Assignment 2. Schedulability Analysis (100 points)

In this assignment, you are required to develop

1. An analysis program implementing various schedulability testing approaches for fixed priority schedule algorithms.
2. A comparative analysis of the schedulability of RM and DM algorithms, and fixed priority scheduling with least slack time using synthetic tasks sets.

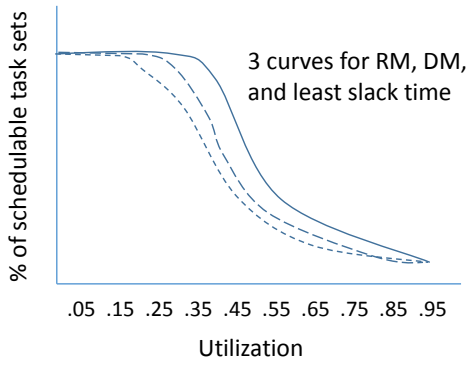
The analysis program written in C/C++, running in Linux without any IDE environment, reads in task parameters from an input text file and reports that each task set is schedulable or not. In this assignment, we assume that deadline is less than or equal to period for each task. An example input is:

7	\\ this file contains 7 task sets
3	\\ the 1st task set consists of 3 tasks
10.5 20.8 50	\\ WCET, deadline, and period of 1st task
5.2 18.9 60	\\ WCET, deadline, and period of 2nd task
2.4 100 205	\\ WCET, deadline, and period of 3rd task
4	\\ the 2nd task set consists of 4 tasks
……..	\\ task parameters for the rest of the 6 task sets

For each task set, the analysis program should consider RM and DM algorithms, as well as the fixed priority scheduling that assigns priorities according to slack times (deadline-WCET), i.e., the task with the least slack has the highest priority. Your program should choose utilization based analysis first. If the attempt is inconclusive, the program should apply response time based test.

For each task set, the report should indicate a sequence of methods that have been applied and the analysis result from each method. If any response time-based analysis method is used, the computed worst-case response times should be reported.

For the comparative analysis, you should present XY plots, as shown below, to illustrate the percentage of random task sets of various utilizations that are schedulable under RM and DM algorithms, and the fixed priority scheduling with least slack time. The analysis needs to generate synthetic tasks



sets and test the schedulability. To generate synthetic task set, you should adopt the same approach in [1, 2], and consider the following cases:

1. The task deadlines are uniformly distributed in $[C_i, T_i]$, and $[C_i+(T_i-C_i)/2, T_i]$.
2. Each task set consists of 12, 24, and 48 tasks.

You should use the analysis program from the first part of the assignment to determine the schedulability of each task set. You can modify the analysis program to automate the analysis process. For instance, the program can internally generate the task sets instead of reading them from an input file. Note that you need to present 6 XY plots as we need to consider combinations of the above cases:

- Plot 1: 12 tasks in each task set and the deadline distribution of $[C_i, T_i]$
- Plot 2: 24 tasks in each task set and the deadline distribution of $[C_i, T_i]$
- Plot 3: 48 tasks in each task set and the deadline distribution of $[C_i, T_i]$
- Plot 4: 12 tasks in each task set and the deadline distribution of $[C_i+(T_i-C_i)/2, T_i]$
- Plot 5: 24 tasks in each task set and the deadline distribution of $[C_i+(T_i-C_i)/2, T_i]$
- Plot 6: 48 tasks in each task set and the deadline distribution of $[C_i+(T_i-C_i)/2, T_i]$

For each plot, the utilization ranges from 0.05 to 0.95 with 0.1 step. For each utilization of each plot, you need to use 10,000 task sets. Periods of tasks for each task set are determined as shown in Section 6 of [1] with M=3. UUniFast algorithm [2] is used to determine utilizations of tasks. Then, WCET of each task can be calculated with the period and the utilization, i.e., WCET = Utilization*Period.

Reference

[1] Robert I. Davis, Attila Zabos, Alan Burns, "Efficient Exact Schedulability Tests for Fixed Priority Real-Time Systems," IEEE Transactions on Computers, vol. 57, no. 9, pp. 1261-1276, September, 2008.

[2] E. Bini and G.C. Buttazzo. “Measuring the Performance of Schedulability tests”. *Real-Time Systems*, vol. 30, no. 1–2, pp. 129–154, May 2005.

Due Date

This is assignment is due at 11:59pm on Mar. 06.

What to Turn in for Grading

- Create a working directory similar to Assignment 1 to include your source file(s), makefile, a test case of multiple task sets and the test result (in text), readme.txt, and the comparative analysis report (in pdf).
- Compress the directory into a zip file named *cse522-firstname-lastname.zip*. Please note that, for convenience, we only accept zip files. Points will be deducted if the name conversion and the directory structure are not followed.
- The comparative analysis report should include your analysis/explanation for the 6 XY plots. The analysis/explanation basically is to justify your results, e.g., why one scheduling algorithm has higher schedulability than the others?
- Comment your source files properly and write the readme file to describe how to use your software. Also, make sure there will be no warnings when compiling your source code.
- Submit the zip archive to Blackboard by the due date and time.