CS 314-002 Operating Systems
Spring 2017
Quiz #5 on February 28, 2017 (SOLUTIONS)

Your Last Three Digits: ________________
(please do NOT write all of your student ID or your name)

Grade: ______

**********************************************************************

(1) How is “process deadlock” different from “(process) starvation”?

In “processes deadlocks”, all processes are stuck (waiting for themselves), while in “process
starvations”, there always is at least one process that keeps running.

(2) What are the four necessary conditions for a process deadlock to occur?

- Non-preemptive resource
- Hold-and-wait (processes hold resources while they wait for other resources)
- Mutually-exclusive resource
- Circular-wait

(3) Why is it difficult to eliminate the condition of “mutual exclusion” to prevent a process deadlock
from occurring?

It is difficult, since converting non-preemptive resources to preemptive resources will ruin
the operations of such non-preemptive resources. For example, DVD-burners and printers
are preemptive resources. Making them preemptive resources will screw up the results if
their use.
(4) In the classroom, we discussed what we can do to make sure one of the four necessary conditions for a deadlock is not satisfied. Is it possible to have a technique that never causes “circular wait”? If yes, describe how.

One of the techniques is to avoid a circular-wait from happening is as follows. First, we assign a unique ID for each resource available in a computer system. Then, the operating system requires each process to make requests to resources either in the ascending or the descending order of the resource IDs. This will prevent a circular-wait from happening.

(5) Suppose a system has the following status of the resource:

<table>
<thead>
<tr>
<th>Process</th>
<th>Assigned</th>
<th>MAX Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>

If process B requests 1 (one) additional instances of this resource, is the system safe? If it is safe, prove that it is safe. If not, prove that it is unsafe.

The request for 1 (one) more instance of the resource by process B still keeps the system safe. It is because there is still a way to finish all three processes after the request is approved by the operating system. The following is how.

(a) After 1 (one) more instance of the resource is assigned to B, we will have 3 (three) more resource available in the system (i.e., “Available = 3”). Process B will have 3 (three) instances of the resource.

(b) After (a), process B can be finished by assigning 2 (two) more instance of the resource and B can finish. As soon as B finishes, all 5 (five) instances of the resources will be available (i.e., “Available = 5”).

(c) With 5 (five) resources, process C can finish.

(d) After process C finishes, we will have 11 (eleven) resources in the system (i.e., “Available = 11”). Using the 11 (eleven) instances of the resource, process A can finish.