# CS140 Operating Systems and Systems Programming
## Midterm Exam

## October 29th, 2004

## (Total time = 50 minutes, Total Points = 50)

**Name: (please print)_____**

**In recognition of and in the spirit of the Stanford University Honor Code, I certify that I will neither give nor receive unpermitted aid on this exam.**

**Signature:_____**

**This examination is close notes and close book. You may not collaborate in any manner on this exam. You have 50 minutes to complete the exam. Before starting, please check to make sure that you have all 8 pages.**

| | |
|---|---|
| **1** | |
| **2** | |
| **3** | |
| **4** | |
| **5** | |
| **6** | |
| **7** | |
| **8** | |
| **9** | |
| **10** | |
| **Total** | |

Name:_____

1. (9 points) You are implementing an OS with a partner. You ask your partner to write some synchronization functions:
   lock(L) - Locks L.
   unlock(L) - Unlocks L.
   sleep() - Makes the current thread sleep until woken with wake().
   wake(T) - Wakes thread T.

   Your partner decides include several combination functions:
   lock_sleep(L) – Atomically lock(L) then sleep().
   unlock_sleep(L) - Atomically unlock(L) then sleep().
   lock_wake(L,T) – Atomically lock(L) then wake(T).
   unlock_wake(L,T) – Atomically unlock(L) then wake(T).
   You may assume that all of these functions, but no others, execute atomically. Use them to implement the semaphore operations.

Data fields and initial values:

P(): V():

2. (6 points) A common trick used in computer systems is to interpose or "wrap" an interface. To wrap an interface A, every call to that interface is redirected to another interface B which does some processing and passes the call on to interface A. For example, the interface A has two functions:

   ```
   int foo1(int x);
   int foo2(double y);
   ```

   then interface B would have functions that look like:

   ```
   int foo1(int x) { /* do processing */ return A->foo1(x); }
   int foo2(double y) { /* do processing */ return A->foo2(y); }
   ```

   One example usage of wrapping would be an interface that counted all the calls to methods of A. Interface B would be given to all places that interface A was referenced and hence would get invoked before A was called each time.

   Assume your project partner adds some general support for wrapping interfaces to your OS. The question comes up what happens if the interface being wrapped (e.g. A in our example) is actually a monitor. Your partner claims that if the interface is a monitor, the wrapping interface (e.g. B in our example) should also be a monitor. You claim the wrapping interface should never be a monitor. Are you, your partner or neither of you right in this? Justify your answer.

3. (4 points) Is it possible to implement a critical section with wait-free/non-blocking synchronization? Justify your answer. (Hint: Think about the critical section definition.)

4. (4 points)
   (a) Under what conditions (if any) does a non-preemptive CPU scheduler move a process from the ready state to the running state?
   (b) How about a preemptive CPU scheduler?

5. (4 points) Can a system with all jobs having the same priority ever suffer from the priority inversion problem? Justify your answer.

6. (4 points) Explain why a multiprocessor CPU scheduler might assign jobs to run on particular processors rather than having a global run queue serving all the processors.

7. (6 points) Describe how or where the operating system gets the following information about a process.
   (a) Where in a process to start executing when a program started running?
   (b) The first available heap memory address?
   (c) The initial value for the stack pointer for the process?

8. (4 points)
   (a) Does the placement of a process's page in physical memory by the virtual memory subsystem affect the amount external fragmentation?
   (b) How about the amount of internal fragmentation?
   Justify your answer.

9. (4 points)
   (a) Can adding prepaging to a demand page virtual memory system decrease the number of page faults?
   (b) How about can it increase the number of page faults?
   Justify the answer.

10. (5 points) Which of the following conditions would necessarily be present in a system that is thrashing.
    (a) Paging disk nearly 100% active.
    (b) Paging disk near 100% of capacity
    (c) Some processes on the machine are blocked much of the time.
    (d) CPU is 100% busy
    (e) CPU is 100% idle.
    Justify your answer for each.