Hello,
I'm just going to be talking about the synchronization algorith in this report as that's what is required. I started by writing the function that will process the critical section. But I didn't want to write two functions so I used the differences of 'X' and 'O' messages to differentiate between the threads. I used one big mutex lock around the whole critical section so my locking mechanism is coarse-grained.I used a global 'turn' variable to make the threads alternate by calling 'pthread_cond_wait' and 'pthread_cond_signal' with their respective turn value so that my locking algorithm was fair. I toggled the 'turn variable' whenever I was done, called 'pthread_cond_signal' to wake up the other thread and released the mutex. I didn't forget to put the same ending process when I was going to return from my function so that the other thread wouldn't sit idle in a queue when the latter thread's job was finished and destroyed.Thanks.