CS140 Operating Systems and Systems Programming
Final Exam

March 22, 2002

(Total time = 165 minutes, Total Points = 165)

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 165 minutes (2 hours and 45 minutes) to complete the exam. Before starting, please check to make sure that you have all 19 pages.

1	
2	
3	
4	
Total	

1. File Systems (54 points)

(a) (9 points) Explain why a logging file system can perform significantly more writes to a disk than a non-logging file system yet have better write performance as seen by the applications.

(b) (9 points) Is it possible to have a disk contain over 50% fragmentation when using a contiguous allocation file system? How about a linked file system? Justify your answer.

(c) (9 points) Explain why it would be faster to read the first block of every file in the file system than it would be to read the last block of every file. Give as many reasons as you can think of. For simplicity, you can assume an Unix-like file system format.

(d) (9 points) Explain why the BSD Unix file system carefully writes the inode of a newly allocated file to disk before it writes the other changes caused by the file creation.

(e) (8 points) What would be the hazard in a Unix file systems of allowing hard links to directories?

(f) (10 points) Describe the disk I/O (e.g. what file system blocks would be read or written and in what order) that would be performed to complete the following system calls:

```
int fd = open("/foo",0);   // Assume that the file exists and is more than 512 bytes
                           // The second argument specifies read-only.
read(fd,buf,512);
```

You may assume a BSD Unix file system implementation and an empty file cache.

2. Protection and security (49 points)

(a) (12 points) For each of the following system descriptions, say if you think the system would be better classified as a capability-based system or as an access control list-system. Be sure to justify your answer.

(1) Each file contains a list of valid passwords. A user must specify one of the passwords from the list to the open system call in order to access the file.

(2) All files are given names containing 256 randomly generated characters. The only way to access a file its to know its name.

(3) The administrator sets the list of users that cannot access a file. This is done on a per file basis.

(b) (9 points) Describe how a buffer overflow attack works.

(c) (9 points) Logging has been used to solve problems in both file systems and security. Describe the similarities and the differences in how logging is used in each area.

(d) (10 points) Describe a way that:
(1) You can authenticate yourself to a computer system.

(2) The computer system can authenticate itself to you.

(e) (9 points) Why would you ever want to publish a detailed description of the workings of your computer's protection system?

3. Networking (34 points)

(a) (8 points) Is it possible for a TCP connection to have fewer ACK packets than data packets? If so, explain how. If not, why?

(b) (8 points) What is “bit stuffing” and how is it used in a link-level protocol?

(c) (9 points) Explain how an Ethernet decides which machine goes next after a collision.

(d) (9 points) Explain why a packet sent between two machines on the Internet using the IP protocol might arrive in multiple pieces spread across multiple packets.

4. Miscellaneous (28 points)

(a) (9 points) Given a virtual memory system that always gives equal portions of the physical memory to each process, would the system necessitate a global or local replacement policy? Justify your answer.

(b) (10 points) Describe how the following virtual memory implementation algorithms decide when to give and when to take away memory from a process:
(1) Working set algorithm

(2) Page fault frequency algorithm.

(c) (9 points) Explain how a RAID5 disk subsystem might generate many disk read requests even though the OS might only ever write data to the disk device.