Matthew London and Ismaeel Varis

Dr. Ali Elkateeb

CIS 450

15 August 2018

Course Project report

**Problems Ran into**

1.) When looping in the program there were memory leaks displaying incorrect data and an infinite loop printing the same string of numbers. The solution we found was by checking our logic for the looping and making sure that random numbers were being generated and removed.

2.) We had an error where the thread buffer was consistently full because remove function was not removing the correct numbers from the thread. The solution we had was making sure that the correct numbers were being removed and that if a buffer was close to being full it would go to different threads.

3.) We had minor syntax errors when displaying the data to the console window but were able to find a fix by looking up syntax on stackoverflow and other sites.

4.) There was a mysterious error that would occur every 3 executions or so, it was a memory access error being triggered by the semaphores. The problem appears to be solved by launching a pthread with the address of the producer/consumer function and not the function itself.

(for example: pthread\_create(&Producer, NULL, &Produce, (void\*)(i + 1)); vs. pthread\_create(&Producer, NULL, Produce, (void\*)(i + 1)); )

**Pthread system calls used in program**

To run the program we used **Visual studio 2013** with the pthread packages listed in step 1 of the steps to run the program section of this document. For our Consumer and Producer functions we used semaphores to check when the thread was full or empty. We also used mutex locks/unlocks using the pthread\_mutex\_lock commands. We used 2 functions remove and insert to add/remove data to our threads and called them in the consumer/Producer functions. To create the threads we used Pthread\_create on both of the consumer/producer functions.

**Summary**

We used semaphores to check the if the threads were empty or full and used mutex locks to make sure that mutual exclusion was kept. Along with these we used the pthread package library as stated above to populate and regulate data in the threads

**Steps to run program**

1.) Install Pthread https://www.youtube.com/watch?v=4GdTcqE0iqg

2.) Connect <include> <lib> and linker dependencies to folder for pthread

3.) Rebuild program and Debug program

4.) Producer and Consumer will add to the threads endlessly. If empty or full is triggered the program will \*\*sleep\*\* according to document specification.

**Screen shots**

