National Textile University

Department of Computer Science

Subject:

Operating system

Submitted to:

Sir Nasir Mahmood

Submitted by:

Areeba Tariq

Reg number:

23-NTU-CS-1139

Lab no:

3rd

Semester:

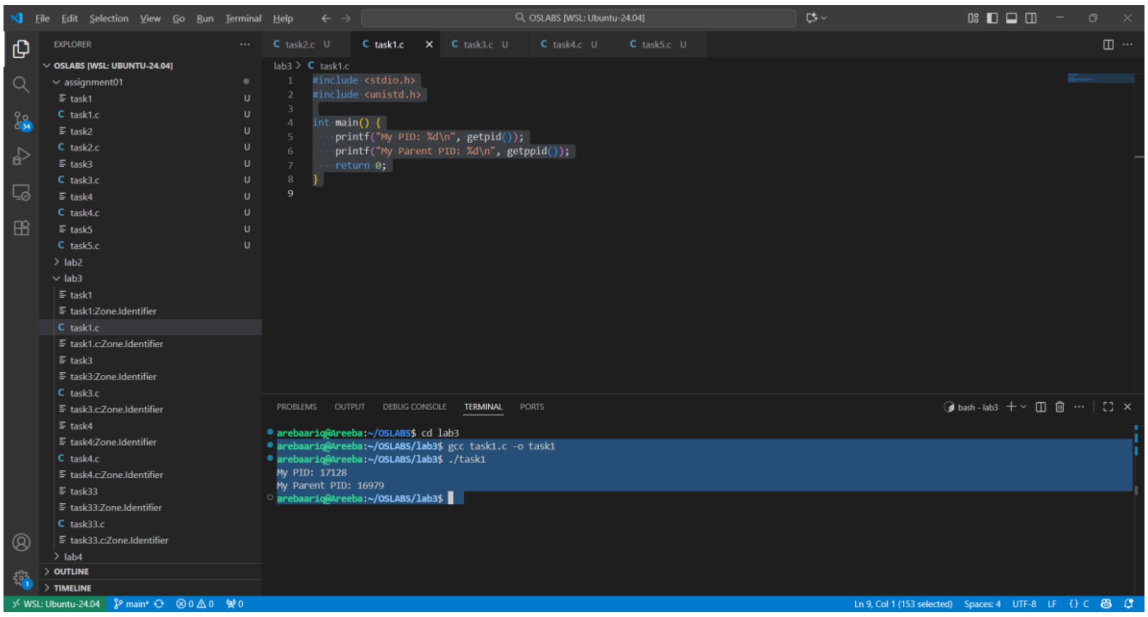
5th

TASK1:
CODE:

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

int main() {
    printf("My PID: %d\n", getpid());
    printf("My Parent PID: %d\n", getppid());
    return 0;
}
```

Output:



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

int main() {
    printf("My PID: %d\n", getpid());
    printf("My Parent PID: %d\n", getppid());
    return 0;
}
```

```
arebaariq@Areeba:~/OSLABS$ cd lab3
arebaariq@Areeba:~/OSLABS/lab3$ gcc task1.c -o task1
arebaariq@Areeba:~/OSLABS/lab3$ ./task1
My PID: 17128
My Parent PID: 16979
arebaariq@Areeba:~/OSLABS/lab3$
```

Remarks:
This program prints its own **Process ID (PID)** and the **Parent Process ID (PPID)** using Linux system calls.

TASK2:
CODE:

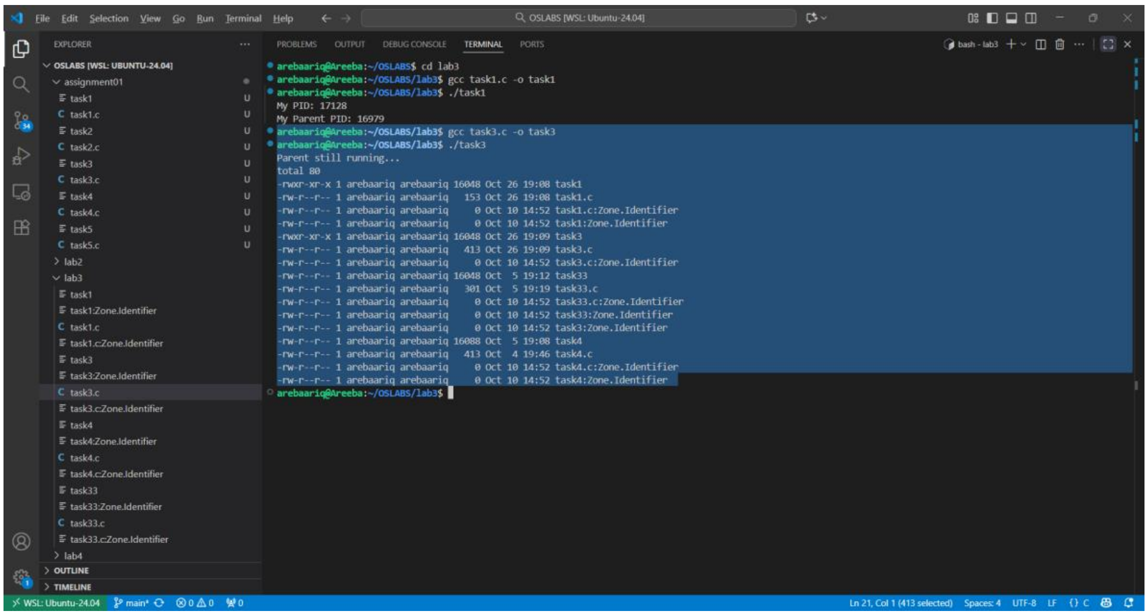
```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>   // for pid_t

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

    if (pid == 0) {
        // Child process
        execlp("ls", "ls", "-l", NULL);

        // Agar exec fail hua to yeh chalega
        printf("This will not print if exec succeeds.\n");
    } else {
        // Parent process
        printf("Parent still running...\n");
    }
    return 0;
}
```

Output:



```
arebaariq@Areeba:~/OSLABS$ cd lab3
arebaariq@Areeba:~/OSLABS/lab3$ gcc task1.c -o task1
arebaariq@Areeba:~/OSLABS/lab3$ ./task1
My PID: 17128
My Parent PID: 16979
arebaariq@Areeba:~/OSLABS/lab3$ gcc task3.c -o task3
arebaariq@Areeba:~/OSLABS/lab3$ ./task3
Parent still running...
total 80
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:08 task1
-rw-r--r-- 1 arebaariq arebaariq   153 Oct 26 19:08 task1.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:09 task3
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:09 task3.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16048 Oct  5 19:12 task33
-rw-r--r-- 1 arebaariq arebaariq   301 Oct  5 19:19 task33.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16088 Oct  5 19:08 task4
-rw-r--r-- 1 arebaariq arebaariq   413 Oct  4 19:46 task4.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4:Zone.Identifier
arebaariq@Areeba:~/OSLABS/lab3$
```

Remarks:
The program creates a child process using fork().
Child runs the **ls -l** command using execlp(), while the parent prints **"Parent still running..."**.

TASK3:
CODE:

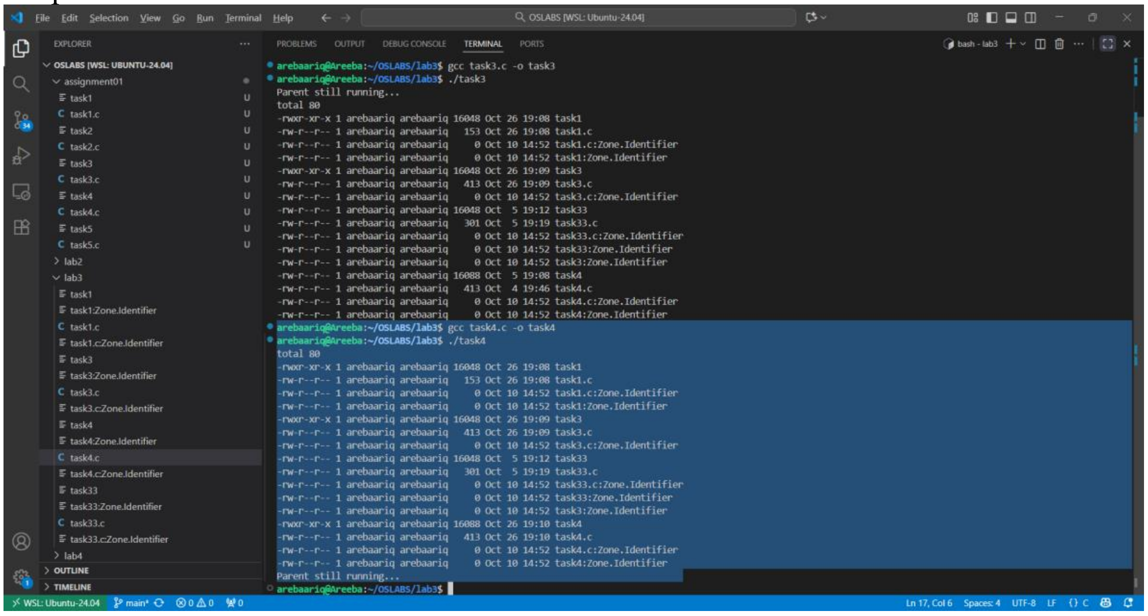
```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

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

    if (pid == 0) {
        // Child process
        execlp("ls", "ls", "-l", NULL);
        printf("This will not print if exec succeeds.\n");
    } else {
        // Parent waits for child
        waitpid(pid, NULL, 0);
        printf("Parent still running...\n");
    }

    return 0;
}
```

Output:



```
arebaariq@Areeba:~/OSLABS/lab3$ gcc task3.c -o task3
arebaariq@Areeba:~/OSLABS/lab3$ ./task3
Parent still running...
total 80
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:08 task1
-rw-r--r-- 1 arebaariq arebaariq   153 Oct 26 19:08 task1.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:09 task3
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:09 task3.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16048 Oct  5 19:12 task33
-rw-r--r-- 1 arebaariq arebaariq   301 Oct  5 19:19 task33.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16088 Oct  5 19:08 task4
-rw-r--r-- 1 arebaariq arebaariq   413 Oct  4 19:46 task4.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4:Zone.Identifier
arebaariq@Areeba:~/OSLABS/lab3$ gcc task4.c -o task4
arebaariq@Areeba:~/OSLABS/lab3$ ./task4
total 80
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:08 task1
-rw-r--r-- 1 arebaariq arebaariq   153 Oct 26 19:08 task1.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:09 task3
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:09 task3.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16048 Oct  5 19:12 task33
-rw-r--r-- 1 arebaariq arebaariq   301 Oct  5 19:19 task33.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16088 Oct 26 19:10 task4
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:10 task4.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4:Zone.Identifier
Parent still running...
arebaariq@Areeba:~/OSLABS/lab3$
```

Remarks:
After fork(), the child again runs **ls -l**.
The parent uses waitpid() to **wait for the child to finish**, then prints a message.

TASK4:
CODE:

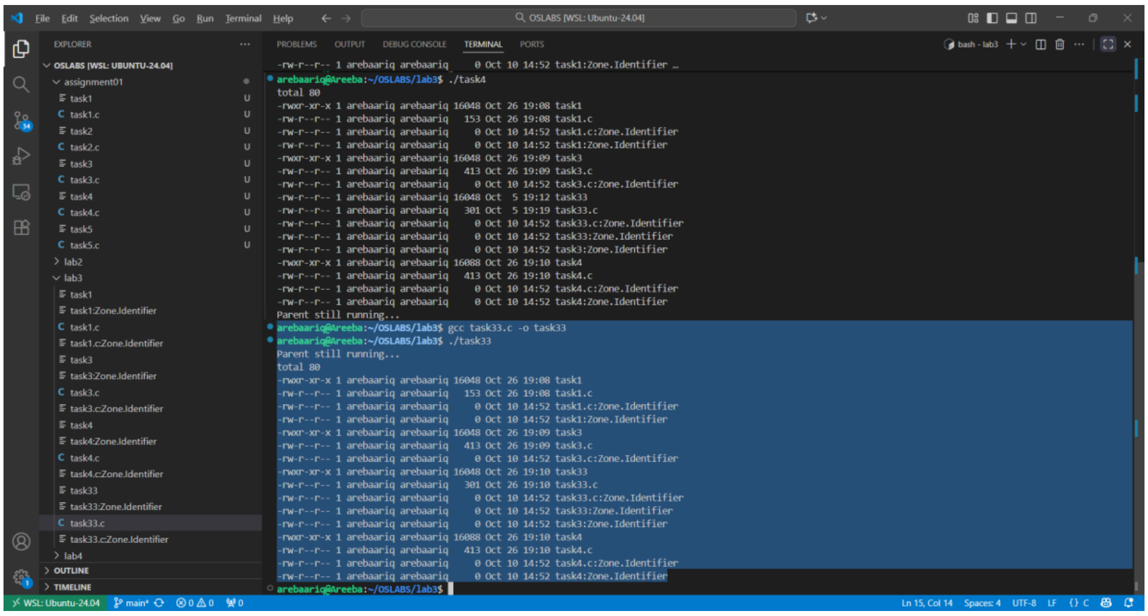
```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

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

    if (pid == 0) {
        execlp("ls", "ls", "-l", NULL);
        printf("This will not print if exec succeeds.\n");
    } else {
        printf("Parent still running...\n");
    }

    return 0;
}
```

Output:



```
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier …
arebaariq@Areeba:~/OSLABS/lab3$ ./task4
total 80
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:08 task1
-rw-r--r-- 1 arebaariq arebaariq   153 Oct 26 19:08 task1.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:09 task3
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:09 task3.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq 16048 Oct  5 19:12 task33
-rw-r--r-- 1 arebaariq arebaariq   301 Oct  5 19:19 task33.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16088 Oct 26 19:10 task4
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:10 task4.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4:Zone.Identifier
Parent still running...
arebaariq@Areeba:~/OSLABS/lab3$ gcc task33.c -o task33
arebaariq@Areeba:~/OSLABS/lab3$ ./task33
Parent still running...
total 80
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:08 task1
-rw-r--r-- 1 arebaariq arebaariq   153 Oct 26 19:08 task1.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task1:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:09 task3
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:09 task3.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3.c:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16048 Oct 26 19:10 task33
-rw-r--r-- 1 arebaariq arebaariq   301 Oct 26 19:10 task33.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task33:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task3:Zone.Identifier
-rwxr-xr-x 1 arebaariq arebaariq 16088 Oct 26 19:10 task4
-rw-r--r-- 1 arebaariq arebaariq   413 Oct 26 19:10 task4.c
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4.c:Zone.Identifier
-rw-r--r-- 1 arebaariq arebaariq     0 Oct 10 14:52 task4:Zone.Identifier
arebaariq@Areeba:~/OSLABS/lab3$
```

Remarks:

Same behavior as Task 2.

Child executes **ls -l**, and the parent prints a message without waiting.