

**EECS 2011 Section E Fall 2017  
Assignment 2**

**System for Deterministic Modeling of CPU Scheduling Algorithms**

**Due Date: Tuesday November 7, 2017, 9:30 a.m.**

**1. Description of the Assignment**

**1.1 Required Readings**

You are required to read Section 6.1, Basic Concepts, Section 6.2, Scheduling Criteria, Section 6.3, Scheduling Algorithms (pages 261 to 277), Section 6.8.1, Deterministic Modeling, (pages 300-301), of Chapter 6, CPU Scheduling, in the book authored by Silberschatz, Galvin, and Gagne, "Operating System Concepts," Ninth Edition, Wiley, 2013. (A copy of Sections 6.1, 6.2, 6.3, and 6.8.1, of Chapter 6 of Silberschatz, Galvin, and Gagne's book above can be found in the course directory:  
`/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf`)

**1.2. System for Deterministic Modeling of CPU Scheduling Algorithms**

You are required to apply the concepts of data structures and algorithms that you have learned in this course to design, analyze, implement, test, and document a System for Deterministic Modeling of CPU Scheduling Algorithms described below:

**A. Each process *P* in the system can have the following attributes:**

A1. **Arrival Time:** the time when process *P* first makes a request to the CPU Scheduler to execute on the CPU. Each Arrival Time can be expressed as a nonnegative integer.

A2. **Sequence of CPU Computation Time and I/O Time Requirements** which can each be expressed as a nonnegative integer (see example below).

A3. **Process Priority**, expressed as a nonnegative number.  
(Process Priority Numbers will only be used when Priority Scheduling is applied. Lower Process Priority Numbers correspond to higher process priorities.)

For example, for some process *P*:

Arrival\_Time(*P*) = 8;

The Sequence of CPU Computation Time and I/O Time Requirements of process *P* could have 5 components as follows:

- (1) CPU\_time(*P*) = 10; (P needs to execute 10 time units on the CPU at first)
- (2) I/O\_time(*P*, 0) = 15; (then P needs to do I/O for 15 time units on I/O device 0)
- (3) CPU\_time(*P*) = 16; (then P needs to execute 16 time units on the CPU)

- (4)  $I/O\_time(P, 1) = 20$ ; (then P needs to do I/O for 20 time units on I/O device 1)  
(5)  $CPU\_time(P) = 8$ . (then P needs to execute 8 time units on the CPU before terminating)

$Process\_Priority(P) = 2$ .

Please note that the I/O time requirements only includes the time that is needed when a process P is actually doing I/O; it DOES NOT include the time that it may need to spend waiting in an I/O queue before the I/O device becomes available.

**B. At any point in time, each process P in the system can be in any one of the following states,** according to Section 6.1, Basic Concepts, Section 6.2, Scheduling Criteria, Section 6.3, Scheduling Algorithms (pages 261 to 277) of Chapter 6, CPU Scheduling, in the book authored by Silberschatz, Galvin, and Gagne, “Operating System Concepts,” Ninth Edition, Wiley, 2013:

- (a) **New:** P has not arrived yet;
- (b) **Running:** P has been selected by the CPU scheduler for execution on the CPU, and is currently executing on the CPU;
- (c) **Ready:** P has already arrived, and P is currently waiting in the ready queue to be scheduled by the CPU Scheduler for execution on the CPU;
- (d) **Waiting for I/O:** P had previously made an I/O request to the I/O scheduler in order to perform execution on the CPU, and P is either currently performing I/O on an I/O device or is currently waiting in an I/O queue so that it can be scheduled by the I/O Scheduler to perform I/O on an I/O device;
- (e) **Terminated:** P has completed its sequence of CPU computation time and I/O time requirements.

**C. The System for Deterministic Modeling of CPU Scheduling Algorithms must provide the ability to use any one of the scheduling algorithms described in Section 6.3, Scheduling Algorithms of the above mentioned book, to schedule a set of processes as described in Section 6.1, Basic Concepts of the above mentioned book on a single CPU.**

In particular, the System for Deterministic Modeling of CPU Scheduling Algorithms allows any one of the following scheduling algorithms described in Section 6.3, Scheduling Algorithms, of the above mentioned book to be used:

**C1. Nonpreemptive First-Come, First-Served (FCFS) Scheduling**

Nonpreemptive First-Come, First Served (FCFS) scheduling is explained in detail in Section 6.3.1. First-Come, First-Served Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

**C2. Nonpreemptive Shortest-Job-First (SJF) Scheduling**

Nonpreemptive Shortest-Job-First (SJF) scheduling is explained in detail in Section 6.3.2. Shortest-Job-First Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

### **C3. *Preemptive SJF (Shortest-Remaining-Time-First) Scheduling***

Preemptive SJF (Shortest-Remaining-Time-First) scheduling is also explained in detail in Section 6.3.2. Shortest-Job-First Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

### **C4. *Nonpreemptive Priority Scheduling***

Nonpreemptive Priority scheduling is explained in detail in Section 6.3.3. Priority Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

### **C5. *Preemptive Priority Scheduling***

Preemptive Priority scheduling is also explained in detail in Section 6.3.3. Priority Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

### **C6. *Preemptive Round-Robin (RR) Scheduling***

Preemptive Round-Robin (RR) scheduling is explained in detail in Section 6.3.4. Priority Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

### **C7. *Multilevel Queue Scheduling***

Multilevel Queue scheduling is also explained in detail in Section 6.3.5. Multilevel Queue Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

For Multilevel Queue scheduling, you may assume that there exist a total of two (2) queues:

- C7.1. One foreground queue which is scheduled by RR scheduling;
- C7.2. One background queue which is scheduled by FCFS scheduling;
- C7.3. The foreground queue has absolute priority over the background queue.

### **C8. *Multilevel Feedback Queue Scheduling***

Multilevel Feedback Queue scheduling is explained in detail in Section 6.3.6. Multilevel Feedback Queue Scheduling, of the above mentioned book, in [/cs/course/2011E/Operating\\_System\\_Concepts\\_CPU\\_Scheduling.pdf](/cs/course/2011E/Operating_System_Concepts_CPU_Scheduling.pdf).

For Multilevel Feedback Queue scheduling, you may assume that there exist a total of three (3) queues, with an organization similar to that illustrated in Figure 6.7 on page 276 of the above mentioned book in

/cs/course/2011E/Operating\_System\_Concepts\_CPU\_Scheduling.pdf:

C8.1. Queue 0 which is scheduled by RR scheduling with a first given time quantum;

C8.2. Queue 1 which is scheduled by RR scheduling with a second given time quantum;

C8.3. Queue 2 which is scheduled by FCFS scheduling.

C8.4. Processes are scheduled in a manner similar to that illustrated in Figure 6.7 and further explained on pages 275-276 of the above mentioned book.

**D. The following can be assumed regarding I/O devices and the I/O Scheduler:**

D1. There can exist any number of I/O devices in the system;

D2. There is only one I/O queue for each I/O device;

D3. I/O queues are scheduled by Nonpreemptive FCFS scheduling.

**E. For any given set of processes for which the Arrival Time, Sequence of CPU Computation Time and I/O Time Requirements, and Process Priority are given for each process, the system will be able compute the following for any of the CPU scheduling algorithms described above.**

(you may assume that the context switch time is 0) according to the definitions provided in Section 6.2, Scheduling Criteria, p. 265-266 of the above mentioned book in

/cs/course/2011E/Operating\_System\_Concepts\_CPU\_Scheduling.pdf:

E1. **Average Waiting Time;**

E2. **Average Turnaround Time.**

**1.3. Requirements regarding the design, analysis, implementation, testing, and documentation of the software in the System for Deterministic Modeling of CPU Scheduling Algorithms**

(a) When designing the software to implement the System for Deterministic Modeling of CPU Scheduling Algorithms, you must apply best practice software engineering principles and carefully choose appropriate data structures and methods. Furthermore, in your report/documentation you must justify and explain why you chose each particular data structure and method.

(b) You must analyze the asymptotic run times and space usage of your methods in the report.

(c) You must describe in detail any problems or difficulties that you had encountered, and how you solved or were able to overcome those problems or difficulties in the report.

(d) Additional Requirements:

(d1) You must make sure that your code has very detailed comments.

(d2) You must make sure that your code compiles correctly.

(d3) You must make sure that your code does not generate non-recoverable exceptions.

(d4) You must make sure that your code is able to handle incorrect input.

Failure to satisfy all the requirements above will result in a low mark for the assignment.

## **2. Platform on Which The System for Deterministic Modeling of CPU Scheduling Algorithms is to be Implemented**

The programs should to be implemented using the Java programming language and you should make sure that the TAs/markers will be able to run them using the system at York.

## **3. What to Hand In**

Each group is required to hand in both a hard copy and an electronic copy of the following:

1. A written report that identifies and addresses all the important aspects and issues in the design, analysis, implementation, testing, and documentation of the software for the problem described above.
2. The Java source programs.
3. A "Test\_output" file containing the output of any testing your group has done.
4. A "README" file explaining how to compile and run your group's program.
5. In addition to submitting a hard copy, each group is required to use the utility "submit" to submit the electronic version of the above 4 files to the course directory  
/cs/course/2011E/submit/a2

**Important Warning:**

**Only submitting an electronic copy of your assignment is not enough! If you fail to submit a hard copy of your assignment on or before the due date, your assignment will receive a grade of 'F'.**

#### **4. Evaluation of the Assignment**

4.1. The report part of your assignment (70%) will be evaluated according to:

- (a) How well you have satisfied the requirements specified in Section 1.3 above.
- (b) How well you have explained the design and implementation of your system and how well you have justified your design decisions.
- (c) The quality of your design.
- (d) How well you have designed and explained the testing.
- (e) The clarity, and readability of the report.

4.2. The program and testing part of your assignment (30%) will be evaluated according to:

- (a) The quality of the design and implementation of your programs.
- (b) The quality of the testing of your programs.
- (c) Whether your programs satisfy the Additional Requirements in section 1.3 (d), (d1)-(d4) above.

#### **5. Notes**

Please note that the requirements specified in Section 1. Description of the Assignment above, are the *minimum requirements* that must be satisfied by your assignment. Obviously, there are many other possible details of the System for Deterministic Modeling CPU Scheduling Algorithms that have been left unspecified. It is your responsibility to make appropriate design, analysis, implementation, testing, and documentation choices concerning the unspecified details of the System for Deterministic Modeling of CPU Scheduling Algorithms, and justify those decisions in your report.

#### **6. Resources**

/cs/course/2011E/Operating\_System\_Concepts\_CPU\_Scheduling.pdf

