CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 1
Fall 2019
Due Date: October 7, 2019

Purpose

The goals of this assignment are the following:

- Gain more experience with the C programming language from an OS perspective
- Get experience with the *fork(), wait(), execl(), execlp(), and pipe()*system functions

Part I: Parent and Child Processes (50 points)

Write a program in C that will perform the following tasks:

1. Your program will create a *parent* process which will create two *child* processes (e.g., child_1, and child_2)
2. *parent* will wait for child_1 to complete before creating child_2
3. child_1 will create its own child child_1.1
4. Inside child_2, a system call to an external program will be made (let's say this external program is "external_program.out" that prints "Hello World"), and as a result of this external program call, child_2 will be replaced by external_program.out (hint: execl())

The expected output from your program should look like the following:

parent process (PID 2255) created child_1 (PID 2256)
parent (PID 2255) is waiting for child_1 (PID 2256) to complete before creating child_2
child_1 (PID 2256) created child_1.1 (PID 2257)
child_1 (PID 2256) is now completed
parent (PID 2255) created child_2 (PID 2258)
child_2 (PID 2258) is calling an external program external_program.out and leaving child_2…
From the external program: Hello World..

Your submission file **must** be assignment1_part1.c.

Hints: fork(), wait(), getpid(), getppid(), execl()

Part II: Inter-Processes Communications (50 points)

Write a C program that will accept two integer values from the user as **command line arguments** (for example, X and Y). Your program will create a *parent* and *child* where the *parent* process will read X and the *child* process will read Y. Now, *child* process will send Y to *parent* process by communicating parent process through a *pipe* (i.e., shared memory). Then the parent process will add Y to X and produce the sum. The expected output from your program should look like the following:

1. A pipe is created for communication between *parent* (PID 2255) and *child*
2. *parent* (PID 2255) created a c*hild* (PID 2256)
3. *parent* (PID 2255) Reading *X* = 10 from the user
4. *child* (PID 2256) Reading *Y* = 20 from the user
5. *child* (PID 2256) Writing *Y* into the *pipe*
6. *parent* (PID 2255) Reading *Y* from the *pipe* (*Y* = 20)
7. *parent* (PID 2256) adding *X* + *Y* = 30

Your submission file **must** be assignment1_part2.c.

Hints: fork(), wait(), getpid(), getppid(), pipe(), write(), read()

Mark Distribution

This section describes a tentative allocation of marks assigned for the desired features.

- **Part I: Parent and Child Processes (50 points)**
    a) A Parent process will create two Child processes: 10 points
    b) Child_1 will create its own child Child_1.1: 10 points
    c) Parent will wait for Child_1 to complete before creating Child_2: 10 points
    d) Child_2 will make a system call to external_program.out: 20 points

- **Part II: Inter-Processes Communications (50 points)**
    a) Parent reads X from user: 5 points
    b) Child reads Y from user: 5 points
    c) A pipe is created for communication between Parent and Child: 15 points
    d) Child writes Y into the pipe: 10 points
    e) Parent reads Y from the pipe: 10 points
    f) Parent adding two variables X+Y: 5 points

Assignment computing platform

Please visit the course website for assignment related specific technical instructions (Assignment submission, Assignment computing platform, relevant materials etc.). Also, consult TAs, and Instructor for any question you may have regarding this assignment.