

CS5541 - Computer Systems
Assignment Name: Scheduling
Spring 2018
Due: Monday, April 23, 2018

Jason Eric Johnson

03/27/2018

1 Introduction

Your task will be to implement a scheduling simulator. Given an input file with a list of process descriptions, your program will output the timing of the processes under various scheduling policies. These policies are as follows:

- Uniprocessor Policies
 - First-Come-First-Served (FCFS)
 - Round Robin (RR)
 - Shortest Process Next (SPN)
 - Shortest Remaining Time (SRT)
 - Highest Response Ratio Next (HRRN)
- Real-Time Aperiodic Policies
 - Earliest Deadline (ED)
 - Earliest Deadline with Unforced Idle times (EDUI)
 - First-Come-First-Served (FCFS)
- Real-Time Periodic Policies
 - Fixed-Priority (FP)
 - Earliest Deadline using Completion Deadlines (EDCD)

2 Handout Instructions

There is no handout for this assignment. You will start from a "blank page" for this assignment, so all of your code should be your own. You can look around for ideas and to understand concepts by finding implementations on the internet or in books, but your final code should be your own. **NO COPYING AND PASTING!** Please include references to any code, websites, and books you use to get ideas and increase your understanding.

3 Assignment

Your simulator will read an input text file and output the timing of the processes under the various policies listed in the introduction. Your program should use the specified input file to determine what kind of scheduling should be performed and the characteristics of the processes to be scheduled. The format of the input files and the expected output appears below. The examples below come directly from chapters 9 and 10 of your Operating Systems textbook.

3.1 Uniprocessor Scheduling

The input file for Uniprocessor Scheduling should look like this:

```
U, 5, 4
A, 0, 3
B, 2, 6
C, 4, 4
D, 6, 5
E, 8, 2
```

The first line specifies the type of scheduling (U for uniprocessor), the number of processes to be scheduled (5), and the quantum for Round Robin scheduling (4).

The remaining lines specify the characteristics of one process per line in the following format:

```
Process, Arrival Time, Service Time
```

The output for the input file shown above should look like this:

```
FCFS:
A: 0->3
B: 3->9
C: 9->13
D: 13->18
E: 18->20
```

RR:

A:0->3

B:3->7

C:7->11

D:11->15

B:15->17

E:17->19

D:19->20

SPN:

A:0->3

B:3->9

E:9->11

C:11->15

D:15->20

SRT:

A:0->3

B:3->4

C:4->8

E:8->10

B:10->15

D:15->20

HRRN:

A:0->3

B:3->9

C:9->13

E:13->15

D:15->20

For each of the policies listed in the Introduction output the abbreviation, then a line for each change in actively running process. The first line from the FCFS section above:

A:0->3

indicates that process A runs from time 0 to time 3.

3.2 Real-Time Aperiodic

The input file for Real-Time Aperiodic Scheduling should look like this:

```
RA, 5
A, 10, 20, 110
B, 20, 20, 20
C, 40, 20, 50
D, 50, 20, 90
E, 60, 20, 70
```

The first line specifies the type of scheduling (RA for real-time aperiodic scheduling), and the number of processes to be scheduled (5).

The remaining lines specify the characteristics of one process per line in the following format:

```
Process,Arrival Time,Execution Time,Starting Deadline
```

The output for the input file shown above should look like this:

```
ED:
A:10->30
B:Missed
C:40->60
E:60->80
D:80->100

EDUI:
B:20->40
C:40->60
E:60->80
D:80->100
A:100->120

FCFS:
A:10->30
B:Missed
C:40->60
D:60->80
E:Missed
```

For each of the policies listed in the Introduction output the abbreviation, then a line for each change in actively running process. The first line from the ED section above:

```
A:10->30
```

indicates that process A runs from time 10 to time 30.

Note - You should indicate any processes that miss their deadlines as shown in the output above.

3.3 Real-Time Periodic

The input file for Real-Time Periodic Scheduling should look like this:

```
RP, 2, 100
A, 0, 10, 20
B, 0, 25, 50
```

The first line specifies the type of scheduling (RP for real-time periodic scheduling), the number of recurring tasks to be scheduled (2), and the length of the simulation in time units (100).

The remaining lines specify the characteristics of one recurring task per line in the following format:

```
Task, Arrival Time, Execution Time, Ending Deadline
```

The output for the input file shown above should look like this:

```
FP:
A(1):0->10
B(1):10->20
A(2):20->30
B(1):30->40
A(3):40->50
B(1):Missed
B(2):50->60
A(4):60->70
B(2):70->80
A(5):80->90
B(2):90->95

EDCD:
A(1):0->10
B(1):10->20
A(2):20->30
B(1):30->45
A(3):45->55
B(2):55->60
A(4):60->70
B(2):70->90
A(5):90->100
```

For each of the policies listed in the Introduction output the abbreviation, then a line for each change in actively running task. The first line from the FP section above:

```
A (1) :0->10
```

indicates that first instance of process A runs from time 0 to time 10.

Note - You should indicate any processes that miss their deadlines as shown in the output above.

Note that the tasks should be listed in order of precedence for fixed priority scheduling. For the example above task A would have priority, followed by task B. If other tasks were specified, they would have precedence indicated by their order in the input file. So, for the following input file:

```
RP, 4, 100  
C, 0, 20, 30  
A, 0, 10, 20  
B, 0, 25, 50  
X, 0, 25, 75
```

the order of precedence for fixed priority scheduling would be C, A, B, X.

4 Programming Rules

This is an individual assignment. Work on your own and do not discuss code with your classmates. As always, you may discuss concepts with me and your classmates during class time, but do not share code.

You may use any language you wish to implement this assignment.

Remember that all of your code should be your own. You can look around for ideas and to understand concepts by finding implementations on the internet or in books, but your final code should be your own. **NO COPYING AND PASTING!** Please include references to any code you use to get ideas and/or increase your understanding.

5 Evaluation

This assignment will be worth 100 points.

6 Handin Instructions

Submit Scheduling using Elearning. Follow all the instructions on the "Homework Guidelines" page carefully.

You will submit your assignment as a single `.zip` file, named as specified in the "Homework Guidelines". It should include your code and `README.txt` file which will contain authorship information and any references.

Please name your `.zip` file in the following way:

`CS5541-[YourName]-Scheduling.zip`