Spring 2007
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN-MADISON
PH. D. QUALIFYING EXAMINATION
Operating Systems
Monday, February 5, 2007
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.

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 **Spring 2007**

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

Question 1. Security / protection:

An OS may provide security and protection either through hardware mechanisms, such as privileged mode bits and access bits in a page table, or through software, such as runtime type checks.

1A. Explain how each of the following hardware features contributes to protection and describe how the same function can be performed purely in software.

- Address space
- Privileged mode of operation
- System calls

1B. What are the benefits of using software protection instead of hardware protection? Give a good example application where it makes sense.

1C. What are the drawbacks of using software protection? Give an example where hardware is better.

Question 2. File system interfaces:

Sixty-four bit processors are becoming common. This large address space eliminates one of the major problems with memory-mapped file access, in that you couldn't map all of a large file at once.

Given this trend, it is possible to replace the common Unix file system API with two different methods: streams for sequential access and memory mapping for random access.

2A. Explain the relative benefits of this interface compared to the standard Unix interface (`read`, `write`, `lseek`).

2B. Some large workloads request direct I/O, in which no buffering is performed. Explain how you might implement this functionality with memory-mapped I/O.

Question 3. Disk Layout:

You have a system with two primary workloads. One performs sequential access to a set of large files. The other performs small, independent, random reads to a separate set of large files. To satisfy these two workloads, you have a set of 8 disks. Both applications can saturate the peak bandwidth of your disk subsystem (i.e., they are I/O bound). Each disk can provide 100MB/s and has an average seek time of 10ms.

3A. How should you organize these disks and lay out your data to provide best performance for the sequential workload?

3B. How should you organize these disks and lay out your data to provide best performance for the random workload?

3C. If you want to run both applications simultaneously and achieve the maximum average bandwidth for each application, how should you organize these disks and data?

3D. Suppose the random reads are dependent, in that each request cannot be submitted before the previous one completes. Does this change the answer to 3C? If so, how?

Question 4. Virtual Machine Migration:

Process migration has been implemented in many operating systems and scheduling systems.

4A. Describe two applications of process migration.

4B. A key difficulty is often the identification and isolation of resources used by a process so that it can be packaged for movement. In addition, there is the challenge of providing continued access to open files, devices, and network connections after migration. Identify some of these resources and describe some difficulties in moving a process that is accessing these resources.

4C. Suppose that the unit of migration was a virtual machine partition instead of an individual process. What parts of implementing process migration would this use of virtual machines simplify and how?

Question 5. Deadlock:

5A. Define the term "deadlock". There are four conditions that must hold before deadlock is possible. Name them.

5B. Outline an algorithm that detects whether there is a deadlock. The algorithm should be able to cope with multiple resource classes, each of which has some limited number of units available.

5C. When should the algorithm be invoked? The answer to this question depends on the characteristics of the system to which it is to be applied, such as the rate of resource requests, the granularity of resources, and the expected rate of deadlock. List three possible choices and discuss the criteria you would use to choose among them.

Question 6. Synchronization:

Suppose we replace `wait()` and `signal()` operations of monitors with a single construct `waitFor(B)` where `B` is a general Boolean expression that causes the process executing it to wait until `B` becomes true.

6A. Write a monitor using this scheme to implement the readers-writers problem.

6B. Explain why, in general, this construct cannot be implemented efficiently.

6C. What restrictions need to be put on the `waitFor()` statement so that it can be implemented efficiently? (Hint: Restrict the generality of `B`).