

Experiment [2]: [Linux file systems permissions and essential commands]

Name: Aarush Prasad Roll.: 590027630 Date: 2025-10-30

AIM:

- [To Learn linux file systems permissions and essential commands]

Requirements:

- [Any Linux Distro, any kind of text editor (vs code, vim, notepad, nano, etc,)]

Theory:

- [Basic Linux file systems permissions and essential commands]

Procedure & Observations

TASK 1: [Directory Navigation]

Task Statement:

- [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.]

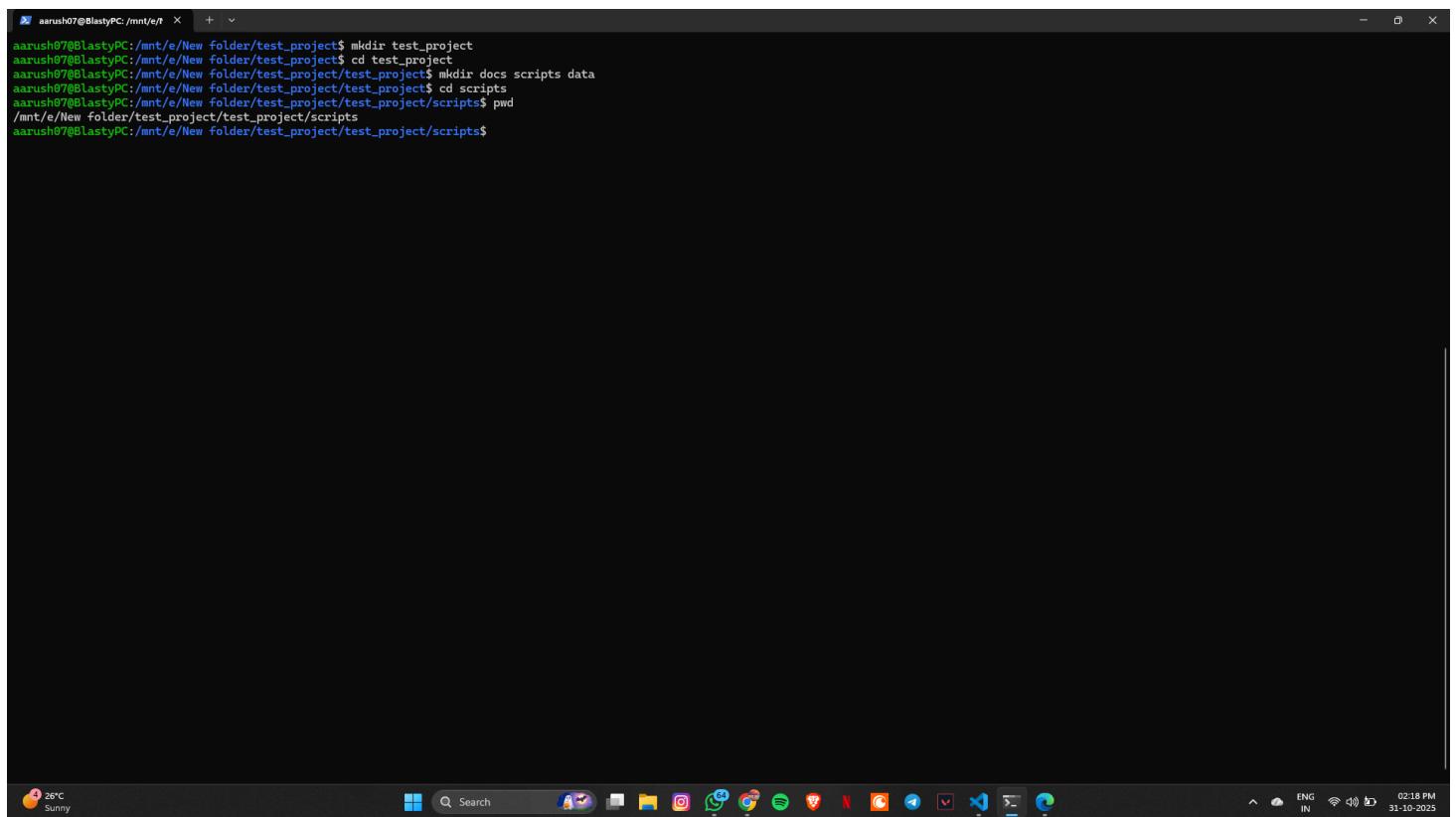
Explanation:

- [Use mkdir to create the wanted directory we can use cd to navigate and use pwd to show current path]

Command(s):

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

Output:



The screenshot shows a terminal window titled "aarush07@BlastyPC: /mnt/e/" with the following command history:

```
aarush07@BlastyPC:/mnt/e/New folder/test_project$ mkdir test_project  
aarush07@BlastyPC:/mnt/e/New folder/test_project$ cd test_project  
aarush07@BlastyPC:/mnt/e/New folder/test_project/test_project$ mkdir docs scripts data  
aarush07@BlastyPC:/mnt/e/New folder/test_project/test_project$ cd scripts  
aarush07@BlastyPC:/mnt/e/New folder/test_project/test_project/scripts$ pwd  
/mnt/e/New folder/test_project/test_project/scripts  
aarush07@BlastyPC:/mnt/e/New folder/test_project/test_project/scripts$
```

The terminal window is set against a dark background. The desktop environment at the bottom includes icons for various applications like File Explorer, Edge, and Task View, along with system status indicators for battery level (26°C), network, and date/time (31-10-2025).

TASK 2: [FILE Creation and Content]

Task Statement:

- [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.]

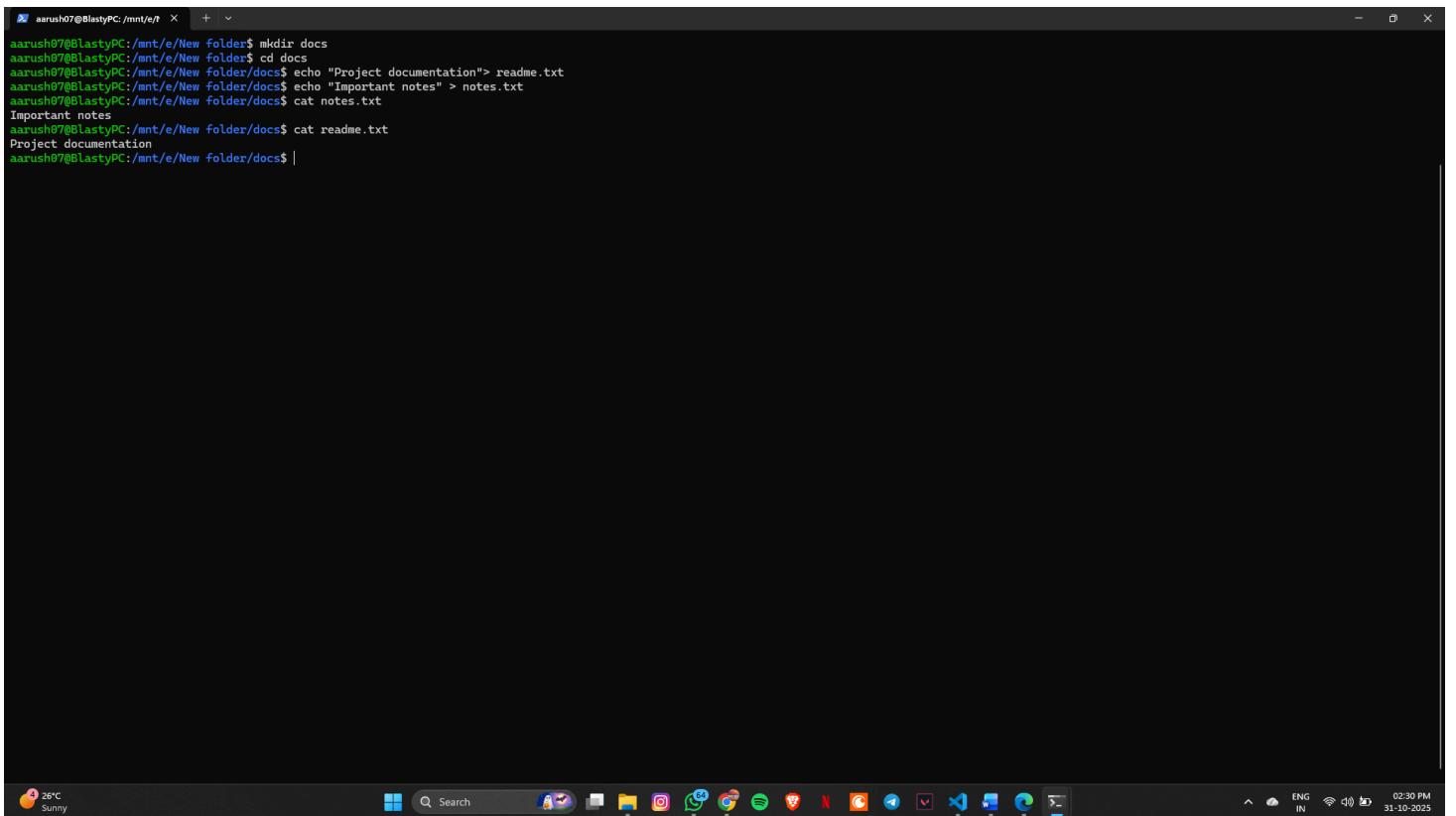
Explanation:

- [We can use touch to create empty files and using echo "text" > file.txt to add content to a file and using 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
cat readme.txt
```

Output:



```
aarush07@BlastyPC:/mnt/e/New folder$ mkdir docs
aarush07@BlastyPC:/mnt/e/New folder$ cd docs
aarush07@BlastyPC:/mnt/e/New folder/docs$ echo "Project documentation"> readme.txt
aarush07@BlastyPC:/mnt/e/New folder/docs$ echo "Important notes" > notes.txt
aarush07@BlastyPC:/mnt/e/New folder/docs$ cat notes.txt
Important notes
aarush07@BlastyPC:/mnt/e/New folder/docs$ cat readme.txt
Project documentation
aarush07@BlastyPC:/mnt/e/New folder/docs$ |
```

The screenshot shows a terminal window on a Linux desktop environment. The terminal history displays the following commands and their outputs:

- `mkdir docs`: Creates a new directory named "docs".
- `cd docs`: Changes the current working directory to "docs".
- `echo "Project documentation"> readme.txt`: Writes the string "Project documentation" to the file "readme.txt".
- `echo "Important notes" > notes.txt`: Writes the string "Important notes" to the file "notes.txt".
- `cat notes.txt`: Prints the contents of the "notes.txt" file, which is "Important notes".
- `cat readme.txt`: Prints the contents of the "readme.txt" file, which is "Project documentation".
- `|`: A final command separator character.

The desktop interface at the bottom includes a weather widget (26°C Sunny), a taskbar with various application icons (Search, File Explorer, Instagram, Spotify, Google Chrome, Netflix, etc.), and system status indicators (ENG IN, battery level, network, 02:30 PM, 31-10-2025).

TASK 3: [FILE Operations]

Task Statement:

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

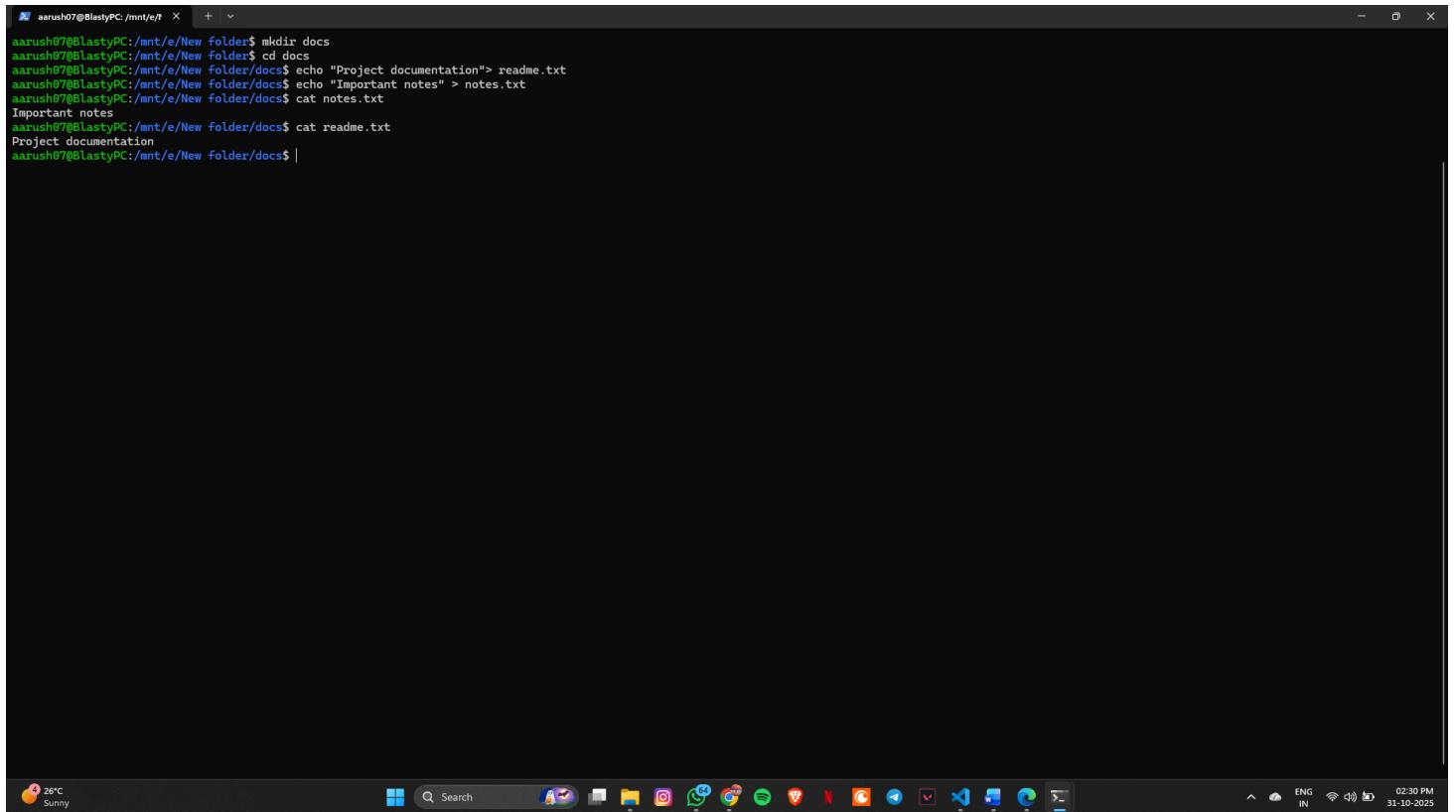
Explanation:

- [- We can use the cp source destination to copy files and using the mv oldname newname to rename files also using the same command mv file directory/ to move files to another directory we can also combine copy and rename: cp file.txt newdir/newname.txt]

Command(s):

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

Output:



The screenshot shows a terminal window on a Windows operating system. The command entered was `cp readme.txt data/project_info.txt`. The terminal output shows the creation of a 'docs' folder, navigating into it, and then copying the 'readme.txt' file to 'project_info.txt'. The terminal window has a dark theme and includes standard Windows taskbar icons at the bottom.

```
aarush07@BlastyPC:/mnt/e/t ~ + - x
aarush07@BlastyPC:/mnt/e/New folder$ mkdir docs
aarush07@BlastyPC:/mnt/e/New folder$ cd docs
aarush07@BlastyPC:/mnt/e/New folder/docs$ echo "Project documentation"> readme.txt
aarush07@BlastyPC:/mnt/e/New folder/docs$ echo "Important notes" > notes.txt
aarush07@BlastyPC:/mnt/e/New folder/docs$ cat notes.txt
Important notes
aarush07@BlastyPC:/mnt/e/New folder/docs$ cat readme.txt
Project documentation
aarush07@BlastyPC:/mnt/e/New folder/docs$ |
```

TASK 4: [FILE Permissions]

Task Statement:

- [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:

- [Using `chmod u+x filename` we can make the file executable for user only using `ls -l` to check for permissions also script files typically need executable permission to run]

Command(s):

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

Output:



TASK 5: [FILE Viewing]

Task Statement:

- [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:

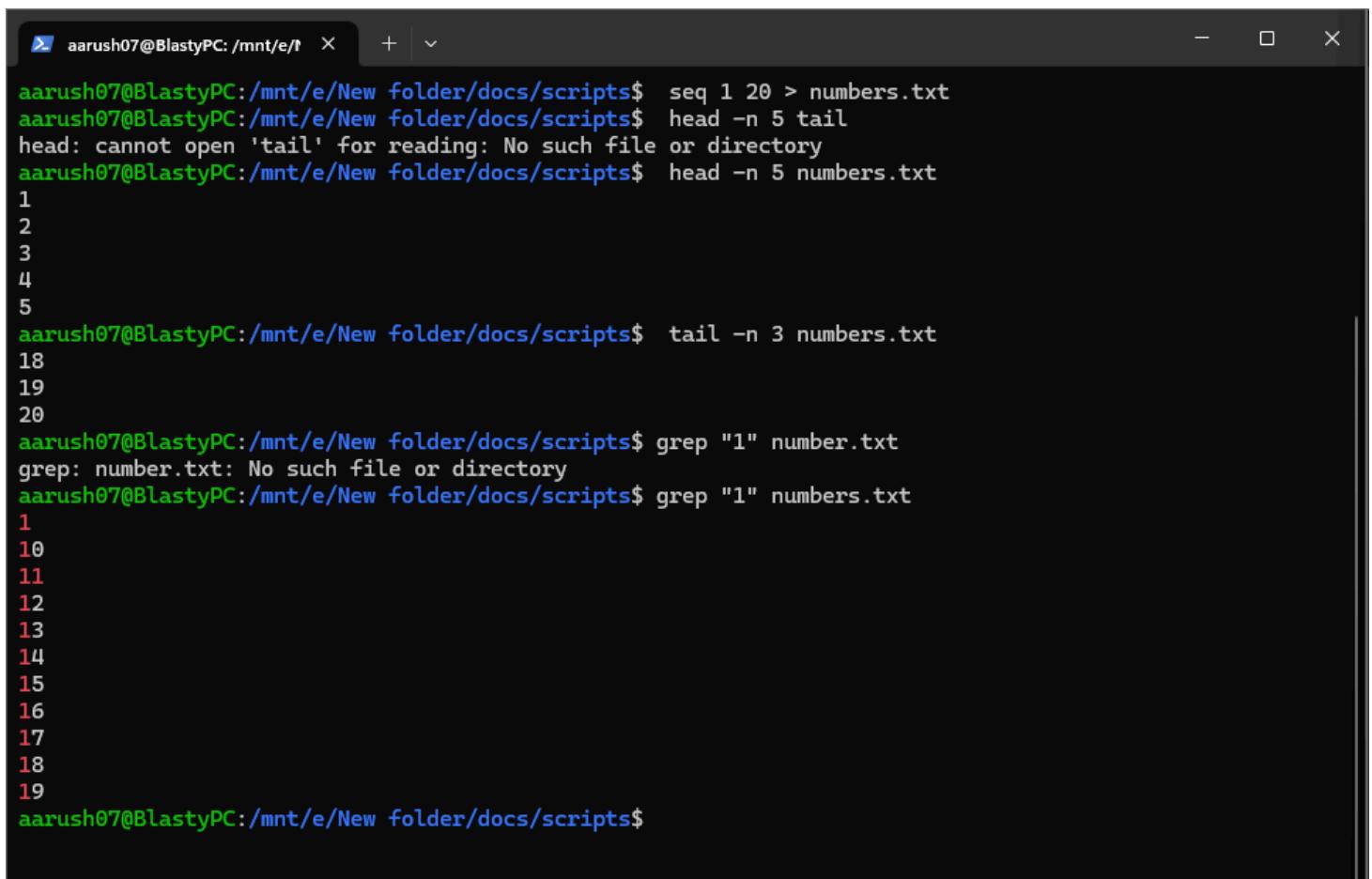
- [I can quickly generate a list of numbers by running `seq 1 20 > numbers.txt`. To check the first few numbers, I use `head -n 5` to see the first 5 lines, and `tail -n 3` to see the last 3 lines. If I want to find

all numbers containing a "1", I can use grep "1". Alternatively, I could create the list manually by using multiple echo commands.]

Command(s):

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

Output:



A screenshot of a terminal window titled "aarush07@BlastyPC: /mnt/e/". The window contains the following text:

```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ seq 1 20 > numbers.txt  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ head -n 5 tail  
head: cannot open 'tail' for reading: No such file or directory  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ head -n 5 numbers.txt  
1  
2  
3  
4  
5  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ tail -n 3 numbers.txt  
18  
19  
20  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ grep "1" number.txt  
grep: number.txt: No such file or directory  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ grep "1" numbers.txt  
1  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$
```

TASK 6: [Text Editing]

Task Statement:

- [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:

- [I open a file in Nano using nano filename.txt and type my content normally. Once I'm done, I press Ctrl+O to save the file and Ctrl+X to exit Nano. After that, I use cat to check the contents and make sure everything was saved correctly.]

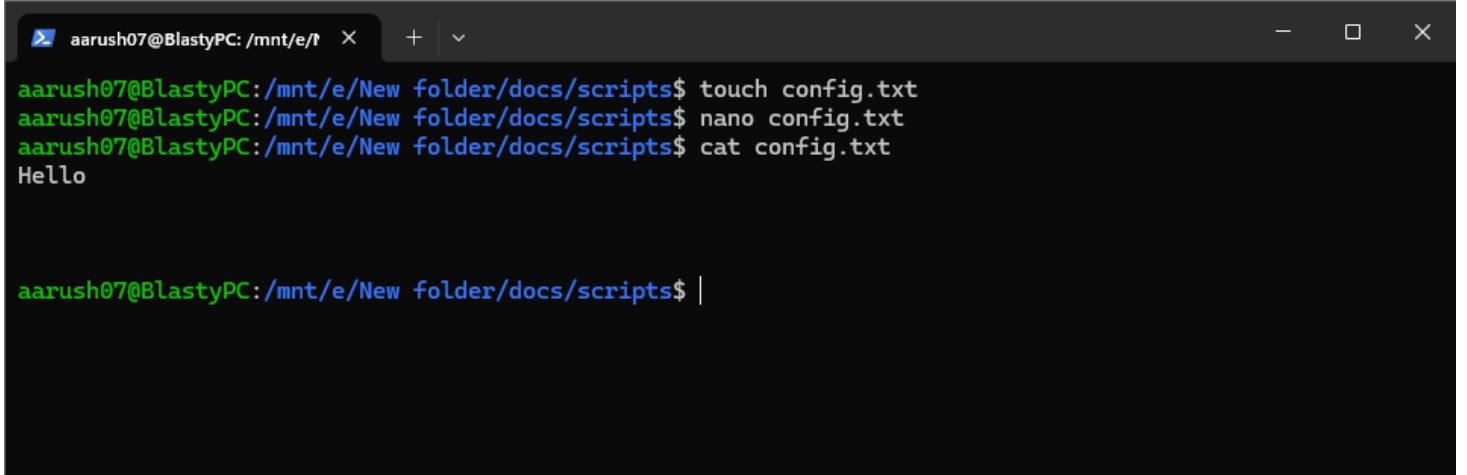
Command(s):

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

Alternatively

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

Output:



```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ touch config.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ nano config.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ cat config.txt
Hello

aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ |
```

A screenshot of a terminal window titled "aarush07@BlastyPC: /mnt/e/New folder/docs/scripts". The window shows a sequence of commands being run: "touch config.txt", "nano config.txt", "cat config.txt", and finally a partially visible command starting with "|". The output of the "cat config.txt" command is "Hello".

TASK 7: [System Information]

Task Statement:

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

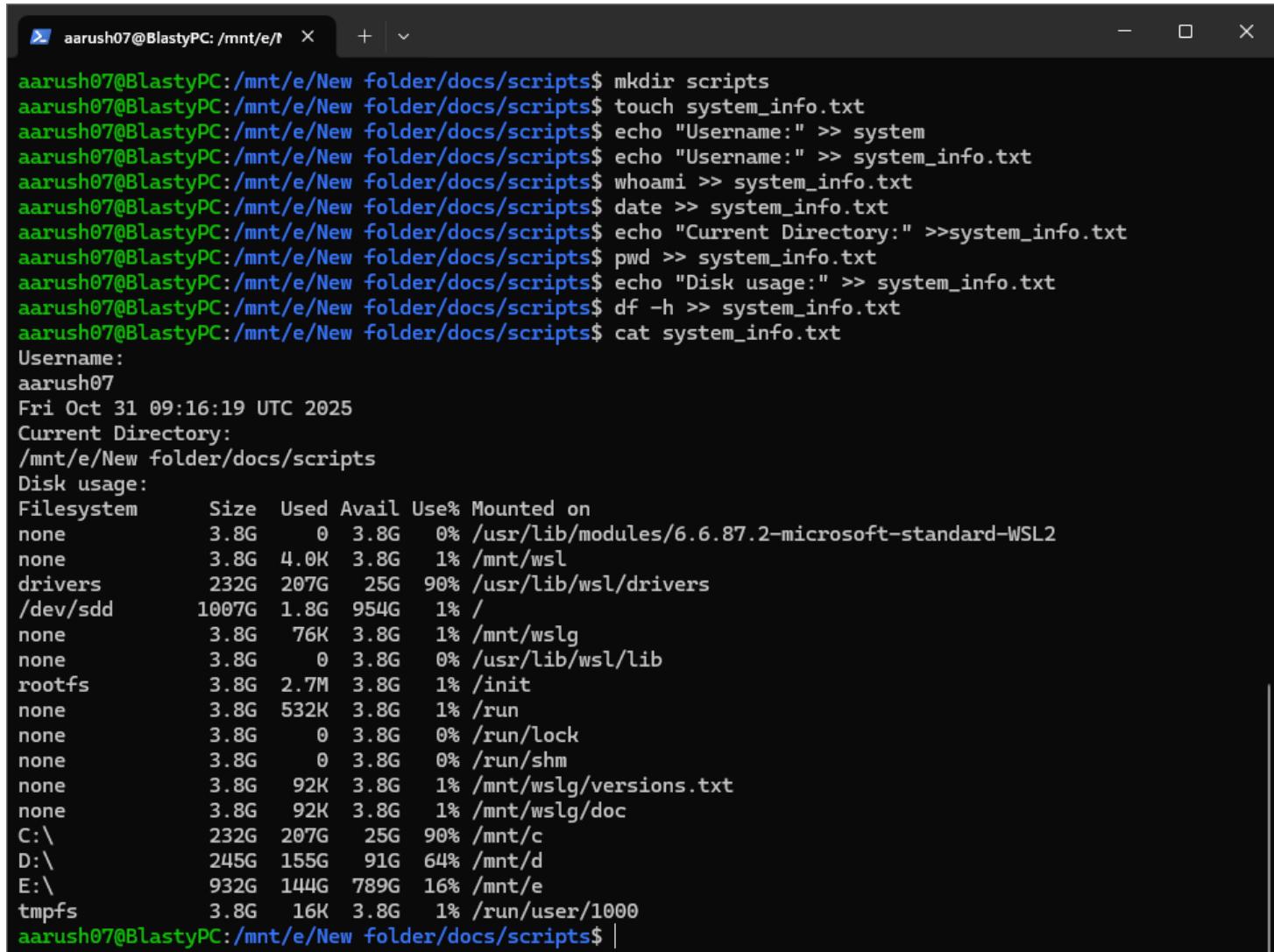
Explanation:

- [I can use whoami to check my username, date to see the current date, and pwd to know my current directory. To check disk usage, I use df -h. I can save the output of any command to a file by using redirection like command >> filename.txt. If I want to add labels, I use echo like this: echo "Username:" >> file.txt.]

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:



```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ mkdir scripts
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ touch system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "Username:" >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "Username:" >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ whoami >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ date >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "Current Directory:" >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ pwd >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "Disk usage:" >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ df -h >> system_info.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ cat system_info.txt
Username:
aarush07
Fri Oct 31 09:16:19 UTC 2025
Current Directory:
/mnt/e/New folder/docs/scripts
Disk usage:
Filesystem      Size  Used Avail Use% Mounted on
none            3.8G    0   3.8G  0% /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2
none            3.8G  4.0K  3.8G  1% /mnt/wsl
drivers         232G  207G   25G 90% /usr/lib/wsl/drivers
/dev/sdd        1007G  1.8G  954G  1% /
none            3.8G   76K  3.8G  1% /mnt/wslg
none            3.8G    0   3.8G  0% /usr/lib/wsl/lib
rootfs          3.8G   2.7M  3.8G  1% /init
none            3.8G  532K  3.8G  1% /run
none            3.8G    0   3.8G  0% /run/lock
none            3.8G    0   3.8G  0% /run/shm
none            3.8G   92K  3.8G  1% /mnt/wslg/versions.txt
none            3.8G   92K  3.8G  1% /mnt/wslg/doc
C:\             232G  207G   25G 90% /mnt/c
D:\             245G  155G   91G 64% /mnt/d
E:\             932G  144G  789G 16% /mnt/e
tmpfs           3.8G   16K  3.8G  1% /run/user/1000
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ |
```

TASK 8: [File Organisation]

Task Statement:

- [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.
]

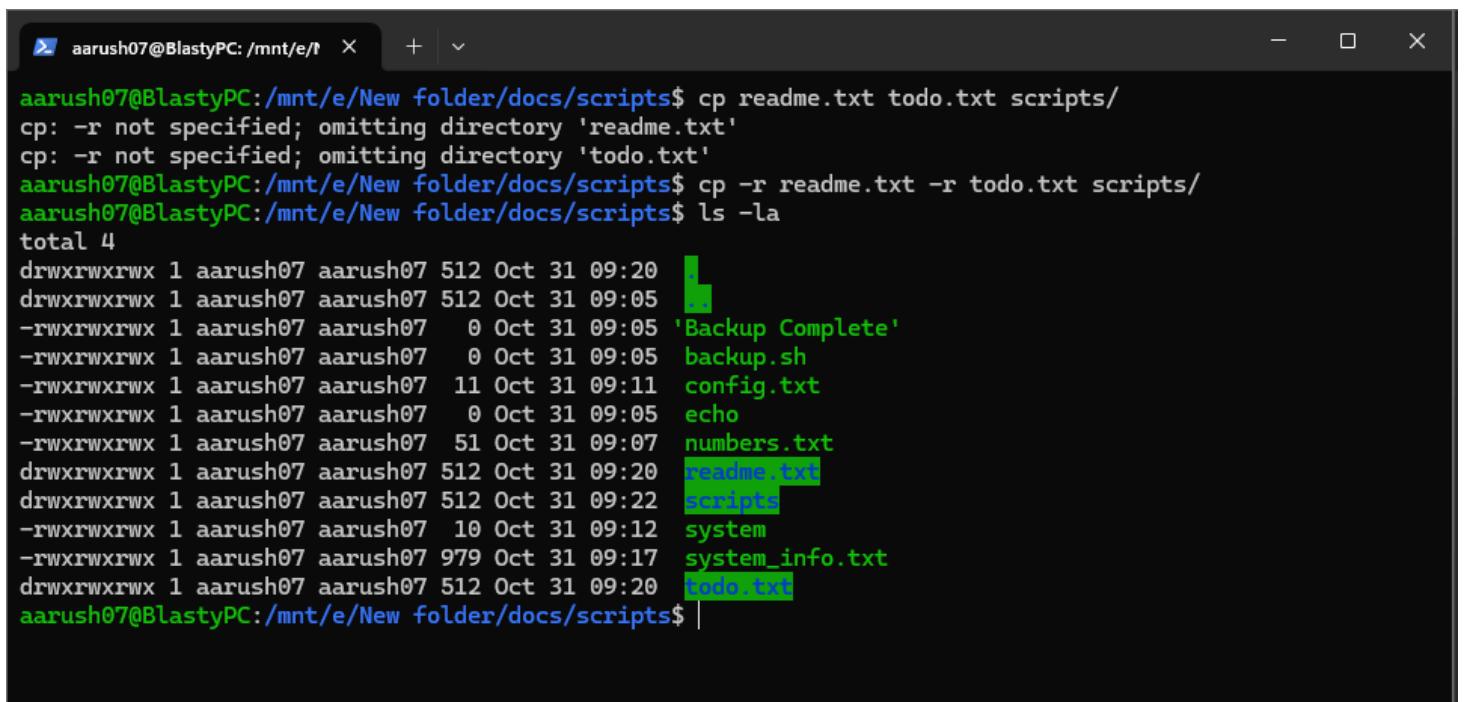
Explanation:

- [I can use find . -name "*.txt" to locate all .txt files. Alternatively, I can navigate to each directory and copy files manually. To copy multiple files at once, I use cp file1.txt file2.txt destination/. If I want detailed information about the files, I use ls -la. The wildcard *.txt helps me match all files that end with .txt.]

Command(s):

```
cp test_project/data/project_info.txt      test_project/docs/notes.txt      test_project/docs/readme.txt
```

Output:



```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ cp readme.txt todo.txt scripts/
cp: -r not specified; omitting directory 'readme.txt'
cp: -r not specified; omitting directory 'todo.txt'
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ cp -r readme.txt -r todo.txt scripts/
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ ls -la
total 4
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:20 .
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:05 .
-rwxrwxrwx 1 aarush07 aarush07  0 Oct 31 09:05 'Backup Complete'
-rwxrwxrwx 1 aarush07 aarush07  0 Oct 31 09:05 backup.sh
-rwxrwxrwx 1 aarush07 aarush07 11 Oct 31 09:11 config.txt
-rwxrwxrwx 1 aarush07 aarush07  0 Oct 31 09:05 echo
-rwxrwxrwx 1 aarush07 aarush07 51 Oct 31 09:07 numbers.txt
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:20 readme.txt
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:22 scripts
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:22 system
drwxrwxrwx 1 aarush07 aarush07 979 Oct 31 09:17 system_info.txt
drwxrwxrwx 1 aarush07 aarush07 512 Oct 31 09:20 todo.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ |
```

TASK 9: [Process and History]

Task Statement:

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

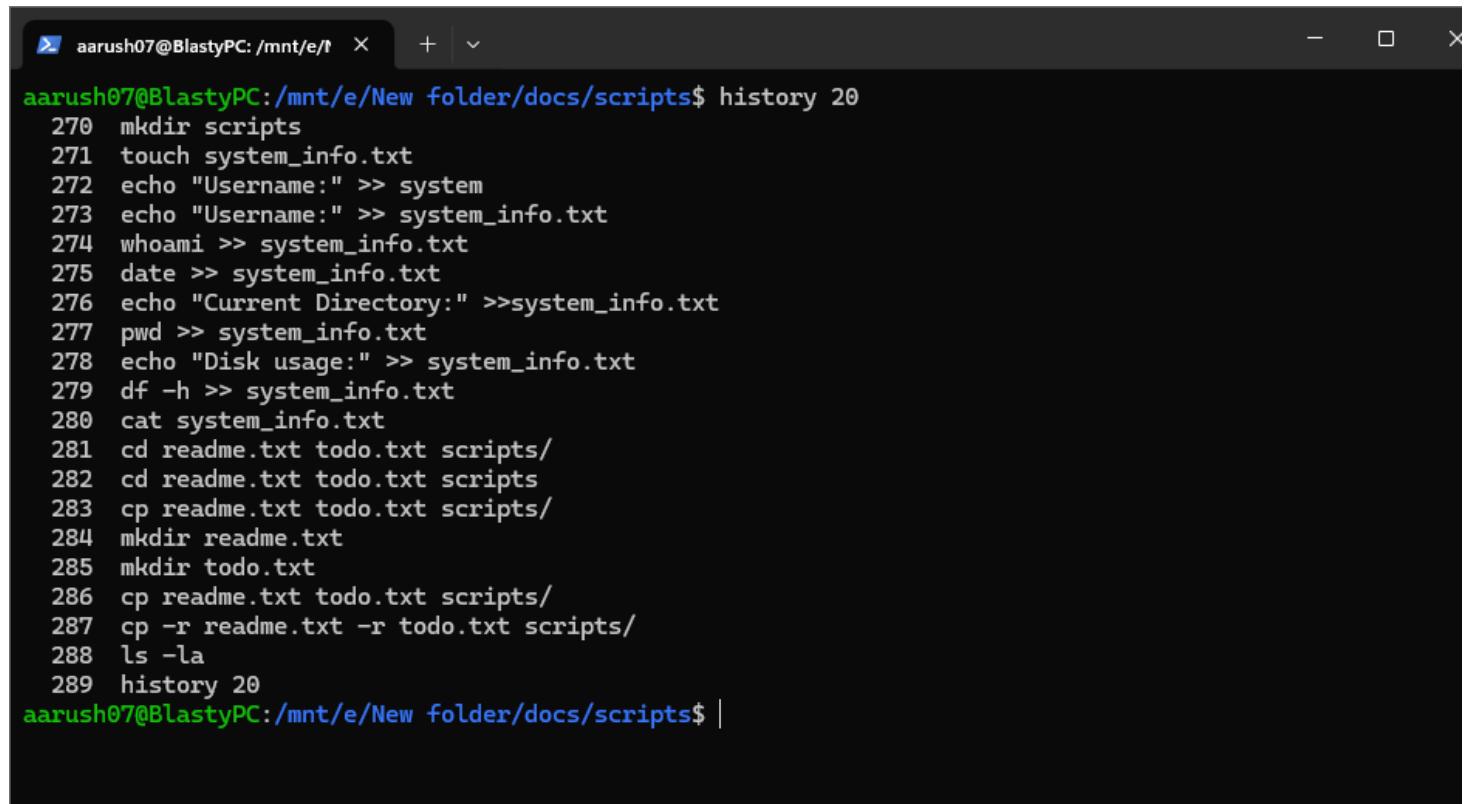
Explanation:

- [I can use history to see all the commands I've typed. To count the total number of commands, I use history | wc -l. If I want to view just the last 10 commands, I can use history 10 or history | tail -10. The wc -l command simply counts the number of lines in the output.]

Command(s):

```
history 10
```

Output:



A screenshot of a terminal window titled "aarush07@BlastyPC: /mnt/e/New folder/docs/scripts\$". The window shows a command history of 289 entries. The entries are numbered from 270 to 289. The commands include file operations like mkdir, touch, cp, and cd, as well as system information commands like whoami, date, and df. The terminal window has a dark background with light-colored text.

```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ history 20
270  mkdir scripts
271  touch system_info.txt
272  echo "Username:" >> system
273  echo "Username:" >> system_info.txt
274  whoami >> system_info.txt
275  date >> system_info.txt
276  echo "Current Directory:" >>system_info.txt
277  pwd >> system_info.txt
278  echo "Disk usage:" >> system_info.txt
279  df -h >> system_info.txt
280  cat system_info.txt
281  cd readme.txt todo.txt scripts/
282  cd readme.txt todo.txt scripts
283  cp readme.txt todo.txt scripts/
284  mkdir readme.txt
285  mkdir todo.txt
286  cp readme.txt todo.txt scripts/
287  cp -r readme.txt -r todo.txt scripts/
288  ls -la
289  history 20
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ |
```

TASK 10: [Comprehensive Cleanup]

Task Statement:

- [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:

- [I can set permissions for `backup.sh` using `chmod 754 backup.sh` to give `rwxr-xr--` permissions. Alternatively, I can use `chmod u=rwx,g=rx,o=r backup.sh`. To count all files, I use `find . -type f | wc -l`, and to count directories, I use `find . -type d | wc -l`. If I want to see the full directory structure recursively, I use `ls -R`. I can also combine multiple commands with `&&` or save the outputs to a summary file for later reference.]

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:

```
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "total Files:" > summary.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ find . -type f | wc -l >>summary.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ echo "total Directories:" >.summary.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ find . -type d | wc -l >> summary .txt
wc: .txt: No such file or directory
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ find . -type d | wc -l >> summary.txt
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ cat summary.txt
total Files:
8
6
aarush07@BlastyPC:/mnt/e/New folder/docs/scripts$ |
```