Operating Systems (Fall 2015)

Instructor: Dr. Hanh Pham

Project Requirements:

Write a program in JAVA to implement a CPU scheduling algorithm (see the assignment table below). This program should read data about the processes from "input.data"(see the format below) and define the schedule data for a Gant chart in "output.data"(see the format below)

Format of "input.data":

```
10 (number of processes)
0 (if not RR nor preemptive, 1 for opposite) 2 (=quantum)
3 (arriving time for P1) 12(burst time) 2(priority)
…
0 (arriving time for P10) 7(burst time) 5(priority)
```

Format of "output.data":

```
0(start) 7(end) 10 (process #)
7(start) 21(end) 1(process #)
…
102(start) 115(end) 8(process #)
```

Format of the report:

+Title page: student name, assigned scheduling method
+Table of contents

1. Description of Scheduling algorithm (also give an example)
2. Description of Implementation (your program)
3. Experiments (run your program with 3 different input data sets, provide results, draw Gantt chat based on that, calculate average waiting time, and average response time for each set) => see MORE on 3rd page !!!
4. Conclusion
5. References

What to submit:

1) Report file in word format (named as "ID-LastName-FirstInitial.doc")
2) RAR folder which contains (named as "ID-LastName-FirstInitial.rar"):
   - Input.data
   - Output.data
   - Scheduler.class (executable code)
   - Scheduler.java (source code)

Where to submit:

http://cs.newpaltz.edu/~phamh/aos/sub/

Deadline: November 4th, 2015 (hard deadline, late work will not be graded !!!)

Assignments:

Please check at Blackboard for the assignment of your scheduling algorithm which could be one of the following:

Code	Scheduling Algorithms
1	SJF: Shortest Job First
2	PH: Priority High (higher number means higher priority)
3	PL: Priority Low (lower number means higher priority)
4	RR: Round Robin
5	pSJF: Preemptive Shortest Job First
6	pPH: Preemptive Priority High (higher number means higher priority)
7	pPL: Preemptive Priority Low (lower number means higher priority)

EXAMPLES:

In the REPORT:

…

3. EXPERIMENTS:

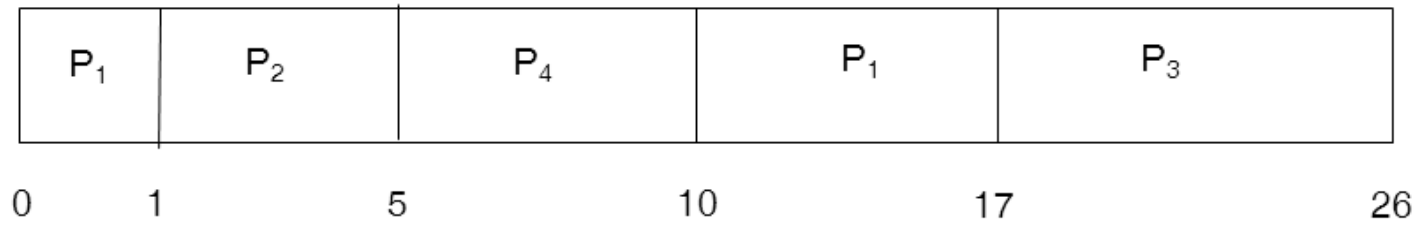
Experiment #1:

a) INPUT DATA:

```
“input.data"
4
1 1
0 8 5
1 4 1
2 9 4
3 5 3
```

Process	Arrival Time	Burst Time	Priority
P_1	0	8	5
P_2	1	4	1
P_3	2	9	4
P_4	3	5	3

b) Gantt Chart (Your Assigned Schedule Algorithm: example *Preemptive* SJF)



c) Average waiting time

= [P1(10-1) + P2(1-1) + P3(17-2)+ P4(5-3)]/4 = 26/4 = **6.5**

d) SHOULD-BE Output:

"output.data"

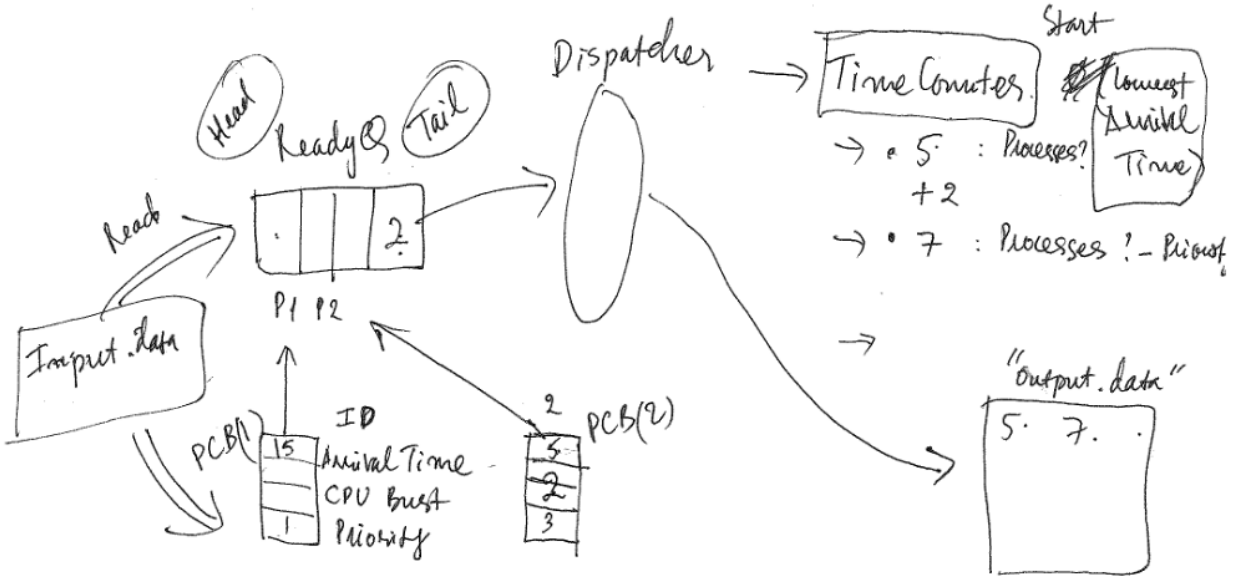
```
0 1 1
1 5 2
5 10 4
10 17 1
17 26 3
```

e) MY Output (produced by program):

My "output.data"

```
0 1 1
1 5 2
5 10 4
10 17 1
17 26 3
```

An example of how the scheduler can be built:



(1) Read "Input.data" & Fill-in PCBs
(2) Build the ReadyQueue
(3) Has Dispatcher look/pick a process/element to assign to CPU
(write to "output.data")

An example of how the scheduler would work: