

Fundamentos de Sistemas de Operação

MIEI 2017/2018

Homework Assignment 3

Deadline and Delivery

This assignment is to be performed **individually** by each student – any detected frauds will cause failing the discipline. The code has to be submitted for evaluation via the Mooshak system (<http://mooshak.di.fct.unl.pt/~mooshak/>) using each student's individual account -- the deadline is **23h59, November 26th, 2017 (Sunday)**.

Description

The goal of this assignment is to implement a process's scheduling algorithm in an OS simulator. The algorithm is similar to the MLFQ presented in the classes and in your reference book (OSTEP) [1]. The simulator is written in Java and includes the simulation of several hardware and OS concepts in a very simplified way. In this assignment, you will only have contact with a small part of the simulator, namely concerning process scheduling.

Scheduling algorithm

The scheduler to be implemented is based on the MLFQ presented in your book [1], chap. 8, pag. 10:

1. Every process starts with the maximum priority;
2. The scheduler dispatches for execution one of the processes with highest priority, using Round-Robin among the ones with the same priority and its execution is limited by a *time-slice* (uses preemption);
3. After running an accumulated *time-quota* of CPU time, the process' priority drops one level;
4. At every *time period*, all processes have their priority upgraded by two levels;
5. When the IO of a process finishes, that process is added to the end of its queue.

Your scheduler should define 4 priority levels, all using *time-slices* of 10 clock ticks, *time-quotas* of 20 clock ticks, and *time-periods* of 80 clock ticks for the upgrades (ascending two priority levels as described). The calculation of all those times can only be made when the scheduler is called, namely:

- When a process is blocked or preempted, you should decrement the used time of that process' quota; if the quota becomes zero or bellow, then the process gets its priority lowered one level and a new quota;
- It is necessary to remember the clock time of the last priority upgrade (`lastUpgradeTime`). At the end of each scheduler method (see next section), the clock time has to be inspected to see if a new *time-period* has passed to perform the priority upgrade to all processes in case that happens. All processes have their quota restored, their priority and queue updated. The `lastUpgradeTime` is also updated.

The Simulator

Your work will be confined to the implementation of class `caoss.simulator.os.scheduling.FSOScheduler`, of which a skeleton is already given. This class implements the `caoss.simulator.os.Scheduler` interface that comprises the following 5 methods (that trigger the several processes' state changes):

- `newProcess(Program prog)` – This method is called by the simulator upon a program's execution request and must create a process to execute the program `prog`.
- `ioRequest(ProcessControlBlock<SchedulingState> pcb)` – It is invoked when the process (with control block `pcb`) that is running in the CPU requests an input/output operation.
- `ioConcluded(ProcessControlBlock<SchedulingState> pcb)` – This one is called when the input/output operation requested by the process (with control block `pcb`) ends.
- `quantumExpired(ProcessControlBlock<SchedulingState> pcb)` – This method is invoked when the process (with control block `pcb`) running in the CPU exhausts its quantum (*time-slice*).
- `processConcluded(ProcessControlBlock<SchedulingState> pcb)` – It is called when the process (with control block `pcb`) has concluded its execution.

To accomplish your assignment, you must also study other classes of the simulator, namely:

- `caoss.simulator.Program` – Describes the program executed by a process.

- `caoss.simulator.os.ProcessControlBlock` – Defines the information required by the system to manage a process' execution. For instance, the *pid* and the process' time of arrival to the system.
- `caoss.simulator.os.scheduling.SchedulingState` – The scheduling information that must be kept for each process in the system is represented in this class. In the context of MLFQ, examples of the information you may use are the *time-quota* given to the process, its priority, and the last time when the process was scheduled.
- `caoss.simulator.os.Dispatcher` – Loads the execution context of a process in the target CPU. If there is no process to dispatch, your code will **mandatorily** have to dispatch the idle process, i.e. `null`:

`Dispatcher.dispatch(some_pcb)` or `Dispatcher.dispatch(null)`

- `caoss.simulator.hardware.Clock` – Defines the computer's clock. The method to obtain the current time is:

`Hardware.clock.getTime()`
- `caoss.simulator.hardware.Timer` – Implements the computer's timer. It may be programmed to notify the scheduler that the quantum (*time-slice*) assigned to a process has expired.
 - To obtain the simulator's timer use the following line of code:

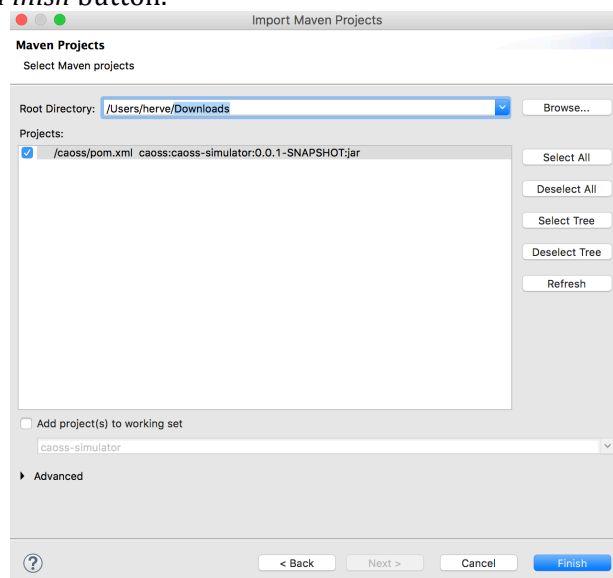
`Timer timer = (Timer) Hardware.devices.get(DeviceId.TIMER);`
 - To program the timer, use the `set()` method. For instance, `timer.set(10)` programs the timer to interrupt the CPU and gives control to the Operating System, in 10 time units.
- `caoss.simulator.os.Logger` – This is the Operating System's logger. You must use it to output the result of your scheduling algorithm. Concretely, you must use the static method `Logger.info()`. You may change the detail level of the logger by altering line 34 of class `caoss.simulator.os.Logger` to `LOGGER.setLevel(Level.ALL)`. This modification will not have impact on your submission to Mooshak.
- The `caoss.simulator.os.OperatingSystem.OS` static variable holds a reference to the Operating System object that features several useful methods, such as the `getRunningProcess()` to obtain the process that is currently running in the CPU.

To simplify your understanding of the simulator, you may find the implementation of a Round Robin scheduler in class `caoss.simulator.os.scheduling.RoundRobinScheduler`. You can test this scheduler by copying its code to the `caoss.simulator.os.scheduling.FS0Scheduler` class or by altering the field `scheduler` of class `caoss.simulator.os.FS0OS`.

You also have *javadoc* information at your disposal in folder *doc*.

Development environment, compilation and execution

The simulator's source code (available from CLIP) is a Maven managed project. You may import it to Eclipse or some other IDE. For instance, in Eclipse, use *File* → *Import ...* → *Maven* → *Existing Maven Project*, select the *caoss* project, and click on the *Finish* button.



Subsequently, you will be able to execute the simulator by running class `caoss.simulator.CAOSS`.

If you prefer to compile and run the project from the command line, compile it using the `mvn` command:

```
cd caoss
mvn compile
```

To run the `caoss.simulator.CAOSS` class type (in directory `caoss`):

```
java -cp target/classes caoss.simulator.CAOSS
```

The simulator presents a command line from where you may simulate a program execution:

```
exec examples/ex1.caoss
```

or multiple programs at once:

```
exec examples/ex1.caoss examples/ex2.caoss examples/ex3.caoss
```

To terminate the simulator's execution type:

```
shutdown
```

Submission to Mooshak

You only have to submit your implementation of class `caoss.simulator.os.scheduling.FSOScheduler` (`FSOScheduler.java`).

Bibliography

[1] Chapter 8 of the recommended book, "Operating Systems: Three Easy Pieces Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau"

Example of a Schedule

See below the example of a schedule using the MLFQ algorithm that you will have to implement.

Command: `exec examples/ex1.caoss examples/ex1.caoss examples/ex4.caoss`

Result:

```
Create process 0 to run program examples/ex1.caoss
Run process 0 (quantum=10, quota=20)
Upgrade priorities
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Create process 1 to run program examples/ex1.caoss
Queue 0: [1]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Create process 2 to run program examples/ex4.caoss
Queue 0: [1, 2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 1 (quantum=10, quota=20)
Queue 0: [2, 0]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=20)
Queue 0: [0, 1]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
```

```

Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: [1]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked []
Process 1: quantum expired
Process 1: quota expired
Run process 2 (quantum=10, quota=17)
Queue 0: []
Queue 1: [0, 1]
Queue 2: []
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: [1]
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=20)
Queue 0: []
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=16)
Queue 0: []
Queue 1: [0, 1]
Queue 2: []
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: [1]
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 1: quantum expired
Process 1: quota expired
Run process 2 (quantum=10, quota=13)
Queue 0: []
Queue 1: []
Queue 2: [0, 1]

```

```

Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: [1]
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=12)
Queue 0: []
Queue 1: []
Queue 2: [0, 1]
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: [1]
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 1: quantum expired
Process 1: quota expired
Run process 2 (quantum=10, quota=9)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0, 1]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [1]
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked [2]
Process 2: IO concluded
Upgrade priorities
Queue 0: [2]
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=20)
Queue 0: []
Queue 1: [0, 1]

```

```

Queue 2: []
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: [1]
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked []
Process 1: quantum expired
Process 1: quota expired
Run process 2 (quantum=10, quota=17)
Queue 0: []
Queue 1: [0]
Queue 2: [1]
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: [1]
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 1 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=16)
Queue 0: []
Queue 1: []
Queue 2: [0, 1]
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: [1]
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 1: quantum expired
Process 1: quota expired
Run process 2 (quantum=10, quota=13)
Queue 0: []

```

```

Queue 1: []
Queue 2: [0]
Queue 3: [1]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [1]
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 1 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 1: quantum expired
Run process 2 (quantum=10, quota=12)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0, 1]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [1]
Blocked [2]
Process 0: quantum expired
Run process 1 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked [2]
Process 1: execution concluded
Process 1: turnaround time: 265
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Process 0: quota expired
Run process 2 (quantum=10, quota=9)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded

```

```

Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Process 0: quota expired
Run process 2 (quantum=10, quota=8)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 0 (quantum=10, quota=10)
Upgrade priorities
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=20)
Queue 0: []
Queue 1: [0]
Queue 2: []
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=19)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []

```



```

Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=16)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=15)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=12)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []

```

```

Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=11)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 2 (quantum=10, quota=8)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: [0]
Blocked []
Process 2: IO request
Run process 0 (quantum=10, quota=10)
Upgrade priorities
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 0: quantum expired
Run process 0 (quantum=10, quota=10)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked [2]
Process 2: IO concluded
Queue 0: [2]
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Process 0: quota expired
Run process 2 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: [0]
Queue 3: []
Blocked []
Process 2: execution concluded
Process 2: turnaround time: 483
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Run process 0 (quantum=10, quota=10)
Queue 0: []

```

```
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: quantum expired
Process 0: quota expired
Run process 0 (quantum=10, quota=20)
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
Process 0: execution concluded
Process 0: turnaround time: 504
Queue 0: []
Queue 1: []
Queue 2: []
Queue 3: []
Blocked []
```