

CSCI 40300/ECE 40800
Operating Systems– Fall 2016
Quiz 5
Solutions

Name: _____

Question:	1	2	3	Total
Points:	5	5	5	15
Score:				

Normalized Total to 100 = $100 \times \text{Total}/15 =$ _____ (what will appear in Canvas gradebook).

1. (5 points) Which of the following does this code fragment fail to achieve?

```
// Process 0
while(TRUE) {
    while(turn != 0) {}; /*loop*/
    critical_region();
    turn = 1;
    noncritical_region();
}

// Process 1
while(TRUE) {
    while(turn != 1) {}; /*loop*/
    critical_region();
    turn = 0;
    noncritical_region();
}
```

- A. Mutual exclusion
- B. Progress**
- C. Bounded wait
- D. None of the above

Answer: We said in class that strict alternation as in this case violates the progress requirement. When process i wants to enter its critical region and process j is in its noncritical region, process i should be able to proceed and enter its critical region any number of times. But with strict alternation this does not work. They take turns, whether the other process is in its noncritical region or not.

2. (5 points) Within a process, a critical section consists of any sequence of instructions that:

- A. Produce a result necessary for the successful completion of a subsequent instruction.
- B. Are frequently executed on behalf of the process.

C. Must be accessed on a mutually exclusive basis.

D. Have a high-priority value associated with them.

3. (5 points) Suppose a thread is running in a critical section of code, meaning that it has acquired all the locks through proper arbitration. Can it get context switched? Why or why not? Explain in a few sentences.

Answer: Yes it can get context switched. Locks are independent of the scheduler. In fact, locks were the solution to make the critical section problems independent of scheduler issues. The idea is that even if the thread holding the lock and in its critical section gets preempted and context-switched, other threads trying to enter their critical sections will not be able to get the lock and, therefore, not enter their critical sections, even if they are given the CPU. On the other hand the new thread scheduled on the CPU, if it is in its non-C.S. code, it will be able to execute.