

Systems Comprehensive Exam, Fall 2003

Tuesday, August 19, 2003

1 Instructions

This is a closed-book, closed-notes exam with a total of 75 points. You may not use any external source for answering these questions. Please direct any questions about this exam to Professor Bridges, who may be reached either in person in his office in 345G Farris, by phone at 277-3032, or by email at bridges@cs.unm.edu. Turn your exam in to Professor Bridges by 5:00 PM MDT on Tuesday, August 19, 2003. Exams *will not* be accepted after this time except by prior arrangement with Professor Bridges.

Type or write your answers to the stated number of questions in each of the following three sections. Make any *reasonable* assumptions necessary to answer the question, but be sure to state any assumptions that you make.

2 Short Answer - Answer 3 of 4 (30 points)

Briefly answer 3 of the following 4 questions. Your answer to each question should be no longer than *one* paragraph.

1. Consider a machine with a standard 5-stage pipeline (perhaps the DLX pipeline described by Hennessy and Patterson—Fetch, Decode, Execute, Memory, and Writeback). Suppose you split every stage into 2 stages which took half as long to run and as a result doubled the clock speed of the processor. Explain why programs would not necessarily run twice as fast on this processor despite the doubled clock frequency.
2. Consider the inner-most backward branch in a set of nested loops, where the program exits the loop when that branch is not taken. Why will a two-bit branch predictor generally perform better in this case than a one-bit predictor?
3. In the context of RPC, what is marshalling and how is it done?
4. In a transport protocol, what is the difference between congestion control and flow control?

3 Medium Answer - Answer 2 of 3 (20 points)

Provided detailed answers to two of the following three questions. Be sure to state any assumptions you make and to fully justify your answers.

3.1 Priority Inversion

Priority inversion is an important problem in scheduling and synchronization. Define what priority inversion is, and carefully illustrate how it could occur in a specific situation. How is priority inversion usually prevented?

3.2 Consistency

A variety of different consistency models have been defined for use in distributed systems, the most important of which are *sequential consistency* and various forms of *weak consistency*.

- (a) Define sequential consistency. Pick and describe one specific weak consistency model.
- (b) Construct and show an example in which sequential consistency and your chosen form of weak consistency will return different results.

3.3 Caching in Distributed File Systems

Different distributed file systems cache data at different granularities. For example, the Andrew File System (AFS) caches whole files, while the Sprite file system and most versions of NFS cache individual file blocks.

- (a) What are the advantages and disadvantages of each approach?
- (b) How does the choice of a caching mechanism relate to options for maintaining consistency and coherency in the file system?
- (c) How suitable is of each approach to dealing with disconnected operation (i.e., the access of cached data when disconnected from the network)?

4 Design - Answer 1 of 2 (25 points)

Provide a *detailed* answer to one of the following two questions. Be sure to state any assumptions you make and to fully justify your answer.

4.1 System I/O Hardware and Software

The company you work for is designing a new computer system design and heavily customizing the operating system to take advantage of this system. Two typed of non-volatile media appropriate for secondary storage are being considered for use in the system's file system: a high-latency, high-bandwidth medium (e.g. modern disk drives) and a low-latency, low bandwidth medium (similar to e.g., compact flash memory). Both formats have the same price/megabyte. How would you customize the operating system to best take advantage of these types of storage, and approximately how much each type of memory would you suggest be included in low-end and high-end versions system?

4.2 Stupid Paging Tricks

Copy-on-write is frequently used to optimize performance of data passing and sharing between multiple processes. (Recall that in a copy-on-write system, "copied" data is actually mapped read-only into multiple processes, and an actual copy is made for a process only when it attempts to write to the data.)

- (a) Assuming a two-level page table, show the actions necessary for providing copy-on-write sharing for one page of data between two processes.
- (b) In some cases, immediately copying the data may be quicker than remapping the data. Which factors should you consider in determining which approach is better, and how do each of these factors effect the performance of each approach?