Stanford University
Computer Science Department
CS 240 Quiz 1
Spring 2006

May 3, 2006

This is an open-book exam. You have 50 minutes to answer eight out of ten questions. Write all of your answers directly on the paper. Make your answers as concise as possible. Sentence fragments ok.

NOTE: We will take off points if a correct answer also includes incorrect or irrelevant information. (I.e., don't put in everything you know in hopes of saying the correct buzzword.)

Question	Score
1 - 5	
6 - 10	
total	

Stanford University Honor Code

In accordance with both the letter and the spirit of the Honor Code, I did not cheat on this exam nor will I assist someone else cheating.

Name and Stanford ID:

Signature:

Answer eight of the following ten questions and, in a sentence or two, *say why your answer holds.* (5 points each).

1. Given a program P that runs on Caprpriccio and does no explicit locking of its own. What does P correspond to in terms of MESA? What is the most straightforward way you could adapt Eraser to find race conditions in such programs? (Hint: think of how it checked OS code.) How useful do you expect this to be?

2. Consider the Therac threading model. How does it compare to Events in Ousterhout? To threads in Capriccio? Does it contradict or support the arguments of these people?

3. Consider very-broken code that uses the "double-check" lock idiom:

```
0:   int *p = 0;
....
1:   if(!p)
2:      lock(l);
3:      if(!p) {
4:         int *t = malloc(sizeof *t);
5:         *t = 3;
6:         p = t;
7:      }
8:      unlock(l);
9:   }
10:  x = x / *p;
```

How can an optimizing compiler of the sort that Boehm complains about interact badly with the use of the `t` temporary? Assume you don't use an optimizing compiler: if another thread has done a `free` previously, what can happen if the initializing thread gets context switched immediately after line 6 and another thread executes line 10 on another processor?

4. Your 240 cohort sees the number 98.2 in the VMS paper and says "that is strange." Why? Assume the methodology that obtained it was accurate: What is a plausible reason the number is what it is?

5. Superpages: Jimbob says the most important thing to know about the future is when an object will die. Bobjim says that the most important thing is how an object will be populated. How could you use these facts? Give an experimental reason for why one of these guys is more right.

6. Most OSes don't have superpages. VMware hires Navarro to hack ESX so that ESX will trick guest OSes into using superpages. At a high-level: how would he do so? Do you expect this approach to work well? Why or why not?

7. What would suck if ESX only used a "slow" average for measuring active memory?

8. You use the Safe-C compiler in the Rinard et al papers to implement memory protection in Nooks. Give three things you are pretty sure would change or not change in Figure 8, "Nooks" (and why!).

9. "Take from the head, put on the tail." Give three places we have seen such a data structure and, in each case, what high-level policy it was attempting to achieve. Give one place where we might have seen this but did not (and why).

10. You see the code:

```
1:    p = p + k;
2:    if(p == q) {
3:          x = *p;
4:          y = *q;
5:    }
```

How could the dereference on line 3 be illegal, but the one on line 4 be legal? Give pseudo-code for how the expression on line 1 will be rewritten by the failure-oblivious compiler.