CS 350: First Midterm (Spring 01) - 3/22/2001

Answer 100 points worth of problems. If you answer more, I will throw away/scale down your worst answer. If necessary, feel free to make assumptions (clearly state them) if they do not trivialize the problem.

Problem 1: (16 pts; 15 minutes)
Briefly explain the following terms and their importance to Operating Systems:
- Medium term scheduler
- Condition locks
- Spooling
- DMA

Problem 2: (20 points; 15 minutes)
(a) (9 points) An operating system is replete with queues. Name three different queues that a process may belong to; under what situation/state would you expect the process to be there? (Name 3 more for a 3 point bonus)
(b) (11 points) Consider the following program fragments being executed by two threads; what is the range of final values that x can have.

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>for(i=0; i &lt; 10; i++)</td>
<td>for(j=0; j&lt; 10; j++)</td>
</tr>
<tr>
<td>x = x + 2;</td>
<td>x = x - 1;</td>
</tr>
</tbody>
</table>

Problem 3: (30 points; 20 minutes) Briefly explain any 4 of the following potentially wrong statements. Be careful with this problem.
1. Busy waiting is more expensive than blocking
2. It is beneficial to have “user-level” processes in the same way we have “user-level” threads
3. Preemptive scheduling would work well in the long term scheduler if we were interested in response time
4. Mesa and Hoare semantics in monitors make no difference to the programmer
5. A higher quantum size in round-robin increases the overhead and the response time

Problem 4: (25 pts; 15 minutes) You are designing an operating system for use in distributed applications. The primary requirement on your scheduling is that it is fair. To that end, a bright engineer in your group suggested that an enhanced version of Highest Response Ratio Next that she developed for best fairness. In her policy called (Maximum Turnaround Time Next), instead of running the process that has the highest response ratio (call it Rmax), you assume that you run that process, and find the response ratios of all the other processes when it finishes (call them R1, ... Rn). You run the process with max(Rmax, R1, ... Rn).

(a) (15 points) Define a reasonable metric for fairness. Explain why this policy is more fair than HRRN. Also show one example proving that.
(b) (10 points) Would a preemptive version work better? Explain.

Problem 5: (30 pts; 25 minutes) Two or more people are playing a game of frisbee with two or more frisbees. Initially, the frisbees are given to different people and the number of people and frisbees is known. The players stand around in a circle and the frisbees are thrown clockwise along the circle. Each person receives a frisbee and throws it to the next person over. Unfortunately, if a frisbee is thrown at a person when he already has another, he will drop it – we do not want that case to happen.

- (15 points) Implement a pseudocode simulation of this problem (you can use locks, semaphores or condition locks)
- (5 points) Is there a problem if the number of frisbees equals the number of players? Explain
- (10 points) Suppose that not all the throws are accurate, and sometimes, a frisbee is lost. We would like to continue throwing the other frisbees around. Describe how your implementation would change (don’t worry about how the frisbee is lost).