Christopher Schaefer 3/17/2021
CS 3053 Operating Systems Dr. Mauricio Papa
Project 1 Report

The bulk of the project was spent understanding the existing nachos code that included the partial thread system. I spent most of my time sifting through the existing classes for the project as well as online material for nachos and gaining an understanding from that (as well as existing knowledge from the course). Implementation of this project wasn't difficult or long in terms of time taken or lines of code.

For task 1, I set the thread up to wait and add itself to a ThreadQueue in join(). Ten in finish() it checks to see if there's a thread waiting and wakes it up, before finishing. For task 2 I used a hashmap instead of a tuple to store the pairs of Threads and wakeupTimes. In waitUntil, the Thread is put to sleep and stored with its wakeupTime as a hash. In timerInterrupt it iterates over the hashmap, and readies and removes any threads that need to be woken up. I completed task 4 before task 3 so I could use Condition2 for Communicator, since we weren't allowed to use semaphores and therefore couldn't use Condition in it. For task 4, it was mainly just adding interrupt enable and disable and adding locks and then looping this for wakeAll. For Task 3 speak() acquires the lock and adds its word to a list. It then waits for a listener before waking the listener, decrementing the number of listeners and releasing its lock. In listen() it acquires the lock, increments the listeners, and waits for a speaker. It then wakes up the speaker, releases the lock, and returns the first entry in the list.

The only test I was not able to resolve and pass was test 3 for the Communicator. There is some synchronization issue. After some debugging and testing I eliminated the possibility that it was due to my Condition2, but was unable to resolve the error in my Communicator before submitting.