FALL 2006
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN-MADISON
PH. D. QUALIFYING EXAMINATION
Operating Systems
Monday, September 18, 2006
3:00-7:00 PM

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 ALL six questions.

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tries 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.

UNIVERSITY OF WISCONSIN-MADISON
Computer Sciences Department

Operating Systems
Depth Exam

Fall 2006

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

Question 1. Scheduling:

Computer scientists have studied process and thread scheduling for decades. One reason is that we cannot agree on what to optimize.

A. Give **three** examples of goals for which the schedule can be optimized. For each goal, describe (i) a workload where that goal is important, and (ii) explain a scheduling algorithm that targets that goal.
B. Pick two of the three scheduling algorithms from your answer to part A, and explain how best you can integrate them into a single system. Does this achieve both goals under any circumstances? If so, when?

Question 2. Layers of Indirection:

The classic OS technique to solve any problem is to add a layer of indirection. More recently, the solution is to remove a layer of indirection. For each of the four cases listed below, give (i) an example of where this technique has been used, and (ii) explain what the layer was and how it improved the system.

A. Adding a layer and improving performance.
B. Adding a layer and improving security.
C. Adding a layer and improving reliability.
D. Removing a layer and improving performance.

For example for part A, (i) Unix added a buffer cache between the file system and the disk, (ii) which improved performance by returning data from memory instead of accessing the disk on every request. You should provide answer all four parts (A through D) above.

Question 3. Synchronization:

A. One approach for providing mutual exclusion is to rely upon a *non-preemptive scheduler*. What are the problems with using this approach to protect critical sections?
B. Another alternative for providing mutual exclusion and synchronization is to use *monitors* and *condition variables*. When Hoare introduced monitors, his definition required that a process waiting on a condition variable must run immediately when another process signals that variable, and that the signaling process runs as soon as the waiter leaves the monitor. However, when monitors and were implemented in Mesa, the developers decided to change the definition such that the waiting process

may not be scheduled immediately after it is notified. Java uses the same definition. What are the advantages and disadvantages of the Mesa/Java implementation compared to the definition by Hoare? What are the implications for application developers?

Question 4. Copy-on-Write:

Copy-on-write is a way for processes to share pages (or segments) that are logically distinct. When one process sends some data to a second process with copy semantics, nothing needs to be copied immediately, or perhaps ever.

A. Explain how standard memory-management hardware can be used to implement copy-on-write cheaply.
B. Linus Torvalds dislikes copy-on-write and claims "Once you play games with page tables, you are generally better off copying the data". Describe the extra costs of copy-on-write, and explain when it is better to just copy the data directly.
C. Describe how copy-on-write can be used to improve system performance for: (i) message passing, (ii) forking a new process, and (iii) file I/O operations.

Question 5. AFS Under Pressure:

Consider the AFS file system. This system was designed to be scalable so as to handle many clients.

A. When the designers were concerned with scale, what resources concerned them the most? Describe several aspects of the AFS design that promoted scalability.
B. What assumptions did the designer of AFS make about the client workload and access patterns to support these design decisions?
C. For each of the following workloads, describe how AFS would handle them and, if necessary, what changes might be needed in AFS to handle them efficiently.
    - Sequential read access to extremely large files.
    - Random read access to extremely large files.
    - Random read write access to extremely large files.
    - Searching ("grepping") through a directory with many small files.

Question 6. Cyber Cafe Safety:

You are in a cyber cafe, using one of their kiosk computers to read your email. Of course, you *cannot* trust the operators of the cafe, and they are likely take advantage of any information that you expose while using their computers.

A. Assuming that you are using a web-based interface to your email in the cafe, describe the security vulnerabilities that you create by using their computer. Consider vulnerabilities such as violation of privacy, unauthorized access, or loss of integrity of data.
B. Which of these vulnerabilities would be prevented if you were using an authentication scheme that uses a key card device that generates passwords that are used only once?
C. Assume that you bring your own laptop computer to the cafe and use their Internet connection to access your email. Which of these vulnerabilities would be prevented in this case? Explain. Would any new ones be created? If so, explain.