Lab 5 - Description

(Signal Processing in C)

Lab Overview:

For this lab, we will be learning how to process signals in C. A signal is a special integer that the operating system uses to communicate directives to its processes. Consider an example we are familiar with: `SIGTERM.` This is the signal that is sent to a process to terminate it. You can see for yourself if you start a program then enter CRTL-C. This lab will focus on how to send signals to our child process to accomplish specific tasks.

You will be building on top of your lab4 or Part1 of your project.

Core Tasks:

1. Make each process wait before the exec call.
2. Create a signaler function that sends the given signal to every child process.

Task Details:

1. Spawn a minimum of 5 children processes. Each child process will wait until it receives the SIGUSR1 signal.
    a. Use **sigwait(3):** http://man7.org/linux/man-pages/man3/sigwait.3.html
    b. Before waiting the child must print the following:
        i. **Child Process: <pid> - Waiting for SIGUSR1…**
    c. After the child receives the signal print the following:
        i. **Child Process: <pid> - Received signal: SIGUSR1 - Calling exec().**
2. Create a signaler function that is called from the parent process.
    a. The signaler function will take the process pool, size of the pool and a signal as parameters.
    b. Your signaller **must** then loop through each child and do the following:
        i. Sleep for 2 second. See the sleep function mentioned in lab 4.
        ii. Print: **Parent process: <pid> - Sending signal: <signal> to child process: <pid>**

        iii. Send the signal to each child. Using kill(2): http://man7.org/linux/man-pages/man2/kill.2.html

3. Send signals to the children.
    a. First, we will send the `SIGSTOP` signal to all children.
    b. Wait for 5 seconds, send the `SIGCONT` signal to all children.
    c. After this, wait for another 3 seconds and we will send the `SIGINT` signal to all children to terminate them.

Remarks:

If you are finished with part 2 of the project you can add the above-mentioned print statements and show us that as well.

Submission Requirements:

1. Files include
    a. main.c
    b. makefile
    c. a readme.txt briefly explain how your program is run
        i. specify if you are building on top of lab4 or part1
2. Screenshot of Compiling running, and system information
3. Submit your lab 5 folder as a tar.gz to Canvas.
    a. The submission must take place before midnight Friday, Nov 6th.