Lab 5 Report – Scheduling Simulation

**Procedure**

For this lab, we simulated three scheduling algorithms: rate monotonic (RMS), earliest deadline first (EDF), and least laxity first (LLF). To begin, we read an input file provided by the user on the command line. Then with that file we performed the three scheduling methods. Finally, we output the information in each algorithm for the user to analyze.

There were a lot of things we had to consider with how we wanted the simulation to function. Some questions we considered were: Should a task end if it misses a deadline? If a task doesn’t finish when interrupted, should it pick up where it left off, or start over? We decided to terminate a task if it missed a deadline, and when the task would pick up again, it would start from the beginning. In the output text file, for each algorithm, the information we reported includes missed deadlines, tasks released, preemptions, and tasks starting and ending.

For RMS, we took the tasks provided in the input file and calculated the priority and then created a schedule for the program to follow. Once the schedule is created, we then run the simulation and report the information. Since RMS is fixed priority, there are no preemptions. Once we got periodic tasks working, we then added aperiodic tasks that would run during slack time. The aperiodic tasks were given a deadline of 500 ms after their release time.

For EDF, we combined the aperiodic and periodic tasks because EDF does not care if a task is periodic or not. The simulation would then begin. During EDF we would calculate deadlines every time tick and switch tasks appropriately.

For LLF, we started with our EDF simulation and adjusted it to calculate slack on every time tick, using a strict approach. We did this to reduce the number of times a task would be preempted. If a task was already first in the priority, it would retain that priority if a task with the same slack time was there.

**Results and Conclusion**

The results for this lab were expected. For RMS we were able to create the schedule and prioritize the tasks. When we added the aperiodic tasks at first it would just run if it found any slack time but we quickly realized that this needed to be modified. The aperiodic task would run in slack causing the tasks to eventually miss their deadlines. In RMS there tended to be more missed deadlines, whereas with EDF and LLF there were fewer. With LLF there tended to be more preemptions and fewer missed deadlines than EDF. This showed that LLF was using more scheduling time, since there were more preemptions.

In conclusion, this lab taught us how the three different scheduling methods worked. We learned that LLF was always worse than EDF because it was more complicated to prioritize based on slack instead of deadlines, and had more preemptions. All three methods are valid for creating a schedule of tasks. RMA has the advantage of being simple because it does not dynamically change the priorities. EDF is more complex than RMA but it also allows dynamic priorities which can be helpful for a program that creates subtasks. LLF did not perform better than EDF and it would be better to use EDF for scheduling.