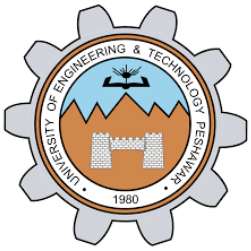
Operating Systems Lab-6

Process Creation, Execution and Termination



Submitted By: Awais Saddiqui

Registration# 21pwcse1993

Section: "A"

Submitted to:

Mam Madiha Sher

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar.

CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	Exceeds expectation (2.5)	Meets expectation (1.5)	Does not meet expectation (0)	Sc
1. Correctness	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 81-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-80% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	
2. Delivery	Delivered on time, and in correct format (disk, email, hard copy etc.)	Not delivered on time, or slightly incorrect format.	Not delivered on time or not in correct format.	
3. Coding Standards	Proper indentation, whitespace, line length, wrapping, comments and references.	Missing some of whitespace, line length, wrapping, comments or references.	Poor use of whitespace, line length, wrapping, comments and references.	
4. Presentation of document	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting and excellently organized.	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting.	No name, date, or assignment title included. No task titles, no objectives, no output screenshots, poor formatting.	

Instructor:

Name: __Engr. Madiha Sher___ Signature: _____

Operating Systems Lab

What is a Process Creation :

Process creation is the act of creating a new process. A process is a unit of execution in an operating system. It is a program in execution with its own address space, stack, registers, and resources.

What is a Process execution:

Process execution is the act of carrying out a process. It is the process of taking a set of instructions and carrying them out in order to achieve a specific outcome. Process execution can be manual or automated.

What is a Process Termination:

Process termination is the end of a process's life cycle. It can happen for a variety of reasons, such as the process reaching the end of its code, receiving a signal, or being killed by the operating system.

Orphan Process:

If a parent dies before its child, the child (orphan process) is automatically adopted by the original “init” process whose PID is 1.

Objectives:

This lab describes how a program can create, terminate, and control child processes. Actually, there are a few distinct operations involved: creating a new child process, and coordinating the completion of the child process with the original program.

Task #1:

Write a C program that executes ls -l command in the child process. Parent process shall wait for the child process.

Code:

awais@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int main(){
        int pid;
        pid=fork();
       // printf("Process Id is %d and parrent ID is %d \n",getpid(),getppid());
        if(pid==0)
                execlp("ls","ls","-l",NULL);
        else
                wait(NULL);

        return 0;
}
```

Output:

awais@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6

```
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ gcc Task.c -o Task.o
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ ./Task.o
total 64
-rwxrwxrwx 1 awais awais 26112 May 17 21:28 'Operating Systems lab06.doc'
-rwxrwxrwx 1 awais awais   173 May 18 00:04  Task.c
-rwxrwxrwx 1 awais awais 16040 May 18 00:04  Task.o
-rwxrwxrwx 1 awais awais 19625 May 17 23:35  Task1.png
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$
```

Task #2:

a. Write a C program that finds the max of an array.
b. Write a C program that creates a child process and executes the above program in child process. Parent shall wait for the child process.

Code:

awais@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int  maximum(int arr[],int size){
        int temp=arr[0];
        for(int i=0; i<size; i++){
                if(arr[i]>temp)
                        temp = arr[i];
        }
                        return temp;
}
int main(){
        int array[5]={4,12,55,3,40};
        int pid;
        int result=maximum(array, 5);
        pid = fork();
        if(pid==0)
                printf("Maximum is %d \n",result);
        else
                wait(NULL);
```

Output:

```
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ vim Task2.c
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ ./Task2.o
Maximum is 55
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$
```

Task #3:

Create a fan of N processes. Take N as input from the user. Make sure there are no orphan processes.

Code:

awais@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6

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

int main(){
        int input, pid;
                printf("How many processes you want to create ...");
                scanf("%d",&input);
                for(int i=0; i<input;i++){
                        pid=fork();
                        if(pid==0){
                        printf("I'm child and my ID is %d and my PPID is %d \n",getpid(),getppid());
                        break;
                        }
                        for(int j=0; j<input;j++){
                         wait(NULL);
                                printf("Child is Terminated .... \n");
                        }
                }
        return 0;
}
-- REPLACE --                                                    1,1           All
```

Output:

```
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ vim Task3.c
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ ./Task3.o
How many processes you want to create ...3
I'm child and my ID is 173 and my PPID is 172
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
I'm child and my ID is 174 and my PPID is 172
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
I'm child and my ID is 175 and my PPID is 172
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$
```

Task #4:

Create a chain of N processes. Take N as input from user. Make sure there are no orphan processes.

Code:

awais@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6

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

int main(){
        int input, pid;
                printf("How many processes you want to create ...");
                scanf("%d",&input);
                for(int i=0; i<input;i++){
                        pid=fork();
                        if(pid==0){
                        printf("I'm child and my ID is %d and my PPID is %d \n",getpid(),getppid());
                        sleep(30);
                        }
                        for(int j=0; j<input;j++){
                         wait(NULL);
                                printf("Child is Terminated .... \n");
                        }
                }
        return 0;
}
-- REPLACE --                                                    1,1           All
```

Output:

```
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ gcc Task4.c -o Task4.o
awais@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-6$ ./Task4.o
How many processes you want to create ...3
I'm child and my ID is 156 and my PPID is 155
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
I'm child and my ID is 157 and my PPID is 156
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
I'm child and my ID is 158 and my PPID is 157
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
Child is Terminated ....
```