CS 350: Homework #7

Due 3PM, Tuesday, March 29

Kristel Tan (ktan@bu.edu)

LATE SUBMISSION

Question 3

Part d

Please find the source code for this problem in the Java file named 'Q3D.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which five processes are interleaved and served in a FIFO fashion:

```
P0 is requesting CS
P0 is entering CS on iteration 1
P4 is requesting CS
P3 is requesting CS
P2 is requesting CS
P1 is requesting CS
P0 is exiting the CS
P0 is requesting CS
P4 is entering CS on iteration 1
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 1
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 1
P2 is exiting the CS
P2 is requesting CS
P1 is entering CS on iteration 1
P1 is exiting the CS
P1 is requesting CS
P0 is entering CS on iteration 2
P0 is exiting the CS
P0 is requesting CS
P4 is entering CS on iteration 2
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 2
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 2
P2 is exiting the CS
P2 is requesting CS
P1 is entering CS on iteration 2
P1 is exiting the CS
P1 is requesting CS
P0 is entering CS on iteration 3
P0 is exiting the CS
P0 is requesting CS
P4 is entering CS on iteration 3
```

```
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 3
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 3
P2 is exiting the CS
P2 is requesting CS
P1 is entering CS on iteration 3
P1 is exiting the CS
P1 is requesting CS
P0 is entering CS on iteration 4
P0 is exiting the CS
P0 is requesting CS
P4 is entering CS on iteration 4
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 4
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 4
P2 is exiting the CS
P2 is requesting CS
P1 is entering CS on iteration 4
P1 is exiting the CS

. . .
```

This pattern is then printed out in like fashion for the remaining iterations and for each thread. As you can see, the processes are served based on when they arrive. I implemented a queue in order to keep track of this. The process at the head of the queue is always served when `newSignal()` is called.

Question 4

Please find the source code for this problem in the Java file named 'Q4.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which there is a fixed upper bound of "N" on the out-of-order use of the priority semaphore:

```
P1 is requesting CS
P4 is requesting CS
P3 is requesting CS
P2 is requesting CS
P0 is requesting CS
P1 is entering CS on iteration 1
P1 is exiting the CS
P1 is requesting CS
P4 is entering CS on iteration 1
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 1
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 1
P2 is exiting the CS
P2 is requesting CS
```

```
P0 is entering CS on iteration 1
P0 is exiting the CS
P1 is entering CS on iteration 2
P0 is requesting CS
P1 is exiting the CS
P1 is requesting CS
P4 is entering CS on iteration 2
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 2
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 2
P2 is exiting the CS
P2 is requesting CS
P0 is entering CS on iteration 2
P0 is exiting the CS
P1 is entering CS on iteration 3
P0 is requesting CS
P1 is exiting the CS
P4 is entering CS on iteration 3
P1 is requesting CS
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 3
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 3
P2 is exiting the CS
P2 is requesting CS
P0 is entering CS on iteration 3
P0 is exiting the CS
P1 is entering CS on iteration 4
P0 is requesting CS
P1 is exiting the CS
P1 is requesting CS
P4 is entering CS on iteration 4
P4 is exiting the CS
P4 is requesting CS
P3 is entering CS on iteration 4
P3 is exiting the CS
P3 is requesting CS
P2 is entering CS on iteration 4
P2 is exiting the CS
P2 is requesting CS
P0 is entering CS on iteration 4
P0 is exiting the CS

. . .
```

To view the entire output, you can run the program and view the console output. With the exception of the beginning, you can see that no more than N=3 processes enter the critical section before 0. An example of this is the highlighted section of the output above. In the code, you will also see that I implemented a variable called `bound` in order to keep track of how many processes have entered the critical section before P0 must use it again.