National Textile University, Faisalabad

Department of Computer Science

Name	Dawood Saif
Class	SE-5th (A)
Reg. No.	23-NTU-CS-1145
Course	Operating system
Submitted To	Sir Nasir
Submission Date	9/10/2025
Lab No.	3 (Home Work)

Lab No. 3 (Home Work)

Part 1: File and Directory Operations:

Create the following directory structure in your home directory:

```
Lab_3/
├── docs/
│   └── drafts/
├── data/
│   ├── raw/
│   └── processed/
└── scripts/
```

2. Inside docs/:

Create three files: intro.txt, notes.txt, summary.txt.

Add at least **two lines of text** into each using echo >>.

Copy summary.txt into the drafts/ folder using cp command.

3. Inside data/raw/:

Create two files: raw1.txt, raw2.txt.

Append the **current date** into raw1.txt using the date command.

Move raw2.txt into processed/ using mv. The syntax is:

```
mv source destination
```

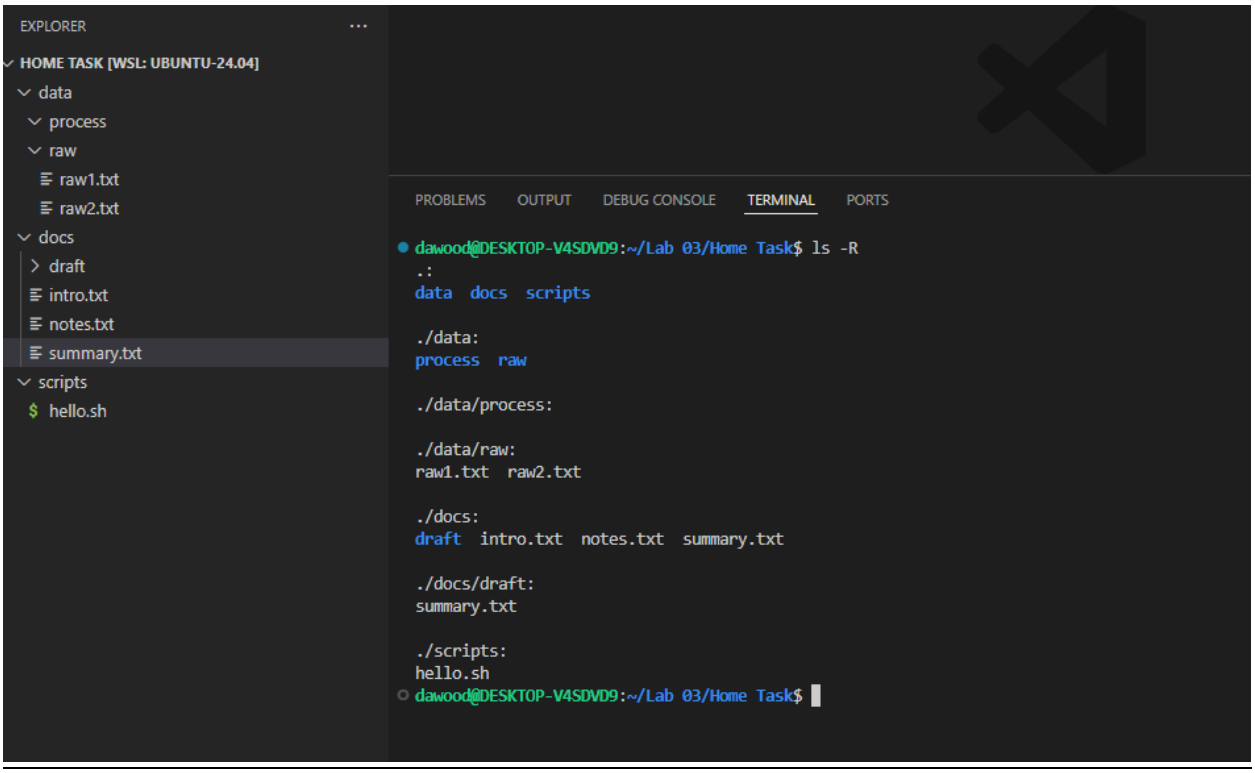
4. Inside scripts/:

Create a script named hello.sh with the following content:

```
echo "Hello World"
pwd
ls -lh
```

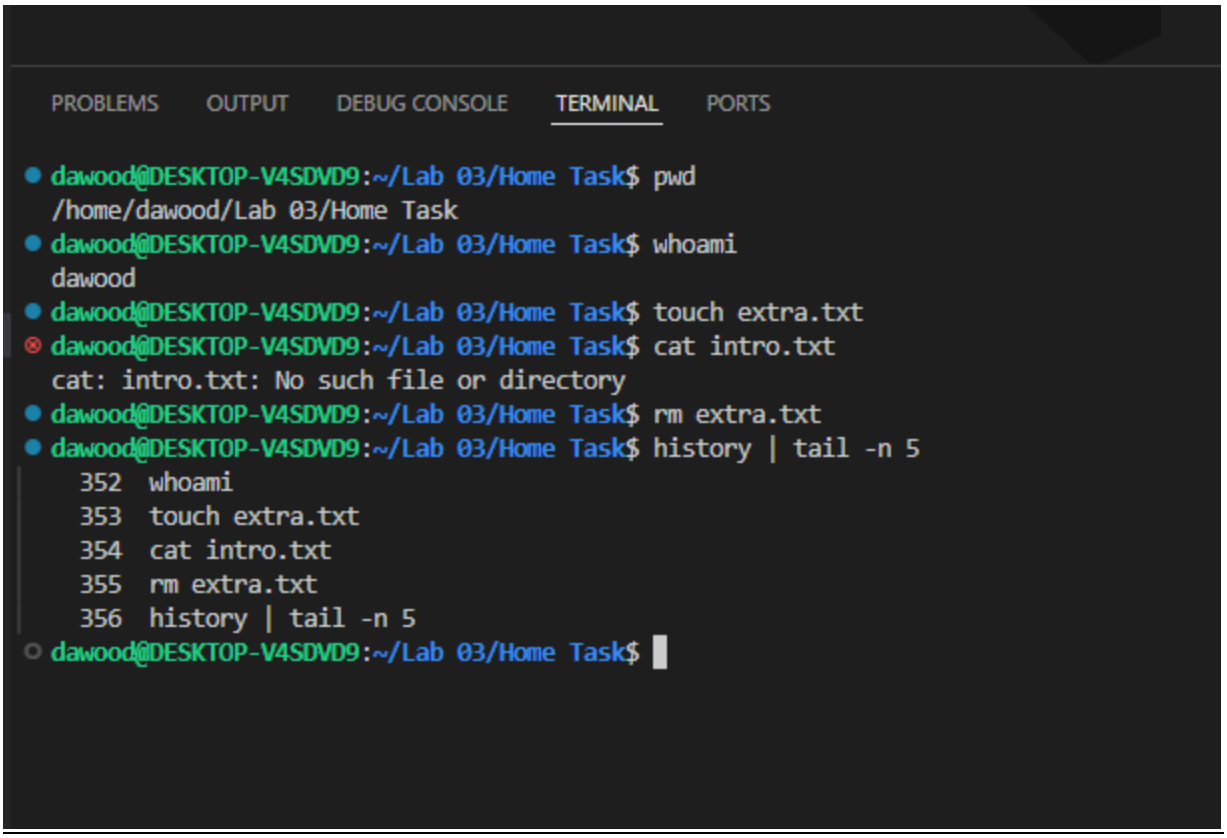
Later, you will make it executable (in Part 3).

5. Display the directory structure recursively and take a Screenshot:



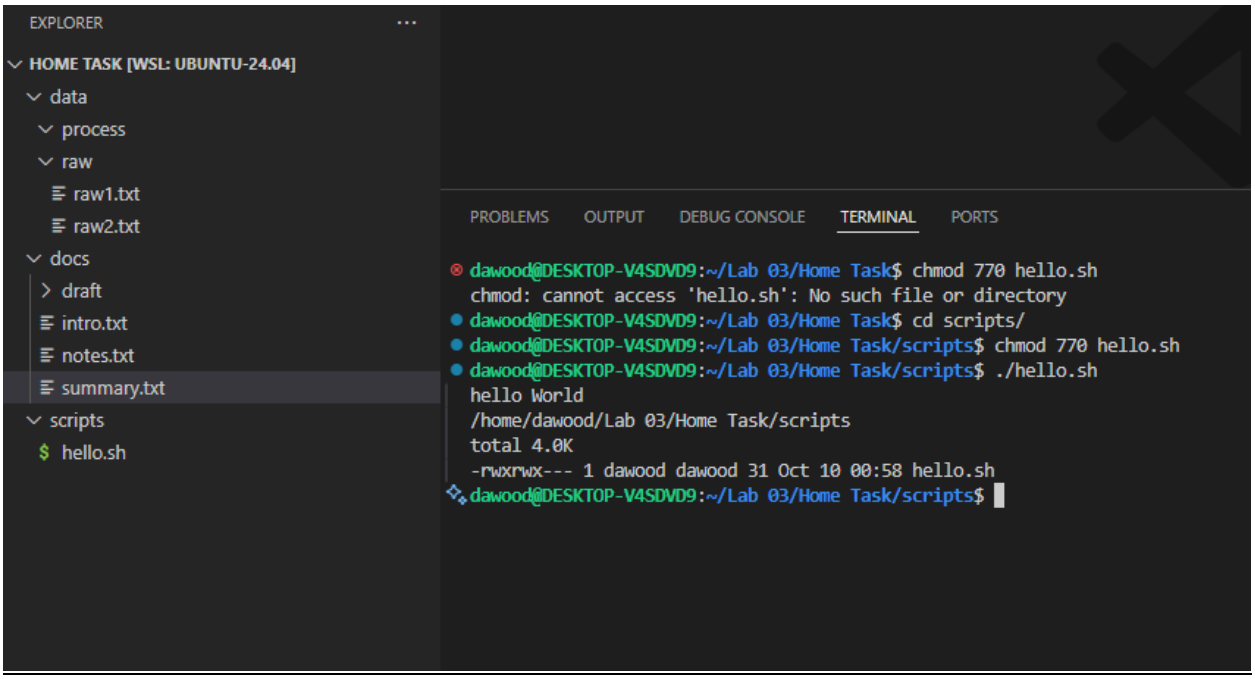
Part 2: Practice with Basic Linux Commands

Screenshot:

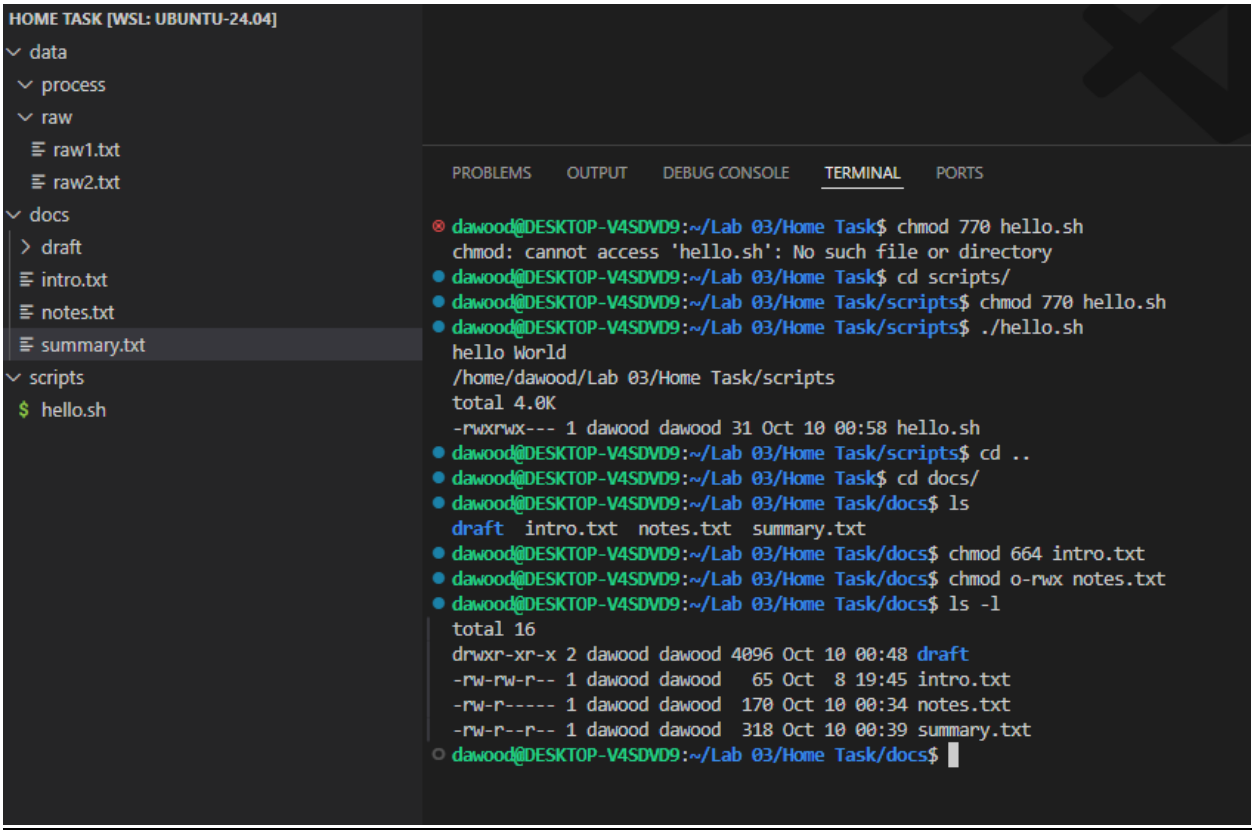


Part 3: File Permissions and Ownership

1. Changing Permissions of hello.sh using 770 in chmod



- Change the permissions of intro.txt using **numeric notation** so that
- Change the permissions of notes.txt using **symbolic notation** so that others don't have any permission on it.



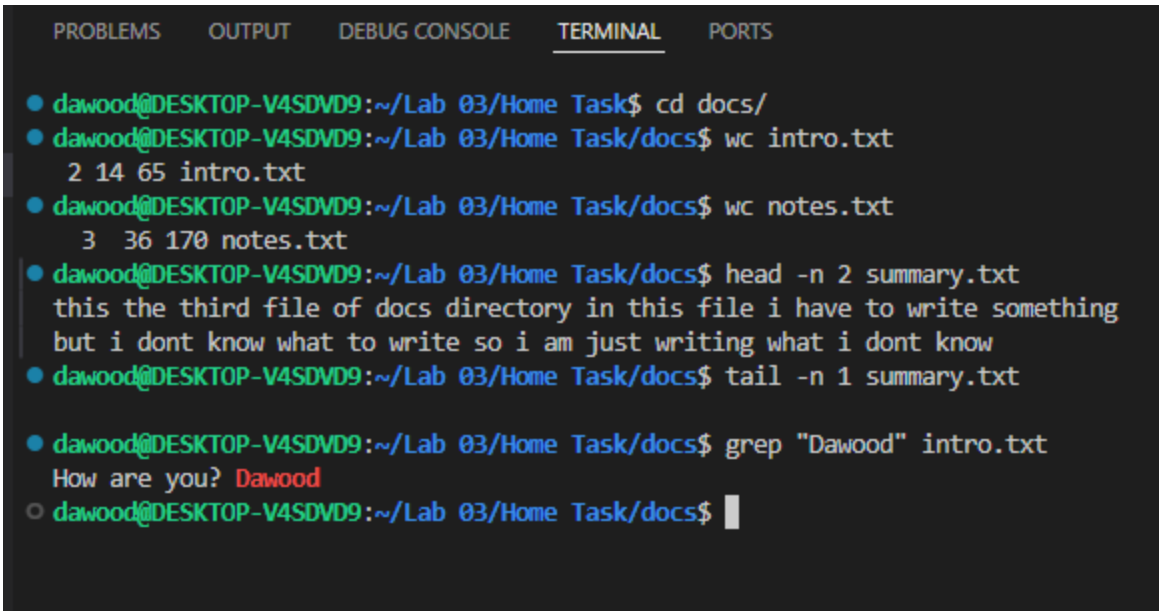
Part 4: Reading & Searching Files:

1. Count the number of lines, words, and characters in notes.txt using wc .
2. Show only the first 2 lines of summary.txt using head -n 2 .

Take screenshots.

3. Show the last line of summary.txt using tail -n 1 .

4. Search for a keyword (of your choice) in intro.txt using grep



Part 5: Linux Process Commands

1. **Exploring Processes**

Use `ps -ef` and identify **3 processes** running on your system. Note their **PID,**

PPID, and command.

Run `top` for 20–30 seconds. Write down:

Which process is consuming the most CPU.

Which process is consuming the most memory.

2. **Practice with Infinite Process**

Start:

`yes > /dev/null &`

Locate its PID using `ps -ef | grep yes` .

Kill it using `kill <PID>` and verify using `ps` .

3. **Foreground & Background Jobs**

Run `sleep 60` in **foreground** and terminate it with **Ctrl + C**.

Run `sleep 60 &` in **background**, bring it to foreground with `fg` , stop with **Ctrl + Z**,

then resume in background using `bg` .

Part 6: C Programs on Processes

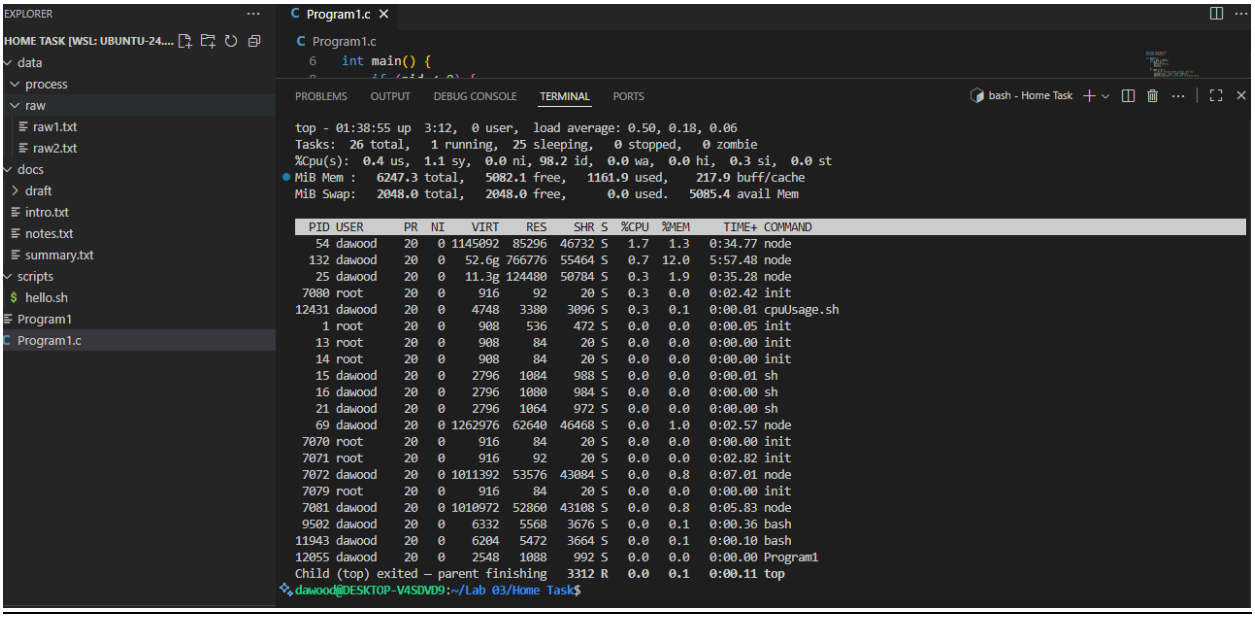
Program 1 – Exec with top

Modify the exec program so that the child runs top instead of ls -l .

Run the program.

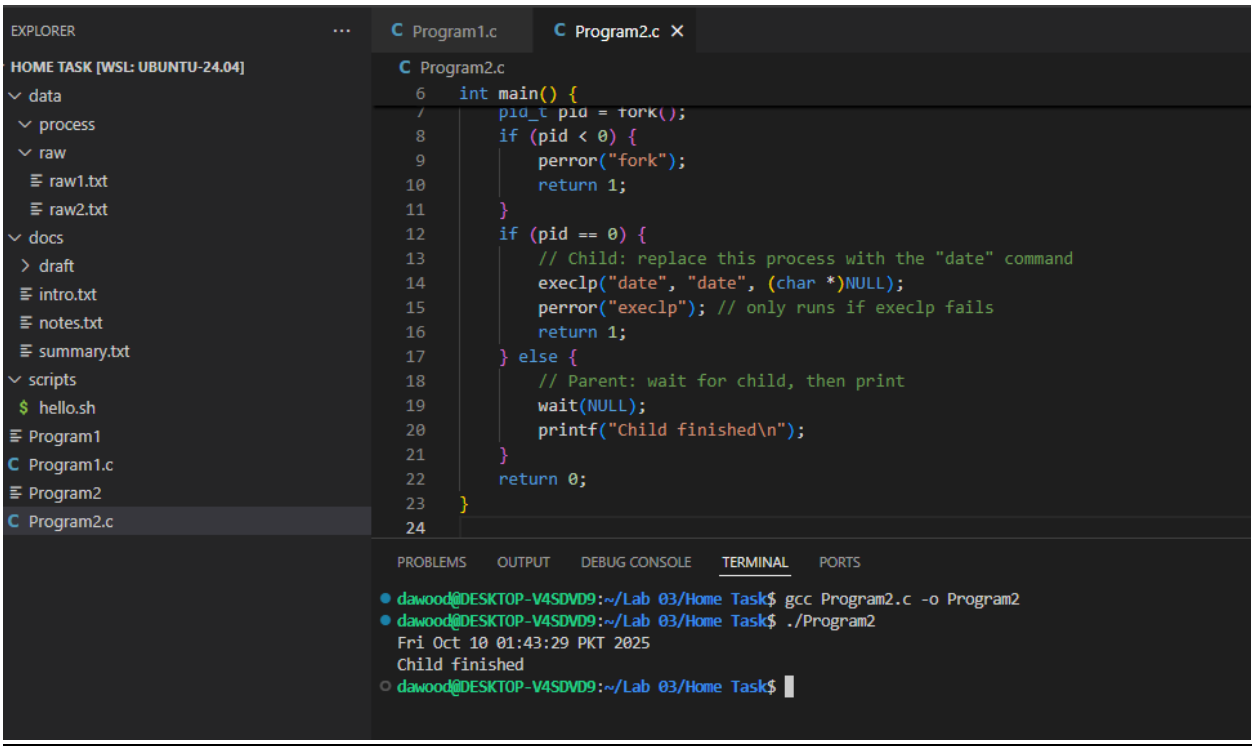
In another terminal, use ps -ef | grep top (or run top) to find the child's PID.

Use the child's process ID to kill it manually.



Program 2 – Incomplete Program

Task: Complete the missing parts, run the program, and take a screenshot of the output.


EXPLORER
HOME TASK [WSL: UBUNTU-24.04]
data
process
raw
raw1.txt
raw2.txt
docs
draft
intro.txt
notes.txt
summary.txt
scripts
hello.sh
Program1
Program1.c
Program2
Program2.c
Program1.c
Program2.c
Program2.c
int main() {
pid_t pid = fork();
if (pid < 0) {
perror("fork");
return 1;
}
if (pid == 0) {
// Child: replace this process with the "date" command
execlp("date", "date", (char *)NULL);
perror("execlp"); // only runs if execlp fails
return 1;
} else {
// Parent: wait for child, then print
wait(NULL);
printf("Child finished\n");
}
return 0;
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
dawood@DESKTOP-V4SDVD9:~/Lab 03/Home Task$ gcc Program2.c -o Program2
dawood@DESKTOP-V4SDVD9:~/Lab 03/Home Task$ ./Program2
Fri Oct 10 01:43:29 PKT 2025
Child finished
dawood@DESKTOP-V4SDVD9:~/Lab 03/Home Task$