CPS 210 Final Exam

Fall 1996

Instructions. Answer each question with a terse statement, providing as much detail as you can in the time available. Answers will be graded for substance and clarity. Redundancy and vague generalities are considered bad form. The first seven questions are equally weighted at 10 points each. The last question is worth 20 points. You receive a holiday gift of 10 points just by signing your name on your exam paper. Total points: 100.

This is a closed book examination. You have 2.5 hours.

1. Object-oriented concepts are often useful in kernel design. In this course, we discussed several instances of "generic" internal kernel interfaces that can have multiple implementations at runtime, selected by indirect calls through a vector of procedure pointers associated with some kernel object. Discuss three real examples of this technique in use. For each example, briefly outline the nature and purpose of (1) the kernel interface that was made generic, and (2) two specific implementations of the generic interface.

2. List three broad design goals of the Mach virtual memory system, and briefly outline how the design meets those goals.

3. Larry McVoy says that "file system design is 99% block allocation". Do you agree or disagree with this statement? Explain why the block allocation strategy often determines the performance of file read operations.

4. Margo Seltzer's experiments with the Fast File System (FFS) and the log-structured file system (LFS) showed that FFS is faster at sequentially overwriting large files than it is at sequentially writing new files of the same size. Why does this happen in FFS? Does it also happen in LFS? Why or why not?

5. What happens when an Network File System (NFS) server fails and restarts? How does the failure appear to the client process? What actions does the client take to recover from the failure? What actions does the server take to recover from the failure?

6. Earlier today I typed "ls" to an old shell, and a message appeared that said: ". not found: Stale NFS file handle". What does this message mean? Outline a sequence of events that could have caused this error to occur. Why can't NFS prevent this problem?

7. What is global memory management? Why might a user want global memory management? Why might a user want to avoid it even if it works properly? What assumptions does global memory management make about the network? What assumptions does it make about the hosts connected to the network?

8. What does the kernel do when a process terminates? A Unix process can terminate by invoking the *exit* system call. Exit system call handling is long and involved and affects most major kernel data structures. List the actions the kernel must take on *exit* in as much detail as you can. (20 points)