

Name \_\_\_\_\_

All questions are worth 10 points. Maximum score: 80

1. **Process A** is running on a preemptive multiprocessing system. **Process B**, is waiting for I/O, and **Process C** is ready. What are the steps that must occur for **Process A** to stop running, **Process B** to become ready, and **Process C** to start running? Be as specific as possible.

2. What is the difference between pre-emptive and non pre-emptive scheduling?

3. Explain the meaning of the term "Context." Where is a context stored when a process is running? Where is the context stored when the process is not running?

4. What is busy waiting? Give an example from the bounded-buffer problem.

5. Suppose the following jobs arrive in the following order at time zero. Give the GANTT chart and the average wait time for the following algorithms: First Come First Served, Shortest Job First, and Round Robin with a quantum of 4. Assume that a process switch requires zero time. Calculate the average wait time for both algorithms.

<b>Job Number</b>	<b>Burst Time</b>
<b>1</b>	<b>5</b>
<b>2</b>	<b>9</b>
<b>3</b>	<b>7</b>
<b>4</b>	<b>2</b>
<b>5</b>	<b>3</b>

6. Suppose the following jobs enter the ready queue at the times given. Give two GANTT charts for these jobs, one for the non-preemptive Shortest Job First algorithm, and one for the preemptive Shortest Job First (aka Shortest Remaining Time First). Compute the average waiting time for each.

<b>Job Number</b>	<b>Burst Time</b>	<b>Arrival Time</b>
<b>1</b>	<b>8</b>	<b>1</b>
<b>2</b>	<b>2</b>	<b>3</b>
<b>3</b>	<b>3</b>	<b>4</b>
<b>4</b>	<b>1</b>	<b>6</b>
<b>5</b>	<b>5</b>	<b>8</b>

7. The following two processes have a shared variable  $S$  which has the value 5. After the following two operations have both been executed once,  $S$  can have three different values. What are they and how do they occur?

**Process 1**

...  
 $S = S * 2;$   
...

**Process 2**

...  
 $S = S * 3;$   
...

8. What is starvation? Give an example of a scheduling algorithm that suffers from starvation. Also give an example of an algorithm that *does not* suffer from starvation.