NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Eman Faisal 23-NTU-CS-1149

SECTION SE: 5th (A)

Operating systems Lab3-homatask

SUBMITTED TO:

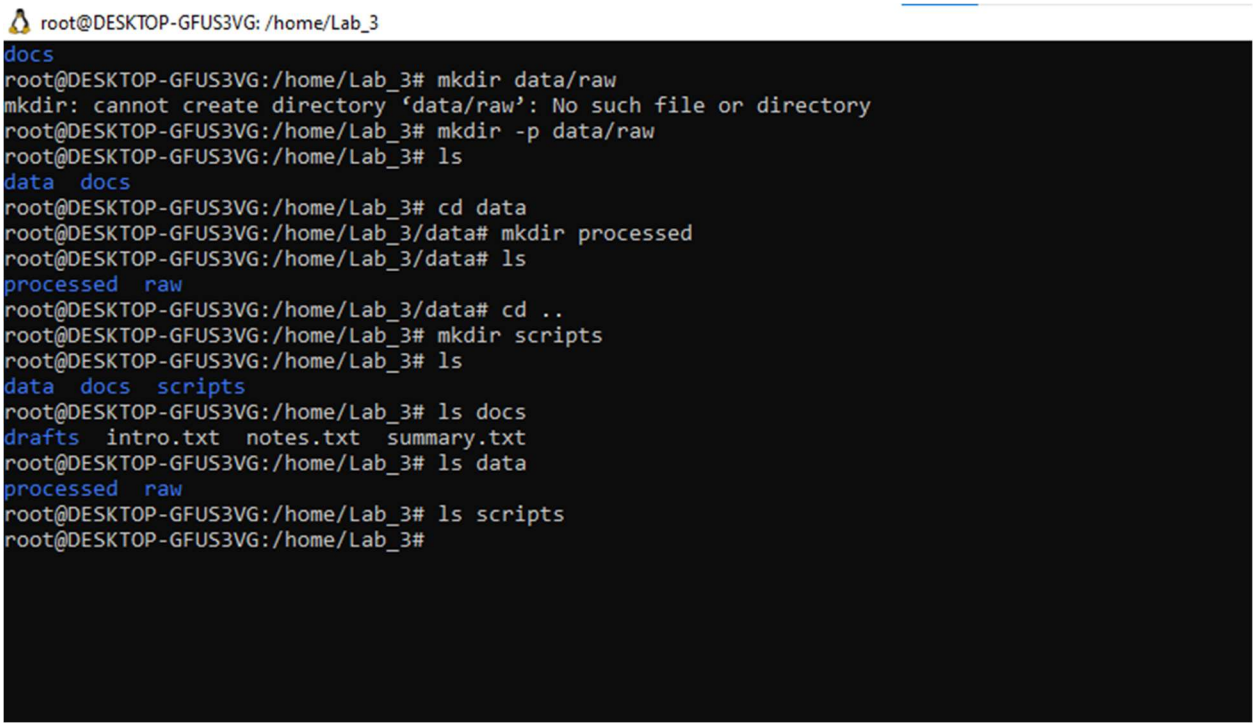
Sir Nasir mahmood

SUBMISSION DATE: 10/6/25

Part 1: File and Directory Operations

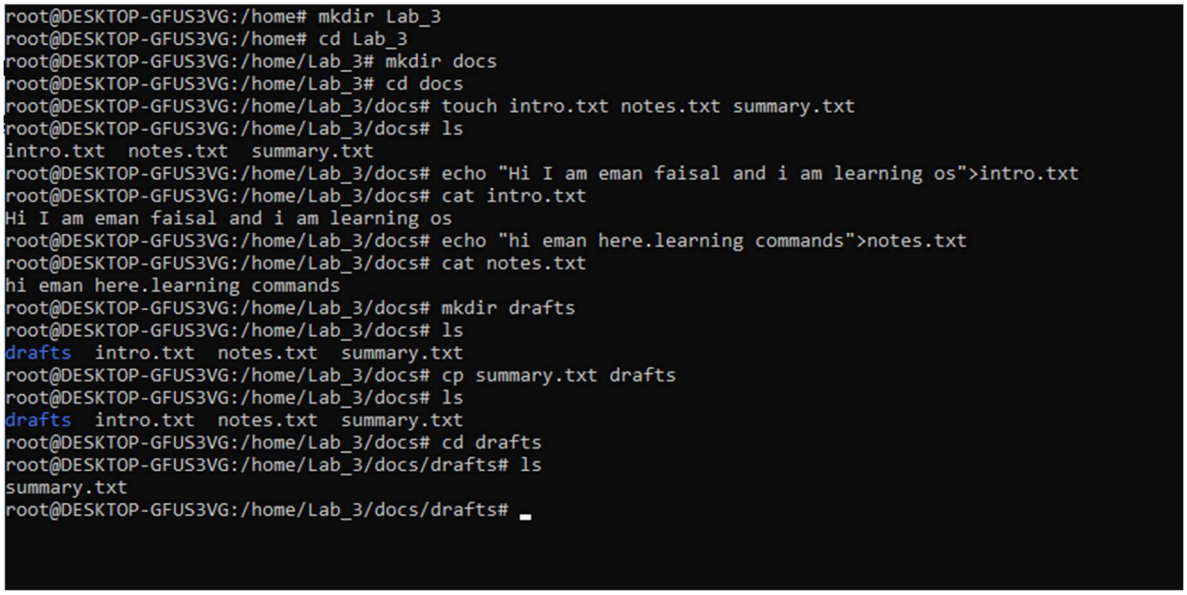
1. Create the following directory structure in your home directory.

```
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.

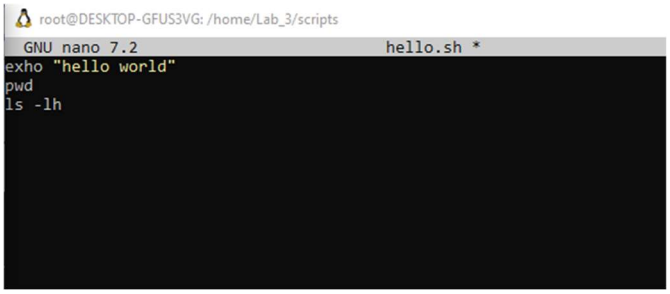
IN this,I used **touch and echo** to create and write into files.Also ,I used **cp** to copy summary.txt into drafts and I showed my results using **cat and ls** command



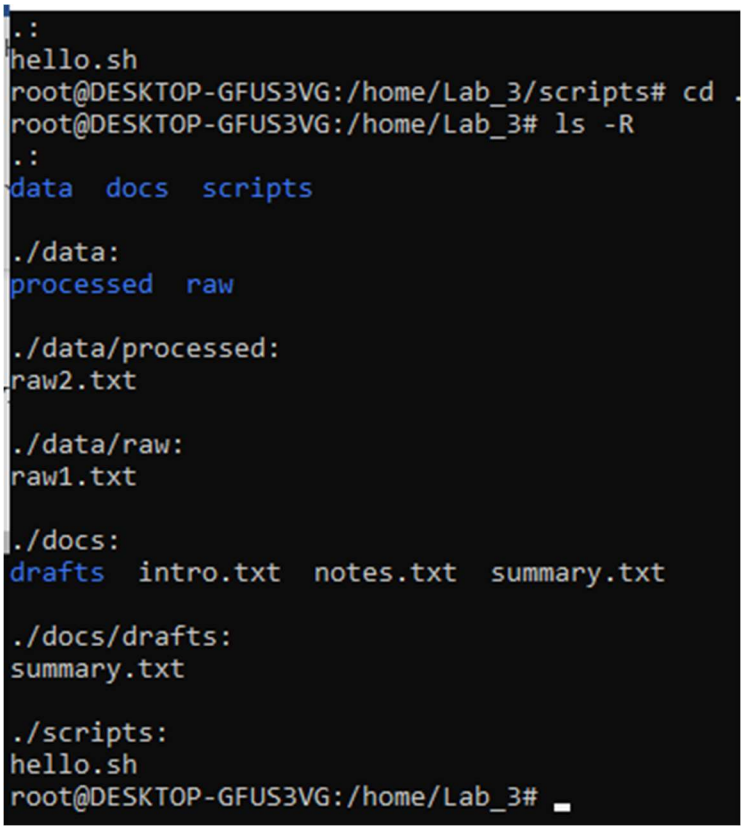
3. Inside data/raw/ : Create two files: raw1.txt , raw2.txt . Append the current date into raw1.txt using the date command. Move raw2.txt into processed/ using mv . The syntax is: mv source destination.

```
root@DESKTOP-GFUS3VG: /home/Lab_3/data
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# date >> raw1.txt
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# cat raw1.txt
Mon Oct  6 22:29:41 PDT 2025
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# cd..
cd..: command not found
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# cd ..
root@DESKTOP-GFUS3VG:/home/Lab_3/data# mv raw2.txt processed
mv: cannot stat 'raw2.txt': No such file or directory
root@DESKTOP-GFUS3VG:/home/Lab_3/data# mv raw/raw2.txt processed
root@DESKTOP-GFUS3VG:/home/Lab_3/data# ls
processed  raw
root@DESKTOP-GFUS3VG:/home/Lab_3/data# cd raw
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# ls
raw1.txt
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# ls processed
ls: cannot access 'processed': No such file or directory
root@DESKTOP-GFUS3VG:/home/Lab_3/data/raw# cd ..
root@DESKTOP-GFUS3VG:/home/Lab_3/data# ls processed
raw2.txt
root@DESKTOP-GFUS3VG:/home/Lab_3/data#
```

4. Inside scripts/ : Create a script named hello.sh with the following content: echo "Hello World" pwd ls -lh



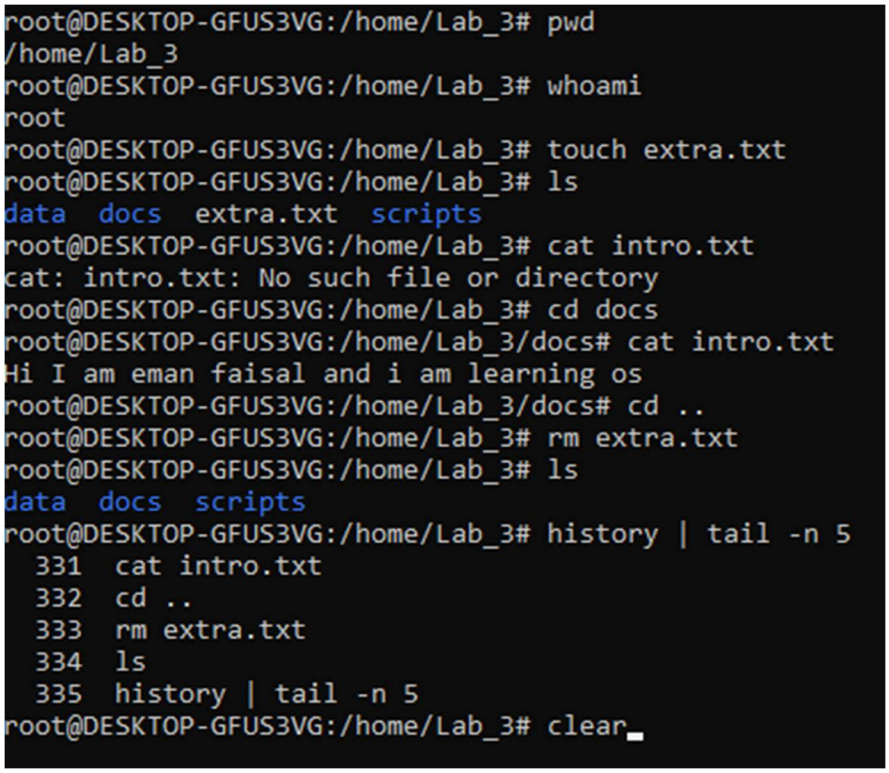
5. Display the directory structure recursively and take a screenshot: ls -R



Part 2: Practice with Basic Linux Commands Run the following commands inside Lab_3/ and note their outputs.

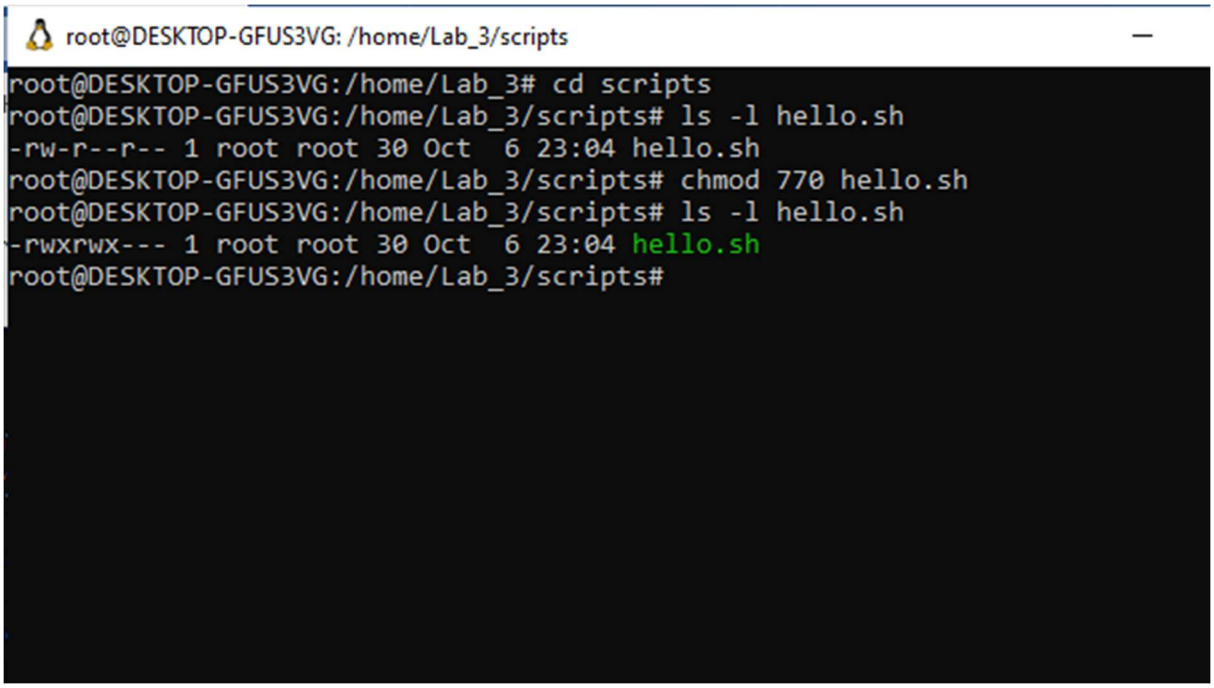
- `pwd` → Show current working directory.
- `whoami` → Display the current logged-in user.
- `touch extra.txt` → Create an empty file.
- `cat intro.txt` → Display file contents.
- `rm extra.txt` → Delete a file.
- `history | tail -n 5` → Show your last 5 executed commands.
- `clear` → Clear the terminal.

Take screenshots of commands and outputs.



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: Take a screenshot of its output.



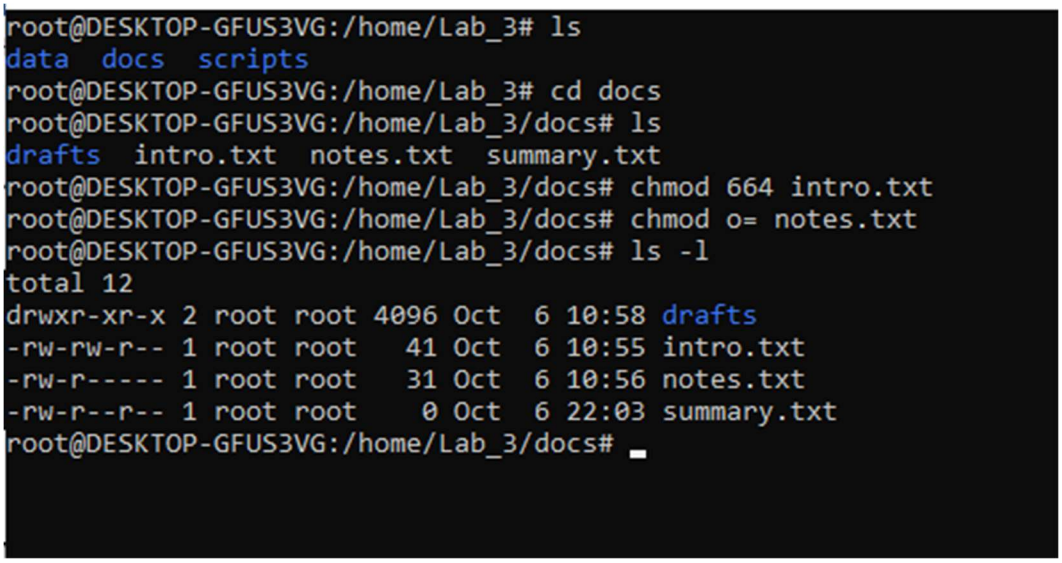
Part 2,3 and 4:

2. Change the permissions of intro.txt using numeric notation so that: Owner → Read & Write Group → Read & Write Others → Read only

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

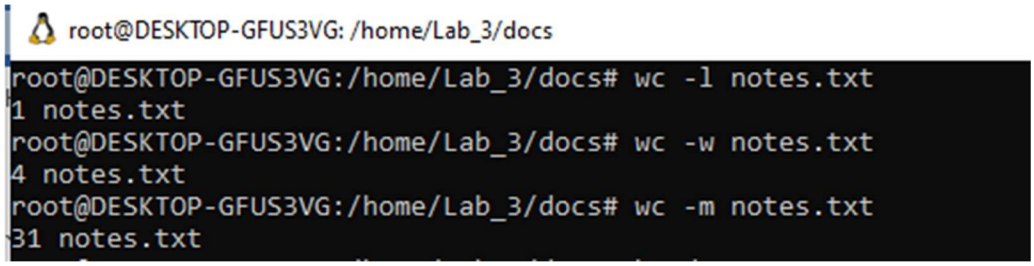
4. Verify all changes with: ls-l

COMMANDS AND OUTPUT:

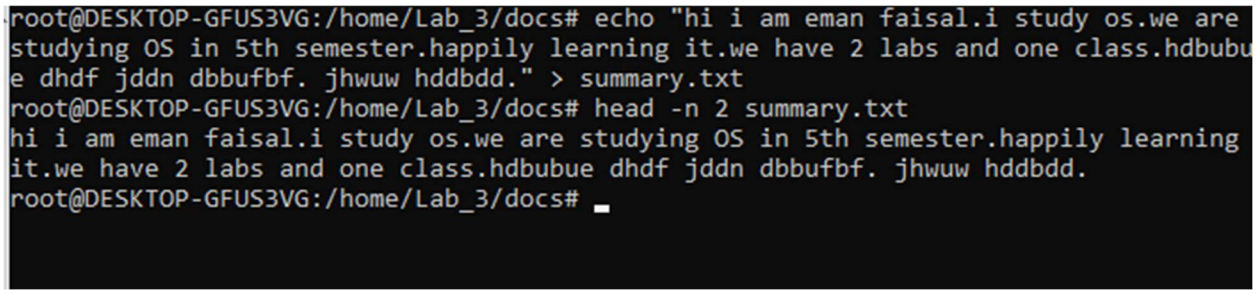


PART 4:

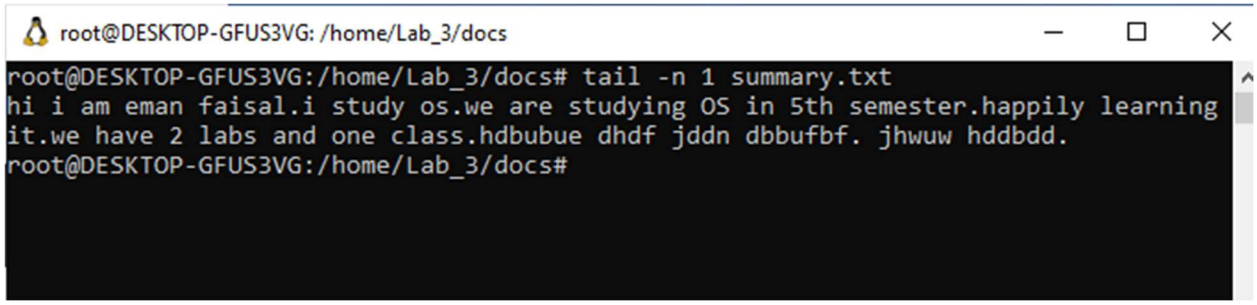
1. **Count the number of lines, words, and characters in notes.txt using wc .**



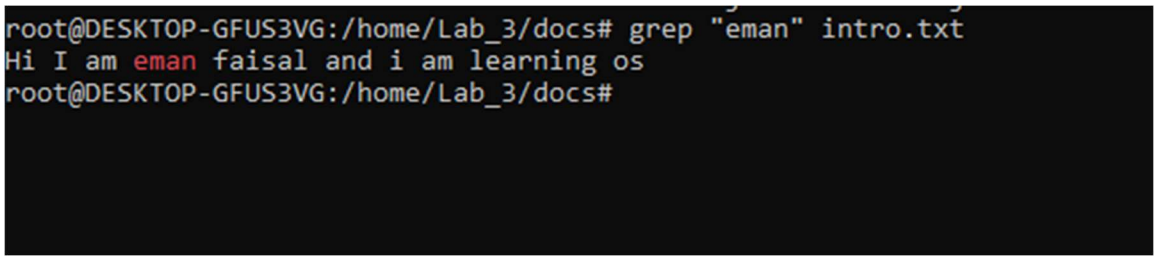
2. **Show only the first 2 lines of summary.txt using head -n 2 . Take screenshots.**



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

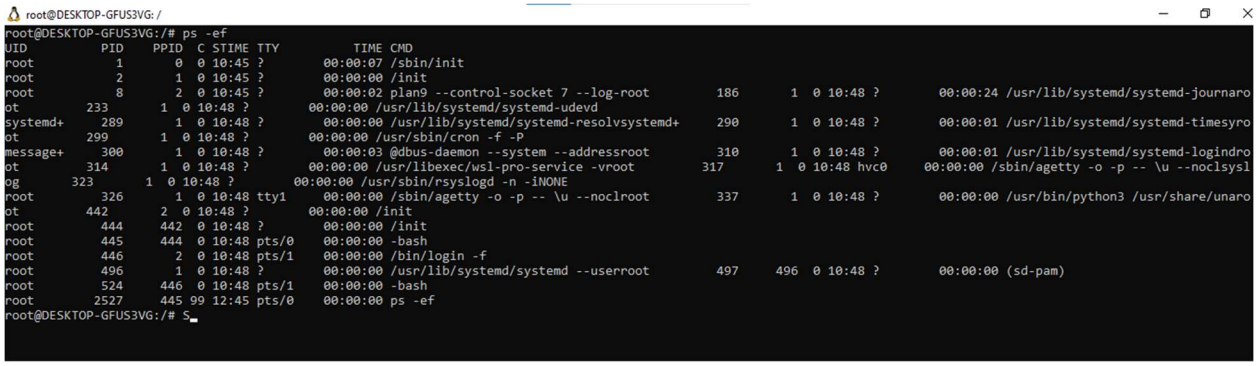


4. **Search for a keyword (of your choice) in intro.txt using grep**



Part 5: Linux Process Commands

1.Explore Processes Use ps -ef and identify 3 processes running on your system. Note their PID, PPID, and command.

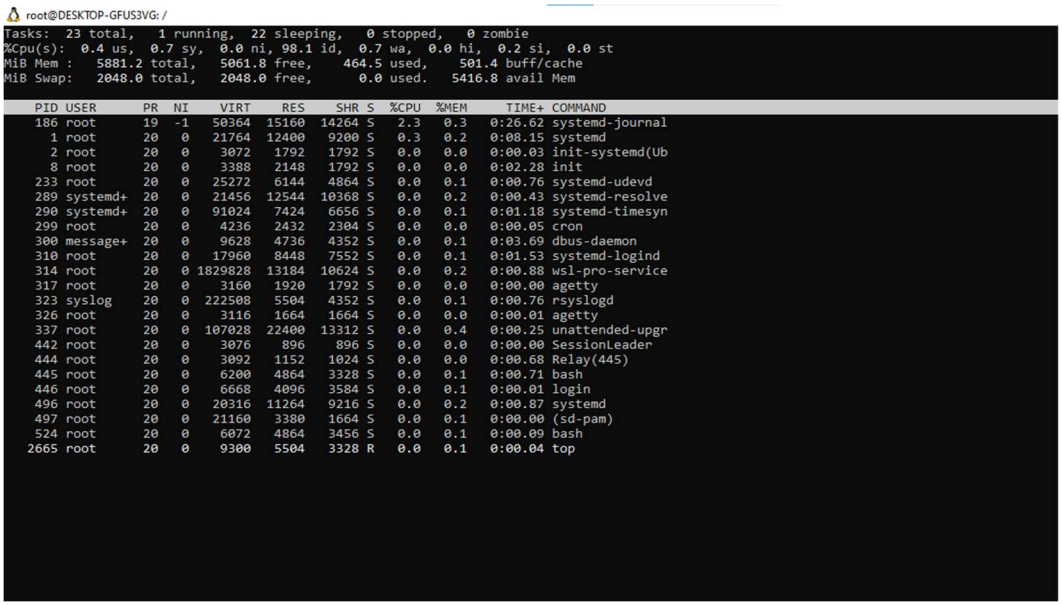


PID:1 PPID:0 CMD: /sbin/init

PID:2 PPID:1 CMD: /init

PID:444 PPID:552 CMD: /init

2.Run top for 20–30 seconds. Write down: Which process is consuming the most CPU. Which process is consuming the most memory.



Highest cpu: 2.3% by 186 process: system journal

Highest memory: 0.3% system journal

2. Practice with Infinite Process Start: yes > /dev/null & Locate its PID using ps -ef | grep yes . Kill it using kill and verify using ps.

```
root@DESKTOP-GFUS3VG:/# yes > /dev/null &
[1] 3142
root@DESKTOP-GFUS3VG:/# ps -ef | grep yes
root        3142     445 99 00:09 pts/0    00:00:13 yes
root        3146     445  0 00:09 pts/0    00:00:00 grep --color=auto yes
root@DESKTOP-GFUS3VG:/# kill 3142
root@DESKTOP-GFUS3VG:/# ps -ef | grep yes
root        3148     445  0 00:10 pts/0    00:00:00 grep --color=auto yes
[1]+  Terminated              yes > /dev/null
root@DESKTOP-GFUS3VG:/# ps
    PID TTY          TIME CMD
    445 pts/0    00:00:00 bash
   3157 pts/0    00:00:00 ps
root@DESKTOP-GFUS3VG:/#
```

3. Foreground & Background Jobs Run sleep 60 in foreground and terminate it with Ctrl + C

```
root@DESKTOP-GFUS3VG:/# sleep 60
^C
root@DESKTOP-GFUS3VG:/#
```

Run sleep 60 & in background, bring it to foreground with fg , stop with Ctrl + Z, then resume in background using bg .

```
root@DESKTOP-GFUS3VG: /
root@DESKTOP-GFUS3VG:/# sleep 60
^C
root@DESKTOP-GFUS3VG:/# sleep 60 &
[1] 3216
root@DESKTOP-GFUS3VG:/# fg
sleep 60
^Z
[1]+  Stopped                 sleep 60
root@DESKTOP-GFUS3VG:/# bg
[1]+ sleep 60 &
root@DESKTOP-GFUS3VG:/#
```

PART 6:
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.

Terminal:

```
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# ps -ef | grep top
root       5568    5203  0 01:01 pts/7    00:00:00 grep --color=auto top
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# kill 5203
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework#
```

Code:

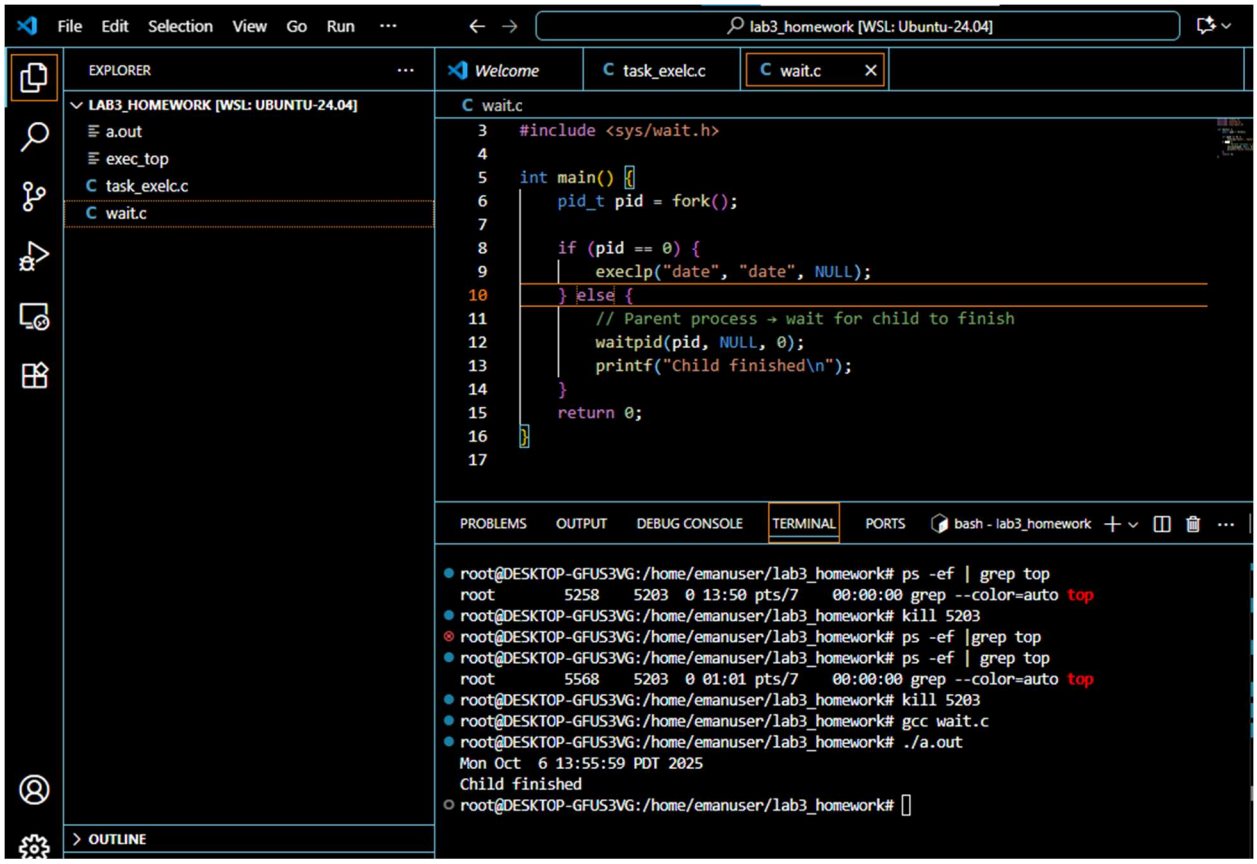
```c
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid = fork();  // create a child process

    if (pid == 0) {
        // Child process: replace its code with the 'top' program
        execlp("top", "top", NULL);
        printf("This will not print if exec succeeds.\n");
    } else if (pid > 0) {
        // Parent process
        printf("Parent still running...\n");
    } else {
        // fork() failed
        perror("fork failed");
    }

    return 0;
}
```

PART2:


File Edit Selection View Go Run ···
lab3_homework [WSL: Ubuntu-24.04]
EXPLORER
LAB3_HOMEWORK [WSL: UBUNTU-24.04]
a.out
exec_top
task_exelc.c
wait.c
Welcome
task_exelc.c
wait.c
wait.c
#include <sys/wait.h>
int main() {
pid_t pid = fork();
if (pid == 0) {
execlp("date", "date", NULL);
} else {
// Parent process → wait for child to finish
waitpid(pid, NULL, 0);
printf("Child finished\n");
}
return 0;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - lab3_homework
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# ps -ef | grep top
root 5258 5203 0 13:50 pts/7 00:00:00 grep --color=auto top
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# kill 5203
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# ps -ef |grep top
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# ps -ef | grep top
root 5568 5203 0 01:01 pts/7 00:00:00 grep --color=auto top
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# kill 5203
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# gcc wait.c
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework# ./a.out
Mon Oct 6 13:55:59 PDT 2025
Child finished
root@DESKTOP-GFUS3VG:/home/emanuser/lab3_homework#
OUTLINE