

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.]

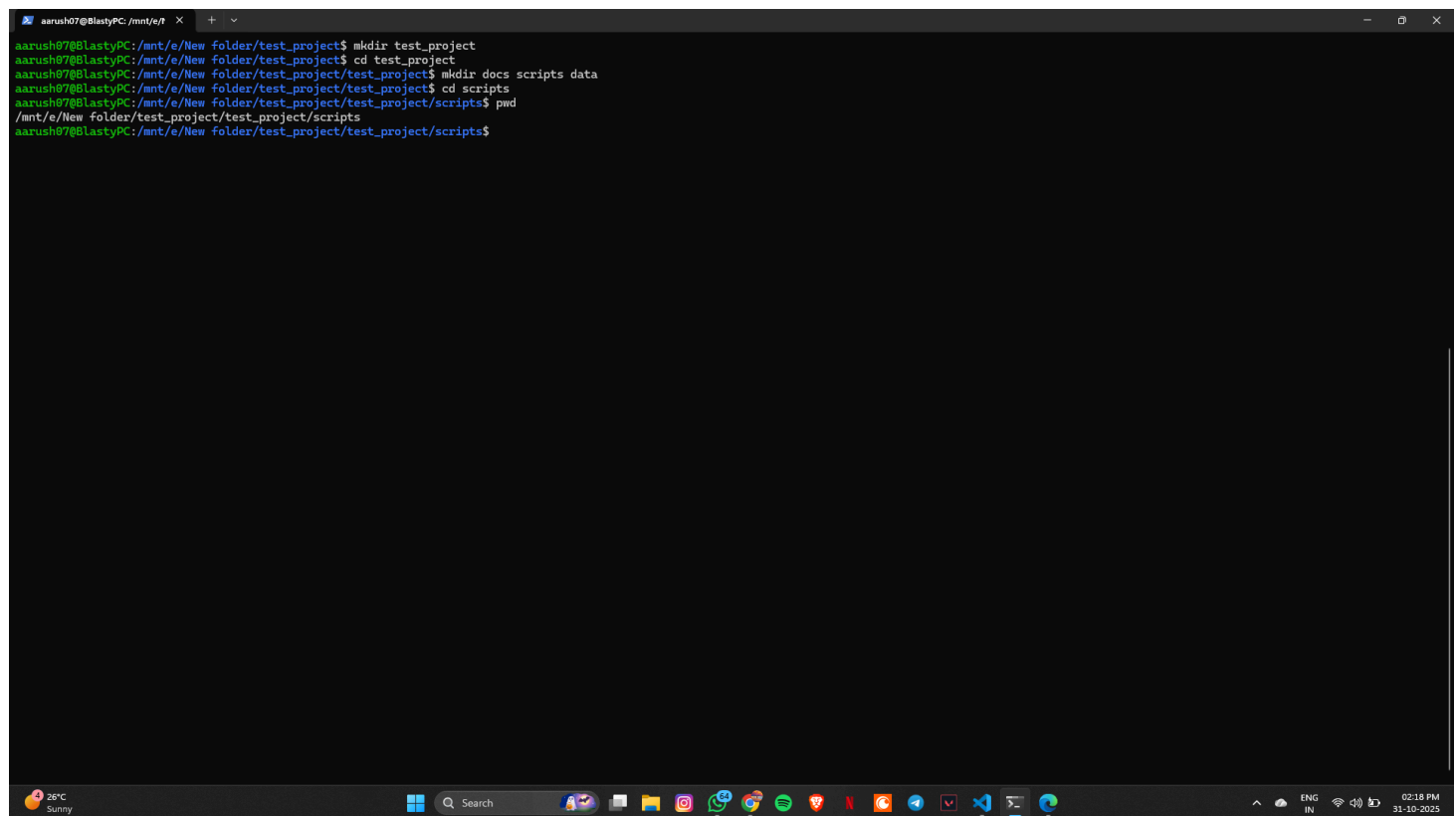
Explanation:

- [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 Windows terminal window with a dark background. The title bar indicates the user is 'aarush07@BlastyPC' in the directory '/mnt/e/t'. The terminal displays the following commands and their outputs:

```
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 Windows taskbar is visible at the bottom, showing the Start button, a search bar, and various application icons. The system tray on the right shows the date and time as 02:18 PM on 31-10-2023.

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.]

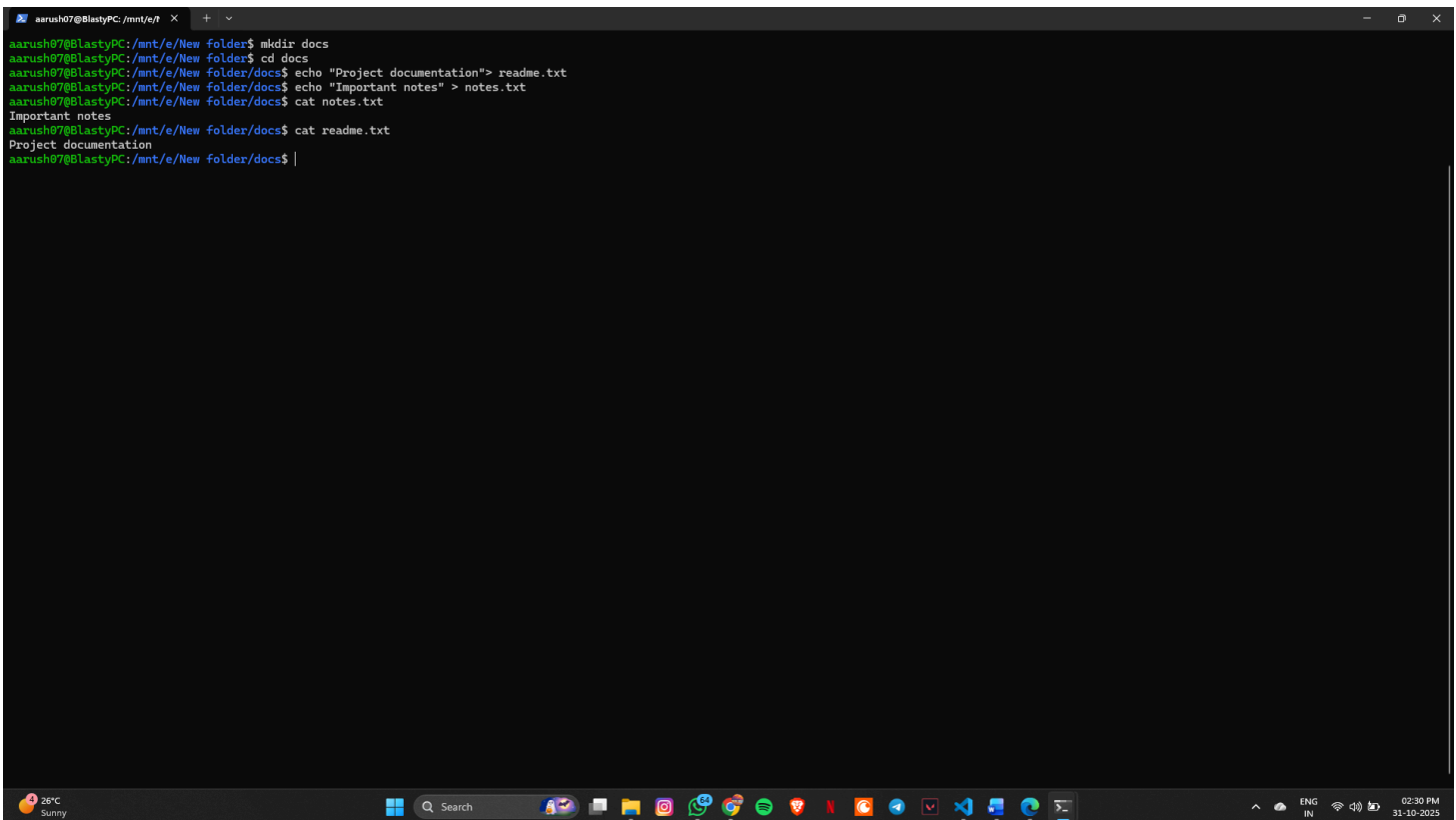
Explanation:

- [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:

A screenshot of a Windows terminal window with a dark background. The window title is 'aarush07@BlastyPC: /mnt/e/r'. The terminal shows a series of commands and their outputs. The commands are: 'mkdir docs', 'cd docs', 'echo "Project documentation"> readme.txt', 'echo "Important notes" > notes.txt', 'cat notes.txt', and 'cat readme.txt'. The outputs are: 'Important notes' and 'Project documentation'. The terminal window is open on a Windows desktop, with the taskbar visible at the bottom showing various application icons and system status information like '25°C Sunny' and '02:30 PM 31-10-2025'.

```
aarush07@BlastyPC: /mnt/e/r
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 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.]

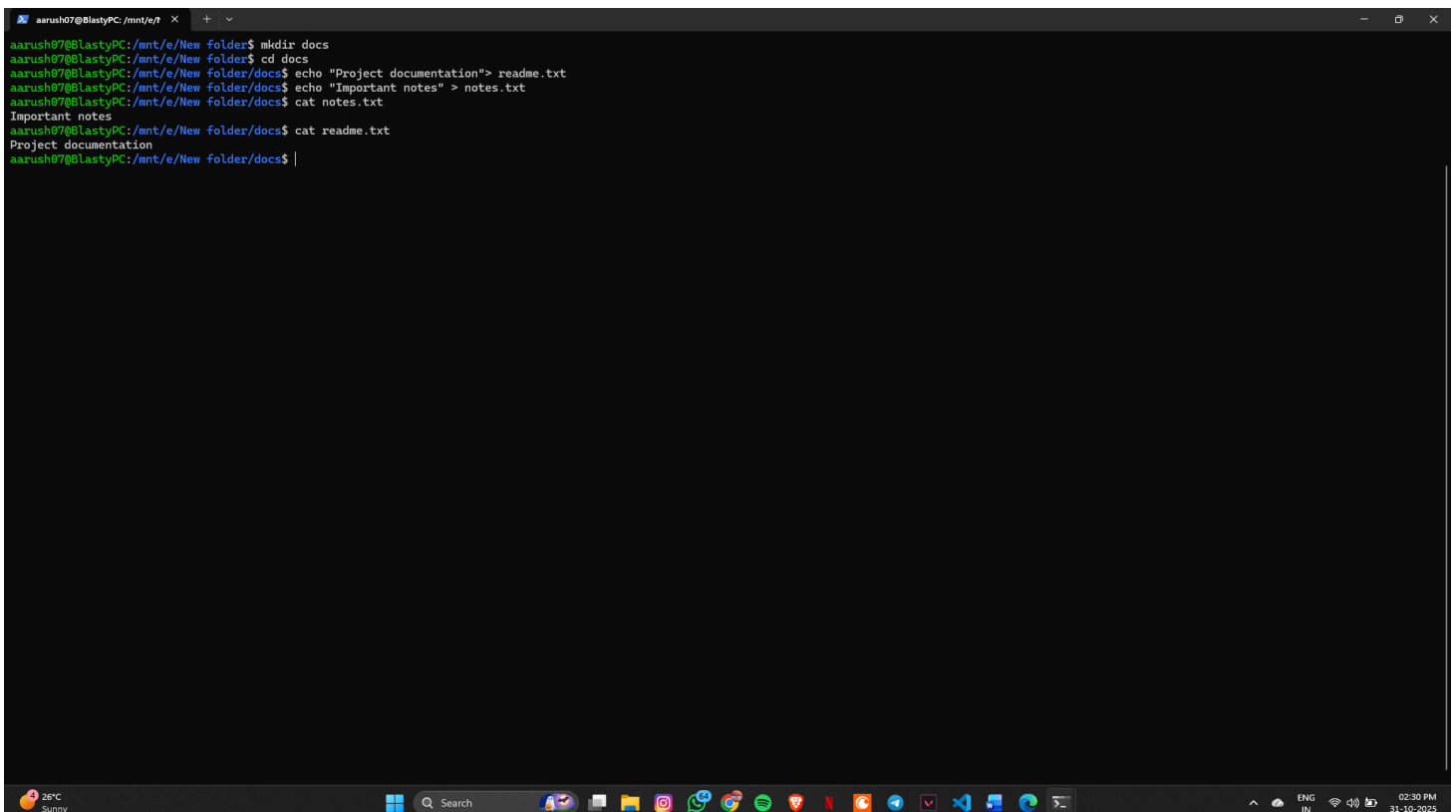
Explanation:

- [- 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:



```
aarush07@BlastyPC: /mnt/e/t  x
+ - v
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 Windows terminal window with a dark theme. The title bar indicates the user is 'aarush07@BlastyPC' and the current directory is '/mnt/e/t'. The terminal content shows a series of commands and their outputs: 'mkdir docs' creates a new directory; 'cd docs' changes the current directory to 'docs'; 'echo "Project documentation"> readme.txt' creates a file named 'readme.txt' with the text 'Project documentation'; 'echo "Important notes" > notes.txt' creates a file named 'notes.txt' with the text 'Important notes'; 'cat notes.txt' displays the contents of 'notes.txt'; and 'cat readme.txt' displays the contents of 'readme.txt'. The terminal window is open over a Windows desktop background, with the taskbar visible at the bottom showing various application icons and system status information like temperature (26°C) and time (02:30 PM).

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:

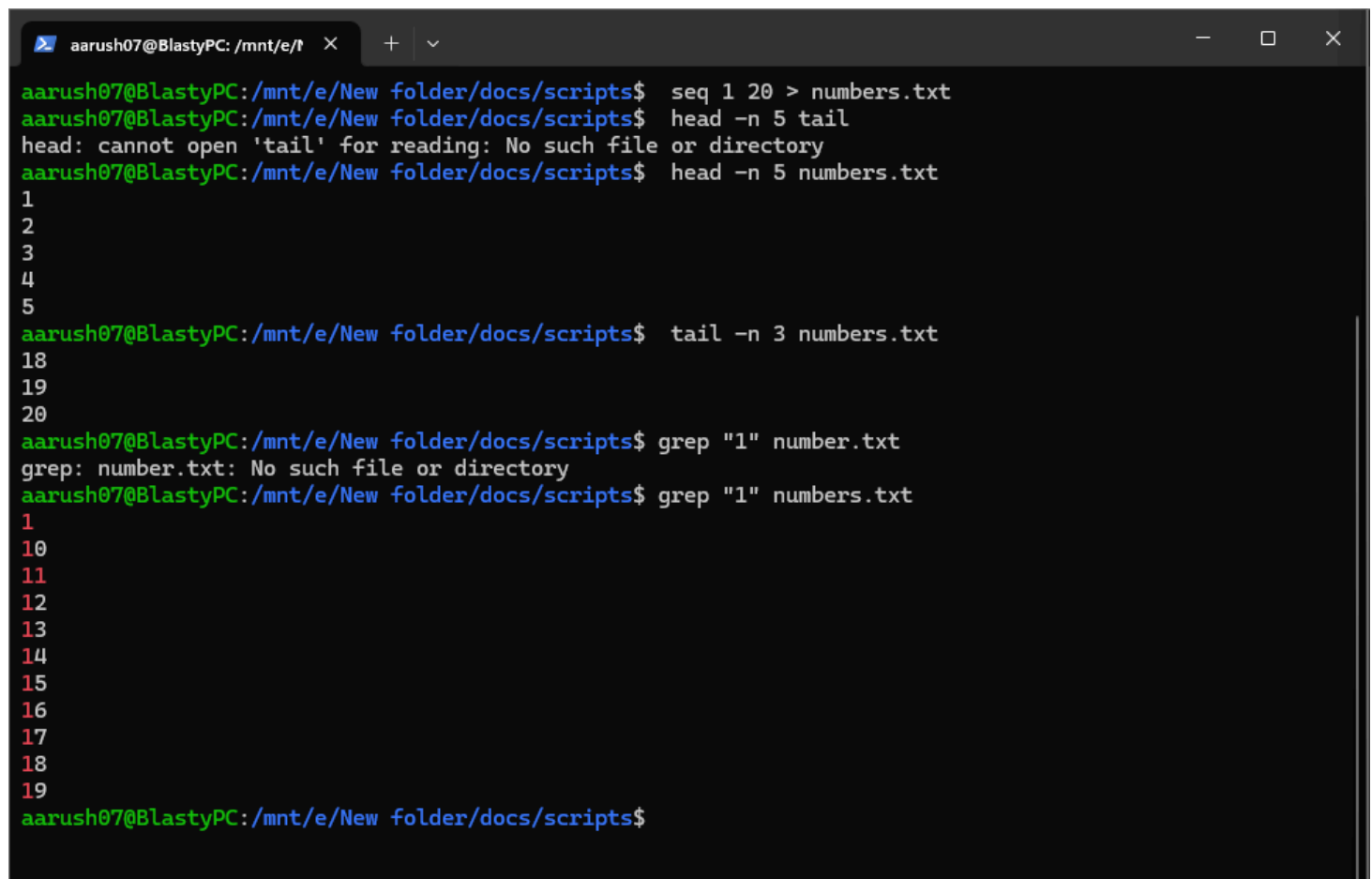
- [I can quickly generate a list of numbers by running `seq 1 20 > numbers.txt`. To check the first few numbers, I use `head -n 5` to see the first 5 lines, and `tail -n 3` to see the last 3 lines. If I want to find

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 terminal window with a dark background and light-colored text. The window title is 'aarush07@BlastyPC: /mnt/e/t'. The terminal shows the following commands and their outputs:
1. Command: `seq 1 20 > numbers.txt`
2. Command: `head -n 5 tail`
 Output: `head: cannot open 'tail' for reading: No such file or directory`
3. Command: `head -n 5 numbers.txt`
 Output: `1
2
3
4
5`
4. Command: `tail -n 3 numbers.txt`
 Output: `18
19
20`
5. Command: `grep "1" number.txt`
 Output: `grep: number.txt: No such file or directory`
6. Command: `grep "1" numbers.txt`
 Output: `1
10
11
12
13
14
15
16
17
18
19`
The terminal ends with the prompt `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.]

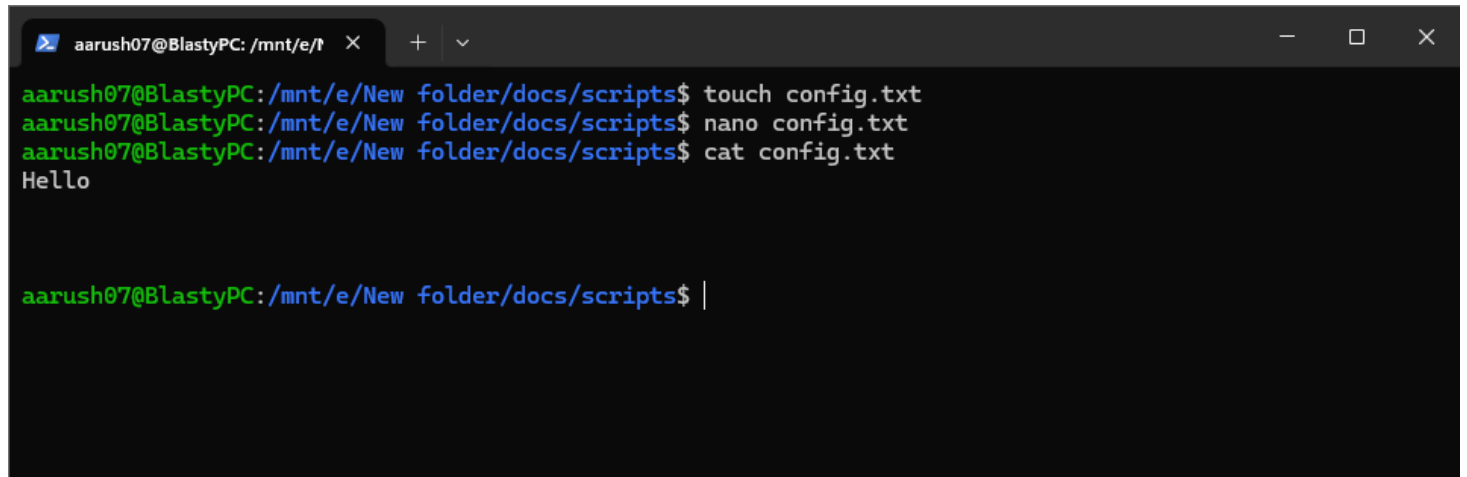
Command(s):

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

Alternatively

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

Output:

A terminal window with a dark background. The title bar shows 'aarush07@BlastyPC: /mnt/e/t' and window control buttons. The terminal text shows a sequence of commands: 'touch config.txt', 'nano config.txt', and 'cat config.txt'. The output of the 'cat' command is 'Hello'. The prompt is '/mnt/e/New folder/docs/scripts\$'.

```
aarush07@BlastyPC: /mnt/e/t X + v
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$ |
```

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.
]

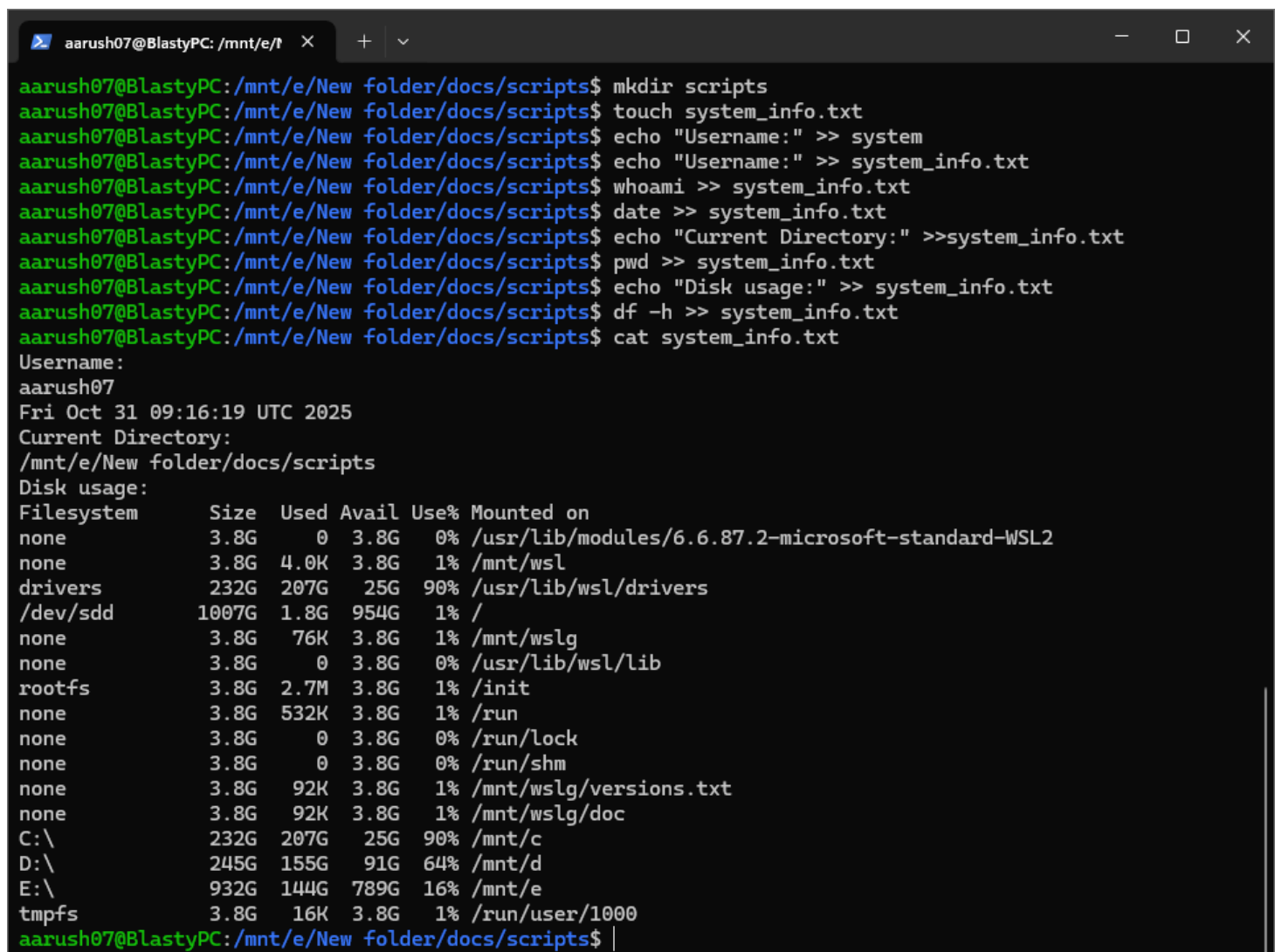
Explanation:

- [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:

A terminal window titled 'aarush07@BlastyPC: /mnt/e/t' with standard window controls. The terminal shows a series of commands being executed in a directory path: /mnt/e/New folder/docs/scripts. The commands are: mkdir 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, and cat system_info.txt. The output of these commands is displayed below, showing the username 'aarush07', the date 'Fri Oct 31 09:16:19 UTC 2025', the current directory '/mnt/e/New folder/docs/scripts', and a detailed disk usage table. The table lists various filesystems with their sizes, used space, available space, usage percentage, and mount points. The command prompt at the bottom is '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
```

Output:

```
aarush07@BlastyPC: /mnt/e/t  X + v
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
-rwxrwxrwx 1 aarush07 aarush07 10 Oct 31 09:12 system
-rwxrwxrwx 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.

]

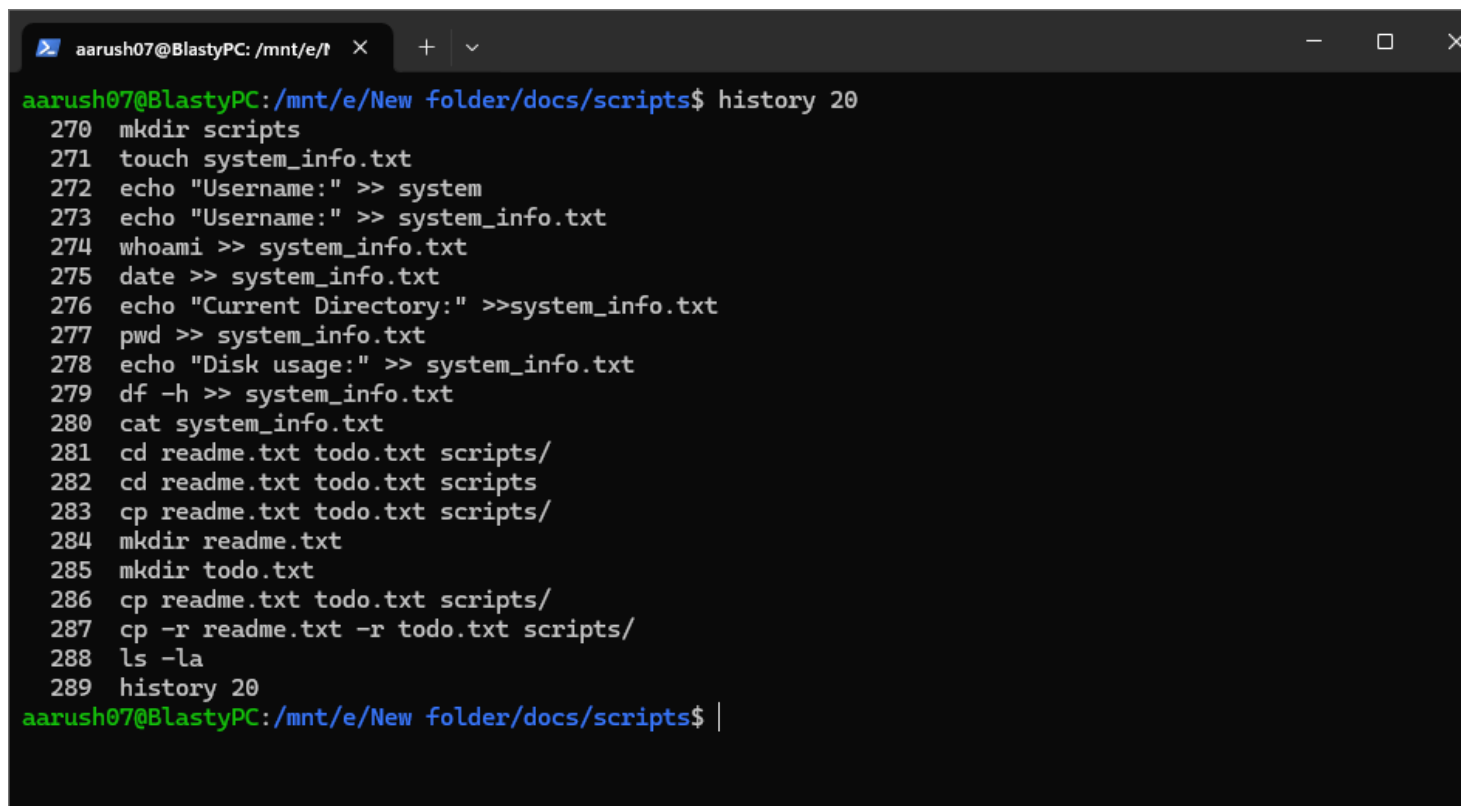
Explanation:

- [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 terminal window with a dark background. The title bar shows 'aarush07@BlastyPC: /mnt/e/...' with window control buttons. The prompt is 'aarush07@BlastyPC: /mnt/e/New folder/docs/scripts\$'. The command 'history 20' has been entered. The output shows a list of 20 commands, each preceded by a line number from 270 to 289. The commands include 'mkdir scripts', 'touch system_info.txt', 'echo "Username:" >> system', 'echo "Username:" >> system_info.txt', 'whoami >> 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', 'cat system_info.txt', 'cd readme.txt todo.txt scripts/', 'cd readme.txt todo.txt scripts', 'cp readme.txt todo.txt scripts/', 'mkdir readme.txt', 'mkdir todo.txt', 'cp readme.txt todo.txt scripts/', 'cp -r readme.txt -r todo.txt scripts/', 'ls -la', and 'history 20'. The prompt is followed by a vertical bar '|'.

```
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 `rw-r--r--` 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/t  X + v
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$ |
```