Programming Assignment 1 [30 pts + bonus 5pts]

COMP4270/6270 Operating Systems – Spring'20

Due 3/16/2020

Instructor: Myounggyu Won

1. [Preparation – 5 pts] Install a POSIX-compliant operating system such as Ubuntu, Linux, MacOS. You can use a free virtualization software such as VMWare Player or VirtualBox to install a POSIX-compliant operating system if needed. Note that your programs will be graded on a POSIX-compliant operating system.

2. [Process Creation – 10 pts] Write a C program that creates one child process. The parent process should display its process ID continuously in an infinite loop. The child process should display its process ID and its parent process ID continuously in an infinite loop. Observe how these two processes share the CPU time and briefly explain your observation in your report (one or two sentences should be enough).

Use the system calls pid() and ppid() to get the process id and the parent process id. Here is an example.

```
#include <stdio.h>

int main( void )

{

        int pid;

        printf( "ORIGINAL: PID=%d PPID=%d\n", getpid(), getppid() );

        return( 1 );

}
```

Your program will be terminated using Ctrl+C (SIGINT). The output of this program may look like this:

[Parent process] PID=1234

[Child process] PID=1235, Parent-PID=1234

[Parent process] PID=1234

[Parent process] PID=1234

[Child process] PID=1235, Parent-PID=1234

…

3. [Process Creation – 10 pts] Write a C program that creates a child process. In this program, the child process takes the user input, a single word that represents a program name (e.g., ls, pwd, …). The child process then executes the program. After executing the program, the child process continues to take another user input. In particular, if the user types 'exit', the child process should be terminated. The parent process should be waiting for the child process to terminate. Once the child process is terminated, the parent process is terminated.

In solving this problem, you will have to use the exec() system call. Review the course slides and the manual about the exec system call to learn about how to replace the memory space of a child process with a new program: https://linux.die.net/man/3/exec

The output of this program may look like this:

home/mwon/> ls

. .. a.txt b.txt

home/mwon> pwd

/home/mwon

home/mwon> exit

bye!

4. [Zombie Process – 10 pts] Create a C program that forks a child process that ultimately becomes a zombie process. Make the zombie process be maintained in the system for 1 minute. Provide a screenshot of the results of the command 'ps -l'. Indicate that there is a zombie process with the state 'Z'. Please include a snapshot of the results of the command 'ps -l' in your report.

Files to submit:

- Firstname_p2.c
- Firstname_p3.c
- Firstname_p4.c
- 1-page report