SPRING 1998
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON
PH. D. QUALIFYING EXAMINATION

Operating Systems
Qualifying Examination

Monday, February 2, 1998
3:00 – 7:00 PM
1263 Computer Sciences and Statistics

GENERAL INSTRUCTIONS:

1. Answer each question in a separate book.
2. Indicate on the cover of *each* book the area of the exam, your code number, and the question answered in that book. On *one* of your books list the numbers of *all* the questions answered. *Do not write your name on any answer book.*
3. Return all answer books in the folder provided. Additional answer books are available if needed.

SPECIFIC INSTRUCTIONS:

Answer any *six* of the *seven* questions. Before beginning to answer a question, be sure to read it carefully. If you are confused about what the question means, state any assumptions you have made in formulating your answer. Good Luck!

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tires to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the *first hour* of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is nontrivial.

Operating Systems Exam

Instructions:
Answer any six of the following seven questions.

Question 1. Distributed File Systems:

The original AFS file server had the client contact the server on each file open and close. The subsequent AFS version used "call backs."

(1a) What are "call backs" and how do they differ from the original AFS design?

(1b) Compare and contrast the two schemes discussed in part (1a). Discuss this issue from the following views: performance, file system complexity, and crash recovery.

Question 2. Process Migration:

Consider a distributed operating system that supports process migration. Given a process's ID (pid), the operating system must be able to find on which host the process is currently executing.

Describe how the distributed operating system would find processes, given a pid. Describe the internal structure (if any) of the pid, the data structures and algorithms used by the operating system.

Be sure to consider scenarios where

- the process may migrate several times,
- hosts can fail,
- the process currently has message channels to other processes, and
- messages are in transit while the process migrates.

Question 3. File Systems:

Consider the Log Structured File System (LFS) proposed by Rosenblum and Ousterhout.

(3a) Describe the basic design and principles of LFS.

(3b) What are the assumptions about trends in hardware and workloads that motivate the LFS design? Are these trends likely to make LFS more or less desirable in the next few years? Defend your claim.

Question 4. Security:

Your goal is to open a credit card service (CCS) on the Internet. Your system will have processes representing retail customers (CUST), retail stores (STORE), the customers' banks (C-BANK), and the stores' banks (S-BANK), as well as a CCS process. These processes will be running on different computers geographically dispersed around the Internet.

CUST will use the Web to make a purchase from the Web page of the STORE. Assume that a customer interacting with process CUST wants to purchase a new Fromzle from STORE for great price of $69.95.

- CUST will provide STORE with something that will authorize it to transfer $69.95 from CUST's account in C-BANK to STORE's account in S-BANK.
- Assume that CCS is a *Trusted Authority*: every party (CUST, STORE, C-BANK, and S-BANK) trusts CCS.
- Assume that none of the other parties trust each other.
- CUST must be able to authorize STORE to get exactly $69.95.
- STORE can use the authorization exactly once.
- STORE can use the authorization only at (or near) the time it was issued.
- The authorization can only be used by STORE and no other party.
- You will use public key or conventional (secret key) encryption as the underlying technology.

Describe the protocol between the various parties to implement this CCS.

Question 5. Hardware Protection:

(5a) Explain the terms *user mode* and *kernel mode* (also called *supervisor mode* or *protected mode*). How is the kernel mode different from the user mode? Are there memory addresses that can be accessed in user mode but cannot be accessed in kernel mode? What about the other way around? Give one example of a privileged instruction that can only be executed in kernel mode.

(5b) When a user program issues a system call such as **read** or **write**, it in fact generates a software interrupt. Describe what the hardware does when the interrupt happens, what the kernel does when it processes the system call, what it does after the finishing the processing, and how the CPU is switched back from the kernel mode to user mode.

(5c) How does the kernel access the parameters passed to the system call?

Question 6. Copy Avoidance:

When a Web server handles a request for a file, it typically reads the file from the file system and sends it over the network. To satisfy the request, the server may copy each block of the file several times: from the disk to the kernel's file-buffer cache, to an array in the Web-server process, to network output buffers, to the network interface.

(6a) How can the operating systems kernel avoid some of these copies? In particular, how can it avoid copying the data into and out of the Web server's address space? (Hint: consider the paging hardware). For this part of this question, assume that only the kernel *implementation* is being optimized; the kernel *interface* must remain unchanged. How can the Web server be written to help the kernel optimize these operations? What problems are caused by the need to add network protocol headers to output packets?

(6b) Can you further reduce the overhead of processing the request by establishing a new system call for the Web server to use?

(6c) Assume that the Web server can identify the set of files that are very hot (that is, being referenced very frequently). Describe various optimization techniques that can reduce the overhead of processing Web requests. You should explore both optimizations that do and do not involve changing the kernel. Also mention any hardware features that may help.

Question 7. Page Replacement:

(7a) Describe the Working Set *model* of program behavior. How is the working set of a process defined? What are the main features of any memory management algorithm inspired by the working set model?

(7b) Describe the Working Set (WS) *algorithm* for memory management. Why is the WS algorithm, as strictly defined, impractical to implement in a real system? What sort of approximations or compromises are commonly used to create "working-set-like" algorithms?

(7c) Any useful memory allocation algorithm must have an component that exercises "load control": controlling the number of active processes. Why? What is the WS policy on load control?

(7d) The WS-CLOCK algorithm of Carr and Hennessy combines features of the CLOCK (also called "Second Chance") algorithm with features of WS. One difference between WS and WS-CLOCK is that WS-CLOCK only keeps track of the resident portion of the working set. How does this fact interfere with load control? What does WS-CLOCK do to mitigate this problem?