Assignment 4. Schedulability Analysis (100 points)

In this assignment, you are required to develop an analysis program implementing various schedulability testing approaches for EDF, RM, and DM schedule algorithms, and running on a Linux host machine (e.g. Ubuntu).

The analysis program should be written in C/C++, running in Linux without any IDE environment. It should reads in task parameters from an input text file and reports (to be written in an output text file) 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 task parameters 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. The report should also 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.

Due Date

The due date is 11:59pm, April 23.

What to Turn in for Grading

- Your submission is a zip archive, named RTES-LastName-FirstInitial_04.zip, that includes source files (.c/.cpp and .h), Makefile, and readme. The zip archive should be submitted to Canvas by the due date. The executable generated from Makefile should be named as “assignment4”. Note that any object code or temporary build files should not be included in the submission.
- 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 your 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 Canvas. If you have multiple submissions, only the newest one will be graded. If needed, you can send an email to the instructor to drop a submission.
- The assignment must be done individually. No collaboration is allowed, except the open discussion in the forum on Canvas. The instructor reserves the right to ask any student to explain the work and adjust the grade accordingly.
- Failure to follow these instructions may cause deduction of points.
- Here are few general rule for deductions:
  - Cannot apply patch or compilation error -- 0 point for 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.

How your program will be graded:

1. Unzip your submission
2. Run make
3. Run your analysis program with command "./assignment4 inputfile outputfile"
4. Examine your source code and report.