IT308: CPU Scheduling – Exercises and Prog assignment

This lab will allow you to explore the basics of CPU scheduling. To get checked off on this lab, you will have to solve Exercise 1 and 2 below, and show your work to a TA.

This handout also includes a description of the programming assignment 1 in which you will write your own CPU scheduling simulator. This assignment is to be submitted via moodle and is due on 24th January, 23:50 hrs.

Exercise 1

Suppose we have 3 processes.

Process ID	Required CPU time	Arrival Time
A	200 msec	50 msec
B	400 msec	150 msec
C	300 msec	0 msec

For each of the following scheduling algorithms, draw the execution timeline and compute the **average response time**, **average wait time**, and **average turnaround time**. Assume round robin has 100 msec time slice. Assume zero context switching cost. You should use the following definition of **wait time**: the total time the process is waiting in the ready queue.

(a) FIFO (A: 200, 200, 500)
(b) Round-Robin (A: 66.7, 300, 600)
(c) SJF (A: 200, 200, 500)
(d) SCTF (A: 117, 183, 483)

Exercise 2

You are given 3 processes P1, P2 and P3 with the same arrival time. The scheduler will break ties according to process-number and start with the one with the smallest number i.e. P1.

- P1 alternates between 1 second of CPU burst and 1 second of I/O, for a total of 3 seconds of computation and 2 seconds of I/O. (i.e. it starts and ends with a CPU burst).
- P2 has 2 seconds CPU burst size and does no I/O.
- P3 has 3 seconds CPU burst size and does no I/O.

Draw the execution timeline for the MLFQ scheduling policy; clearly show which queues the processes are running in. There are two queues which use round robin scheduling. Queue-1 has 1 second time slice and Queue-2 has 3 seconds time slice. If a process executing in Queue-1 does not finish its CPU burst within its time quantum, then it is moved to Queue-2. A process in Queue-1 has priority over the one in Queue-2 and will preempt it as soon as it becomes ready to run.

What is the average turnaround time? (A: 6.3) What is the average waiting time? (A: 3). Note: time spent doing I/O is not to be counted in the waiting time.

Programming Assignment: CPU Scheduling Simulator

You will write a C program (you may also use Java) to implement a simulator with different scheduling algorithms. The simulator selects a task to run from ready queue based on the scheduling algorithm. Since the assignment intends to **simulate** a CPU scheduler, it does **not** require any actual process creation or execution. When a task is scheduled, the simulator will simply print out what task is selected to run at a time.

The selected scheduling algorithms to implement in this assignment are

- First Come First Serve (**FCFS**)
- Round Robin (**RR**)
- Shortest Job First (**SJF)**

Process Information

The task information will be read from an input file. The format is

```
pid arrival_time burst_time
```

All of fields are of integer type where:

`pid` is a unique numeric process ID

`arrival_time` is the time when the task arrives in the unit of milliseconds

`burst_time` is the CPU time requested by a task, in the unit of milliseconds

The time unit for **`arrival_time`**, **`burst_time`** and **`time_quantum`** is millisecond.

Here is a sample input file:

```
1 0 10
2 0 9
3 3 5
4 7 4
5 10 6
6 10 7
```

Command-line Usage and Examples

Usage: **`sched input_file [FCFS|RR|SJF] [time_quantum]`**

where *input_file* is the file name with task information. FCFS and RR are the names of scheduling algorithms. The **`time_quantum`** only applies to RR. FCFS, SJF are non-preemptive while RR is **preemptive**. **The last argument is needed only for RR.** (See following table for more examples)

Examples Description:

`sched input.1 FCFS` FCFS scheduling with the data file "input.1"

`sched input.1 RR 2` Simulate RR scheduling with time quantum 2 milliseconds (4th parameter is required even for quantum 1 millisecond) with the data file "input.1"

`sched input.1 SJF` Shortest Job First Scheduling with data file "input.1"

Design Hints

No sample code is given for this project. However, here is a possible design logic you can refer to. The simulator first reads task information from the input file and stores all data in a data structure. Then it starts simulating a scheduling algorithm in a **time driven** manner. At each time unit (or slot), it adds any newly arrived task(s) into the ready queue and calls a specific scheduler algorithm in order to select appropriate task from ready queue. When a task is chosen to run, the simulator prints out a message indicating what process ID is chosen to execute for this time slot. If no task is running (i.e. empty ready queue), it prints out an "idle" message. Before advancing to the next time unit, the simulator should make all necessary changes in task and ready queue status.

Requirements:

The project requires you to simulate FCFS, SJF and RR scheduling for given tasks and to compute the **average waiting time, average response time and average turnaround time**.

- **Implement scheduling algorithm for FCFS, SJF and RR.** The program should schedule tasks and print progress of task every unit time (millisecond) as shown in sample outputs.
- **Print statistical information**. As soon as all tasks are completed, the program should compute and print 1) **average waiting time**, 2) **average response time** and 3) **average turnaround time**.

Note: if you use static array to implement ready queue structure, you can assume the maximum queue length is 20.

Suggested Methodology

- Implement one scheduling algorithm at each step.
- You can use a circular linked list as the queue structure.
- You can use a data structure similar to a Process Control Block (PCB) for each task, though it will be much simpler.

Sample outputs

Here is a sample input file:

```
% more input.1
1 0 10
2 0 9
3 3 5
4 7 4
5 10 6
6 10 7

% sched
Usage: sched input_file FCFS|SJF|RR [quantum]
```

And here is a sample output:

```
% sched input.1 FCFS
Schdeuling algorithm: FCFS
Total 6 tasks are read from "input.1". press 'enter' to start...
=================================================================
<system time 0> process 1 is running
<system time 1> process 1 is running
<system time 2> process 1 is running
<system time 3> process 1 is running
<system time 4> process 1 is running
<system time 5> process 1 is running
<system time 6> process 1 is running
<system time 7> process 1 is running
<system time 8> process 1 is running
<system time 9> process 1 is running
<system time 10> process 1 is finished.......
<system time 10> process 2 is running
<system time 11> process 2 is running
<system time 12> process 2 is running
<system time 13> process 2 is running
<system time 14> process 2 is running
<system time 15> process 2 is running
<system time 16> process 2 is running
<system time 17> process 2 is running
<system time 18> process 2 is running
<system time 19> process 2 is finished.......
<system time 19> process 3 is running
<system time 20> process 3 is running
<system time 21> process 3 is running
<system time 22> process 3 is running
```

```
<system time 23> process 3 is running
<system time 24> process 3 is finished.......
<system time 24> process 4 is running
<system time 25> process 4 is running
<system time 26> process 4 is running
<system time 27> process 4 is running
<system time 28> process 4 is finished.......
<system time 28> process 5 is running
<system time 29> process 5 is running
<system time 30> process 5 is running
<system time 31> process 5 is running
<system time 32> process 5 is running
<system time 33> process 5 is running
<system time 34> process 5 is finished.......
<system time 34> process 6 is running
<system time 35> process 6 is running
<system time 36> process 6 is running
<system time 37> process 6 is running
<system time 38> process 6 is running
<system time 39> process 6 is running
<system time 40> process 6 is running
<system time 41> process 6 is finished.......
<system time 41> All processes finish ....................
============================================================
Average waiting time : 14.17
Average response time : 14.17
Average turnaround time: 21.00
============================================================

% sched input.1 RR 2
Schdeuling algorithm: RR
Total 6 tasks are read from "input.1". press 'enter' to start...
============================================================
<system time 0> process 1 is running
<system time 1> process 1 is running
<system time 2> process 2 is running
<system time 3> process 2 is running
<system time 4> process 1 is running
<system time 5> process 1 is running
<system time 6> process 3 is running
<system time 7> process 3 is running
<system time 8> process 2 is running
<system time 9> process 2 is running
<system time 10> process 1 is running
<system time 11> process 1 is running
<system time 12> process 4 is running
<system time 13> process 4 is running
<system time 14> process 3 is running
<system time 15> process 3 is running
<system time 16> process 5 is running
<system time 17> process 5 is running
<system time 18> process 6 is running
<system time 19> process 6 is running
<system time 20> process 2 is running
```

```
<system time 21> process 2 is running
<system time 22> process 1 is running
<system time 23> process 1 is running
<system time 24> process 4 is running
<system time 25> process 4 is running
<system time 26> process 4 is finished.......
<system time 26> process 3 is running
<system time 27> process 3 is finished.......
<system time 27> process 5 is running
<system time 28> process 5 is running
<system time 29> process 6 is running
<system time 30> process 6 is running
<system time 31> process 2 is running
<system time 32> process 2 is running
<system time 33> process 1 is running
<system time 34> process 1 is running
<system time 35> process 1 is finished.......
<system time 35> process 5 is running
<system time 36> process 5 is running
<system time 37> process 5 is finished.......
<system time 37> process 6 is running
<system time 38> process 6 is running
<system time 39> process 2 is running
<system time 40> process 2 is finished.......
<system time 40> process 6 is running
<system time 41> process 6 is finished.......
<system time 41> All processes finish .....................
============================================================
Average waiting time : 22.50
Average response time : 4.00
Average turnaround time: 29.33
============================================================
```