Lab 8: Synchronization

Start Assignment

- Due Nov 17 by 11:59pm
- Points 10
- Submitting a file upload
- Available after Aug 26 at 12am

Threads Synchronization

Problem 1: Mutex Lock

Create a number of threads and have them share an integer variable, and have each thread update (e.g., increment) the value of the shared variable, and then in the main thread, prints out the final value of the shared variable after all child threads complete, use a mutex lock to avoid race condition.

Problem 2: Condition Variable

In Lab 6, you wrote a multithreaded program to generate the Fibonacci sequence. The program required the parent thread to wait for the child thread to completely finish its execution before printing out the computed Fibonacci sequence. If we let the parent thread access the Fibonacci numbers ***as soon as*** they were computed by the child thread—***rather than waiting for the child thread to terminate***—what changes would be necessary for the solution to this problem? Implement your modified solution. (Hint: consider using a condition variable, and then think about what value should we ***initialize*** it?)

Submission requirement:

- source code file(s)
- screenshot(s) of testing results, one for each problem