Zach Jagoda

Professor Hansen

May 11, 2018

Project 3: Rate Monotonic Scheduler (RMS)

This project was designed using multiple java files, starting with RMS.java which kept a minimal amount of code and worked to start off the process of creating and starting the Threads in Scheduler.java. From Scheduler.java, the threads tried to create, start and join. Within that, they were sent to WorkThread.java where the calculations were done. The amount of completions (numCompletions) and overruns were counted here and run through createThread and doWork. printOut was then called at the end of RMS.java and printed out from Scheduler.java. In the background, Timer.java was also running, keeping track of the system time and taking in Semaphores in the function run().

Thread 1 ran for 1, Thread 2 ran for 2, Thread 3 ran for 4, and Thread 4 ran for 16 completions. Within the program, all Threads are created and started at the same time. They are then run individually given the time of the program, until one is completed the others lie in wait to kickoff; thus the use of the timer. The one thing I was not able to accomplish was writing a Thread that did not have any overruns, given more time to focus on this project solely I feel as if I may have been able to solve my issue as I believe I was close to getting the correct output for that specific case.