PROCESSES

OPERATING SYSTEMS LABS

LAB TASK # 09

Submitted By

SAWERA YOUSAF

(19P-0007)

Submitted to

MR. MUHAMMAD ABDULLAH

(COMPUTER INSTRUCTOR)

DEPARTMENT OF COMPUTER SCIENCE

FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCINCES, PESHAWAR

Session 2019-2023

Question 1:

CODE:

```
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/wait.h>
int main()
{
int pid;
pid = fork();
if(pid == 0)
{
printf("Child PID = %d\n", pid);
printf("I am the Child with PID = %d\n", getpid());
kill(getppid(),SIGTERM);
}
else
{
wait(NULL);
printf("Parent of = %d\n", pid);
printf("I am the Parent with PPID = %d\n",getppid());
}
return 0;
}
```

OUTPUT:

```
(base) transient@arewas-thinkpad-x230:~/Desktop$ gcc lab_9.c
(base) transient@arewas-thinkpad-x230:~/Desktop$ ./a.out
Child PID = 0
I am the Child with PID = 29591
Terminated
(base) transient@arewas-thinkpad-x230:~/Desktop$
```

Explanation:

In this code **signals.h** header file is used to send software termination signal via kill to parent. **kill(getppid(),SIGTERM)** this line of code sends termination signal to parent. If new process has pid 0 so the **if block** for child will execute and it will send termination signal to parent.

We have used **sys/types.h** library to include **kill()** system call.In this case if the **else block** is executed first so parent will wait until child gets done with its function. In this case the child is terminated so the program ends and control is not given back to parent. **sys/wait.h** is used to include **wait()** system call. **wait(NULL)** waits for the child to finish its work.