# CS140 Operating Systems and Systems Programming
## Midterm Exam

**February 10th, 2006**

**(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 6 pages.**

| | |
|---|---|
| **1** | |
| **2** | |
| **3** | |
| **4** | |
| **5** | |
| **Total** | |

Name:_______________

1. (12 points total) Answer the following three-part question. Be sure to justify your answers.
   (a) Describe a CPU scheduling algorithm that guarantees that no process will starve given a finite number of processes.
   (b) Describe a CPU scheduling algorithm that can result in starvation. Provide an example workload that will result in starvation.
   (c) Is it possible that a CPU scheduling algorithm can result in a deadlock?

2. (10 points total) Two-part question:
   (a) Some machines put user process page tables into virtual memory so the OS can easily page the page table to disk. In such a system an application can take a page fault on either its memory or its page table. Describe how an OS could page the page tables on the x86 architecture where the page tables are stored in physical memory. Be sure to include a description of when and how pages are saved to and retrieved from disk.
   (b) Assuming an architecture like the x86 with a hierarchical page table, is it possible for the page table of a process to be bigger than the physical memory? Justify your answer.

3. (8 points) In Pintos we disable interrupts to synchronize access to some kernel data structures. Your partner decides to make Pintos multiprocessor-ready by replacing the disable interrupts/enable interrupts we use for synchronization with a spin lock. Would this work? Justify your answer.

4. (12 points) Write a monitor that implements an alarm clock functionality where a thread can call the WaitFor(int *ticks*) method that causes that thread to wait for *ticks* timer ticks. You can assume the monitor has a method Tick() that is called on every timer tick (but not at interrupt level.) Extra credit (2 points) is available for using only one condition variable.

```
monitor struct {
    void WaitFor(int ticks);
     void Tick();
 } AlarmClock;
```

Show the code and data for the WaitFor() and Tick() functions in a C-like language that contains support for monitors.

5. (8 points total) For the following error messages, state if the message would be generated during the first or second pass of the linker. Be sure to justify your answer.
   (a) “External Reference FOO not found.”
   (b) “Duplicate symbol definition FOO.”