German International University
Faculty of Media Engineering and Technology
Dr. Alia El Bolock

CSEN 602 Operating Systems, Spring 2022
Milestone 2

Due Date: 18.06.2022 at 11:59PM

Scheduling

A Scheduling Algorithm is an algorithm used to schedule processes to ensure that all processes get a chance to execute. As mentioned in the lecture, there are many different scheduling algorithms to schedule processes.

Milestone 2

In this milestone, you are asked to create these three scheduling algorithms using JAVA:

1. **First Come First Serve (FCFS):** Processes are executed on a first come, first serve basis (queue, based on arrival time).
2. **Shortest Job First (non-preemptive):** Processes are executed depending on their execution time, the shortest process being the first to execute. The processor should know in advance how much time each process needs for execution, so that it can order the processes accordingly.
3. **Round Robin:** Each process is assigned a fixed time slot in a cyclic way.

Program Syntax

We need only THREE methods created, one for each scheduling algorithm, these methods are to follow the following conventions:

- **Method names:** Scheduler_FCFS(), Scheduler_SJF() and Scheduler_RR() with Q=2.
- **Method parameters/inputs:** Each method takes one input which is a string representing the processes, time to run and arrival time. The string will have a specific format which is discussed later.
- **Method return value:** Each method will return a string, which represents the order in which the processes will execute. The output string format is also discussed later.

String Formats

The **INPUT** string will be formatted as follows. For example "a,b,c;0,0,0;10,20,30" is a valid input string. It has three sections, each is separated with a ";"

1. **Section 1** represents the names of the processes and is comma separated. In the above example we have three processes named *a, b & c*.

2. **Section 2** represents the arrival time of the processes and is comma separated. In the above example we have all processes arriving at time 0.

3. **Section 3** represents the time needed for each process to execute and is comma separated. Here we have process **a** requiring **10** time units, process **b** requiring **20** time units and process **c** requiring **30** time units. The unit of time is irrelevant, we only care about the number itself and 0 represents the beginning of time.

For the **OUTPUT** string, we only want the order and execution time in which the processes will be executed. For example "a(10),b(20),c(30)" is a valid output string. It shows that process a will execute for 10 time units, followed by process b for 20 time units and finally process c for 30 time units. Time units between brackets is important when it comes to Round Robin as it is a preemptive algorithm and the process does not execute to completion. You must note that the naming of the processes is **case-sensitive**. This means that if the input has a process named **"a"**, it should also be named **"a"** in the output and we will consider **"A"** to be a different process and vice versa. The naming is very crucial and must be preserved, otherwise the test cases might not pass.

Example Runs

You can assume we will only use valid strings. No need to handle wrong/invalid inputs within your code. You must follow the below convention for both the input and output so please follow it precisely.

Input: String input = "A,B,C,D,E;0,2,4,5,8;3,6,4,5,2"
Outputs:

- *Scheduler_FCFS(input)* = "A(3),B(6),C(4),D(5),E(2)"
- *Scheduler_SJF(input)* = "A(3),B(6),E(2),C(4),D(5)"
- *Scheduler_RR(input)* = "A(2),B(2),A(1),C(2),B(2),D(2),C(2),E(2),B(2),D(3)" OR "A(2),B(2),A(1),C(2),B(2),D(2),C(2),E(2),B(2),D(2),D(1)" both are accepted.

Input: String input = "Hi,Bye;0,0;4,2"
Outputs:

- *Scheduler_FCFS(input)* = "Hi(4),Bye(2)"
- *Scheduler_SJF(input)* = "Bye(2),Hi(4)"
- *Scheduler_RR(input)* = "Hi(2),Bye(2),Hi(2)"

Hints

You can implement the project with none of these hints, however you might find it helpful to lookup these topics to assist you in the project.

- String split in Java: https://www.geeksforgeeks.org/split-string-java-examples/
- Queues in Java: https://www.geeksforgeeks.org/queue-interface-java/

Work Distribution & Grading

The grading for this assignment will be completely automated, meaning, we will not be having any face-to-face evaluations nor will we inspect your code. Your code will be tested using specified test cases and you will be graded accordingly. You are advised to split the work by assigning a method/algorithm to each member as they are completely independent, but do keep in mind that Round Robin is the most challenging of the bunch and the grade will be equal on all team members.

Project Deliverable and Submission

The project should be submitted as ONE zip folder containing the JAVA files you created. Please make sure to name your folder as follows, Team_Name (example: Team_01). Late submissions will not be accepted. Submission will be through the following link: https://forms.gle/A39hkD8k2RVCd1GW9