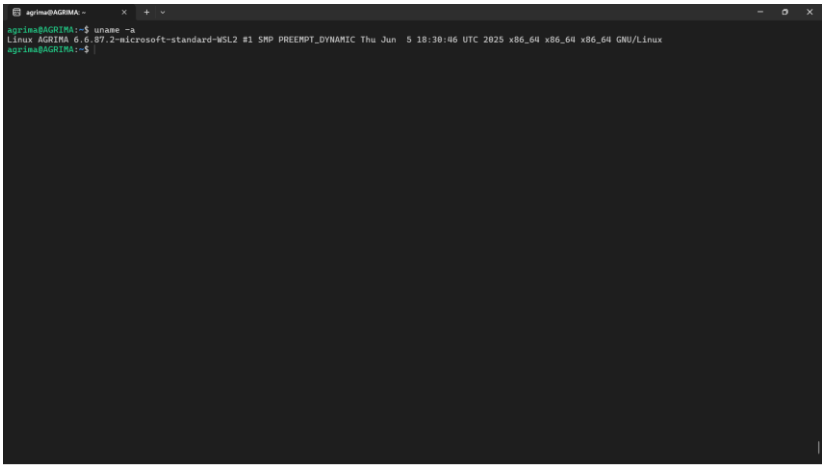
Check free space before large operations like backups.



6). uname -a - Show kernel and system info.

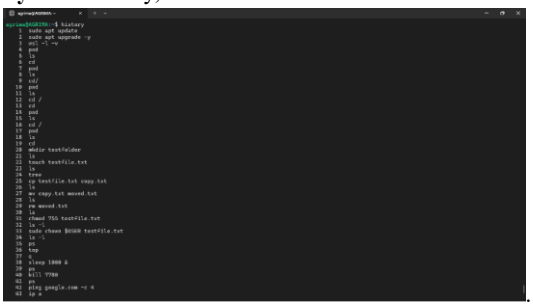
Syntax- uname -a

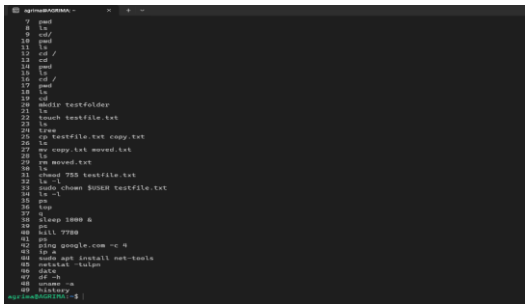
Record system details for documentation or debugging.



7). history- Show command history (useful evidence).

Syntax- history, **use-** Provide an audit trail of commands you executed



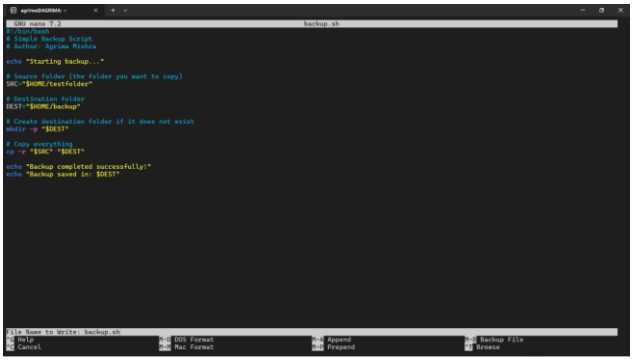


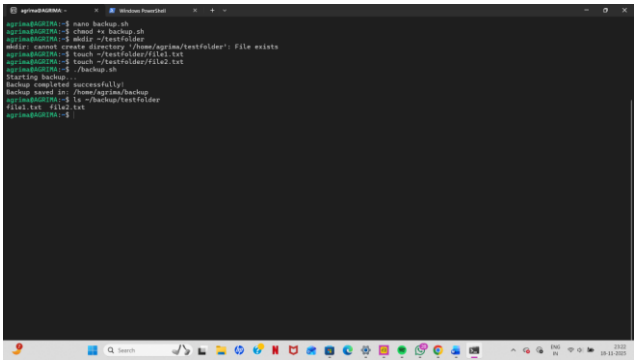
Script Shells

1). Backup a directory - backup.sh

Purpose: copy a specified folder into a backups folder named with timestamp.

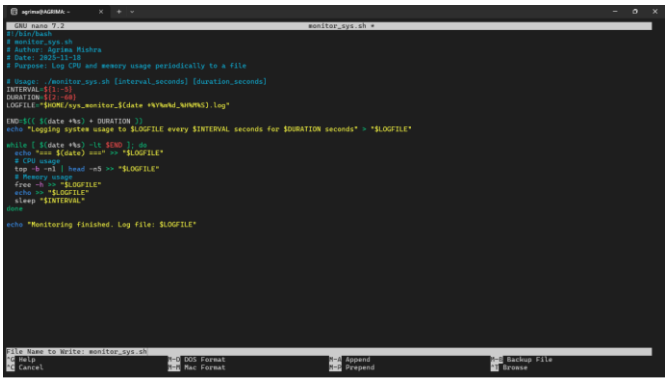
This script copies a folder called **testfolder** into a folder called **backup** inside your home directory.





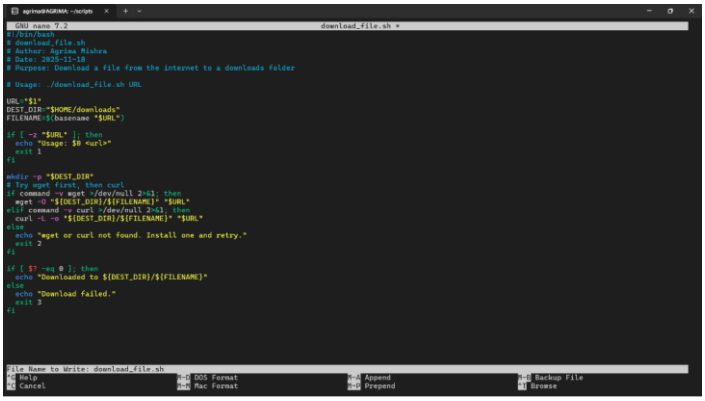
2). **CPU/Memory monitor — monitor_sys.sh**

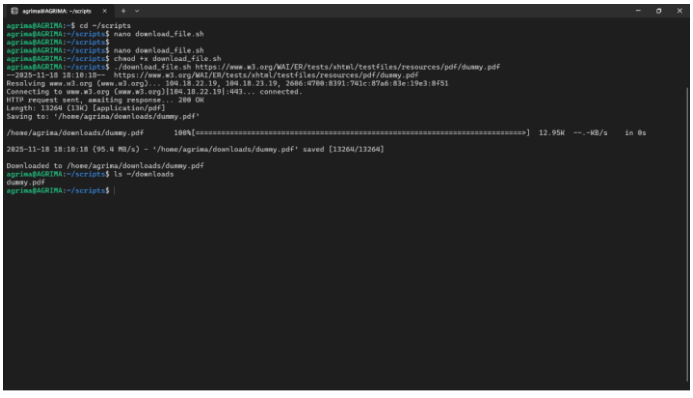
Purpose: log CPU and memory usage every N seconds to a log file.



3). Automated download — download_file.sh

Purpose: download a file with wget or curl into a predefined directory.





Challenges faced

Setting up WSL and choosing Ubuntu 24.04 (initially tried 22.04), capturing screenshots while installing, and learning permission management (chmod/chown).

Learning outcomes: Gained experience installing Ubuntu via WSL, using 20 core Linux commands, writing shell scripts for backup/monitoring/download, and publishing to GitHub.

Real-world applications: System administration, DevOps automation, creating routine backups, monitoring servers, and automating deployments or data retrieval.

Reflection

I now know how Linux can function properly on a Windows system and how WSL operates. I learned how effective Linux is at managing files, keeping an eye on system resources, verifying networks, and managing processes throughout the assignment. I also practiced a number of fundamental commands.
One of the most beneficial aspects was writing shell scripts, which demonstrated to me how tasks can be automated with a few lines of code. Future courses like programming, cloud computing, ethical hacking, and cybersecurity will benefit from these abilities. All things considered, this assignment has given me a solid basis for learning Linux and delving into more complex subjects. I feel more equipped now to use Linux in practical academic projects and technical professions.

I felt more comfortable using Linux after creating folders, copying files, and examining system details. The part I most enjoyed was writing shell scripts because it demonstrated to me how little scripts can carry out practical tasks automatically. It helped me understand why Linux is preferred by professionals because it offers complete control and completes tasks quickly.
I also encountered some problems, such as missing packages and permission errors, but resolving them taught me even more. My familiarity with Linux has improved as a result of this assignment, and my interest in ethical hacking and cybersecurity has grown. I am eager to learn more commands, tools, and scripting techniques because I believe that learning Linux will be very beneficial for my future career.