

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

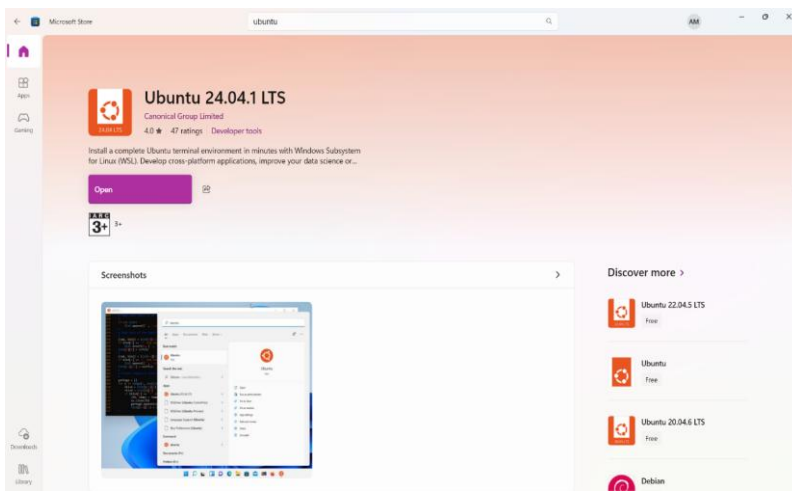
Command (syntax)	Short description	When / why used
pwd	Print Working Directory — shows full path of current directory.	Confirm where you are in the filesystem before running file operations or giving paths.
ls / ls -l / ls -a	List files and directories (long format, show hidden files).	Inspect directory 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.

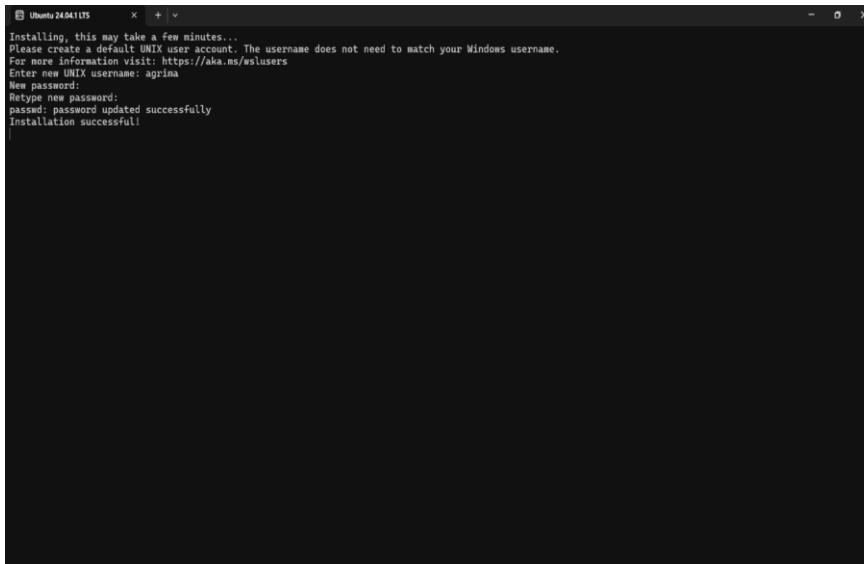
LINUX INSTALLATION DOCUMENTATION

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

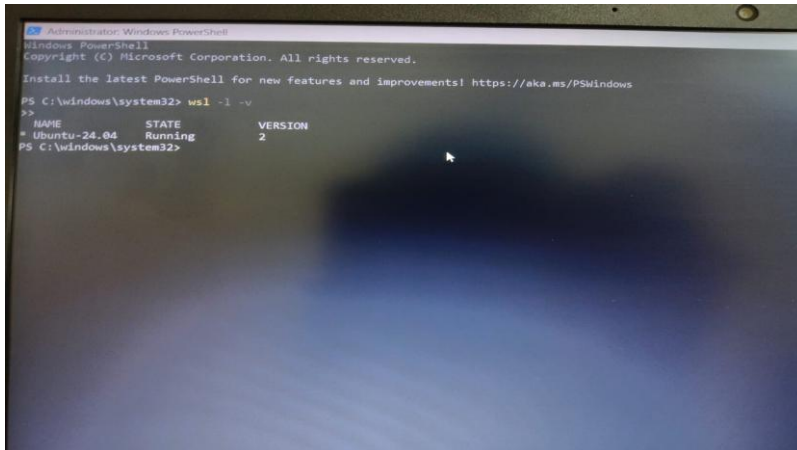
```
agrima@AGRI19A:~$ sudo apt update
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: agrima
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

agrima@AGRI19A:~$ sudo apt update
sudo apt upgrade -y
[sudo] password for agrima:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1389 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [215 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9016 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [989 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [294 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19.4 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2180 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [486 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [568 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [27.4 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5956 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
Get:22 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5082 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
69% [6 Packages store 0 B] [23 Components-amd64 23.0 kB/3871 kB 1%]
```

```
agrima@AGRI19A:~$ sudo apt upgrade -y
Setting up landscape-common (24.02-0ubuntu5.6) ...
Setting up netplan.io (1.1.2-2-ubuntu24.04.2) ...
Setting up libegl-mesa0:amd64 (25.0-7-0ubuntu0.24.04.2) ...
Setting up python3-software-properties (0.99.49.3) ...
Setting up python3-requests (2.31.0dfsg-1ubuntu1.1) ...
Setting up landscape-client (24.02-0ubuntu5.6) ...
Setting up ubuntu-pro-client (24ubuntu0-24.04) ...
Installing new version of config file /etc/apparmor.d/ubuntu_pro_apt_news ...
Installing new version of config file /etc/apparmor.d/ubuntu_pro_esa_cache ...
Installing new version of config file /etc/apt/apt.conf.d/20apt-esa-hook.conf ...
Setting up libgtk-3-common (3.24.41-4ubuntu1.3) ...
Setting up ubuntu-pro-client-l10n (36ubuntu0-24.04) ...
Setting up wsl-pre-service (0.1.18-24.04.2) ...
Setting up libgl-mesa0:amd64 (25.0-7-0ubuntu0.24.04.2) ...
Setting up cloud-init (25.2-0ubuntu1-24.04.1) ...
Installing new version of config file /etc/cloud/templates/sources.list.debian.deb822.tpl ...
Installing new version of config file /etc/cloud/templates/sources.list.ubuntu.deb822.tpl ...
Setting up ubuntu-minimal (1.539.2) ...
Setting up python3-apport (2.28.1-0ubuntu3.8) ...
Setting up libgtk-3-0:amd64 (3.24.41-4ubuntu1.3) ...
Setting up wsl-setup (0.5.18-24.04) ...
Installing new version of config file /etc/update-motd.d/99-wsl ...
Setting up libgtk-3-bin (3.24.41-4ubuntu1.3) ...
Setting up apport-core-dump-handler (2.28.1-0ubuntu3.8) ...
apport-autoreport.service is a disabled or a static unit not running, not starting it.
Setting up python3-distupgrade (1:24.04.27) ...
Setting up ubuntu-release-upgrader-core (1:24.04.27) ...
Setting up python3-update-manager (1:24.04.12) ...
Setting up update-manager-core (1:24.04.12) ...
Processing triggers for sgml-base (1.31) ...
Processing triggers for install-info (7:1.3build2) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for systemd (255.4-0ubuntu8.11) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for dbus (1.14.18-0ubuntu4.1) ...
Setting up packagekit (1.2.8-2ubuntu1.2) ...
Setting up packagekit-tools (1.2.8-2ubuntu1.2) ...
Setting up software-properties-common (0.99.49.3) ...
Setting up ubuntu-wsl (1.539.2) ...
agrima@AGRI19A:~$
```

Step 4- Checking WSL version using the command

`wsl -l -v`



The screenshot shows a Windows PowerShell terminal window. The title bar reads "Administrator: Windows PowerShell". The window content displays the following text:

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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> wsl -l -v
>>
  NAME                STATE      VERSION
  -
  Ubuntu-24.04         Running    2
PS C:\Windows\system32>
```

LINUX COMMANDS

1. File Navigation:

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

Used - Use to confirm your current location in filesystem.

```
aprima@AGRIHA:~$ pwd
/home/aprima
aprima@AGRIHA:~$
```

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

Used - Inspect files and directories.

```
U: aprima@AGRIHA:~$ pwd
/home/aprima
aprima@AGRIHA:~$ ls
aprima@AGRIHA:~$
```

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.

```
agrina@AGRIMA: /  
agrina@AGRIMA:~$ pwd  
/home/agrina  
agrina@AGRIMA:~$ ls  
agrina@AGRIMA:~$ cd /  
agrina@AGRIMA:/$  
agrina@AGRIMA:/$ ls  
bin          boot      etc       init      lib.usr-is-merged  lost+found  mnt      proc      run     /sbin.usr-is-merged  srv      tmp      var  
bin.usr-is-merged  dev      home     lib       lib64      media      opt      root     sbin     snap      sys      usr
```

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.

```
agrina@AGRIMA:~$ pwd  
/home/agrina  
agrina@AGRIMA:~$ ls  
agrina@AGRIMA:~$ cd /  
agrina@AGRIMA:/$  
agrina@AGRIMA:/$ ls  
bin          boot      etc       init      lib.usr-is-merged  lost+found  mnt      proc      run     /sbin.usr-is-merged  srv      tmp      var  
bin.usr-is-merged  dev      home     lib       lib64      media      opt      root     sbin     snap      sys      usr
```

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

A screenshot of a Windows 10 desktop environment. A terminal window titled 'mprow@BAGHIMA: ~' is open, displaying the command 'mprow@BAGHIMA:~\$ sudo apt install tree'. The terminal output is currently empty. The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several pinned application icons including File Explorer, Microsoft Edge, and various utility tools. The system tray on the right shows the date and time as 10:11 on 10/11/2023.

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

```

@ubuntu082019:~$ sudo apt install tree
[sudo] password for aprite:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

The following packages were automatically installed and are no longer required:
  libbsd0 libbsd-dev libbsd-lsb3 libbsd0 libbsd0-dev libbsd0-lsb3
Use 'sudo apt autoremove' to remove them.

The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 47.4 kB of archives.
After this operation, 111 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 tree amd64 2.1.1-2ubuntu2.20.04.2 [47.4 kB]
748B+500kB=507kB
Fetched 47.4 kB in 2s (25.6 kB/s)
Selecting previously unselected package tree.
(Reading database ... 5002 files and directories currently installed.)
Preparing to unpack .../tree-2.1.1-2ubuntu2.20.04.2_amd64.deb ...
Unpacking tree (2.1.1-2ubuntu2.20.04.2) ...
Setting up tree (2.1.1-2ubuntu2.20.04.2) ...
Processing triggers for man-db (2.12.0-3build2) ...
aprite@a082019:~$

```

File & Directory Management Commands:

1). **mkdir**- make directory.

mkdir testfolder created a folder.

Syntax- mkdir <folder name>

Used- Create project or storage directories.

```
agrima@AGIRIMA: ~  
agrima@AGIRIMA:~$ mkdir testfolder  
agrima@AGIRIMA:~$ ls  
testfolder  
agrima@AGIRIMA:~$
```

2). **touch** - Create an empty file.

Syntax- touch <file>

--touch testfile.txt.

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

```
agrima@AGIRIMA: ~  
agrima@AGIRIMA:~$ mkdir testfolder  
agrima@AGIRIMA:~$ ls  
testfolder  
agrima@AGIRIMA:~$ touch testfile.txt  
agrima@AGIRIMA:~$ ls  
testfile.txt testfolder  
agrima@AGIRIMA:~$
```

3). **cp** – Copy files

Syntax- cp source dest

Used- Duplicate files for backup or editing safely.

```
agrina@AGRIHA:~$ cp testfile.txt copy.txt
agrina@AGRIHA:~$ ls
copy.txt  testfile.txt  testfolder
agrina@AGRIHA:~$
```

4). **mv** - Move or rename files.

Rename files or relocate them.

Syntax- mv old.txt new.txt

--- mv copy.txt moved.txt

```
agrina@AGRIHA:~$ mv copy.txt moved.txt
agrina@AGRIHA:~$ ls
moved.txt  testfile.txt  testfolder
agrina@AGRIHA:~$
```

5). **rm** – remove files

Syntax- rm <file>

Delete unwanted files

```
aprima@AGRIHA:~$ mv copy.txt moved.txt
aprima@AGRIHA:~$ ls
moved.txt  testfile.txt  testfolder
aprima@AGRIHA:~$ rm moved.txt
aprima@AGRIHA:~$ ls
testfile.txt  testfolder
aprima@AGRIHA:~$
```

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:

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

```
agrima@AGRIMA:~$ chmod 755 testfile.txt
agrima@AGRIMA:~$ ls -l
total 4
-rwxr-xr-x 1 agrima agrima 0 Nov 18 12:08 testfile.txt
drwxr-xr-x 2 agrima agrima 4096 Nov 18 12:06 testfolder
agrima@AGRIMA:~$
```

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

2). **chown**- change the ownership of the file

Syntax- sudo chown \$USER file

Fix ownership issues or set for deployment.

```
agrima@AGRIMA:~$ sudo chown $USER testfile.txt
[sudo] password for agrima:
agrima@AGRIMA:~$ ls -l
total 4
-rwxr-xr-x 1 agrima agrima 0 Nov 18 12:08 testfile.txt
drwxr-xr-x 2 agrima agrima 4096 Nov 18 12:06 testfolder
agrima@AGRIMA:~$
```

PROCESS MONITORING COMMANDS

1). **ps** - Show current shell processes.

ps (or ps aux)

Check running processes and their PIDs.

```
agrina@AGRIHA:~$ ps
PID TTY          TIME CMD
295 pts/0      00:00:00 bash
7751 pts/0      00:00:00 ps
agrina@AGRIHA:~$
```

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

top

Monitor CPU, memory and running processes.

```
agrina@AGRIHA:~$ top
top - 12:16:23 up 30 min, 1 user, load average: 0.00, 0.00, 0.00
tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%cm(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 7763.2 total, 7163.1 free, 540.0 used, 227.2 buff/cache
MiB Swap: 2048.0 total, 2048.0 free,  0.0 used, 7232.2 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+ COMMAND
    1 root        0   0 22076 11184 9600  S   0.0   0.2   0:01.96 systemd
    2 root        0   0  3072  1664   0  0  0:00.00 init-systemd(Ub
    6 root        0   0  3888  1920  1920  S   0.0   0.0   0:00.00 init
   176 root        0   0  4236  2432  2304  S   0.0   0.0   0:00.01 cron
   177 message+    0   0  9608  4992  4352  S   0.0   0.1   0:00.79 dbus-daemon
   184 root        0   0  17968 8320  7424  S   0.0   0.1   0:00.27 systemd-logind
   282 root        0   0  3168  1920  1792  S   0.0   0.0   0:00.01 agetty
   288 root        0   0  3116  1792  1664  S   0.0   0.0   0:00.00 agetty
   289 root        0   0 186992 22816 12928  S   0.0   0.3   0:00.10 unattended-upgr
   293 root        0   0  3888   896   896  S   0.0   0.0   0:00.00 sessionleader
   294 root        0   0  3896  1624   896  S   0.0   0.0   0:00.13 Relay(295)
   295 agrina      0   0  6856  5248  3520  S   0.0   0.1   0:00.00 bash
   296 root        0   0  6696  4480  3712  S   0.0   0.1   0:00.01 login
   328 agrina      0   0 28032 18880 9888  S   0.0   0.1   0:00.12 systemd
   329 agrina      0   0 21168  3520  1792  S   0.0   0.0   0:00.00 (sd-pam)
   353 agrina      0   0  6856  4992  3456  S   0.0   0.1   0:00.01 bash
  1438 root       19  -1 58576 10464 13568  S   0.0   0.2   0:00.11 systemd-journal
  4980 syslog     0   0 222560  5120  4080  S   0.0   0.1   0:00.03 rsyslogd
 5921 systemd+    0   0  91824  7888  6912  S   0.0   0.1   0:00.04 systemd-timesyn
 5972 root        0   0 24976  5212  5808  S   0.0   0.1   0:00.06 systemd-udev
 6243 polkitd+    0   0 21456 12928 10880  S   0.0   0.2   0:00.00 systemd-resolve
 6422 polkitd     0   0 388164  7888  7888  S   0.0   0.1   0:00.18 polkitd
 7754 agrina      0   0  9308  5632  3456  R   0.0   0.1   0:00.02 top
```

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

Syntax- sleep 1000 & + kill <PID>

Demonstrates starting background tasks and stopping them safely.

```
april@kali:~$ sleep 1000 &
[1] 7788
april@kali:~$

april@kali:~$ ps
  PID TTY          TIME CMD
 255 pts/0    00:00:00 bash
 7788 pts/0    00:00:00 sleep
 7789 pts/0    00:00:00 ps
april@kali:~$

april@kali:~$ ps
  PID TTY          TIME CMD
 255 pts/0    00:00:00 bash
 7788 pts/0    00:00:00 sleep
 7789 pts/0    00:00:00 ps
april@kali:~$ kill 7788
april@kali:~$ ps
  PID TTY          TIME CMD
 255 pts/0    00:00:00 bash
 7788 pts/0    00:00:00 ps
[1]+  Terminated                  sleep 1000
april@kali:~$
```

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.

```
agrina@AGRIMA:~$ ping google.com -c 4
PING google.com (142.251.223.142) 56(84) bytes of data:
64 bytes from tzdela-be-in-f14.1e100.net (142.251.223.142): icmp_seq=1 ttl=116 time=16.5 ms
64 bytes from tzdela-be-in-f14.1e100.net (142.251.223.142): icmp_seq=2 ttl=116 time=5.67 ms
64 bytes from tzdela-be-in-f14.1e100.net (142.251.223.142): icmp_seq=3 ttl=116 time=13.3 ms
64 bytes from tzdela-be-in-f14.1e100.net (142.251.223.142): icmp_seq=4 ttl=116 time=6.38 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3806ms
rtt min/avg/max/mdev = 5.671/10.861/16.537/4.605 ms
agrina@AGRIMA:~$
```

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

Syntax- ip a

Check IP, interface status.

```
agrina@AGRIMA:~$ 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
    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 08:00:15:5d:b0:5e-8a brd ff:ff:ff:ff:ff:ff
    inet 172.25.46.10/20 brd 172.25.47.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fab0:8ebd/64 scope Link
        valid_lft forever preferred_lft forever
agrina@AGRIMA:~$
```

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

Syntax- netstat -tulpn

find services bound to ports and troubleshoot network services.

```

$ sudo apt-get install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libidn2-0 libidn3 libidn3-dev libidn3-extra libidn3-idn libidn3-idn2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 204 kB of archives:
after this operation 811 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 net-tools amd64 2.10-0.1ubuntu4.0 [204 kB]
debconf: unable to initialize frontend: Dialog
debconf: falling back to frontend: Readline
Selecting previously unselected package net-tools.
(Reading database ... 49000 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4.0_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4) ...
Setting up net-tools (2.10-0.1ubuntu4) ...
Processing triggers for man-db (2.12.0-4ubuntu2) ...
$ sudo apt-get install net-tools

```

```

sgrima@Sgrima:~$ netstat -tulnp
#net: All processes could be identified, non-owned process info
#will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	*:::0.0.0.0	*:0.0.0.0:*	LISTEN	-
tcp	0	0	*:::0.0.0.0:53	*:0.0.0.0:*	LISTEN	-
tcp	0	0	10.255.255.254:53	0.0.0.0:*	LISTEN	-
udp	0	0	*:::0.0.0.0:53	*:0.0.0.0:*	-	-
udp	0	0	*:::0.0.0.0:53	*:0.0.0.0:*	-	-
udp	0	0	10.255.255.254:53	0.0.0.0:*	-	-
udp	0	0	*:::0.0.0.0:1323	*:0.0.0.0:*	-	-
udp	0	0	:::1.323	:::*	-	-

```

sgrima@Sgrima:~$

```

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

Syntax- date

Log timestamps; used in scripts for naming backups.

```
agrima@AGRIMA:~$ date
Tue Nov 18 12:22:16 UTC 2025
agrima@AGRIMA:~$
```

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

Syntax- df -h

Check free space before large operations like backups.

```
agrima@AGRIMA:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.8G  0  3.8G   0% /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2
none            3.8G  4.0K  3.8G   1% /mnt/WSL
drivers          476G  183G  294G  22% /usr/lib/WSL/drivers
/dev/sdd        1007G  1.7G  994G   1% /
none            3.8G  84K  3.8G   1% /mnt/WSLg
none            3.8G  0  3.8G   0% /usr/lib/WSL/Lib
rootfs          3.8G  2.7M  3.8G   1% /init
none            3.8G  536K  3.8G   1% /run
none            3.8G  0  3.8G   0% /run/lock
none            3.8G  0  3.8G   0% /run/shm
none            3.8G  76K  3.8G   1% /mnt/WSLg/versions.txt
none            3.8G  76K  3.8G   1% /mnt/WSLg/doc
C:\              476G  183G  294G  22% /mnt/c
tmpfs           777M  28K  777M   1% /run/user/1000
agrima@AGRIMA:~$
```

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

Syntax- uname -a

Record system details for documentation or debugging.

```
agrima@AGRIMA:~$ uname -a
Linux AGRIMA 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:39:46 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
agrima@AGRIMA:~$
```

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

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

```

13 root@ubuntu:~# history
14 sudo apt update
15 sudo apt upgrade -y
16 apt -y --no-install-recommends
17 ls
18 mv
19 mv
20 mv
21 cd /
22 cd /
23 cd /
24 mv
25 cd
26 mkdir testFolder
27 touch testFile.txt
28 ls
29 cp testFile.txt copy.txt
30 mv copy.txt moved.txt
31 rm moved.txt
32 mv moved 755 testFile.txt
33 ls
34 head -n50 $HOME/testFile.txt
35 ls -l
36 top
37 sleep 1000 s
38 kill 7788
39 top
40 ping google.com -c 4
41 df -h

```

```

1 y
2 mv
3 mv
4 mv
5 mv
6 mv
7 mv
8 mv
9 mv
10 mv
11 ls
12 cd /
13 mv
14 mv
15 mv
16 mv
17 mv
18 mv
19 mv
20 mv
21 mv
22 mv
23 mv
24 mv
25 mv
26 mv
27 mv
28 mv
29 mv
30 mv
31 mv
32 mv
33 mv
34 mv
35 mv
36 mv
37 mv
38 mv
39 mv
40 mv
41 mv
42 mv
43 mv
44 mv
45 mv
46 mv
47 mv
48 mv
49 mv
50 mv
51 mv
52 mv
53 mv
54 mv
55 mv
56 mv
57 mv
58 mv
59 mv
60 mv
61 mv
62 mv
63 mv
64 mv
65 mv
66 mv
67 mv
68 mv
69 mv
70 mv
71 mv
72 mv
73 mv
74 mv
75 mv
76 mv
77 mv
78 mv
79 mv
80 mv
81 mv
82 mv
83 mv
84 mv
85 mv
86 mv
87 mv
88 mv
89 mv
90 mv
91 mv
92 mv
93 mv
94 mv
95 mv
96 mv
97 mv
98 mv
99 mv
100 mv

```

Script Shells

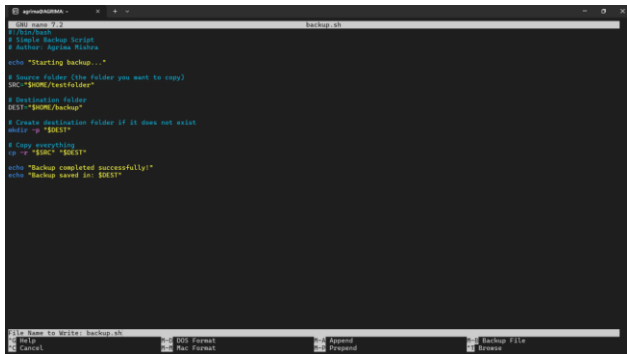
1). Backup a directory - backup.sh

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.



```
aprima@primas:~$ nano backup.sh
```



```
#!/bin/bash
# Simple Backup Script
# Author: Aprima Primas

echo "Starting backup..."

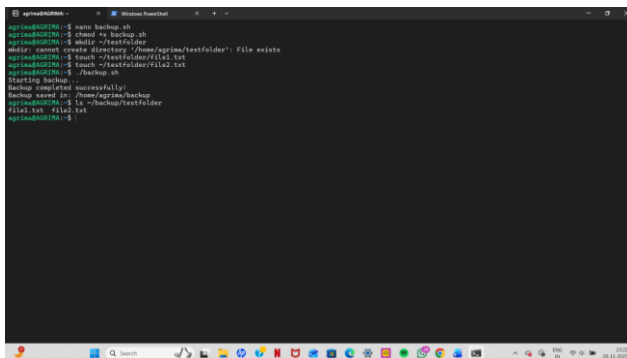
# Source folder (the folder you want to copy)
SRC="$HOME/testfolder"

# Destination folder
DEST="$HOME/Backup"

# Create destination folder if it does not exist
mkdir -p "$DEST"

# Copy everything
cp -r "$SRC" "$DEST"

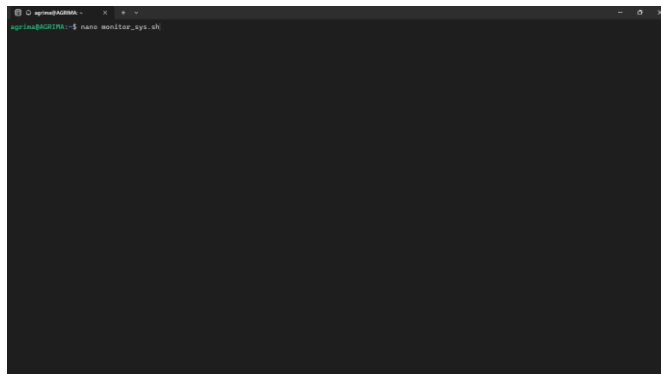
echo "Backup completed successfully!"
echo "Backup saved in: $DEST"
```



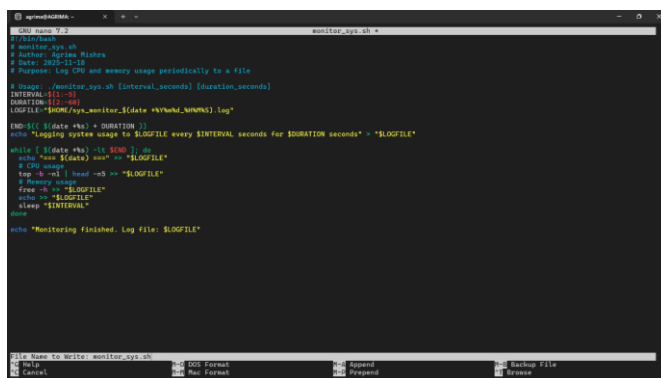
```
aprima@primas:~$ nano backup.sh
aprima@primas:~$ chmod +x backup.sh
aprima@primas:~$ mkdir -p testfolder
mkdir: cannot create directory '/home/aprima/testfolder': File exists
aprima@primas:~$ touch ~/testfolder/file1.txt
aprima@primas:~$ touch ~/testfolder/file2.txt
aprima@primas:~$ ./backup.sh
Starting Backup
Backup completed successfully!
Backup saved in: /home/aprima/Backup
aprima@primas:~$ ls -la ~/Backup/testfolder
file1.txt file2.txt
aprima@primas:~$
```

2). CPU/Memory monitor — monitor_sys.sh

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



```
april@AGRI19A:~$ nano monitor_sys.sh
```



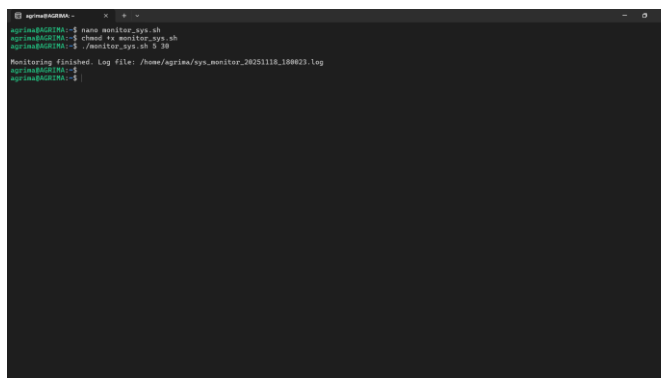
```
#!/bin/bash
# monitor_sys.sh
# Author: April Rishra
# Date: 2023-11-18
# Purpose: Log CPU and memory usage periodically to a file

# Usage: ./monitor_sys.sh [interval_seconds] [duration_seconds]
INTERVAL=${1:-1}
DURATION=${2:-99}
LOGFILE="/home/april/sys_monitor_${date +%Y%m%d_%H%M%S}.log"

CMD=$( [ $(date +%s) > DURATION ] && echo "Logging system usage to $LOGFILE every $INTERVAL seconds for $DURATION seconds" > "$LOGFILE" )

while [ $(date +%s) -lt $DUR ]; do
  echo "Time $(date) mem" >> "$LOGFILE"
  # CPU usage
  CPU=$(cat /proc/stat | grep ^cpu | awk '{print $2/$4 * 100.0}')
  # Memory usage
  MEM=$(cat /proc/meminfo | grep MemTotal | awk '{print $2/$4 * 100.0}')
  echo "$CPU $MEM" >> "$LOGFILE"
  sleep $INTERVAL
done

echo "Monitoring finished. Log file: $LOGFILE"
```



```
april@AGRI19A:~$ nano monitor_sys.sh
april@AGRI19A:~$ chmod +x monitor_sys.sh
april@AGRI19A:~$ ./monitor_sys.sh 5 30

Monitoring finished. Log file: /home/april/sys_monitor_20231118_100023.log
april@AGRI19A:~$
```

3). Automated download — download_file.sh

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

```
agrinab@agrinu:~$ cd ~/scripts
agrinab@agrinu:~/scripts$ nano download_file.sh
agrinab@agrinu:~/scripts$
```

```
#!/bin/bash
# download_file.sh
# Author: Agrina Mishra
# Date: 2025-11-18
# Purpose: Download a file from the internet to a downloads folder

# Usage: ./download_file.sh URL

URL="$1"
DEST_DIR="$HOME/downloads"
FILENAME=$(basename "$URL")

if [ -z "$URL" ]; then
    echo "Usage: $0 <url>"
    exit 1
fi

mkdir -p "$DEST_DIR"
# Try wget first, then curl
if command -v wget >/dev/null 2>/dev/null; then
    wget -O "${DEST_DIR}/${FILENAME}" "$URL"
elif command -v curl >/dev/null 2>/dev/null; then
    curl -L -o "${DEST_DIR}/${FILENAME}" "$URL"
else
    echo "wget or curl not found. Install one and retry."
    exit 2
fi

if [ $? -eq 0 ]; then
    echo "Downloaded to ${DEST_DIR}/${FILENAME}"
else
    echo "Download failed."
    exit 3
fi
```

```
agrinab@agrinu:~$ cd ~/scripts
agrinab@agrinu:~/scripts$ nano download_file.sh
agrinab@agrinu:~/scripts$ ./download_file.sh
agrinab@agrinu:~/scripts$ nano download_file.sh
agrinab@agrinu:~/scripts$ chmod +x download_file.sh
agrinab@agrinu:~/scripts$ ./download_file.sh https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy.pdf
--2025-11-18 18:18:18-- https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy.pdf
Resolving www.w3.org (www.w3.org)... 104.18.22.19, 104.18.21.19, 2086.4760.8391.741c:87ab:83e:19e3:8f51
Connecting to www.w3.org (www.w3.org)[104.18.22.19]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13264 (13K) [application/pdf]
Saving to: '/home/agrina/downloads/dummy.pdf'

/home/agrina/downloads/dummy.pdf 100%[=====] 12.95K --.-KB/s in 0s

2025-11-18 18:18:18 (95.4 MB/s) - '/home/agrina/downloads/dummy.pdf' saved [13264/13264]

Downloaded to /home/agrina/downloads/dummy.pdf
agrinab@agrinu:~/scripts$ ls ~/downloads
dummy.pdf
agrinab@agrinu:~/scripts$
```

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.