



National Textile University
Department of Computer Science

Subject:

Operating system

Submitted to:

Dr. Nasir Mehmood

Submitted by:

Dawar Abbas

Reg number:

Lab no. :

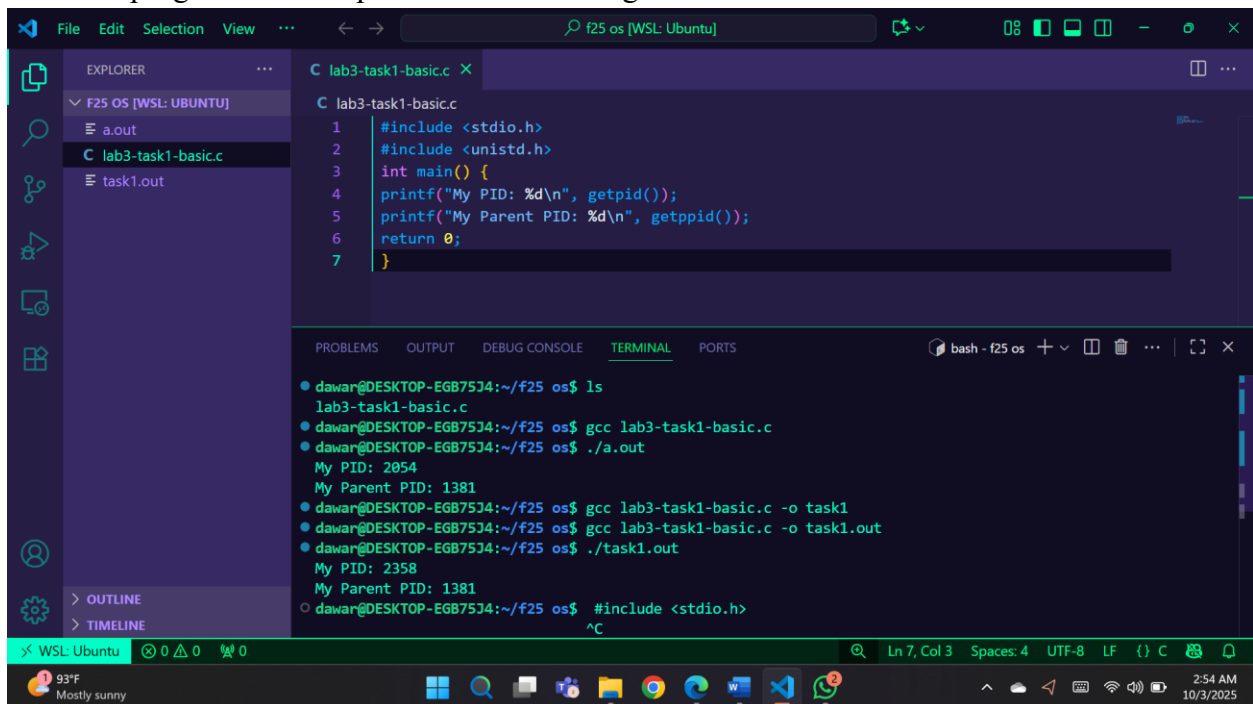
3

Semester:

5th

Lab 3

Write a c program and compile it and run it using command ./a.out:



The screenshot shows the Visual Studio Code interface with a C program named `lab3-task1-basic.c` open in the editor. The program's content is as follows:

```
1 #include <stdio.h>
2 #include <unistd.h>
3 int main() {
4     printf("My PID: %d\n", getpid());
5     printf("My Parent PID: %d\n", getppid());
6     return 0;
7 }
```

The terminal window at the bottom displays the execution of the program. The commands and their outputs are:

```
dawar@DESKTOP-EG875J4:~/f25 os$ ls
lab3-task1-basic.c
dawar@DESKTOP-EG875J4:~/f25 os$ gcc lab3-task1-basic.c
dawar@DESKTOP-EG875J4:~/f25 os$ ./a.out
My PID: 2054
My Parent PID: 1381
dawar@DESKTOP-EG875J4:~/f25 os$ gcc lab3-task1-basic.c -o task1
dawar@DESKTOP-EG875J4:~/f25 os$ gcc lab3-task1-basic.c -o task1.out
dawar@DESKTOP-EG875J4:~/f25 os$ ./task1.out
My PID: 2358
My Parent PID: 1381
dawar@DESKTOP-EG875J4:~/f25 os$ #include <stdio.h>
^C
```

The status bar at the bottom indicates the file is in the `f25 os` workspace, with the cursor at line 7, column 3. The system tray shows the temperature as 93°F and the date as 10/3/2025.

Getting parent and child of processes:

The screenshot shows a Visual Studio Code editor with the file 'simple-classes.c' open. The code is as follows:

```
1 #include <stdio.h>
2 #include <unistd.h>
3 int main() {
4     pid_t pid = fork();
5     if (pid == 0) {
6         // This block runs in the child process
7         printf("Child: PID=%d, Parent=%d\n", getpid(), getppid());
8     } else {
9         // This block runs in the parent process
10        printf("Parent: PID=%d, Child=%d\n", getpid(), pid);
11    }
12    return 0;
13 }
```

The terminal output shows the execution of the program:

```
My Parent PID: 1381
dawar@DESKTOP-EG875J4:~/f25 os$ #include <stdio.h>
^C
dawar@DESKTOP-EG875J4:~/f25 os$ gcc simple-classes.c -o classes.out
dawar@DESKTOP-EG875J4:~/f25 os$ ./classes.out
Parent: PID=3536, Child=3537
Child: PID=3537, Parent=3536
dawar@DESKTOP-EG875J4:~/f25 os$ gcc p-execute.c -o execute.out
dawar@DESKTOP-EG875J4:~/f25 os$ ./execute.out
Parent still running...
total 76
```

In the third program we have to execute a program with fork and execute.

The screenshot shows a Visual Studio Code editor with the file 'p-execute.c' open. The code is as follows:

```
3 int main() {
4     pid_t pid = fork();
5
6     if (pid == 0) {
7         execlp("ls", "ls", "-l", NULL);
8     } else {
9         printf("This will not print if exec succeeds.\n");
10    }
11    printf("Parent still running...\n");
12    return 0;
13 }
```

The terminal output shows the execution of the program:

```
dawar@DESKTOP-EG875J4:~/f25 os$ gcc p-execute.c -o execute.out
dawar@DESKTOP-EG875J4:~/f25 os$ ./execute.out
Parent still running...
total 76
-rwxr-xr-x 1 dawar dawar 16064 Oct 3 02:42 a.out
-rwxr-xr-x 1 dawar dawar 16096 Oct 3 02:57 classes.out
-rwxr-xr-x 1 dawar dawar 16048 Oct 3 03:02 execute.out
-rw-r--r-- 1 dawar dawar 146 Oct 3 02:37 lab3-task1-basic.c
-rw-r--r-- 1 dawar dawar 242 Oct 3 03:02 p-execute.c
-rw-r--r-- 1 dawar dawar 315 Oct 3 02:56 simple-classes.c
-rwxr-xr-x 1 dawar dawar 16064 Oct 3 02:44 task1.out
dawar@DESKTOP-EG875J4:~/f25 os$ gcc p-wait.c -o wait
dawar@DESKTOP-EG875J4:~/f25 os$ ./wait
```

In Q3 and Q4 the difference is that because of wait command in output parent will run in Q3 but do not wait for the child to complete but in Q4 the parent run after the child is completed because

we have used wait command.

```
1 #include <stdio.h>
5 pid_t pid = fork();
6 if (pid == 0) {
7     execlp("ls", "ls", "-l", NULL);
8     printf("This will not print if exec succeeds.\n");
9 } else {
10    waitpid(pid, NULL, 0); // Wait for the child process to finish
11    printf("Parent still running...\n");
12 }
13 return 0;
14 }
```

total 96
-rwxr-xr-x 1 dawar dawar 16064 Oct 3 02:42 a.out
-rwxr-xr-x 1 dawar dawar 16096 Oct 3 02:57 classes.out
-rwxr-xr-x 1 dawar dawar 16048 Oct 3 03:02 execute.out
-rw-r--r-- 1 dawar dawar 146 Oct 3 02:37 lab3-task1-basic.c
-rw-r--r-- 1 dawar dawar 242 Oct 3 03:02 p-execute.c
-rw-r--r-- 1 dawar dawar 327 Oct 3 03:06 p-wait.c
-rw-r--r-- 1 dawar dawar 315 Oct 3 02:56 simple-classes.c
-rwxr-xr-x 1 dawar dawar 16064 Oct 3 02:44 task1.out
-rwxr-xr-x 1 dawar dawar 16088 Oct 3 03:07 wait
Parent still running...