**Operating Systems – COC 3071L**

# SE 5th A – Fall 2025 Part 1: File and Directory Operations

1. Create the following directory structure in your home directory:

Lab

\_

3

/

├──

do

c

s

/

│

└──

dr

a

fts

/

├──

d

a

t

a

/

│

├──

r

a

w

/

│

└──

pro

c

essed

/

└──

s

c

ripts

/

2

.

Inside

do

c

s

/

:

Create three files:

intro

.

txt

,

notes

.

txt

,

summ

a

ry

.

txt

.

Add at least

**two lines of text**

into each using

e

c

ho

>>

.

Copy

summ

a

ry

.

txt

into the

dr

a

fts

/

folder using

c

p

command.

3

.

Inside

d

a

t

a

/

r

a

w

/

:

Create two files:

r

a

w

1

.

txt

,

r

a

w

2

.

txt

.

Append the

**current date**

into

r

a

w

1

.

txt

using the

d

a

te

command.

Move

r

a

w

2

.

txt

into

pro

c

essed

/

using

mv

. The syntax is:

mv

sour

c

e

destin

a

tion

4

.

Inside

s

c

ripts

/

:

Create a script named

hello

.

sh

with the following content:

e

c

ho

"

H

ello

W

orld

"

pwd

ls

-

lh

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

5

.

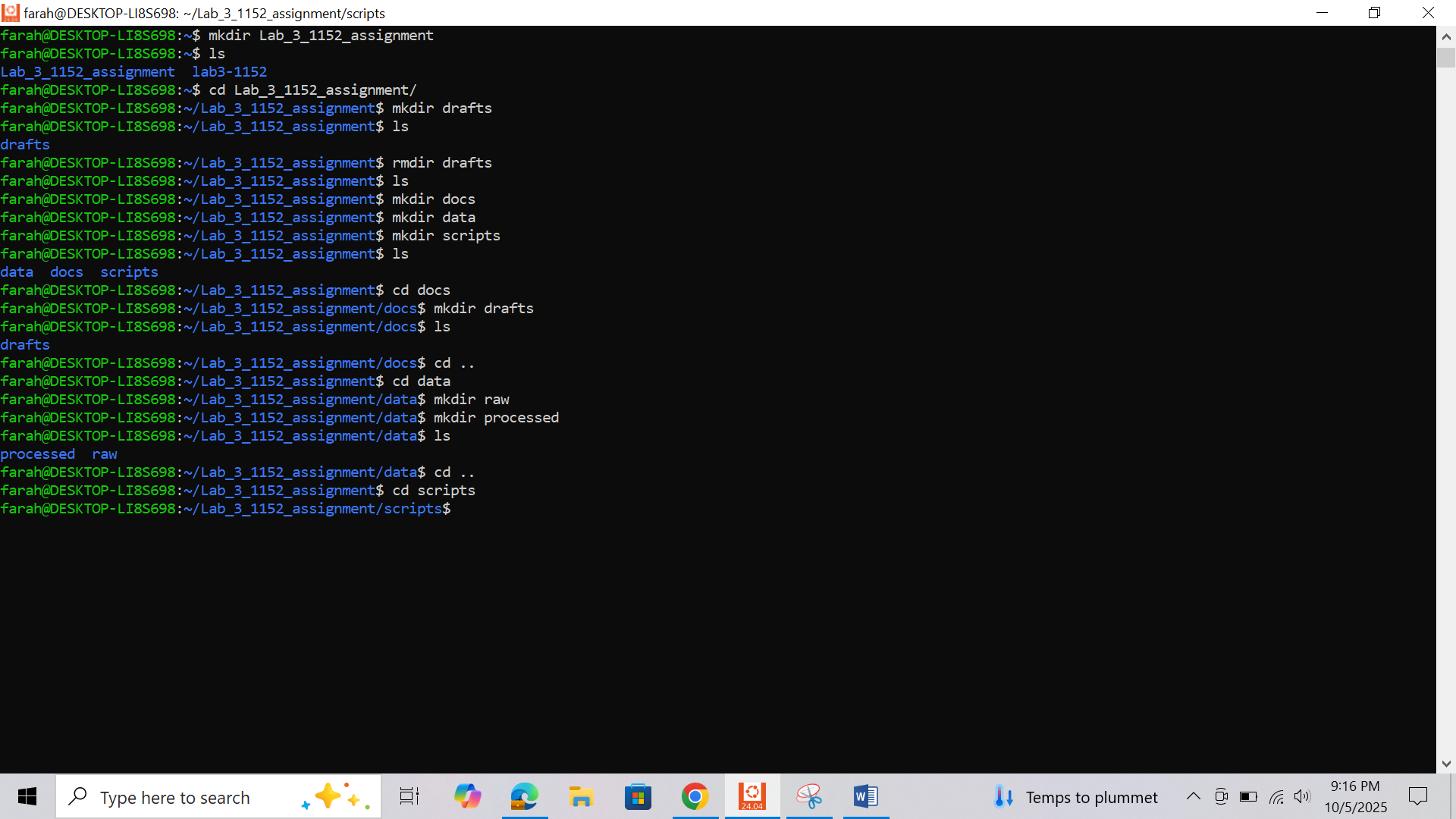
Display the directory structure recursively and take a screenshot:

ls

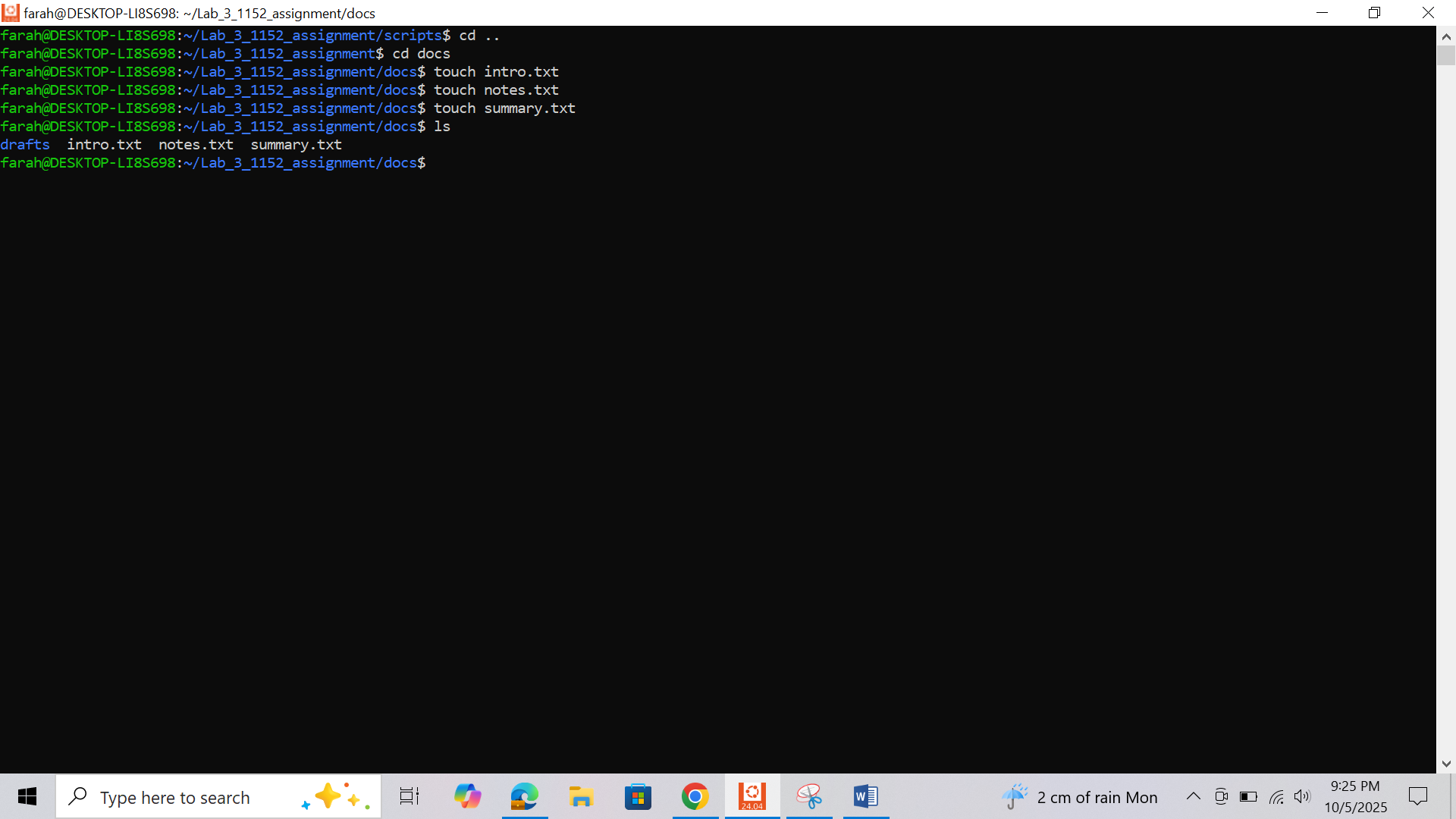
-

R

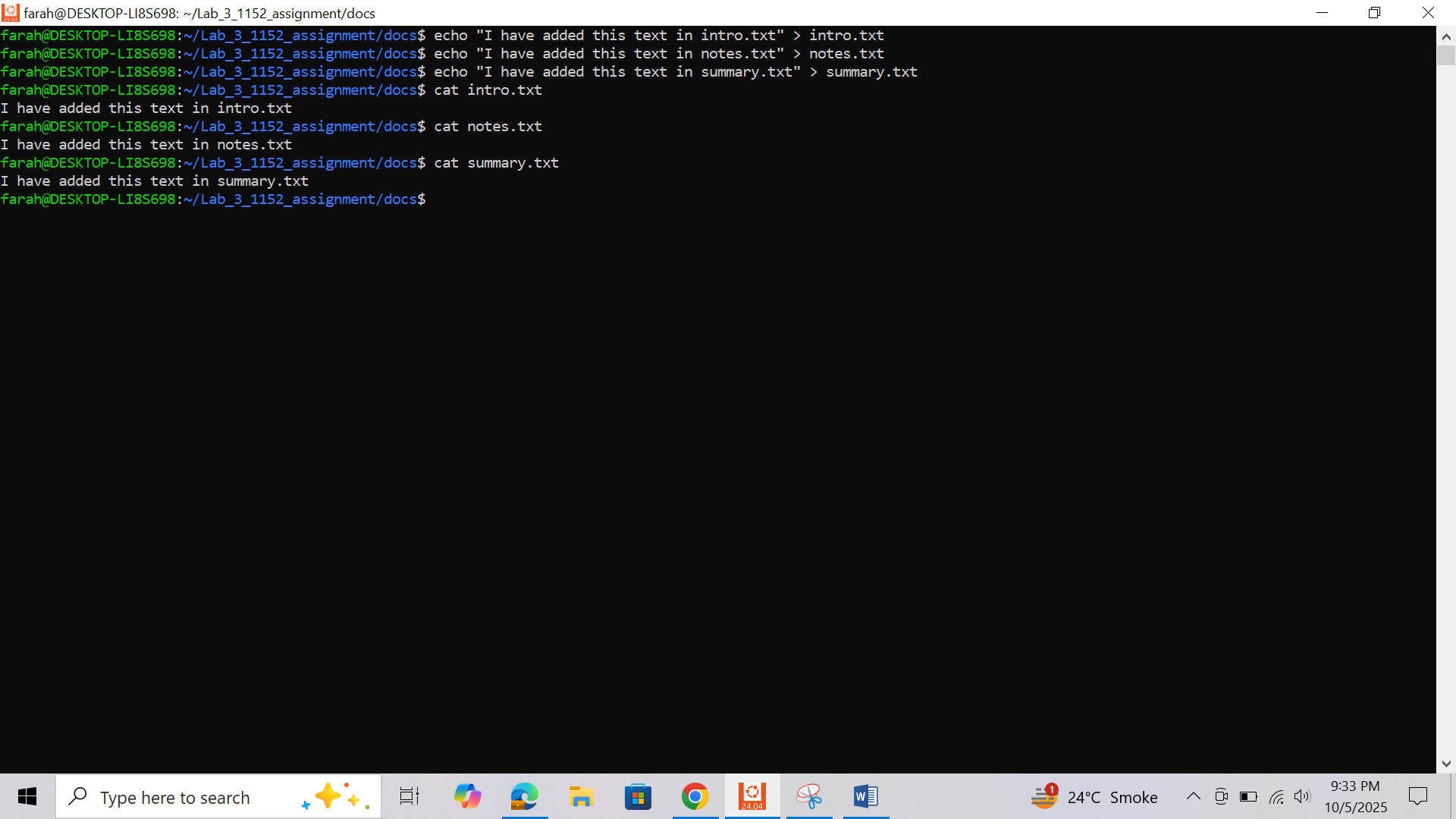
1. **Create directory structure:**



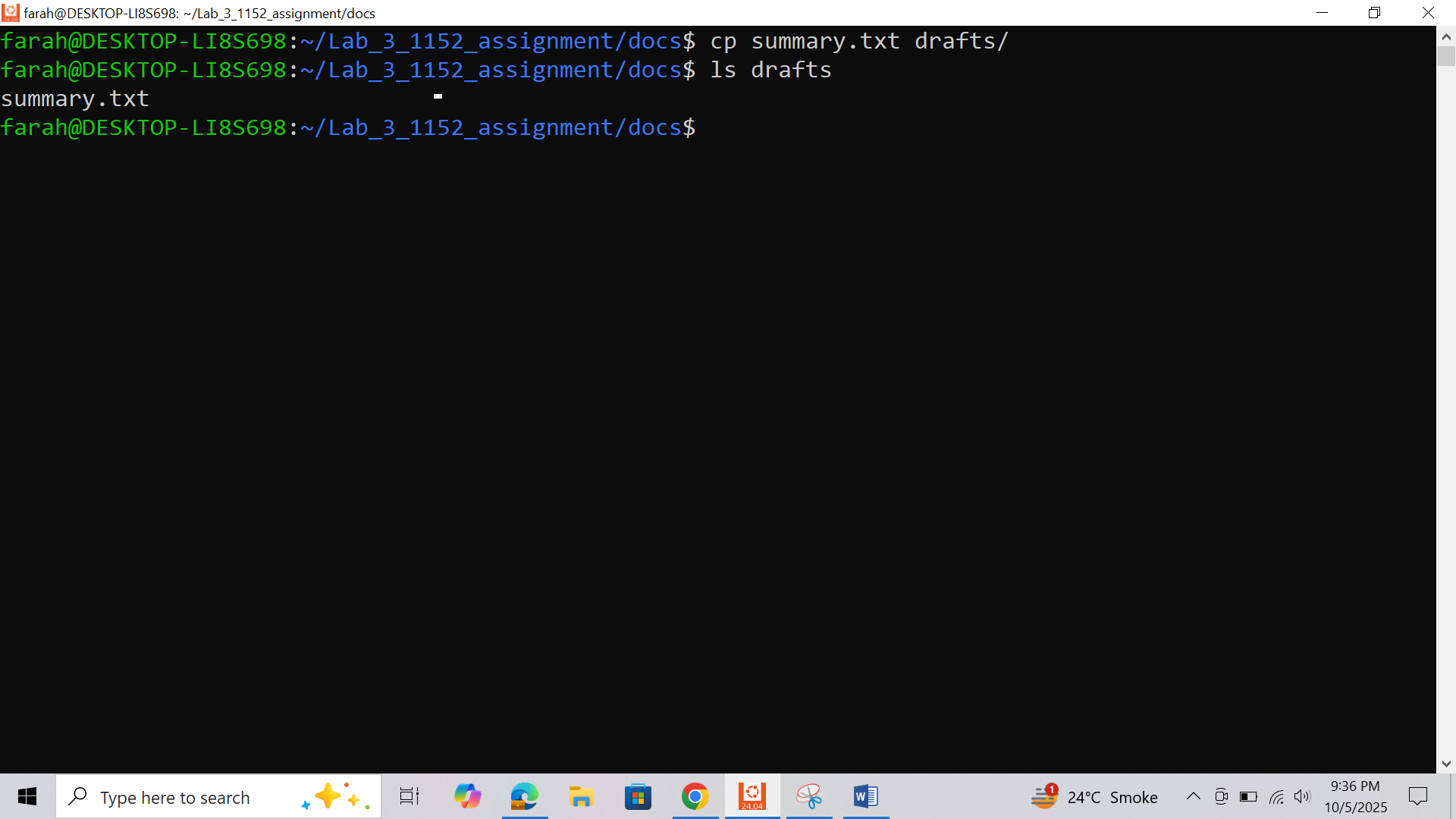
1. **Inside docs:**
   * 1. Create three files inside docs:



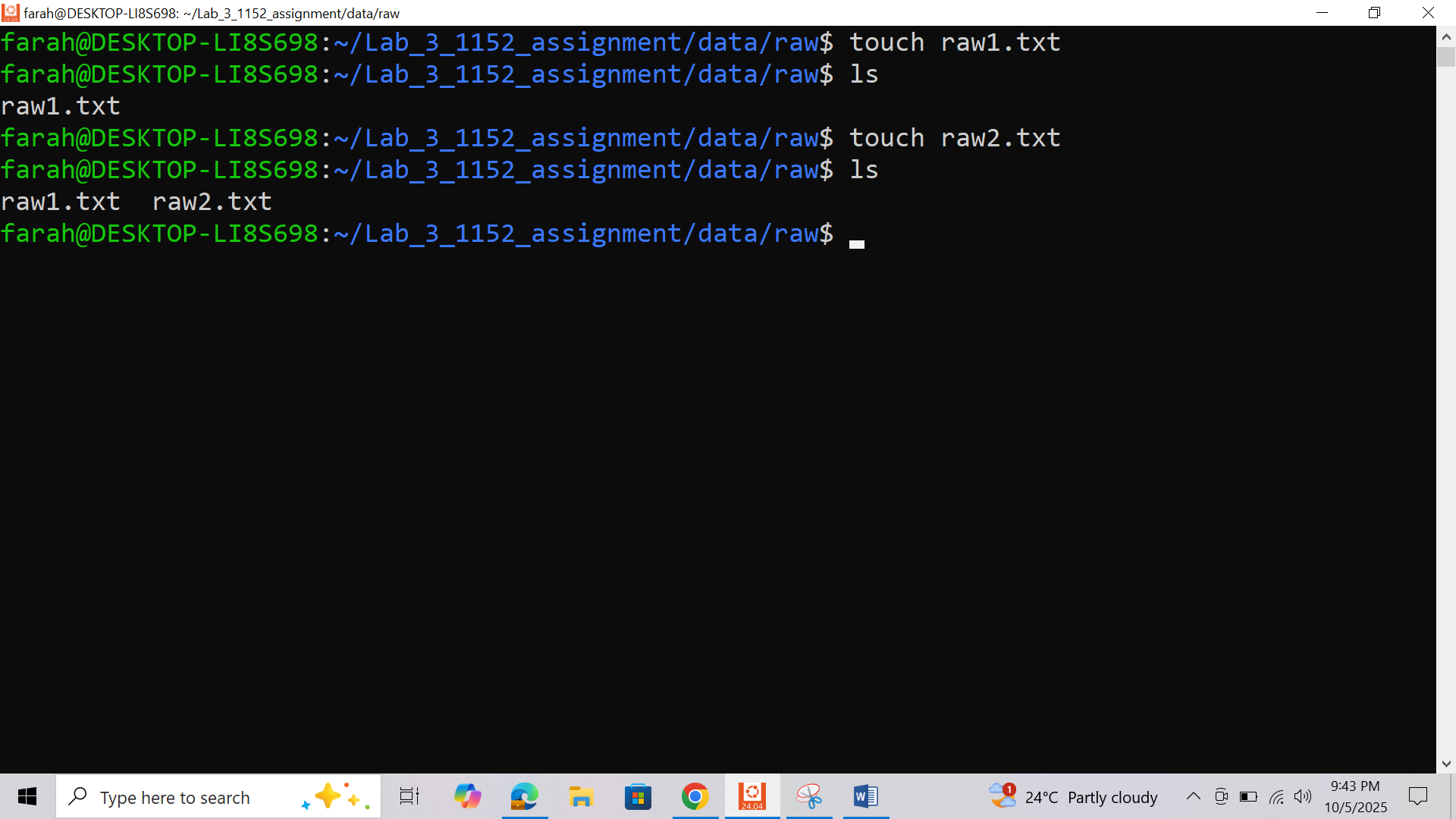
* + 1. Add at least two lines of text into each using echo command:



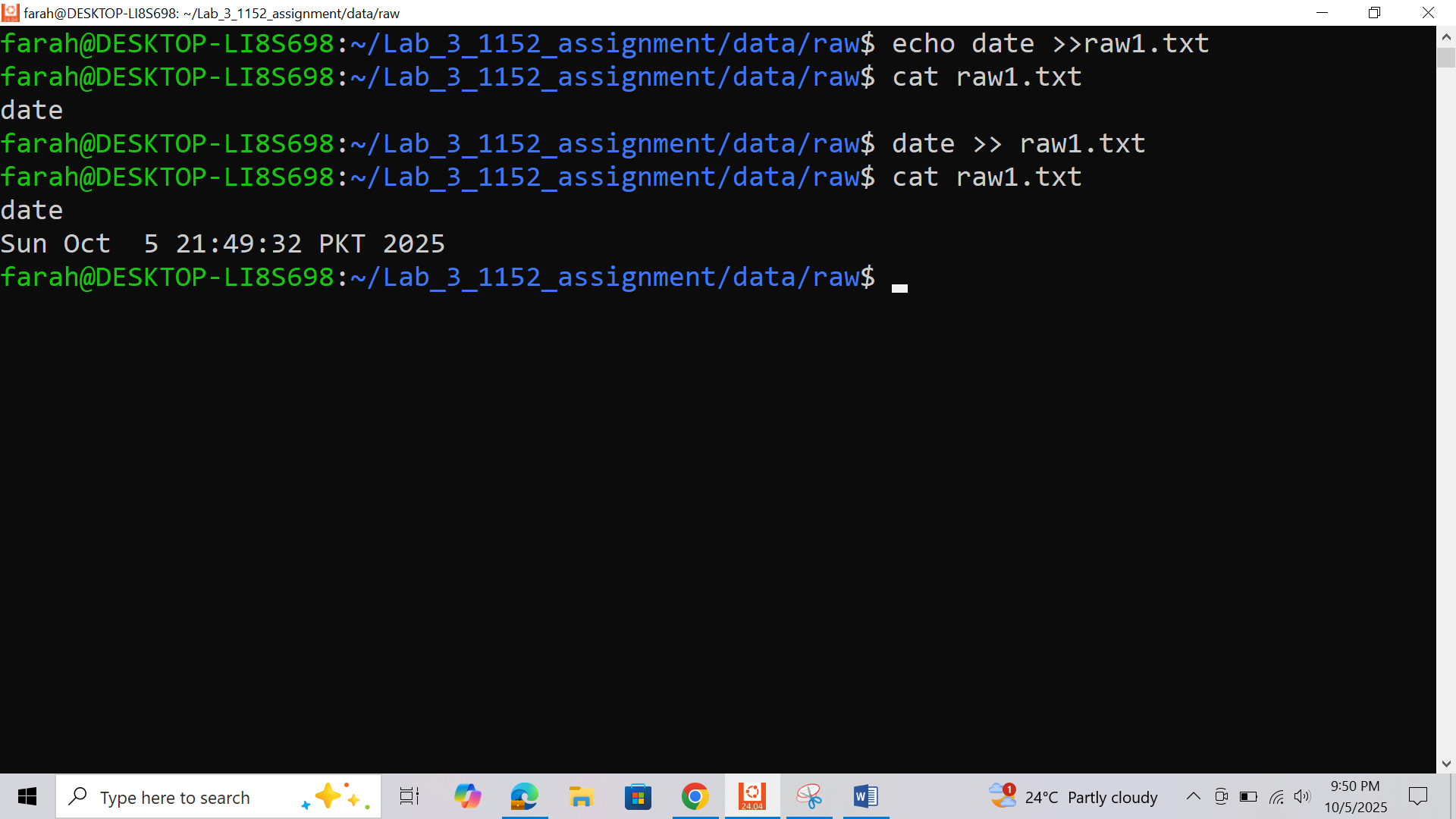
* + 1. Copy summary.txt in the docs folder:



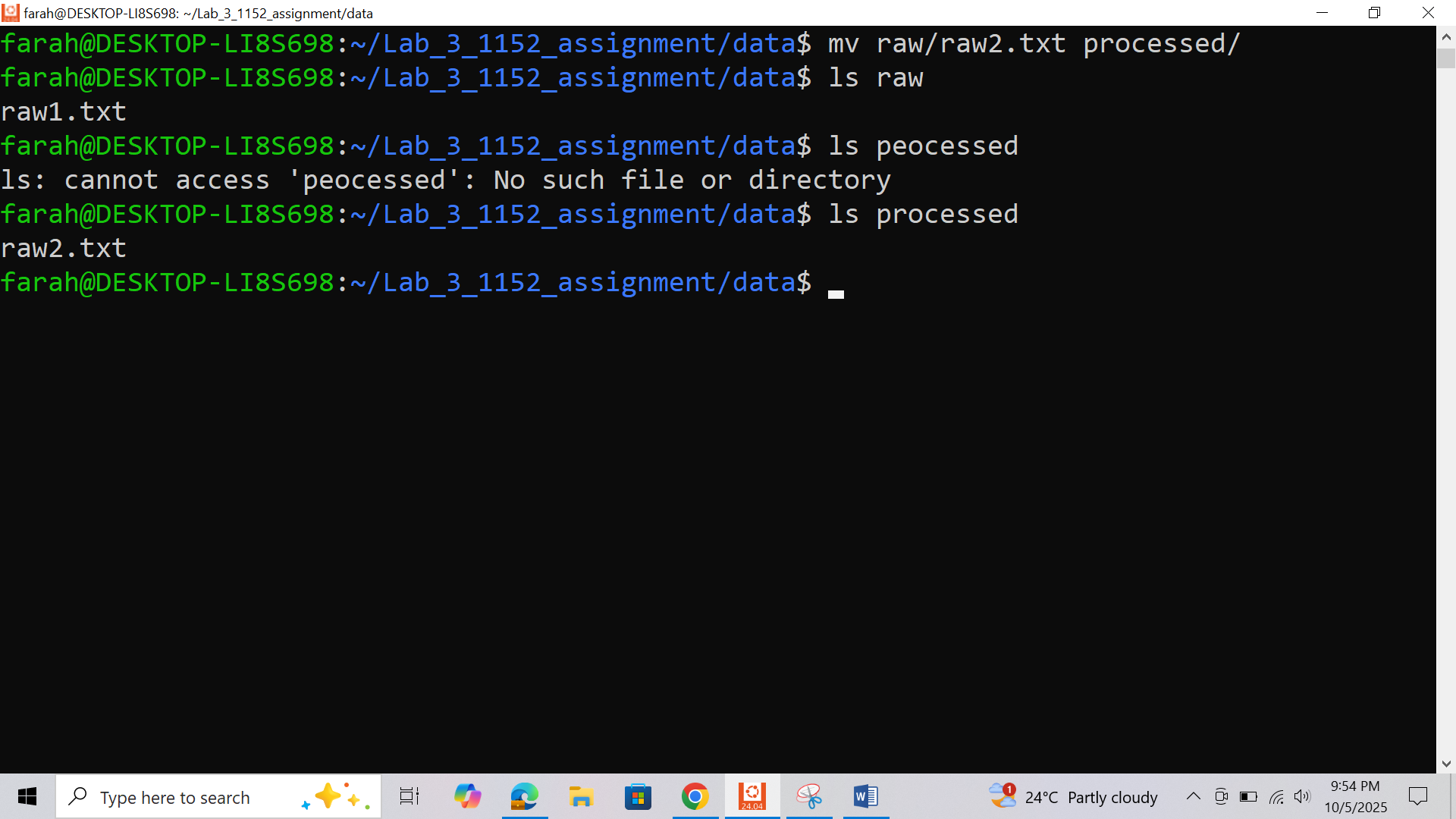
1. **Inside data/raw/:**
2. Create two files raw1.txt, raw2.txt:



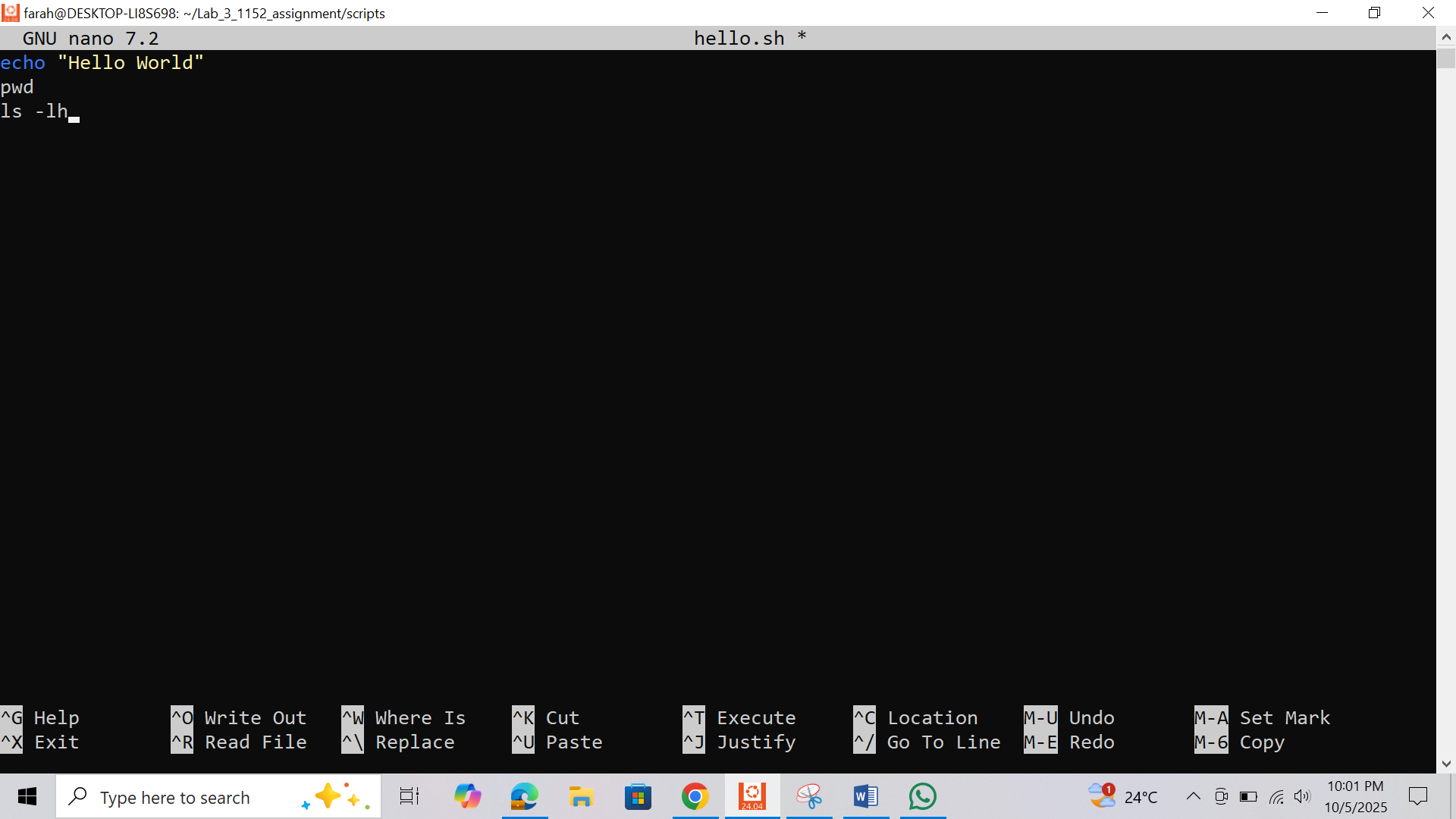
1. Append the current date into raw1.txt using date command



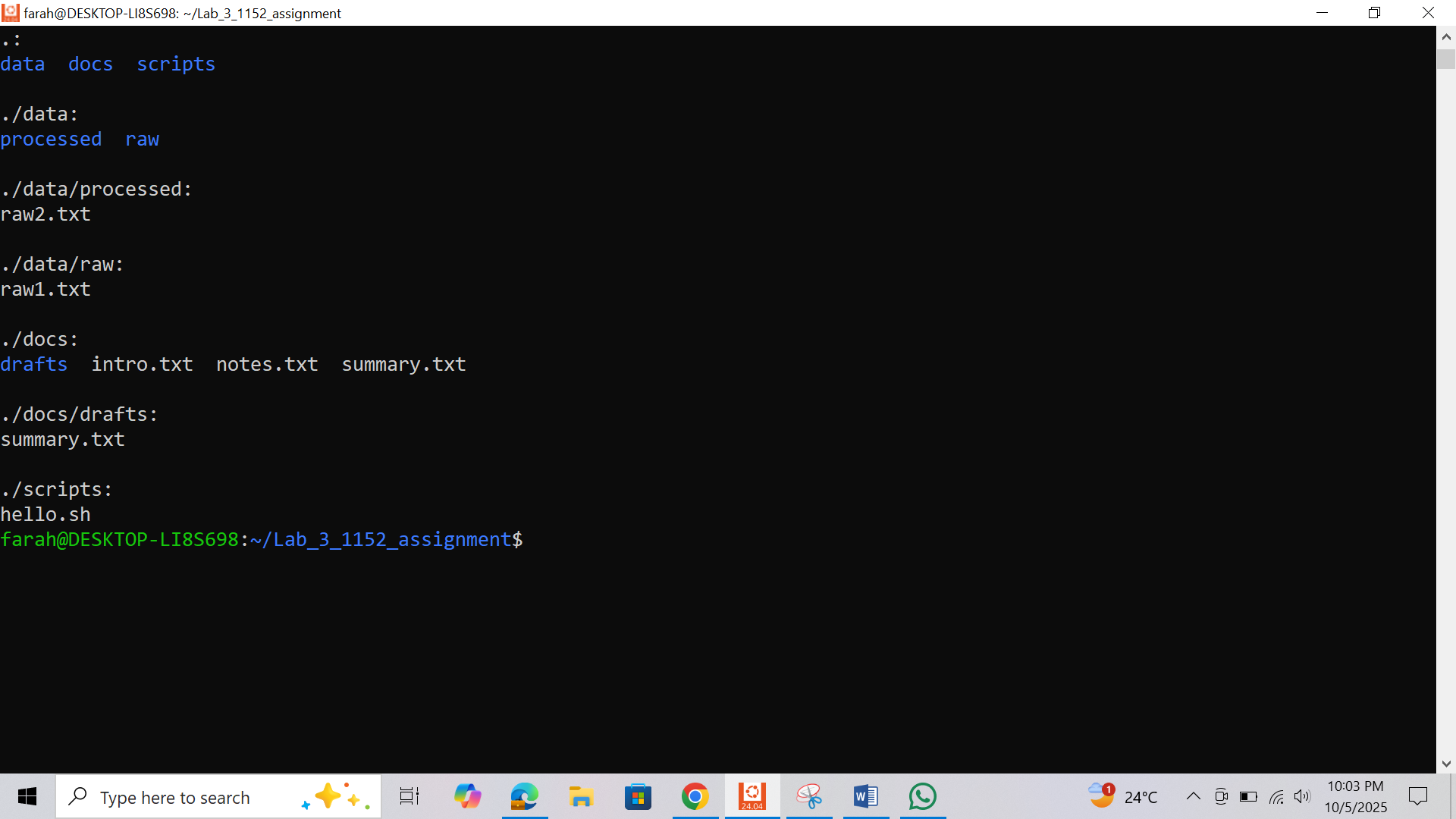
1. Move raw2.txt in processed using mv command:



1. **Inside scripts:**
2. Create a script named hello.sh:



1. **Display directory structure:**



# Part 2: Practice with Basic Linux Commands

Run the following commands inside Lab\_3/ and note their outputs:

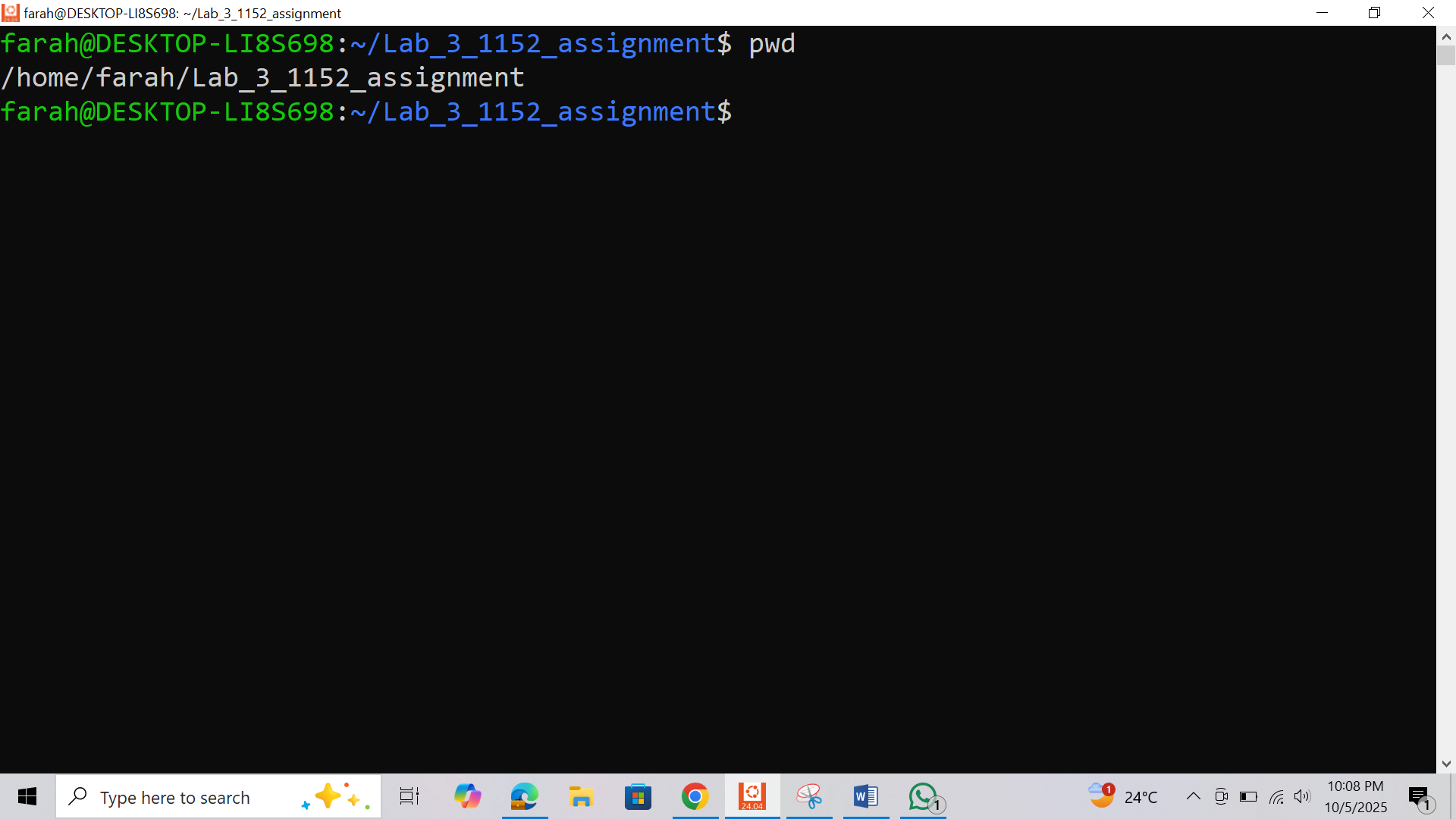
pwd → Show current working directory.

whoami → Display the current logged-in user. touch extra.txt → Create an empty file. cat intro.txt → Display file contents. rm extra.txt → Delete a file.

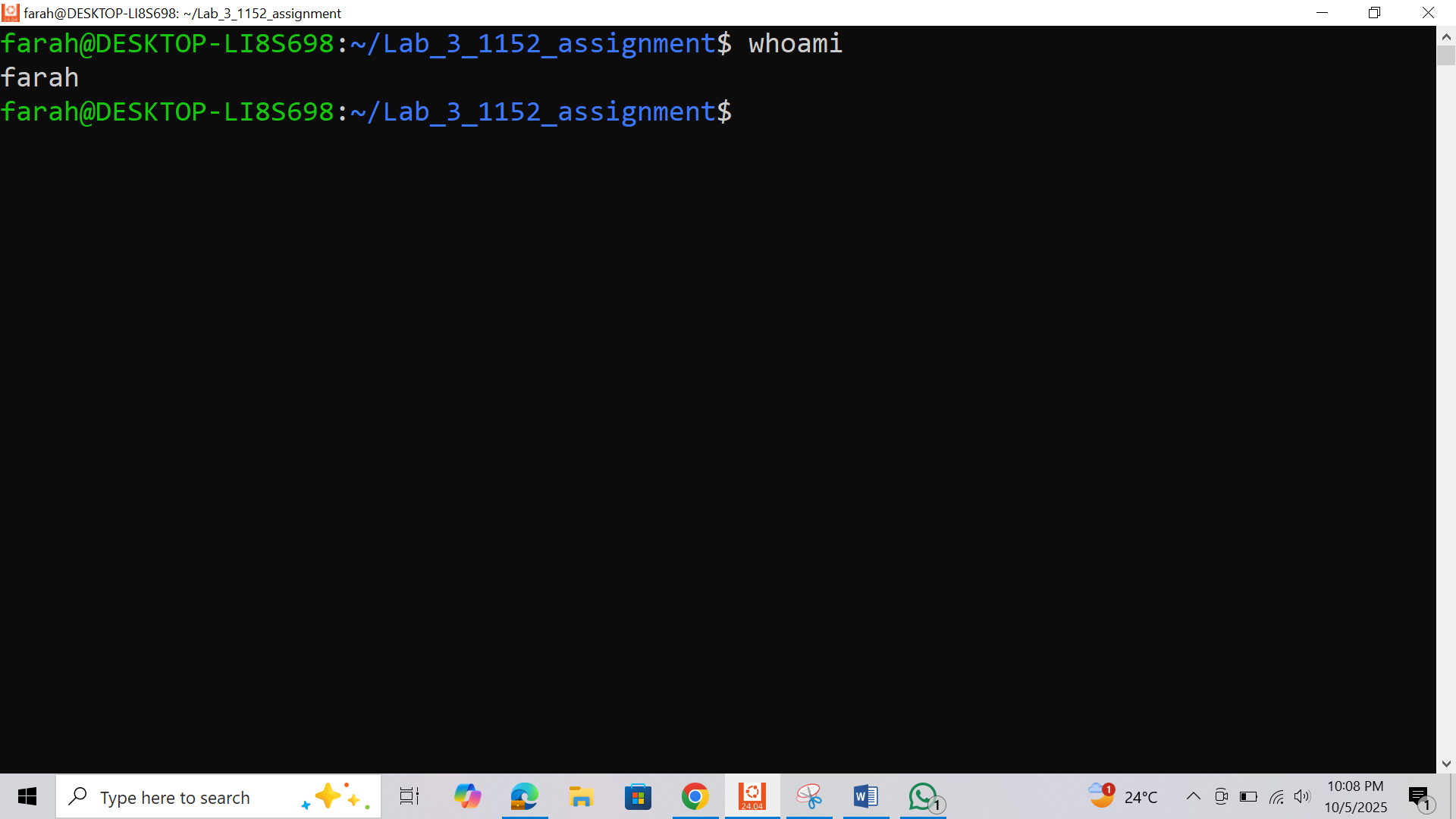
history | tail -n 5 → Show your last 5 executed commands. clear → Clear the terminal.

Take screenshots of commands and outputs.

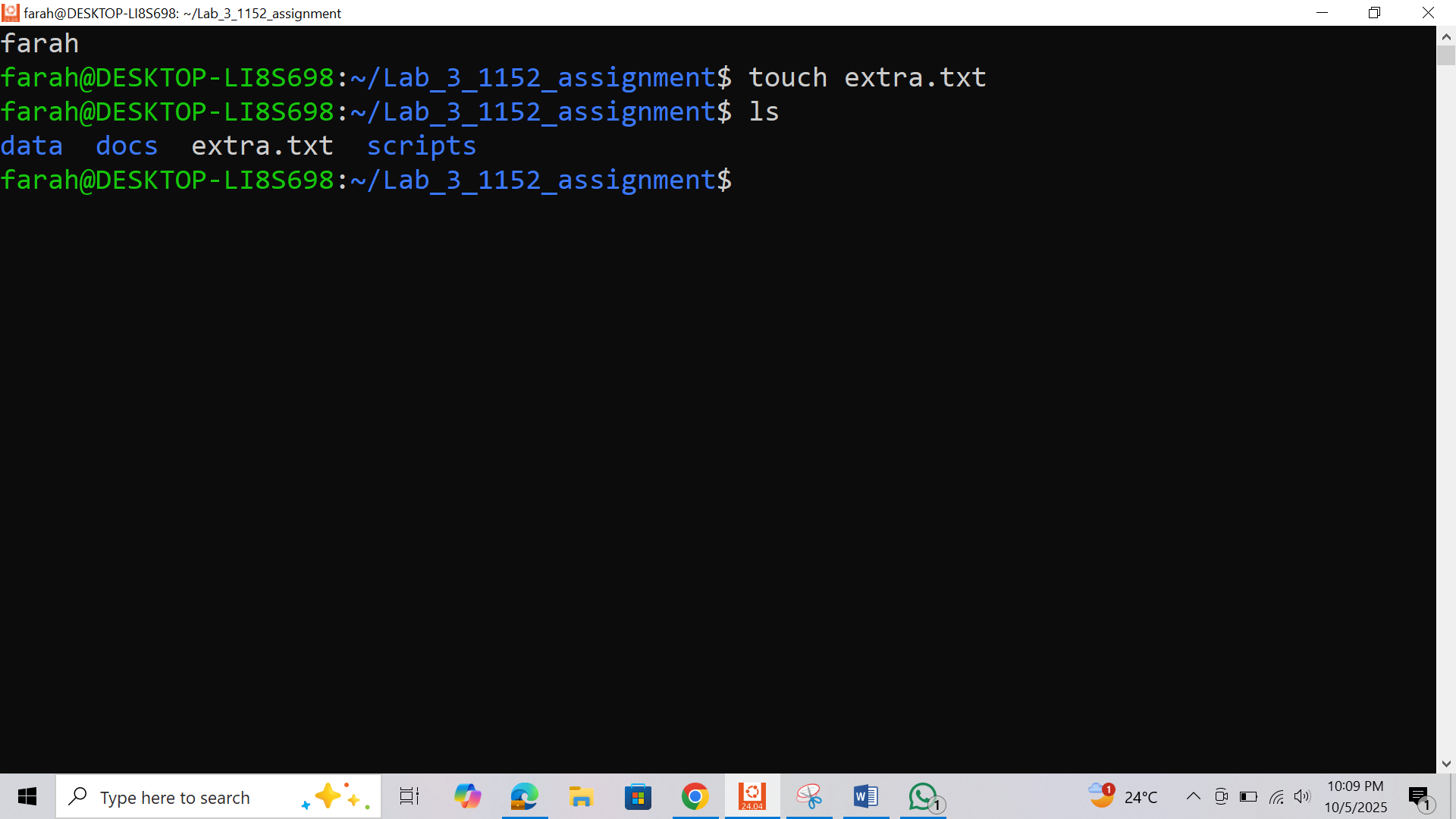
1. **Pwd:**



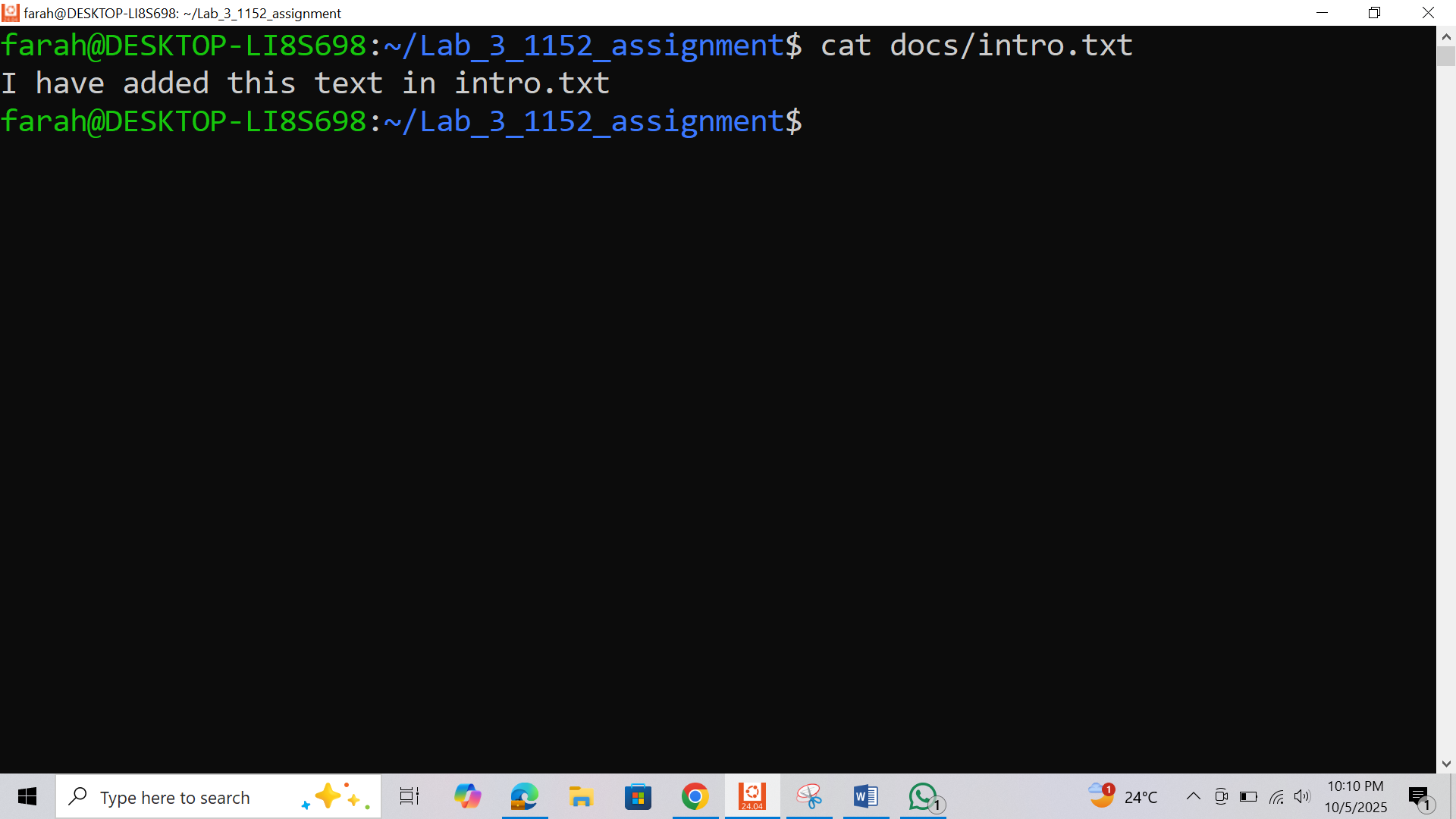
1. **Whoami:**



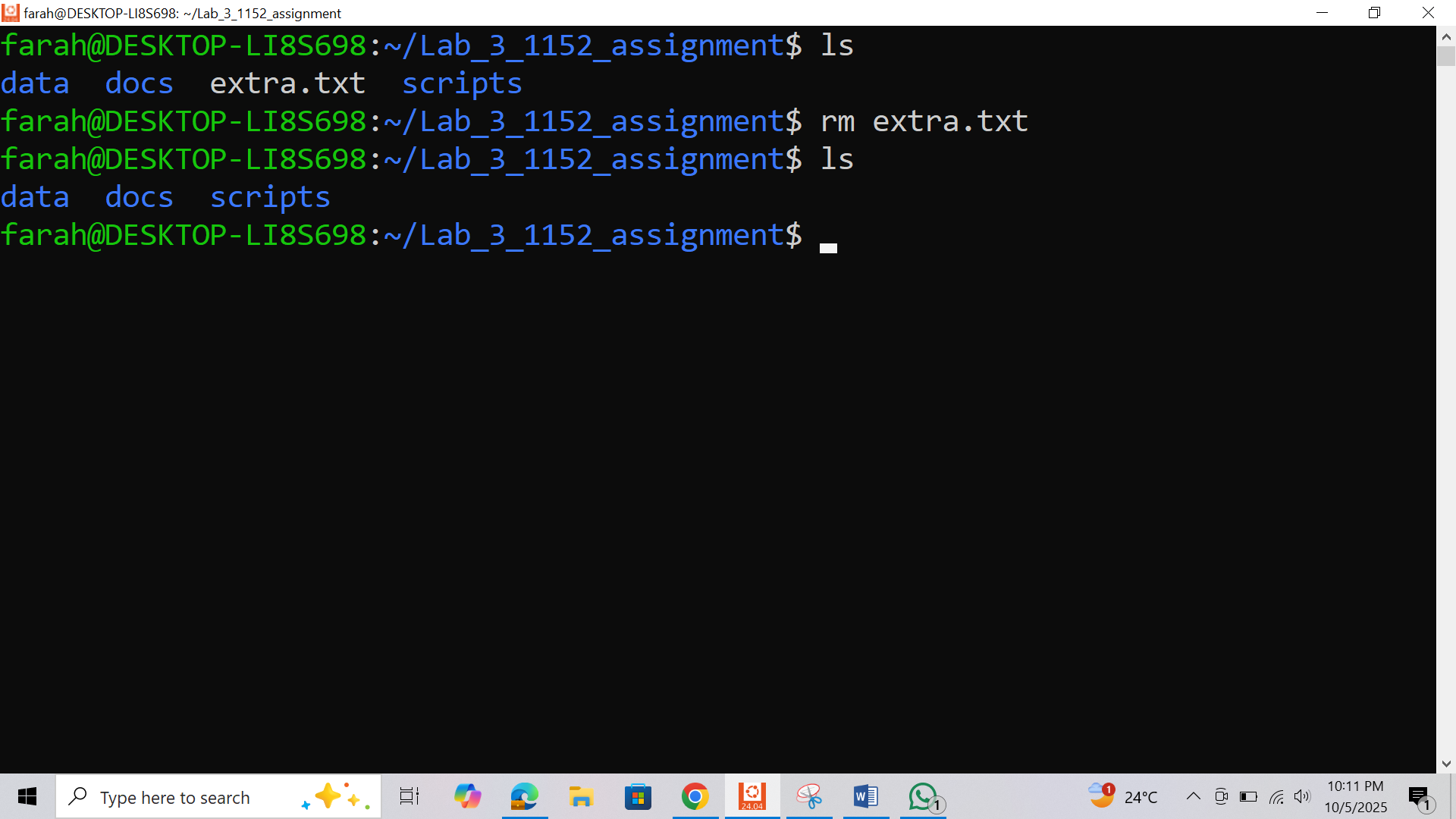
1. **Touch extra.txt**



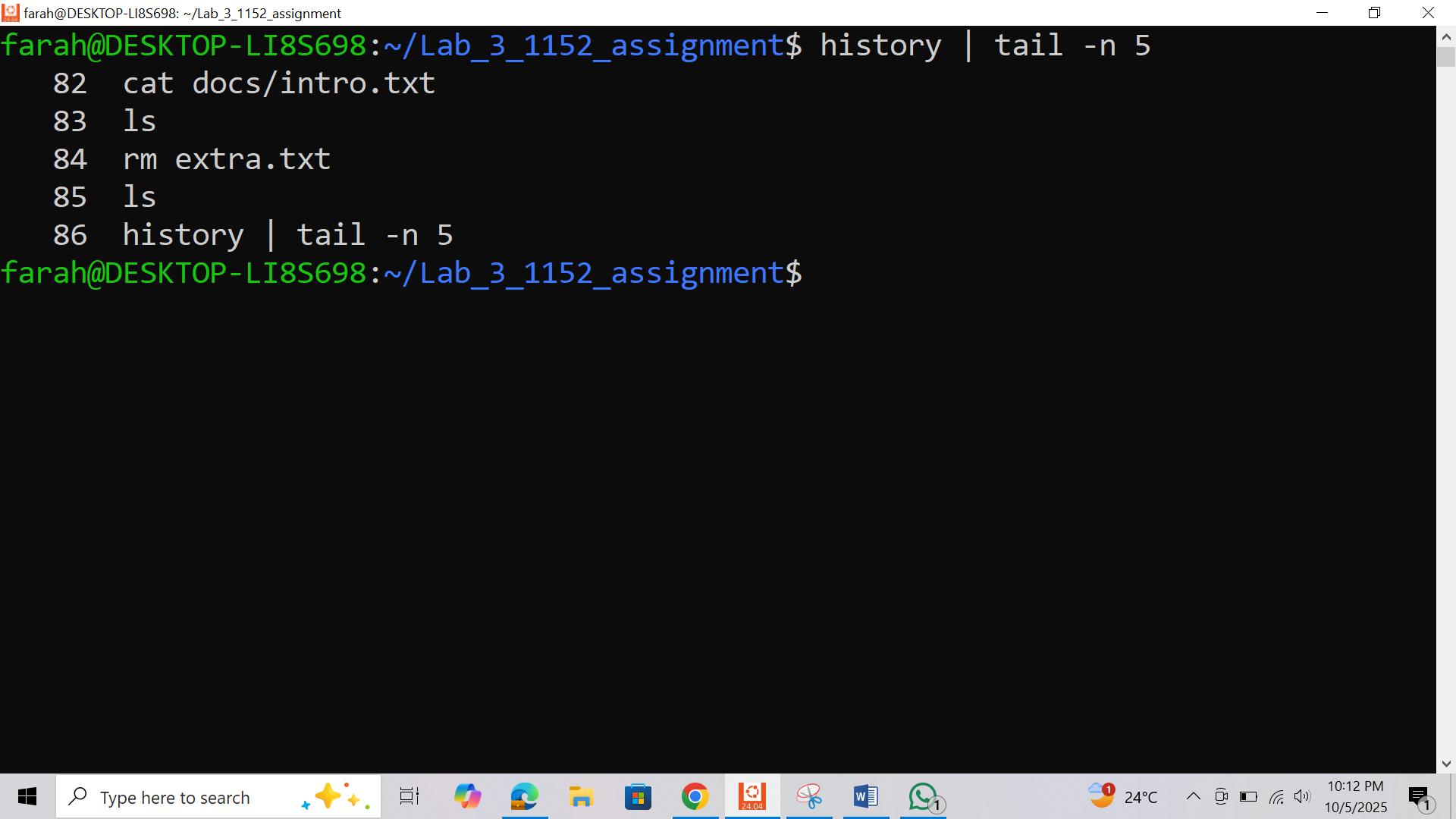
1. **Display intro.txt**



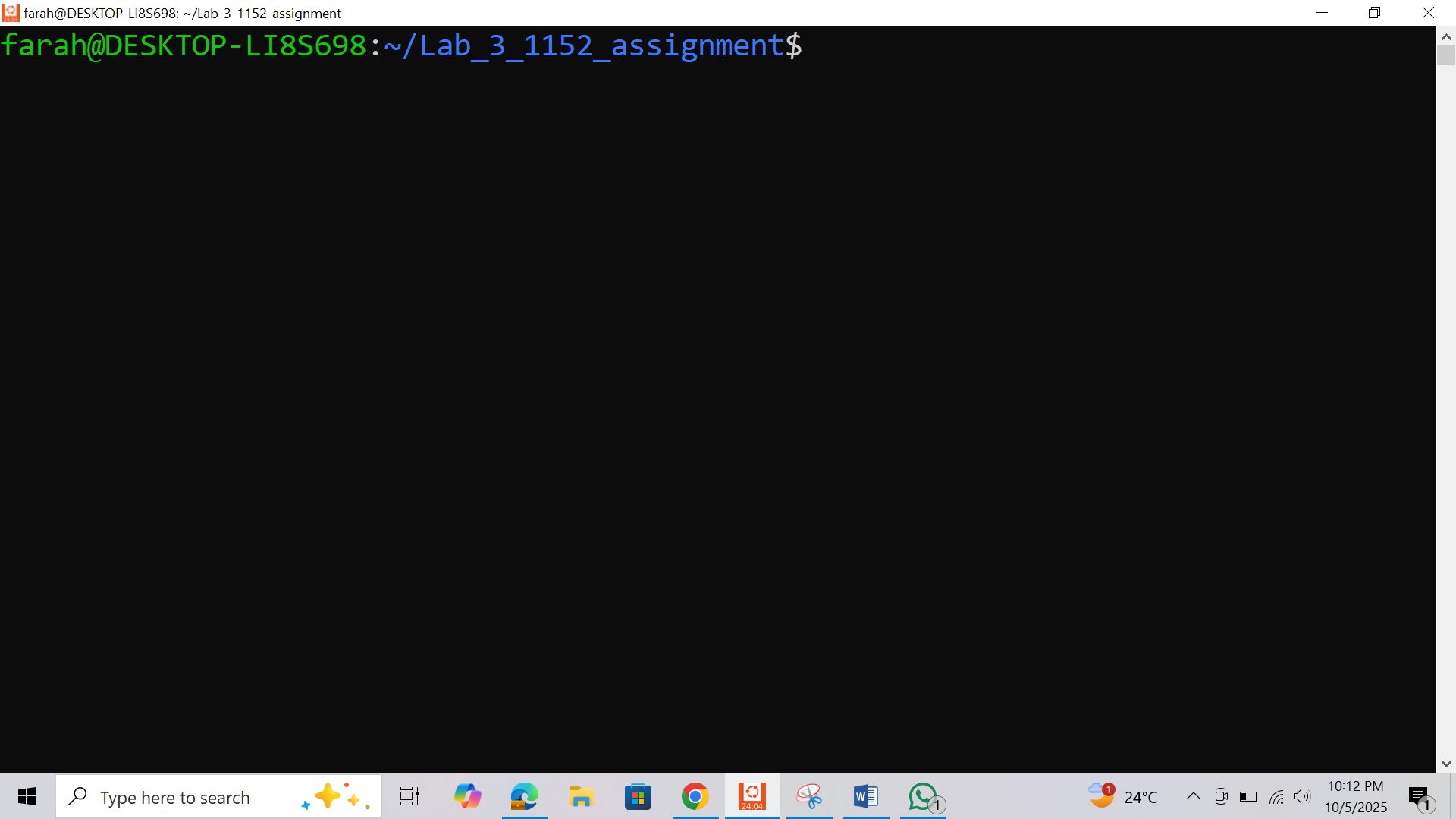
1. **Delete a file -> rm extra.txt:**



1. **history | tail -n 5 → Show your last 5 executed commands.**



1. **clear → Clear the terminal.**



# Part 3: File Permissions and Ownership

1

.

Change the permissions of

hello

.

sh

so that:

Owner → Read, Write & Execute

Group → Read, Write & Execute

Others → No permissions

Run the script using:

./

hello

.

sh

Take a screenshot of its output.

1. Change the permissions of intro.txt using **numeric notation** so that:

Owner → Read & Write

Group → Read & Write

Others → Read only

1. Change the permissions of notes.txt using **symbolic notation** so that others don't have any permission on it.
2. Verify all changes with:

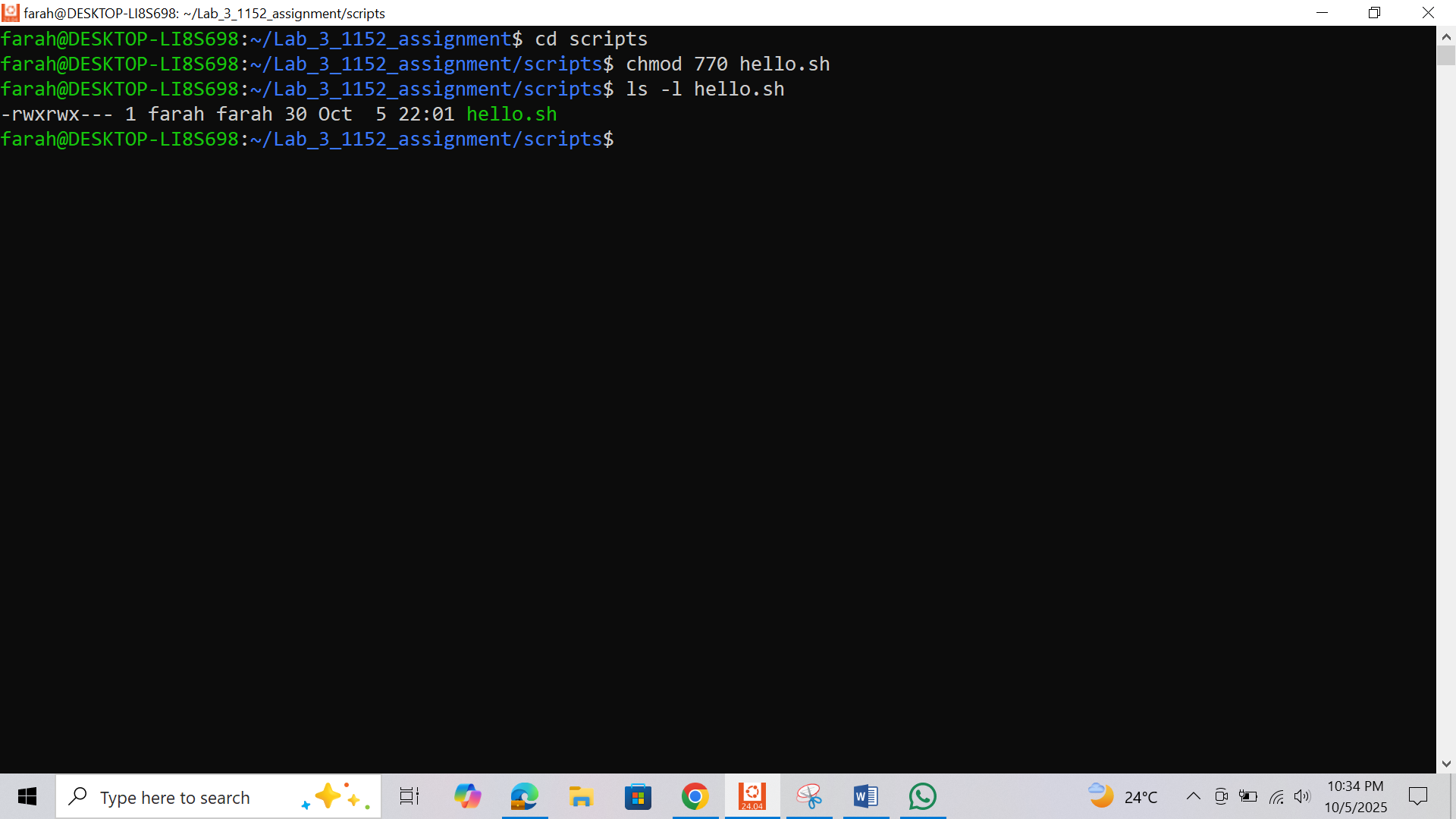
ls

-

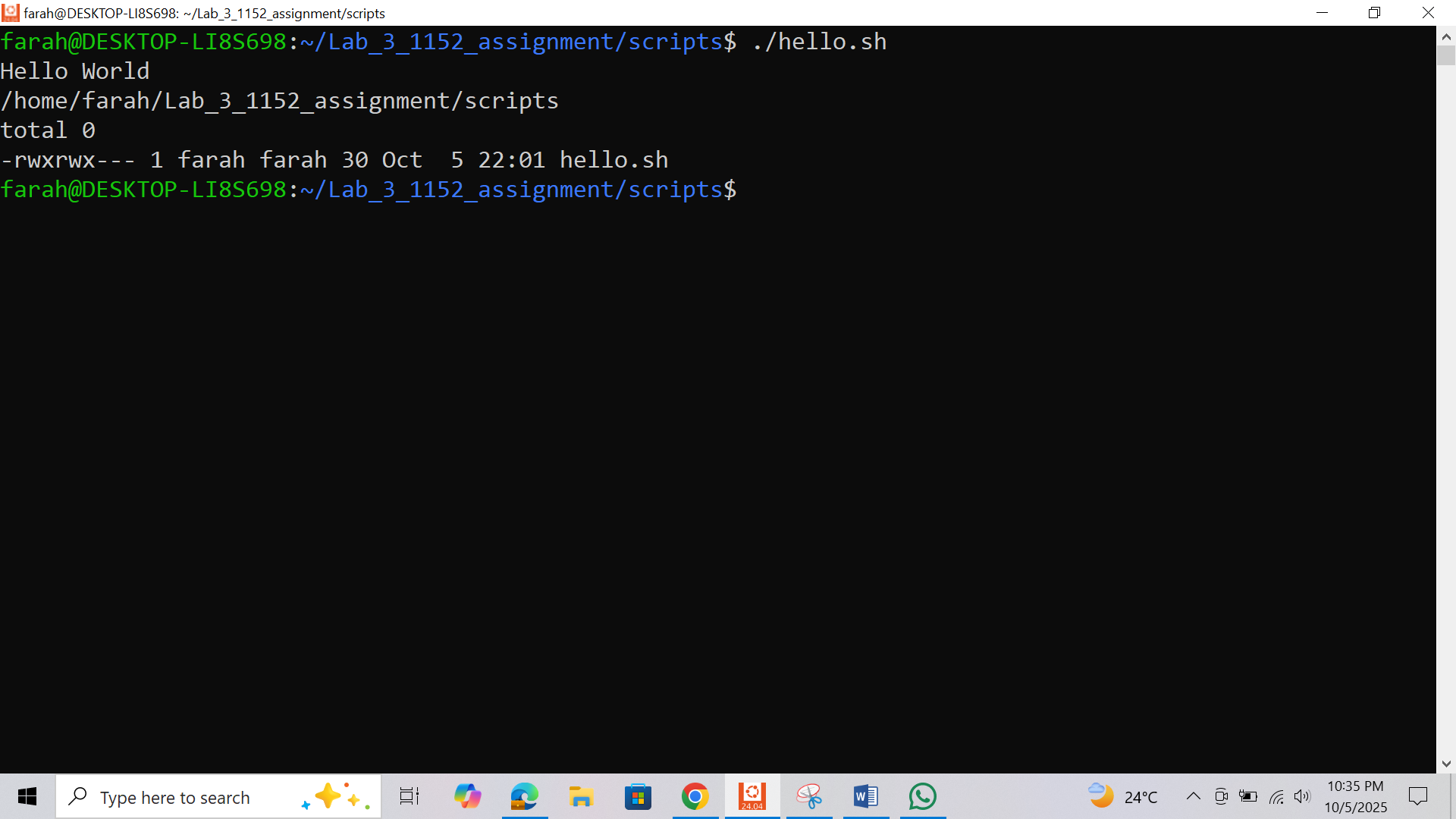
l

Take a screenshot of the output.

**Change the permissions of hello.sh such that:**



Run the script file:

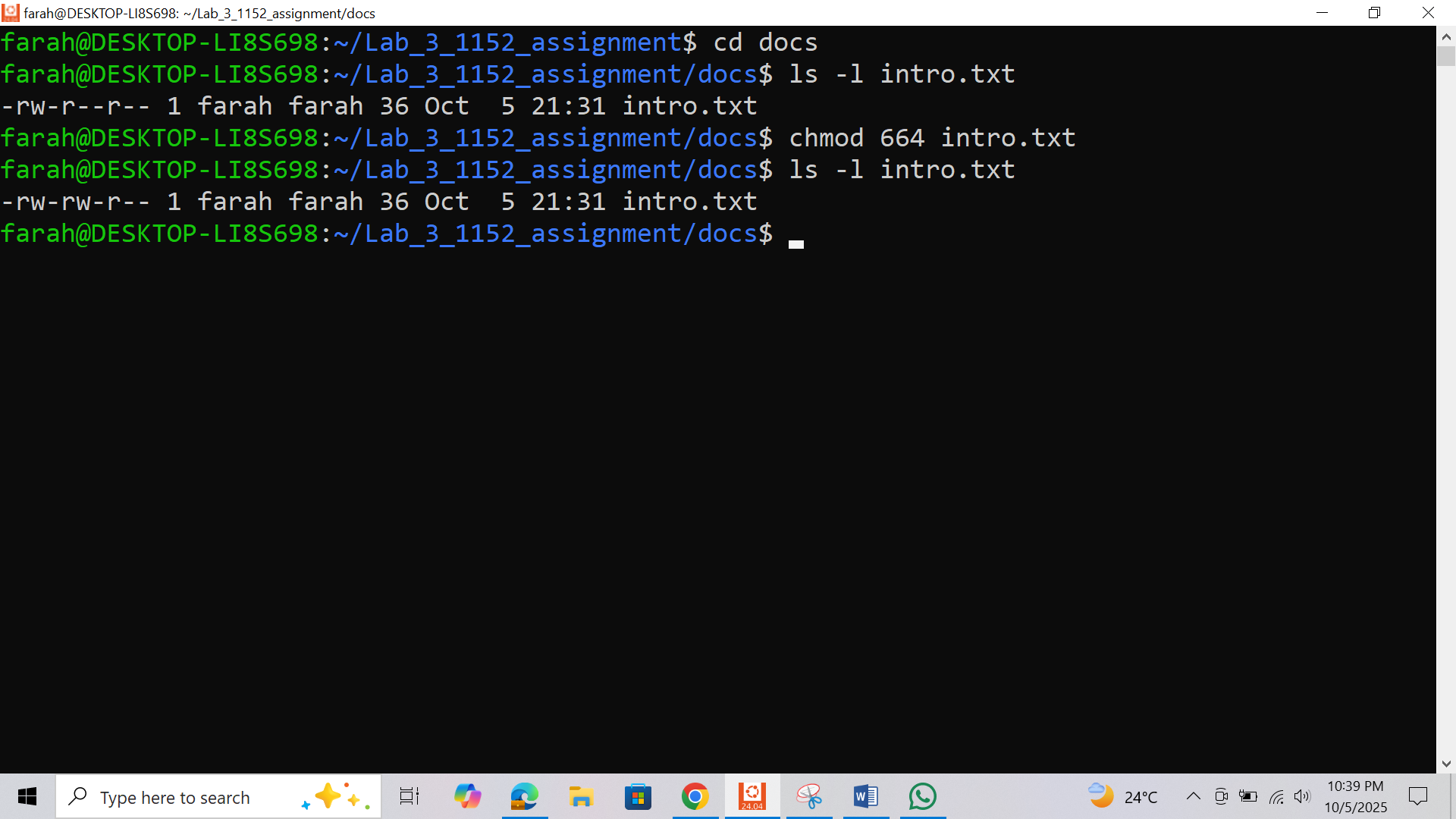


**Change the permissions of intro.txt using numeric notation so that:**

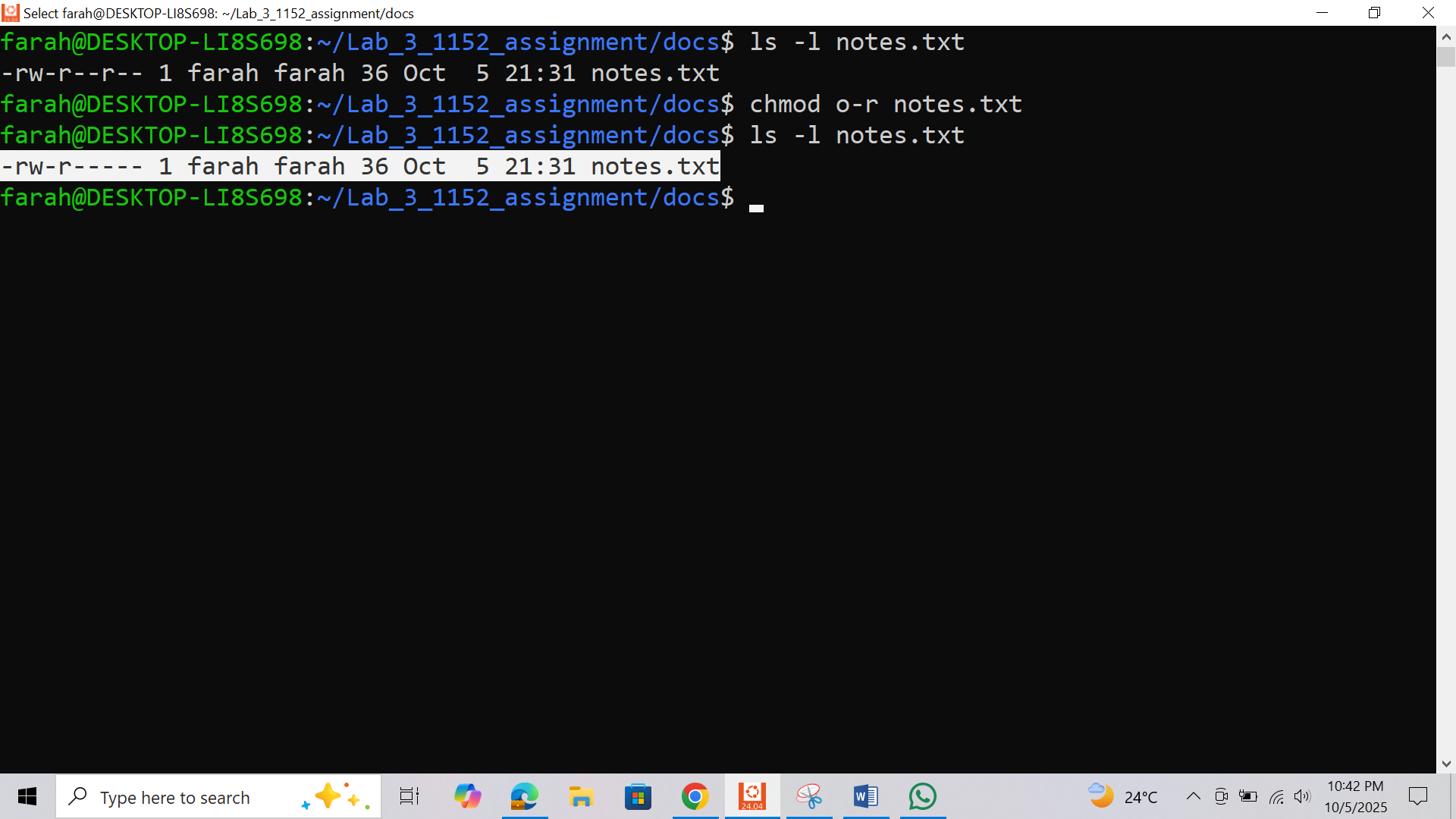
Owner → Read & Write

Group → Read & Write

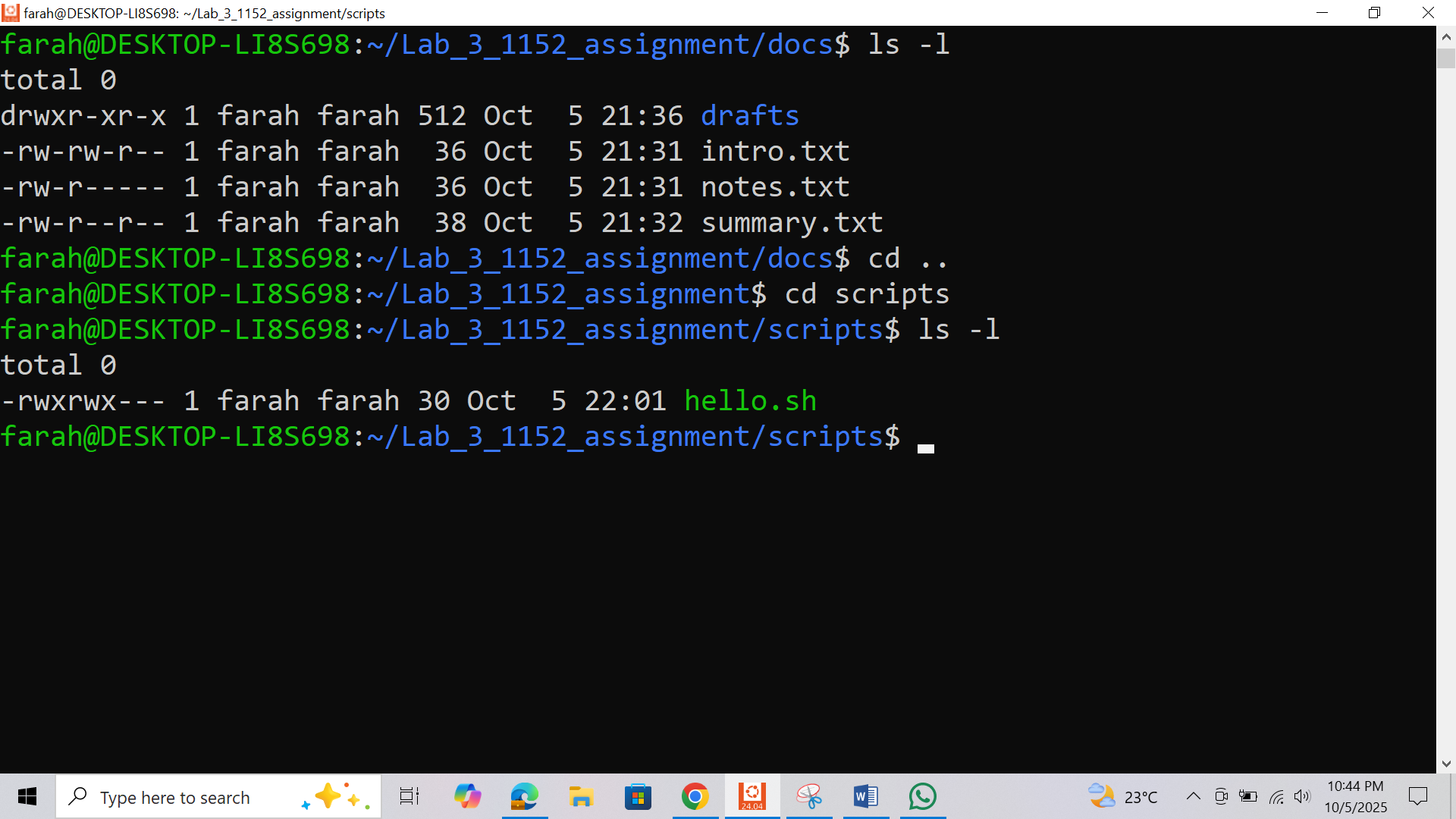
Others → Read only



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



**Verify all changes with: ls -l**



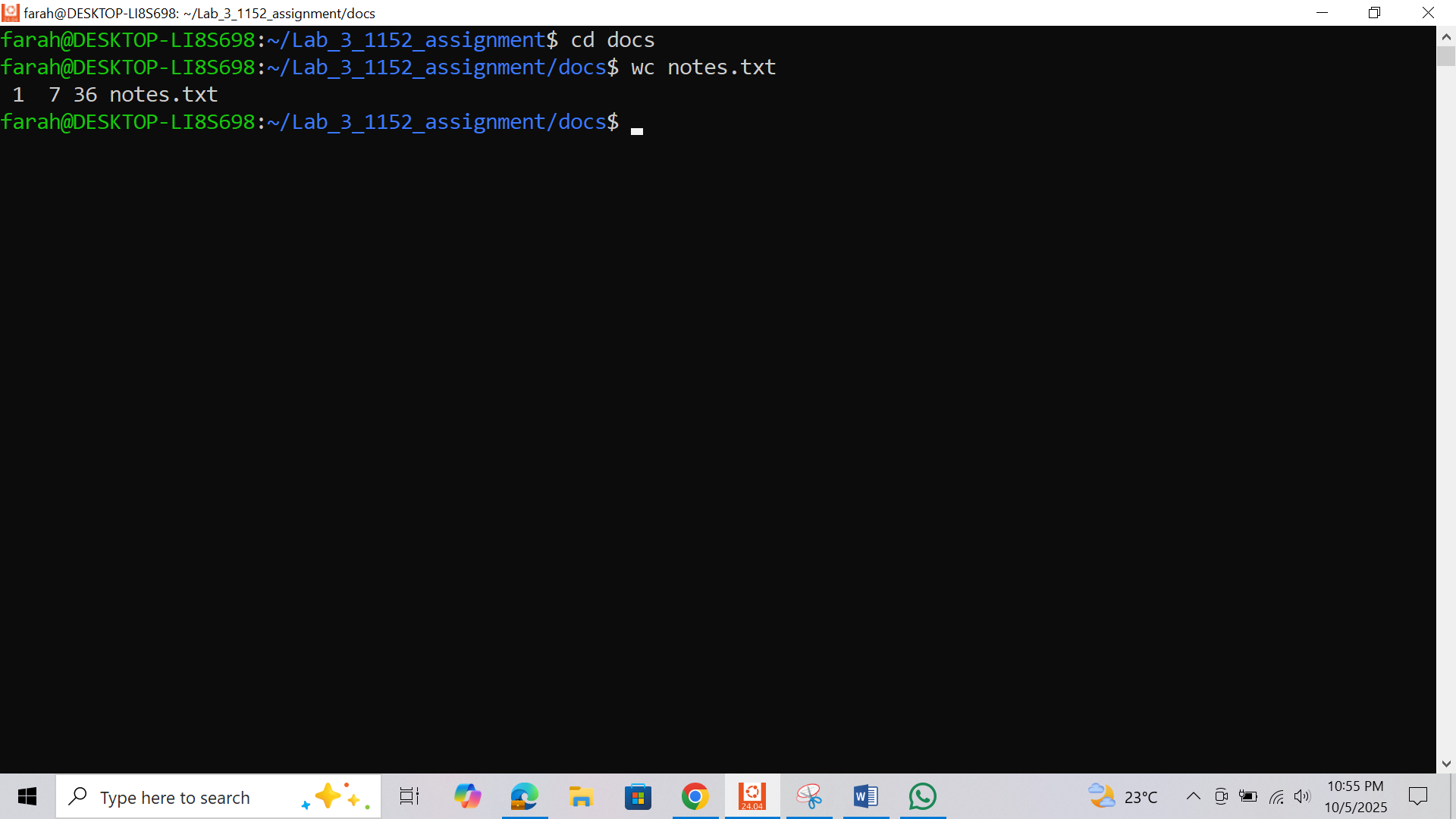
# Part 4: Reading & Searching Files

Inside docs/ :

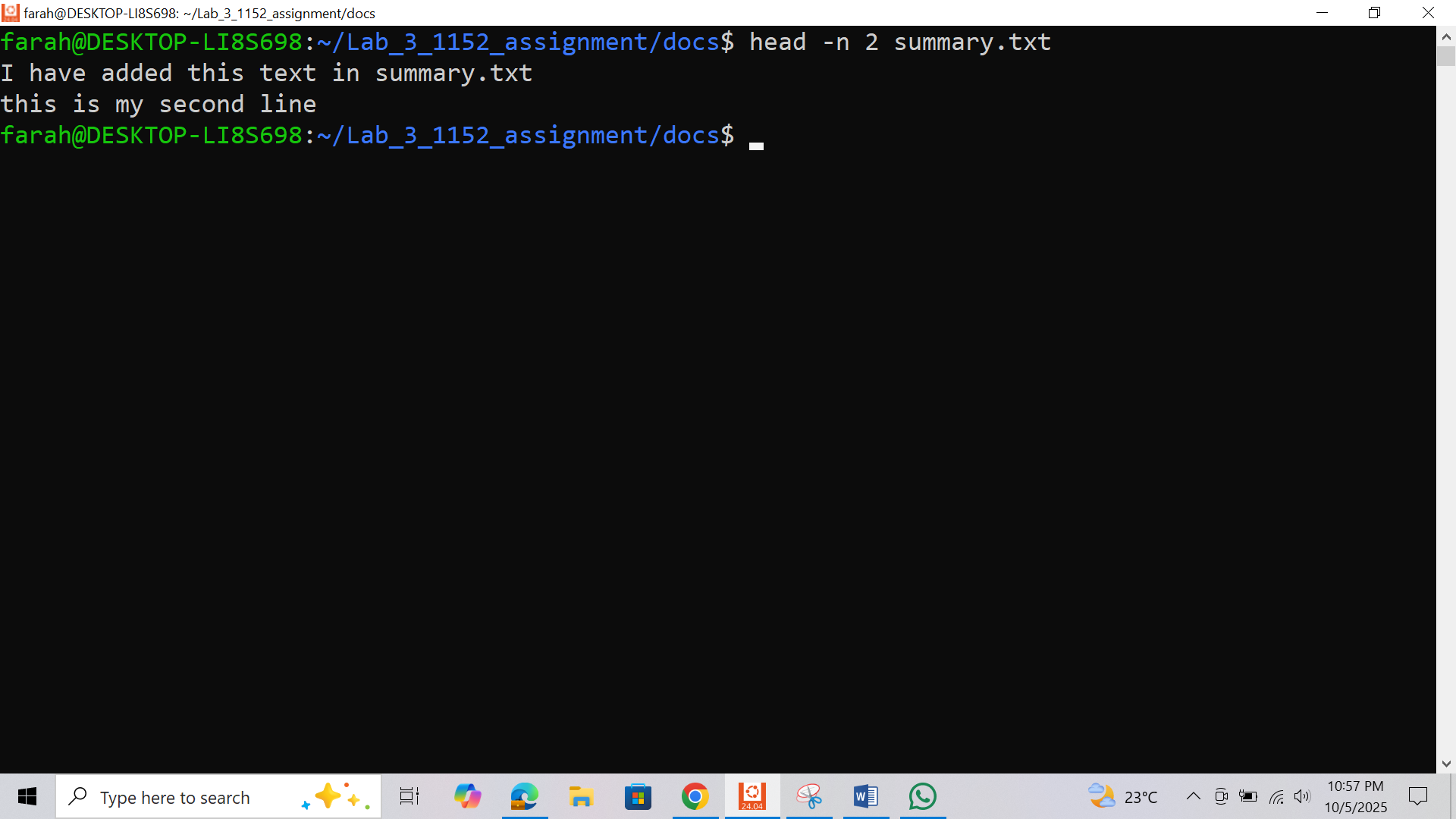
1. Count the number of lines, words, and characters in notes.txt using wc .
2. Show only the **first 2 lines** of summary.txt using head -n 2 .
3. Show the **last line** of summary.txt using tail -n 1 .
4. Search for a keyword (of your choice) in intro.txt using grep .

Take screenshots.

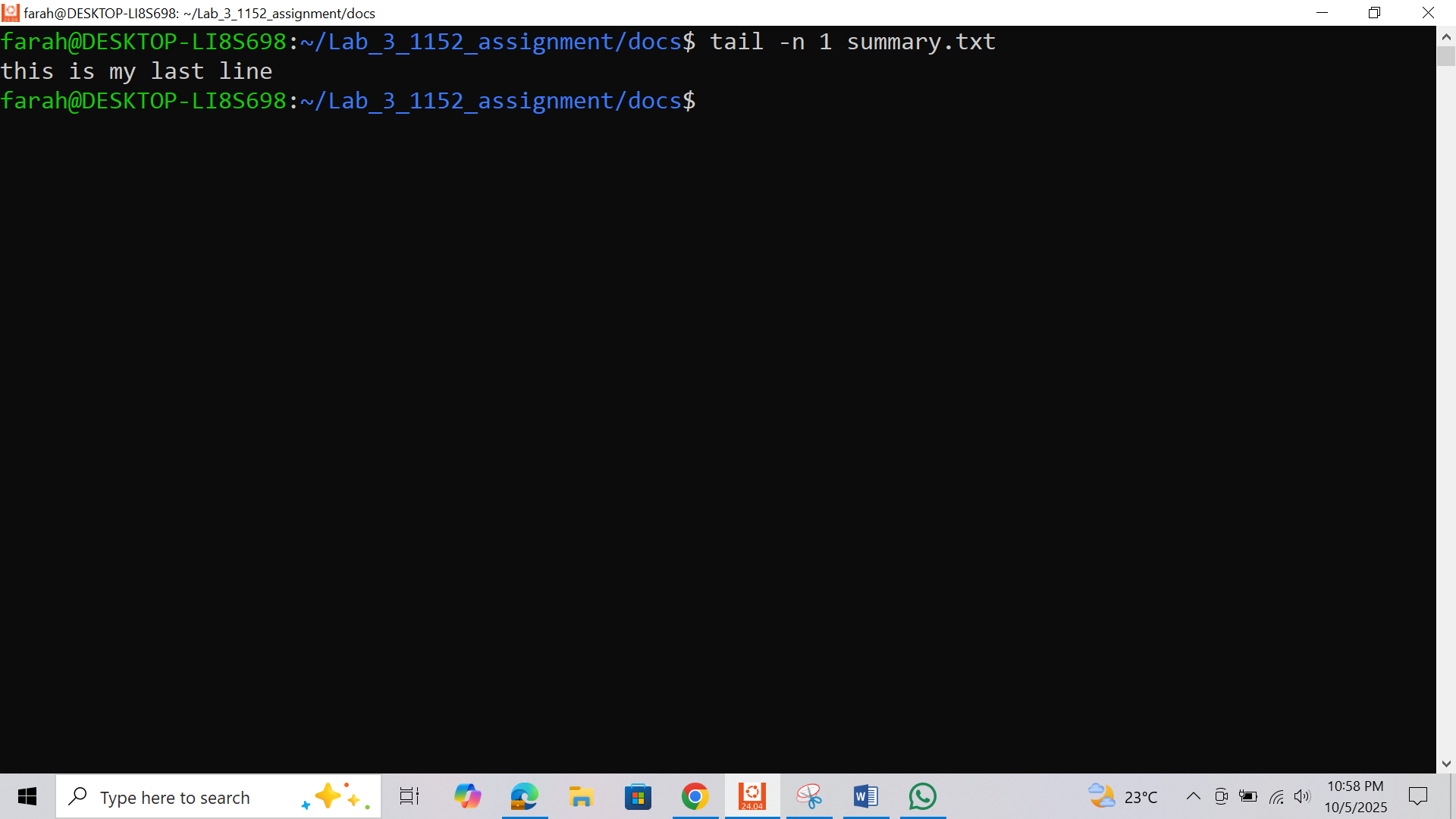
1. Count the number of lines, words, and characters in notes.txt using wc



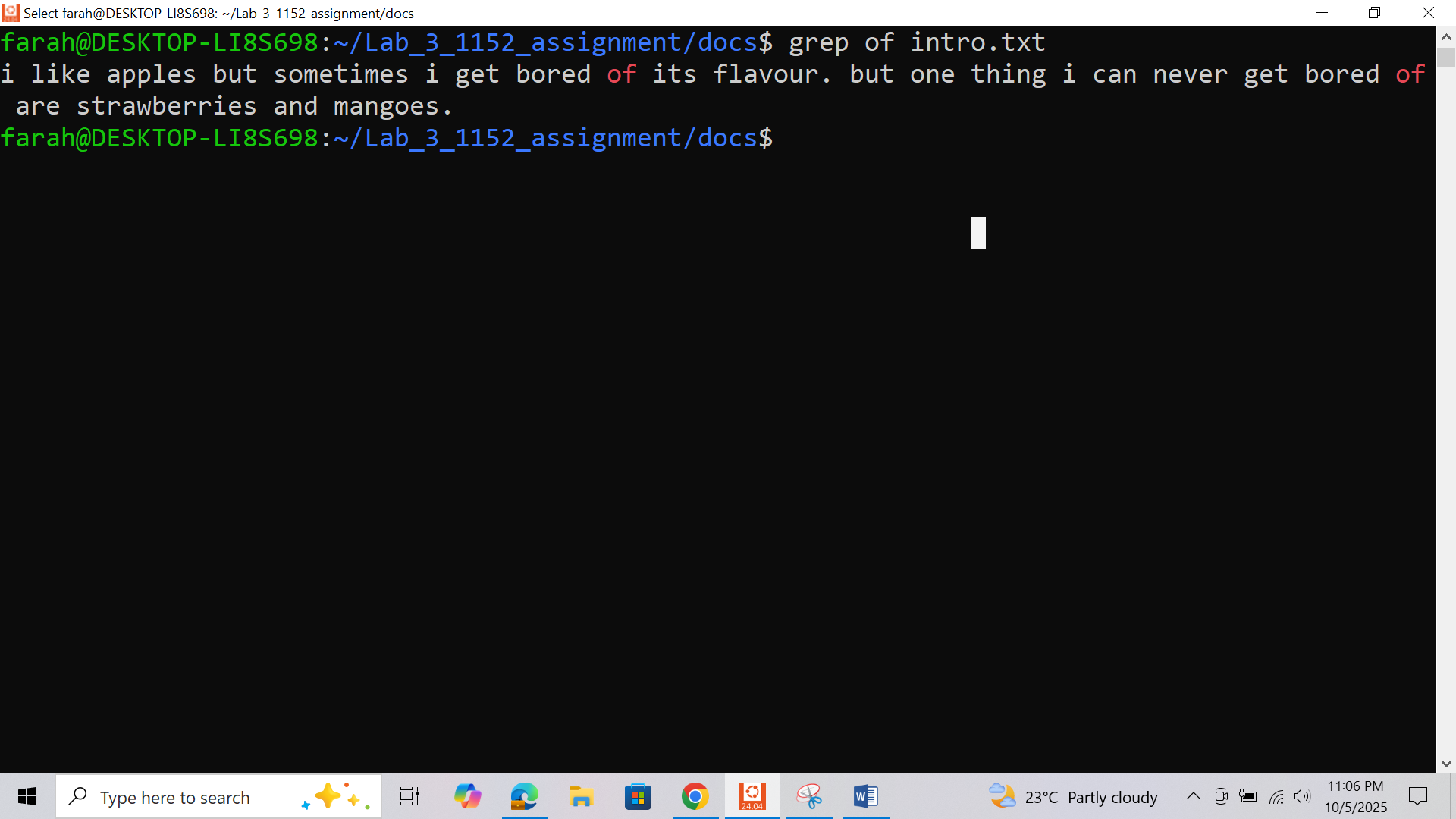
1. Show only the **first 2 lines** of summary.txt using head -n 2 .



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



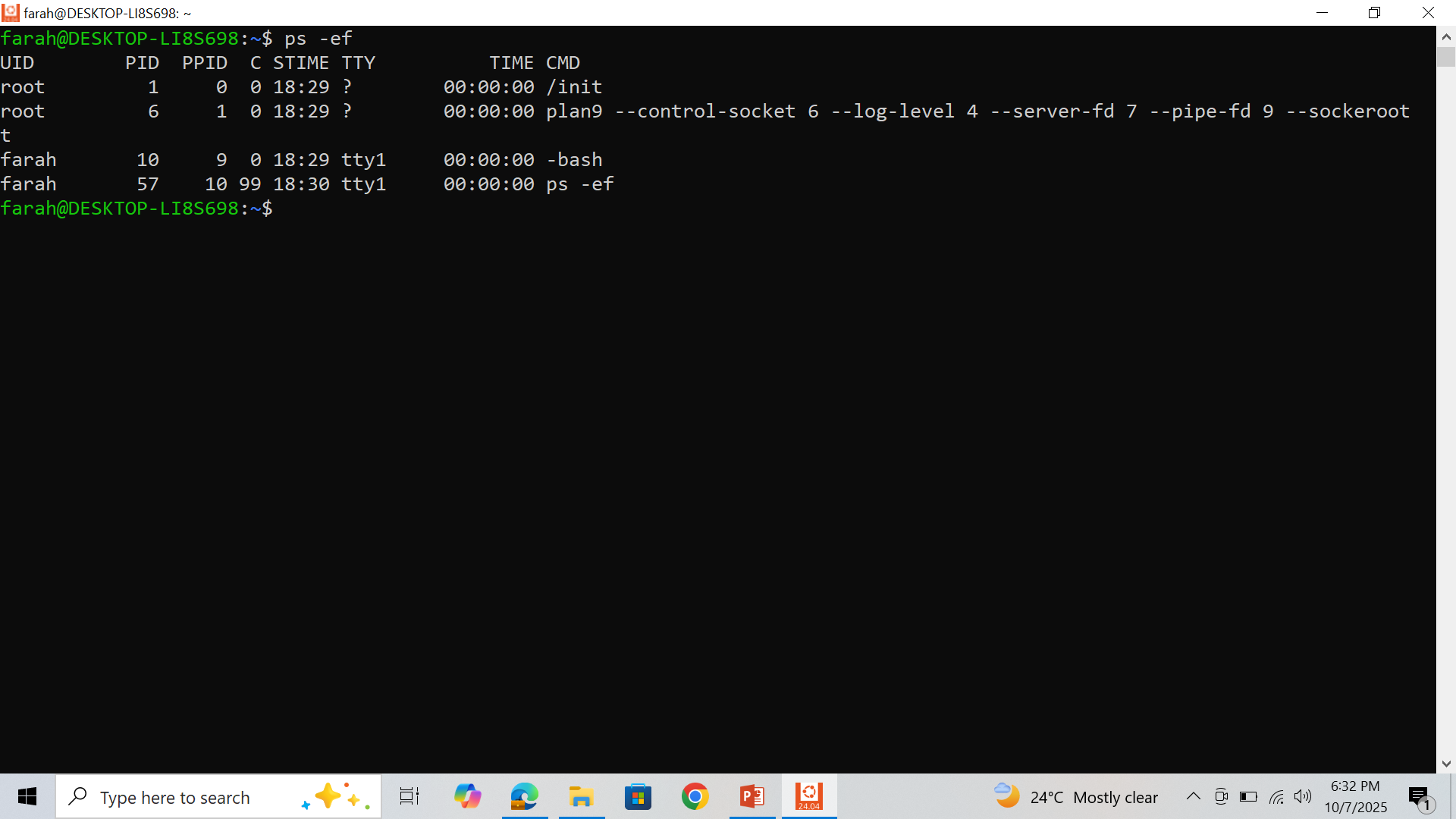
1. Search for a keyword (of your choice) in intro.txt using grep .



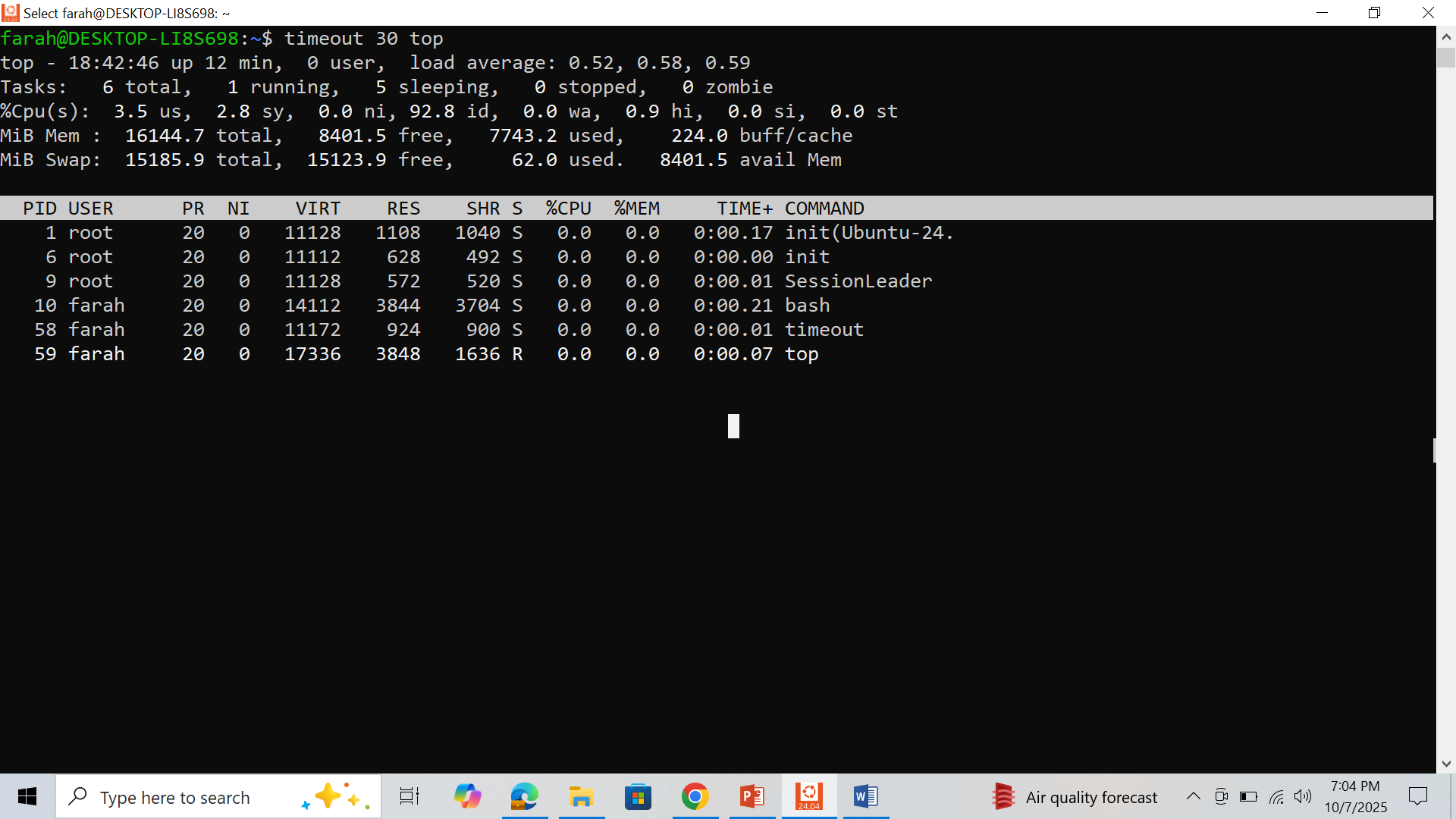
# Part 5: Linux Process Commands

## 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.

System is idle

Which process is consuming the most memory.

System is idle

## 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**

,

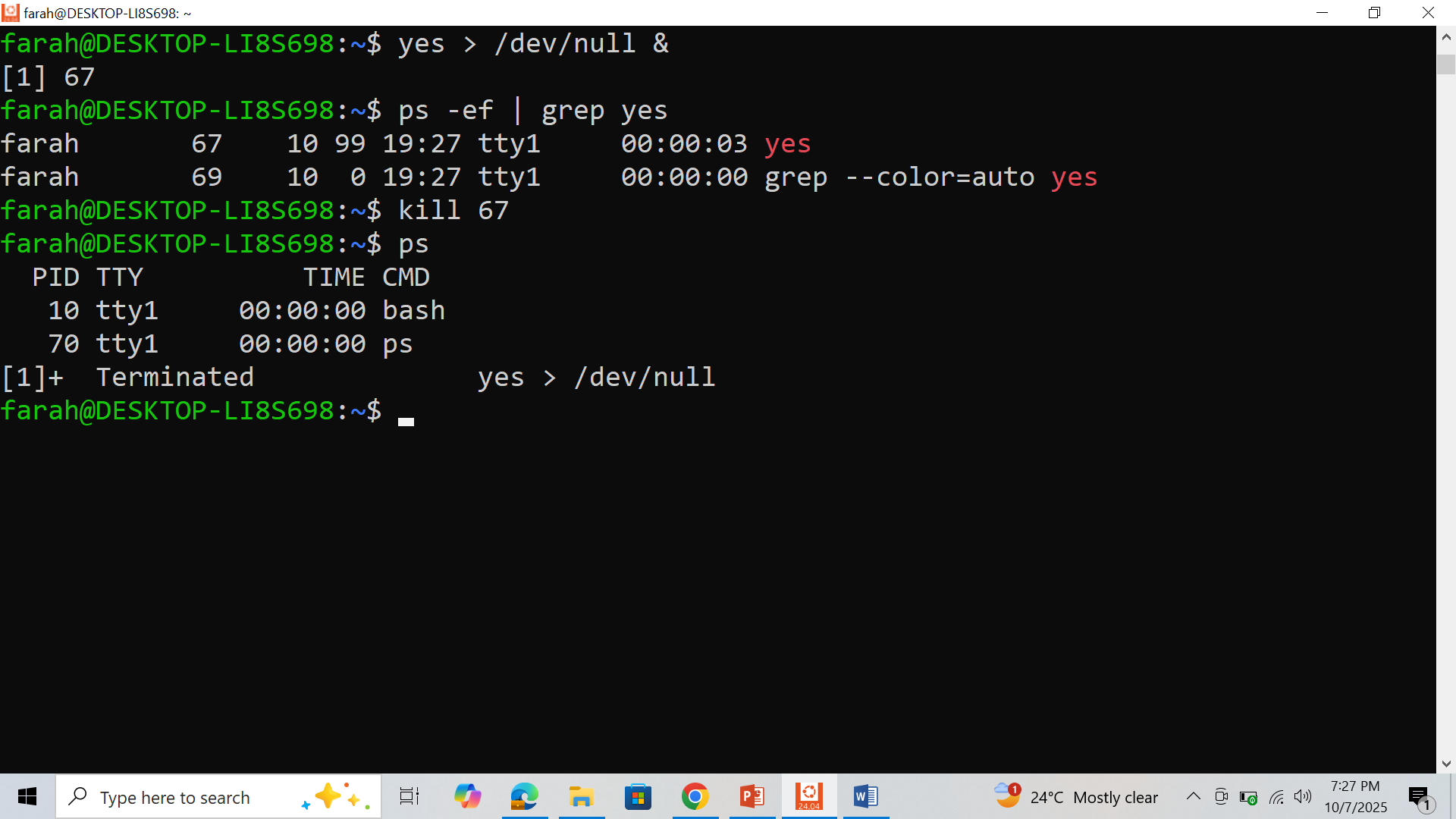
then resume in background using

b

g

.

2. Practice with Infinite processes:



3. Foreground and Background processes:



# Part 6: C Programs on Processes

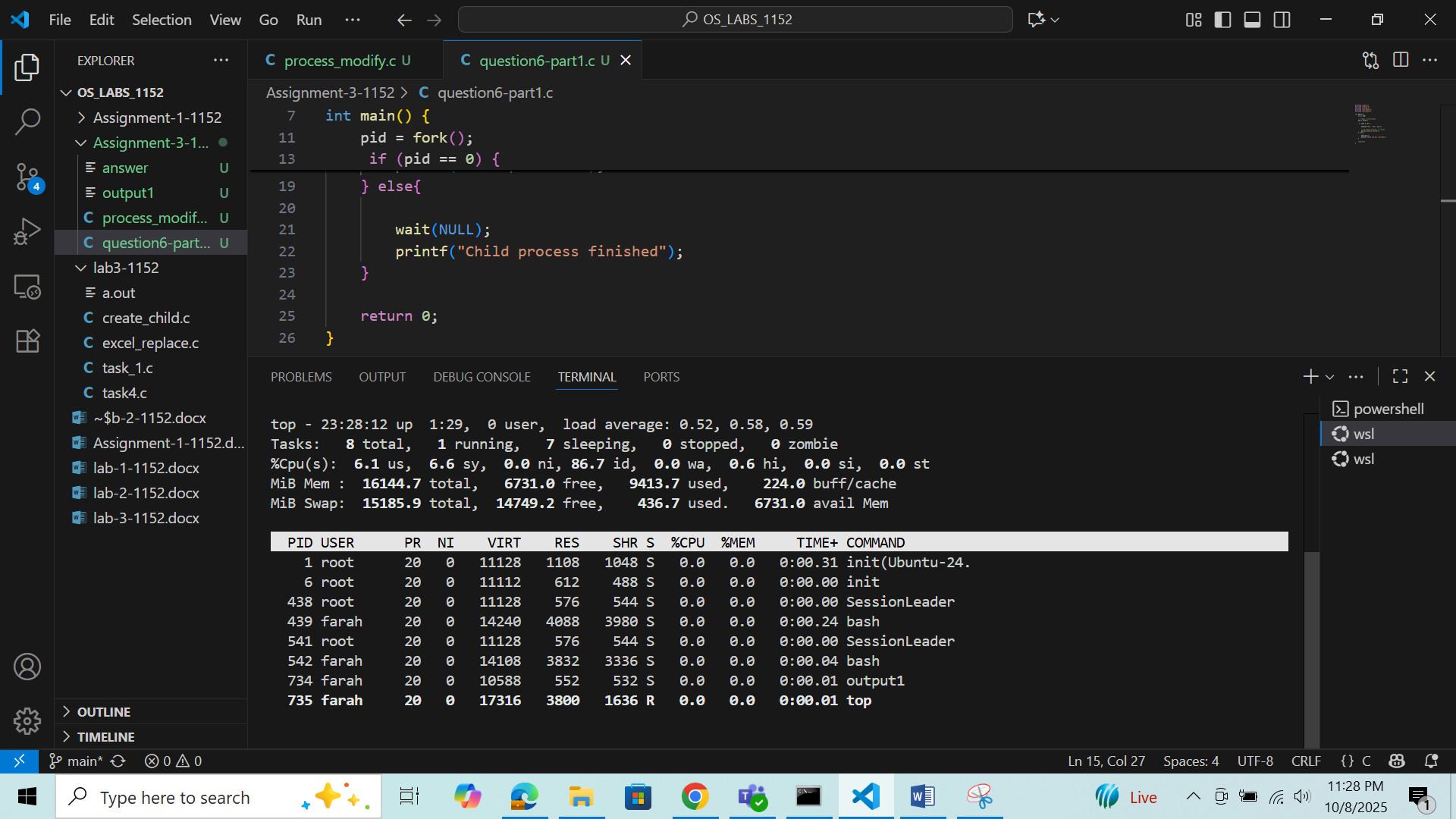
**Program 1 – Exec with top**

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

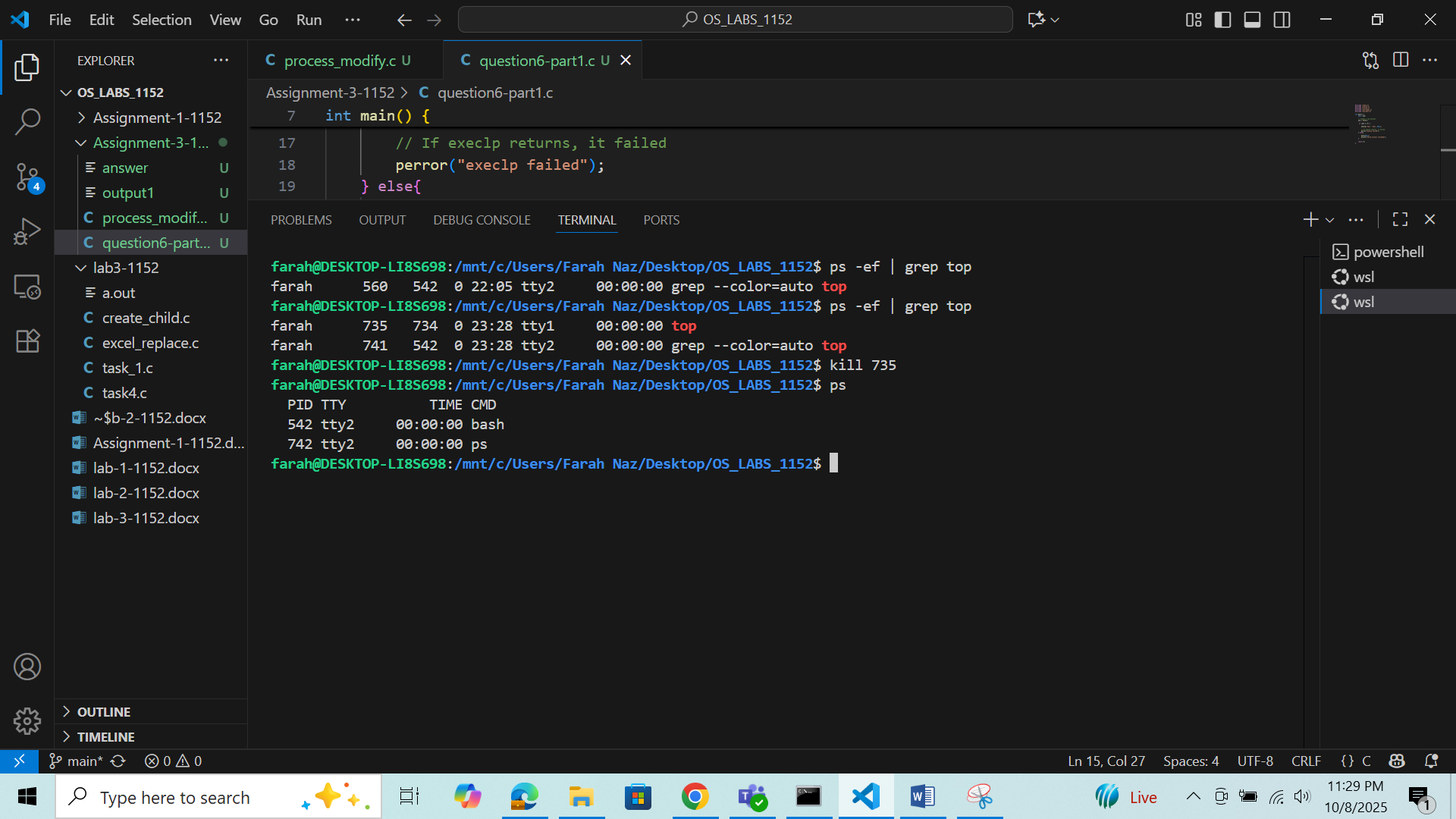
Run the program.

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.



**Killed the child with ID 735 immediately:**



**Program 2 – Incomplete Program**

pid

\_

t

pid

=

fork

(

)

;

if

(

pid

==

0

)

{

//

TODO

:

R

epl

ac

e

this

c

hild

pro

c

ess

with

the

"

d

a

te

"

c

omm

a

nd

using

exe

c

lp

//

H

int

:

exe

c

lp

(

"

d

a

te

", "

d

a

te

",

NULL

)

;

}

else

{

//

TODO

:

Ma

ke

p

a

rent

w

a

it

for

c

hild

b

efore

printing

"

C

hild

finished

"

}

return

0

;

}

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

