CS 140: Operating Systems and Systems Programming
Midterm Exam

Ben Pfaff
Stanford University
Department of Computer Science

July 19, 2005

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.

Name (please print): _____

Signature: _____

- This exam is **closed notes** and **closed book**.
- **No collaboration** of any kind is permitted.
- You have **60 minutes** to complete the exam.
- There are **6 questions** totalling **60 points**. Some questions have multiple parts.
- Please check that you have all **6 pages**.
- Before starting, **write your initials on each page**, in case they become separated during grading.
- Please print or **write legibly**.
- Answers may not require all the space provided. Complete but **concise answers are encouraged**.
- **SCPD students:** If you wish to have the exam returned to you at your company, please attach an SCPD routing slip. Otherwise, we will assume that you will pick up your exam in class. Routing slips can be obtained just outside the classroom (if you take the exam on campus) or on the SCPD website.

Question	Points	Score
1	13	
2	8	
3	15	
4	5	
5	9	
6	10	
Total	60	

1. Condition Variables.

   (a) (3 points) To be semantically correct, the `wait`() operation on a condition variable must execute three operations. Describe, in 10 words or less, each of these operations in their order of execution:
   (1)

   (2)

   (3)

   (b) (3 points) Which of the above operations must execute atomically? Describe a race condition that would result if these steps were interrupted.

   (c) (5 points) The value of a semaphore records a "history" of how many times P and V operations have completed, but a condition variable does not have history. How, then, does the user of a monitor avoid missing an important event?

2. The DeathStation 8000 is a uniprocessor machine designed from the ground up to support cooperative multitasking under DeathOS, a multiprocessing, multithreaded OS. The DeathStation 9000, now in development, will support preemptive multitasking. This obviously requires changes in DeathOS. Answer the following questions about other changes that are also necessary.

   (a) (5 points) What changes might the upgrade require in application software, to keep it from malfunctioning in this new environment? Consider how the assumptions that processes and threads may make change between cooperative and preemptive multitasking environments.

   (b) (4 points) What additional hardware, or modifications to hardware, might the upgrade require?

3. Each of the events listed below interrupts the flow of execution. Answer the following questions about each event:

   - Is the event synchronous or asynchronous?
   - What state must be saved and restored?
   - Who (caller, callee, etc.) must save and restore this state?

   Your answer should discuss generic components of machines (e.g. "the instruction pointer"), instead of being oriented toward any particular machine architecture (e.g. "EBX in the 80x86 SVR4 ABI").

   (a) (5 points) Procedure call

   (b) (5 points) System call

   (c) (5 points) Process switch caused by a time slice expiring

4. (6 points) Some coding standards for applications require that every block obtained by a memory allocation function (e.g. `malloc`()) must be released with a corresponding deallocation function (e.g. `free`()) before the process terminates. Explain why this practice may cause additional work for the OS's virtual memory subsystem and thus slow down the system.

5. Ben's allocator uses a linked list to track free blocks. In the allocator, all blocks are a multiple of 4 bytes in length. Answer the following questions assuming that the free list contains just two blocks, of 28 and 16 bytes each, in that order. Ignore any space required by bookkeeping overhead.

   (a) (3 points) List a sequence of `malloc`() calls that would succeed given first-fit allocation, but fail with best fit.

   (b) (3 points) List a sequence that would succeed with best fit but fail with first fit.

   (c) (3 points) List a sequence that would succeed with worst fit but fail with best fit.

6. Segmentation & Paging.

   (a) (5 points) The IBM 370 and Intel $80x86$ architectures implement segmentation and paging at the same time, in very different ways, but both architectures perform segmentation before paging. Would it ever make sense to perform paging before segmentation? If so, outline a hypothetical MMU architecture of that form and the rationale behind it. If not, explain why not.

   (b) (5 points) Most architectures designed recently do not support segmentation. Can segmentation be simulated with paging? Describe how so. Specify your assumptions.