ASSIGMENT : UNIT 2

NAME : PRIYANSHI

ROLL NO : 25010216

COURSE : BTECH CSE(core)

SECTION : A

SUBJECT : COMPUTATIONAL SCIENCE FUNDAMENTALS AND CAREER PATH WAYS

SUBMITTED TO: MR. RAJESH BANDRA

#INTRODUCTION

This assignment is designed to provide practical, hands -on experience with the Linux Operating System.

Linux is an essential skill for computer science and IT professionals due to its inherent flexibility, security, and powerful scripting capabilities. The core objective is to demonstrate proficiency across three key areas:

1) Successfully establishing a Linux environment (Installation)

2) Executing and documenting fundamental shell commands

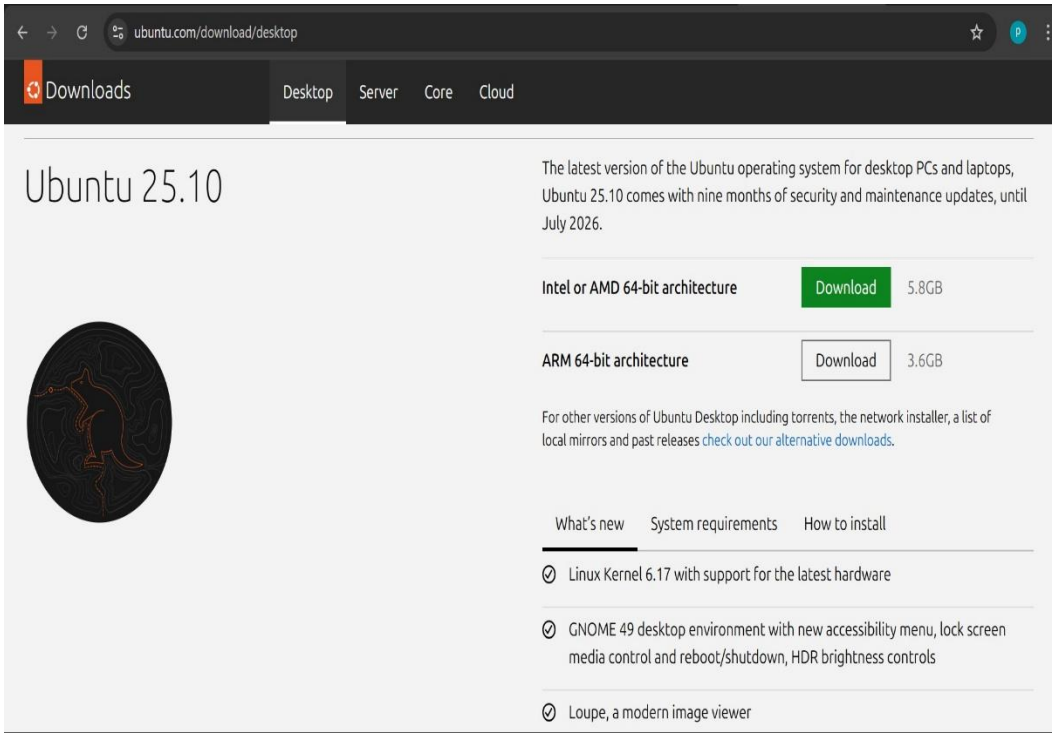
3) Developing functional Bash shell scripts to automate system administration tasks

#STEP 1

How to download Ubuntu and virtual box and make a machine to operate linux

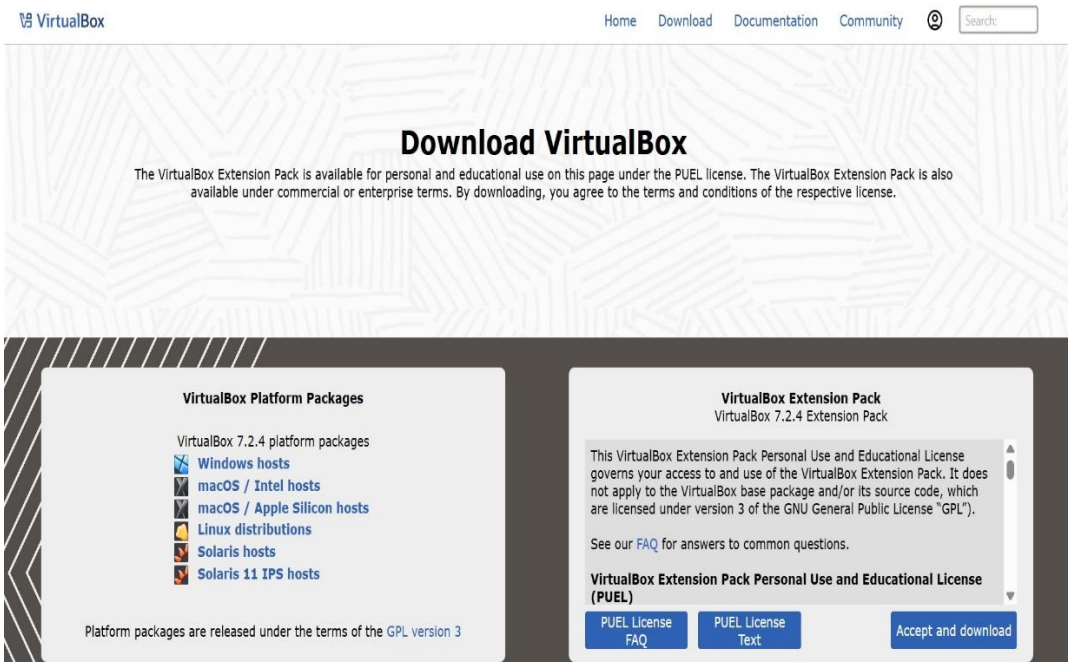
Step 1: browse download Ubuntu

Step2: download it in the window according to the system

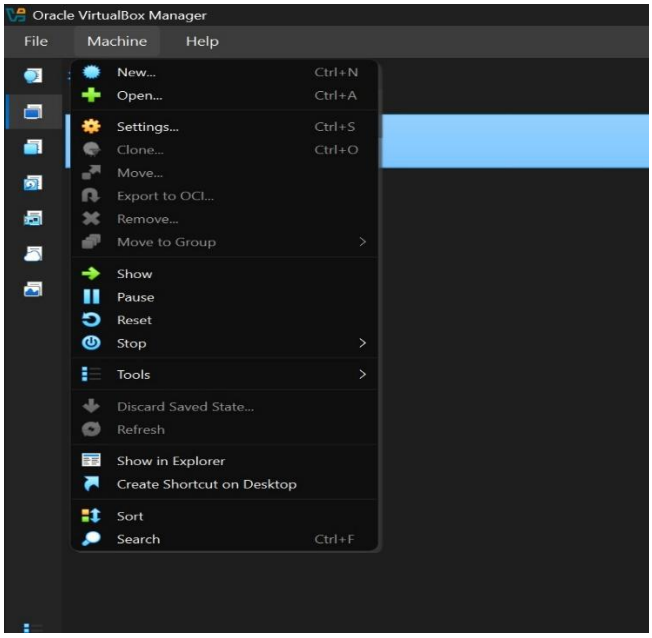


Step 3: browse download virtual box

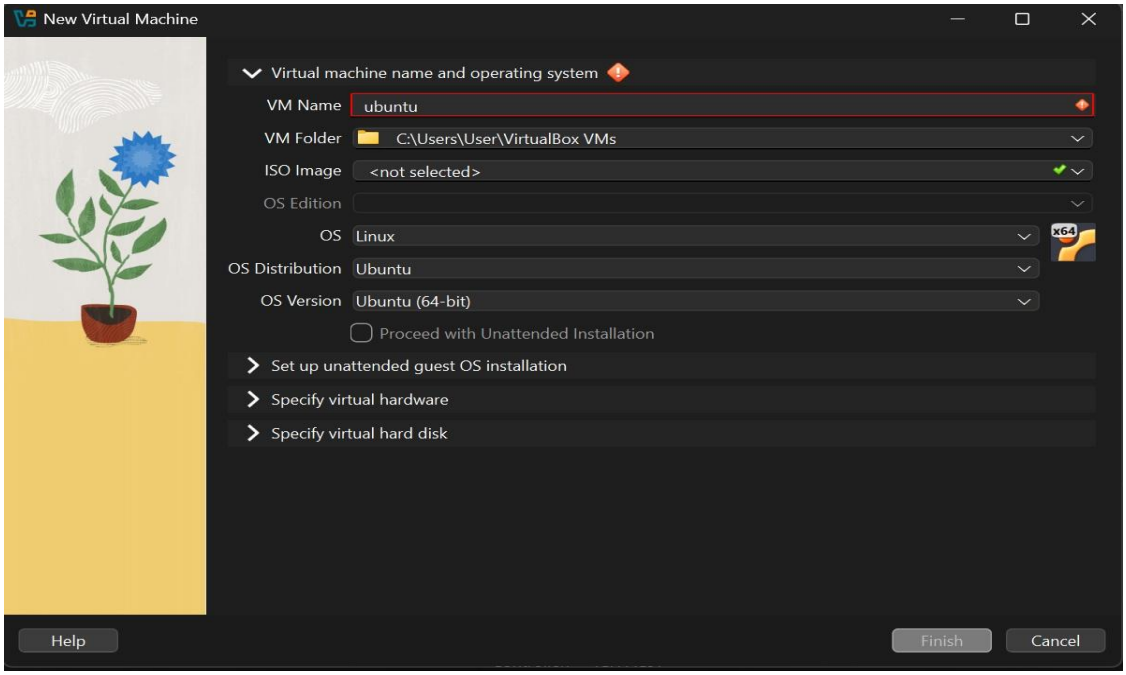
Step 4: download it in window



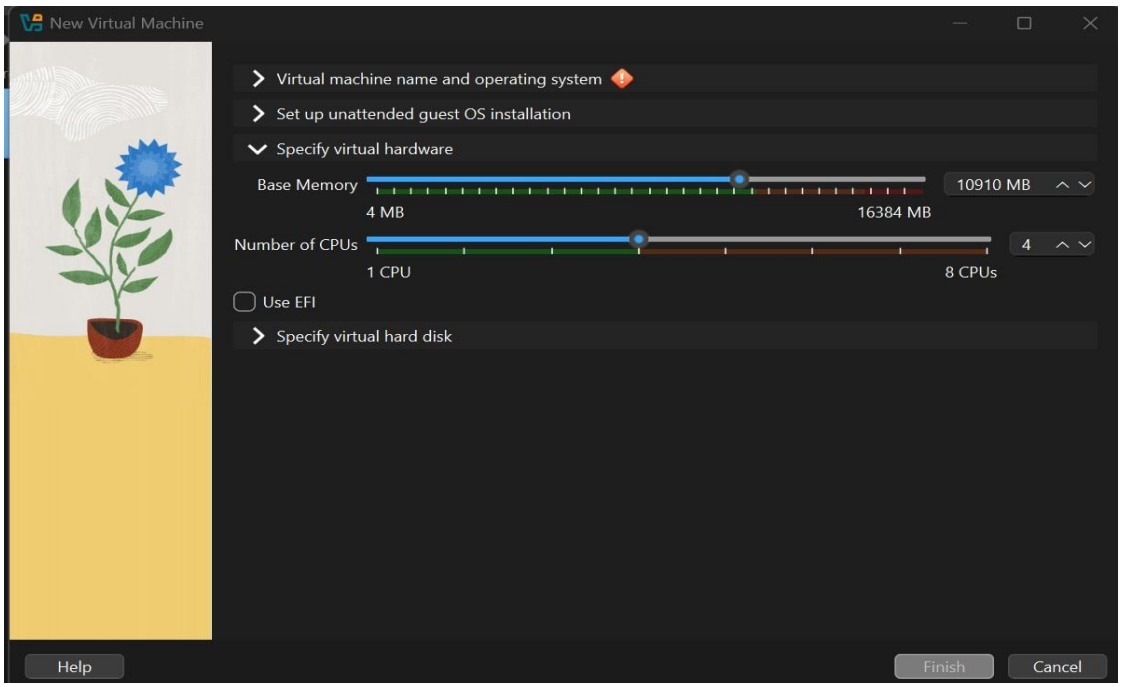
Step 5: open the virtual box manager and click the machine



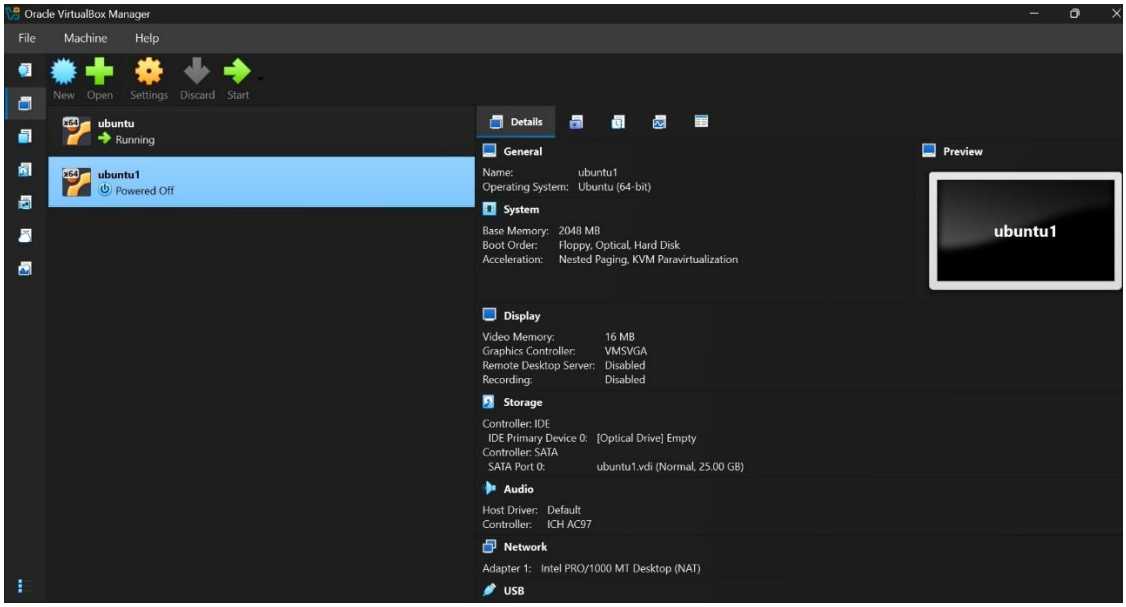
Step 6: select new machine and create a new machine giving it name as Ubuntu



Step 7: click on specify virtual hardware than set up base memory and number and CPUs as follows

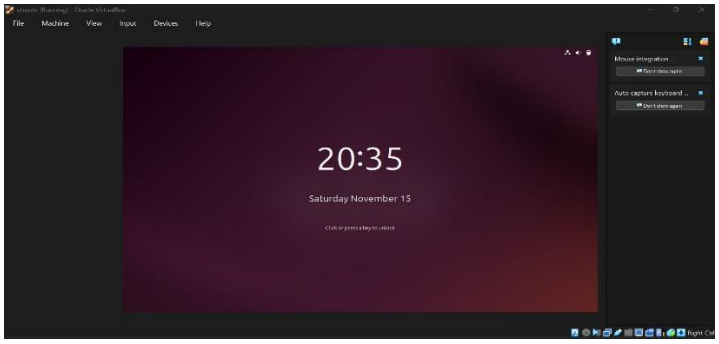


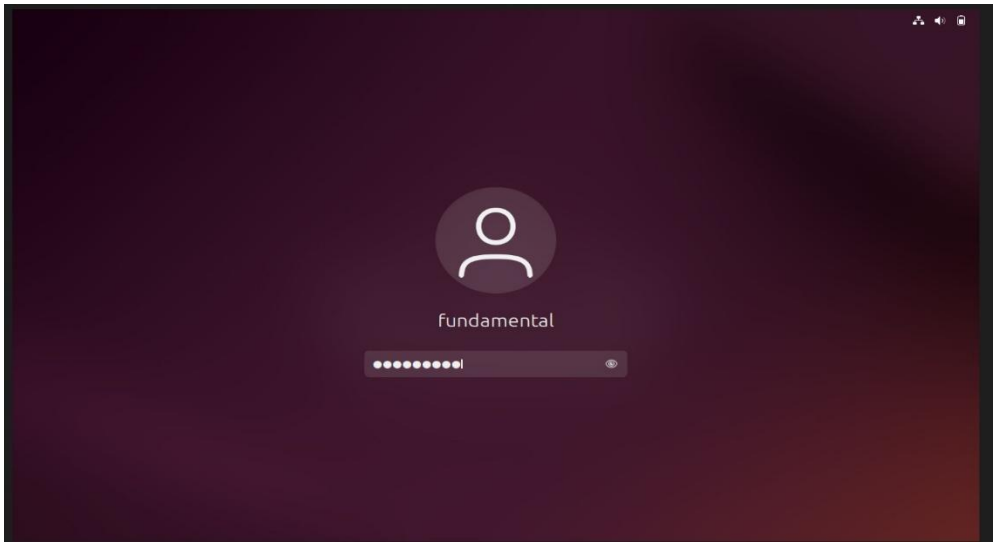
Step 8: add a strong password and click on finish than new machine will be created to start that machine click on start.



S

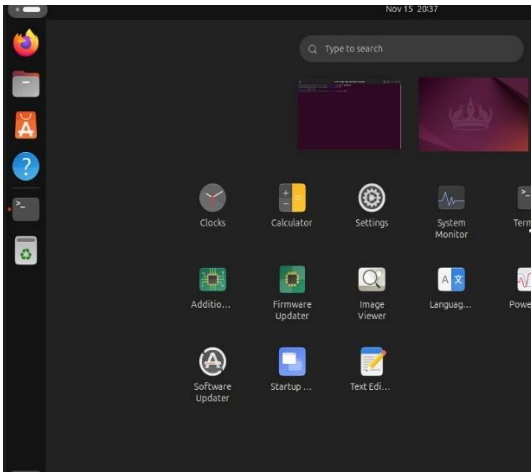
Step 9: start the machine this would take 20-30 minutes to set up the machine than click and select all the default setting and than start the machine again it will show the interface as follows

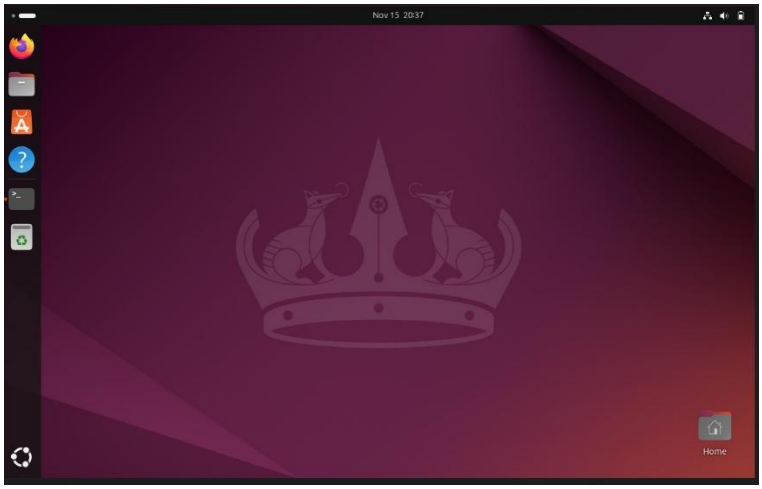



fundamental

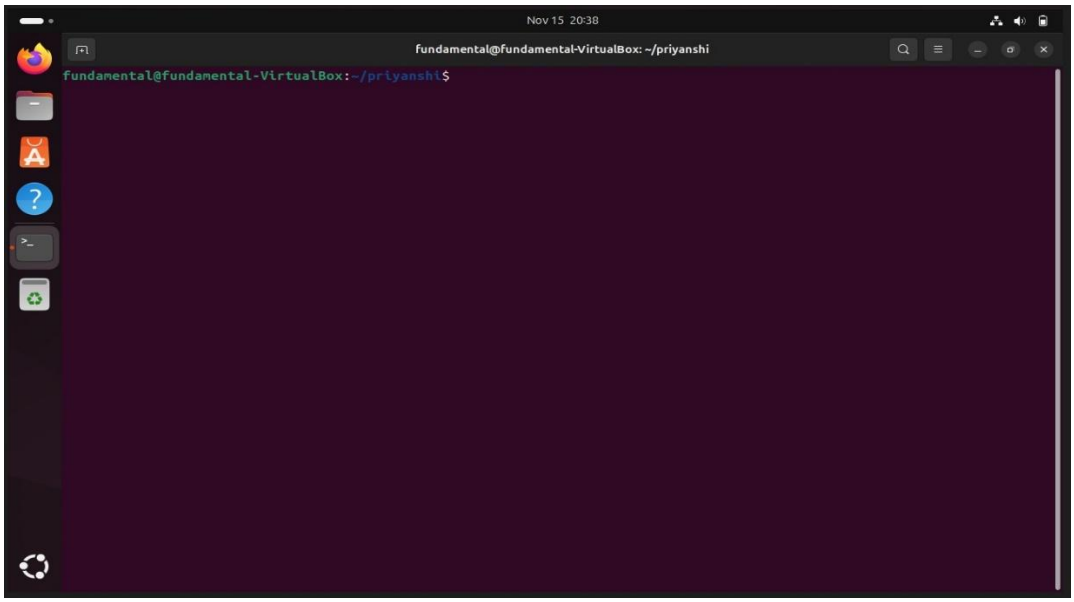
Step 10: add the password that was created in step 8

Step 11: this will appear click on the circle type icon and search terminal and there we can practice all the commands



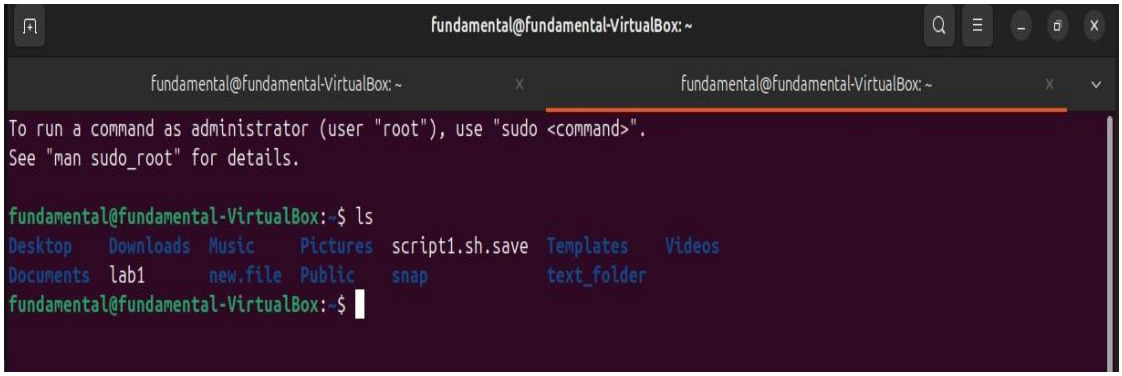


Step 12: here is the terminal

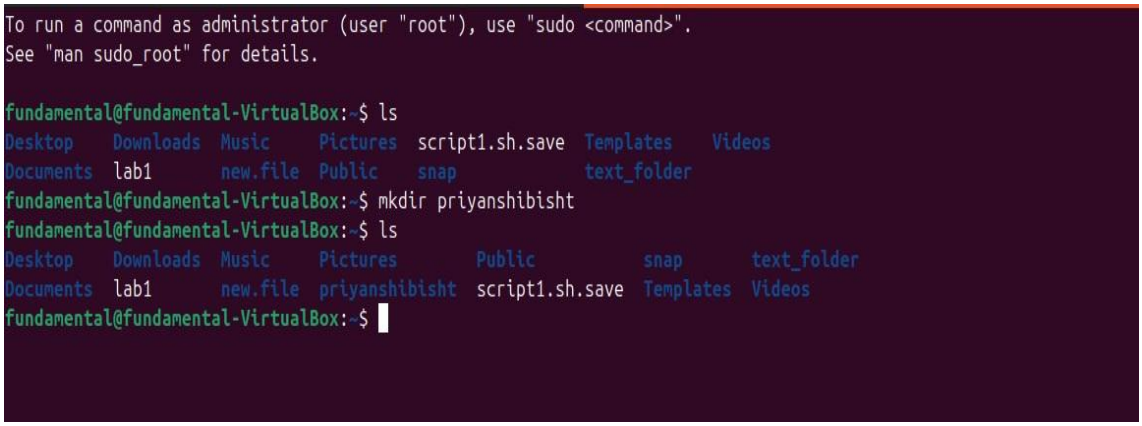


commands practice

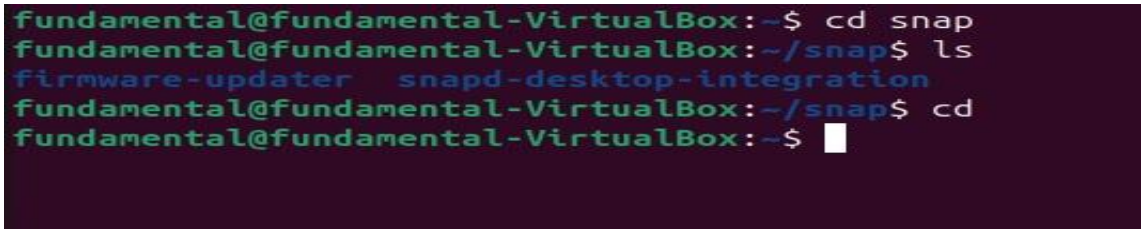
1. Ls
   - Syntax: ls
   - Description: this command is used to list files or folders in current directory
   - Use Case: To see all files including hidden
   - Implementation:

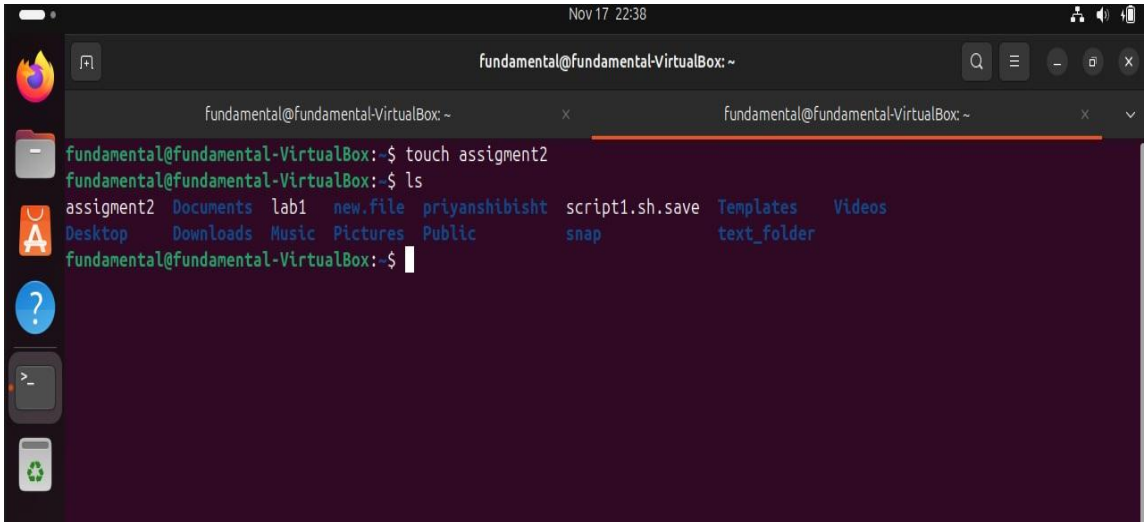


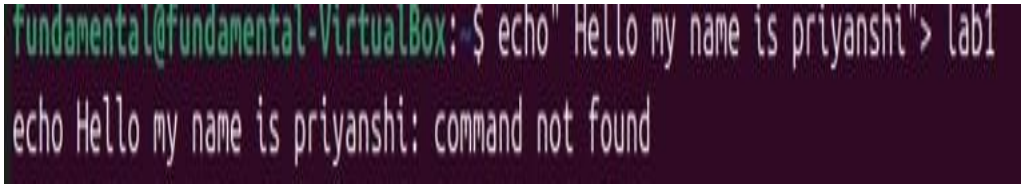
2. mkdir
   - Syntax: mkdir [directory name]
   - Description: it is used to create a new directory or folder
   - Use Case: it is useful for organizing files by creating one or multiple directories quickly
   - Implementation:
   -



3. cd
   - Syntax: cd [options] [directory]
   - Description: it is used to change the current working directory
   - Use Case: It is commonly used whenever you need to navigate to a different directory to access or manage files
   - Implementation:

- 

4. touch
   - Syntax: touch <filename>
   - Use Case: it is used to create an empty file or update file timestamp
   - Description: create multiple files for practice
   - Implementation:
   - 

5. Echo
   - Syntax: echo “text”
   - echo “text” > file
   - Echo “text” >> file
   - Use case: Create a quick file with content
   - Description: Prints output to screen or writes/appends text to a file.
   - Implementation:
   - 

6. **Mv**
   - Syntax: mv <source> <destination>
   - Use case: Moves or renames files/directories
   - Description: The word “mv” stands for “move files” this command in Linux is used to move or rename files and directories
   - Implementation**:**

```
fundamental@fundamental-VirtualBox:~$ mkdir old.file
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  Pictures        Public           snap       text_folder
Desktop     Downloads  Music  old.file  priyanshibisht  script1.sh.save  Templates  Videos
fundamental@fundamental-VirtualBox:~$ mv old.file new.file
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  priyanshibisht  script1.sh.save  Templates    Videos
Desktop     Downloads  Music  Pictures  Public          snap             text_folder
fundamental@fundamental-VirtualBox:~$
```

7. **Cat**
   - Syntax: : cat [filename], for Combining files :- cat [file1] [file2]
   - Use case: The cat command in Linux is used to display the content of the file on the terminal or combine multiple files
   - Description: The name cat command means “concatenate”. It is used to display the content of files, concatenate multiple files, and redirect output to create or append to files
   - Implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  priyanshibisht  script1.sh.save  Templates    Videos
Desktop     Downloads  Music  Pictures  Public          snap             text_folder
fundamental@fundamental-VirtualBox:~$ cat > assigment2
here in assigment2 i have practiced and learned how linux is operated
```

8. Cp
   - Syntax: cp <source> <destination>
   - Use case: Create backup of a config file before editing
   - Description: Copies files or folders
   - Implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  priyanshibisht  script1.sh.save  Templates    Videos
Desktop     Downloads  Music  Pictures  Public          snap             text_folder
fundamental@fundamental-VirtualBox:~$ cp lab1 old.file
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  Pictures        Public           snap       text_folder
Desktop     Downloads  Music  old.file  priyanshibisht  script1.sh.save  Templates  Videos
fundamental@fundamental-VirtualBox:~$
```

9. Cp –r
   - Syntax: cp –r <source_ folder> <destination_ folder>
   - Use case: Duplicate a folder including subfolders and files
   - Description: copies files or folders
   - Implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  Pictures        Public           snap       text_folder
Desktop     Downloads  Music  old.file  priyanshibisht  script1.sh.save  Templates  Videos
fundamental@fundamental-VirtualBox:~$ cp -r lab1 old.file
fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  Pictures        Public           snap       text_folder
Desktop     Downloads  Music  old.file  priyanshibisht  script1.sh.save  Templates  Videos
fundamental@fundamental-VirtualBox:~$
```

10. Rm
    - Syntax: rm <filename>
    - Use case: Delete a file no longer required
    - Description: Deletes a file permanently (no recycle bin)
    - Implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  priyanshi  Public  snap  Templates  text_folder
fundamental@fundamental-VirtualBox:~$ rm -r priyanshi
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  Public  snap  Templates  text_folder  Videos
fundamental@fundamental-VirtualBox:~$ rm -f new.file
rm: cannot remove 'new.file': Is a directory
```

11. rm –f
    - syntax: rm –f <filename>
    - use case: delete a corrupted or permission restricted file
    - description : Force deletes file without prompting even if write-protected
    - implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  priyanshi  Public  snap  Templates  text_folder
fundamental@fundamental-VirtualBox:~$ rm -r priyanshi
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  Public  snap  Templates  text_folder  Videos
fundamental@fundamental-VirtualBox:~$ rm -f new.file
rm: cannot remove 'new.file': Is a directory
```

12. rm –r
    - syntax: rm –r <directoryname>
    - use case : Remove a node_modules folder or build output directory
    - description: deletes folder and all its content recursively
    - implementation:

```
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  priyanshi  Public  snap  Templates  text_folder
fundamental@fundamental-VirtualBox:~$ rm -r priyanshi
fundamental@fundamental-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  new.file  Pictures  Public  snap  Templates  text_folder  Videos
fundamental@fundamental-VirtualBox:~$ rm -f new.file
rm: cannot remove 'new.file': Is a directory
```

13. chmod
    - syntax: chmod <mode> <file>
    - use case: make a script executable
    - description: changes access permissions(read , write, execute)
    - implementation:

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

fundamental@fundamental-VirtualBox:~$ ls
assigment2  Documents  lab1   new.file  Pictures        Public           snap       text_folde
Desktop     Downloads  Music  old.file  priyanshibisht  script1.sh.save  Templates  Videos
fundamental@fundamental-VirtualBox:~$ chmod 755 assigment2
fundamental@fundamental-VirtualBox:~$ ls -l
total 64
-rwxr-xr-x 1 fundamental fundamental    70 Nov 17 22:53 assigment2
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Desktop
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Documents
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Downloads
-rw-rw-r-- 1 fundamental fundamental    53 Nov 17 22:43 lab1
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Music
drwxr-xr-x 3 fundamental fundamental 4096 Nov 17 22:57 new.file
-rw-rw-r-- 1 fundamental fundamental    53 Nov 17 23:06 old.file
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Pictures
drwxrwxr-x 2 fundamental fundamental 4096 Nov 17 22:37 priyanshibisht
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Public
-rw------- 1 fundamental fundamental     1 Nov 17 22:30 script1.sh.save
drwx------ 4 fundamental fundamental 4096 Nov 15 21:24 snap
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Templates
drwxrwxr-x 2 fundamental fundamental 4096 Nov 15 19:56 text_folder
drwxr-xr-x 2 fundamental fundamental 4096 Nov  9 21:15 Videos
fundamental@fundamental-VirtualBox:~$
```

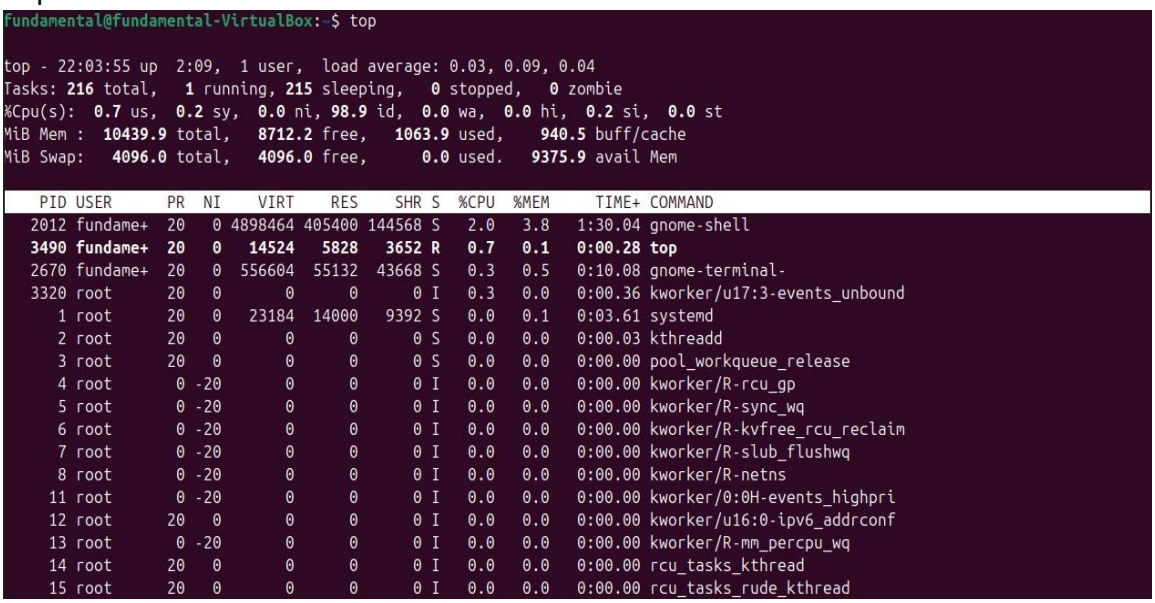
14. ps aux
    - syntax: ps aux
    - use case: to find resource consuming processes
    - description: shows full list of running processes with memory , CPU usage and owner
    - implementation:

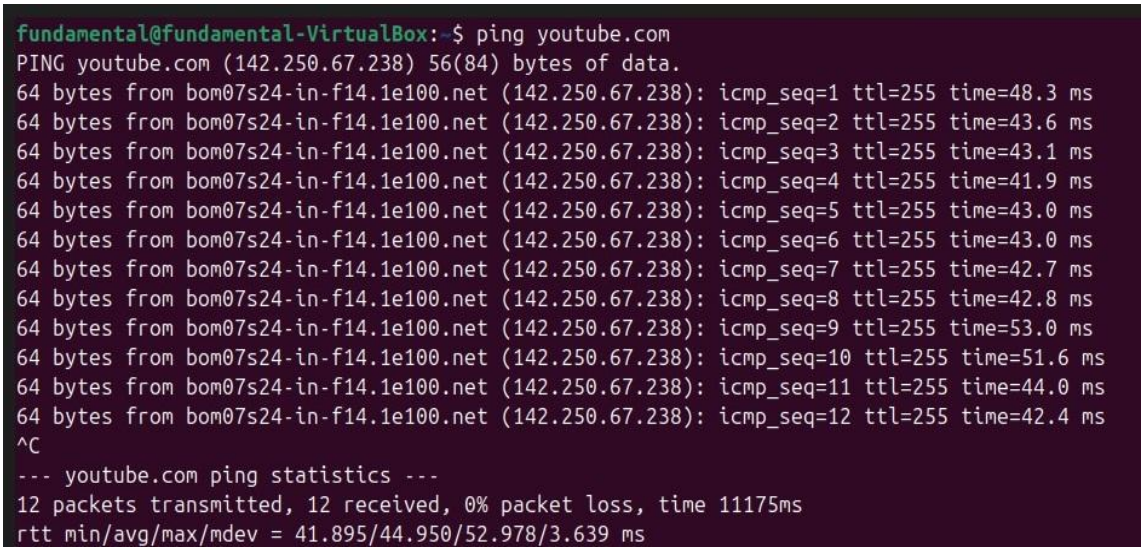
```
Desktop  Documents  Downloads  lab1  Music  new.file  Pictures  Public  snap  Templates  text_folder  Videos
fundamental@fundamental-VirtualBox:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.1  23184 14000 ?        Ss   19:54   0:03 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    19:54   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    19:54   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-rcu_gp]
root           5  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-sync_wq]
root           6  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-kvfree_rcu_reclaim]
root           7  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-slub_flushwq]
root           8  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-netns]
root          11  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/0:0H-events_highpri]
root          12  0.0  0.0      0     0 ?        I    19:54   0:00 [kworker/u16:0-ipv6_addrconf]
root          13  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/R-mm_percpu_wq]
root          14  0.0  0.0      0     0 ?        I    19:54   0:00 [rcu_tasks_kthread]
root          15  0.0  0.0      0     0 ?        I    19:54   0:00 [rcu_tasks_rude_kthread]
root          16  0.0  0.0      0     0 ?        I    19:54   0:00 [rcu_tasks_trace_kthread]
root          17  0.0  0.0      0     0 ?        S    19:54   0:00 [ksoftirqd/0]
root          18  0.0  0.0      0     0 ?        I    19:54   0:01 [rcu_preempt]
root          19  0.0  0.0      0     0 ?        S    19:54   0:00 [rcu_exp_par_gp_kthread_worker/0]
root          20  0.0  0.0      0     0 ?        S    19:54   0:00 [rcu_exp_gp_kthread_worker]
root          21  0.0  0.0      0     0 ?        S    19:54   0:00 [migration/0]
root          22  0.0  0.0      0     0 ?        S    19:54   0:00 [idle_inject/0]
root          23  0.0  0.0      0     0 ?        S    19:54   0:00 [cpuhp/0]
root          24  0.0  0.0      0     0 ?        S    19:54   0:00 [cpuhp/1]
root          25  0.0  0.0      0     0 ?        S    19:54   0:00 [idle_inject/1]
root          26  0.0  0.0      0     0 ?        S    19:54   0:01 [migration/1]
root          27  0.0  0.0      0     0 ?        S    19:54   0:00 [ksoftirqd/1]
root          29  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/1:0H-events_highpri]
root          30  0.0  0.0      0     0 ?        S    19:54   0:00 [cpuhp/2]
root          31  0.0  0.0      0     0 ?        S    19:54   0:00 [idle_inject/2]
root          32  0.0  0.0      0     0 ?        S    19:54   0:01 [migration/2]
root          33  0.0  0.0      0     0 ?        S    19:54   0:00 [ksoftirqd/2]
root          35  0.0  0.0      0     0 ?        I<   19:54   0:00 [kworker/2:0H-kblockd]
root          36  0.0  0.0      0     0 ?        S    19:54   0:00 [cpuhp/3]
root          37  0.0  0.0      0     0 ?        S    19:54   0:00 [idle_inject/3]
```

15. ps
    - syntax: ps
    - use case: Check if a program started from where terminal is running
    - description: shows only processes related to current terminal session
    - implementation:

```
fundamental@fundamental-VirtualBox:~$ ps
    PID TTY          TIME CMD
   2677 pts/0    00:00:00 bash
   3529 pts/0    00:00:00 ps
fundamental@fundamental-VirtualBox:~$
```

16. top

- syntax: top
- use case: monitor CPU/memory while running apps
- description: displays real time performance statistics
- implementation:
- 

17. ping
    - syntax: ping <host>
    - use case: test internet connections
    - description: checks network connectivity and response time
    - implementation:
    - 

18. netstat –a
    - syntax: netstat –a
    - use case: troubleshoot port conflicts between applications. And check if server app started listening
    - description: displays all active connections and listening ports
    - implementations:

```
fundamental@fundamental-VirtualBox:~$ netsat -a
Command 'netsat' not found, did you mean:
  command 'netcat' from deb netcat-openbsd (1.226-1ubuntu1)
  command 'netcat' from deb ncat (7.94+git20230807.3be01efb1+dfsg-2)
  command 'netcat' from deb netcat-traditional (1.10-48)
  command 'netstat' from deb net-tools (2.10-0.1ubuntu4.4)
Try: sudo apt install <deb name>
fundamental@fundamental-VirtualBox:~$ netsat -tulpn
Command 'netsat' not found, did you mean:
  command 'netcat' from deb netcat-openbsd (1.226-1ubuntu1)
  command 'netcat' from deb ncat (7.94+git20230807.3be01efb1+dfsg-2)
  command 'netcat' from deb netcat-traditional (1.10-48)
  command 'netstat' from deb net-tools (2.10-0.1ubuntu4.4)
Try: sudo apt install <deb name>
```

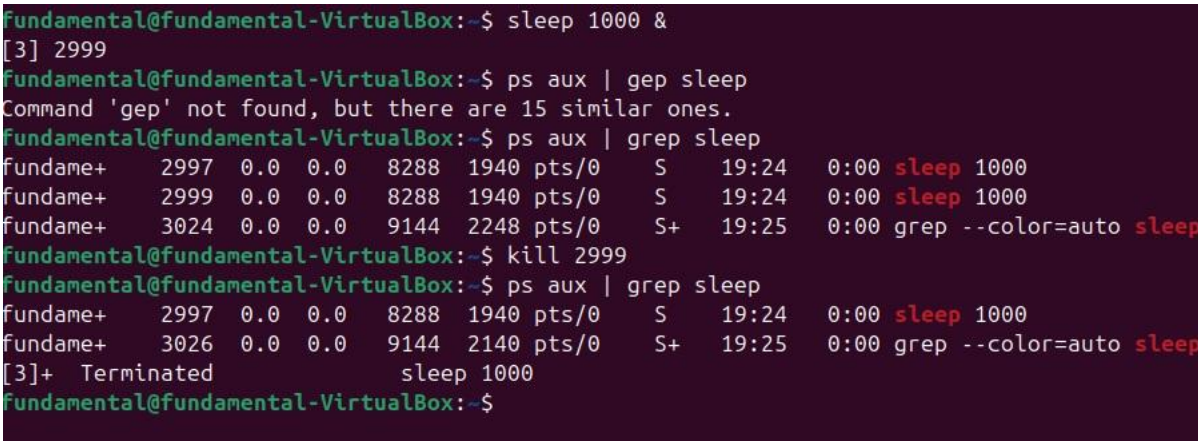
19. ip addr
    - syntax : ip addr
    - use case: find system IP address on LAN
    - descriptions: shows IP details of system network interfaces
    - implementation:

```
fundamental@fundamental-VirtualBox:~$ ip addr
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 noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d4:33:59 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 78143sec preferred_lft 78143sec
    inet6 fd17:625c:f037:2:2451:6415:8838:8d4c/64 scope global temporary dynamic
       valid_lft 85918sec preferred_lft 13918sec
    inet6 fd17:625c:f037:2:a00:27ff:fed4:3359/64 scope global dynamic mngtmpaddr
       valid_lft 85918sec preferred_lft 13918sec
    inet6 fe80::a00:27ff:fed4:3359/64 scope link
       valid_lft forever preferred_lft forever
```

20. ip link
    - syntax: ip link
      ip link set <interface> up/down
    - use case: disable Wi-Fi interface temporarily
    - description: shows and configures network interfaces
    - implementation:

```
fundamental@fundamental-VirtualBox:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:d4:33:59 brd ff:ff:ff:ff:ff:ff
```

21. kill
    - syntax: kill [signal] <PID>
    - use case: This command is essential for managing and controlling the processes that run on the Linux system

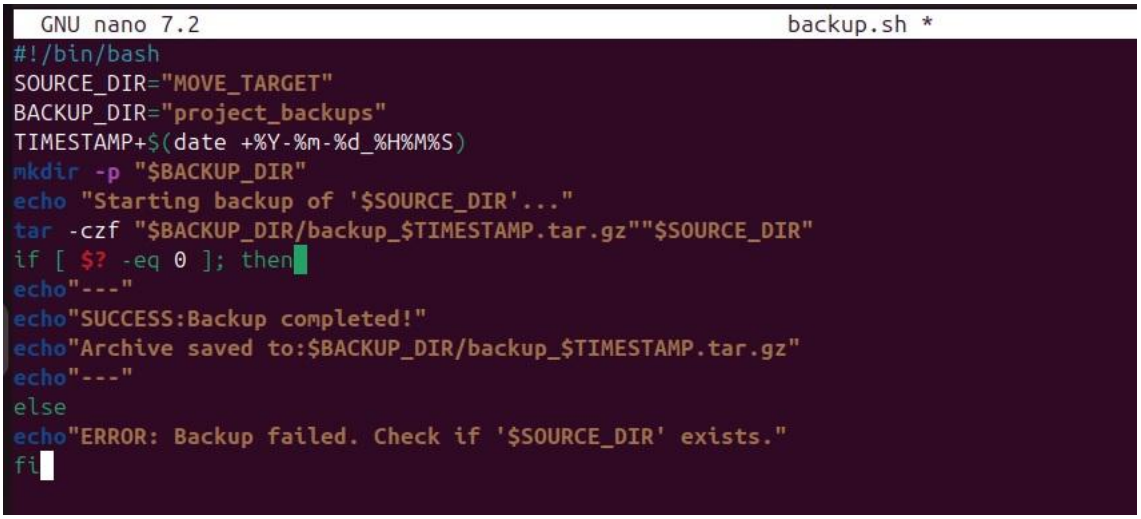
- description : The “kill” command is used to send signals to processes, typically to terminate them.
- Implementation:
- 

#Scripting

Script 1 : Backup a Directory

Purpose: To create a backup of a specified directory by copying all its contents into a backup folder with a timestamp. This helps in maintaining safe copies of important files

SCRIPT:



Output:

```
fundamental@fundamental-VirtualBox:~$ nano backup.sh
fundamental@fundamental-VirtualBox:~$ ./backup.sh
./backup.sh: line 4: TIMESTAMP+2025-11-18_211405: command not found
Starting backup of 'MOVE_TARGET'...
tar: Cowardly refusing to create an empty archive
Try 'tar --help' or 'tar --usage' for more information.
./backup.sh: line 14: echoERROR: Backup failed. Check if 'MOVE_TARGET' exists.: command not found
fundamental@fundamental-VirtualBox:~$
```

Explanation : This script uses the tar command to combine and compress the files (-c for create, -z for gzip compression, -f for specifying the filename). It uses the date command output to ensure every backup file has a unique name

SCRIPT 2: CPU / Memory Monitoring

Purpose: To log CPU and memory usage at regular intervals into a log file. This is useful for monitoring system performance

Script :

```
 GNU nano 7.2                                    monitor.sh *
#!/bin/bash
echo "---System Status Log (New Entry)---" >> performance_report.txt
date >> performance_report.txt
echo "---Memory Usage (MB)---" >> performance_report.txt
free -m >> performance_report.txt
echo "---System Uptime---" >> performance_report.txt
uptime >> performance_report.txt
echo "Log saved."
```

Output :

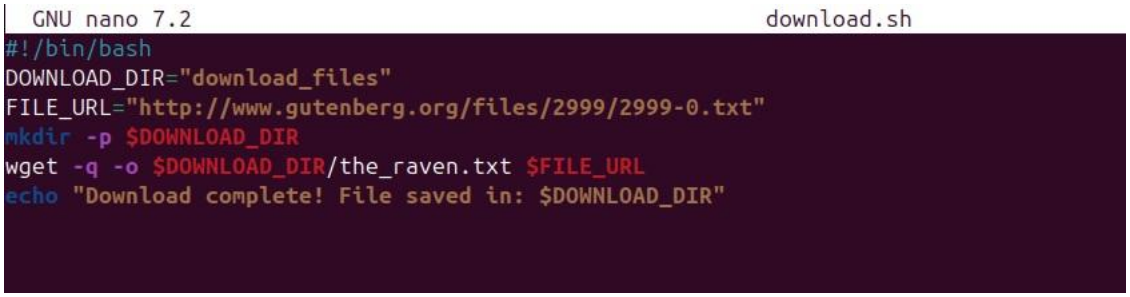
```
fundamental@fundamental-VirtualBox:~$ ./download.sh
Download complete! File saved in: download_files
fundamental@fundamental-VirtualBox:~$ ls download_files
the_raven.txt
fundamental@fundamental-VirtualBox:~$ nano monitor.sh
fundamental@fundamental-VirtualBox:~$ touch monitor.sh
fundamental@fundamental-VirtualBox:~$ nano monitor.sh
fundamental@fundamental-VirtualBox:~$ chmod +x monitor.sh
fundamental@fundamental-VirtualBox:~$ ./monitor.sh
Log saved.
fundamental@fundamental-VirtualBox:~$ ./monitor.sh
Log saved.
fundamental@fundamental-VirtualBox:~$ cat performance_reprt.txt
cat: performance_reprt.txt: No such file or directory
fundamental@fundamental-VirtualBox:~$ cat performance_report.txt
---System Status Log (New Entry)---
Tue Nov 18 08:59:45 PM IST 2025
---Memory Usage (MB)---
               total        used        free      shared  buff/cache   available
Mem:           10439        1049        8740          33         924        9389
Swap:           4095           0        4095
---System Uptime---
 20:59:45 up  1:51,  1 user,  load average: 0.04, 0.09, 0.08
---System Status Log (New Entry)---
Tue Nov 18 08:59:51 PM IST 2025
---Memory Usage (MB)---
               total        used        free      shared  buff/cache   available
Mem:           10439        1054        8735          33         924        9385
Swap:           4095           0        4095
---System Uptime---
 20:59:51 up  1:51,  1 user,  load average: 0.04, 0.08, 0.08
fundamental@fundamental-VirtualBox:~$
```

Explanation : This script uses the redirection operator >> to append the output of date, free-m, and uptime commands to the performance_report.txt file without overwriting previous entries

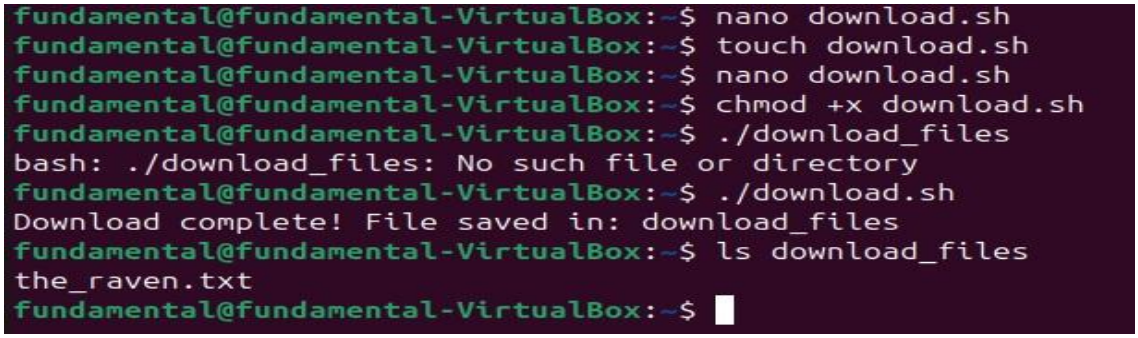
SCRIPT 3 : Automated download task

Purpose : To download a file from the internet using either wget or curl and save it inside a predefined download directory

Script :



Output :



Explanation : The script uses mkdir -p to guarantee the target folder exists. The wget command handles

the actual file transfer from the defines URL directly into the download_files folder.

How to upload file and submission of github repositories

To complete this step, I have created a GitHub repository to store all my Linux Shell Scripts, screenshots, and the assignment report. GitHub helps in saving code online and provides version control. 4.1: Repository Creation :

1. I logged into my Github account at(https://github.com)
2. I clicked on “New Repository”
3. I named the repository linux-shell-assignment
4. I selected Public so it can be viewed easily 5. I clicked Create Repository.
5. Here is my github repository link : https://github.com/priyanshi220383-max/linux-CSFCAP-assigment

Uploading Files to github

Inside the repository, I uploaded the required files:

- All three shell scripts
  - backup_dir.sh
  - monitor_sys.sh
  - auto_download.sh
- Screenshots (Linux installation, commands, scripts outputs)
- My assignment report (DOCX/PDF)

REFLECTION

The practical phase of this assignment primarily challenged me to master shell script, where issues like missing spaces in conditional checks required dedicated debugging. This process reinforced the importance of precise automation logic and led me directly to proficiency in essential techniques, including command chaining and output redirection (>>) for creating persistent logs. These skills translate directly into critical real-world functions: the monitor.sh script models essential server health checks (CPU/memory), the backup.sh script ensures data resilience and disaster recovery through timestamped backups, and the download.sh script demonstrates fetching dependencies required for DevOps pipelines. Successfully implementing these scripts confirms a foundational understanding of Linux as a powerful automation platform.