Computer Science Department
Stanford University
Comprehensive Examination in Software Systems
Autumn 1991

October 11, 1991

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. Be sure you have all the pages of this exam. There are 2 pages.

3. The number of POINTS for each problem indicates how elaborate an answer is expected. For example, an essay-type question worth 6 points or less doesn't deserve an extremely detailed answer, even though a person can expound at length on just about any topic in computer science.

4. The total number of points is 60, and the exam takes 60 minutes. This "coincidence" can help you plan your time.

5. This exam is OPEN BOOK. You may use notes, articles, or books—but no help from other sentient agents such as other humans or robots.

6. Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out; you can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when it is actually false. But no partial credit can be given if you write nothing.

Comprehensive Exam: Software Systems (60 points) **Autumn 1991**

Note: If you are convinced you need to make an assumption to answer the question, state your assumption(s) as well as the answer.

1. (20 points total) *Concurrency*

   (a) (7 points) Bulgarian Computer & Communication Inc. (BCCI) has just produced a multiprocessor machine which lacks any type of indivisible read-modify-write instructions. John Meredith, a well-known computer consultant, has claimed that the whole thing has to be scrapped (at the cost of much wasted time and money) because of the above omission. You are called as a second opinion. Describe what you would recommend and why, both for this machine and what they do in the future, including any issues with scale and applications.

   (b) (7 points) Garn Smidley, a young, somewhat overconfident programmer, claims that he needs no synchronization in the sense of locks, semaphores, etc. in the concurrent program he is writing for you because there is only a single writer/update process for each shared data structure — the other processes just read/query the shared data structures. Give an example of how Smidley can still get himself in trouble if not careful, and an example of a non-trivial type of trick he might be using to make concurrent operations correct.

   (c) (6 points) You have the pleasure of meeting Paula Abdul at a party, and discover to your surprise that she has been studying concurrent programming between hit records! There is one thing she is confused on (which she proceeds to ask you). "How can signaling on a condition variable inside a monitor avoid resuming the waiting process inside the monitor, so there are multiple processes in the monitor (the signaler and the signaled), while still assuring the signaled process that the condition is true when it enters the monitor". she asks, explaining to the roadies that you are in the Ph.D. program at Stanford. As all look on in awe, what do you say?

2. (20 points total) **Virtual Memory** Consider the simulation of a large wind tunnel on a uniprocessor machine with 100 Megabytes of memory. (For the purposes of this question, assume all the memory is available for application data, and the code size is negligible.) The simulation works by scanning and updating the entire wind tunnel state, represented as K megabytes of data, on each time step. Suppose that a page fault costs 10 milliseconds, with a page size of 4 kilobytes, 8.3. milliseconds rotational latency and 1.7 milliseconds transfer time (for round numbers). A timestep takes approximately 1 second of CPU time for each 10 megabytes of wind tunnel without page faults.

   (a) (7 points) Calculate an estimate of the elapsed time per timestep with 130 megabytes of windtunnel, assuming a simple LRU scheme being used in the virtual memory system. Describe any factors/assumptions that might affect your estimate.

(b) (7 points) Suppose your job was to speed up this program (without getting more hardware), and that the operating system provided virtual memory controls to prepage and flush out portions of the address space. How would you (re)structure the program to take advantage of these facilities, and what sort of performance might you expect when done? Assume each prepage and flush operation cost 1 millisecond of CPU time to start/complete I/O transfers, etc.

(c) (6 points) Describe all the additional (real world) factors that might further limit the performance to be below what seems reasonable from the basic numbers above.

3. (20 points total) **File Systems** Himmel Smackly, a new Masters student, claims that file systems are totally irrelevant for the future because, with 64-bit addressing in processors, we can finally have true single-level store systems, and keep all "objects", including files in the virtual memory system. Your roommate laughs heartily at this, knowing that you missed some great parties to study how file systems work for this exam, and now the technology is obsolete. Put them both in their place by describing:

(a) (7 points) the key functions of file systems.

(b) (9 points) why the basic modules and techniques won't disappear just because of 64-bit addressing.

(c) (4 points) why single-level store may not be the great salvation even for users, applications, etc. that Smackly claims, even if every machine converted to 64-bit addressing tomorrow.

(Note: you may answer this question as three separate parts, or as one combined answer covering all at once.)