PLEASE SUBMIT THE ASSIGNMENT IN THE LAB SECTION ON CAMINO
Programming Assignment #1: Creating Processes

Due Date: Demo **Schedule a time with your TA before your next lab (April 19th and 20th)**,

submit code and documentation **before the end of that day**

The Basic Programming Task (up to 20 points)

The goal of this assignment is to make sure you fully comprehend (and are able to realize) the creation of processes with Unix-like operating systems. To this end you will be using the fork() system call.

You are to write a very simple shell program, similar to the example covered in lecture. The shell should terminate if the user enters "exit" at the command line, and should attempt to launch a program in response to any other input string. You do not have to parse any parameters, nor do you need to support background execution of processes. The simplest possible shell to demonstrate is all that is required. You may wish to write a "hello world" program, so that you have an executable file that you could launch from your shell.

The Second Programming Task (up to 10 additional points):

You are to write a program that will create exactly **eight** child processes (not including the initial program itself). You are **not** to allow any single process to create more than two child processes.

The Third Programming Task (up to an additional 20 points)

You are to write a program that, when launched, will result in a total of **twelve** child processes to run. This is the same task as described above, but this time you are **not** to allow any single process which creates children to create **less** than two child processes, or **more** than three. In other words, the program you write can only launch a maximum of three child processes directly, and any other "children" will have to be created by the children of this parent process (subject to the restriction of only creating two or three processes each). Once again, any process that creates other processes must create either two or three processes only, no more and no less.

The Fourth Programming Task (up to an additional 20 points)

You are to write a program that, when launched, will result in a total of **seventeen** child processes to run. This is the same task as described above, again this time you are **not** to allow any single process which creates children to create **less** than two child processes, or

more than three. In other words, the program you write can only launch a maximum of three child processes directly, and any other "children" will have to be created by the children of this parent process (subject to the restriction of only creating two or three processes each). Once again, any process that creates other processes must create either two or three processes only, no more and no less.

The Demo (up to an additional 30 points)

You are to demo your final solutions to the TA. Be prepared to answer questions about your code, as this will account for the final 30 points of this assignment.

The Details

You must submit the following:

- The source code for your solution.
- Documentation of your solution, which should include how it works, and how you tested your solution. Where necessary, your documentation should assume your code will be tested on the linux workstations in the design center.

Failure to provide complete documentation (either within the source file, as a detailed header comment, or as an accompanying README file), will result in a penalty, as would providing code that does not solve the problem, or is not readily testable on the school linux systems. It is your responsibility to make sure that you submit the code and documentation before the deadline (next week), and that you are prepared to demo your code for the TA in lab.