NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Subject

Operating System

SUBMITTED BY:

Fatima Waseem: 23-NTU-CS1155

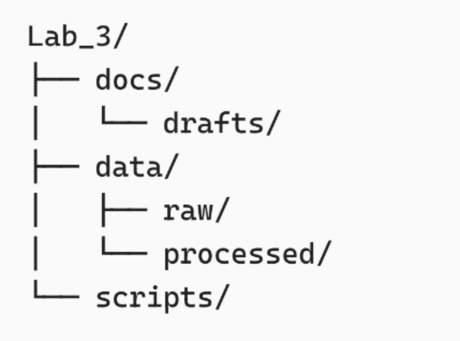
SECTION SE: 5th (A)

SUBMITTED TO:

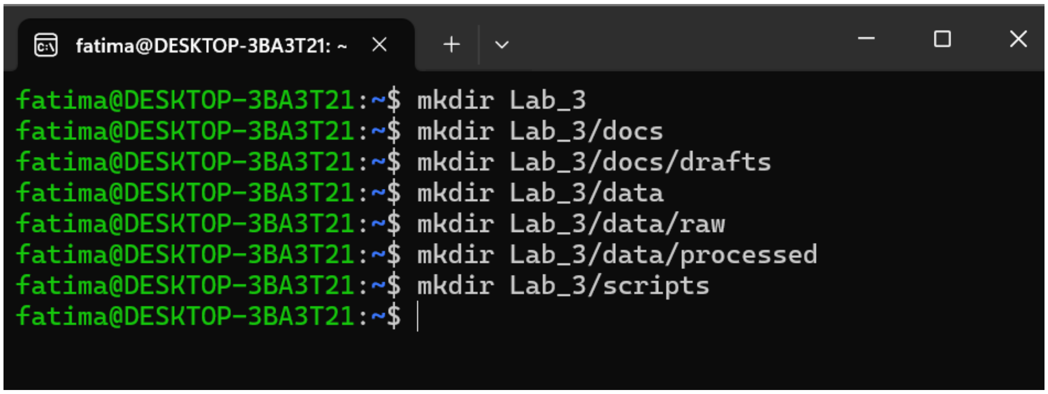
Sir Nasir

Home Task

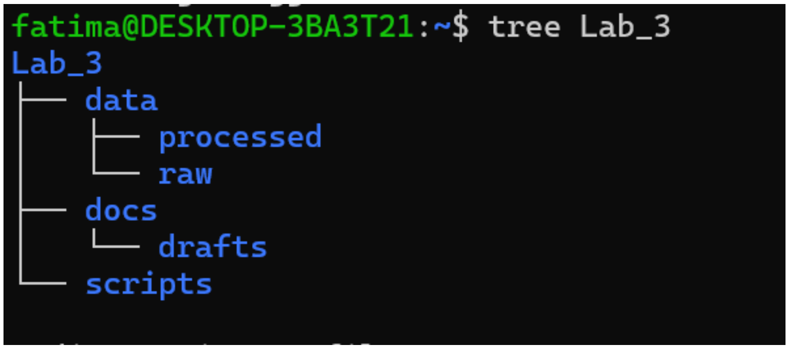
1. Create the following directory structure in your home directory:



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



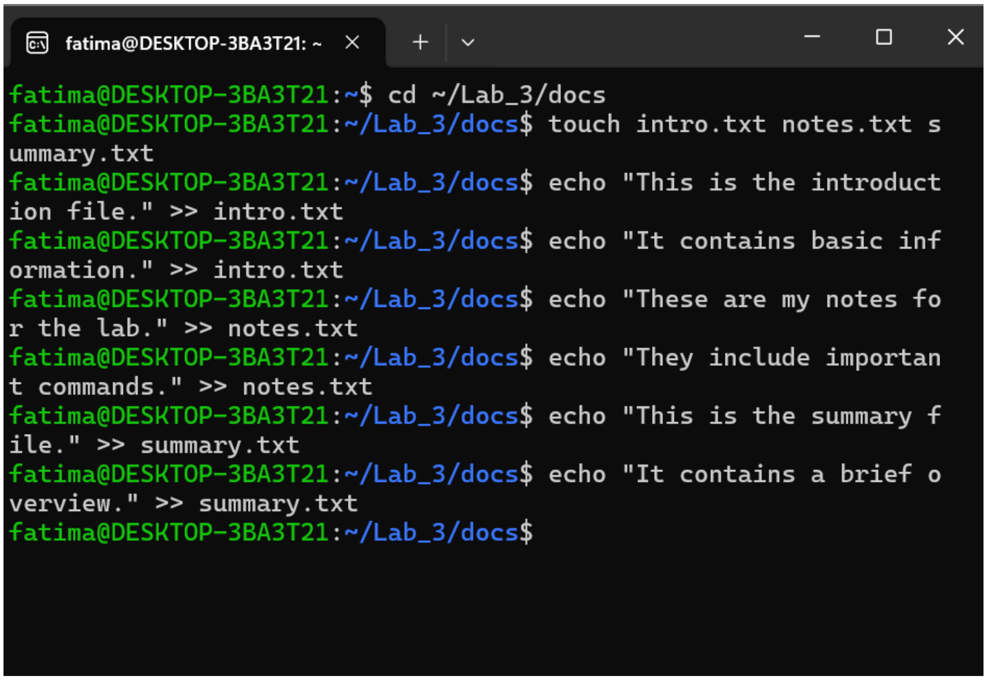
```
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/docs
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/docs/drafts
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/data
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/data/raw
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/data/processed
fatima@DESKTOP-3BA3T21:~$ mkdir Lab_3/scripts
fatima@DESKTOP-3BA3T21:~$
```



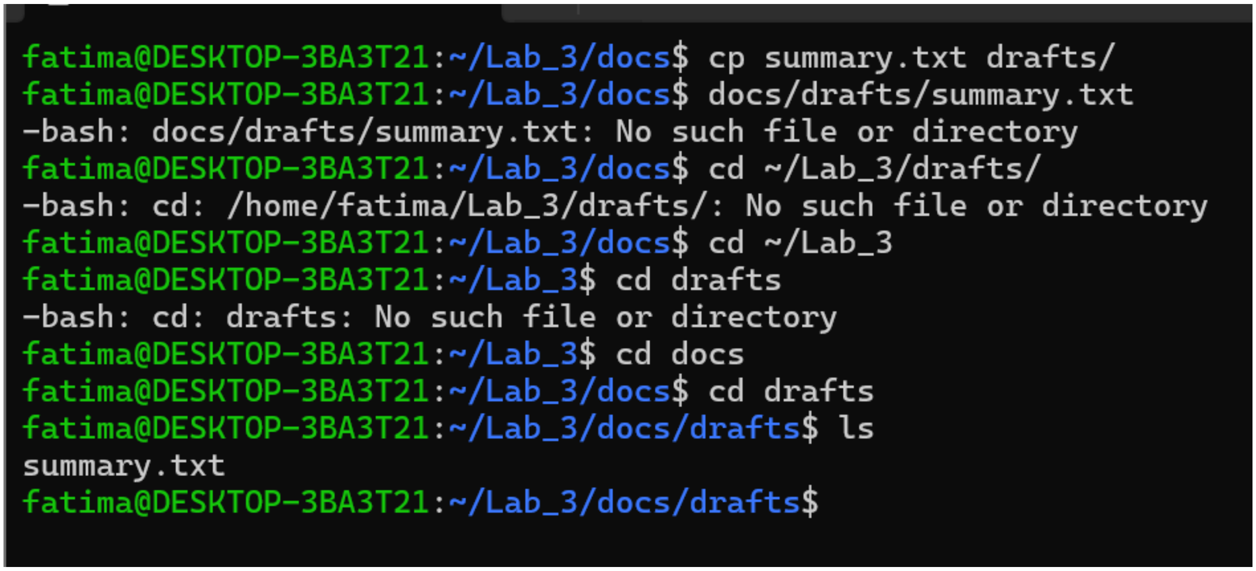
```
fatima@DESKTOP-3BA3T21:~$ tree Lab_3
Lab_3
├── data
│   ├── processed
│   └── raw
├── docs
│   └── drafts
└── scripts
```

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.



```
fatima@DESKTOP-3BA3T21:~$ cd ~/Lab_3/docs
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ touch intro.txt notes.txt s
ummary.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "This is the introduct
ion file." >> intro.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "It contains basic inf
ormation." >> intro.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "These are my notes fo
r the lab." >> notes.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "They include importan
t commands." >> notes.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "This is the summary f
ile." >> summary.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ echo "It contains a brief o
verview." >> summary.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$
```



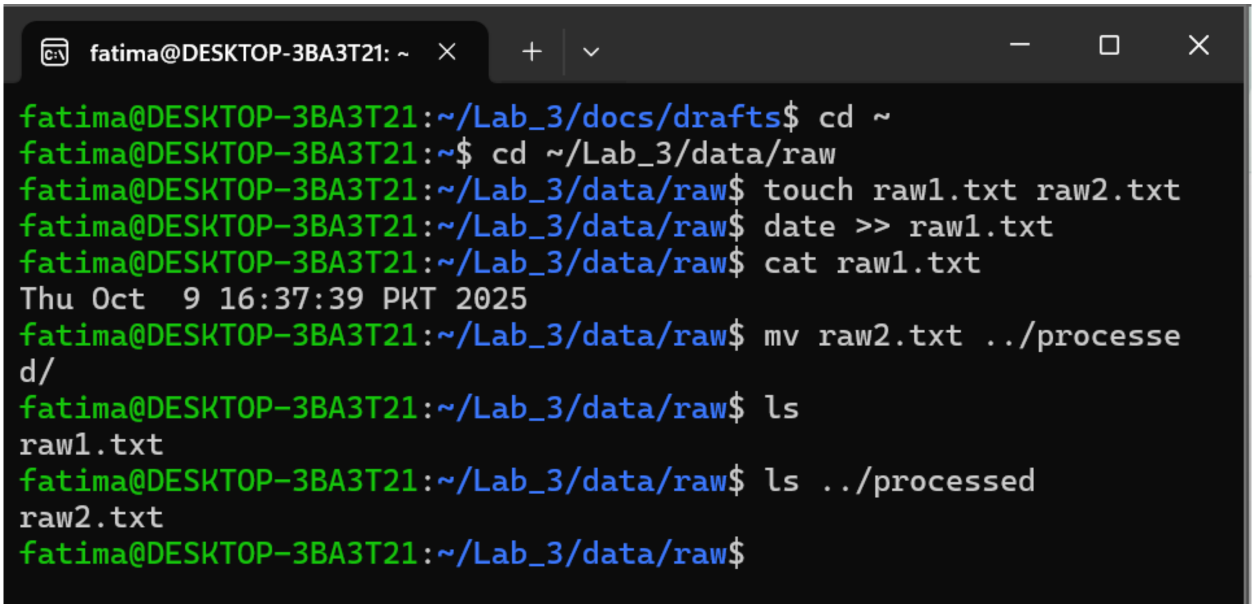
```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cp summary.txt drafts/
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ docs/drafts/summary.txt
-bash: docs/drafts/summary.txt: No such file or directory
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cd ~/Lab_3/drafts/
-bash: cd: /home/fatima/Lab_3/drafts/: No such file or directory
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cd ~/Lab_3
fatima@DESKTOP-3BA3T21:~/Lab_3$ cd drafts
-bash: cd: drafts: No such file or directory
fatima@DESKTOP-3BA3T21:~/Lab_3$ cd docs
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cd drafts
fatima@DESKTOP-3BA3T21:~/Lab_3/docs/drafts$ ls
summary.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs/drafts$
```

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
```



```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs/drafts$ cd ~
fatima@DESKTOP-3BA3T21:~$ cd ~/Lab_3/data/raw
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ touch raw1.txt raw2.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ date >> raw1.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ cat raw1.txt
Thu Oct  9 16:37:39 PKT 2025
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ mv raw2.txt ../processe
d/
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ ls
raw1.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ ls ../processed
raw2.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$
```

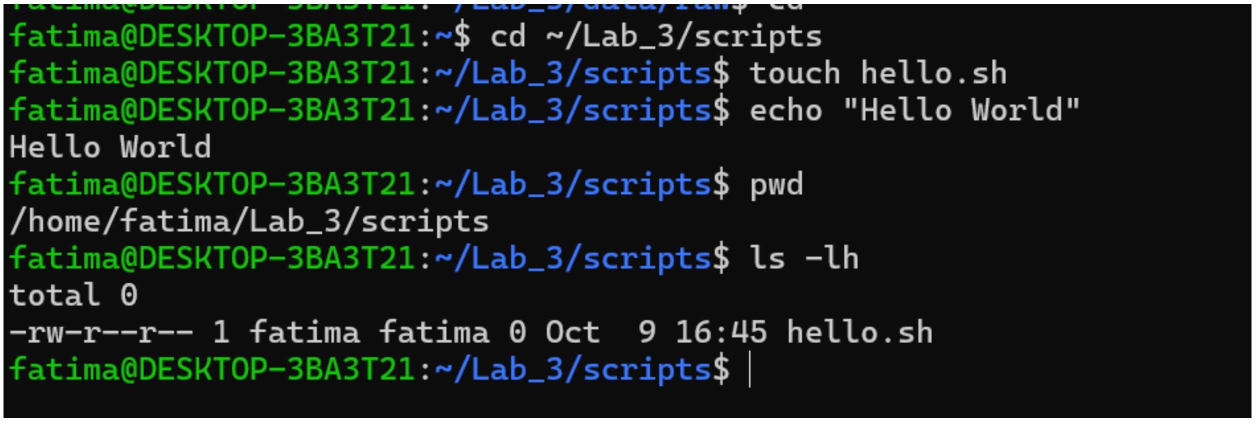
4.

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).

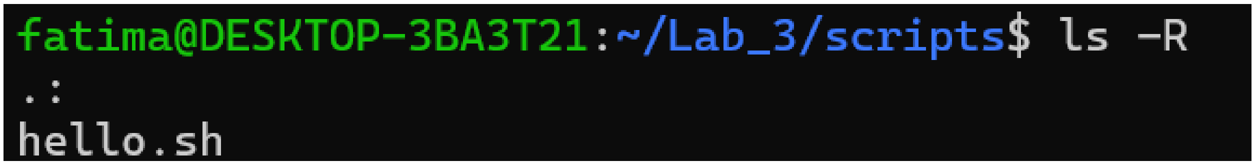


```
fatima@DESKTOP-3BA3T21:~/Lab_3/data/raw$ cd
fatima@DESKTOP-3BA3T21:~$ cd ~/Lab_3/scripts
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ touch hello.sh
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ echo "Hello World"
Hello World
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ pwd
/home/fatima/Lab_3/scripts
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ ls -lh
total 0
-rw-r--r-- 1 fatima fatima 0 Oct  9 16:45 hello.sh
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ 
```

5.

Display the directory structure recursively and take a screenshot:

```
ls -R
```



```
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ ls -R
.:
hello.sh
```

Part 2: Practice with basic linux commands.

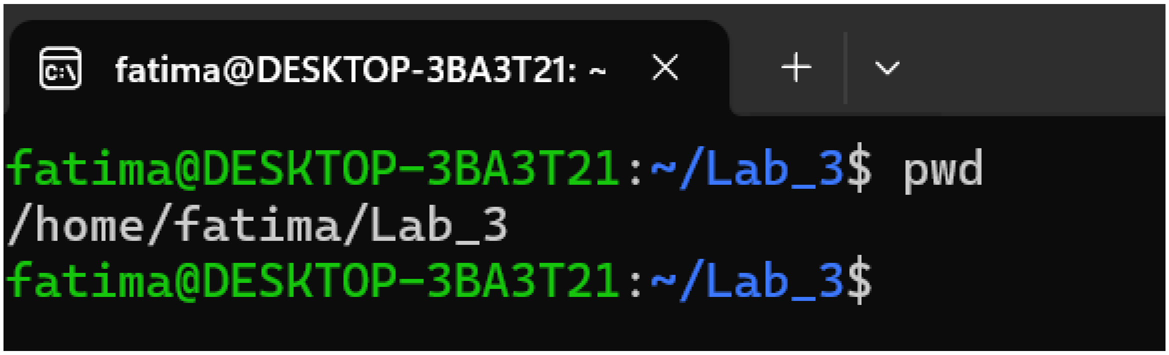
Run the following commands inside `Lab_3/` and note their outputs:

- `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.

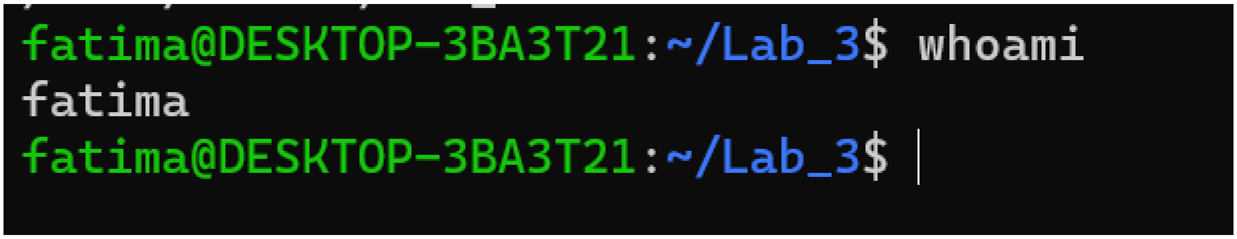
Commands:

Pwd:



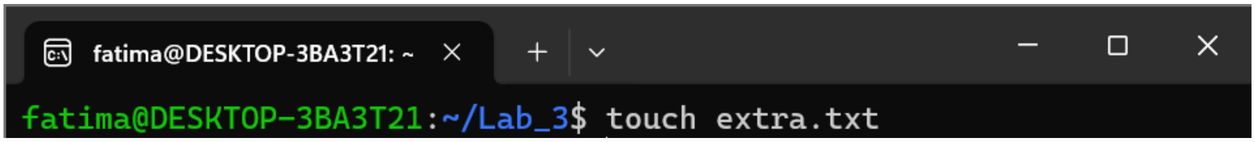
```
fatima@DESKTOP-3BA3T21:~/Lab_3$ pwd
/home/fatima/Lab_3
fatima@DESKTOP-3BA3T21:~/Lab_3$
```

Whoami:



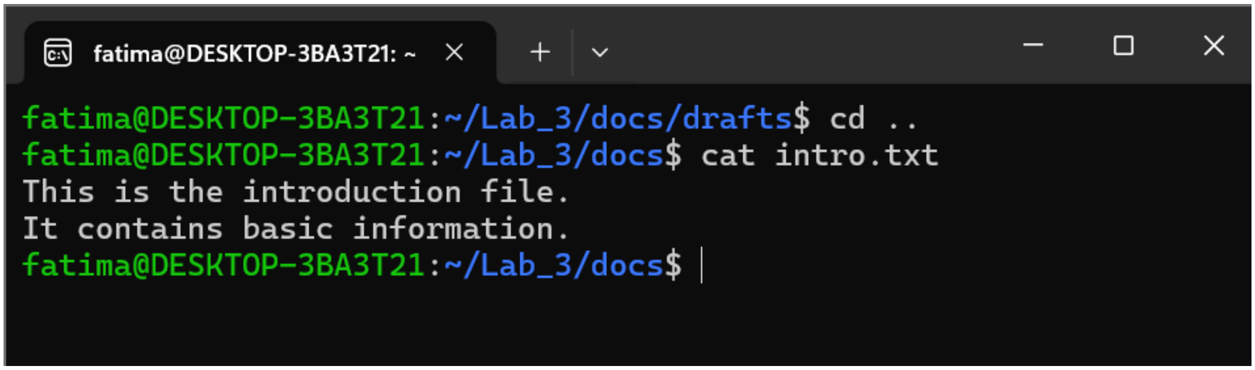
```
fatima@DESKTOP-3BA3T21:~/Lab_3$ whoami
fatima
fatima@DESKTOP-3BA3T21:~/Lab_3$
```

Touch extra.txt



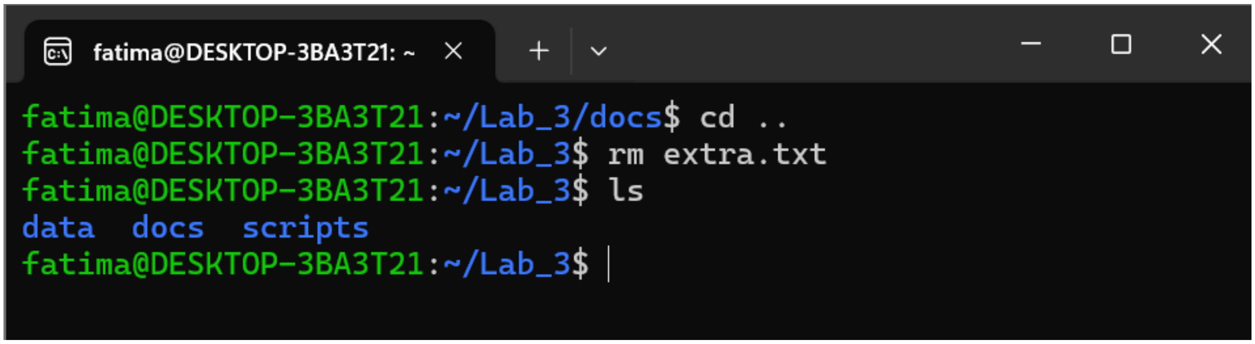
```
fatima@DESKTOP-3BA3T21:~/Lab_3$ touch extra.txt
```

Cat intro.txt



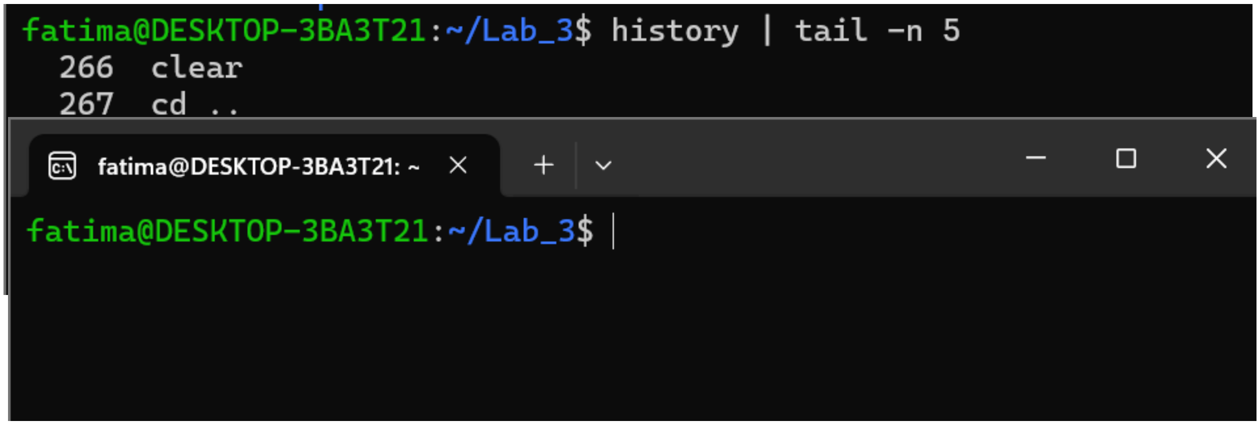
```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs/drafts$ cd ..
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cat intro.txt
This is the introduction file.
It contains basic information.
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$
```

Rm extra.txt



```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ cd ..
fatima@DESKTOP-3BA3T21:~/Lab_3$ rm extra.txt
fatima@DESKTOP-3BA3T21:~/Lab_3$ ls
data  docs  scripts
fatima@DESKTOP-3BA3T21:~/Lab_3$ 
```

History | tail -n 5:



```
fatima@DESKTOP-3BA3T21:~/Lab_3$ history | tail -n 5
  266  clear
  267  cd ..
```

```
fatima@DESKTOP-3BA3T21:~/Lab_3$ 
```

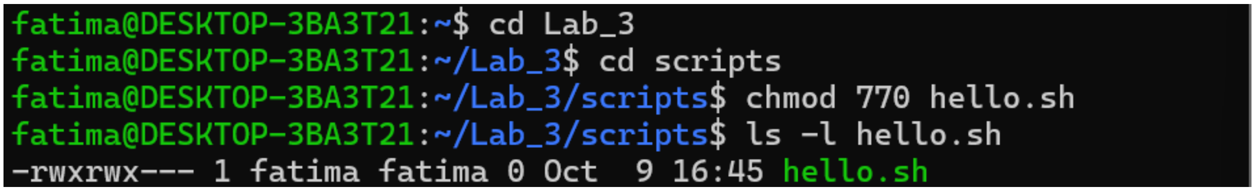
Clear:

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
    ```

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

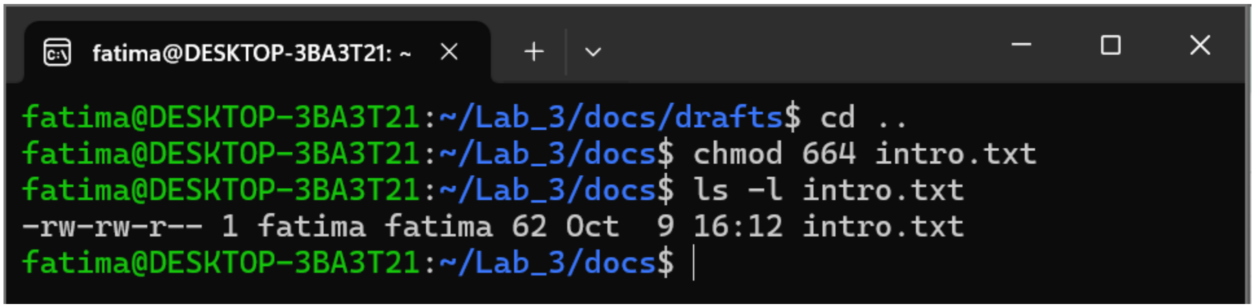


```
fatima@DESKTOP-3BA3T21:~$ cd Lab_3
fatima@DESKTOP-3BA3T21:~/Lab_3$ cd scripts
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ chmod 770 hello.sh
fatima@DESKTOP-3BA3T21:~/Lab_3/scripts$ ls -l hello.sh
-rwxrwx--- 1 fatima fatima 0 Oct  9 16:45 hello.sh
```

Owner → Read & Write

Group → Read & Write

Others → Read only

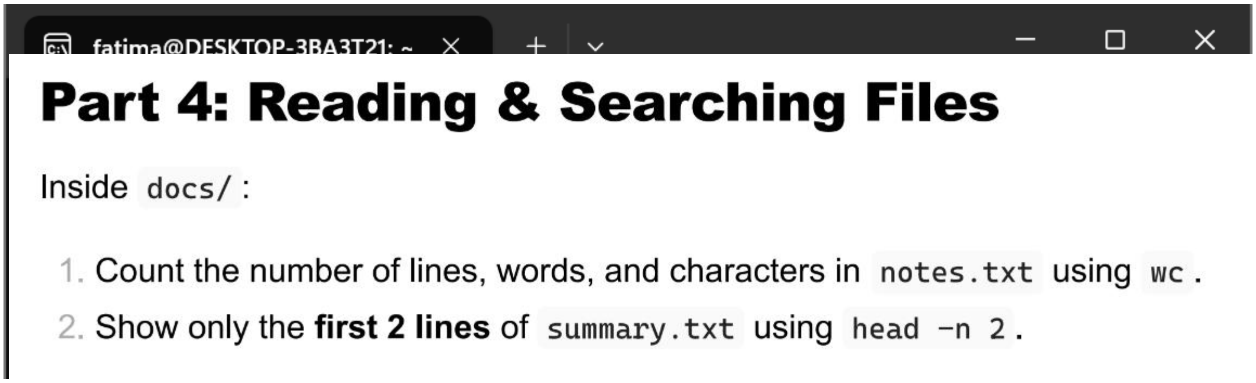


```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs/drafts$ cd ..
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ chmod 664 intro.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ ls -l intro.txt
-rw-rw-r-- 1 fatima fatima 62 Oct  9 16:12 intro.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$
```

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

   ```
   ls -l
   ```

   Take a screenshot of the output.

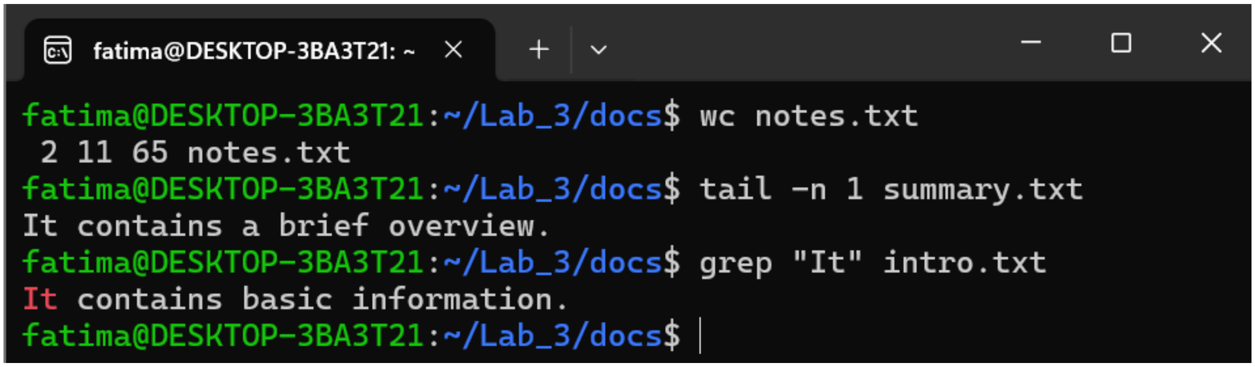


Part 4: Reading & Searching Files

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`.

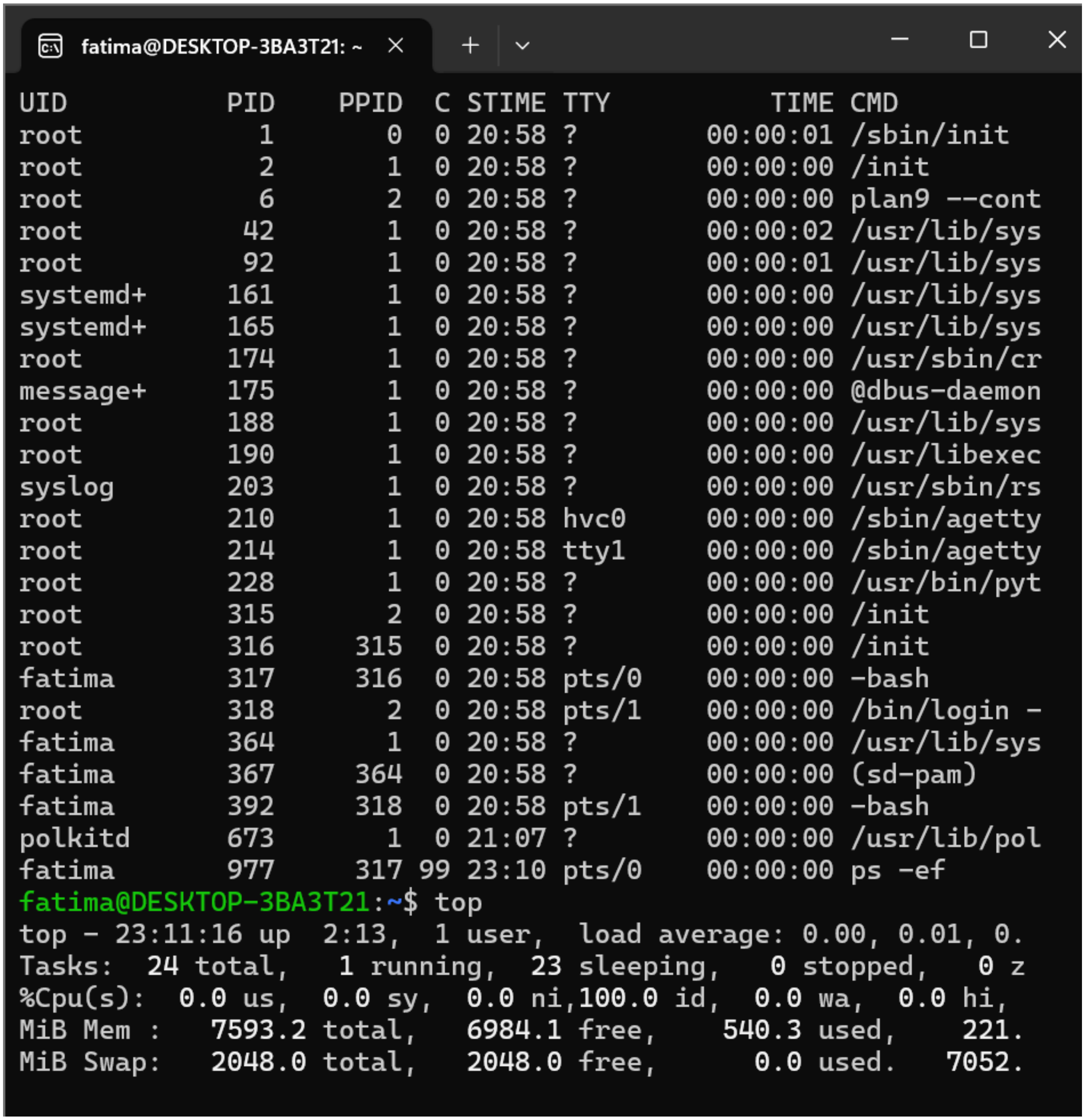
Take screenshots.



```
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ wc notes.txt
 2 11 65 notes.txt
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ tail -n 1 summary.txt
It contains a brief overview.
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$ grep "It" intro.txt
It contains basic information.
fatima@DESKTOP-3BA3T21:~/Lab_3/docs$
```

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.



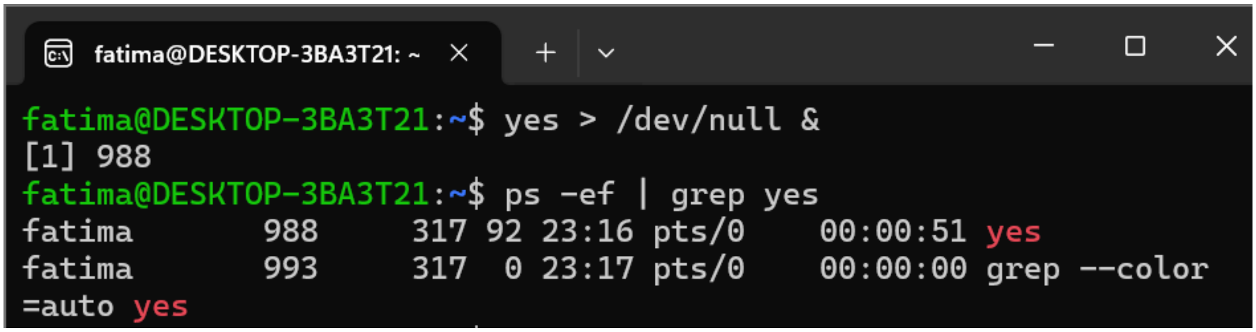
```
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 20:58 ?        00:00:01 /sbin/init
root           2       1  0 20:58 ?        00:00:00 /init
root           6       2  0 20:58 ?        00:00:00 plan9 --cont
root          42       1  0 20:58 ?        00:00:02 /usr/lib/sys
root          92       1  0 20:58 ?        00:00:01 /usr/lib/sys
systemd+     161       1  0 20:58 ?        00:00:00 /usr/lib/sys
systemd+     165       1  0 20:58 ?        00:00:00 /usr/lib/sys
root         174       1  0 20:58 ?        00:00:00 /usr/sbin/cr
message+     175       1  0 20:58 ?        00:00:00 @dbus-daemon
root         188       1  0 20:58 ?        00:00:00 /usr/lib/sys
root         190       1  0 20:58 ?        00:00:00 /usr/libexec
syslog       203       1  0 20:58 ?        00:00:00 /usr/sbin/rs
root         210       1  0 20:58 hvc0     00:00:00 /sbin/agetty
root         214       1  0 20:58 tty1     00:00:00 /sbin/agetty
root         228       1  0 20:58 ?        00:00:00 /usr/bin/pyt
root         315       2  0 20:58 ?        00:00:00 /init
root         316     315  0 20:58 ?        00:00:00 /init
fatima       317     316  0 20:58 pts/0    00:00:00 -bash
root         318       2  0 20:58 pts/1    00:00:00 /bin/login -
fatima       364       1  0 20:58 ?        00:00:00 /usr/lib/sys
fatima       367     364  0 20:58 ?        00:00:00 (sd-pam)
fatima       392     318  0 20:58 pts/1    00:00:00 -bash
polkitd      673       1  0 21:07 ?        00:00:00 /usr/lib/pol
fatima       977     317 99 23:10 pts/0    00:00:00 ps -ef
fatima@DESKTOP-3BA3T21:~$ top
top - 23:11:16 up  2:13,  1 user,  load average: 0.00, 0.01, 0.
Tasks:  24 total,   1 running,  23 sleeping,   0 stopped,   0 z
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,
MiB Mem :   7593.2 total,   6984.1 free,    540.3 used,    221.
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   7052.
```

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`.



```
fatima@DESKTOP-3BA3T21:~$ yes > /dev/null &
[1] 988
fatima@DESKTOP-3BA3T21:~$ ps -ef | grep yes
fatima       988     317 92 23:16 pts/0    00:00:51 yes
fatima       993     317  0 23:17 pts/0    00:00:00 grep --color
=auto yes
```

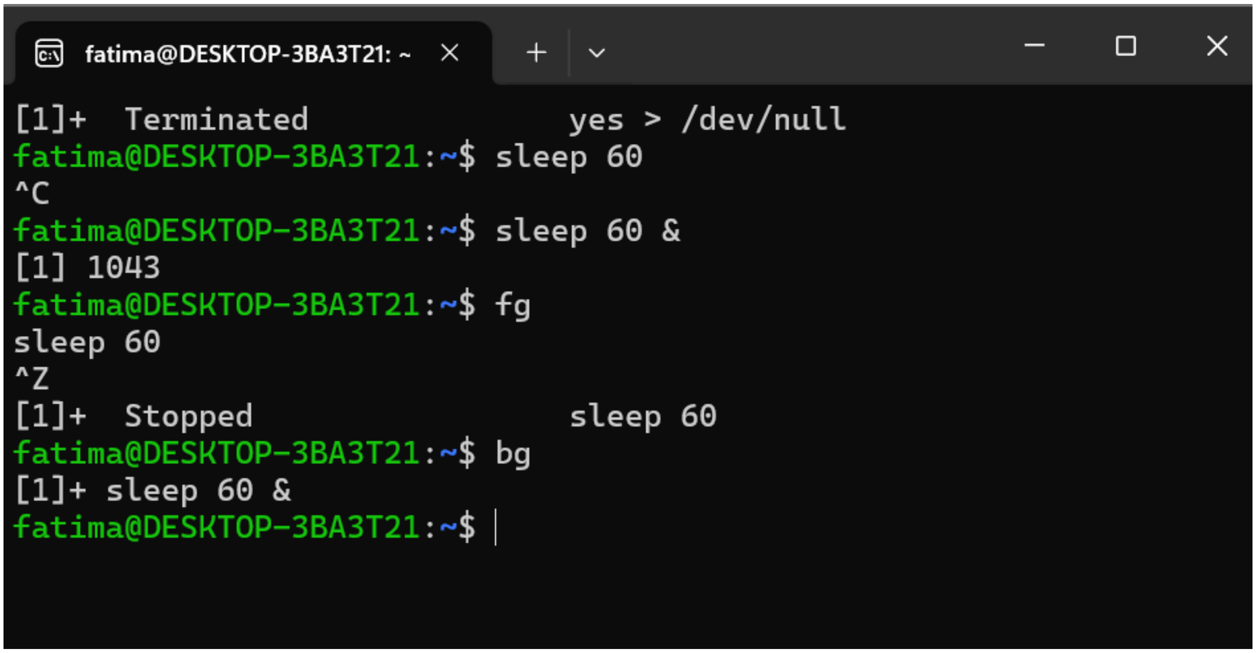
2. Practice with Infinite Process

- 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`.



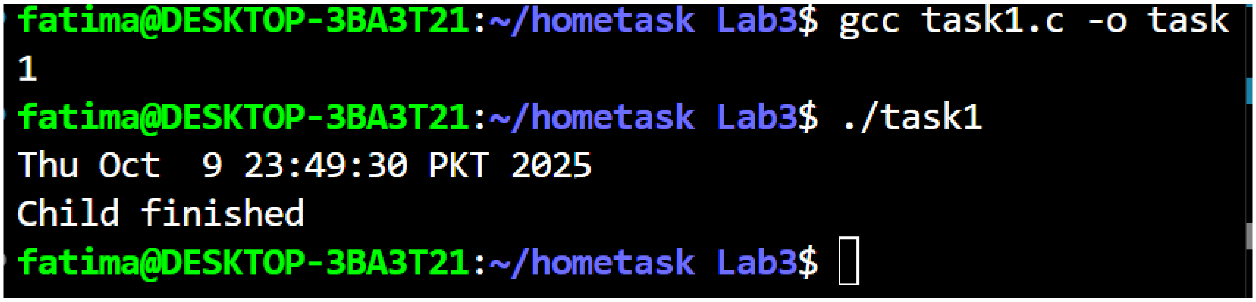
```
[1]+  Terminated              yes > /dev/null
fatima@DESKTOP-3BA3T21:~$ sleep 60
^C
fatima@DESKTOP-3BA3T21:~$ sleep 60 &
[1] 1043
fatima@DESKTOP-3BA3T21:~$ fg
sleep 60
^Z
[1]+  Stopped                 sleep 60
fatima@DESKTOP-3BA3T21:~$ bg
[1]+ sleep 60 &
fatima@DESKTOP-3BA3T21:~$
```

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



```
fatima@DESKTOP-3BA3T21:~/hometask Lab3$ gcc task1.c -o task
1
fatima@DESKTOP-3BA3T21:~/hometask Lab3$ ./task1
Thu Oct  9 23:49:30 PKT 2025
Child finished
fatima@DESKTOP-3BA3T21:~/hometask Lab3$
```