CSOPESY Term 2, AY 2020 – 2021

Long Exam 1 – Multilevel Feedback Queue

Groupings:	Same grouping as Mini MP 1
Deadline:	April 29 , 2021 (Thursday) 11:59 PM
Percentage:	20%

Deliverables:

`Zip` file containing:

- Program source codes – `c` file/s

Submission guidelines: Submit the `zip/c` file to AnimoSpace

Place the following comment at the beginning of the file

```
***************************************************************
Names:  <Surname1, Name1>
        <Surname2, Name2>
Group:   <Group Number>
Section:  <Section>
***************************************************************
```

Filename format:

```
CSOPESY-Project1-<Section>-Group<#>.c
                 or
CSOPESY-Project1-<Section>-Group<#>.zip
```

SPECIFICATIONS

For convention and consistency, the project should be implemented in **C**.

Implement a simulation of a multilevel feedback queue(MLFQ). The MLFQ should follow these set of rules:

Rule 1: If Priority(A) > Priority(B), A runs (B doesn't).

Rule 2: If Priority(A) = Priority(B), A & B run in RR

Rule 3: When a job enters the system, it is placed at the highest priority (the topmost queue).

Rule 4: Once a job uses up its **time quantum** at a given level (regardless of how many times it has given up the CPU e.g. during an IO burst), its priority is reduced (i.e., it moves down one queue).

Rule 5: After some time period **S**, move all the jobs in the system to the topmost queue. This is what we refer to as a **priority boost**.

Input

The program accepts the filename of an input text file which contains the following:

- The first line contains 3 integers separated by space $X\ Y\ S$
- X denotes the number of queues where $2 \leq \mathrm{X} \leq 5$
- Y denotes the number of processes where $3 \leq Y \leq 100$
- S denotes the time period for the **priority boost**.
- There will be X lines of space-separated integers $A\ B\ C$ where A is the queue ID, B is the priority, and C is the time quantum for this queue.
- All queues should use round robin scheduling and all times are in milliseconds.
- There will be Y lines of space-separated integers F G H I J where F is the process ID, G is the arrival time, and H is the total execution time and I is the length of IO burst time and J dictates how often(every N milliseconds) the IO burst happens. For CPU bound processes with no I/O, I and J should be 0. Assume that context switch time is 0.
  - For example the input **`1 1 20 1 2`** means 1 is the process ID, arrival time is 1 ms, the total execution time is 20 ms, the length of I/O is 1 ms, and lastly IO happens every 2 ms.

Output

- The output should print Y lines of processes that show print the following information: Notes: Q[X] is the queue ID and [IO] corresponds to an IO burst.

```
P[A]
Q[X] Start time: <S1> End time: <E1>
Q[X] Start time: <S2> End time: <E2>
[IO]  Start time: <S3> End time: <E4>
…
Q[X]  Start time: <SN> End time: <EN>
Waiting time: <WT>
Turnaround time: <TT>
************************************
P[A]
Q[X]  Start time: <S1> End time: <E1>
Q[X]  Start time: <S2> End time: <E2>
…
Q[X]  Start time: <SN> End time: <EN>
Waiting time: <WT>
Turnaround time: <TT>
************************************
```

Figure 1. Output per process

- An additional last line should show:

```
Average waiting time: <AWT>
```

Figure 2. Last line of the output should display the average waiting time.

where *AWT* indicates the average waiting time of the processes. The output does **NOT** have to be sorted according to the process ID *A*.

REQUIRED PROGRAM INTERACTION

There should be minimal program interaction as shown in the sample runs below. The program will just ask the user to input the name of the input text file. If the text file exists, its contents will be processed, the output will be displayed, and the program terminates. If the text file does not exist, the program outputs "`<FILENAME.TXT> not found.`" error message and then terminates.

DELIVERABLES

Submit a zip file or a c file containing the source code files via AnimoSpace. **DO NOT INCLUDE ANY EXECUTABLE FILE in your zip file submission.**

HONESTY POLICY AND INTELLECTUAL PROPERTY RIGHTS

Honesty policy applies. Please take note that you are **NOT allowed to borrow and/or copy-and-paste** – in full or in part any existing related program code from the internet or other sources (such as printed materials like books, or source codes by other people that are not online). You should develop your own codes from scratch by yourselves, i.e., in cooperation with your groupmates.

According to the handbook (5.2.4.2), "faculty members have the right to demand the presentation of a student's ID, to give a grade of 0.0, and to deny admission to class of any student caught cheating under Sec. 5.3.1.1 to Sec. 5.3.1.1.6. The student should immediately be informed of his/her grade and barred from further attending his/her classes."

RUBRIC FOR GRADING

Test cases will be provided during the demo.