University of Wisconsin–Madison
Department of Computer Sciences

Ph.D. Qualifying Exam: Operating Systems
Spring 2014

General Instructions

Answer each question in a separate book.

Indicate on the cover of each book the area (OS) of the exam, your code number, and the question number answered in that book. Do not write your name on any answer book.

Return all answer books in the folder provided. Additional answer books are available if needed.

Specific Instructions

Answer all 6 questions.

Before beginning to answer a question make sure that you read it carefully. If you are confused about what the question means, state any assumptions that you have made in formulating your answer. Good luck!

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 that 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.

1. Scalability

A system can usually handle more tasks, when provided with more resources. This property is often described as "scalability".

A) In a single-core machine, system throughput can often increase with the size of physical memory. Please explain the reason.

B) In multi-core systems, "scalable" often means that system throughput can increase (almost) linearly with the number of cores. Imagine a multi-threaded program with more threads than cores. First, describe how the presence of shared variables across threads impacts scalability on a multi-core machine. Second, describe how synchronization providing mutual exclusion, such as Mesa monitors, impacts scalability. What types of approaches for providing mutual exclusion will lead to better scalability?

C) In distributed systems, "scalable" often means that system throughput can increase (almost) linearly with the number of machines in the system. Please give an example of scalable design in Grapevine.

2. Chord

A distributed system often needs an algorithm to partition data across different nodes. When clients request a particular data item, a corresponding algorithm is needed to locate the node that holds that data item.

A) Explain how data is distributed and looked up using consistent hashing.

B) Describe the advantages of consistent hashing over basic hashing in data partitioning and lookup.

C) Explain how data is partitioned and looked up in Chord.

D) Describe the advantages of Chord over consistent hashing in data partitioning and lookup.

3. File System Heuristics

File systems use heuristics and approximations all the time in decision making. This question explores the usage of various heuristics in various file systems you have studied.

A) In FFS, the Fast File System, a heuristic is used to decide upon file placement. What is this file-placement policy? When does it work well, and when does it not work so well?

B) FFS also uses a heuristic when placing "large" files. What does FFS do differently for large files? Why does it treat large files differently?

C) LFS, the log structured file system, uses heuristics as well. For example, LFS buffers some amount of data before writing the data to disk. How much does it buffer (roughly)? Given a particular disk with average position time of P milliseconds and transfer rate of T megabytes/second, how much should LFS buffer before writing?

D) LFS also uses many heuristics when cleaning. Describe the heuristics LFS uses in cleaning.

4. Everything is a File

One of the great contributions of the original Unix operating system was to use the file as a unifying abstraction. While there were several places in which early Unix systems did not use the file abstraction, later operating systems, including 8th Edition Unix and ultimately Plan 9 carried this file abstraction to more areas.

Name three (3) areas where the use of the file abstraction was expanded in later systems from the original Unix (for example, in Plan 9). For each of these areas, describe the advantage to using the file abstraction and how this affected the design of the operating system.

5. MULTICS

The MULTICS operating system enabled the sharing of procedures and data in segments across multiple processes. Imagine the scenario where a process (named α) executing a procedure P wishes to reference data that is external to the procedure P; the first time the process makes this reference from P, it will be using the symbolic address for the data (in MULTICS notation, the symbolic address is <D> | [x] with the symbolic segment name <D> and the symbolic external address [x]). On the first reference to this symbolic name, the name is "made known" and translated into a generalized address (in the MULTICS notation, the generalized address is of the format d#α|x).

A) What is the motivation for beginning with a symbolic name? What is the motivation for translating the name into a generalized address? In MULTICS, why is it not acceptable to place the generalized address within procedure P?

B) In MULTICS, what supporting structure(s) are used to enable this translation and what are their properties? Draw two simple figures to illustrate P and the associated supporting structure(s); one figure should illustrate when the reference is symbolic and one when the reference is generalized.

C) How does MULTICS set up the associated supporting structure(s) when a procedure Q is first entered? Draw one figure to support your explanation.

D) What addressing support is used by MULTICS to enable these operations?

6. Memory Management in Virtual Machines

Disco and the VMware ESX server have some similarities and some differences. You will examine how these two systems handle memory management, in particular, how they identify duplicate pages.

A) Why is it important for a virtual machine monitor to identify duplicate pages? As related to this issue, what are the different assumptions made by Disco versus ESX server?

B) How does Disco handle page sharing? Specifically, describe how identical pages are found and the data structures that are used to track these pages.

C) How does the VMware ESX server handle page sharing? Specifically, describe how identical pages are found and the data structures that are used to track these pages.

D) What are the advantages and disadvantages of these two approaches?