Stanford University
Computer Science Department
CS 240 Quiz 2
Winter 2003

March 6, 2003

This is an open-book exam. You have 50 minutes to answer eight out of ten questions. 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.)

Question	Score
1 - 2	
3 - 4	
5 - 6	
7 - 8	
9 - 10	
total	

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 **8 of the following 10 questions** (i.e., skip two) and, in a sentence or two, say *why* your answer holds. (5 points each).

1. Your 240 study partner, Jimbo, enthusiastically states that since the end-to-end argument says to push functionality out to the ends, each application should implement their own TCP. Is that so or should he fail 240? Please give your reasoning, but not more than 4-5 sentences of it.

2. Since the original publication of Google's mechanisms, the Google guys have added (1) a "negation" operator that discards all pages that have a match and (2) join operator that forces two words to be adjacent. Consider the query: "vice-president -gore" (i.e., "vice president" must be adjacent and "gore" cannot appear in the result). List the steps needed to perform this query, indicating the data structures involved and how they are used.

3. Ethernet guarantees only best-effort, i.e., not 100% reliable, transmission. Please give two errors that result in packet loss that it cannot prevent.

4. Given the following pseudo-code:

```
// runs when there is a network interrupt.
net_interrupt() {
        while eth1 has packets on receive Q
                // process packets;
        while eth2 has packets on receive Q
                // process packets;
        while eth1 has packets to send
                // send packet
        while eth2 has packets to send
                // send packet
}
```

List several problems with this code from a livelock point of view and suggest how to rewrite it to eliminate these problems.

5. Flash uses the `mincore()` system call to determine if a page is currently in memory. Jimbo knows that system calls are expensive, so he replaces it with a library call that does nothing other than return "true" when called. How would you expect the "Flash" line in Figure 9 to change?

6. LFS grabs several segments and starts cleaning them. One of the segments contains a block that is at the end of a very large file (i.e., it is reached by indexing off of a triple indirect block). In the worst case, how many other segments will the cleaner have to touch in order to move this block to a new location? Besides extra disk accesses what problem can this cause?

7. LBFS makes a big deal out of saving bandwidth while providing traditional file system semantics. Give two ways you could trade weaker (reasonable) semantics for reduced bandwidth.

8. Describe a update pattern in which EFS would incur significant space overhead for its directory metadata. How could EFS's directory representation be changed to accommodate such a pattern and reduce the space overhead?

9. Because NFS is stateless it relies on requests being *idempotent* on the persistent file system state (i.e., you can repeat them over and over with the same effect: the statement "x = 4" is an example of an idempotent operation). Which NFS operations are not, in fact, idempotent? Give one example of how this could cause different behavior than if you were running on a local file system.

10. As discussed in class, RAID0 ("striping"), simply stripes data across disks in the same manner as levels 4 and 5, but without adding any redundancy. Compare the performance of RAID0 with RAID5 for small reads, writes, and r-m-w's.