Introduction to Real-Time Systems
실시간 시스템 개론

Jinkyu Lee

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

Homework 2

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

- Due date: 11/27 Wed 23:59 (no late submission accepted)
  - We will discuss this homework in the class on 11/28 10:30-11:45

- Compress the code and report into a yourid_HW2.zip file and submit it to i-campus.

- Code [40 points] (8 test cases, 5 points each)
  - Submission format
    - Format: yourid_HW2.py
    - You file should work on Python 3.8.10 at Linux.

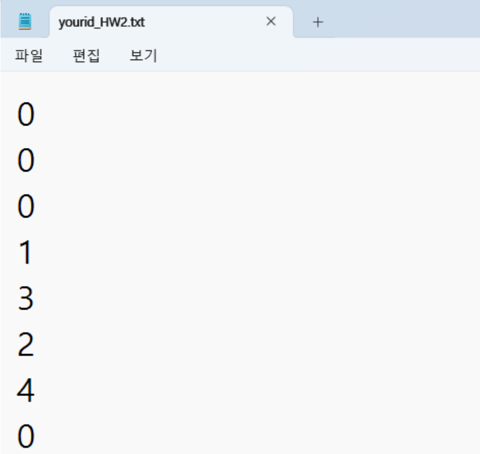
- Report [10 points]
  - Submission format
    - Format: yourid_HW2.pdf
    - Use the same template as exercise (power point style). But, your should submit it as the PDF file type.
  - If suspected of copying the code, the teaching assistant may ask for an explanation based on the report.

Homework 2

- [Code] Make a scheduler that executes the set of real-time tasks generated by HW1, satisfying the conditions below.
  - The scheduler program takes three command line inputs: the *“input_file_name.txt”*, a prioritization policy (one of *FCFS*, *SJF*, *RM* and *EDF*), and a preemption option *p* or *np* (*p* for preemptive and *np* for non-preemptive).
  - The scheduler considers only a single-processor environment, so only one task is executed at any given time instant.
  - The schedule is work-conserving; in other words, the computing platform should not be idle as long as there is at least one active task.
  - The scheduler is preemptive or non-preemptive which is given by command line input.
  - The scheduler can select the following prioritization policy: FCFS, SJF, RM and EDF, also given by command line input.
  - The scheduler schedules each set of tasks for 100,000 time units. If a deadline miss occurs during this period, it outputs the index of the missed task (with task indices starting from 1). If no deadline miss occurs, it outputs 0.
  - The program takes task set file which is same format from Homework 1 and generates “yourid_HW2.txt” at ‘./output’ folder.
  - If some jobs have the same priority, the job with a lower task index is assigned a a higher priority. (Task indices are assigned based on their order in the input.)

Homework 2

- [Code] Make a program, which is a scheduler that satisfies the condition
  - The number of lines of the input file and the output file should be same. For example, if the input file has 100 lines, the ‘yourid_HW2.txt’ should also have 100 lines. And there is only one number for each line.
    - The input file: see Homework 1
  - Example of program execution.
    - # python3 ‘yourid_HW2.py’ ‘input_file.txt’ EDF p
    - As explained in the previous slide, the input file name is ‘input_file.txt’, the prioritization policy is EDF and p means the scheduler is preemptive. If there is np instead of p, the scheduler should be non-preemptive.
  - Example of output file.



```
0
0
0
1
3
2
4
0
```

Homework 2

- [Evaluation] The total score is 50 points, with 40 points allocated to the code and 10 points to the report.
  - The code score is determined by 8 test cases, each worth 5 points.
  - The report may include an analysis of the implemented prioritization policy, code, and other relevant aspects, and is graded between a minimum of 0 and a maximum of 10 points.
  - If you have any questions regarding the Homework 2, please contact the TA via the email below.
    - E-mail : yuns0509@g.skku.edu