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:

- 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:

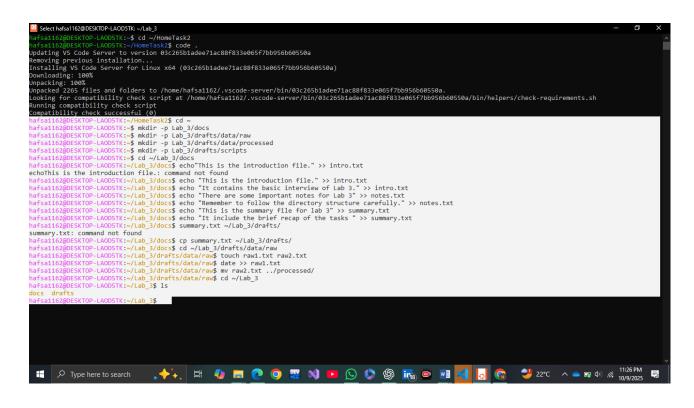
```
mv source destination
```

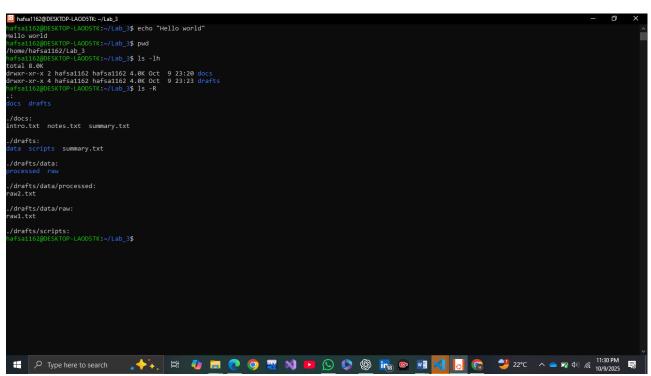
- 4. Inside scripts/:
 - Create a script named hello-sh with the following content:

```
echo "Hello World"
pwd
Is -Ih
```

- Later, you will make it executable (in Part 3).
- 5. Display the directory structure recursively and take a screenshot:

```
ls -R
```





```
## Author Company Com
```

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 \rightarrow Create an empty file. cat
- intro.txt → Display file contents. rm extra.txt →
- Delete a file.
- history | tail -n 5 \rightarrow Show your last 5 executed commands. clear \rightarrow Clear
 - the terminal.

Take screenshots of commands and outputs.

```
### Participations of Participation | Partici
```

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.

- 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 -I
```

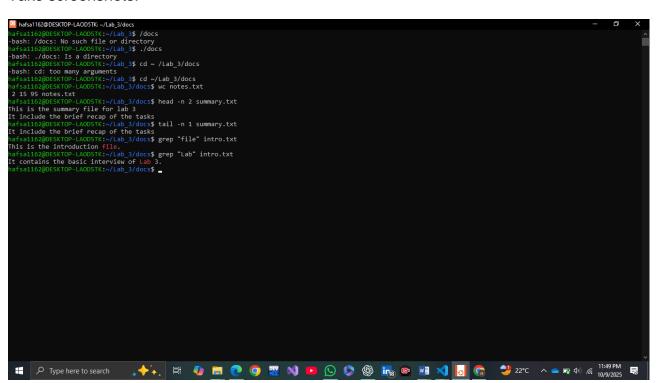
Take a screenshot of the output.

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.

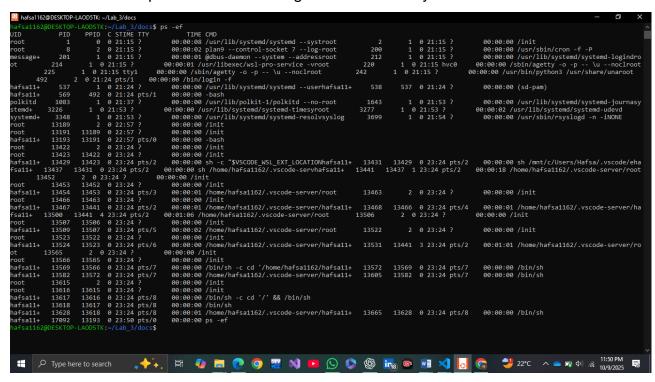
Take screenshots.

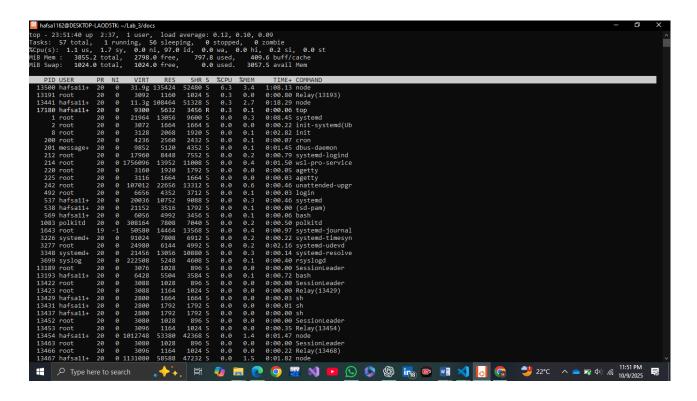


Part 5: Linux Process Commands

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

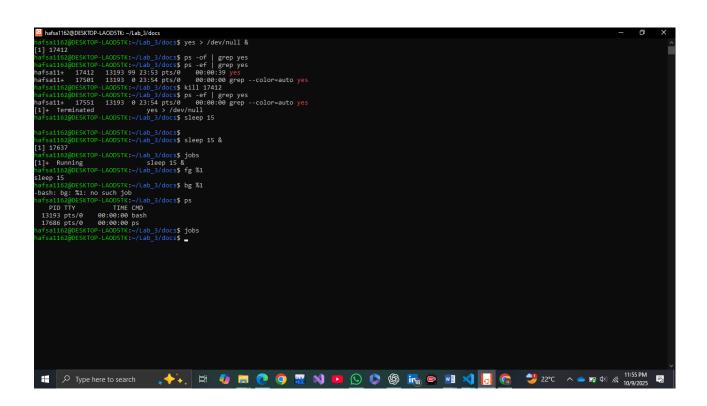
Start:

```
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 Is -1.
- 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

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {

    pid t pid = fork();

    if (pid == 0) {

        // TODO: Replace this child process with the "date" command using execlp

        // Hint: execlp("date", "date", NULL);
    } else {

        // TODO: Make parent wait for child before printing "Child finished" }

    return 0;
}
```

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

```
O: 🔳 🗖 🖽
                                                                                   C Task6.c X
                                                                                                                                                                                                                                          □ ...
                                                               Welcome
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
Q
         ≣ output1
         C Task6.c
ညီ
                                                                              main() {
pid_t pid = fork();
$
                                                                                  (pid == 0) {
瞪
                                                                                   // Child process: replace with "date" command
execlp("date", "date", NULL);
// If execlp fails:
<u>[</u>8
                                                                 11
12
13
14
                                                                                | pernor("execlp failed");
else if (pid > 0) {
    // Parent process: wait for child to finish
Ó
                                                                                        DEBUG CONSOLE TERMINAL PORTS
                                                                                                                                                                                             🍞 bash - HomeTask2 + ∨ 🖽 🗓 … │ 🖸 🗴
                                                                               ESKTOP-LAOD5TK:~/HomeTask2$ gcc Task6.c -o output1
                                                                                         stray '\302' in program
30A0><U+00A0><U+00A0>return 0;
                                                                                       or: stray (\302' in program
|\00000>\U+0000>\U+0000>return 0;
                                                                         :22:2:
                                                                                          stray '\302' in program
0A0><<del>U+00A0></del><U+00A0>return 0;
                                                                          :22:4: error: stray '\302' in program 
<U+00A0><U+00A0><U+00A0><U+00A0>return 0;
                                                                Task6.c:22:4:
                                                                           2@DESKTOP-LAOD5TK:~/HomeTask2$ gcc Task6.c -o output1
2@DESKTOP-LAOD5TK:~/HomeTask2$ ./output1
8
       > OUTLINE
> TIMELINE
                                                                                               K:~/HomeTask2$
> WSL: Ubuntu-24.04 ⊗ 0 △ 0 😾 0
                                                                                                                                                                            Ln 11, Col 28 Spaces: 2 UTF-8 LF {} C ♣ Finish Setup ♀

      22°C
      ^ ● ☞ Φ)
      12:09 AM 10/10/2025
      ■

                                                             Ħ 🥠 🥫 📀
                                                                                          🧿 💹 刘 🔼 🕓 🔇 🎼 🙉 🔟 刘 😭 👩
```

Submission Guidelines

- Submit a single PDF file including:
 - Screenshots of all said commands & outputs.
 - Modified & completed C program code and outputs.
- Deadline: 9th October, 2025, 11:59 PM.