Systems Comprehensive Exam, Spring 2006

August 16, 2006

1 Instructions

This is a closed-book, closed-notes exam with a total of 100 points. You may not use any external source for answering these questions, including but not limited to the Internet, books, notes, or othee people. Please direct any questions about this exam to Professor Bridges. Professor Bridges may be reached either in person in his office in 301B Farris, by phone at 277-3032 or 363-8798, or by email at `bridges@cs.unm.edu`. Turn your exam in to Professor Bridges or the front office by 5:00 PM MDT on Wednesday, August 16, 2006. 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 should be no longer than *one* paragraph.

1. **Storage Management.** The UNIX fast file system, and most file systems, stores files as a list (or tree) of blocks. What problem does this present particularly for small files and how does the Berkeley Fast File System attempt to address this?

2. **Caches and Multiprocessors.** What are the main advantages and disadvantages of write-though vs. write-back caches on mulitprocessor systems?

3. **Computer Networks.** When might a layer-3 (network-layer) switch, a.k.a. a router, be preferable to a layer-2 (MAC-level) switch, a.k.a. a bridge, and why?

4. **Memory Mangement.** Assuming a standard two-level hierarchical page table, when a page is marked copy-on-write and then subsequently written to and copied, what page table manipulations are involved in the page tables of the source and destination processes?

3 Medium Answer - Answer 2 of 3 (40 points)

Provide detailed answers to two of the following three questions. Be sure to state any assumptions you make and to fully justify your answers. Limit your answers to approximately one to two pages in length.

1. **Routing.** The *count-to-infinity* problem is a well-known problem in distance-vector (i.e. Bellman-Ford) routing. Define the count-to-infinity problem, provide an example where the count-to-infinity problem occurs, and briefly mention at least one possible workaround to this problem.

2. **Distributed Clocks.** Lamport timestamps are a common method for partially ordering operations in a distributed system based on the *happens-before* relation. Describe Lamport's technique for implementing logical clocks using timestamps, where every operation $a, b, \ldots$ is given a logical clock time $C(a), C(b), ...$

3. **Message Passing and Scheduling.** Older message-passing systems, for example in the original Mach kernel, had synchronous RPC where a sender sent an RPC request to a waiting receiver thread that processed incoming messages. In contrast, most modern message passing systems, for example in the K42 and L4 operating systems, have caller threads in a synchronous RPC directly execute the called funciton in the memory protection context of the called server for efficiency reasons. This optimization is sometimes referred to as "migrating threads".

   (a) Assuming short messages that can fit in registers, what steps are involved in the complete Mach-style synchronous RPC, and which steps can be removed in the optimized migrating threads version?

   (b) One difficulty with a migrating threads model of RPC is that the thread usually runs entirely in the resource allocation context of the caller. With multiple callers to a single service, show how this can lead to priority inversion.

4 Design - Answer 1 of 2 (30 points)

Provide a *full* and *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 Device Support for Virtualization

Hardware device virtualization is one of the most difficult issues in virtualization systems like Xen and VMWare. Because most commodity devices, for example Ethernet cards and disk controllers, have no concept of virtulized operating systems, the hypervisor (virtualization) layer must generally be involved with every device request to isolate and allocate resources between guest operating systems. This can greatly slow system device performance and increase virtualization overhead, unfortunately, particularly on high-speed devices.

For two different devices, namely a high-speed ethernet network device and a hard-disk controller, design and describe hardware extensions that will:

- Allow guest operating systems to set up virtualized sessions with each device using virtulization layer and,
- After these sessions are established, allow guest OSes to use their share of the device without per-operation intervention from the hypervisor.

Be sure to describe the interface between the device and the hypervisor, the guest OS and the hypervisor, and the guest OS and the device for supporting this functionality.

4.2 Evaluating the Log-Structured File System

Recall that the Log-Structured File System (LFS) organizes disk data structures as a logically-infinte log, that the system buffer cache groups writes together into large contiguous write, and that a user-level program called the *cleaner* is responsible for discovering free space to extend the log. In the mid-90s, there was considerable controversy in the fairness of comparisons between LFS and newer extensions to the Berkeley Fast File System (FFS) that optimize large writes by grouping sequential disk blocks into large *extents*. By writing entire extents at a time, this extent-based file system (EFS) optimized large writes while maintaining standard FFS-like on-disk structures.

Design and describe experiments that will highlight the strengths and weaknesses and quantify the tradeoffs made by both EFS and LFS so that system analysts can choose the correct file system for a system. Be sure to specify all relevant metrics that you propose to collect, and the results that you expect in each experiment on each metric.