CS140 Operating Systems and Systems Programming
Final Exam

March 19, 2007

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 closed notes and closed book. You may not collaborate in any manner on this exam. You have 180 minutes (3 hours) to complete the exam. Before starting, please check to make sure that you have all 21 pages.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
Total	

(1) (20 points) – Software running in a virtual machine monitor (VMM) can peer inside the virtual machine (VM) to look at memory and registers including the page tables for the processes running inside the VM. One proposed use of this visibility is to look for mistakes or bugs in operating systems. For each of the observations below describe if it is a bug or if not what normal operation would cause the observed behavior. Assume you have an Unix-like operating system running in the VM on a x86 architecture system.

(a) The VMM observes the same physical page present in multiple different page tables for different processes.

Continued on next page…

Question 1 continued…

(b) The VMM observes the same physical page present in multiple different places in the same page table.

Continued on next page…

Question 1) continued….

(c) An application process's page that is known to be both read and written by the application is mapped with a read-only page table entry in the process's page table.

Continued on next page…

Question 1 continued…

(d) A physical page that is mapped both in an application's address space and the file system buffer cache.

(2) (16 points) When talking about user level memory allocation algorithms we mentioned that the fragmentation of a good algorithm is around M*log(*n*) where M is the size of the allocated data and *n* is the ratio between the smallest and largest blocks.

(a) Since the log(1) is 0, this seems to imply that there is no fragmentation when *n* is one. Why is this?

Continued on next page…

Question 2 continued…

(b) It is possible for a system where *n* is large (say 1000) to still have no fragmentation. Explain why this is possible?

(3) (18 points) Assume that a disk drive manufacturer has invented an *atomic disk* drive. It is identical to a normal disk drive except it can take groups of write commands and perform them atomically. Once the drive has accepted the group-write request all the writes will all be written to the disk even if the power fails in the middle.

(a) Describe how a new atomic disk could be used by a file system to solve some of the problems of crash recovery. Give examples of how the file system would use it to implement the creation of a file.

Continued on next page…

Question 3 continued…

(b) There is a disagreement among the file system designers of what the atomic disk means for the write ahead logging technique. Some say it is still a valuable technique even with an atomic disk while others claim it is pointless now. Which is right? Justify.

Continued on next page …

Question 3 continued…

(c) Describe how the disk drive manufacturers could make group writes of the atomic disk go faster than if it was given the write requests one at a time.

(4) (18 points) In class we described three file descriptor structures:
   (a) Indexed files.
   (b) Linked files.
   (c) Contiguous (extent-based) allocation.

Each of the structures has its advantages and disadvantages depending on the goals for the file system and the expected file access pattern.

For each of the following situations, rank the three structures in order of preference. Be sure to include the justification for your rankings.

(a) You have a file system where the most important criteria is the performance of sequential access to very large files.

Continued on next page …

Question 4 continued…

For each of the following situations, rank the three structures in order of preference. Be sure to include the justification for your rankings.

(b) You have a file system where the most important criteria is the performance of random access to very large files.

Continued on next page …

Question 4 continued…

For each of the following situations, rank the three structures in order of preference. Be sure to include the justification for your rankings.

(c) You have a file system where the most important criteria is the utilization of the disk capacity (i.e. getting the most file bytes on the disk).

(5) (12 points) The Unix and Linux operating system support the ability to pass open file handles (e.g. the value return from the open() system call) between processes. Assume that you have a process that receives a file handle from another process that opened a directory. Describe what you would have to do to determine the name of the directory if you are only given an open file handle for the directory.

(6) (10 points) Describe the trick used in disk arrays (e.g RAID devices) to have redundancy for tolerating failures yet don't actually keep a replica the block on the disks.

(7) (10 points) Explain the reasoning behind the use of the LRU algorithm for virtual memory page replacement and file system buffer caches.

(8) (16 points) For each of the following protection systems, describe if they more resemble access control lists (ACLs) or capabilities. Be sure to justify your answer.

(a) A door with a lock that takes a key to unlock.

(b) A door with a combination lock.

(c) A door with a badge reader an employee needs to swipe to unlock.

(e) A door with a biometric fingerprint reader that unlocks for the right employee's fingers

(9) (14 points) Recently researchers have gotten interested in building storage systems out of untrusted storage. Untrusted storage is a storage system (i.e. disk or file system) that might return the wrong contents for a request. Some of the systems designed to handle untrusted storage have adapted a weird naming system where the message digest of an object is used to name the object. In order to request an object from the untrusted storage you ask for it by its name/message digest. Describe how this approach helps handle the challenges of untrusted storage.

(10) (14 points) Assume you are given a box of Ethernet adaptors that some of which are broken in one of two different ways.

(a) Some of the adaptors have a broken exponential back-off algorithm that will frequently generate the same back-off wait value every time.

(b) Some of the adaptors have broken destination address match functionality that reports all packets match the destination address.

Assume that you have to use some of the broken cards in order to connect a collection of machines together. Which of these problem cards would choose? Justify your answer.

(11) (16 points) The Internet Protocol (IP) was designed to handle all kind of problems you might see in a big inter-network. For each of the challenges below, describe what the IP protocol does to handle it.

(a) A cycle in the routing graph of the network.

(b) Network links that have a small maximum packet size.

(c) Routing table size in networks with billions of hosts.

(12) (16 points) The TCP/IP protocol exports an abstraction of a reliable byte stream using acknowledgement packets. Assume you are watching the network between two machines with a TCP/IP connection. What would cause the following observations:

(a) You see more than one acknowledgement packet for a packet sent.

(b) The number of acknowledgement packets you see sent to a machine is only ¼ the number of packets that it sent out.