

SYSC 3303 Summer 2014 Quiz #2

Name:

Student Number:

| Q1(/10) | Q2(/15) | Q3(/15) | Total(/40) |
|---------|---------|---------|------------|
| | | | |

Question 1 [10 marks]

Imagine that we have three Java threads in a concurrent system being run on a uni-processor machine. Also, assume that context switching between threads and all other overhead managed by the scheduler take no time, and that the priority inheritance protocol is used. Here is the information on our three threads:

- t1: priority 4, needs to run for a total of 5ms; needs o's lock for 2ms: from time 2 to 4 of its 5ms; ready to run at 0ms (when we start)
- t2: priority 6, needs to run for a total of 3ms; doesn't need any locks; ready to run at 3ms
- t3: priority 8, needs to run for a total of 2ms; needs o's lock for the whole time; ready to run at 5ms

You are to break down the 10ms execution of the above threads into 1ms intervals. For each 1ms interval you are to specify:

- the thread (t1, t2, or t3) that is currently running (i.e. has the processor)
- its priority
- the lock(s) the thread is holding, if any

Intervals (in ms):

0-1: t1, 4, no locks

1-2: t1, 4, no locks

2-3: t1, 4, o's lock

3-4: t2, 6, no locks

4-5: t2, 6, no locks

5-6: t1, 8, o's lock [t3 wants the lock, so t1 has its
priority increased]

6-7: t3, 8, o's lock

7-8: t3, 8, o's lock

8-9: t2, 6, no locks

9-10: t1, 4, no locks [note that t1's priority is back to
usual now]


```

// constructor sets the lockPrio after checking it (5 marks total)
public BoxPC( int prio ) { [1 mark]
    if (prio<Thread.MIN_PRIORITY||prio>Thread.MAX_PRIORITY)
        // or 1 and 10 in place of constants
        prio=Thread.MAX_PRIORITY - 1; // if invalid pick a priority
        // (or throw exception) [2 marks]
    }

    lockPrio = prio; [2 marks]
}

// when the box is empty, puts the given object into the box
// ensures that the priority is equal to the lock priority
// throughout the critical section (8 marks total)
public void put( Object obj ) {
    int origPrio = Thread.currentThread().getPriority();
    // save the original priority [3 marks]
    // (deduct 1.5 marks if this is a field, as that
    // will not work)
    if (origPrio<lockPrio) { // don't want to lower it and no need
        // and no need to change if it's equal
        Thread.currentThread().setPriority(lockPrio);
        // change priority before critical section [2 marks]
        // no deduction if no "if" (as question states you can
        // assume that the user follows the rules)
    }
    synchronized (this) { // [1 mark]
        while (!empty) {
            try {
                wait();
            } catch (InterruptedException e) {return;}
        }
        contents = obj;
        empty = false;
        notifyAll();
    } // end critical section
    if (origPrio<lockPrio) {
        Thread.currentThread().setPriority(origPrio);
        // reset priority after critical section [2 marks]
        // ("if" optional)
    }
}
}
}

```

Note: Question 3 is on the back of this page.

Question 3 [15 marks]

File "quiz2.txt" containing 2025 bytes is being transferred from the client to the server.

- The first time ACK block #1 is sent the packet is lost.
- The first time DATA block #2 is sent, the packet is duplicated and both packets arrive one after the other.
- The disk becomes full when writing DATA block #3.

Draw a timing diagram showing what packets are sent and/or received by the client and/or server starting with the request packet and ending with the last packet in the transfer. Include information on threads and/or ports created and destroyed during the transfer. For each packet, give the packet type and opcode, source and destination TID, block number (if applicable), error code (if applicable), and number of bytes of data (if applicable).

Deduct 1 mark for each incorrect / missing / extra TFTP packet and deduct 1 mark for each of any missing/incorrect TID (listener must use 69, others are student's choice), opcode, error code, block number, data bytes, create/destroy, etc.

