UNIVERSITY OF WISCONSIN—MADISON
Computer Sciences Department

Operating Systems **Spring 2017**

Instructions: There are six questions on this exam. You must answer ***all*** of them and explain your answers.

Question #1: Distributed File Systems

NFSv2 and AFS are two classic distributed file systems that made very different design decisions. This question examines two of those decisions: how each distributed file system tracks state on the server and whether clients cache whole files or individual blocks.

A. Server State.
    1. What state does an NFS server and an AFS server track?
    2. What are the advantages and disadvantages of each of these two approaches?
B. Whole files vs. individual blocks.
    1. How does client caching of whole files versus individual blocks impact the design of the two file systems?
    2. What are the advantages and disadvantages of each approach?

Question #2. Digital Signatures

Needham and Schroeder describe protocols for digitally signing a message, using both public key and secret key encryption algorithms. The goal of such a protocol is for **A** to send **B** such that only **B** can read the message and that only **A** could have generated it.

A. Describe how such a digital signature would work when using public key encryption.
B. Describe how such a digital signature would work when using secret key encryption. For this solution, you may need an additional participant in the protocol, beyond A and B. What would be the role of this additional participant?
C. Suppose that A wanted to repudiate their signature. How would A do this for each of the two cases above?

Question #3: Operating System Design

Operating system advances are often driven by advances in other hardware or software technology.

A. **Past trends:** For these four systems, please explain what hardware trends motivated the work:
    1. Disco
    2. Remote Procedure Calls
    3. Google File System
    4. Scheduler Activations

B. **Future trends:** New memory technologies are being proposed that may be much cheaper than DRAM. Suppose these technologies lead to systems with 1000 times more memory than today's largest systems.

    Explain how operating system designs could change? For example, what components would stay the same, change, or be removed, and why?

Question #4: Log-structured File Systems

LFS is a log-structured file system that writes all modifications to persistent storage sequentially in a log-like structure.

A. Compared to the approach taken by FFS-based file systems, why is writing to a log advantageous for hard disk drives and RAIDs?
B. LFS writes many of its relevant data structures in a log-like fashion.
    1. What data structures does LFS require that FFS-like file systems do not? Why? What is the role of each of these data structures?
    2. Which of these data structures cannot be written only to the log? Why not? How are these data structures handled?
C. Compared to FFS-based file systems, why is using a log advantageous for crash recovery?
D. Assume a crash has occurred.
    1. What operations must be performed in LFS for recovery?
    2. How does this differ from the operations performed by FFS?
    3. What are the trade-offs in the design of LFS pertaining to crash recovery?

Question #5: Isolation

Isolation in computer systems seeks to ensure that the isolated entities cannot interfere with each other, where interference may be directly through changing data or control flow. This can also happen through performance, when one entity affects the performance of another. However, providing performance isolation often conflicts with achieving high efficiency or performance.

A. How does VMware ESX Server's memory management system balance performance isolation against high performance?
B. If you wanted to make its isolation stronger, at the cost of performance, what could you do?
C. If you wanted to make it performance or efficiency higher at the cost of isolation, what would you do?
D. If you were a malicious virtual machine and wanted to have the maximum performance impact on other virtual machines, what would you do?

Question G: MapReduce

MapReduce (MR) is a parallel programming environment for batch-style large data processing. In this question, we'll investigate MR and some of the assumptions it makes in its design.

A. One key phase of MR is called "shuffle".
    1. Describe what shuffle is and how it works.
    2. In what situations can the shuffle cause performance problems?
B. Is shuffle always needed? Describe the types of applications that could work well even without a shuffle phase.
C. One problem with MR computations is the "straggler" problem. Describe the straggler problem and how it can negatively harm MR computations.
D. The paper suggests one type of solution to the straggler problem known as "backup tasks".
    1. What is a backup task?
    2. When does it work well?
    3. In what environment or configuration would backup tasks actually make things worse?