Computer Science and Engineering Department
University of South Florida
Ph.D. Qualifiers Exam Spring 2010
Operating Systems
January 29

You have 3 hours to solve 10 problems, all equal weight. If you believe that you cannot answer a question without making some assumptions, state those assumptions in your answer.

Irrelevant verbosity will not gain you points. Clear and crisp answers will be appreciated.

This is a closed books, closed notes exam.

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. 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 not, when?

2. **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.
    a. How should you organize these disks and lay out your data to provide best performance for the sequential workload?
    b. How should you organize these disks and lay out your data to provide best performance for the random workload?
    c. 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?
    d. Suppose the random reads are dependent, in that each request cannot be submitted before the previous one completes. Does this change the answer to question c)? If so, how?

3. **Distributed file systems**:
    a. Most distributed file systems cache recently used file data in client memory. What are the performance benefits of file caching? What are the performance costs?
    b. File caching introduces the problem of cache consistency when files are shared across the network. Explain the problem and demonstrate it with an example.
    c. Outline a plausible scheme for dealing with the problem of cache consistency in a distributed file system. You may ignore the problem of failures in your answer (but see next question). Your scheme need not be identical to that used in any specific distributed file system.
    d. What are the limitations of the file caching scheme you described for previous question? Explain how your scheme can handle failures (crashes), assuming that systems fail only by stopping, discarding the contents of their memory, and restarting.

4. **Overview:** You have been given a user account on a remote machine to which you are porting some "high performance application". All you know about this machine is that it supports the POSIX interface; you have no useful documentation, no one is answering your e-mail about the machine, and any interfaces you know of for acquiring information about the machine appear to be broken.

   To obtain reasonable performance for your application, you believe that it would be useful to know the following properties about the machine or OS:

   a. Number of CPUs
   b. Page size (in bytes)
   c. Amount of physical memory (in bytes)
   d. How memory is allocated between the virtual memory system and the file cache; specifically, is a fixed amount of memory given to each or does the amount vary?

   For each property, give a brief but precise description of a benchmark program you develop that allows you to infer each property. Also, clearly state any assumptions that you are making about the machine or OS and any limitations of your benchmark.

5. **RAID**s are often used in data storage. A "software" RAID system is an in-kernel pseudo-device driver that emulates RAID functionality on top of disks that are attached to the machine. To the file system above, it appears as a single, large disk, but internally, the software RAID driver spreads disk requests across the devices. What are some of the major differences between software and hardware RAID? (what advantages and disadvantages does each type of RAID have?)

6. **IPC:** Show (using pseudocode) that mutexes and counting semaphores are equivalent.

7. **Memory:** Can the size of the working set of a process affect how much CPU time it gets and how often it gets the CPU? Support your answers with specific examples.

8. **Deadlock in Distributed Systems:** Your company is building a distributed system, and you are asked to develop a scheme for dealing with the deadlock problem.
    a. Would you use a deadlock-detection scheme or a deadlock-prevention scheme? Explain your choice.
    b. If you were to use a deadlock-prevention scheme, which one would you use? Explain your choice.
    c. If you were to use a deadlock-detection scheme, which one would you use? Explain your choice.
    d. Suppose in a distributed systems there are several resource allocators, each responsible for some set of resource classes. A client process sends each request to the appropriate allocator. (Assume there is some mechanism for clients to figure out the appropriate allocator for each request.) A check_for_deadlock request can be sent to *any* allocator. Allocators may exchange messages with each other to detect deadlock. Outline a deadlock detection algorithm in this environment. You may ignore the possibility of process or communication failures, but assume that communication between allocators is relatively expensive.

9. **Protection:** Compare file protection based on user groups (as in Unix) with access-control lists, in which individual users are designated as having specific access privileges to a file. Consider complexity of implementation, ease of use, flexibility, and security in your answer.

10. **Trends:** What challenges do multicore processors pose to operating system design? Address each relevant component of an operating system.