

## Programming Assignment Two

Write a program to implement CPU scheduling with Multilevel Feedback Queue. Please refer to slides 28-29 of Lecture 5 *ch5\_CPU\_Scheduling.pdf* for Multilevel Feedback Queue and its example.

At the beginning of all .c and .h files, put the comment including the information: author, student id, date and file purpose.

### Task 1

Write a program that include three queues, Q1, Q2, and Q3.

- Q1, Q2 both use FCFS+RR, Q3 uses FCFS.
- Q1: quantum is 4, Q2: quantum is 8.
- Each process will go to Q1 first.
- If a process is not finished in one quantum in Q1, it will be moved to Q2; If it is not finished in one quantum in Q2, it will be moved to Q3 where it will be finished according to FCFS rule.
- Processes in Q1 have highest priority while in Q3 have lowest priority.

In your program, you should ask the user to enter the burst time in the sequence of their arrival (assume that they all arrive at 0 in the sequence P1, P2, P3, ..., Pn). Then your program will output the processes in the sequence they use CPU. Here are two examples:

#### Example 1:

```
=====
Enter the number of processes to schedule: 3
Enter the burst time of P1: 17
Enter the burst time of P2: 15
Enter the burst time of P3: 7
The scheduling sequence is: P1, P2, P3, P1, P2, P3, P1, P2
=====
```

#### Example 2:

```
=====
Enter the number of processes to schedule: 2
Enter the burst time of P1: 3
Enter the burst time of P2: 10
The scheduling sequence is: P1, P2, P2
=====
```

### Task 2

Task 2 is like Task 1, using Multilevel feedback queue to CPU scheduling with more rules.

- 1) We assume that there are 5 levels of priority, indicated from 1 to 5, where 1 is the highest priority and 5 is the lowest priority.
- 2) User will input the burst time and the priority for each process. There are at most 5 queues, from Q1 to Q5, each contains processes with corresponding priority, i.e., Q1 for processes with priority 1, Q2 for processes with priority 2, and so on.
- 3) The queue with priority 5 uses FCFS algorithm, while others use FCFS+RR. User inputs RR quantum.
- 4) If a process cannot finish during the given quantum, its priority number will be increased by 1 and move to the end of corresponding queue.

5) Assume that all processes arrive at time 0.

**Example:**

=====

Enter the quantum for Q1, Q2, Q3, and Q4: 6 4 8 7

Enter the number of processes to schedule: 3

Enter the burst time and priority of P1: 24, 1

Enter the burst time and priority of P2: 10, 5

Enter the burst time and priority of P3: 17, 3

The scheduling sequence is: P1, P1, P3, P1, P3, P1, P2, P3

=====

**Submission**

1. Given a startup program, merge main functions of task1 and task2 into one, so that users can test both task1 and task2 in one program.

**Example:**

=====

Enter a task number (1 or 2) or 0 to exit: 3

Please enter 0, 1 or 2.

Enter a task number (1 or 2) or 0 to exit: 2

Enter the quantum for Q1, Q2, Q3, and Q4: 6 4 8 7

Enter the number of processes to schedule: 3

Enter the burst time and priority of P1: 24,1

Enter the burst time and priority of P2: 10,5

Enter the burst time and priority of P3: 17,3

The scheduling sequence is: P1, P1, P3, P1, P3, P1, P2, P3

Enter a task number (1 or 2) or 0 to exit: 1

Enter the number of processes to schedule: 2

Enter the burst time of P1: 3

Enter the burst time of P2: 10

The scheduling sequence is: P1, P2, P2

Enter a task number (1 or 2) or 0 to exit: 1

Enter the number of processes to schedule: 3

Enter the burst time of P1: 17

Enter the burst time of P2: 15

Enter the burst time of P3: 7

The scheduling sequence is: P1, P2, P3, P1, P2, P3, P1, P2

Enter a task number (1 or 2) or 0 to exit: 0

Press any key to continue...

=====

2. Submit **only** .c and .h files into iSpace before the deadline.