**Concordia University**

**comp346 - Summer 2020**

**Operating Systems**

**Programming assignment 2**

**Deadline:** By 11:59pm, Friday July 26 2020

**Late Submission:** No late submission

**Teams:** The assignment can be done individually or in teams of 2 or 3. Submit only one assignment per team.

**Purpose:** The purpose of this assignment is to apply develop the different CPU scheduling algorithms and compare all of them to each other.

* **Problem specification.**

In this assignment, you will create a simulation for a CPU scheduler. The number of CPU’s and the list of processes and their info will be read from a text file. The output, of your simulator will display the execution of the processes on the different available CPU’s. The simulator should also display:

* The given info of each process
* CPU utilization
* The average wait time
* Turnaround time for each process
* CPU response time for each process
* **Implementation.**

You need to start your demo with a UML diagram.

You need to implement the following algorithms:

* FCFS
* SJB
* SRTF
* RR with a user given q
* **Sample input file**

numOfCPUs: 4

// list of processes to be scheduled

// processID arrivalTime totalExecTime IO\_RequestAtTime IO\_RequestAtTime ...

p0 0 10 2 5 8

p1 2 3

p2 10 7 1 2 3

The first line indicates how many CPU’s/cores do we have in the system. This sample file indicates that we have 4 CPU’s/cores

Any line that starts with // is a comment

Then we have a list of the processes. The first line:

p0 0 10 2 5 8

means we have process with process ID p0 that arrives at time 0. It needs 10 time on the CPU to finish executing its code. 2 time units after it has started, it asks for I/O. Any I/O request is fulfilled in 2 time units. At time 5 of its execution, p0 asks for I/O again. At time 8 of its execution, it asks for I/O again. There is no limit for how many times a process asks for I/O.

p1 does not ask for any I/O.

p2 arrives at time 10. 1 time unit after it stared execution, it requests I/O.

* **Evaluation.**

You will be evaluated mostly on the implementation of the required functionality, the implementation of the algorithms, and the correctness of the output.

***If you miss your demo time, you will receive 0 credit for the assignment.***

***Evaluation Criteria***

|  |  |
| --- | --- |
| **Criteria** | **Marks** |
| The overall design of the project | 20% |
| Implementation of FCFS | 15% |
| Implementation of SJB | 20% |
| Implementation of SRTF | 20% |
| Implementation of RR with a user given q | 25% |

**Required documents.**

* Source code in Java
* Output test cases

**Submission.**

Create one zip file, containing the necessary files (.java, .txt and test cases). If the assignment is done individually, your file should be called pa1\_studentID, where pa1 is the number of the assignment and studentID is your student IDnumber. If the work is done in a team of 2 or 3 people, the zip file should be called pa1\_studentID1\_studentID2 or pa1\_studentID1\_studentID2\_studentID3 where studentID1, studentID2, and studentID3 are the studentID numbers of each student.

The zip file should be uploaded to Moodle before the due date. No late submissions are accepted.