CECS 326 Assignment 3 (10 points)

Due: October 15, 2019 by class time on BeachBoard

Cooperating processes need to communicate between them. One way Linux supports interprocess communication is message queue. A message queue must first be acquired from the operating system by calling **msgget**. Control operations, e.g., remove, can be performed on an existing message queue by calling **msgctl**. Processes with appropriate permissions may send and/or receive messages via the message queue by calling **msgsnd** and **msgrcv**. Please consult the man pages of these system calls for details.

For this assignment you need to write three C++ programs named *master.cpp*, *sender.cpp*, and *receiver.cpp*, which should be compiled into executables *master*, *sender*, and *receiver*, respectively. Together they should do the following:

The *master* process should first acquire a message queue from the operating system, followed by creating two child processes, with one to execute *sender* and the other to execute *receiver*, using **fork** and **exec** system calls. The *master* process should output the message queue ID, the process IDs of the *sender* and *receiver* processes it has created, then waits for both child processes to return. Then *master* will remove the message queue before exit.

When execution begins, *sender* and *receiver* should output a message as follows:

Sender (or Receiver), PID xxxxx, begins execution

The *sender* process should prompt user for a line of input, then send the input line to *receiver* via the message queue created by *master*. Make sure that the prompt identifies the process making the prompt is *sender* along with its PID.

The *receiver* process retrieves a message from the message queue, and outputs the message on the screen. Make sure the output identifies that the line is printed by *receiver* along with PID.

The program must run successfully on a Linux machine in the CECS Lab.

Do the following for this assignment:

1. Develop three C++ programs (*master.cpp*, *sender.cpp*, and *receiver.cpp*) as described above. Make sure your programs are adequately commented.
2. Submit on BeachBoard the three C++ programs, with a cover page that provides: your name, your student ID, course # and section, assignment #, due date, submission date, and a clear program description. Format of the cover page should follow the cover page template on BeachBoard. All three programs must be properly formatted and adequately commented to enhance readability and understanding.
3. Demonstrate your program on a Linux machine in the Lab and explain details of your implementation. Bring a hardcopy of the cover page for the demo.