CS 452 Sample Exam

Name: (print)_____________________________________________

Instructions: Failure to follow instructions will result in a ten (10) point penalty. Print (and not scribble or sign) your name and ID. Put all your answers in the blue book. Only answers in the blue book will be graded. All pages of this test must be left intact and turned in with the blue book.

1. For the following set of execution profiles, develop Gantt charts and the average waiting and turnaround times for each of the following scheduling policies:

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival</th>
<th>Burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Round Robin (quantum = 3)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(b) Shortest Remaining Time First</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

2. Consider a multilevel feedback queue in a single-CPU system. The first scheduling algorithm uses the MFQS algorithm described in the text, i.e., the first two levels uses Round Robin scheduling with time quantum of 8 and 16 ms respectively. The third queue uses FCFS. With the following process characteristics:

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>25</td>
<td>3</td>
</tr>
</tbody>
</table>

(a) Show the Gantt chart for the entire session.

(b) Compute the average waiting time

(c) Compute the average turnaround time.
3. Using semaphores, show how to synchronize five processes whose permissible execution order is specified by the following precedence graph.

Minimize the number of semaphores that you use, and do not impose constraints not specified in the graph. For example, T₂ and T₃ can execute concurrently after T₁ completes.

4. You have just been hired by Mother Nature to help her out with the chemical reaction to form water, which she doesn’t seem to be able to get right due to synchronization problems. The trick is to get two H atoms and one O atom together at the same time. The atoms are threads. Each H atom invokes a procedure hReady when it is ready to react, and each O atom invokes procedure oReady when it is ready. For this problem, you are to write the code for hReady and oReady. The procedure must delay until there are at least two H atoms and one O atom present, and then one of the procedures must call the procedure makeWater, which just prints a debug message that water is made. After the makeWater call, two instances of hReady and one instance of oReady should return. Write the code for hReady and oReady using semaphores for synchronization. Your solution must avoid starvation and busy-waiting. Use the following data structures:

```c
Semaphore hArrived;  // initially 0, for oxygen threads to wait on
Semaphore hCanLeave; // initially 0, for oxygen threads to wait on
Semaphore mutex;     // initially 1

void hReady ( ) {
    :
}

void oReady( ) {
    :
}
```

5. A system is composed of four processes, p₁ through p₄, and three types of resources, R₁ through R₃. There is one unit each of R₁ and R₃ available.

a. p₁ requests a unit of R₁ and a unit of R₃.

b. p₂ produces a unit of R₁ and a unit of R₃ and requests one unit of R₂.

c. p₃ requests a unit of R₁ and a unit of R₃.

d. p₄ produces a unit of R₂ and requests one unit of R₃.
Show the resource allocation graph to represent the system state. Which, if any, of the processes are deadlocked in this state? Identify the processes in the cycle if deadlocked. Otherwise show any safe sequence of execution.

6. Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the following algorithms place processes of 212K, 417K, 112K, and 426K (in order)?

   a) First-fit
   b) Best-fit
   c) Worst-fit
   d) Which algorithm makes the most efficient use of memory? Justify your answer.

7. A computer uses the buddy system for memory management. Initially it has one block of 256K at address 0. Consider the following requests and releases:

   (a) A requests 5K
   (b) B requests 25K,
   (c) C requests 35K
   (d) D requests 20K
   (e) B terminates
   (f) E requests 65
   (g) A terminates
   (h) F requests 17

How many blocks are left at this point and what are their sizes and addresses? (Graphical interpretation required).