UNIVERSITY OF WISCONSIN
Computer Sciences Department

Operating Systems
Qualifying Exam

Fall 2010

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

Question 1. The Log-Structured File System:

A. In LFS, all data and metadata are grouped into chunks (called "segments") and then written to disk (at some point) to a free spot on disk. How big should a segment be in order to ensure high performance? (explain/justify)
B. One major difficulty of such an approach is that LFS makes it harder to find items on disk (e.g., inodes), as they could have been written anywhere on disk. What additional structures are needed to allow LFS to find the location of an inode?
C. Once the location of an inode is known, is there any difference between LFS and more traditional file systems (such as FFS) in terms of read performance?
D. An oft-cited problem with LFS is the need for garbage collection, which is performed by the LFS cleaner. Describe how the cleaner works. How does it find garbage and make segments available for future use?

Question 2. Memory Management:

Both virtual machine monitors and operating systems need to worry about memory management and paging. However, there are fundamental differences between the two.

A. Describe the mechanism VMware server uses to decide (i) how much memory to allocate to a virtual machine, and (ii) which pages to reclaim when removing pages from a virtual machine.
B. Explain why operating system policies, such as WS-clock, are inappropriate for use by virtual machine monitors.
C. A key benefit of virtual machines is the ability to run unmodified operating systems. Most operating systems now support the use of superpages (larger than normal pages).
    1. What is the benefit of using superpages over normal size pages for an operating system? What are the costs?
    2. Explain what changes are needed to a VMware's management of virtual memory to support guests that use superpages.

Question 3. Authentication:

Suppose that your department or group is using Kerberos for authenticated access to services.

A. Initial access: What do you have to do to be able to use Kerberos the first time. In other words, how do you initially authenticate yourself to Kerberos?
B. Revocation: How would the system administrator revoke access for a particular user to a particular set of services? Describe two ways that you might implement such revocation, comparing their efficiency and ability to cause a timely revocation of access.
C. Multiple administrative domains: Two computer science departments, D_1 and D_2, at different universities are using Kerberos to provide access to their services. Each department has its own list of users and services, and controls access to its own services.

   We would like to be able to configure Kerberos so that a user *U* in one department can log on (authenticated itself) once to Kerberos and then access services in both department. What characteristics of Kerberos' design make this difficult? How could you change Kerberos to make this simpler?

Question 4. Software Dependability:

A. "Reliability" and "availability" are slightly different concepts. Define each term and explain the difference between the two.
B. Explain one technique used by NFS that improves system availability but not system reliability.
C. Process pairs are a technique to improve system availability. They typically include two processes executing the same program and maintaining approximately the same state on two different machines. Some people suggest the use of virtual machines to host a process pair on one physical machine in order to save the code of hardware. How and when does this affect the resulting system's availability?

Question 5. Distributed Storage:

Your goal is to design a system that stores and provides access to tens of millions of files submitted by users of the distributed storage system. The files are going to be stored on hundreds of servers scattered around the planet. The purpose of distributing the files is spread out load, eliminate single points of failure that could take down the whole system, and increase availability of the data.

For this question, we will not consider authentication or access control.

A. The first issue to resolve is naming. A simple hierarchical name space is a problem in this contexct, as it does not help with helping users find files or helping avoid name conflicts. Propose a file identification scheme that would allow users to provide intuitive names for files, while allowing easy search for the files and ability to disambiguate between similarly named files from different users.
B. The next issue to resolve is the index data structure that allow users to find copies of a particular file. Propose a design for a distributed index structure that allows users to quickly add, remove, and find files.
C. Replication of this index will be key to performance and availability. What is your consistency model for replicated index data and how will you handle additions and removals from the index?

Question 6. About Operating Systems:

Imagine that on your flight to OSDI your ticket is upgraded to first class and you find yourself sitting next to a well-dressed VIP (very important person) whose job it is to advise some high-ranking government official (they won't say who). After you tell the VIP that you are on your way to a conference about Operating System Design and Implementation, they are in the mood to talk. They ask you the following questions, which you do your best to answer using terms and language that are understandable to the non-technical (but still smart!) VIP.

A. What is an operating system and what does it usually do? Why are operating systems important?
B. Has any research in **(single-node) operating systems** helped out any businesses or people in their everyday lives? If so, give an example and explain the basic conceptual ideas.
C. Has any research in **distributed systems** helped out any businesses or people in their everyday lives? If so, give an example and explain the basic conceptual ideas.
D. Is systems research complete as this point? If not, what are some of the most important problems that remain to be solved?
