pThreads Write-Up

Andy Lum

February 28, 2023

0.0.1 Analysis

- Mutex is a lock that is set before using a shared resource and release after using it. Once the lock is set, no other threads can access the locked region of code. The mutex can be unlocked and destroyed by calling : the first **pthread_mutex_unlock(&lock);** releases the lock and the second **pthread_mutex_destroy(&lock);** destroys the lock so that it cannot be used anywhere else. For the no synchronization file, there is a counts function that will take the y value, and then goes through a loop that adds to the count and it will print the count out.

0.0.2 Experiment

- four scenarios: 1. lock a locked mutex in same thread, 2. unlock a unlocked mutex, 3. (Wait to) Lock a locked mutex owned by another thread (normal use case), and 4. Unlock a locked mutex owned by another thread. Comparing Mutex to Busy-Wait, Mutex had a better performance. When the counter indicates that every thread has entered the critical section, threads can leave the critical section. In conclusion, a mutex can be used to avoid conflicting access to critical sections as well. Ideally, think of it as a lock on a critical section, since mutexes arrange for mutually exclusive access to a critical section.