

Assignment 3

In this project we were to implement a rate monotonic scheduler with four threads, each with a different period of units. Each thread will execute the same `doWork()` function in which a 10x10 matrix composed of ones will multiply the board by itself in a specific column order. Each of the four threads have a counter that keeps track of how many times it runs and how many times it overruns. There is also a thread for the scheduler that it creates after the threads. Semaphores were initialized per each of the four threads (`sme#`). The scheduler works by implementing a for loop until it reaches 160 to cycle through 10 periods of 16 units. The scheduler priorities and increments the four threads. The four threads are incremented how many times they run in a time frame by the if statements using the remainder. The threads are prioritized by the semaphores using `sem_wait` in each of the thread creation and `sem_post` in the scheduler. The semaphores were utilized in conjunction with the boolean `running#` so that the scheduler can keep track of the amount of times the thread had overrun if it was still running.

Other Notes:

I struggled with the behind the test cases and how exactly to implement them. I don't think there should be overruns in the first two threads. When I created the scheduler thread before the other threads there were no overruns in the first two threads, but the first thread wouldn't run at all.