1 Review for Mid-term Exam 3

Most frequently asked question about any exam: *How many questions are there on the exam?*

3 (on this exam)

Other facts about the organization:

- Exam takes 50 minutes.
- No written references or calculators are allowed.
- To receive any partial credit, show your work!

2 Things to Study

(an incomplete list)

1. Caches
   - Principles of locality, associativity, least recently used (LRU).
   - Basic cache design, multi-level, write-back, etc.
   - Computing the size of block offset, index and tag fields.
   - Computing the size of a cache (amount of data it can hold) and the number of bits to implement it.
   - Calculating cache performance and AMAT calculations.
   - Determining cache performance given patterns of usage.
2. Virtual Memory
   - What is indirection? (Can you give non-VM examples?)
   - How does VM enable: processes larger than memory? processes to use the same addresses? isolation of one process from another? controlled sharing between processes?
   - What is the difference between a virtual address and a physical address?
   - What is a page table? What is a translation look-aside buffer (TLB)? (How are they accessed? what do they contain?)
   - How does a hierarchical page table work? Why are they used?
   - What is a page fault? How does the cost of a page fault affect the design of virtual memory? (e.g., page size, associativity).
3. Hard Disks
   - What are platters, heads, tracks, and sectors?
   - The four components of an access: seek time, rotational delay, transfer time, and overhead (and how to compute them).
   - How random read/write performance differs from the access time of sequential sectors on the same track?

4. Error Correcting Codes
    - Error detection vs. Error correction.
    - Parity: how it is computed.
    - Hamming distance: what it is and what degree of protection/correction a given Hamming distance enables?
    - SECDED.

3 Problems

1. (a) A program repeatedly performs a three-step process: It reads in a 4-KB block of data from disk, does some processing on that data, and then writes out the result as another 4-KB block elsewhere on the disk. Each block is laid out using contiguous sectors and randomly located on a single track on the disk. Each sector is 512 bytes. The disk rotates at 7200 RPM, has an average seek time of 8 ms, and has a transfer rate of 20 MB/sec. the controller overhead is 2 ms. The processing step takes 20 million clock cycles, and the clock rate is 400 MHz. What is the overall speed of the system in blocks processed per second?

   (b) Now suppose the program reads in a 400-KB block of data that is heavily fragmented and writes back one contiguous block of data to disk. The processing step now takes 2 billion clock cycles. What is the overall speed of the system in blocks processed per second?