

[590023785]Exp2_Script\[590023785]Exp2_Script.md

Experiment 2: Linux File Systems, Permissions, and Essential Commands

Name: Gayatri Bhatt **Roll No.:** 590023785 **Date:** 2025-09-23

Aim:

- To gain an understanding of Linux file system layout, file permission settings, and crucial commands.
- To learn how to manage file access rights by using commands such as `ls`, `chmod`, and `umask`.

Requirements

- A Linux system (Ubuntu, Fedora, or equivalent) with a bash shell environment.
- Basic user-level permissions to create, modify, and delete files and directories.

Theory

Linux arranges files and directories into a hierarchical tree starting at the root directory `/`. Each file and directory has specific permissions that restrict or allow read, write, or execute access. These permissions are classified for the **owner**, **group**, and **others**. Commands like `ls`, `pwd`, `cd`, `mkdir`, `chmod`, and `umask` help users navigate the file system and modify permissions.

Procedure & Observations

Exercise 1: Explore Linux File System

Task Statement:

Browse through the Linux directory structure and list key directories such as `/bin`, `/etc`, `/home`, `/var`, and `/tmp`, understanding their roles.

Explanation:

The `ls` command is used to list the contents of the root and other directories. Each directory serves a specific function; for instance, `/bin` contains executable programs, `/etc` holds configuration files, `/home` stores user files, `/var` keeps log files, and `/tmp` is used for temporary storage.

Command(s):

```
ls /  
ls /bin  
ls /etc  
ls /home  
ls /var  
ls /tmp
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ vim lexp1.sh
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ bash lexp1.sh
bin                lib                proc              sys
bin.usr-is-merged  lib.usr-is-merged root              tmp
boot              lib64              run               usr
dev               lost+found         sbin             var
etc               media              sbin.usr-is-merged
home             mnt                snap
init             opt                srv
NF
X11
['
aa-enabled
aa-exec
aa-features-abi
add-apt-repository
addpart
addr2line
apport-bug
apport-cli
apport-collect
apport-unpack
appstreamcli
apropos
apt
apt-add-repository
apt-cache
apt-cdrom
apt-config
apt-extracttemplates
apt-ftparchive
apt-get
apt-key
apt-mark
apt-sortpkgs
ar
```

Exercise 2: Understanding File Permissions

Task Statement:

Inspect and comprehend the permissions assigned to files using the `ls -l` command.

Explanation:

The `ls -l` command displays detailed information about files, including the permission string, ownership, group, size, and modification timestamps. Permissions appear as a sequence of 10 characters where the first shows the file type and the following nine represent read (r), write (w), and execute (x) permissions divided among owner, group, and others.

Command(s):

```
ls -l
```

Output:

```

gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ vim 1exp2.sh
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ bash 1exp2.sh
total 384
drwxrwxrwx 1 gayatri10 gayatri10 4096 Sep 22 19:38 100codes
-rwxrwxrwx 1 gayatri10 gayatri10 76 Sep 24 06:34 1ex1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 34 Sep 24 06:35 1ex2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 55 Sep 24 06:35 1ex3.sh
-rwxrwxrwx 1 gayatri10 gayatri10 33 Sep 24 06:38 1ex4.sh
-rwxrwxrwx 1 gayatri10 gayatri10 81 Sep 24 06:39 1ex5.sh
-rwxrwxrwx 1 gayatri10 gayatri10 57 Sep 24 06:40 1ex6.sh
-rwxrwxrwx 1 gayatri10 gayatri10 58 Sep 24 12:54 1exp1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 18 Sep 24 12:55 1exp2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 103 Sep 24 12:15 3ex1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 117 Sep 24 12:15 3ex2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 33 Sep 24 12:04 4ex1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 126 Sep 24 12:05 4ex2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 235 Sep 24 12:06 4ex3.sh
-rwxrwxrwx 1 gayatri10 gayatri10 163 Sep 24 12:06 4ex4.sh
-rwxrwxrwx 1 gayatri10 gayatri10 61 Sep 23 17:58 6ex1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 79 Sep 23 18:34 6ex2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 55 Sep 23 18:35 6ex3.sh
-rwxrwxrwx 1 gayatri10 gayatri10 277 Sep 23 18:40 6ex4.sh
-rwxrwxrwx 1 gayatri10 gayatri10 88 Sep 23 18:46 6ex5.sh
-rwxrwxrwx 1 gayatri10 gayatri10 201 Sep 23 18:47 6ex6.sh
-rwxrwxrwx 1 gayatri10 gayatri10 97 Sep 23 18:49 6ex7.sh
-rwxrwxrwx 1 gayatri10 gayatri10 151 Sep 24 07:07 8ex1.sh
-rwxrwxrwx 1 gayatri10 gayatri10 229 Sep 24 07:08 8ex2.sh
-rwxrwxrwx 1 gayatri10 gayatri10 146 Sep 24 08:15 8ex3.sh
-rwxrwxrwx 1 gayatri10 gayatri10 251 Sep 24 08:16 8ex4.sh
drwxrwxrwx 1 gayatri10 gayatri10 4096 Sep 24 12:44 Experiments
-rwxrwxrwx 1 gayatri10 gayatri10 131 Aug 27 12:51 HELLO.C
-rwxrwxrwx 1 gayatri10 gayatri10 41871 Aug 27 12:49 HELLO.exe
-rwxrwxrwx 1 gayatri10 gayatri10 225 Sep 6 05:26 Untitled-2.c
-rwxrwxrwx 1 gayatri10 gayatri10 40851 Sep 6 05:26 Untitled-2.exe
-rwxrwxrwx 1 gayatri10 gayatri10 331 Sep 6 05:35 Untitled-3.c
-rwxrwxrwx 1 gayatri10 gayatri10 40851 Aug 27 12:54 a.exe
-rwxrwxrwx 1 gayatri10 gayatri10 258 Sep 6 05:38 areaofcircleusingpi.c
-rwxrwxrwx 1 gayatri10 gayatri10 319 Sep 23 17:29 armstrong.sh
-rwxrwxrwx 1 gayatri10 gayatri10 0 Jan 15 2024 dated_file.txt

```

Exercise 3: Changing Permissions with chmod

Task Statement:

Modify file permissions using the chmod command.

Explanation:

The chmod command changes file access rights. Permissions can be set symbolically (like u+x for adding execute permission to the user) or numerically (e.g., 755). For example, chmod 755 file1 grants read, write, and execute rights to the file's owner, and read and execute rights to group and others.

Command(s):

```

chmod 755 file1
ls -l file1
chmod u-w file1
ls -l file1

```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ vim 1exp3.sh
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ bash 1exp3.sh
chmod: cannot access 'file1': No such file or directory
ls: cannot access 'file1': No such file or directory
chmod: cannot access 'file1': No such file or directory
ls: cannot access 'file1': No such file or directory
```

Exercise 4: Default Permissions with umask

Task Statement:

Learn about default permissions using the umask command.

Explanation:

The umask sets which permission bits are taken away when new files or directories are created. For instance, a umask of 022 results in new files with default permissions of 644 and directories with 755.

Command(s):

```
umask
umask 022
touch newfile
ls -l newfile
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ vim 1exp4.sh
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ bash 1exp4.sh
0022
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1$ |
```

Result

- Successfully navigated and investigated the Linux file system structure.
- Viewed and modified file permissions using `ls -l`, `chmod`, and `umask`.
- Acquired knowledge about Linux access control mechanisms.

Challenges Faced & Learning Outcomes

- Difficulty distinguishing symbolic from numeric modes in `chmod` resolved by repeated exercises.
- Clarified how `umask` impacts permission calculation by reviewing the masking process.

Learning:

- Gained proficiency in traversing Linux file system hierarchy.
- Understood the significance of file permissions.
- Became adept at permission management via `chmod` and `umask`.

Conclusion

This experiment highlighted Linux file system organization, permission controls, and basic commands. It emphasized how access rights are managed and provided hands-on experience essential for Linux system administration.

TASK 1: [Directory Navigation]

Task Statement:

- Create a directory named `test_project` inside your home directory, then create subdirectories `docs`, `scripts`, and `data` within it. Navigate into the `scripts` directory and display the current directory path.

Explanation:

- The `mkdir` command is used to create directories. Use `cd` to change directories and `pwd` to print the current directory path.

Command(s):

```
mkdir test_project
cd test_project
mkdir docs scripts data
cd scripts
pwd
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs$ mkdir test_project
mkdir: cannot create directory 'test_project': File exists
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs$ cd test_project
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project$ mkdir docs scripts data
mkdir: cannot create directory 'docs': File exists
mkdir: cannot create directory 'scripts': File exists
mkdir: cannot create directory 'data': File exists
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project$ cd scripts
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ pwd
/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/t
est_project/scripts
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ |
```

TASK 2: [File Creation and Content]

Task Statement:

- Create three files named `readme.txt`, `notes.txt`, and `todo.txt` in the `docs` directory. Write "Project documentation" to `readme.txt` and "Important notes" to `notes.txt`. Display the contents of both files.

Explanation:

- Use `touch` to create empty files. Use `echo "text" > file` to write text to files, and `cat` to display file contents.

Command(s):

```
cd docs
touch readme.txt notes.txt todo.txt
echo "Project documentation" > readme.txt
echo "Important notes" > notes.txt
cat notes.txt
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ cd docs
bash: cd: docs: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ touch readme.txt notes.txt to
do.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ echo "Project documentation"
> readme.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ echo "Important notes" > note
s.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ cat notes.txt
Important notes
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe
riments/Experiments Pdfs/test_project/scripts$ |
```

TASK 3: [File Operations]

Task Statement:

- Copy the file `readme.txt` into the `data` directory renaming it to `project_info.txt`. Move `todo.txt` from `docs` to `scripts`.

Explanation:

- Use `cp source destination` to copy files. Use `mv oldname newname` to rename files or move files to other directories.

Command(s):

```
cp readme.txt data/project_info.txt
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ cp readme.txt data/project_info.txt
cp: cannot create regular file 'data/project_info.txt': No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ |
```

TASK 4: [File Permissions]

Task Statement:

- Create a shell script file called backup.sh in scripts. Insert the lines `#!/bin/bash` and `echo "Backup complete"`. Make the script executable only by the owner.

Explanation:

- `chmod u+x filename` grants execute permission to the file owner. Use `ls -l` to verify permissions.

Command(s):

```
cd scripts
touch backup.sh > echo "Backup complete"
chmod u+x backup.sh
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ cd scripts
bash: cd: scripts: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ touch backup.sh > echo "Backup complete"
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ chmod u+x backup.sh
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Experiments/Experiments Pdfs/test_project/scripts$ |
```

TASK 5: [File Viewing]

Task Statement:

- Create a file called numbers.txt with numbers 1 to 20 each on a new line. Display the first 5 lines, then the last 3 lines, and finally search for lines containing the number "1."

Explanation:

- `seq 1 20 > numbers.txt` generates the list. Use `head -n 5`, `tail -n 3`, and `grep "1"` to view sections and search.

Command(s):

```
seq 1 20 > numbers.txt  
head -n 5  
tail -n 3  
grep "1"
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe  
riments/Experiments Pdfs/test_project/scripts$ seq 1 20 > numbers.txt  
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe  
riments/Experiments Pdfs/test_project/scripts$ head -n 5  
1 2 3 4 5  
1 2 3 4 5  
^C  
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe  
riments/Experiments Pdfs/test_project/scripts$ tail -n 3  
5 4 3 2  
^C  
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/Expe  
riments/Experiments Pdfs/test_project/scripts$ grep "1"
```

TASK 6: [Text Editing]

Task Statement:

- Using nano, create a file named config.txt containing:

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

Save and display the contents of the file.

Explanation:

- Open Nano by running `nano filename.txt`. Type the content, save with `Ctrl+O`, exit with `Ctrl+X`. View contents with `cat config.txt`.

Command(s):

```
vim config.txt  
cat config.txt
```

Alternatively

```
nano config.txt  
cat config.txt
```


Output:

```

" =====
" Netrw Directory Listing                                     (netrw v173
"   /mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1
"   Sorted by      name
"   Sort sequence: [\/]$, \<core\%(\\.d\+\\)\=\>, \.h$, \.c$, \.cpp$, \~\=\*$, *, \.
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:spe
" =====
./
C SEM - 1/
| .git/
| .vscode/
| 100codes/
| C Programming Experiments/
| Experiments/
| | Experiments Pdfs/
| | | file.txt/
| | | test_project/
| | | | data/
| | | | docs/
| | | | scripts/
| | | | | config.txt/
| | | | | test_project/
| | | | | 2ex1.sh*
| | | | | Backup complete*
| | | | | backup.sh*
| | | | | echo*
| | | | | notes.txt*
| | | | | numbers.txt*
| | | | | readme.txt*
| | | | | todo.txt*
| | | 2ex1.sh*
| | | 2exp10.sh*
| | | 2t1.sh*
| | | 6t1.sh*
| | | 6t2.sh*
| | | 6t3.sh*
| | | 6t4.sh*
| | | 7t1.sh*
| | | 7t2.sh*
| | | 7t3.sh*

```

TASK 7: [System Information]**Task Statement:**

- Create a file named system_info.txt containing your username, current date, current directory, and disk usage in human-readable format.

Explanation:

- Use whoami, date, pwd, and df -h. Redirect outputs to the file using >>. Use echo to label each section.

Command(s):

```
cd scripts
touch system_info.txt
echo "Username:" >> system_info.txt
whoami >> system_info.txt
echo "Date:" >> system_info.txt
date >> system_info.txt
echo "Current Directory:" >> system_info.txt
pwd >> system_info.txt
echo "Disk Usage:" >> system_info.txt
df -h >> system_info.txt
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ cd scripts
-bash: cd: scripts: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ touch system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Username:" >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ whoami >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Date:" >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ date >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Current Directory:" >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ pwd >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Disk Usage:" >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ df -h >> system_info.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$
```

TASK 8: [File Organisation]

Task Statement:

- Inside test_project create a backup folder. Copy all .txt files from all subdirectories into this folder. Then list all files with detailed information.

Explanation:

- Use wildcards like *.txt to select text files. Copy multiple files using `cp file1 file2 destination/`. Use `ls -la` to display detailed file information.

Command(s):

```
cp test_project/data/project_info.txt    test_project/docs/notes.txt
test_project/docs/readme.txt    test_project/docs/todo.txt
test_project/scripts/config.txt    test_project/scripts/numbers.txt
test_project/scripts/system_info.txt    test_project/scripts/todo.txt    backup/
```

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ cp test_project/data/project_info.txt
cp: missing destination file operand after 'test_project/data/project_info.t
xt'
Try 'cp --help' for more information.
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/docs/notes.txt
-bash: test_project/docs/notes.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/docs/readme.txt
-bash: test_project/docs/readme.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/docs/todo.txt
-bash: test_project/docs/todo.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/scripts/config.txt
-bash: test_project/scripts/config.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/scripts/numbers.txt
-bash: test_project/scripts/numbers.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/scripts/system_info.txt
-bash: test_project/scripts/system_info.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ test_project/scripts/todo.txt
-bash: test_project/scripts/todo.txt: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ backup/
-bash: backup/: No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$
```

TASK 9: [Process and History]

Task Statement:

- Display your command history and count how many commands have been entered. Show the last 10 commands.

Explanation:

- Use history to display all commands. Count total commands with history | wc -l. Use history 10 or history | tail -10 to show recent commands.

Command(s):

history 10

Output:

```
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr  
ogramming Experiments$ history 10  
468 test_project/docs/notes.txt  
469 test_project/docs/readme.txt  
470 test_project/docs/todo.txt  
471 test_project/scripts/config.txt  
472 test_project/scripts/numbers.txt  
473 test_project/scripts/system_info.txt  
474 test_project/scripts/todo.txt  
475 backup/  
476 clear  
477 history 10  
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr  
ogramming Experiments$
```

TASK 10: [Comprehensive Cleanup]

Task Statement:

- Set permissions on backup.sh to be readable, writable, and executable by owner; readable and executable by group; and readable by others. Create a summary file listing total files and directories in test_project.

Explanation:

- Use `chmod 754 backup.sh` to assign permissions. Count files with `find . -type f | wc -l` and directories with `find . -type d | wc -l`. Save counts to a file using echo and output redirection.

Command(s):

```
chmod 754 backup.sh
```

```
echo "Total files:" > summary.txt  
find . -type f | wc -l >> summary.txt  
echo "Total directories:" >> summary.txt  
find . -type d | wc -l >> summary.txt
```

Output

```
477 history 10
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ chmod 754 backup.sh
chmod: cannot access 'backup.sh': No such file or directory
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Total files:" > summary.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ find . -type f | wc -l >> summary.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ echo "Total directories:" >> summary.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$ find . -type d | wc -l >> summary.txt
gayatri10@LAPTOP-JAMRNMRP:/mnt/c/Users/GAYA1/OneDrive/Desktop/C SEM - 1/C Pr
ogramming Experiments$
```

Result

- Successfully created, copied, moved, and removed files.
- Learned to view file contents and analyze logs.
- Explored file permissions and ownership management.
- Employed `find` and `grep` for searching files and content.
- Worked with file archives and compression.
- Practiced using both hard and symbolic links.

Challenges Faced & Learning Outcomes

- Accidentally deleted files using `rm` without confirmation; learned to prefer `rm -i`.
- Improved understanding of numeric versus symbolic permission notation in `chmod` through practice.

Learning:

- Developed hands-on skills in file operations and permissions.
- Learned efficient search and pattern matching in Linux.
- Gained insight into archiving and linking methods.

Conclusion

The experiment provided practical exposure to Linux file system navigation, permission handling, searching, archiving, and linking. These skills are fundamental for effective Linux use and administration.