

Instructions

- This final is closed book and notes.
- If a question is unclear, write down the point you find ambiguous, make a reasonable interpretation, write down that interpretation, and proceed.
- State your assumptions and show your work. **Write brief, precise, and legible answers.** Rambling brain-dumps are unlikely to be effective. **Think before you start writing** so that you can crisply describe a simple approach rather than muddle your way through a complex description that “works around” each issue as you come to it. **Perhaps jot down an outline** to organize your thoughts. And remember, a **picture can be worth 1000 words.**
- For full credit, show your work and explain your reasoning.
- There are 11 problems on this exam. The first 10 are worth 8 points each and the last is worth 20.
- The exam is scheduled Monday May 12 2-5PM JGP 2.102. The exam is not designed to require the full 3 hours to complete.
- Write your name on this exam

Name:

2

1. (8) In the context of authentication protocols, define the meaning of “jurisdiction” and give an example of its use.

2. (8) What protocol can be used to ensure that two nodes are guaranteed to take the same action at the same time?

3. (8) Why is a bitmap a better choice than a linked list for tracking free disk blocks?

4. (8) In lab 4, the `faultalloc` testprogram allocated a new page at `va` on a page fault for a bad reference to `va`. The `faultalloc` program's `umain` looks like this:

```
void
umain(void)
{
    set_pgfault_handler(handler);
    cprintf("%s\n", (char*)0xDeadBeef);
    cprintf("%s\n", (char*)0xCafeBffe);
}
```

You also ran `faultallocbad`, which replaced the `cprintfs` above with a call `sys_cputs((char *)0xDEADBEEF, 4)`. If things were working properly, `faultallocbad` died rather than allocate a new page. The exercise asked you to understand why `faultalloc` and `faultallocbad` behaved differently.

Why do `faultalloc` and `faultallocbad` behave differently?

5. (8) In file systems, what is a *hard link*?

6. (8) Why is it difficult to implement hard links in the FAT file system?

7. (8) An important operating system architecture is a *virtual machine monitor* or *hypervisor*. A hypervisor can run multiple unmodified operating systems on separate virtual machines—each guest operating system believes it is running on its own machine, and the guest operating systems are isolated from one another so that a bug in one cannot cause a problem in another.

One of the challenges to making this work is that guest operating systems must run with the USER status bit set so that the hypervisor can limit what guest OSes can do. As a result, guest OS code cannot execute privileged instructions like `lcr3()` to install a page table for itself (e.g., the kernel page table for the guest OS) or for processes that the guest OS creates.

Describe how a hypervisor could ensure that unmodified guest operating systems can function properly (while still ensuring isolation among the virtual machines and while still ensuring that processes within the guest OS are isolated from one another.)

8. (8) Consider the following sequence of reads and writes (assume all values are initially 0; each operation occurs at the exact real-time moment specified for each line):

Time	Node 1	Node 2	Node 3	Node 4
0:01	write(A, 12)	write(A, 22)		
0:02	write(B, 13)	write(B, 23)		
0:03	write(C, 14)	write(C, 24)	23 = read(B)	13 = read(B)
0:04	write(D, 15)	write(D, 25)	12 = read(A)	12 = read(A)
0:05			25 = read(D)	25 = read(D)
0:06			13 = read(B)	13 = read(B)

Indicate whether each of the following statements is true or false and explain your answer.

- (a) The above system implements linearizability
- (b) The above system implements sequential consistency
- (c) The above system implements causal consistency
9. (8) Although passwords have limitations, sometimes they must be used for authentication. If you must design a system that uses passwords for authentication, describe three ways to improve their security.

Name:

6

10. (8) The MegaGiga hard disk rotates at 10000 rpm (6ms/rot) with a seek time given by $= 1 + 0.001t$ msec, where t is the number of tracks the arm seeks. Assume a block size of 512 bytes, 1024 sector/track, 8192 tracks, and 4 platters. The disk has a 16MB track buffer. The disk controller can DMA read or write data between memory and the disk device at a rate of 100MB/sec.

(a) What is the storage capacity of this disk?

(b) Estimate the worst case delay to read 512 bytes from this disk

(c) Estimate the expected time to read 20 consecutive MB from a random location on disk.

11. (20) Concurrent programming

Consider the following, simple game: three player each have an unlimited supply of color cards. Player Blue has a big stack of blue cards. Player Green has a big stack of green cards. Player Red has a big stack of red cards. The Dealer (a fourth entity) has a big stack of blue cards, a big stack of green cards, and a big stack of red cards.

The Dealer randomly selects two different colored cards and places them on the table. The player who has the third color removes the two dealer cards from the table and takes the three cards (two from the table and one from the player's hand) and places them in a pile. The Dealer then puts another two cards on the table, and the cycle repeats.

Write a `Table` object to synchronize game play among four threads (one thread each for Dealer, Red, Blue, and Green). The table object has two public methods:

```
public class Table{
    public:
        int const BLUE = 0;
        int const GREEN = 1;
        int const RED = 2;

        dealerPutCards(boolean cardList[3]);
        playerPullCards(int playerCardColor);

        ...
}
```

The method `dealerPutCards()` passes an array of three booleans, two of which are true and one of which is false. It returns once the matching player has played.

The method `playerPullCard()` passes in an `int` representing a color. It returns when the player's color is the winning color.

To receive credit on this problem, you must follow the coding standards described in the handout and project.

List Table's member variables and indicate how they are initialized

Name:

8

Implement dealerPutCards(boolean cardList[3])

Implement playerPullCards(int playerCardColor)