NATIONAL TEXTILE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Eman Faisal 23-NTU-CS-1149

SECTION SE: 5th (A)

Operating System-LAB3

SUBMITTED TO:

Sir Nasir Mahmood

SUBMISSION DATE: 10/3/25

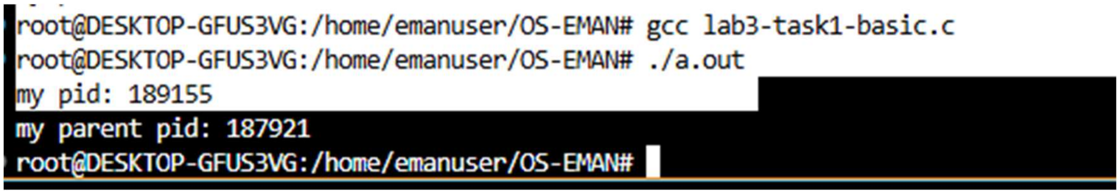
TASK-1

CODE:

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

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

OUTPUT:



```
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# gcc lab3-task1-basic.c
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# ./a.out
my pid: 189155
my parent pid: 187921
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN#
```

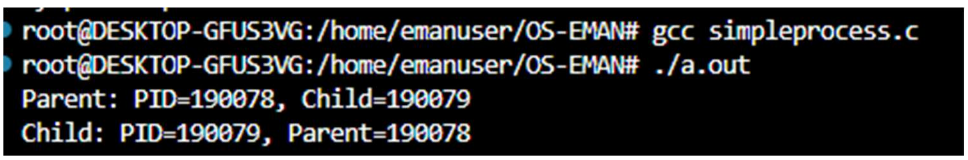
TASK-2

CODE:

```
#include <stdio.h>

#include <unistd.h>
int main() {
pid_t pid = fork();
if (pid == 0) {
// This block runs in the child process
printf("Child: PID=%d, Parent=%d\n", getpid(), getppid());
}
else {
// This block runs in the parent process
printf("Parent: PID=%d, Child=%d\n", getpid(), pid);
}
return 0;
}
```

OUTPUT:



```
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# gcc simpleprocess.c
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# ./a.out
Parent: PID=190078, Child=190079
Child: PID=190079, Parent=190078
```

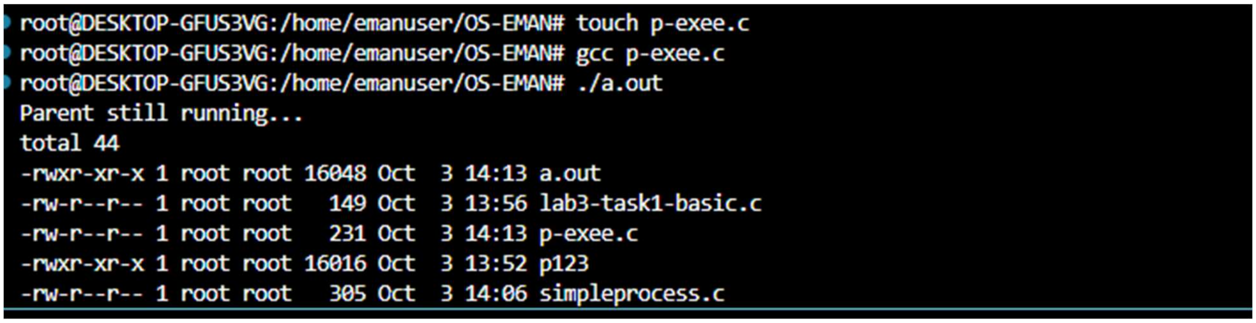
TASK-3

CODE:

```
#include <stdio.h>

#include <unistd.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:



```
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# touch p-exee.c
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# gcc p-exee.c
root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# ./a.out
Parent still running...
total 44
-rwxr-xr-x 1 root root 16048 Oct  3 14:13 a.out
-rw-r--r-- 1 root root   149 Oct  3 13:56 lab3-task1-basic.c
-rw-r--r-- 1 root root   231 Oct  3 14:13 p-exee.c
-rwxr-xr-x 1 root root 16016 Oct  3 13:52 p123
-rw-r--r-- 1 root root   305 Oct  3 14:06 simpleprocess.c
```

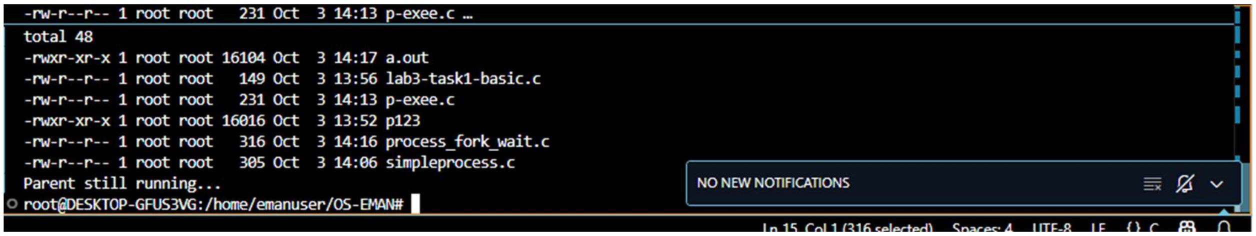
Difference between wait and no wait:

TASK-4

CODE:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.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 {
waitpid(pid, NULL, 0); // Wait for the child process to finish
printf("Parent still running...\n");
}
return 0;
}
```

OUTPUT:



```
 -rw-r--r-- 1 root root   231 Oct  3 14:13 p-exee.c …
 total 48
 -rwxr-xr-x 1 root root 16104 Oct  3 14:17 a.out
 -rw-r--r-- 1 root root   149 Oct  3 13:56 lab3-task1-basic.c
 -rw-r--r-- 1 root root   231 Oct  3 14:13 p-exee.c
 -rwxr-xr-x 1 root root 16016 Oct  3 13:52 p123
 -rw-r--r-- 1 root root   316 Oct  3 14:16 process_fork_wait.c
 -rw-r--r-- 1 root root   305 Oct  3 14:06 simpleprocess.c
 Parent still running...
○ root@DESKTOP-GFUS3VG:/home/emanuser/OS-EMAN# 
```