

Experiment 2: Linux File System Permissions and Essential Commands

Aim

The aim of this experiment is to study Linux file system permissions and to practice some essential Linux commands. By doing this, we learn how to manage files, directories, and their access rights using command line.

Requirements

1. A system with Linux operating system installed (Ubuntu, Fedora, etc.).
2. Access to a terminal (command line interface).
3. Basic knowledge of Linux directory structure.

COMMANDS WITH THEIR EXPLANATION

Command	Purpose	Example
pwd	Show current directory	pwd
ls	List files	ls -la
cd	Change directory	cd /home
mkdir	Create directory	mkdir newfolder
rmdir	Remove empty directory	rmdir oldfolder
touch	Create empty file	touch newfile.txt

Command	Purpose	Example
cp	Copy files	cp file1.txt file2.txt
mv	Move/rename files	mv old.txt new.txt
rm	Delete files	rm -i file.txt
cat	Display file content	cat file.txt
less	View file with pagination	less largefile.txt
head	Show file beginning	head -n 5 file.txt
tail	Show file end	tail -n 5 file.txt
nano	Simple text editor	nano file.txt
chmod	Change permissions	chmod 755 script.sh
chown	Change ownership	sudo chown user:group
whoami	Current user	whoami
sudo	Run as administrator	sudo command
history	Command history	h

Procedure

1. Open the terminal in Linux.
2. Use pwd to check the present working directory.
3. Use ls to list files and directories.
4. Create a directory using mkdir testdir.
5. Change directory with cd testdir.
6. Create an empty file using touch file1.txt.
7. Use ls -l to check file permissions (read, write, execute).
8. Modify permissions with chmod (for example, chmod 755 file1.txt).
9. Change file ownership with chown if needed.
10. Display file content with cat file1.txt.
11. Finally, remove file using rm file1.txt.

LAB EXERCISE

EXERCISE 1: FILE SYSTEM NAVIGATION

EXERCISE 2: FILE OPERATIONS AND PERMISSIONS

EXERCISE 3: TEXT EDITING AND VIEWING

COMMANDS AND OUTPUT

```
pavani@UBUNTU: ~/projects/linux_practice/documents
pavani@UBUNTU:~$ cd
pavani@UBUNTU:~$ pwd
/home/pavani
pavani@UBUNTU:~$ mkdir -p projects/linux_practice/{scripts,documents,backup}
pavani@UBUNTU:~$ cd projects/linux_practice/scripts
pavani@UBUNTU:~/projects/linux_practice/scripts$ touch setup.sh cleanup.sh readme.txt
pavani@UBUNTU:~/projects/linux_practice/scripts$ ls -la
total 8
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:50 .
drwxrwxr-x 6 pavani pavani 4096 Sep 24 18:20 ..
-rw-rw-r-- 1 pavani pavani  0 Sep 24 18:21 cleanup.sh
-rw-rw-r-- 1 pavani pavani  0 Sep 24 18:21 readme.txt
-rw-rw-r-- 1 pavani pavani  0 Sep 24 18:21 setup.sh
pavani@UBUNTU:~/projects/linux_practice/scripts$ cd ..
pavani@UBUNTU:~/projects/linux_practice$ ls -la
total 24
drwxrwxr-x 6 pavani pavani 4096 Sep 24 18:20 .
drwxrwxr-x 3 pavani pavani 4096 Sep 23 14:46 ..
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:46 backup
drwxrwxr-x 2 pavani pavani 4096 Sep 24 18:20 documents
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:46 documents
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:50 scripts
pavani@UBUNTU:~/projects/linux_practice$ cd ~/projects/linux_practice/documents
pavani@UBUNTU:~/projects/linux_practice/documents$ echo "This is a practice document" > practice.txt
pavani@UBUNTU:~/projects/linux_practice/documents$ ls -l practice.txt
-rw-rw-r-- 1 pavani pavani 28 Sep 24 18:22 practice.txt
pavani@UBUNTU:~/projects/linux_practice/documents$ chmod 644 practice.txt
pavani@UBUNTU:~/projects/linux_practice/documents$ cp practice.txt ../backup/
pavani@UBUNTU:~/projects/linux_practice/documents$ cp practice.txt ../backup/practice_backup_$(date +%Y%m%d).txt
pavani@UBUNTU:~/projects/linux_practice/documents$ ls -la ../backup/
total 16
drwxrwxr-x 2 pavani pavani 4096 Sep 24 18:23 .
drwxrwxr-x 6 pavani pavani 4096 Sep 24 18:20 ..
-rw-r--r-- 1 pavani pavani  28 Sep 24 18:23 practice_backup_20250924.txt
-rw-r--r-- 1 pavani pavani  28 Sep 24 18:22 practice.txt
pavani@UBUNTU:~/projects/linux_practice/documents$ cd ~/projects/linux_practice/documents
seq 1 50 > numbers.txt
pavani@UBUNTU:~/projects/linux_practice/documents$ head numbers.txt
1
2
```

EXPERIMENT 4: SYSTEM EXPLORATION

```
pavani@UBUNTU: ~/projects/linux_practice/documents
pavani@UBUNTU:~/projects/linux_practice/documents$ uname -a
Linux UBUNTU 6.8.0-79-generic #79~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 15 16:54:53 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
pavani@UBUNTU:~/projects/linux_practice/documents$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            197M  1.6M  196M   1% /run
/dev/sda3        24G   15G   7.9G  66% /
tmpfs            985M    0  985M   0% /dev/shm
tmpfs            5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2        512M  6.1M  506M   2% /boot/efi
tmpfs            197M  124K  197M   1% /run/user/1000
pavani@UBUNTU:~/projects/linux_practice/documents$ history 10
337 seq 1 50 > numbers.txt
338 head numbers.txt
339 tail -n 5 numbers.txt
340 cat numbers.txt | grep "25"
341 nano numbers.txt
342 cat numbers.txt
top - 18:36:20 up 24 min,  1 user,  load average: 0.04, 0.45, 0.83
Tasks: 199 total,  1 running, 198 sleeping,  0 stopped,  0 zombie
%Cpu(s):  2.9 us,  5.8 sy,  0.0 ni, 82.2 id,  0.0 wa,  0.0 hi,  9.1 si,  0.0 st
MiB Mem : 1968.6 total,  116.9 free, 1271.0 used,  580.7 buff/cache
MiB Swap: 2680.0 total,  2179.3 free,  500.7 used.  637.8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 2645 pavani    20   0 4008832 182360 92532 S  16.0   9.0   2:01.88 gnome-shell
 4139 pavani    20   0  11.6g 464480 139284 S   5.3  23.0   1:37.02 firefox
    54 root       20   0      0      0      0 I   3.3   0.0   0:07.17 kworker/u2:5-events_unbound
 5268 pavani    20   0 2525460 166060 76992 S   2.3   8.2   0:18.96 Isolated Web Co
 5639 pavani    20   0  554048 52228  38980 S   2.3   2.6   0:06.04 gnome-terminal-
   342 systemd+  20   0  14836  5632  5504 S   0.3   0.3   0:01.56 systemd-oond
 2813 pavani    20   0  315096  8512  6912 S   0.3   0.4   0:01.37 ibus-daemon
 3042 pavani    20   0  662336 14068 11000 S   0.3   0.7   0:00.44 xdg-desktop-por
 4451 pavani    20   0 2514268 129488 73312 S   0.3   6.4   0:08.36 Privileged Cont
 6156 pavani    20   0   13176   4096  3328 R   0.3   0.2   0:00.56 top
    1 root       20   0  166816  9308  6748 S   0.0   0.5   0:02.00 systemd
    2 root       20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       20   0      0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
```

EXPERIMENT 5-CLEANUP

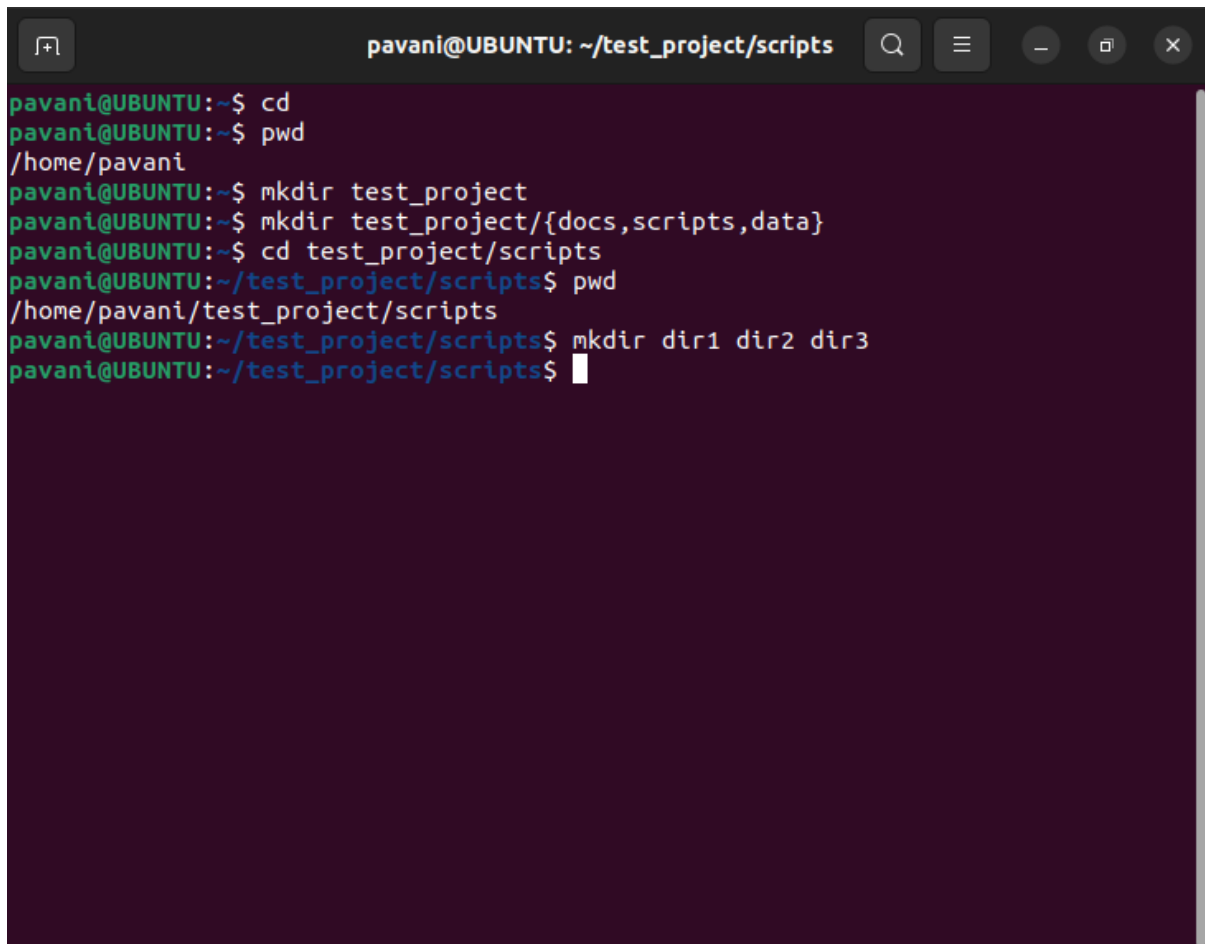
```
pavani@UBUNTU: ~/projects/linux_practice
pavani@UBUNTU: ~/projects/linux_practice$ cd ~/projects/linux_practice
pavani@UBUNTU:~/projects/linux_practice$ rm -i documents/numbers.txt
rm: remove regular file 'documents/numbers.txt'? rmdir backup
pavani@UBUNTU:~/projects/linux_practice$ rm -r backup
pavani@UBUNTU:~/projects/linux_practice$ ls -la
total 20
drwxrwxr-x 5 pavani pavani 4096 Sep 24 18:38 .
drwxrwxr-x 3 pavani pavani 4096 Sep 23 14:46 ..
drwxrwxr-x 2 pavani pavani 4096 Sep 24 18:26 documents
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:46 documents
drwxrwxr-x 2 pavani pavani 4096 Sep 23 14:50 scripts
pavani@UBUNTU:~/projects/linux_practice$ history | tail -20
306 echo "Total files: $(find . -type f | wc -l)" > summary.txt && echo "Total directories: $(find . -type d | wc -l)" >> summary.txt
307 cat summary.txt
308 clear
309 cd ..
310 cp docs/readme.txt data/project_info.txt
311 mv docs/todo.txt scripts/
312 ls data
313 ls scripts/
314 ps
315 cat ps
316 ps aux
317 ps aux | grep bash
318 kill -l
319 kill 11210
320 kill 2560
321 cd ~/projects/linux_practice
322 rm -i documents/numbers.txt
323 rm -r backup
324 ls -la
325 history | tail -20
pavani@UBUNTU:~/projects/linux_practice$
```

LAB TASKS

TASK 1- DIRECTORY NAVIGATION

Create a directory called test_project in your home directory, then create subdirectories docs, scripts, and data inside it. Navigate to the scripts directory and display your current path.

EXPLANATION-Directory navigation means moving around between different folders and files in a computer system. Every file or folder has a specific location (called a *path*), and we use navigation commands to reach them.

A terminal window titled 'pavani@UBUNTU: ~/test_project/scripts' with standard window controls. The terminal shows a series of commands and their outputs: 'cd' followed by 'pwd' returning '/home/pavani'; 'mkdir test_project' followed by 'mkdir test_project/{docs,scripts,data}' followed by 'cd test_project/scripts'; then 'pwd' returning '/home/pavani/test_project/scripts'; and finally 'mkdir dir1 dir2 dir3'. The prompt is currently at the end of the last command.

```
pavani@UBUNTU:~$ cd
pavani@UBUNTU:~$ pwd
/home/pavani
pavani@UBUNTU:~$ mkdir test_project
pavani@UBUNTU:~$ mkdir test_project/{docs,scripts,data}
pavani@UBUNTU:~$ cd test_project/scripts
pavani@UBUNTU:~/test_project/scripts$ pwd
/home/pavani/test_project/scripts
pavani@UBUNTU:~/test_project/scripts$ mkdir dir1 dir2 dir3
pavani@UBUNTU:~/test_project/scripts$
```

Conclusion:

Directory navigation helps us organize and easily find files in the system. With simple commands, we can move from one folder to another without confusion.

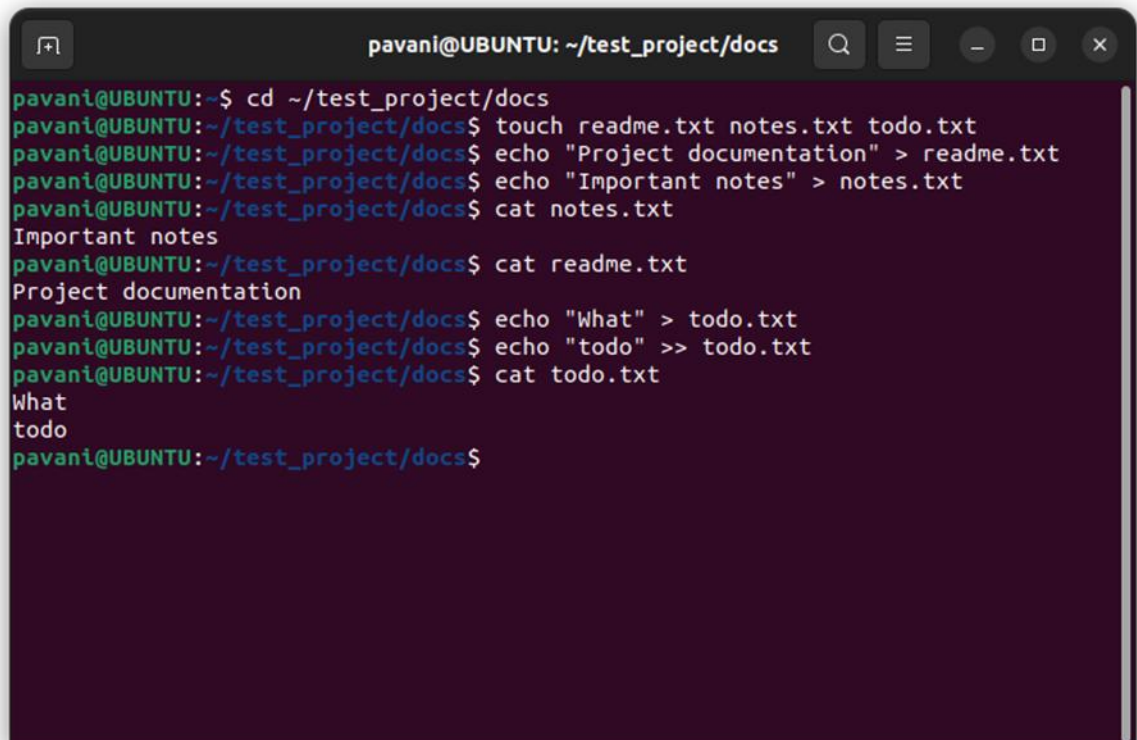
TASK 2 – FILE CREATION AND CONTENT

Create three files in the docs directory: readme.txt, notes.txt, and todo.txt. Add the text "Project documentation" to readme.txt and "Important notes" to notes.txt. Display the contents of both files.

EXPLANATION

- First, we navigated to the **docs** directory.
- Then, we created three files: readme.txt, notes.txt, and todo.txt using the touch command.
- We added the text "**Project documentation**" to readme.txt and "**Important notes**" to notes.txt using the echo command with redirection (>).

- Finally, we displayed the contents of both files using the cat command.

A terminal window titled 'pavani@UBUNTU: ~/test_project/docs' with search, menu, and window control icons. The terminal shows a series of commands and their outputs: changing to the docs directory, creating three empty files (readme.txt, notes.txt, todo.txt), writing 'Project documentation' to readme.txt, writing 'Important notes' to notes.txt, displaying the contents of notes.txt, displaying the contents of readme.txt, writing 'What' to todo.txt, appending 'todo' to todo.txt, and finally displaying the contents of todo.txt.

```
pavani@UBUNTU:~$ cd ~/test_project/docs
pavani@UBUNTU:~/test_project/docs$ touch readme.txt notes.txt todo.txt
pavani@UBUNTU:~/test_project/docs$ echo "Project documentation" > readme.txt
pavani@UBUNTU:~/test_project/docs$ echo "Important notes" > notes.txt
pavani@UBUNTU:~/test_project/docs$ cat notes.txt
Important notes
pavani@UBUNTU:~/test_project/docs$ cat readme.txt
Project documentation
pavani@UBUNTU:~/test_project/docs$ echo "What" > todo.txt
pavani@UBUNTU:~/test_project/docs$ echo "todo" >> todo.txt
pavani@UBUNTU:~/test_project/docs$ cat todo.txt
What
todo
pavani@UBUNTU:~/test_project/docs$
```

Conclusion:

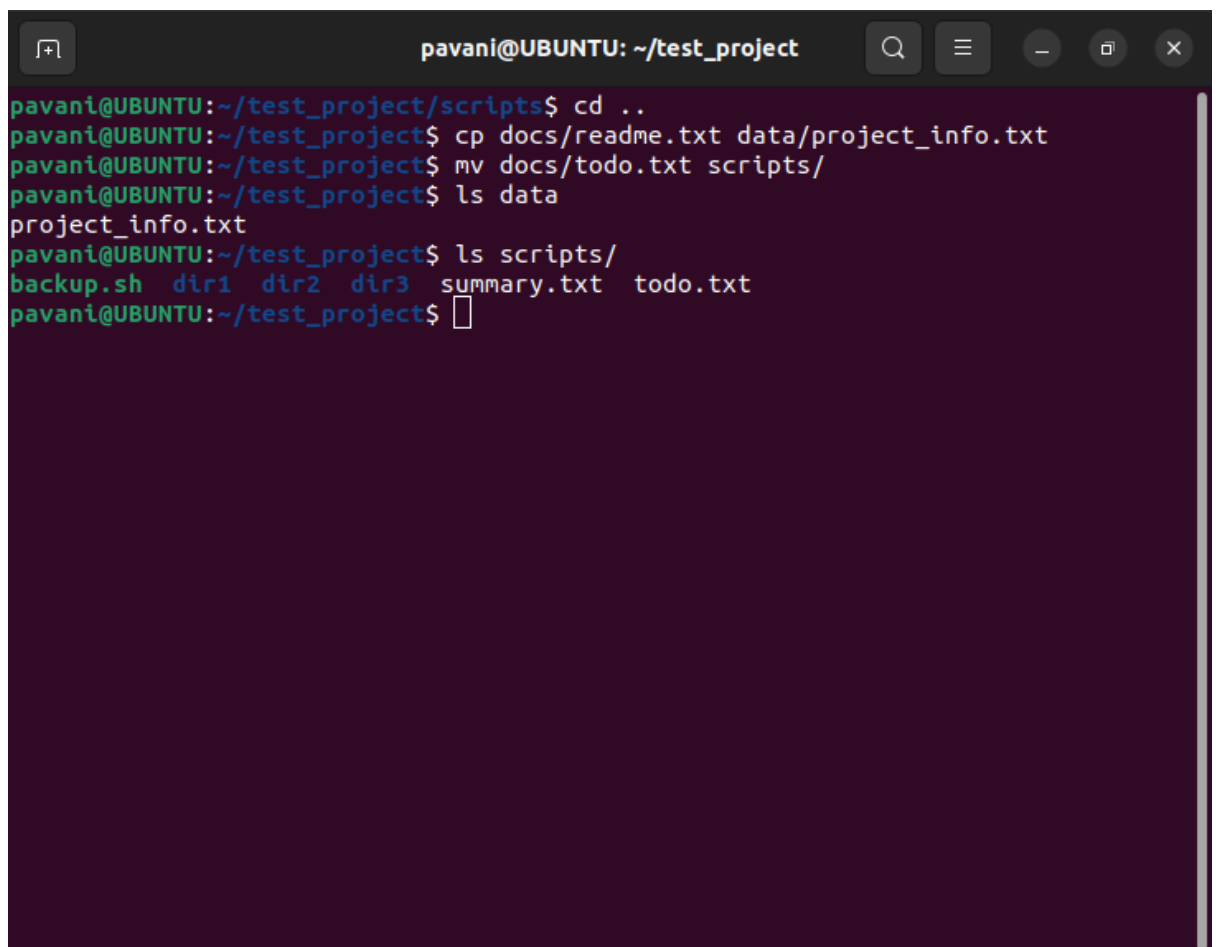
We successfully created files, added content, and viewed their contents in the terminal.

TASK 3 – FILE OPERATIONS

Copy readme.txt to the data directory and rename the copy to project_info.txt. Then move todo.txt from docs to scripts directory.

EXPLANATION

- We copied the file **readme.txt** from the *docs* directory to the *data* directory and renamed it as **project_info.txt** using the `cp` command.
- Next, we moved the file **todo.txt** from the *docs* directory to the *scripts* directory using the `mv` command.

A terminal window titled 'pavani@UBUNTU: ~/test_project' with standard Ubuntu window controls. The terminal shows a series of commands and their outputs. The user navigates to the parent directory, copies 'readme.txt' to 'data/project_info.txt', moves 'docs/todo.txt' to 'scripts/', and lists the contents of both 'data' and 'scripts' directories. The 'data' directory contains 'project_info.txt', and the 'scripts' directory contains 'backup.sh', 'dir1', 'dir2', 'dir3', 'summary.txt', and 'todo.txt'.

```
pavani@UBUNTU:~/test_project/scripts$ cd ..
pavani@UBUNTU:~/test_project$ cp docs/readme.txt data/project_info.txt
pavani@UBUNTU:~/test_project$ mv docs/todo.txt scripts/
pavani@UBUNTU:~/test_project$ ls data
project_info.txt
pavani@UBUNTU:~/test_project$ ls scripts/
backup.sh  dir1  dir2  dir3  summary.txt  todo.txt
pavani@UBUNTU:~/test_project$
```

Conclusion:

We learned how to copy a file to another directory with a new name and how to move a file from one directory to another.

TASK 4 – FILE PERMISSION

Create a shell script file called `backup.sh` in the `scripts` directory. Add the content `#!/bin/bash` and `echo "Backup complete"` to it. Make the file executable only for the owner.

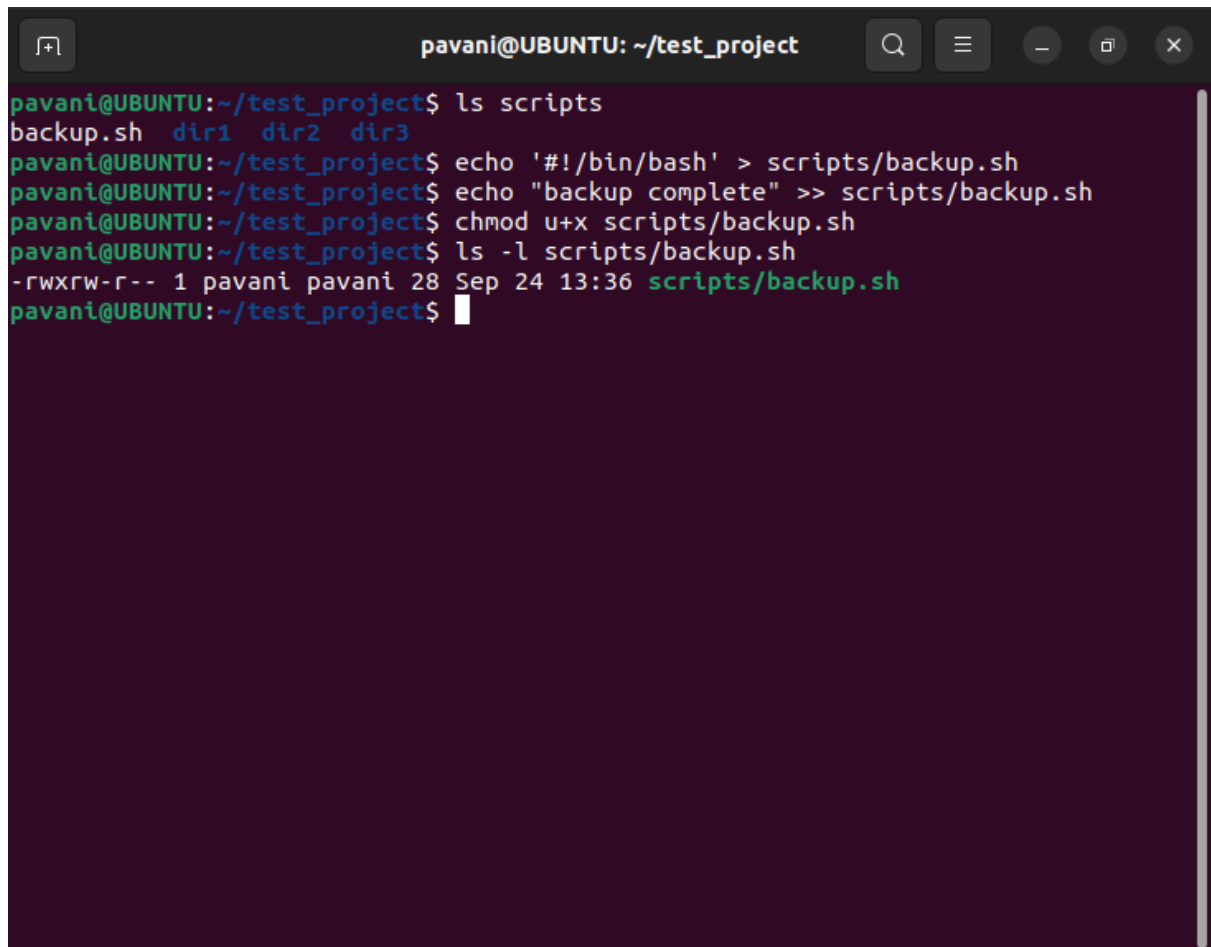
EXPLANATION

- We created a shell script file named **backup.sh** inside the *scripts* directory using the `touch` command.
- Added the content:
 - `#!/bin/bash`
 - `echo "Backup complete"`

using a text editor or redirection.

-

Then, we made the script executable **only for the owner** by setting permissions with the command

A terminal window titled 'pavani@UBUNTU: ~/test_project' with standard Ubuntu window controls. The terminal shows the following commands and output:

```
pavani@UBUNTU:~/test_project$ ls scripts
backup.sh  dir1  dir2  dir3
pavani@UBUNTU:~/test_project$ echo '#!/bin/bash' > scripts/backup.sh
pavani@UBUNTU:~/test_project$ echo "backup complete" >> scripts/backup.sh
pavani@UBUNTU:~/test_project$ chmod u+x scripts/backup.sh
pavani@UBUNTU:~/test_project$ ls -l scripts/backup.sh
-rwxrw-r-- 1 pavani pavani 28 Sep 24 13:36 scripts/backup.sh
pavani@UBUNTU:~/test_project$
```

Conclusion:

We successfully created a shell script, added commands to it, and restricted execution rights to the owner only.

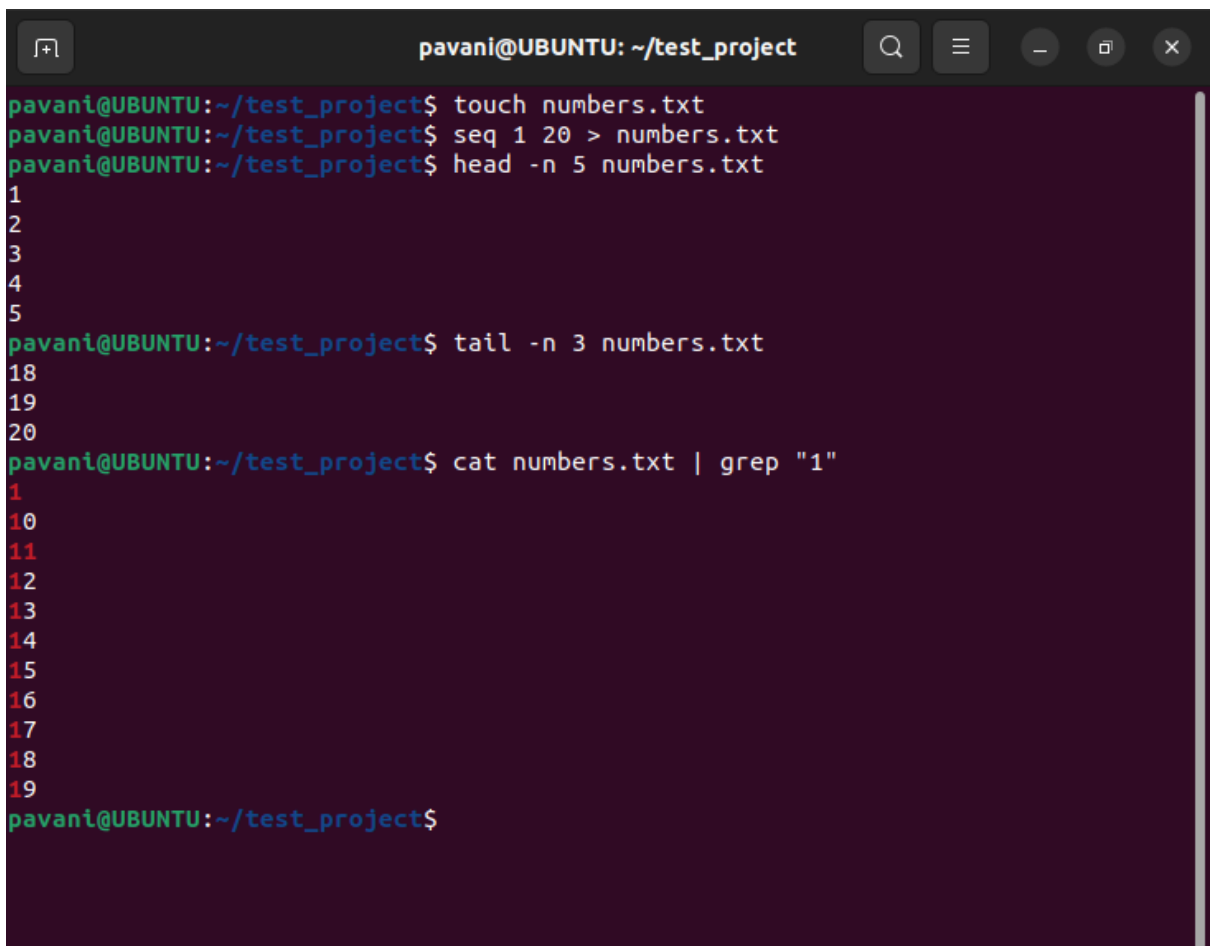
TASK 5 – FILE VIEWING

Create a file called numbers.txt with numbers 1 to 20 (each on a new line). Display only the first 5 lines, then

only the last 3 lines, then search for lines containing the number "1".

EXPLANATION

- A file named **numbers.txt** was created containing numbers from 1 to 20, each written on a new line.
- The **first 5 lines** of the file were displayed.
- The **last 3 lines** of the file were displayed.
- Then, all lines containing the number “1” were searched and shown.

A terminal window titled 'pavani@UBUNTU: ~/test_project' with search, menu, and window control icons. The terminal shows the following commands and output:

```
pavani@UBUNTU:~/test_project$ touch numbers.txt
pavani@UBUNTU:~/test_project$ seq 1 20 > numbers.txt
pavani@UBUNTU:~/test_project$ head -n 5 numbers.txt
1
2
3
4
5
pavani@UBUNTU:~/test_project$ tail -n 3 numbers.txt
18
19
20
pavani@UBUNTU:~/test_project$ cat numbers.txt | grep "1"
1
10
11
12
13
14
15
16
17
18
19
pavani@UBUNTU:~/test_project$
```

Conclusion:

This task helped us understand how to create a file with

data, view only selected parts of it, and search for specific patterns inside a file.

TASK 6 – TEXT EDITING

Using nano, create a file called config.txt with the following content:

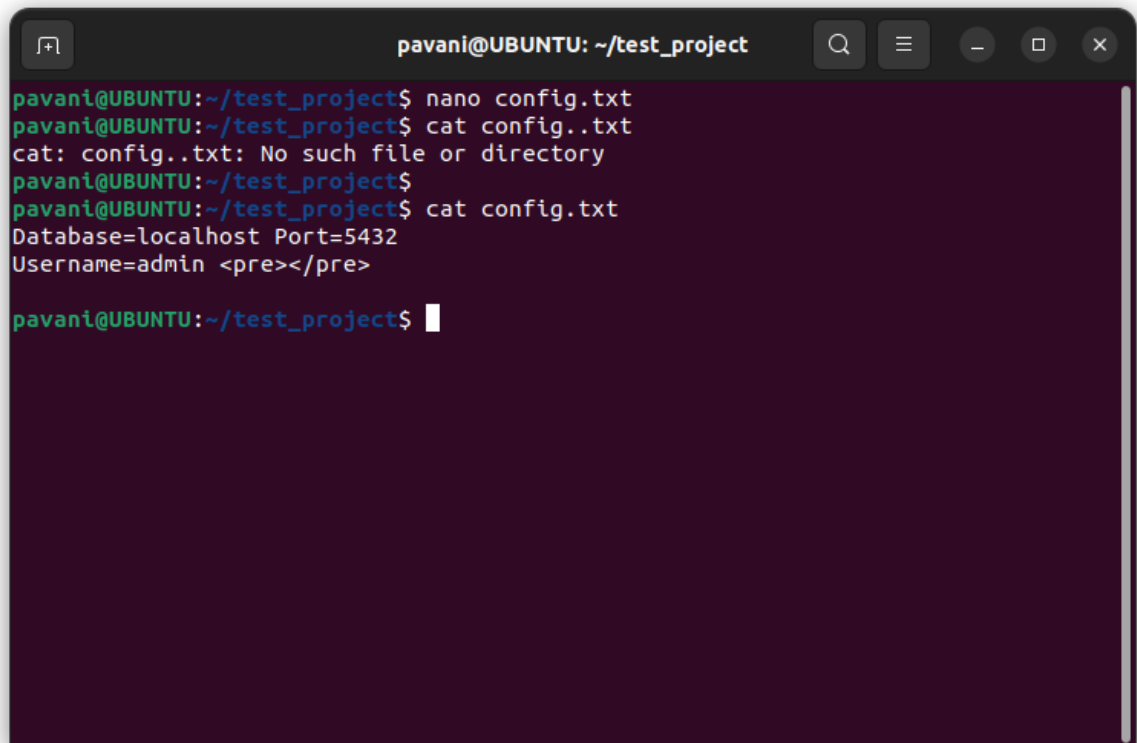
```
Database=localhost Port=5432 Username=admin
```

Save the file and then display its contents.

EXPLANATION

- A file named **config.txt** was created using the nano text editor.
- The following content was written inside the file:
 - Database = localhost
 - Port = 5432
 - Username = admin

- The file was saved and then its contents were displayed.

A terminal window titled 'pavani@UBUNTU: ~/test_project' with standard Ubuntu window controls. The terminal shows a sequence of commands: 'nano config.txt' to create a file, 'cat config..txt' which results in an error 'cat: config..txt: No such file or directory', and 'cat config.txt' which displays the file's contents: 'Database=localhost Port=5432' and 'Username=admin' on separate lines. The prompt returns to 'pavani@UBUNTU:~/test_project\$' with a cursor.

```
pavani@UBUNTU:~/test_project$ nano config.txt
pavani@UBUNTU:~/test_project$ cat config..txt
cat: config..txt: No such file or directory
pavani@UBUNTU:~/test_project$
pavani@UBUNTU:~/test_project$ cat config.txt
Database=localhost Port=5432
Username=admin <pre></pre>
pavani@UBUNTU:~/test_project$
```

Conclusion:

We learned how to create a text file using a text editor, add configuration details to it, save the file, and view its contents.

Task 7: System Information

Create a file called `system_info.txt` that contains: your username, current date, your current directory, and disk usage information in human-readable format.

EXPLANATION

- A file named **system_info.txt** was created.

- The file contains the following information:
 - The current username.
 - The current date.
 - The present working directory.
 - Disk usage details in a human-readable format.
- All this information was saved inside the file and then viewed.

```
pavani@UBUNTU: ~/test_project
pavani@UBUNTU:~/test_project$ touch system_info.txt
pavani@UBUNTU:~/test_project$ echo "Username: " >>
bash: syntax error near unexpected token `newline'
pavani@UBUNTU:~/test_project$ echo "Username: " >> system_info.txt
pavani@UBUNTU:~/test_project$ whoami >> system_info.txt
pavani@UBUNTU:~/test_project$ echo "current Date: " >> system_info.txt
pavani@UBUNTU:~/test_project$ date >> system_info.txt
pavani@UBUNTU:~/test_project$ echo "Current Directory: " >> system_info.txt
pavani@UBUNTU:~/test_project$ pwd >> system_info.txt
pavani@UBUNTU:~/test_project$ echo "Disk usage: " >> system_info.txt
pavani@UBUNTU:~/test_project$ df -h >> system_info.txt
pavani@UBUNTU:~/test_project$ cat system_info.txt
Username:
pavani
Current Date:
Current Date:
Wednesday 24 September 2025 02:26:21 PM IST
Current Directory:
Disk Usage:
Username:
pavani
current Date:
Wednesday 24 September 2025 02:43:27 PM IST
Current Directory:
/home/pavani/test_project
Disk usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            197M  1.6M  196M   1% /run
/dev/sda3        24G   15G   8.1G  65% /
```



```
pavani
Current Date:
Current Date:
Wednesday 24 September 2025 02:26:21 PM IST
Current Directory:
Disk Usage:
Username:
pavani
current Date:
Wednesday 24 September 2025 02:43:27 PM IST
Current Directory:
/home/pavani/test_project
Disk usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            197M  1.6M  196M   1% /run
/dev/sda3        24G   15G   8.1G  65% /
tmpfs            985M    0   985M   0% /dev/shm
tmpfs            5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2        512M  6.1M  506M   2% /boot/efi
tmpfs            197M  128K  197M   1% /run/user/1000
pavani@UBUNTU:~/test_project$
```

Conclusion:

We learned how to gather system-related details and store them in a text file for reference.

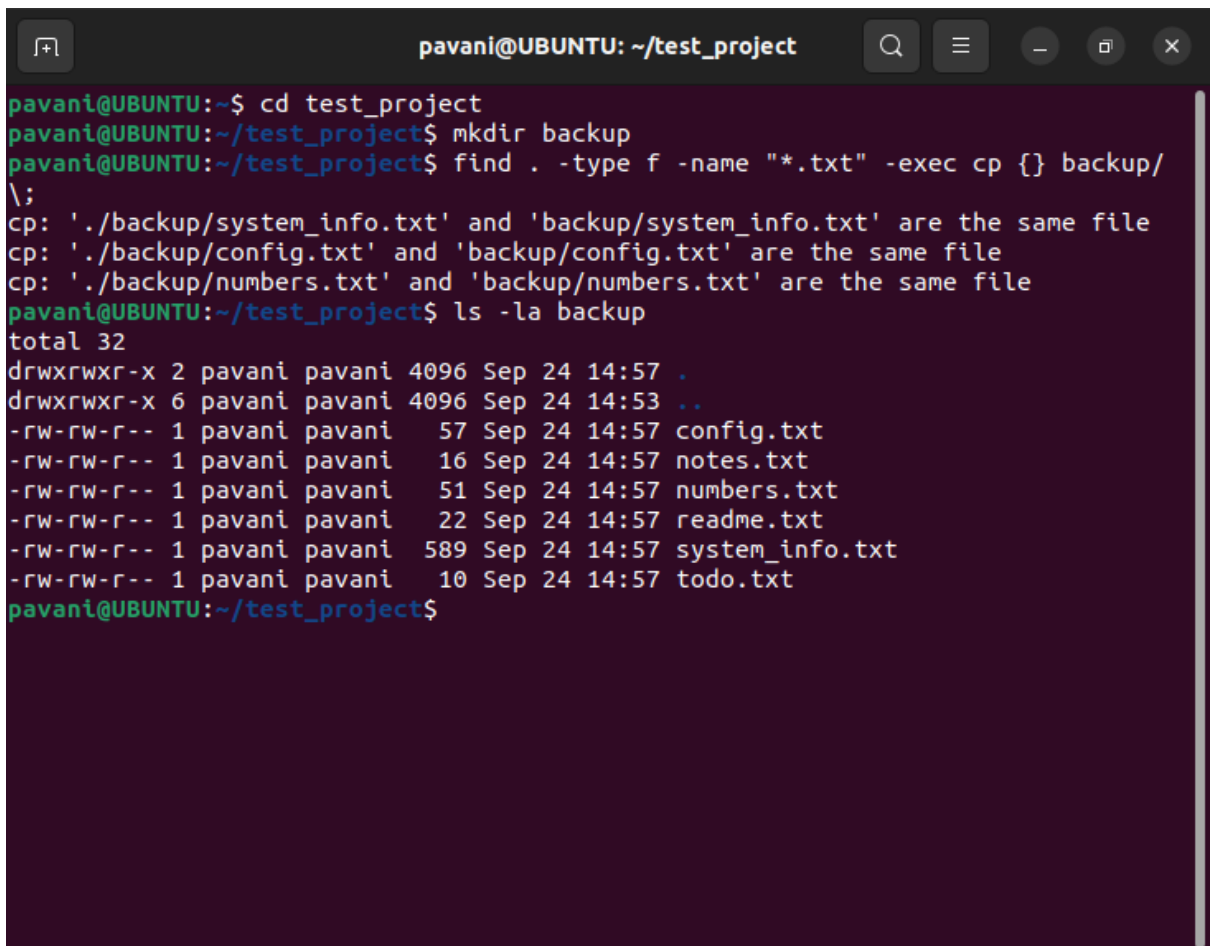
Task 8: File Organization

In your test_project directory, create a backup folder. Copy all .txt files from all subdirectories into this backup folder.

Then list all files in the backup folder with detailed information.

EXPLANATION

- In the **test_project** directory, a folder named **backup** was created.
- All text files (with the extension .txt) from every subdirectory were copied into the backup folder.
- After copying, the contents of the backup folder were listed with detailed information such as file size, permissions, and modification date.

A terminal window titled 'pavani@UBUNTU: ~/test_project' with search, menu, and window control icons. The terminal shows the following commands and output:

```
pavani@UBUNTU:~$ cd test_project
pavani@UBUNTU:~/test_project$ mkdir backup
pavani@UBUNTU:~/test_project$ find . -type f -name "*.txt" -exec cp {} backup/ \;
cp: './backup/system_info.txt' and 'backup/system_info.txt' are the same file
cp: './backup/config.txt' and 'backup/config.txt' are the same file
cp: './backup/numbers.txt' and 'backup/numbers.txt' are the same file
pavani@UBUNTU:~/test_project$ ls -la backup
total 32
drwxrwxr-x 2 pavani pavani 4096 Sep 24 14:57 .
drwxrwxr-x 6 pavani pavani 4096 Sep 24 14:53 ..
-rw-rw-r-- 1 pavani pavani  57 Sep 24 14:57 config.txt
-rw-rw-r-- 1 pavani pavani  16 Sep 24 14:57 notes.txt
-rw-rw-r-- 1 pavani pavani  51 Sep 24 14:57 numbers.txt
-rw-rw-r-- 1 pavani pavani  22 Sep 24 14:57 readme.txt
-rw-rw-r-- 1 pavani pavani 589 Sep 24 14:57 system_info.txt
-rw-rw-r-- 1 pavani pavani  10 Sep 24 14:57 todo.txt
pavani@UBUNTU:~/test_project$
```

Conclusion:

We learned how to organize files by collecting them from different locations into a single folder and how to view detailed file information.

Task 9: Process and History

Display your command history and count how many commands you've executed. Then show the top 10 most recent commands.

EXPLANATION

- The full command history of the session was displayed.
- The total number of commands executed so far was counted.
- Finally, the most recent 10 commands were shown from the history.

```
pavani@UBUNTU: ~  
pavani@UBUNTU:~$ history  
1 sudo apt update  
2 su -  
3 sudo apt update  
4 su -  
5 sudo apt update  
6 sudo passwd -1 root  
7 su -  
8 reboot  
9 sudo apt update  
10 sudo apt upgrade  
11 sudo apt install gedit =y  
12 sudo apt install gedit -y  
13 cd ~  
14 cd Desktop  
15 pwd  
16 cd ~/Desktop  
17 gedit hello.c  
18 ls  
19 gcc hello.c -o hello  
20 gcc Hello.c -o hello  
21 gcc hello.c -o hello  
22 gedit hello.c  
23 gcc hello.c -o hello  
24 ls  
25 ./hello  
26 cd cd file.txt  
27 cd file.txt  
28 cd Desktop
```

```
pavani@UBUNTU: ~  
266 clear  
267 touch system_info.txt  
268 echo "Username: " >>  
269 echo "Username: " >> system_info.txt  
270 whoami >> system_info.txt  
271 echo "current Date: " >> system_info.txt  
272 date >> system_info.txt  
273 echo "Current Directory: " >> system_info.txt  
274 pwd >> system_info.txt  
275 echo "Disk usage: " >> system_info.txt  
276 df -h >> system_info.txt  
277 cat system_info.txt  
278 cd test_project  
279 history  
pavani@UBUNTU:~$ history | wc -l  
280  
pavani@UBUNTU:~$ history 10  
272 date >> system_info.txt  
273 echo "Current Directory: " >> system_info.txt  
274 pwd >> system_info.txt  
275 echo "Disk usage: " >> system_info.txt  
276 df -h >> system_info.txt  
277 cat system_info.txt  
278 cd test_project  
279 history  
280 history | wc -l  
281 history 10  
pavani@UBUNTU:~$ wc -  
█
```

Conclusion:

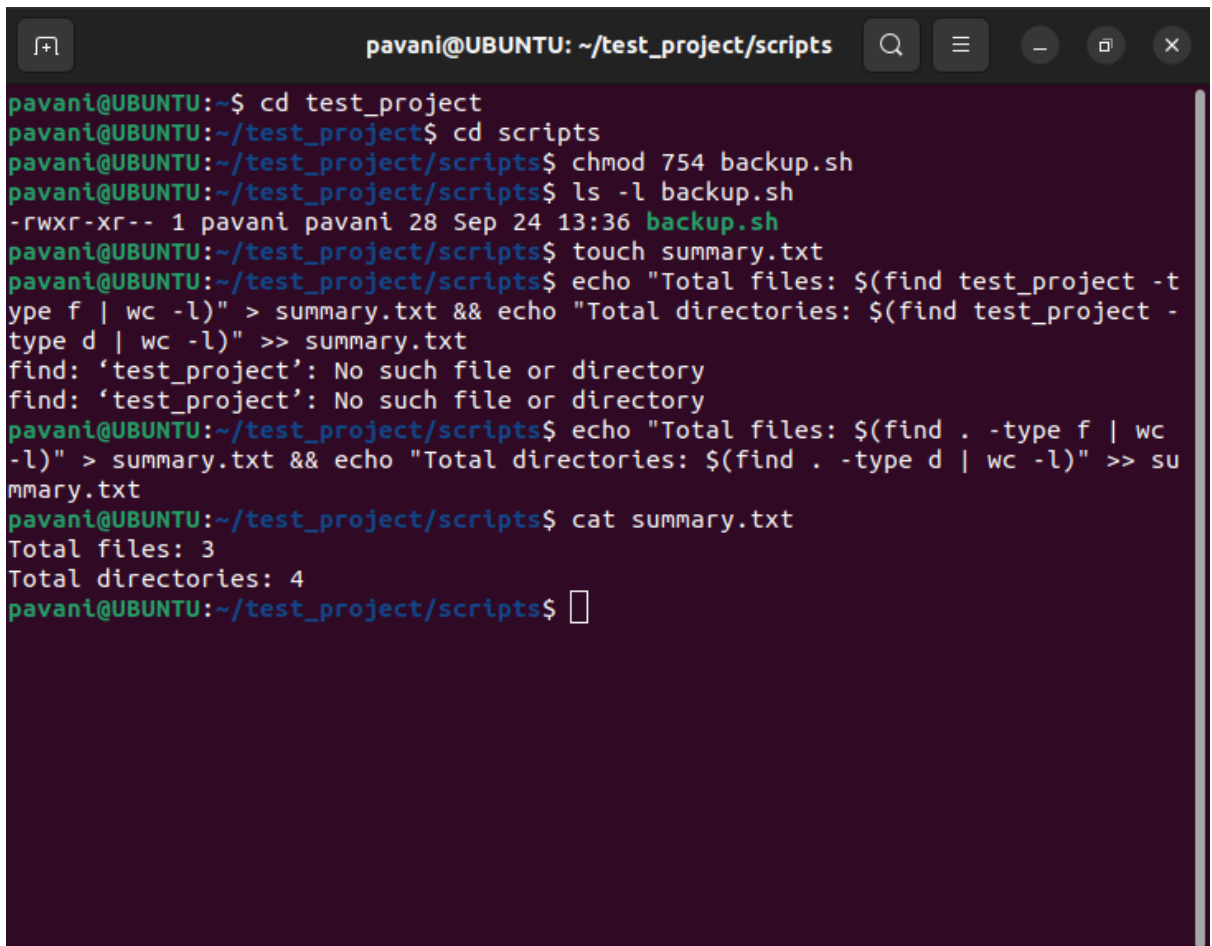
This task helped us understand how to view the command history, count executed commands, and check the latest commands used in the terminal.

Task 10: Comprehensive Cleanup

Set the permissions of your backup.sh script to be readable, writable, and executable by owner, readable and executable by group, and readable by others. Then create a summary file that lists the total number of files and directories in your entire test_project.

EXPLANATION

- The script **backup.sh** was updated so that the **owner** has read, write, and execute permissions, the **group** has read and execute permissions, and **others** have read permission.
- A **summary file** was created that lists the total number of files and directories in the entire **test_project** directory.

A terminal window titled 'pavani@UBUNTU: ~/test_project/scripts' showing a series of commands and their outputs. The user navigates to the 'test_project' directory and then to the 'scripts' subdirectory. They set permissions of 754 on 'backup.sh' and list its details. Then, they create a 'summary.txt' file using 'touch'. Finally, they use 'echo' and 'find' commands to populate the summary file with the total number of files and directories. The final output shows 'Total files: 3' and 'Total directories: 4'.

```
pavani@UBUNTU:~$ cd test_project
pavani@UBUNTU:~/test_project$ cd scripts
pavani@UBUNTU:~/test_project/scripts$ chmod 754 backup.sh
pavani@UBUNTU:~/test_project/scripts$ ls -l backup.sh
-rwxr-xr-- 1 pavani pavani 28 Sep 24 13:36 backup.sh
pavani@UBUNTU:~/test_project/scripts$ touch summary.txt
pavani@UBUNTU:~/test_project/scripts$ echo "Total files: $(find test_project -t
ype f | wc -l)" > summary.txt && echo "Total directories: $(find test_project -
type d | wc -l)" >> summary.txt
find: 'test_project': No such file or directory
find: 'test_project': No such file or directory
pavani@UBUNTU:~/test_project/scripts$ echo "Total files: $(find . -type f | wc
-l)" > summary.txt && echo "Total directories: $(find . -type d | wc -l)" >> su
mmmary.txt
pavani@UBUNTU:~/test_project/scripts$ cat summary.txt
Total files: 3
Total directories: 4
pavani@UBUNTU:~/test_project/scripts$
```

Conclusion:

This task demonstrated how to manage file permissions for different users and how to create a summary report of a project directory's structure.

Quick Verification Commands

After each task, verify your work:

Check current location

```
pwd
```

List files with details

```
ls -la
```

Check file contents

```
cat filename.txt
```

Check permissions

```
ls -l filename
```

Check directory structure

```
ls -R
```

Check command history

```
history 5
```

Common Commands You Should Know After This

Task	Key Commands Used
1	mkdir, cd, pwd
2	touch, echo >, cat
3	cp, mv
4	chmod, ls -l
5	head, tail, grep
6	nano
7	whoami, date, df -h
8	find, wildcard *
9	history, wc -l
10	chmod with numbers, find with type

Observation

Every file and directory in Linux has three types of permissions: read (r), write (w), and execute (x).

These permissions are assigned to user (owner), group, and others. Commands like chmod and chown help in modifying permissions and ownership. The terminal shows permissions in a 10-character string (e.g., -rwxr-xr--).

Conclusion

From this experiment, we conclude that Linux provides a powerful permission system to control access to files and directories. By using essential commands like `ls`, `pwd`, `mkdir`, `touch`, `chmod`, and `chown`, we can create, manage, and secure files. Understanding file system permissions is important for system security and effective usage of Linux.