CS 240 Quiz Questions 2002 - 2004

Stanford University
Computer Science Department

April 19, 2004

These quizes were all open-book exams. In general you were asked to answer somewhere between 8 out of 10 to 10 out of 12 given questions. The questions below are actual questions that were given.

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.)

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 the following questions and, in a sentence or two, say *why* your answer holds.

1. Gabriel uses "pc-losering" as an example of how "worse is better" can triumph even when it makes an interface more complex. Taking his explanation of how Unix works at face value: explain how a luser-level library could make the Unix interface into the Right Thing. Does this invalidate his argument?

2. The Therac paper is a study in severed feedback loops, ranging from AECL not informing customers of problems in the Therac-25 to the video camera being off when a patient was zapped. From the paper, give three (other) examples of such missing feedback loops.

3. What would be a difficulty in detecting the Therac races using an Eraser-type approach?

4. The Therac code in Figure 4 on page 34 shows an overflow error involving the "Class3" variable. If we assume the code in Figure 4 accurately represents the code in the Therac, describe the race that exists between Set-up Test and Lmtchk.

5. The Eraser paper claims that they have never observed more than 10,000 distinct sets of locks for a given program. Give a data structure and locking policy that would pass this limit.

6. Eraser only ensures that data is protected by a consistent set of locks. Give an intuitive sketch of a class of race conditions it will miss, an example, and explain how Eraser could be extended to handle them.

7. Assume we have two threads, T1 and T2, that execute the following code in the following order:

```
int *q = NULL; /* shared global variable */

          T1                                T2
1:   int *p = malloc(sizeof *p);
2:   *p = 0;
3:   q = p;
4:                                     while(q == NULL)
```

```
 5:                                                  ;
 6:                                               *q = 1;
 7:                                               q = 0;
 8:  while(q != NULL)
 9:     ;
10:  printf("p = %d\n", *p);
11:  exit();
12:                                               exit();
```

Assuming naive memory semantics (if you don't know what this means, you are just fine): What will Eraser do for this code? Is its behavior correct?

8. The set of locks that are acquired at any given time is a global property (e.g., switching threads has no effect on it). Given this, why does Eraser use per-thread locksets rather than a single global lockset? Give an intuition and a simple example of the problem a global lockset would cause.

9. The Eraser paper claims "A write access from a new thread changes the state from Exclusive or Shared to the Shared-Modified state..." But Figure 4 says that a write by any thread in the Shared state takes it to the Shared-Modified state. Which is right?

10. Eraser, like most tool papers, defines a race condition as a reference to shared memory without holding a (consistent) lock. Give two intuitive, short examples showing that this definition is both too strong and too weak.

11. For the following snippet of Mesa monitor code:

```
void foo() {
    signal(c);
}
```

If the code runs on a system with different priority processes, what is a potential way that it will generate useless overhead? Is there an automatic way to fix this problem?

12. Would a Mesa programmer have any use for a Mesa version of Eraser?

13. Give two examples where Mesa makes a "New Jersey" style decision.

14. Explain from the Mesa paper: "...[while] any procedure suitable for forking can be called sequentially, the converse is not true."

15. Your ex-140 partner loudly declares that if the semantics of the "wakeup-waiting switch" is good for naked notifies, it must be good for normal notifies, which should be replaced with them. Can you do this substitution and preserve correctness? (List any assumptions you must make.)

16. The mesa paper claims that because of their semantics on waits, "verification is actually made simpler and more localized" (page 11). Give an intuition for why this is true or false.

17. What would Hoare-wakeup semantics correspond to in a message passing system?

18. What is the parallel in the procedure approach to a message sitting on a queue in the message-based approach?

19. The duality paper strongly restricts where waits can be placed in an MPP system. Your mom thinks this is stupid and wants you to be able to place waits arbitrarily. Explain what would be required to do this strictly within the context of MPP (i.e., you cannot change the message loop to use threads, though you will have to do things the duality paper does not).

20. Capriccio, Figure 1: what is a plausible reason that performance for the non-Capriccio systems goes down so quickly?

21. Give two examples of where a static stack allocation scheme would perform better than Capriccio's dynamic approach.

22. Identify several potential weaknesses in the VMS memory system and describe experiments to measure them.

23. You increase the size of the free list to 80% of memory. Describe a workload that will run better, and one that will run worse.

24. When the reference bit of a base page within a superpage is reset, the superpage management system demote the superpage speculatively, recursively performing the demotions with a probability $p = 1$. Suppose we set $p = 0.5$ instead. Give one reason why the system might perform better because of that, and one reason why the system might perform worse because of that.

25. When does the coalesce daemon run in the superpage system? What would the effect of running it more often be?

26. Assume you run a process that has a 513K text segment on the Alpha system described in the superpage paper. What are all the different actions in the superpage system that can be triggered when you jump to the first instruction in this process? (Hint: make sure you consider more than just reservations.)

27. When measuring active memory (for use in idle memory tax calculations), ESX Server keeps three statistical averages: a "slow" average, a "fast" average, and a "very fast" average. Explain the problem that could result if ESX stopped using the "slow" average.

28. Your ex-140 partner (fresh from their triumphant foray into monitor semantics) decides to get the same effect of a balloon driver using an application running on top of the guest OS. On ESX's request the application will aggressively touch many pages of memory, causing them to be "recently used" and the guest OS to page out other, less recently used pages. Is this a bright idea?

29. A balloon driver must use fairly non-portable device driver interfaces. It would have been much more portable if Carl had used a user-level application to do ballooning instead. Would this have worked or not?

30. What bad things would happen if ESX used the average rather than the max of the three moving averages of memory usage?

31. Assume you want to add ESX memory taxation to the VMS system. In a few sentences, describe why this might make sense and how you would integrate it.

32. List two realistic, specific scenarios where running VMS (or copies of VMS) on top of ESX will give better performance.