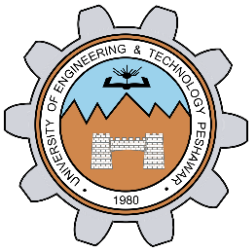
Process Creation, Execution and Termination

LAB # 06



CSE-204L Operating Systems Lab

Spring 2024

Submitted by: **Hassan Zaib Jadoon**

Registration No.: **22PWCSE2144**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Abdullah Hamid

April 25, 2024

Department of Computer Systems Engineering

UET Peshawar

Process Creation, Execution and Termination

Task 01

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

Code:

```
 GNU nano 7.2                                                  code1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork(); // Forking a child process

    if (pid == -1) {
        // Fork failed
        perror("fork");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        // Child process
        execl("/bin/ls", "ls", "-l", (char *)NULL); // Execute ls -l
        // The following code will execute only if execl fails
        perror("execl");
        exit(EXIT_FAILURE);
    } else {
        // Parent process
        int status;
        waitpid(pid, &status, 0); // Wait for the child to complete
        if (WIFEXITED(status)) {
            printf("Child process exited with status: %d\n", WEXITSTATUS(status));
        } else {
            printf("Child process terminated abnormally\n");
        }
    }
    return 0;
}
```

Output

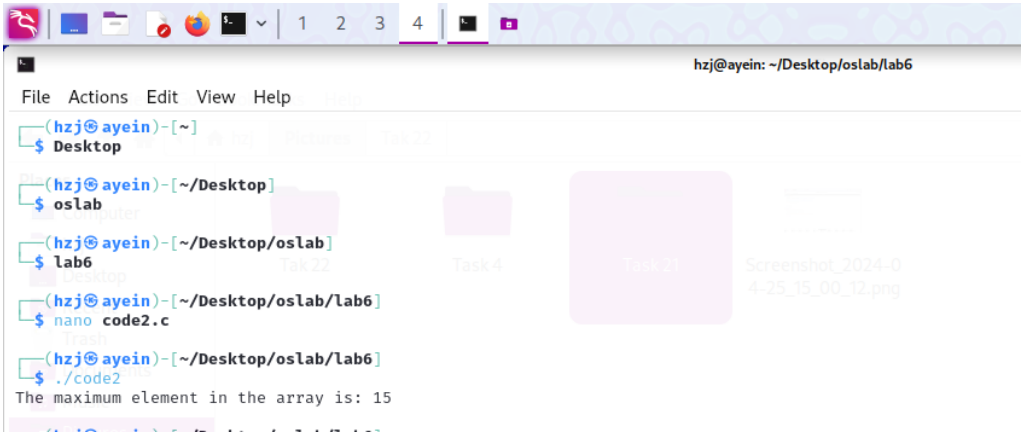
```
┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ nano code1.c

┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ ./code1
total 100
-rwxr-xr-x 1 hzj hzj 16256 Apr 25 14:27 code1
-rw-r--r-- 1 hzj hzj   835 Apr 25 14:27 code1.c
-rwxr-xr-x 1 hzj hzj 15984 Apr 25 14:29 code2
-rw-r--r-- 1 hzj hzj   510 Apr 25 14:29 code2.c
-rwxr-xr-x 1 hzj hzj 16256 Apr 25 14:31 code3
-rw-r--r-- 1 hzj hzj   852 Apr 25 14:31 code3.c
-rwxr-xr-x 1 hzj hzj 16312 Apr 25 14:34 code33
-rw-r--r-- 1 hzj hzj  1093 Apr 25 14:34 code33.c
-rwxr-xr-x 1 hzj hzj 16312 Apr 25 14:36 code4
-rw-r--r-- 1 hzj hzj   868 Apr 25 14:36 code4.c
Child process exited with status: 0

┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$
```

Task 02

a. Write a C program that finds the max of an array.

Code:

Output:



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.

```
hzj@ayein: ~/Desktop/oslab/lab6
File  Actions  Edit  View  Help
 GNU nano 7.2                                  code3.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork(); // Forking a child process

    if (pid == -1) {
        // Fork failed
        perror("fork");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        // Child process
        execl("./find_max", "find_max", (char *)NULL); // Execute the find_max program
        // The following code will execute only if execl fails
        perror("execl");
        exit(EXIT_FAILURE);
    } else {
        // Parent process
        int status;
        waitpid(pid, &status, 0); // Wait for the child to complete
        if (WIFEXITED(status)) {
            printf("Child process exited with status: %d\n", WEXITSTATUS(status));
        } else {
            printf("Child process terminated abnormally\n");
        }
    }
    return 0;
}
```

Output

```
┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ nano code3.c

┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ ./code3
execl: No such file or directory
Child process exited with status: 1
```

Task 03

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

```
 GNU nano 7.2                                  code33.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    int N;

    printf("Enter the number of processes (N): ");
    scanf("%d", &N);

    for (int i = 0; i < N; i++) {
        pid_t pid = fork();

        if (pid == -1) {
            // Fork failed
            perror("fork");
            exit(EXIT_FAILURE);
        } else if (pid == 0) {
            // Child process
            printf("Child process %d created with PID: %d\n", i + 1, getpid());
            // Child process terminates immediately to prevent it from becoming an orphan
            exit(EXIT_SUCCESS);
        }
    }

    // Parent process waits for all child processes to finish
    for (int i = 0; i < N; i++) {
        int status;
        wait(&status);
        if (WIFEXITED(status)) {
            printf("Child process %d with PID %d exited normally\n", i + 1, WEXITSTATUS(status));
        } else {
            printf("Child process %d with PID %d exited abnormally\n", i + 1, WTERMSIG(status));
        }
    }
```

Output

```
┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ ./code33
Enter the number of processes (N): 2
Child process 1 created with PID: 6838
Child process 2 created with PID: 6839
Child process 1 with PID 0 exited normally
Child process 2 with PID 0 exited normally
Parent process finished
```

Task 04

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

```
 GNU nano 7.2                                    code4.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    int N;

    printf("Enter the number of processes (N): ");
    scanf("%d", &N);

    pid_t pid = getpid(); // Get the PID of the parent process

    for (int i = 0; i < N; i++) {
        pid_t child_pid = fork();

        if (child_pid == -1) {
            // Fork failed
            perror("fork");
            exit(EXIT_FAILURE);
        } else if (child_pid == 0) {
            // Child process
            printf("Child process %d with PID: %d, Parent PID: %d\n", i + 1, getpid(), getppid());
        } else {
            // Parent process
            pid = child_pid; // Update parent PID to the new child's PID
        }
    }

    // Parent process waits for the last child process to finish
    if (getpid() != pid) {
        wait(NULL);
    }

    return 0;
}
```

Output

```
File  Actions  Edit  View  Help
┌──(hzj㉿ayein)-[~]
└─$ Desktop

┌──(hzj㉿ayein)-[~/Desktop]
└─$ oslab

┌──(hzj㉿ayein)-[~/Desktop/oslab]
└─$ lab6

┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$ ./code4
Enter the number of processes (N): 4
Child process 1 with PID: 5034, Parent PID: 5017
Child process 2 with PID: 5035, Parent PID: 5017
Child process 2 with PID: 5037, Parent PID: 5034
Child process 3 with PID: 5036, Parent PID: 5017
Child process 3 with PID: 5039, Parent PID: 5035
Child process 3 with PID: 5040, Parent PID: 5034
Child process 4 with PID: 5038, Parent PID: 5017
Child process 4 with PID: 5041, Parent PID: 5035
Child process 4 with PID: 5042, Parent PID: 5034
Child process 4 with PID: 5044, Parent PID: 5036
Child process 4 with PID: 5047, Parent PID: 5040
Child process 3 with PID: 5043, Parent PID: 5037
Child process 4 with PID: 5046, Parent PID: 5039

Child process 4 with PID: 5045, Parent PID: 5037
Child process 4 with PID: 5048, Parent PID: 5043
┌──(hzj㉿ayein)-[~/Desktop/oslab/lab6]
└─$
```

CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	Exceeds expectation (2.5)	Meets expectation (1.5)	Does not meet expectation (0)	Score
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. Abdullah Hamid Signature: _____