Final Study Guide

This is simply a guide of topics that I consider important for the final. I don't promise to ask you about them all, or about any of these in particular; but I may very well ask you about any of these, as well as anything we discussed in class, in discussion section, or that is in the reading.

1. *Anything from the midterm study guide*
2. Memory management
   a. Bare machine, resident monitor, fence addresses, and fence registers
   b. Relocation and address binding
   c. Bounds registers, base and limit registers
   d. Internal vs. external fragmentation
   e. Compaction
   f. Paging and page tables
   g. Segmentation and segment tables
   h. Optimizations: cache, hit ratio, effective memory access time
   i. Views of memory: program vs. operating system, address translation
   j. Protection
   k. Segmented paging (segment the page table)
   l. Paged segmentation (page the segments)
   m. Virtual memory: demand paging, page faults, pure demand paging
   n. Page replacement and victims and dirty bits: FIFO, OPT, LRU, stack algorithms
   o. Minimum number of pages per process
   p. Global vs. local allocation
   q. Working set: thrashing, principle of locality, working set principle and model
   r. Prepaging, I/O interlock, choosing page size, restructuring program
3. File Systems
   a. Virtual vs. physical; names; directory structures
   b. Access control: rights, ACLs, UNIX abbreviations
   c. Access via create, open, close, read, write, rewind, delete system calls or commands
   d. Access methods: sequential, direct mapped, structured
   e. Disk directory: free list implementations, allocation methods (contiguous, linked, indexed)
4. Deadlock
   a. Resource manager, request, release
   b. What is deadlock; difference between it and starvation
   c. Resource types: reusable, consumable
   d. How to deal with deadlock: ignore, detection and recovery, prevention (mutual exclusion, no preemption, circular wait, hold and wait), avoidance
   e. Deadlock recovery: breaking circular wait, break no preemption (i.e., allow preemption)
   f. Deadlock prevention: hierarchical ordering (ordered resource) policy, acquire all resources before running
   g. Deadlock avoidance: bankers algorithm
5. Computer Security
   a. Confidentiality, integrity, availability
   b. Policy vs. mechanism
   c. Saltzer's and Schroeder's Design Principles
   d. Access control: subjects, objects, access control matrix, ACLs, C-Lists, protection rings
   e. Cryptographic attacks: ciphertext only, known plaintext, chosen plaintext
   f. Classical ciphers: Cæsar cipher, Enigma, DES and AES
   g. Public key ciphers: requirements, RSA cryptosystem
   h. Cryptographic hashes
   i. Verification of user identity; the UNIX/MINIX method