CHAPMAN UNIVERSITY, SCHMIDT COLLEGE, OPERATING SYSTEMS

# Project 3

Madison Pahl

May 4, 2017

## 1 RATE MONOTONIC SCHEDULER

**Design:** I created a rate monotonic system that includes four threads 0-3 each executing the doWork function different amounts of times:

1. Thread 0 : 1 time
2. Thread 1 : 2 times
3. Thread 2 : 4 times
4. Thread 3 : 16 times

Each thread class contains its own counter, timer, and run variables to keep track of individual thread actions. By using semaphores throughout the project, we are able to wait and signal the threads in order to sychonize the process, and create priorities for the threads to be accessed upon. Also, by using a timer, a set time is created for each thread to run, and if completion is not done by time the timer ends, then the next thread is to go, and an overrun is accounted for.

**Testing Scenarios:** From test case 2 (overrunning Thread 0) I found that although less runs were made for thread 0 and there was an overrun, the other threads were able to maintain their amounts, and did not reach any overruns themselves. This was the same for test case 3 (overrunning thread 2).

**Note about bonus timer:** I attempted the timer aspect of the project, however, I believe it is not timing and averaging from the 10 iterations, but outputs the last iteration and the time it took for the thread to complete its job.