CS140 Operating Systems and Systems Programming
Final Exam

March 19, 2008

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 19 pages.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
Total	

(1) (10 points) Explain how a network supporting a connection-oriented model (i.e. virtual circuits) can have packet headers that are significantly smaller than those of a datagram network. Be sure to describe how a packet with a header that is even smaller than a destination address can make it to the correct destination.

(2) (10 points) Explain why the time-to-live (TTL) field in the IP protocol header is important for the reliable flooding protocol used to distribute link state information.

(3) (10 points) You and your Pintos partner are discussing the distributed consensus problem on unreliable networks. Your partner mentions that the TCP/IP protocol guarantees that when a process successfully closes a TCP/IP connection, all data sent by that process has been received by the other machine. He suggests having two TCP/IP connections between the parties and having each send and then verify that the data was received by shutting down the connection. He claims that this will solve the consensus problem even in the face of unreliable networks. What is wrong with this argument?

(4) (10 points) Describe the solution used by the HDLC link protocol to handle data that contains a copy of the framing sentinel.

(5) (10 points) Consider the following techniques to store a user's password in a publicly readable password file like was done in Unix. For each of the techniques, state if it is likely to work or if it has a problem. If it has a problem, describe the problem.
(a) Use a secure hash function such as SHA1 to hash the password and store the hash in the file.
(b) Use a shared key crypto system to encrypt the password using the account's user ID as the encryption key. Store the encrypted password in the file.
(c) Use the private key of a public key pair to encrypt the password that is stored in the file. The private key is known only by the password-setting code and the public key is known to all.

(6) (10 points) Some computer protection systems allow the delegation of protection rights where one user can give rights to an object to another user provided they have the rights to give away. Would this kind of system be easier to implement in a system using capabilities or access control lists? Be sure to justify your answer.

(7) (10 points) Describe an algorithm that would take a capability based system and convert it to using access control lists where the user's access rights remain the same.

(8) (10 points) Describe a plausible file access pattern that works poorly with a file system buffer cache that uses an LRU replacement algorithm.

(9) (10 points) Explain two ways a poorly designed file prefetch algorithm can hurt the performance of a system.

(10) (10 points total)

(a) Explain why the original Unix file system showed much better performance on sequential reads of large files when the file system was newly created than after it had been running for a while.

(b) Explain how the BSD file system avoided this problem.

(11) (10 points) Assume you have a file system that uses write-ahead logging. You are given a flash RAM disk and a normal hard disk. Ignoring capacity issues and assuming you can not put both the log and the rest of the file system on the same disk, which disk would you suggest for the log and which would get the rest of the file system. Be sure to justify your answer.

(12) (10 points) Explain why file systems written for floppy disks where the media can be removed at anytime tended to be slower than file systems designed for only non-removable media.

(13) (10 points) Explain why the Unix *ls* command that prints the files in a directory was much faster than the *ls –l* command that prints the file names and their sizes. Assume the original Unix or BSD file system.

(14) (10 points) Explain how it would be possible to add the string “XXX” to every file in a file system yet have the file system report the same amount of disk space in use.

(15) (10 points) Explain the arguments for and against using a last-in-first-out (LIFO) variant of a first fit heap algorithm.

(16) (10 points) Explain why a conservative garbage collector can do the mark and sweep but not the compaction part of normal garbage collector.

(17) (10 points) List the actions that a user process could do that would cause your Pintos operating system to modify a page table. Include a description of the page table modifications caused by each action.

(18) (10 points) Under the page fault frequency algorithm for handling thrashing, how is the amount of memory that is given to a process determined.