National Textile University

Department of Computer Science

Subject: Operating System

Submitted to: Sir Nasir Mahmood

Submitted by: Maha

Reg. number: 23-NTU-CS-1170

Lab no.: lab3 (Home task)

Semester:5th

Lab 3 Home task

To make a folder of lab3_home task we use following commands

As I'm making all labs folder in 1170-lab-os which is in home folder so:

- Home > 1170_OS_LAB > lab3_hometask

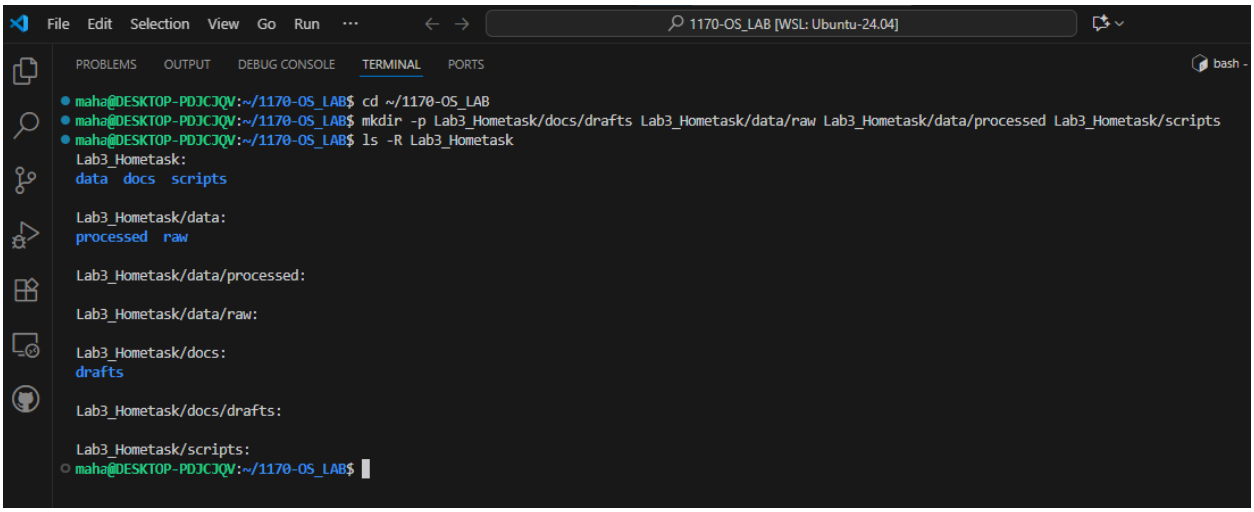
Step 1:

Commands:

- cd ~/1170-OS_LAB
- mkdir -p Lab3_Hometask/docs/drafts Lab3_Hometask/data/raw Lab3_Hometask/data/processed Lab3_Hometask/scripts

To verify folders and subfolder we just created

- ls -R Lab3_Hometask



Step2:

Create files and add text in each file using linux commands

open docx folder

- cd ~/1170-OS_LAB/Lab3_Hometask/docs

Create files

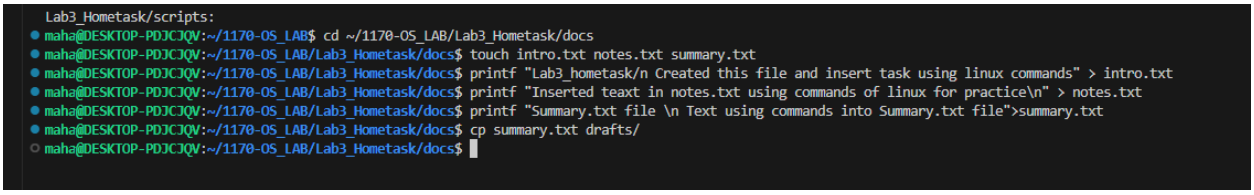
- touch intro.txt notes.txt summary.txt

Add text into each file:

- printf "Intro line 1\nIntro line 2\n" > intro.txt
- printf "Notes line 1\nNotes line 2\n" > notes.txt
- printf "Summary line 1\nSummary line 2\n" > summary.txt

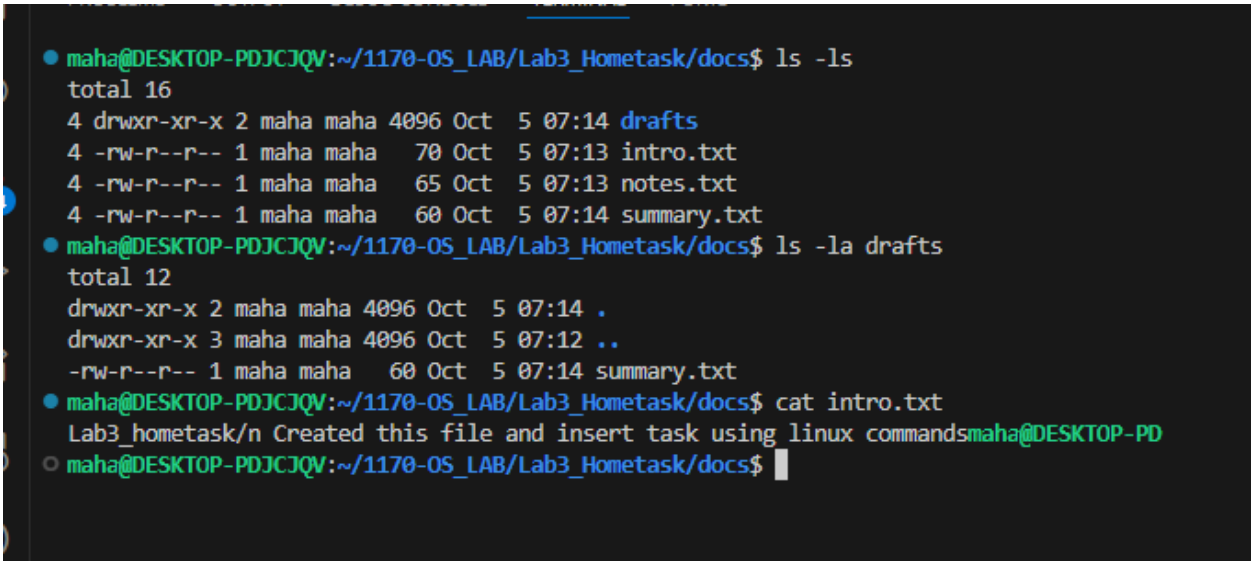
copy summary.txt into drafts folder:

- cp summary.txt drafts/



Verify creation of folders and text inside folders

- ls -la
- ls -la drafts
- cat intro.txt



Step 3:

Operations inside data/raw

- cd ~/1170-OS_LAB/Lab3_Hometask/data/raw

create files:

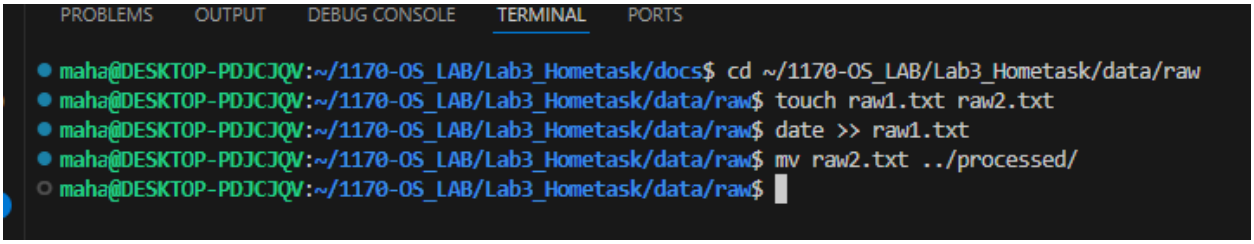
- touch raw1.txt raw2.txt

append current data to raw1.txt

- date >> raw1.txt

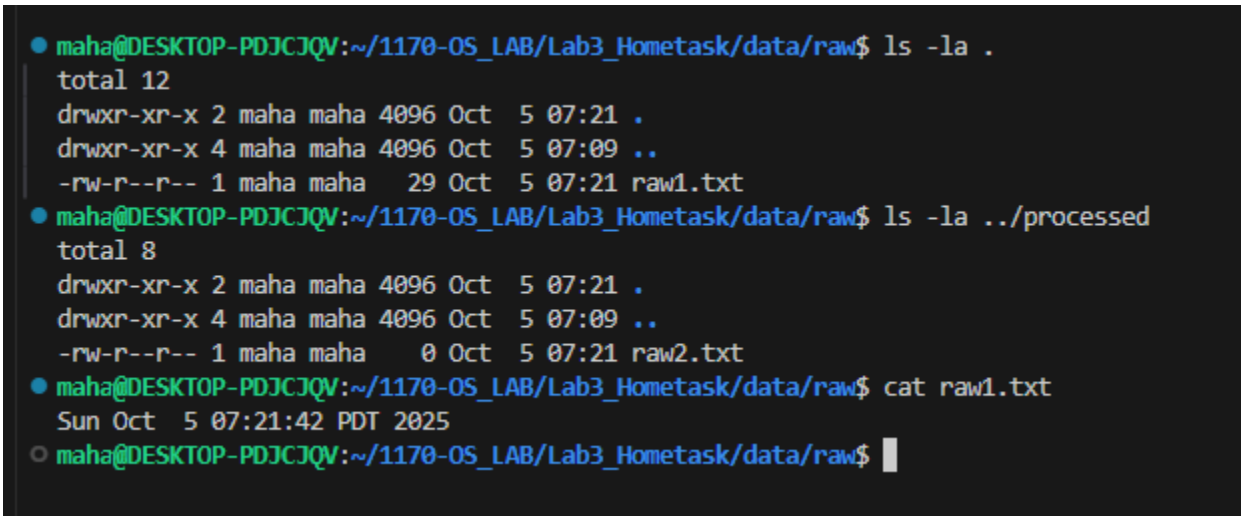
move raw2.txt to processed folder

- mv raw2.txt ../processed/



Verify operation in data/raw

- ls -la .
- ls -la ../processed
- cat raw1.txt



Step 4:

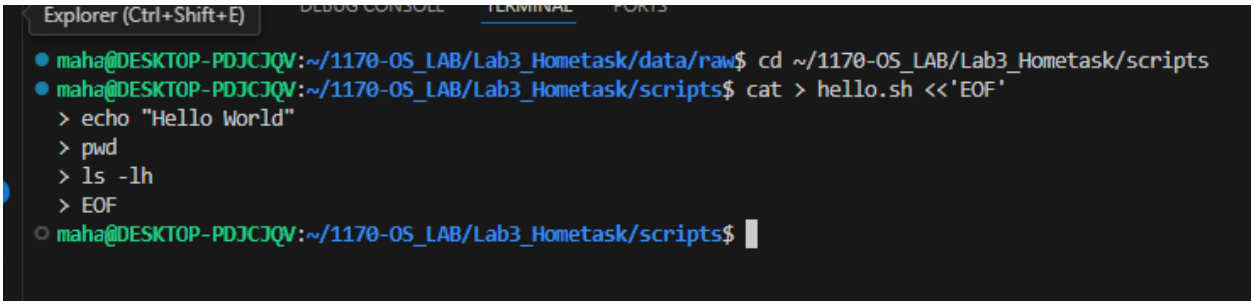
Inside scripts > Create a script named echo "Hello World”

Commands:

- cd ~/1170-OS_LAB/Lab3_Hometask/scripts

create hello.sh

- cat > hello.sh <<'EOF'
- echo "Hello World"
- pwd
- ls -lh
- EOF

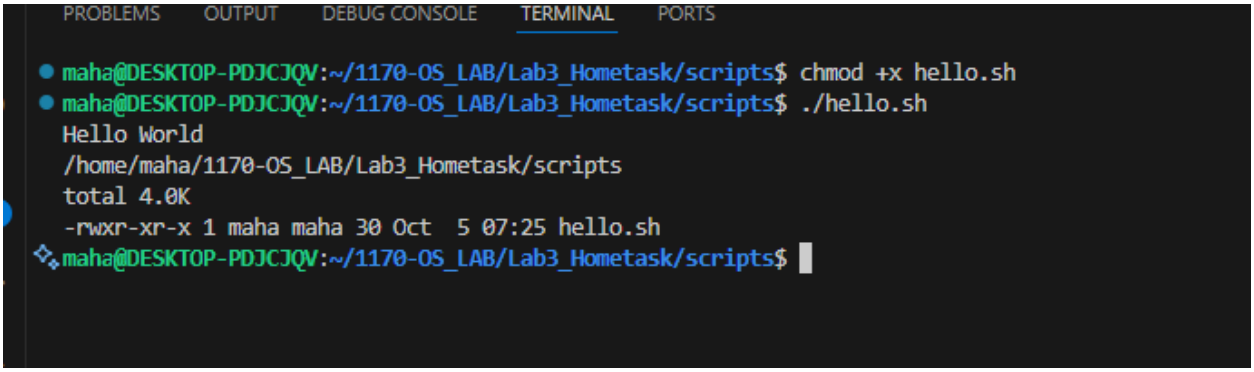


make executable

- chmod +x hello.sh

run

- ./hello.sh



Step 5:

Display the directory structure recursively

Commands:

- cd ~/1170-OS_LAB
- ls -R Lab3_Hometask

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/scripts$ cd ~/1170-OS_LAB
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ ls -R Lab3_Hometask
Lab3_Hometask:
data  docs  scripts

Lab3_Hometask/data:
processed  raw

Lab3_Hometask/data/processed:
raw2.txt

Lab3_Hometask/data/raw:
raw1.txt

Lab3_Hometask/docs:
drafts  intro.txt  notes.txt  summary.txt

Lab3_Hometask/docs/drafts:
summary.txt

Lab3_Hometask/scripts:
hello.sh
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$
```

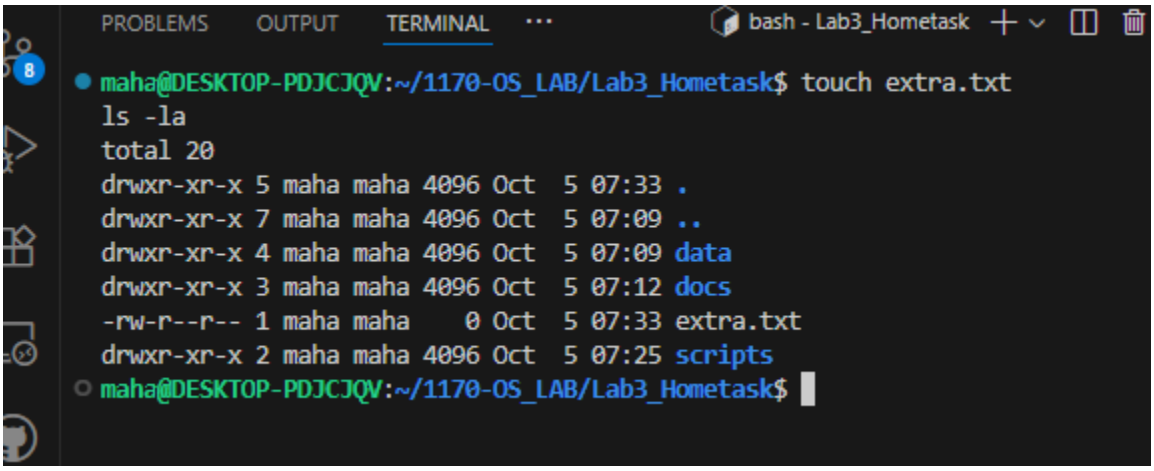
Part 2:

Run the following commands inside Lab_3/ and note their outputs

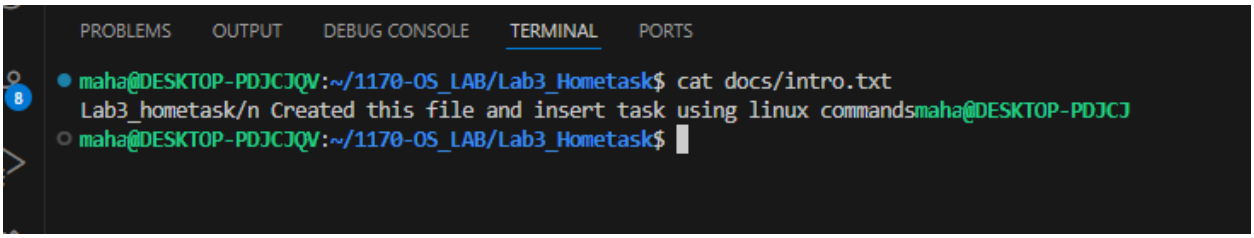
1. pwd
2. whoami

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask$ cd ~/1170-OS_LAB/Lab3_Hometask
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask$ pwd
/home/maha/1170-OS_LAB/Lab3_Hometask
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask$ whoami
maha
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask$
```

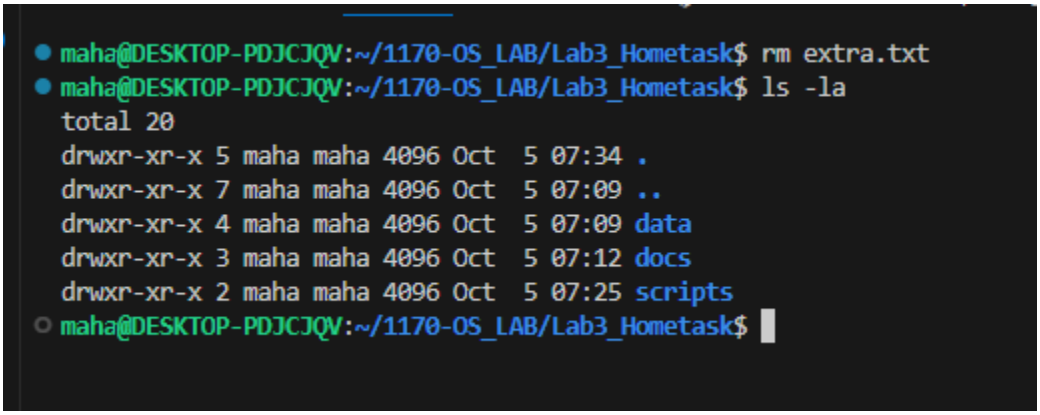
3. touch extra.txt



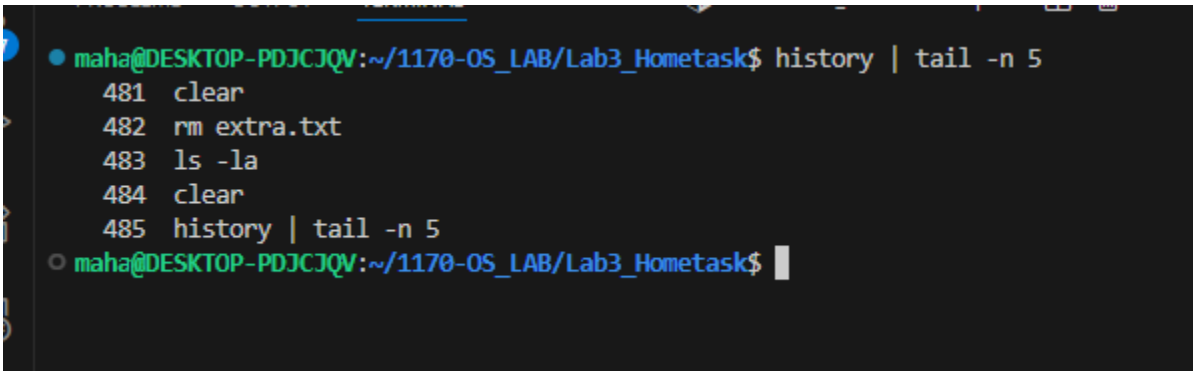
4. cat intro.txt



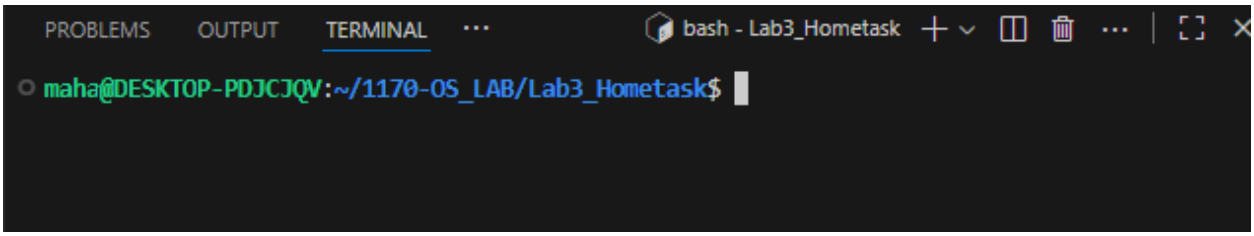
5. rm extra.txt



6. history | tail -n 5



7. clear



Part 3:

File Permissions and Ownerships

Step 1:

Commands:

- cd ~/1170-OS_LAB/Lab3_Hometask/scripts

Owner & Group → Read, Write

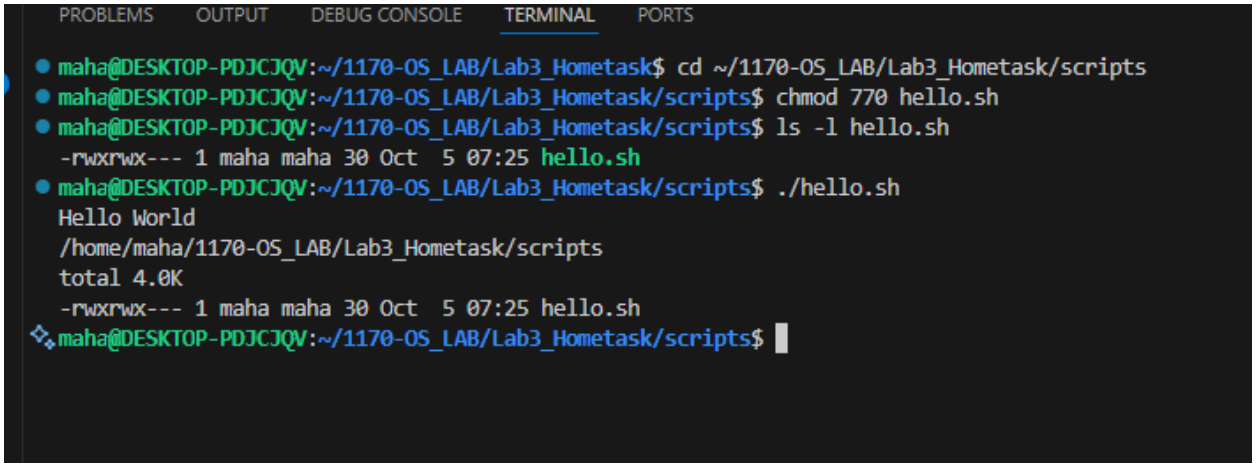
- chmod 770 hello.sh

verify

- ls -l hello.sh

run script

- ./hello.sh



Step 2:

Intro.txt permission

Commands:

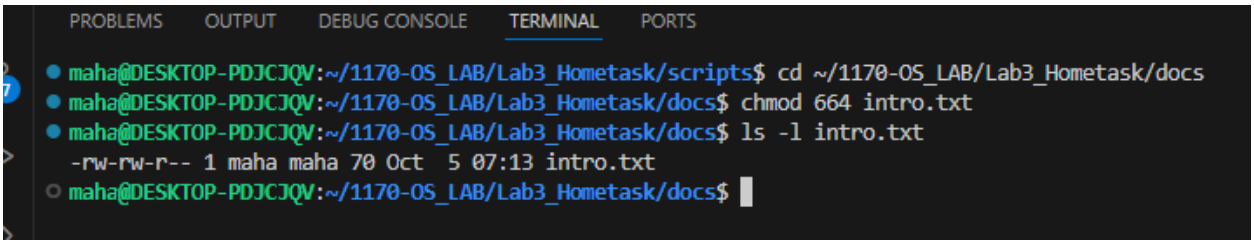
- cd ~/1170-OS_LAB/Lab3_Hometask/docs

Owner → rw, Group → rw, Others → r

- chmod 664 intro.txt

verify

- ls -l intro.txt



Step 3:

Notes.txt permission:

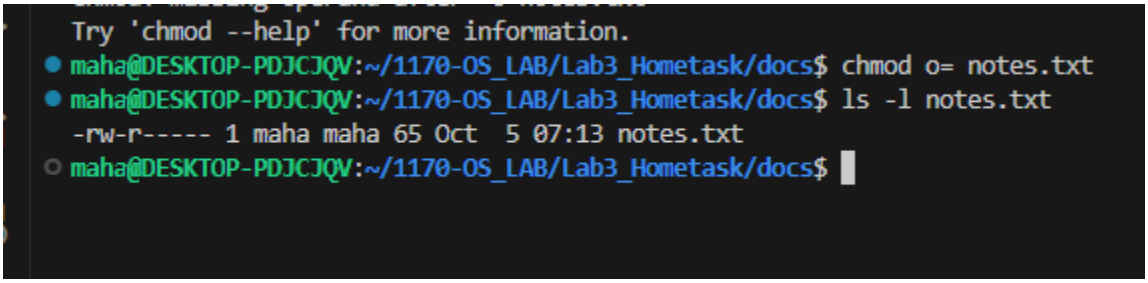
Commands:

remove all permissions for others

- chmod o= notes.txt

verify

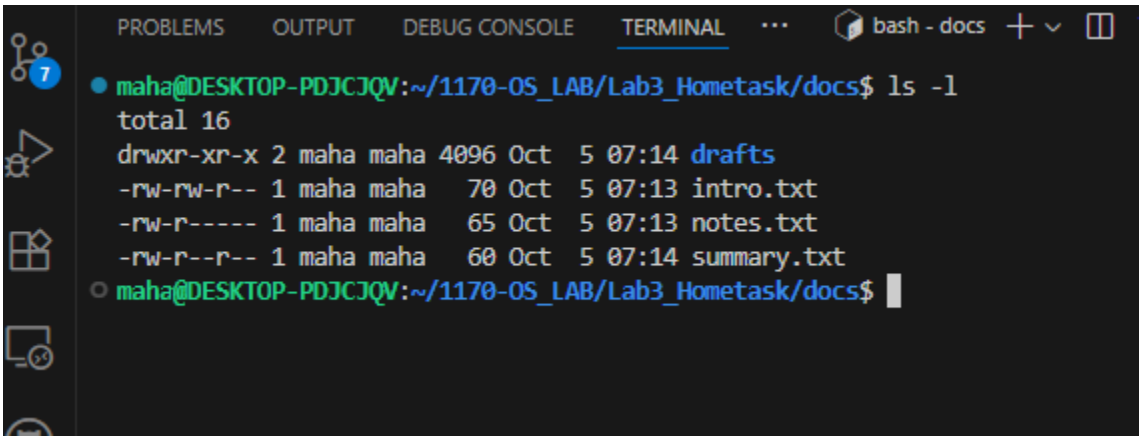
- ls -l notes.txt



Step 4:

Verify all changes

- ls -l



Part 4:

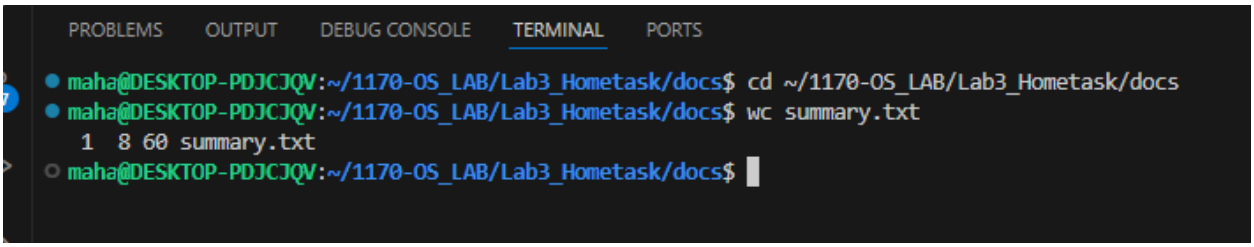
Reading and searching files:

Step 1:

Count lines, Words, Characters in summary.txt

Commands:

- cd ~/1170-OS_LAB/Lab3_Hometask/docs
- wc summary.txt

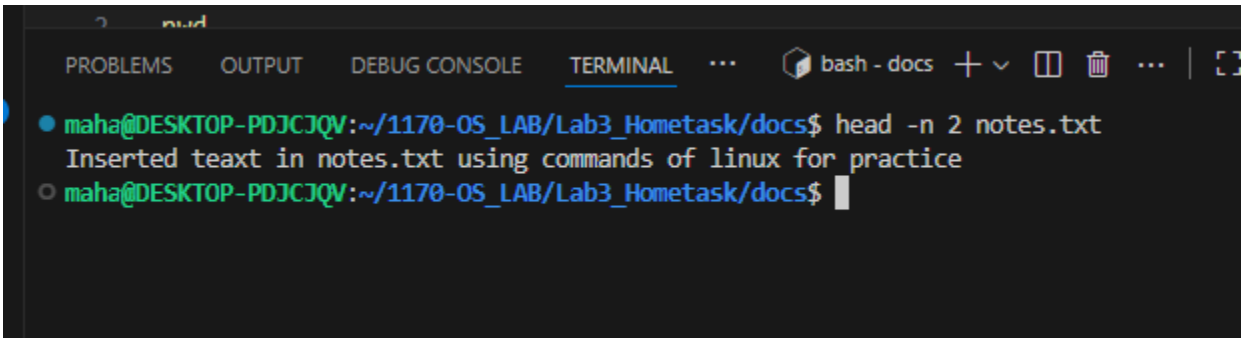


Step 2:

Shows only first two lines of notes.txt

Command:

- Head -n 2 notes.txt

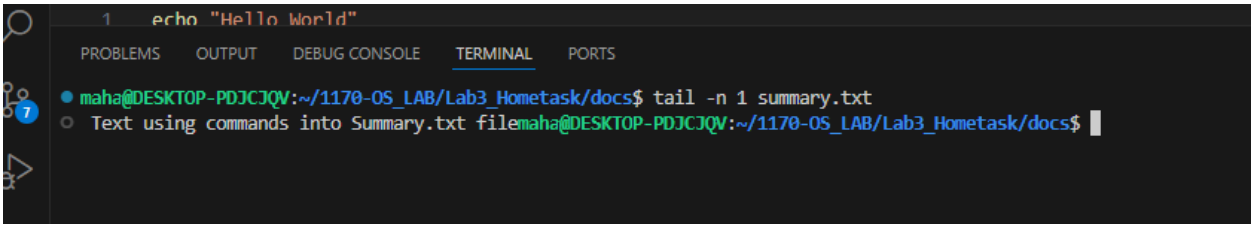


Step 3:

Show the last line of summary.txt

Command:

- Tail -n 1 summary.txt

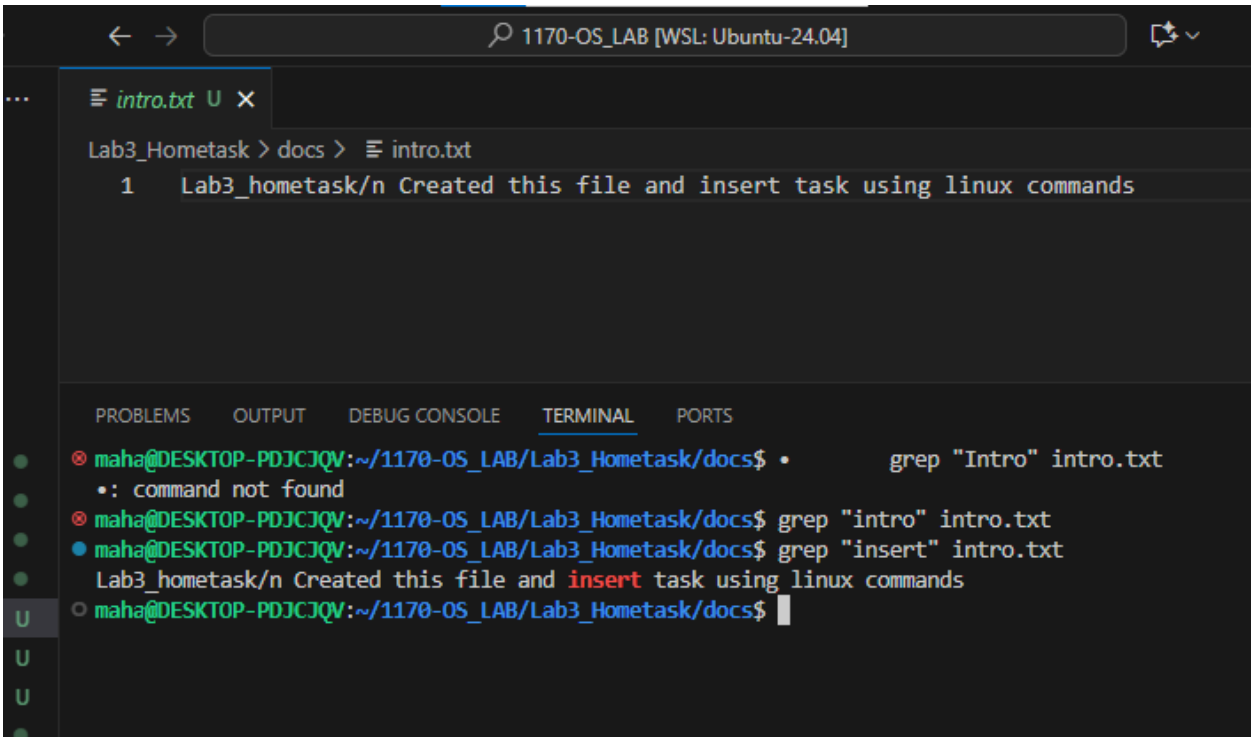


Step 4:

Search for a keyword in intro.txt

Command:

- grep "insert" intro.txt



Part 5:

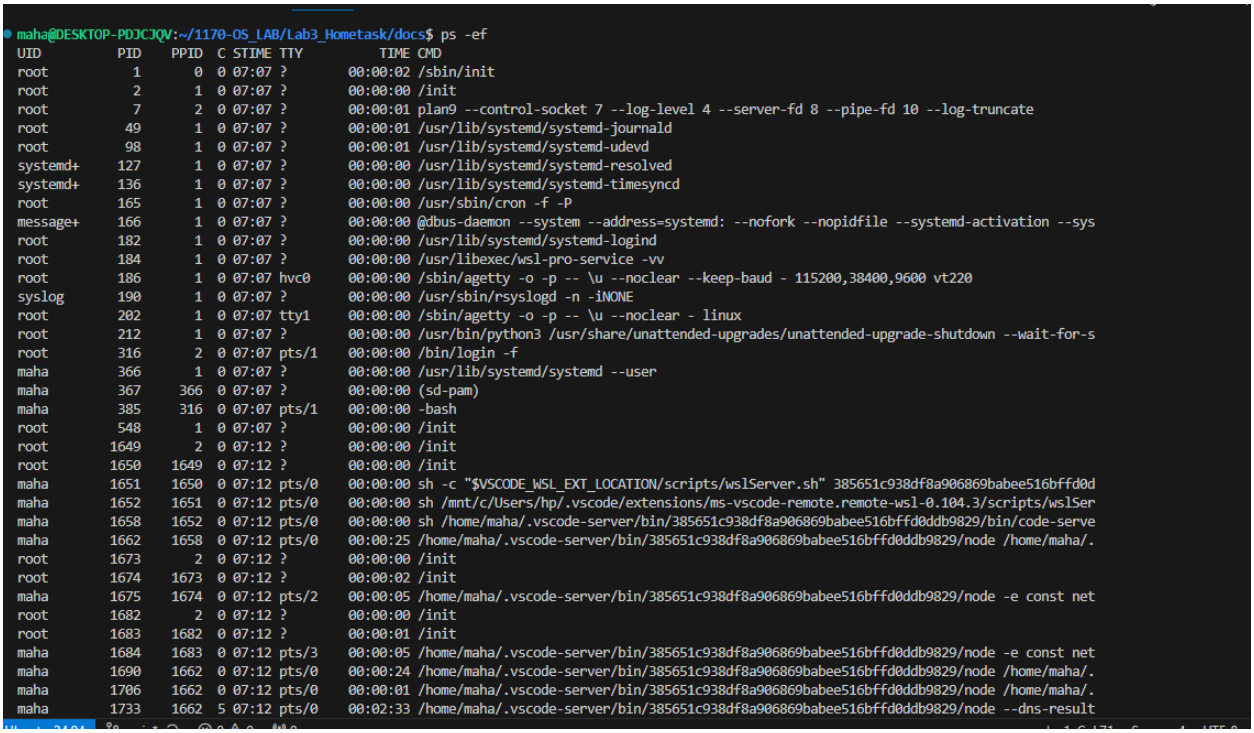
Linux Process Commands

Step 1:

commands:

show all processes with details

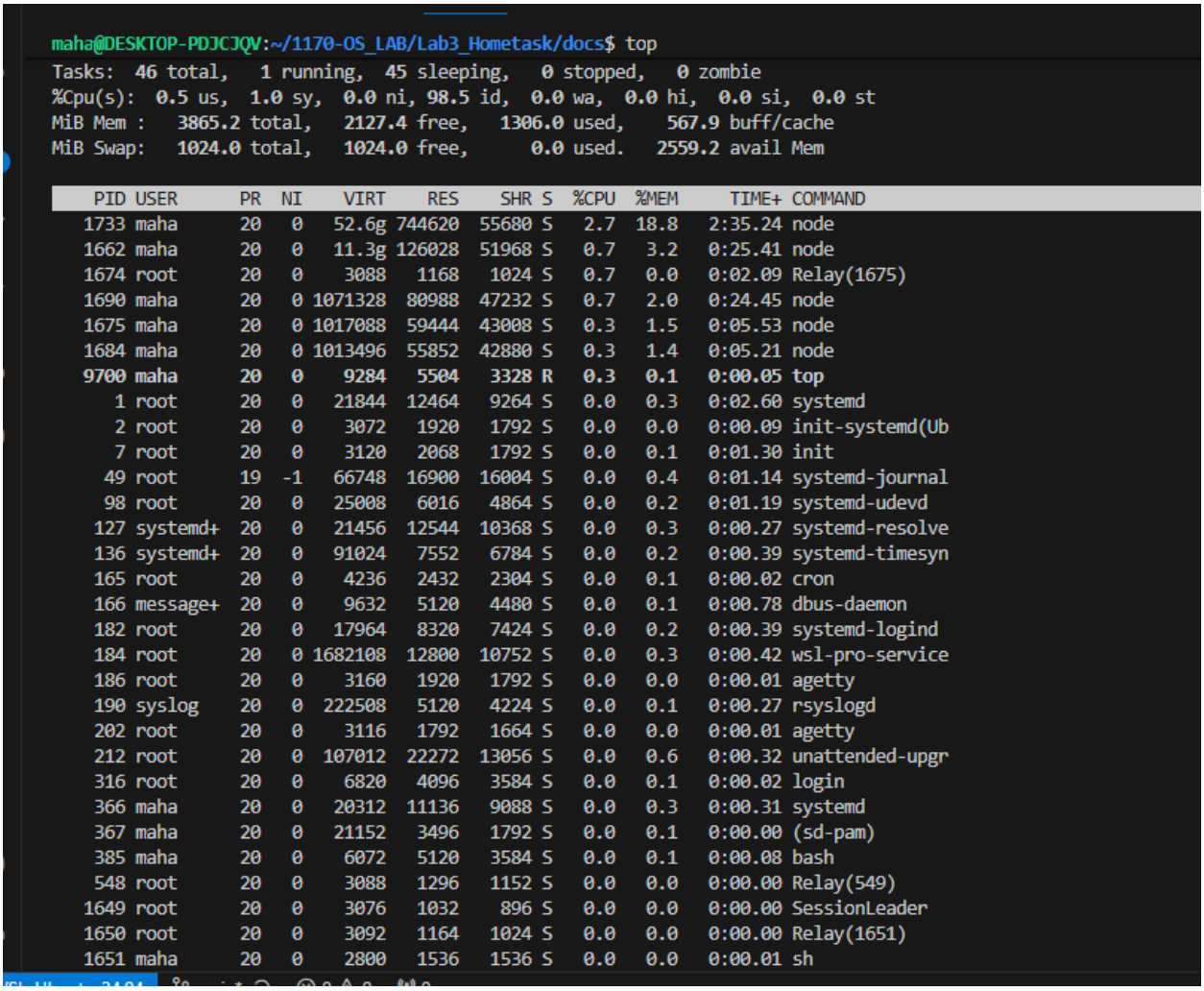
- ps -ef



```
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 07:07 ?        00:00:02 /sbin/init
root           2       1  0 07:07 ?        00:00:00 /init
root           7       2  0 07:07 ?        00:00:01 plan9 --control-socket 7 --log-level 4 --server-fd 8 --pipe-fd 10 --log-truncate
root          49       1  0 07:07 ?        00:00:01 /usr/lib/systemd/systemd-journald
root          98       1  0 07:07 ?        00:00:01 /usr/lib/systemd/systemd-udevd
systemd+     127       1  0 07:07 ?        00:00:00 /usr/lib/systemd/systemd-resolved
systemd+     136       1  0 07:07 ?        00:00:00 /usr/lib/systemd/systemd-timesyncd
root         165       1  0 07:07 ?        00:00:00 /usr/sbin/cron -f -P
message+     166       1  0 07:07 ?        00:00:00 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --sys
root         182       1  0 07:07 ?        00:00:00 /usr/lib/systemd/systemd-logind
root         184       1  0 07:07 ?        00:00:00 /usr/libexec/wsl-pro-service -vv
root         186       1  0 07:07 hvc0     00:00:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 vt220
syslog       190       1  0 07:07 ?        00:00:00 /usr/sbin/rsyslogd -n -iNONE
root         202       1  0 07:07 tty1     00:00:00 /sbin/agetty -o -p -- \u --noclear - linux
root         212       1  0 07:07 ?        00:00:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-s
root         316       2  0 07:07 pts/1    00:00:00 /bin/login -f
maha         366       1  0 07:07 ?        00:00:00 /usr/lib/systemd/systemd --user
maha         367     366  0 07:07 ?        00:00:00 (sd-pam)
maha         385     316  0 07:07 pts/1    00:00:00 -bash
root         548       1  0 07:07 ?        00:00:00 /init
root        1649       2  0 07:12 ?        00:00:00 /init
root        1650    1649  0 07:12 ?        00:00:00 /init
maha        1651    1650  0 07:12 pts/0    00:00:00 sh -c "$VSCODE_WSL_EXT_LOCATION/scripts/wslServer.sh" 385651c938df8a906869babee516bffd0d
maha        1652    1651  0 07:12 pts/0    00:00:00 sh /mnt/c/Users/hp/.vscode/extensions/ms-vscode-remote.remote-wsl-0.104.3/scripts/wslSer
maha        1658    1652  0 07:12 pts/0    00:00:00 sh /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/bin/code-serve
maha        1662    1658  0 07:12 pts/0    00:00:25 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node /home/maha/.
root        1673       2  0 07:12 ?        00:00:00 /init
root        1674    1673  0 07:12 ?        00:00:02 /init
maha        1675    1674  0 07:12 pts/2    00:00:05 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node -e const net
root        1682       2  0 07:12 ?        00:00:00 /init
root        1683    1682  0 07:12 ?        00:00:01 /init
maha        1684    1683  0 07:12 pts/3    00:00:05 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node -e const net
maha        1690    1662  0 07:12 pts/0    00:00:24 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node /home/maha/.
maha        1706    1662  0 07:12 pts/0    00:00:01 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node /home/maha/.
maha        1733    1662  5 07:12 pts/0    00:02:33 /home/maha/.vscode-server/bin/385651c938df8a906869babee516bffd0ddb9829/node --dns-result
```

interactive process monitor

- top



Step 2:

Practice with infinite process

Commands:

Start infinite process

- yes > /dev/null &

Check process id of “yes”

- ps -ef | grep yes

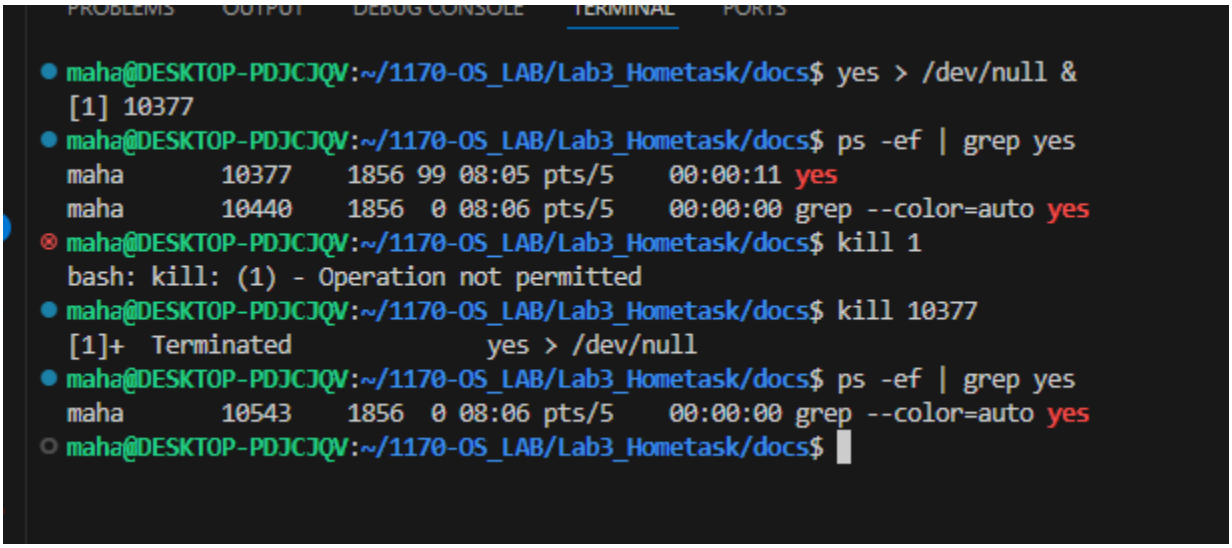
kill the process

- kill <Pid>

- kill

verify killing

- ps -ef | grep yes



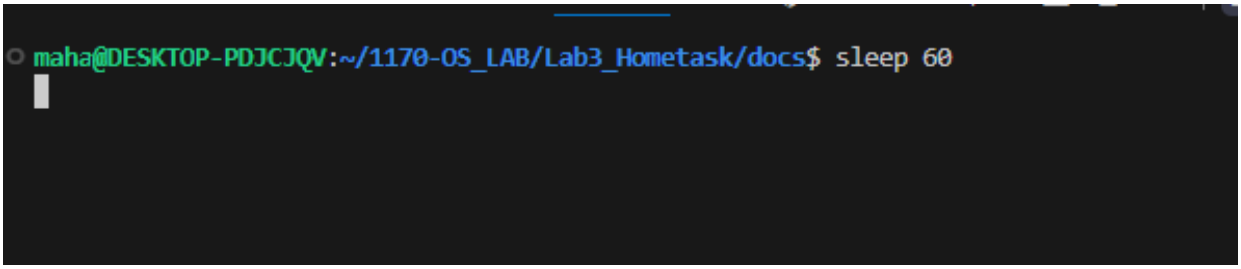
Step 3:

Foreground & Background Jobs

Commands:

Foreground:

- Sleep 60



background

- Sleep 60 &

```
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ sleep 60
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ sleep 60 &
[1] 10991
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$
```

Bring it to foreground

- Fg

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  ···  sleep - docs
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ sleep 60
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ sleep 60 &
[1] 10991
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ fg
sleep 60
```

Send it to background

- Bg

```
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ sleep 10 &
[1] 11238
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$ bg
bash: bg: job 1 already in background
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/Lab3_Hometask/docs$
```

Part 6:

Program 1:

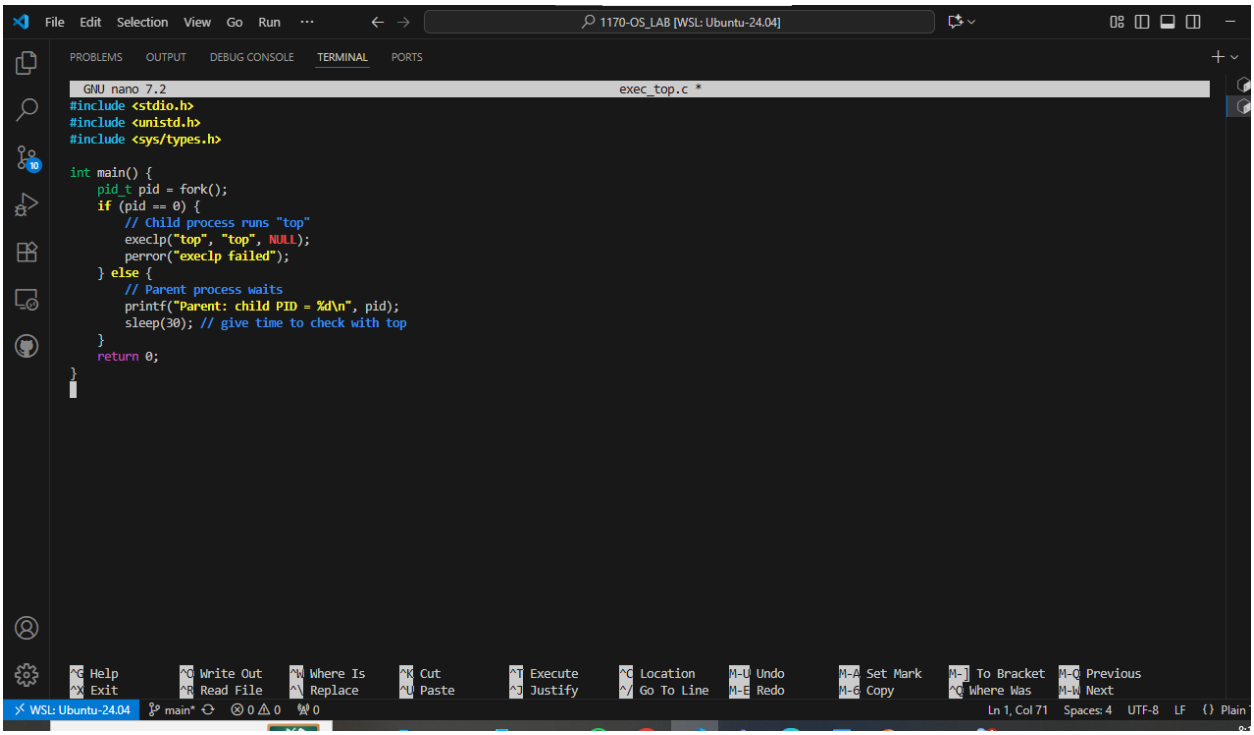
Commands:

Create new c file

- cd ~/1170-OS_LAB/Lab3_Hometask
- nano exec_top.c

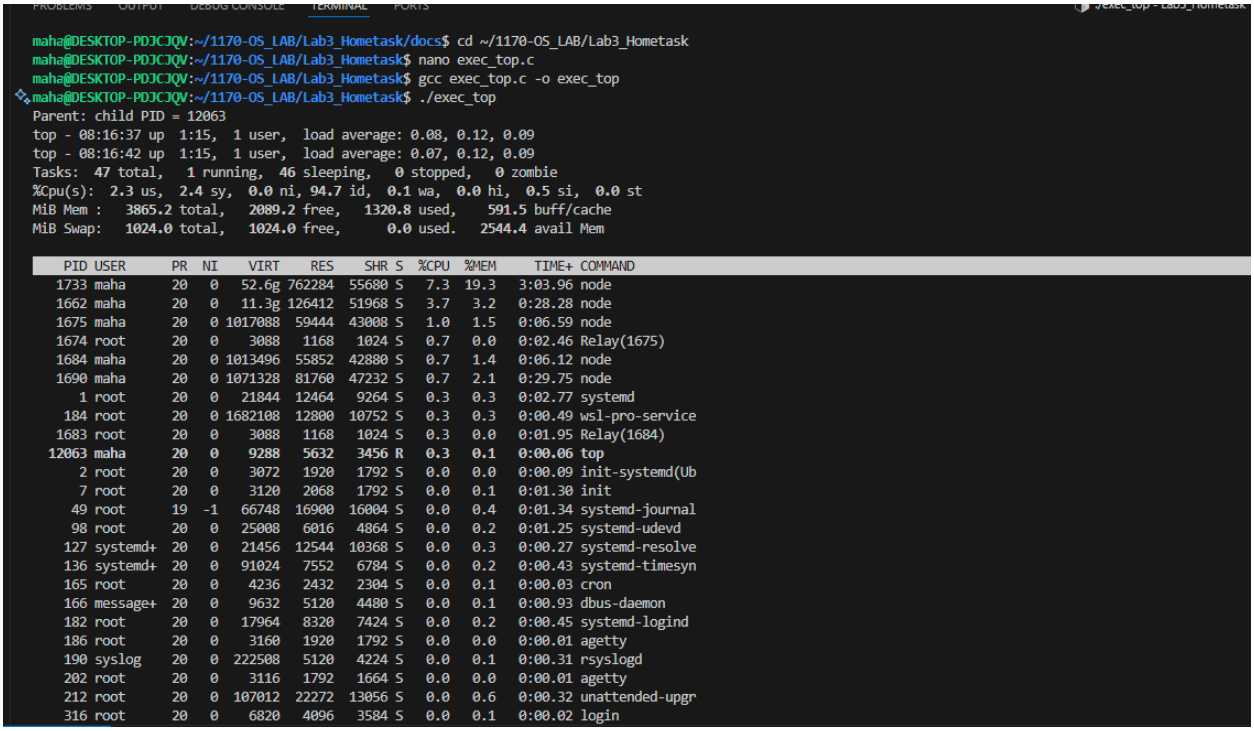
code for c file:

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


int main() {
   pid_t pid = fork();
   if (pid == 0) {
      // Child process runs "top"
      execlp("top", "top", NULL);
      perror("execlp failed");
   } else {
      // Parent process waits
      printf("Parent: child PID = %d\n", pid);
      sleep(30); // give time to check with top
   }
   return 0;
}
```



Compile and run commands:

- gcc exec_top.c -o exec_top
- ./exec_top

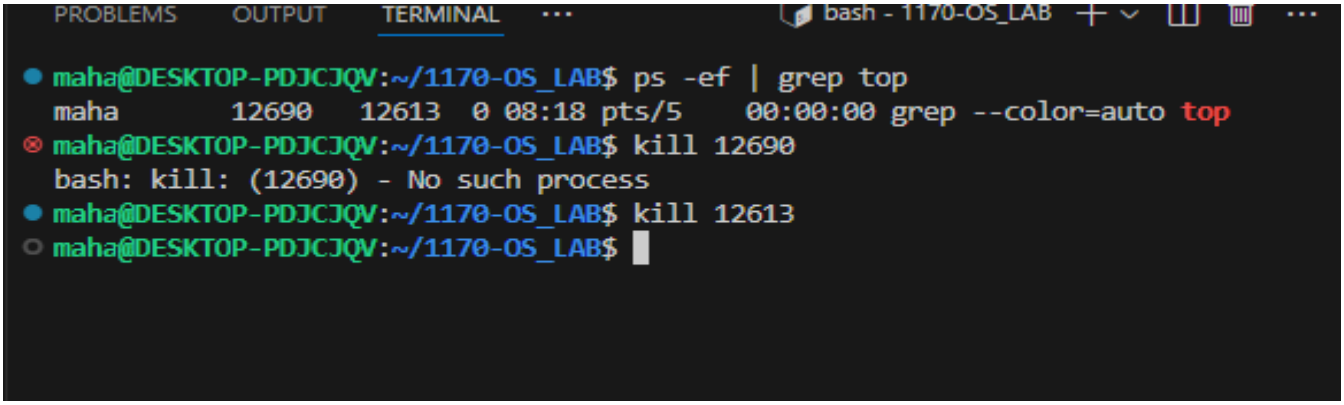


Open new terminal and run:

- ps -ef | grep top

kill child manually:

- kill <PID>
- kill 12613



Step 2:

Incomplete Program

Commands:

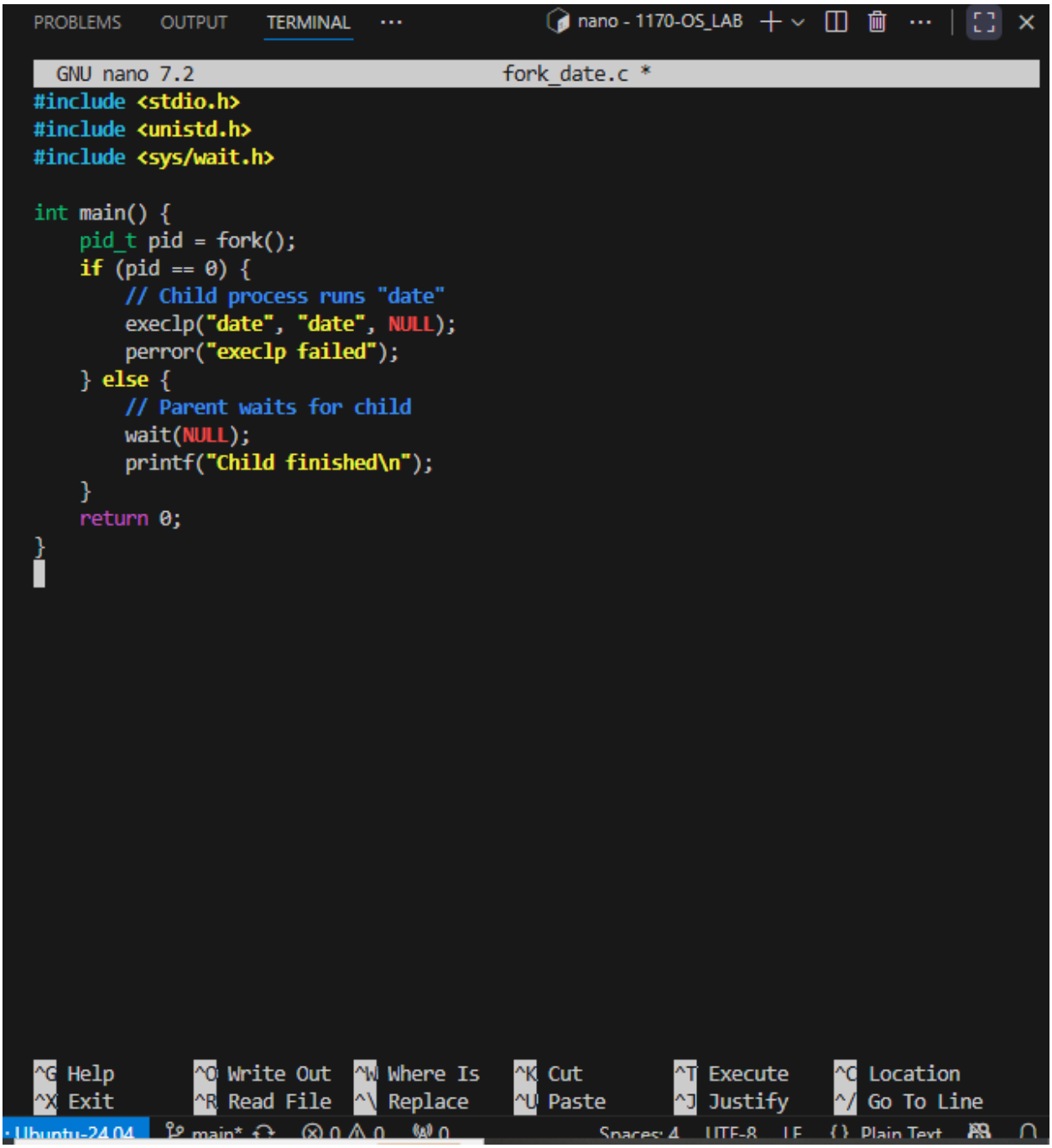
Create new file:

- nano fork_date.c

code:

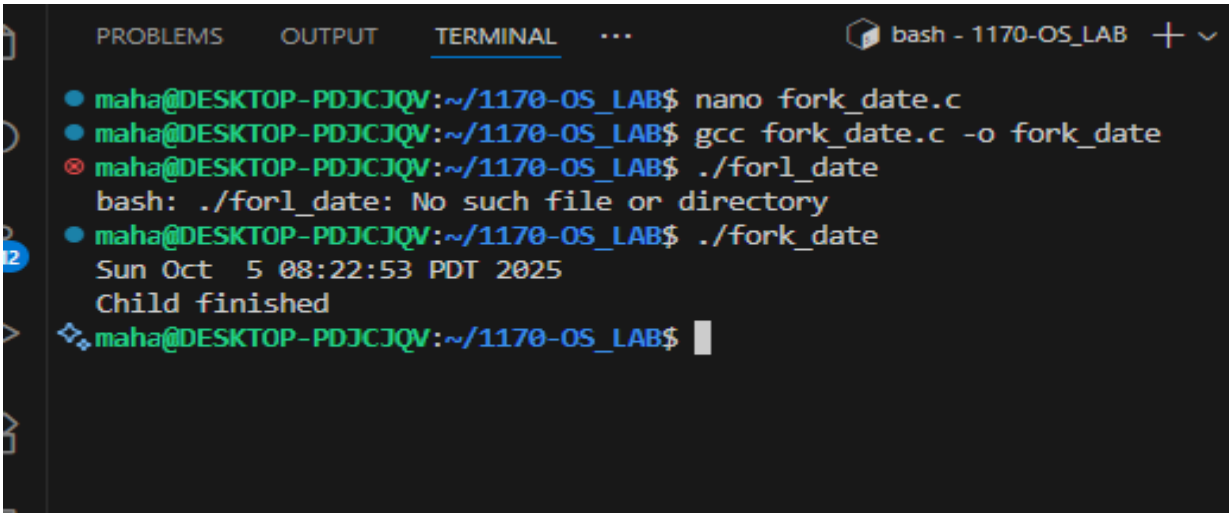
```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork();
```

```
    if (pid == 0) {
        // Child process runs "date"
        execlp("date", "date", NULL);
        perror("execlp failed");
    } else {
        // Parent waits for child
        wait(NULL);
        printf("Child finished\n");
    }
    return 0;
}
```



Compile and run

- gcc fork_date.c -o fork_date
- ./fork_date


PROBLEMS
OUTPUT
TERMINAL
bash - 1170-OS_LAB
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ nano fork_date.c
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ gcc fork_date.c -o fork_date
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ ./forl_date
bash: ./forl_date: No such file or directory
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ ./fork_date
Sun Oct 5 08:22:53 PDT 2025
Child finished
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$

.....