Eshwar Muneshar

CS 340 Project 3

For Project 3, the assignment we were tasked with was to first implement a shared stack of size 10 that would serve to help communicate between 2 threads. The stack was to be implemented using an array and push/pop functions that are based off of homework previously assigned. Thread 1 of the aforementioned threads will push integers starting from 1 going up to 120 onto the stack while thread 2 would pop them off of the stack in a prescribed order.

To start off, the first step in accomplishing this task was to declare variables. The stack was declared as a global array so that way both threads would easily have access to it, as is the variable used to point to the top of the stack. After that I modified the push and pop methods so that they would use a mutex lock to access the array. The mutex lock is important in the sense that it would only allow one method, and by extension, only 1 thread to access the shared stack at a time. This is important to prevent what are known as race conditions, where both threads are trying to change something in the stack at the same time which could lead to data corruption or loss of data.

The output that I got after writing the program was not the output that we were supposed to get, but after speaking with the professor, he said that was ok and to include it in the essay portion. Given more time, getting the output in the correct order would not have been an issue. The output for the project is to be written out to a txt file. While working on the project, I ran into significant issues, but when it all came down to it, it wasn’t because I didn’t understand the assignment or because I didn’t know what to do, it was because I didn’t read the problem carefully enough. The answer to my issue was written in front of me. I spent 3 days trying to fix code that wasn’t broken because I forgot to include a sched_yield call when the stack was full. In hindsight that was a great reminder to always understand the job at hand before I dive head first into it.