CECS 326: Project 4

Objectives: Interprocess communication with semaphores and shared memory.

Implement a system to guarantee the child processes are resumed in accordance to their order in a FIFO queue. That is, a FIFO queue is used to determine the sequence that child processes will be resumed: The child at the front of the queue is next to be resumed.

For simplicity, a child will repeatedly generate a random integer and test if it's a factor of some integer X (see below), until it generates a random integer less than 100 or is divisible by X, then it yields its turn to the next child process. That is, it enqueues itself for its next turn, resumes the next child in the queue and halts.

To make the programming more interesting: Four child processes will be spawned to compute whether some integer is a factor of: U=827395609 or V=962094883. One of the child processes works concurrently on integer U, as another child works on integer V, while the other 2 child processes wait for their turn. No two child processes can operate on the same integer. The parent process (oldest) prompts for exit command: "!wq". All processes will terminate on this input string. (It’s not important what each software outputs on the console.)

When a child is resumed & continues its execution, it needs to determine which of the 2 integers U or V to work on. Again, no two child processes can operate on the same integer. There are different ways to do this.

You have permission to use SEMAPHORE class for this assignment.

Demo your software and submit hard copy of your source code for grading. Along with a software description, include a small paragraph to explain whether or not your solution exhibits starvation or deadlock.