Assignment 2. Schedulability Analysis (100 points)

In this assignment, you are required to develop

1. An analysis program implementing various schedulability testing approaches for EDF, RM, and DM schedule algorithms.
2. A comparative analysis of the schedulability of EDF, RM and DM algorithms using synthetic tasks sets.

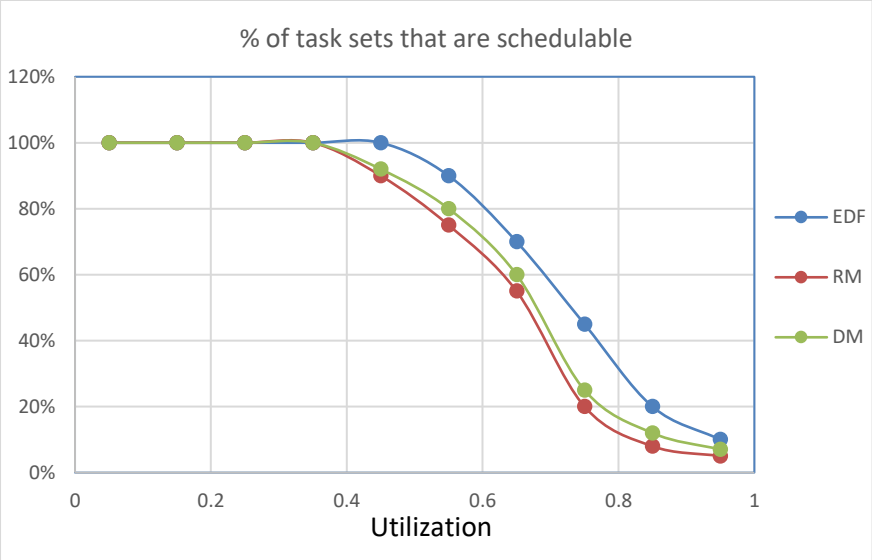
The analysis program written in C/C++, running in Linux without any IDE environment, reads in task parameters from an input text file and reports that each task set is schedulable or not. In this assignment, we assume that deadline is less than or equal to period for each task. An example input is:

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 EDF, RM and DM algorithms. Your program should choose utilization based analysis first. If the attempt is inconclusive, the program should apply the test approaches based on response time and loading factor.

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 loading factor approach, the first missing deadline 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 EDF, RM and DM algorithms. 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 10 and 25 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: 10 tasks in each task set and the deadline distribution of $[C_i, T_i]$
- Plot 2: 25 tasks in each task set and the deadline distribution of $[C_i, T_i]$
- Plot 4: 10 tasks in each task set and the deadline distribution of $[C_i+(T_i-C_i)/2, T_i]$
- Plot 4: 25 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 graph, you need to use 5,000 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

The due date for both parts 1 is 11:59pm, Feb. 26.

What to Turn in for Grading

- Create a working directory, named "cse522-teamX-assgn02", for the assignment to include your source files (.c and .h), makefile(s), readme, and your report (in pdf format). Compress the directory into a zip archive file named cse522-teamX-assgn02.zip. Note that any object code or temporary build files should not be included in the submission. Submit the zip archive to Blackboard by the due date and time.
- Please make sure that you comment the source files properly and the readme file includes a description about how to make and use your software. Don't forget to add each team member's name and ASU id in the readme file.
- There will be 20 points penalty per day if the submission is late. Note that submissions are time stamped by Blackboard. If you have multiple submissions, only the newest one will be graded. If needed, you can send an email to the instructor and TA to drop a submission.
- Your team must work on the assignment without any help from other teams and is responsible to the submission in Blackboard. No collaboration between teams is allowed, except the open discussion in the forum on Blackboard.
- Failure to follow these instructions may cause deduction of points.
- Here are few general rule for deductions:
    - No make file or compilation error -- 0 point for the part of the assignment.
    - Must have "–Wall" flag for compilation -- 5-point deduction for each warning.
    - 10-point deduction if no compilation or execution instruction in README file.
    - Source programs are not commented properly -- 10-20-point deduction.

- ASU Academic Integrity Policy (http://provost.asu.edu/academicintegrity), and FSE Honor Code (http://engineering.asu.edu/integrity) are strictly enforced and followed.