Introduction to Real-Time Systems
실시간 시스템 개론

Jinkyu Lee

Dept. of Computer Science and Engineering,
Sungkyunkwan University (SKKU)

Homework 1: submission information

- 50 points
  - Note that each exercise is worth 10 points.

- Due date: 11/13 Wed 23:59 (no late submission accepted)
  - We will discuss this homework in the class on 11/14 10:30-11:45

- Compress the code and report into a yourid_HW1.zip file and submit it to i-campus.

- Code
  - Submission format
    - Format: yourid_HW1.py
    - You file should work on Python 3.8.10 at Linux.

- Report
  - Submission format
    - Format: yourid_HW1.pdf
    - Use the same template as exercise (power point style). But, your should submit it as the PDF file type.
  - Even if your code evaluation fails, you may get up to 30 points for your report.
    - Examples of code evaluation failure include (i) not to adhere to the required format (including submittion file type), (ii) to fail to execute, (iii) to produce incorrect result, etc.
  - If suspected of copying the code, the teaching assistant may ask for an explanation based on the report.

Homework 1

- [Code] Make a program that randomly generates the parameters of 100 set of real-time tasks (T_i, C_i, D_i) satisfying the conditions below.
  - The program takes three command line inputs: the number of tasks *n*, the utilization of the task set *U* (<1), and *v* which determines task has implicit ($v=0$) or constrained deadline ($v=1$).
  - The utilization of each task is determined by UUniFast algorithm. For the UUniFast algorithm, see Section 3.6 of the paper [1].
  - T_i of each task is between 100 and 1000.
  - C_i should be grater than 0.
  - All the parameters are integers.
  - The difference between the target utilization and the utilization of the generated task set, which arises due to integer parameters, is not subject to evaluation.

[1] E. Bini and G. Buttazzo. 2005. Measuring the performance of schedulability tests. Real-Time Systems (2005).

Homework 1

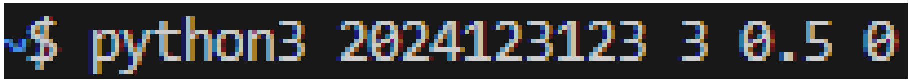
- [Code] Make a program that randomly generates the parameters of 100 set of real-time tasks (T_i, C_i, D_i) satisfying the conditions below.

  - The generated 100 set of tasks should be saved in the ‘*./output*’ folder as ‘*yourid_{n}_{U}_{v}.txt*’. (*n*, *U* and *v* are input values. Please refer to the previous slide for details.)
  - Each line of the text file represents one set of tasks, formatted as follows.
  - {n} {U} {v} {T_1} {C_1} {D_1} … {T_n} {C_n} {D_n}
  - The name of the program should be “yourid_HW1.py”
  - Each program execution should be finished within 5 seconds.

Homework 1

- [Code] Make a program that randomly generates the parameters of 100 set of real-time tasks (T_i, C_i, D_i) satisfying the conditions below.

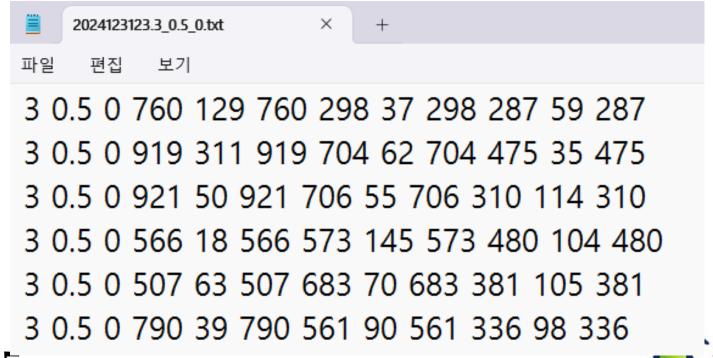
  - Example of program execution.



```
~$ python3 2024123123 3 0.5 0
```

    - When the above command is executed, a text file with 100 lines should be generated. The number of processors, number of tasks and utilization of each set of tasks *n*=3 and *U*=0.5, respectively, and all tasks have implicit deadline (*v*=0).
    - Use “sys” module of python for command line arguments.\

  - Example of output file ‘./output/yourid_3_0.5_0.txt’



```
3 0.5 0 760 129 760 298 37 298 287 59 287
3 0.5 0 919 311 919 704 62 704 475 35 475
3 0.5 0 921 50 921 706 55 706 310 114 310
3 0.5 0 566 18 566 573 145 573 480 104 480
3 0.5 0 507 63 507 683 70 683 381 105 381
3 0.5 0 790 39 790 561 90 561 336 98 336
```