Conner Jordan

August 1st, 2023

CST 334

Lab 6 Write Up

**Explain what happens when you run the threadSync.c program?**

It creates 10 threads, and uses a semaphore to ensure that only one thread at a time can enter its critical section. Each thread simply prints a message indicating that it has entered the critical section, then sleeps for one second, and then leaves the critical section. The main thread waits for all the threads to finish, and then completes. When you run this program, it will print out messages in the order that the threads enter and leave the critical section, which may vary between runs due to the non-deterministic nature of thread scheduling.

**Part 2**

The producer-consumer problem is a synchronization problem where two processes share a common, fixed-size buffer as a queue. The producer's job is to generate data and put it into the buffer, while the consumer's job is to consume the data from the buffer. The challenge is to make sure that the producer doesn't try to add data into the buffer if it's full, and that the consumer doesn't try to remove data from an empty buffer.

The "Lab6" code I’ve implemented solves the producer-consumer problem using semaphores for synchronization.