

# WENTWORTH INSTITUTE OF TECHNOLOGY

College of Engineering and Technology  
Department of Electrical Engineering and Technology

Operating Systems  
Spring 2018

## Lab 3

### PART 1:

Write a function that creates a Fibonacci sequence ([https://en.wikipedia.org/wiki/Fibonacci\\_number](https://en.wikipedia.org/wiki/Fibonacci_number)) of length  $n$ , where  $n$  is a integer parameter passed to the function. Initialize the first two elements to a value of 1.

Then create the main routine of your program, in which the `fork()` system call is used to create a child process that implements the Fibonacci sequence. The length of the sequence  $n$  is obtained from the user as a parameter that is passed to your program when it started (i.e. when you invoke your program from the shell, you pass it one parameter,  $n$ ). The child process should use shared memory (of size  $n * \text{sizeof}(\text{int})$  and created by the parent) to populate values of the sequence and pass it to the parent process. Since we have not studied how to synchronize between processes or threads, the parent process should just wait for the child process to exit and then read the values from the shared memory. Finally (i.e. after the child has terminated), the parent process shall output the sequence to the screen.

### PART 2: (home assignment)

The above program represents a producer-consumer problem similar to the one discussed in class. **Without** using any OS supported synchronization primitives (i.e. mutexes, semaphores, monitors, etc.), modify your code such that the parent process does not wait for the child process to terminate, but instead starts reading and printing values that are ready, from the shared buffer.

### What to hand in (using Blackboard):

- Your “.c” file(s) (with appropriate comments). Do not attach project or make files.
- A screen shot of your terminal window(s).

### RULES:

- Each group may consult with other groups/students about GENERAL concepts or methods, but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied from other students, the internet or any other source).
- Each member of a group is required to contribute, and will be required to explain and defend every part of work done.
- Only one lab report should be submitted for each group.
- **To get full credit, you must attend the lab, show me your progress before you exit the lab (this goes for every student in the group), and submit required files before the posted deadline.**

