CMSC 421/Section 0101: Operating Systems  
Fall 2003  
Krishna Sivalingam  
14 October 2003  
75 Minutes

Name:__________________________________________________________  
Last 5 Digits of UMBC ID:________________________________________

“I certify that my answers on this test are entirely from my efforts.”

(Signature)  
(Printed Name)

This is a closed-book, closed-notes, closed-neighbor exam. Answer all questions in the space provided – you can write on both sides of the paper. If you use additional papers, please STAPLE them to this answerbook and write your name on each additional paper.

1. _____ (10 points)  
2. _____ (15 points)  
3. _____ (25 points)  
4. _____ (25 points)  
Total: _____ (75 points)
1. (10 points) **Basics:**

List the various Operating System functionalities that can classified under the category *Process Management* and *Memory Management*. 
2. (15 points) **Processes and Threads:**

Describe the sequence of actions, with the aid of a diagram if needed, that is carried by the kernel when the `fork()` and `pthread_create()` calls are invoked.
3. Scheduling:

(a) (6) Explain the differences in the degree to which the following scheduling algorithms discriminate in favor of short processes: (a) FCFS; (b) Round Robin, and (c) Multilevel feedback queues
(b) (19) Consider the following set of processes, with the length of the CPU-Burst time given in milliseconds:

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival Time</th>
<th>Burst Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0.0</td>
<td>8</td>
</tr>
<tr>
<td>P2</td>
<td>0.4</td>
<td>4</td>
</tr>
<tr>
<td>P3</td>
<td>1.0</td>
<td>3</td>
</tr>
<tr>
<td>P4</td>
<td>1.2</td>
<td>2</td>
</tr>
</tbody>
</table>

Determine the average turnaround time for each of the following algorithms:

(i) SJF with preemptive scheduling

(ii) Three level feedback queue scheduling with three queues as follows: Queue 1 has quantum of 2; Queue 2 has quantum of 4; and Queue 3 is FCFS. The scheduler first executes process in queue 1; only when queue 1 is empty does it schedule from queue 2.
4. The following algorithm has been proposed for an entry and exit protocol for critical sections. Assume there are two processes \( P_i \) and \( P_j \). The shared variables are \( \text{turn} \) initialized to \( i \), \( \text{flags} \) and a boolean array \( \text{wait} \) with one entry for each process. Assume initially all elements of this array are FALSE. The algorithm for \( P_i \) is:

ENTRY CODE:

\[
\text{flags}[i] = \text{BUSY};
\]

\[
\text{if} \ (\text{flags}[j] == \text{BUSY}) \ {\{}
\]

\[
\text{wait}[i] = \text{TRUE}; \ // \ I \ am \ waiting!
\]

\[
\text{while} \ ((\ \text{flags}[j] == \text{BUSY}) \ \&\& \ (\text{turn} == j) \ || \ \text{wait}[j] == \text{FALSE})) \ {\} ; \ // \ Busy \ Wait
\]

\[
\text{wait}[i] = \text{FALSE};
\]

\}

EXECUTE CRITICAL SECTION CODE

EXIT CODE:

\[
\text{turn} = j;
\]

\[
\text{flags}[i] = \text{FREE};
\]

(a) [12 points] Prove or disprove that the progress condition is satisfied by the above solution.

(b) [12 points] Prove or disprove that the mutual exclusion condition is satisfied by the above solution.
YOU CAN WRITE IN THIS PAGE TOO.