

K.R. MANGALAMUNIVERSITY

Course Name: Computer Science Fundamentals and Career Pathways

Course Code: ETCCCP105

Assignment Number 02: Basics of Linux and Open-Source Tools

Assignment Title: How Can You Demonstrate Linux Setup, Command Usage, and Automation Through Practical Implementation?

Submitted by: Md Shahjad

2501010169

B.Tech CSE Core

Section A

Semester 1

Submitted to: Mr. Rajesh Kumar

★ 1. FILE NAVIGATION COMMANDS

◆ 1.1 Command: ls

Syntax :- ls

Description

ls lists all files and directories present in the current working directory. It helps users see folder contents quickly.

When & Why It Is Used

- **When:** When users need to see the items available in the current directory.
- **Why:** The terminal does not visually display files like a graphical interface, so ls is necessary to inspect directory contents.

◆ 1.2 Command: cd

Syntax :- cd <directory-name>

Description :-

Changes the current working directory. It allows the user to move between different folders in the file system.

When & Why It Is Used

- **When:** When the user needs to work inside a specific folder.
- **Why:** Most commands operate relative to the current directory, so navigation is essential.

◆ 1.3 Command: pwd

Syntax :- pwd

Description

Prints the absolute path of the current directory. This helps confirm the exact location of the user within the file system.

When & Why It Is Used

- **When:** When working inside nested folders and the user needs to verify the current path.
- **Why:** Prevents executing commands in the wrong directory, avoiding accidental data changes.

◆ 1.4 Command: tree

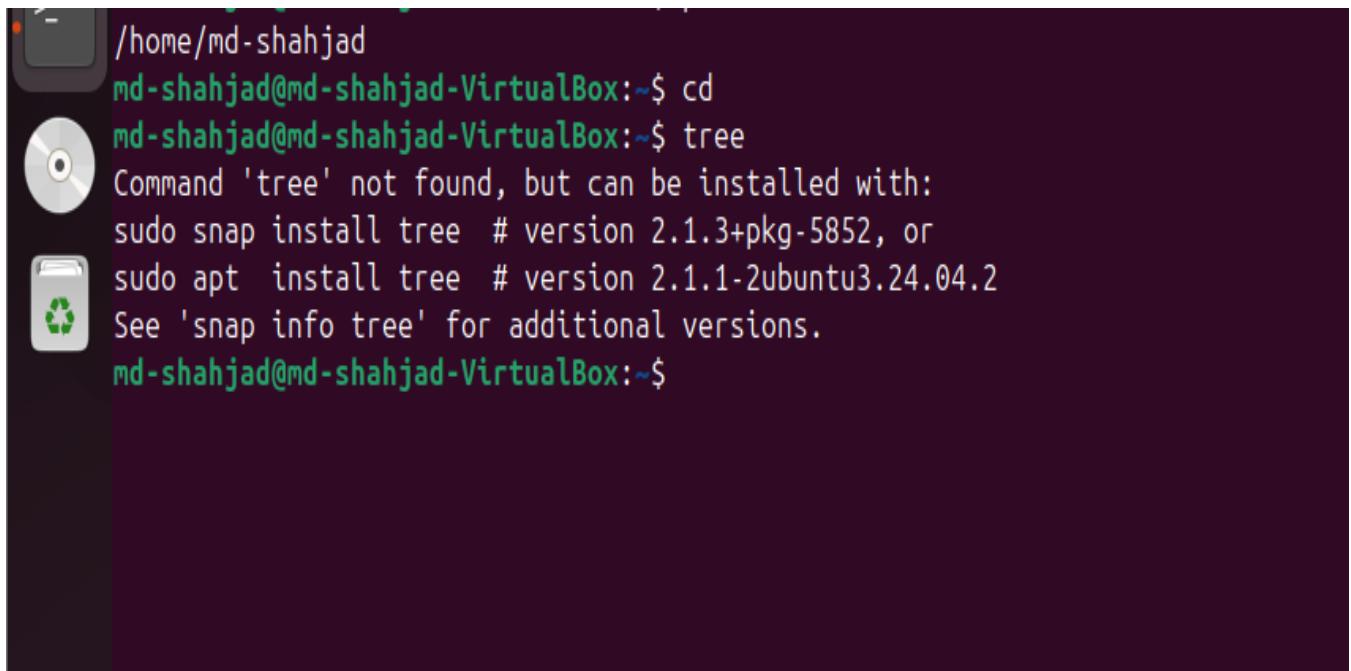
Syntax :- tree

Description

Displays the directory structure in a hierarchical tree format, including nested files and subdirectories.

When & Why It Is Used

- **When:** When the user wants to view the complete structure of a project or directory.
- **Why:** Unlike ls, which shows only one level, tree displays the full hierarchy, making organization easier to understand.



The image shows a terminal window on an Ubuntu desktop. The terminal has a dark background with light-colored text. The user's session is as follows:

```
/home/md-shahjad  
md-shahjad@md-shahjad-VirtualBox:~$ cd  
md-shahjad@md-shahjad-VirtualBox:~$ tree  
Command 'tree' not found, but can be installed with:  
sudo snap install tree # version 2.1.3+pkg-5852, or  
sudo apt install tree # version 2.1.1-2ubuntu3.24.04.2  
See 'snap info tree' for additional versions.  
md-shahjad@md-shahjad-VirtualBox:~$
```

★ 2. FILE & DIRECTORY MANAGEMENT COMMANDS (*mkdir, touch, cp, mv, rm*)

◆ 2.1 Command: **mkdir**

Syntax :- `mkdir <folder-name>`

Description

Creates a new directory inside the file system.

When & Why It Is Used

- **When:** When creating a new folder to organize data.
- **Why:** Proper directory management helps in maintaining an organized file system.

◆ 2.2 Command: **touch**

Syntax :- `touch <file-name>`

Description

Creates an empty file or updates the timestamp of an existing file.

When & Why It Is Used

- **When:** When a new file needs to be created quickly.
- **Why:** Often used before editing or scripting to generate placeholder files.

◆ 2.3 Command: **cp**

Syntax :- `cp <source> <destination>`

Description

Copies files or directories from one location to another.

When & Why It Is Used

- **When:** When creating backups or duplicating files.

- **Why:** To preserve original data while working on copies.

◆ 2.4 Command: mv

Syntax :- mv <source> <destination>

Description

Moves or renames files and directories.

When & Why It Is Used

- **When:** When renaming a file or reorganizing items inside folders.
- **Why:** Helps maintain a clean and structured directory layout.

◆ 2.5 Command: rm

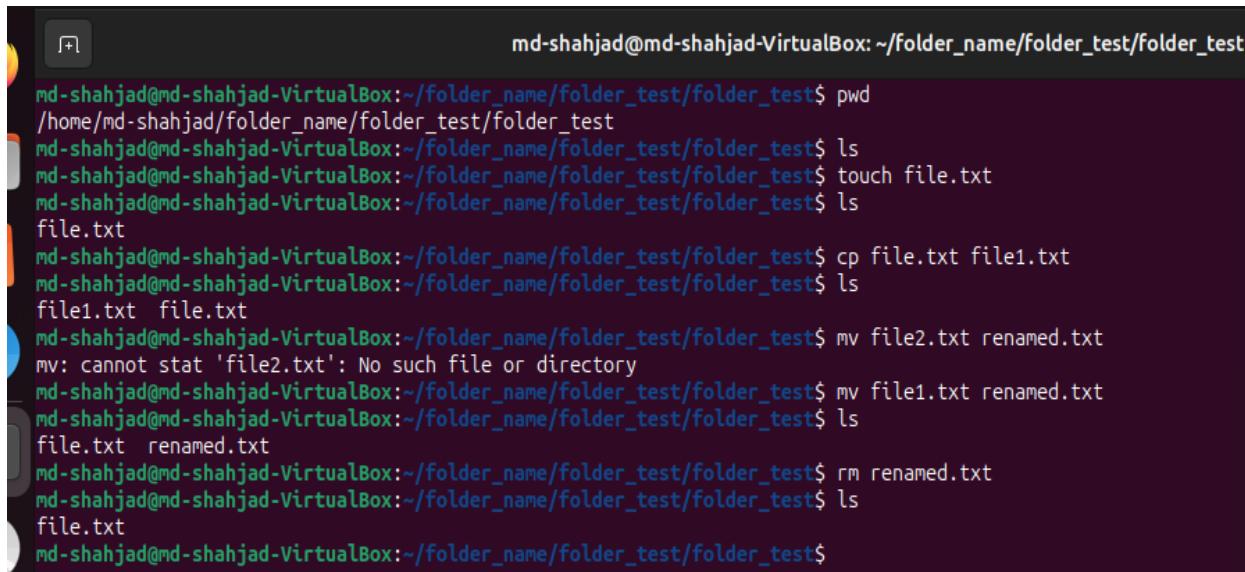
Syntax :- rm <file-name>

Description

Permanently deletes files or directories.

When & Why It Is Used

- **When:** When removing unwanted or unnecessary files.
- **Why:** Helps free storage and keep the file system clean.



```
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ pwd
/home/md-shahjad/folder_name/folder_test/folder_test
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ ls
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ touch file.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ ls
file.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ cp file.txt file1.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ ls
file1.txt file.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ mv file2.txt renamed.txt
mv: cannot stat 'file2.txt': No such file or directory
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ mv file1.txt renamed.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ ls
file.txt renamed.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ rm renamed.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ ls
file.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$
```

★ 3. PERMISSIONS MANAGEMENT COMMANDS(*chmod, chown*)

◆ 3.1 Command: chmod

Syntax :- chmod <permissions> <file>

Description

Changes read, write, and execute permissions for a file or directory.

When & Why It Is Used

- **When:** When a file or script needs specific access rights.
- **Why:** Ensures proper security and prevents unauthorized access.

3.2 Command: chown

Syntax :- chown <user>:<group> <file>

Description

Changes the ownership of a file or directory.

When & Why It Is Used

- **When:** When a file must belong to a specific user or group.
- **Why:** Necessary in multi-user systems to ensure correct access control.

```
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ touch demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ mkdir permission_test
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test$ cd permission_test
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ touch demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ ls -l
total 0
-rw-rw-r-- 1 md-shahjad md-shahjad 0 Nov 16 08:37 demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ chmod u+x demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ ls -l
total 0
-rwxrwxr-- 1 md-shahjad md-shahjad 0 Nov 16 08:37 demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ sudo chown md-shahjad de
```

★ 4. PROCESS MONITORING COMMANDS(*ps, top, kill*)

◆ 4.1 Command: ps

Syntax :- ps

Description

Displays the processes currently running under the active shell session.

When & Why It Is Used

- **When:** When users want to check which processes are running.
- **Why:** Provides the Process ID (PID), which is necessary for monitoring or terminating processes.

◆ 4.2 Command: top

Syntax :- top

Description

Shows real-time information about system performance including CPU usage, memory usage, running tasks, and load average.

When & Why It Is Used

- **When:** When the system becomes slow or performance issues appear.
- **Why:** Helps identify which processes are consuming high system resources

◆ 4.3 Command: kill

Syntax :- kill <PID>

Description

Terminates a running process using its Process ID.

When & Why It Is Used

- **When:** When a program freezes or becomes unresponsive.
- **Why:** Allows users to forcefully stop malfunctioning processes.

```

total 0
-rwxrw-r-- 1 md-shahjad md-shahjad 0 Nov 16 08:37 demo.txt
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ ps
  PID TTY      TIME CMD
 2752 pts/1    00:00:00 bash
 7075 pts/1    00:00:00 ps
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ ps aux
USER      PID %CPU %MEM   VSZ   RSS TTY      STAT START  TIME COMMAND
root        1  0.0  0.3 23272 13908 ?        Ss  03:47  0:10 /sbin/init splash
root        2  0.0  0.0     0   0 ?        S   03:47  0:00 [kthreadd]
root        3  0.0  0.0     0   0 ?        S   03:47  0:00 [pool_workqueue_release]
root        4  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-rCU_gp]
root        5  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-sync_wq]
root        6  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-kvfree_rcu_reclaim]
root        7  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-slab_flushwq]
root        8  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-netns]
root       11  0.0  0.0     0   0 ?        I<  03:47  0:03 [kworker/0:0H-kblockd]
root       12  0.0  0.0     0   0 ?        I   03:47  0:00 [kworker/u12:0]
root       13  0.0  0.0     0   0 ?        I<  03:47  0:00 [kworker/R-mm_percpu_wq]
root       14  0.0  0.0     0   0 ?        I   03:47  0:00 [scu_tasks_kthreadd]

```

```

root      7074  0.0  0.0     0   0 ?        I   08:40  0:00 [kworker/1:2]
md-shah+  7076 100  0.1 22284 4624 pts/1    R+  08:40  0:00 ps aux
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ top

```

```

top - 08:41:08 up 4:53, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 208 total, 2 running, 206 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 1.2 sy, 0.0 ni, 98.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.2 total, 1213.4 free, 1127.7 used, 1839.8 buff/cache
MiB Swap: 3916.0 total, 3916.0 free, 0.0 used. 2788.6 avail Mem
Change delay from 3.0 to

```

```

> - root      7074  0.0  0.0     0   0 ?        I   08:40  0:00 [kworker/1:2]
md-shah+  7078  0.0  0.1 22284 4624 pts/1    R+  08:41  0:00 ps aux
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ kill 7078
bash: kill: (7078) - No such process
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ kill 6917
md-shahjad@md-shahjad-VirtualBox:~/folder_name/folder_test/folder_test/permission_test$ 

```



5. NETWORKING TOOLS (*ping, ifconfig/ip, netstat*)

◆ 5.1 Command: ping

Syntax :- ping <hostname>

Description

Checks connectivity between the local machine and a remote server by sending ICMP packets.

When & Why It Is Used

- **When:** When internet or server connectivity needs to be tested.
- **Why:** Measures packet loss and response time for diagnosing network issues.

◆ 5.2 Command: ip (or ifconfig)

Syntax :- ip a

Description

Displays all network interfaces, IP addresses, and connection details.

When & Why It Is Used

- **When:** When checking the system's IP address or network configuration.
- **Why:** Essential for troubleshooting network issues.

◆ 5.3 Command: netstat

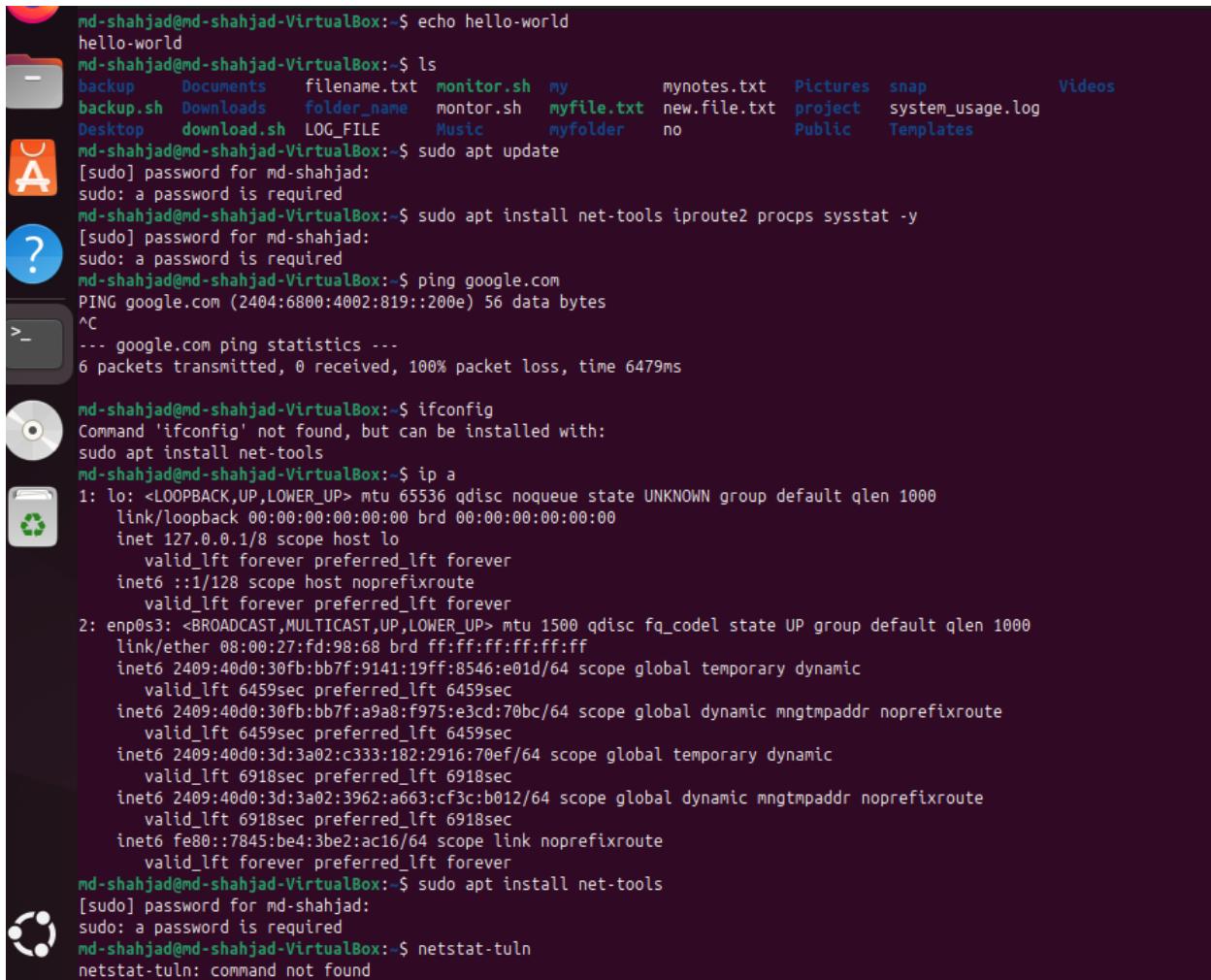
Syntax :- netstat -an

Description

Shows active network connections, open ports, and listening services.

When & Why It Is Used

- **When:** When analyzing network issues or checking for suspicious activity.
- **Why:** Helps monitor network status and diagnose communication problems.



```
md-shahjad@md-shahjad-VirtualBox:~$ echo hello-world
hello-world
md-shahjad@md-shahjad-VirtualBox:~$ ls
backup  Documents  filename.txt  monitor.sh  my  mynotes.txt  Pictures  snap  Videos
backup.sh  Downloads  folder_name  monitor.sh  myfile.txt  new.file.txt  project  system_usage.log
Desktop  download.sh  LOG_FILE  Music  myfolder  no  Public  Templates
md-shahjad@md-shahjad-VirtualBox:~$ sudo apt update
[sudo] password for md-shahjad:
sudo: a password is required
md-shahjad@md-shahjad-VirtualBox:~$ sudo apt install net-tools iproute2 procps sysstat -y
[sudo] password for md-shahjad:
sudo: a password is required
md-shahjad@md-shahjad-VirtualBox:~$ ping google.com
PING google.com (2404:6800:4002:819::200e) 56 data bytes
^C
--- google.com ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 6479ms

md-shahjad@md-shahjad-VirtualBox:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
md-shahjad@md-shahjad-VirtualBox:~$ 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 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:fd:98:68 brd ff:ff:ff:ff:ff:ff
        inet6 2409:40d0:30fb:bb7f:9141:19ff:8546:e01d/64 scope global temporary dynamic
            valid_lft 6459sec preferred_lft 6459sec
        inet6 2409:40d0:30fb:bb7f:a9a8:f975:e3cd:70bc/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 6459sec preferred_lft 6459sec
        inet6 2409:40d0:3d:3a02:c333:182:2916:70ef/64 scope global temporary dynamic
            valid_lft 6918sec preferred_lft 6918sec
        inet6 2409:40d0:3d:3a02:3962:a663:cf3c:b012/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 6918sec preferred_lft 6918sec
        inet6 fe80::7845:be4:3be2:ac16/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
md-shahjad@md-shahjad-VirtualBox:~$ sudo apt install net-tools
[sudo] password for md-shahjad:
sudo: a password is required
md-shahjad@md-shahjad-VirtualBox:~$ netstat-tuln
netstat-tuln: command not found
```

★ SCRIPT 1 — Backup Script (`backup.sh`)

Explanation

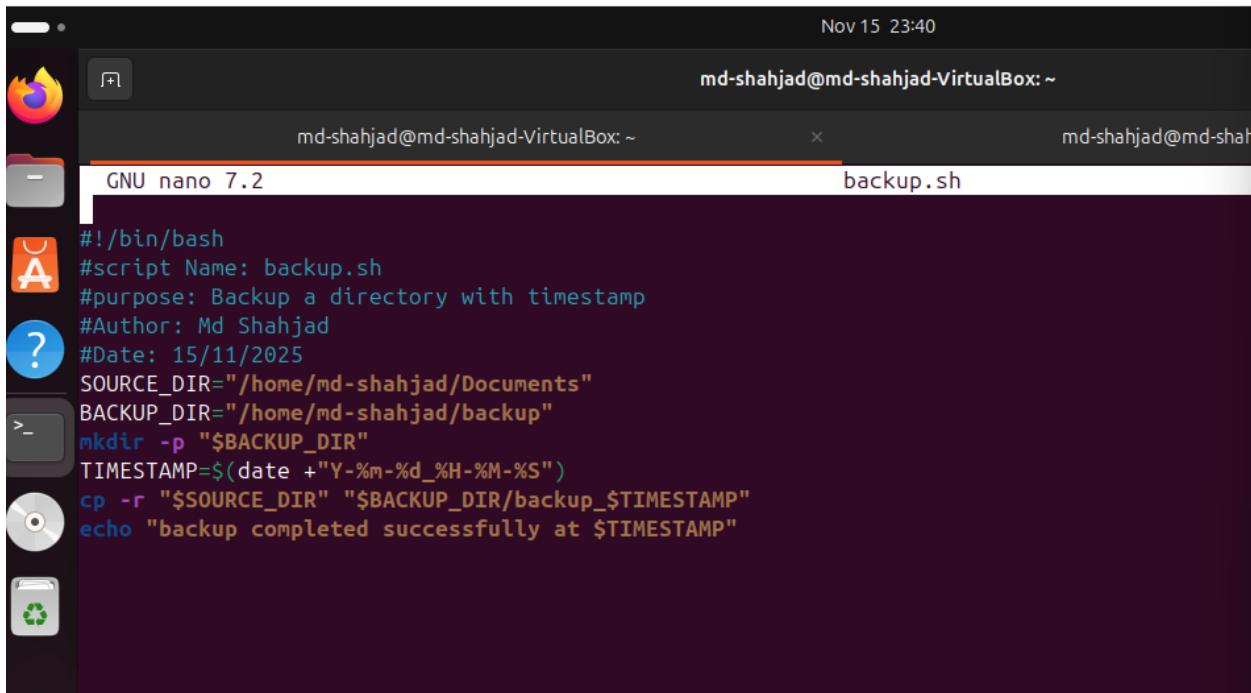
The purpose of the `backup.sh` script is to create a secure backup of a selected directory by copying its entire content to another location. This script helps prevent data loss by keeping a timestamped copy that users can restore later if needed.

The script begins by defining two variables:

- **SOURCE_DIR** – the directory whose data needs to be backed up
- **BACKUP_DIR** – the directory where the backup will be stored

It then ensures that the backup folder exists by using the `mkdir -p` command. A timestamp is generated using the `date` command to create unique backup names, ensuring that no previous backups are overwritten.

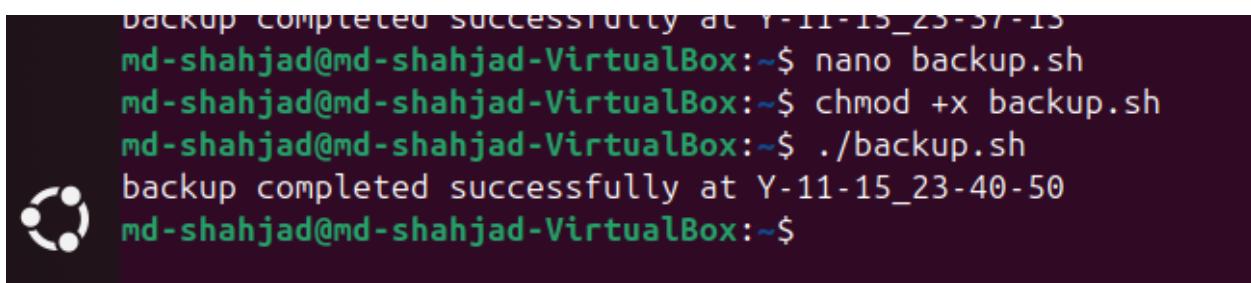
The `cp -r` command performs the actual backup by copying all files and subdirectories from the source to the backup folder. Finally, a success message is displayed along with the timestamp. This script is commonly used in system administration and personal data protection tasks.



Nov 15 23:40
md-shahjad@md-shahjad-VirtualBox: ~

GNU nano 7.2 backup.sh

```
#!/bin/bash
#script Name: backup.sh
#purpose: Backup a directory with timestamp
#Author: Md Shahjad
#Date: 15/11/2025
SOURCE_DIR="/home/md-shahjad/Documents"
BACKUP_DIR="/home/md-shahjad/backup"
mkdir -p "$BACKUP_DIR"
TIMESTAMP=$(date +"Y-%m-%d_%H-%M-%S")
cp -r "$SOURCE_DIR" "$BACKUP_DIR/backup_${TIMESTAMP}"
echo "backup completed successfully at ${TIMESTAMP}"
```



```
Backup completed successfully at Y-11-15_23-37-15
md-shahjad@md-shahjad-VirtualBox:~$ nano backup.sh
md-shahjad@md-shahjad-VirtualBox:~$ chmod +x backup.sh
md-shahjad@md-shahjad-VirtualBox:~$ ./backup.sh
backup completed successfully at Y-11-15_23-40-50
md-shahjad@md-shahjad-VirtualBox:~$
```

★ SCRIPT 2 — System Monitoring Script (`monitor.sh`)

Explanation

The `monitor.sh` script is designed to continuously monitor a system's performance by recording CPU and memory usage at fixed intervals. This type of script is essential for detecting performance bottlenecks, analyzing system behavior, and diagnosing heavy resource consumption.

The script creates or updates a log file called `system_usage.log`, where all monitoring data is stored. Inside an infinite while loop, it performs the following operations every 5 seconds:

1. **Logs the current timestamp**, indicating when the reading was taken.
2. **Captures CPU usage**, using the first few lines of the `top` command in batch mode.
3. **Captures memory usage**, using the `free -h` command.
4. **Adds a separator line** for readability.

By running repeatedly, this script builds a detailed performance history. The user can stop the script manually using `CTRL + C`. Such monitoring scripts are widely used in DevOps, server administration, and system performance analysis.



```
#!/bin/bash
#Script Name : monitor.sh
#Purpose : To log CPU and memory usage at regular intervals.
#Author : Md Shahjad
#Date : 16/11/2025
LOG_FILE="/home/md-shahjad/system_usage.log"
INTERVAL=5
echo "system monitoring started.."
echo "Logging CPU and memory usage every $INTERVAL seconds."
echo "....." >> "$LOG_FILE"
while true
do
    free -b -n 1 | head -5 >> "$LOG_FILE"
    free -h >> "$LOG_FILE"
    echo "...." >> "$LOG_FILE"
    sleep $INTERVAL
done
```

★ SCRIPT 3 — File Download Script (`download.sh`)

Explanation

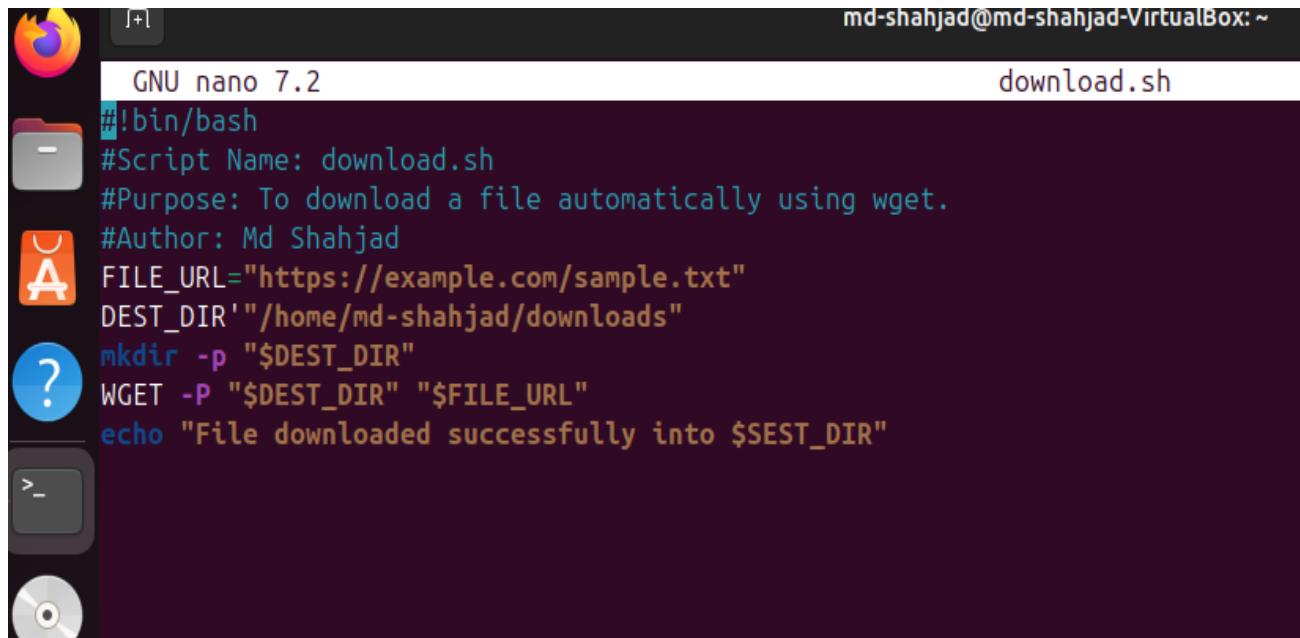
The `download.sh` script automates the process of downloading a file from the internet using the `wget` command. Automating downloads reduces manual effort, ensures consistency, and is particularly useful for scheduled updates, backups, and large-scale system operations.

The script begins by defining two key variables:

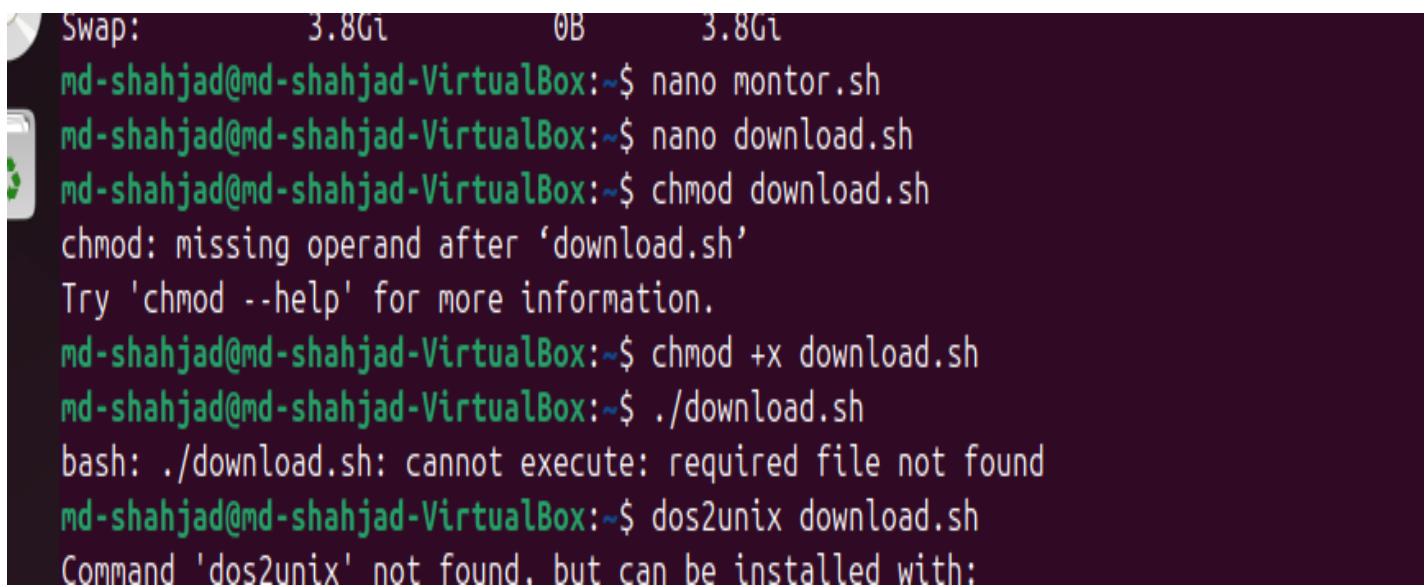
- **FILE_URL** – the link to the file that needs to be downloaded
- **DEST_DIR** – the directory where the downloaded file will be stored

Before downloading, the script checks if the destination directory exists and creates it using `mkdir -p` if necessary. The `wget -P` command downloads the file directly into the specified directory.

After the download completes, a confirmation message is shown to inform the user. This script is useful in environments where frequent or automated downloading is required, such as software updates, backups, or scheduled data collection tasks.



```
GNU nano 7.2 download.sh
#!/bin/bash
#Script Name: download.sh
#Purpose: To download a file automatically using wget.
#Author: Md Shahjad
FILE_URL="https://example.com/sample.txt"
DEST_DIR="/home/md-shahjad/downloads"
mkdir -p "$DEST_DIR"
wget -P "$DEST_DIR" "$FILE_URL"
echo "File downloaded successfully into $DEST_DIR"
```



```
Swap: 3.8Gi 0B 3.8Gi
md-shahjad@md-shahjad-VirtualBox:~$ nano monitor.sh
md-shahjad@md-shahjad-VirtualBox:~$ nano download.sh
md-shahjad@md-shahjad-VirtualBox:~$ chmod download.sh
chmod: missing operand after 'download.sh'
Try 'chmod --help' for more information.
md-shahjad@md-shahjad-VirtualBox:~$ chmod +x download.sh
md-shahjad@md-shahjad-VirtualBox:~$ ./download.sh
bash: ./download.sh: cannot execute: required file not found
md-shahjad@md-shahjad-VirtualBox:~$ dos2unix download.sh
Command 'dos2unix' not found, but can be installed with:
```