CS211 Programming and Operating Systems

Lab 5: Processes

29 March 2021

This lab is a set of exercises, based on ones in Chapter 5 of the textbook; see http://pages.cs.wisc.edu/~remzi/OSTEP/.
Since the use the `fork()` system call, and other UNIX-related system calls, you'll need to complete these exercises using in suitable online compiler; `code::blocks`, with the *mingw* compiler, is not sufficient.
(If you have a Windows computer, and are confident in installing non-standard software, you use `code::blocks` with the `cygwin` C compiler, but it takes a little effort).

The purpose of this lab is to help you get familiar with the concepts of `fork` and signals, and help you prepare for the final lab (after Easter). It is a "low stakes" assignment , contributing about 4% to your over-all grade. To get that 4% submit your solutions to **Exercises 8 and 9** by 5pm, Friday, 9 April.

1. Using an online or desktop compiler, verify that you can run sample programmes from Week 7, in particular, `02WhoAmI.c` and `08Pipes.c`.
2. Write a program that calls `fork()`. Before calling `fork()`, have the main process declare an initialise an `int` variable $x = 100$. What value is the variable in the child process? What happens to the variable when both the child and parent change the value of x?
3. If a process opens a file, does a child process have access to it? What happens if they both try to write to the file at the same time? To answer this, download `02fopen.c` from `http://www.maths.nuigalway.ie/~niall/CS211/lab5`. Notice the use of the `fflush()` system call; how does the output change if that is removed?
4. Write another program using fork(). The child process should print "hello"; the parent process should print "goodbye". You should try to ensure that the child process always prints first; can you do this without calling `wait()` in the parent?

5. Before answering the next question, read, compile and run `04WaitAndCount.c`. See `http://www.maths.nuigalway.ie/~niall/CS211/Week07/`
6. (From OSTEP) Write a program that uses `wait()` to wait for the child process to finish in the parent. What does `wait()` return? What happens if you use `wait()` in the child?
7. Find out what the `waitpid()` function does. Write a programme that exhibits that.
8. [Homework Exercise] Write a C program that works as follows:
   - The parent process `fork`s a child;
   - The child process outputs `1, 2, ..., 9, 10`, but `sleep`s for 1 second between each number.
   - The parent
     - sleeps for 3 seconds, then outputs a message, sends `SIGSTOP` to the child.
     - sleeps for 3 seconds, then outputs another message, sends `SIGCONT`
     - sleeps for 4 seconds, then outputs a final message,
9. [Homework Exercise] Write a C program like `07SIGUSR1.c` from Week 7, but that sets up signal handlers for both `SIGUSR1` and `SIGUSR2`. The child process should prompt the user to input `1` (for SIGUSR1) or 2 (for SIGUSR2), and then send that signal to the parent.