German University in Cairo
Faculty of Media Engineering and Technology
Dr. Aysha Alsafty
Eng. Fadwa Elhussini
Eng. Eslam Osama

CSEN 602-Operating Systems, Spring 2018
Mini-project 2
Deadline: 30.3.2018
Grade: 7%

In this mini-project you are required to simulate the behavior of both the Round-Robin and lottery scheduling algorithms using C. You are provided two baselines C files *RR.c* and *Lottery.c* that you will be augmenting and submitting.

To simplify the simulation, assume all processes given have one long CPU burst only, no I/O time. Furthermore, you may assume that all processes arrive at the same time t_0. You may use an array or other data structures to simulate the Ready queue of the scheduler.

Bonus: 0.75%
Extend your simulation of Round-Robin to support variable process arrival times.

Processes

First of all, you will need to implement a struct for the process, in order to be able to deal with all the attributes of the process as a single entity. The attributes of the process are the following:

- Process ID
- Arrival time
- CPU burst time
- Start time
- End time
- Number of tickets

All times are integers and in milliseconds. You can also add additional attributes if you need to, however the provided ones are the essential ones that should be included in your implementation. You are provided with several input CSV files, that will be used as test cases. For each input file, we provided the expected corresponding output from both scheduling algorithms.The output of your

code should be written to files following the format of the output files provided for each one of the algorithms.

The first step of your implementation is to read in the input files and proceed with creating the processes. You may assume that all input files follow this format:

<quantum length in milliseconds>
<number of total tickets>
<processID ArrivalTime CPU-burst Tickets>
<processID ArrivalTime CPU-burst Tickets>
...
<processID ArrivalTime CPU-burst Tickets>

The total tickets attributes will be used in lottery scheduling only. Please note, that hard-coding the input is not accepted and will lead to substantial deduction. You are also not allowed to change the input files or their format.

Quantum Simulation

Simulating the quantum means keeping the CPU busy while the process is running within it's quantum. In order to achieve that, you will be looking into the timers library in C. The *timer.h* is the place to start looking. As discussed during lectures and tutorials,please assume that when a process completes execution mid-quantum, the scheduler immediately picks the next process ready to run to start execution.

You will notice that the quantum values provided in input files are larger than typical values. This is to show case the scheduling behavior.

Part 1: Round Robin Scheduler

In this part you are to simulate the behavior of the round robin algorithm as discussed in class. You may refer to the text book for more elaboration. As the processes arrive in the same time, please assume the scheduler will place the processes in its Ready queue according to the order they appear in the input file.

After each process is done execution, calculate and print the waiting time and the turn around time for the process. In the end of the simulation you are to output the total average turn around and waiting time for all processes. You will find sample expected output in the files **Output1_RR** and **Output2_RR** for the input files **Input1.in** and **Input2.in** respectively.

German University in Cairo
Faculty of Media Engineering and Technology
Dr. Aysha Alsafty
Eng. Fadwa Elhussini
Eng. Eslam Osama

Part 2: Lottery Scheduler

In this part you are to simulate the behavior of the lottery algorithm. Each process is assigned a fixed number of tickets. A lottery draw is run at the beginning of each quantum to select which process will get this quantum. The process that owns a ticket with the value equal to the ticket drawn in the lottery is selected to execute. To give different processes longer CPU time, each process is assigned a different number of tickets, which makes it more likely for this process to get selected. You may refer to the text book for more elaboration. The input file includes the number of tickets assigned to each process and the total number of tickets available in the system. **Output1_LOT** and **Output2_LOT** for the input files **Input1.in** and **Input2.in** respectively You may also refer to the following link for a numeric example on lottery scheduling: `https://www.geeksforgeeks.org/operating-system-lottery-scheduling/`

Deadline:

- The project deadline is on 30/3 at 11:59 PM .
- **No late** submissions will be accepted.
- Cheating cases will be graded by 0 for all teams involved.
- It is your responsibility to make sure that the files were uploaded successfully to the website.

Submission guidelines:

- The submission will be through the following link: `https://goo.gl/forms/Yq64lvy1HZrGHRRO2`
- The project files should be uploaded on a Google drive and you will be submitting the link to this drive file.
- The drive should have both your C files *RR.c* and *Lottery.c*.
- Please check the updated teams list on the MET website for your team number.