UNIVERSITY OF WISCONSIN Computer Sciences Department

Operating Systems Qualifying Exam **Spring 2013**

Instructions: There are *six* questions on this exam; answer *all six* questions.

Question 1. Virtual Machines and Performance:

Consider a virtual machine monitor that runs on the bare hardware and is able to support multiple partitions, each running a different operating system.

When you run your program on an operating system on a virtual machine, you notice that it runs noticeably slower than it would on an operating system running directly on the bare hardware.

A. Describe three possible reasons why your program might run more slowly than when running on the bare machine.
B. For each reason, describe a possible design change to the virtual machine software (and possibly the operating system or hardware) that might improve the performance.

Question 2. Mapping Tables:

Operating systems are full of tables that map one address space to another. This question explores the similarities and differences between various OS mappings.

A. One such mapping table is found in the virtual memory (VM) system and is known as a page table. Describe a page table, including how it is used, and its common contents.
B. Another such mapping structure is found in the file system (FS) and is known as an inode. Describe an inode, including how it is used, and its common contents.
C. One type of page table is known as a multi-level page table, and is commonly found in modern systems. What is a multi-level page table, and why is it useful?
D. Would this type of multi-level structure be useful in an inode?
E. One type of structure found in modern inodes are extents. What is an extent, and why is it useful?
F. Would extents be useful in a page table?

Question 3. MapReduce:

A. Briefly explain the components of a MapReduce program and how a MapReduce program is executed in a computer cluster.
B. How are failures handled in a MapReduce system? Please separately discuss failures of different system components.
C. Parallel computation is often non-deterministic. That is, given a particular input, different outputs might be generated in different runs. Assuming that the map and reduce functions are internally deterministic, is a MapReduce program overall deterministic (supposing there are no failures)? Please explain your answer.

Question 4. Logical Time:

A. In a distributed system running on different physical machines, we often want to compare the execution order among events that occur on different machines. This comparison is usually **not** based on the physical clock maintained by each machine. Why?
B. Present a logical clock scheme that a distributed system can use to determine the causal order among events. Here, the causal order should reflect message sending-receiving order (i.e., no message can be received before it is sent) and the execution order within each machine.
C. How can the logical clock designed above help detect data races in multi-threaded software? Please define "data race" first and redefine your logical clock scheme for multi-threaded software, if needed.

Question 5. Synchronizing at Large Scale:

The barrier is a type of synchronization operation that requires all processes to reach the barrier call before any of them can proceed. It is key to support a wide variety of supercomputer applications.

For example, in the simplest case, each process in a parallel program calls: `barrier();` and will then block until all other processes all make a similar call. After all the processes reach the barrier, then they all can continue.

Barriers are frequently used in programs that run on large scale supercomputers, with 100,000's (or even millions) of processes. These supercomputers use message passing as the communication mechanism between nodes, supported by high speed, low latency networking. So the implementation of a barrier is built on top of message passing.

To be efficient, you want the barrier mechanism to recognize as quickly as possible that the program reached the barrier; similarly, you want to quickly notify all processes that they can continue execution after they all reach the barrier. In the context of this question, don't concern yourself with failures.

A. What performance measures would you use to evaluate barrier protocol. Explain why these measures are meaningful.
B. Describe an efficient and scalable algorithm for implementing the barrier protocol.
C. Evaluate your protocol in terms of the measures you described in part B.

Question 6. Operating System Organizations:

Historically, there has been a wide range of operating system designs, such as layered, monolithic, and microkernel designs.

A. List and describe three of the most important goals/considerations in designing an operating system.
B. For three operating systems from the list below, explain what goals/considerations the designers had and why.
   i. Nucleus
   ii. Unix
   iii. Hydra
   iv. Pilot
   v. Exokernel
C. Explain how a modern operating system such as Windows, Linux or Mac OS, addresses your goals/considerations.
