

## Embedded Real Time Systems Lab 2

Erik Sargent and Nate Tipton

### 1 Differences between RM and EDF

The total utilization for our `periodictest.txt` is 0.95. We ran this file through the RM and EDF algorithms. RM had a total of 44 preemptions, and EDF had a total of 40 preemptions. Neither algorithm had any deadline misses.

For the `aperiodictest.txt`, the total utilization was also 0.95, with the addition of the aperiodic tasks. The RM algorithm had a total of 52 preemptions and 9 deadline misses. The EDF algorithm had a total of 56 preemptions and 1 deadline miss.

For the `input.txt`, with periodic tasks only, RM had 0 preemptions and 0 deadline misses. EDF had a preemption and 0 deadline misses. With aperiodic tasks, RM had 9 preemptions and EDF had 5 preemptions.

### 2 Observations

When aperiodic tasks were introduced the RM algorithm missed 9 deadlines. The EDF algorithm allowed more preemptions which reduced the deadline misses significantly. This is most likely due to the freedom allowed in EDF to choose which task to schedule when deadlines are shared. EDF also places higher priority on jobs with soonest deadlines as opposed to RM which may continue to delay the job if a shorter period job is released.

### 3 Conclusion

RM and EDF create very similar schedules. In some instances, RM may be the better choice due to simpler implementation. However, from our observations EDF experiences fewer preemptions when handling aperiodic tasks.

### 4 Running the code

The code was developed using `cmake` and `C++11` on a Mac and Linux computer. If `cmake` is available, that would be the easiest method for compiling the program. If `cmake` is unavailable (like on the lab computers), we threw together a simple `Makefile` to compile the project on Windows.

We have included several input files, including two designed to test the aperiodic and periodic functions under high utilization (`aperiodictest.txt`, and `periodictest.txt`). The file `input.txt` includes the example given in the assignment description.