Your name:

CSE 30341 Operating Systems: Module 2 Exam

OPEN BOOK, OPEN NOTES, CLOSED ELECTRONIC SEARCHES

INDIVIDUAL EFFORT

DURATION: 30 MINUTES

All questions carry equal weight

---

1. The critical section designates regions in which many processes might be changing common variables. According to the textbook, a solution to the critical section problem must satisfy the three requirements of mutual exclusion, progress and bounded waiting. Suppose you design a new solution which guaranteed *mutual exclusion* and *progress* but not the *bounded wait* requirement. This means that your solution will violate critical section requirements and allow multiple processes to modify the variables protected by the critical section. True or False? Justify.

2. Let us suppose that you are proficient in using assembly code within C and so you write a (non-portable) code that uses the TestandSet instruction to directly implement semaphores without requiring the services of the OS. Your goal is to avoid a context switch to make a system call. Assume a single processor machine for this question. Hence, you wish to avoid a spinlock. Can you achieve good performance by replacing the block() with a sleep() system call? The sleep(n) system call will allow your process to sleep for *n* seconds (i.e., your current process will be moved to the wait state for *n* seconds). Explain.
[Page: 202 ‘if (S->value < 0) { add this process to S->list; block;}’ to ‘while (S->value < 0) sleep(n);]

3. The Dining philospher’s problem can potentially lead to deadlocks. Which of the following variants are deadlock free? Why?
   a. Even number of philosophers, odd number of forks
   b. Odd number of philosophers, even number of chopsticks

Your name:

4. A monitor construct can be used to solve the critical section problem. Can a monitor function deadlock by recursively calling itself? Explain.

5. Suppose that you run the Banker's algorithm safety check and find out that a requested allocation will leave the system in an unsafe state. This means that
    a. Proceeding with the allocation will cause a deadlock
    b. Proceeding with the allocation will cause a deadlock with high probability
    c. Proceeding with the allocation might cause a deadlock
    d. Proceeding with the allocation will not cause a deadlock

6. Exercise 7.6: Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.