CS 198:416: Operating Systems Design

Sample FInal

FINAL EXAM

Prof. B. R. Badrinath

Instructions:

1. Put your name and student number on the exam books NOW!
2. The exam is closed book.
3. You have 150 minutes to complete the exam. **Be a smart exam taker**— if you get stuck on one problem go on to another problem. Also, don't waste your time giving irrelevant (or not requested) details or material.
4. Show all your work. Partial credit is possible for an answer, but only if you show the intermediate steps in obtaining the answer.
5. Good Luck!

---

1. "Quickies" (20 points, 20 minutes)

1. What is a memory mapped file? What is the benefit of using memory mapped file? What are some of its drawbacks?
2. Differentiate between capabilities and access control lists for implementing protection.
3. Differentiate between i-node and super block.
4. Explain the differences in caching policies used in NFS and AFS.
5. Differentiate between user level threads and kernel level threads.

2. "Asynchronous Concurrent Processes" (30 points, 40 minutes)

1. A barber shop consists of a waiting room with n chairs, and the barber shop has two barbers, John and Mary. John specializes in Men's haircut and Mary specializes in Women's haircut. If there are no customers, the barbers go to sleep. If a customer (male or female) enters the barber shop and all the chairs in the waiting room are occupied, then the customer leaves the shop. If a male (female) customer finds John (Mary) busy, but chairs are available then the male (female) customer sits in one of the free chairs in the waiting room. If barber John is free (asleep), the male customer wakes up

John (becomes busy). If barber Mary is free (asleep), the female customer wakes up Mary (becomes busy). Write a program to coordinate the barbers (John and Mary) and the customers using either monitors **or** semaphores. You can either have two types of customer processes (male and female) or have one customer process and assume that the type (male or female) is given as a parameter to customer process.

2. Consider a system consisting of N processes $P_1, P_2, \ldots, P_n$ each of which have a unique priority number. Write a monitor that allocates three identical printers to these processes, using the priority number for deciding the order of allocation. For example, when a printer becomes free and two or more processes are waiting, then the process with the highest priority among them should be allocated the printer. If all three printers are busy, then the requesting process is made to wait. Assume priority 1 implies highest and priority n implies lowest. Also, the printer request will be through a monitor procedure *printer-id* **acquire**(*int priority*); this procedure returns the id of the printer that has been allocated and the monitor procedure to release a printer is
release(*int printer-id*)

3. "Memory Management" (30 points, 40 minutes)

1. A CPU supports three, 32 bit segment registers and a 32-bit offset. The contents of the segment register specify the starting address of a segment. What is the maximum virtual address space seen by a process?

2. An operating system is running on a machine with 16M of physical memory. The virtual memory is implemented as a pure paging scheme. What is the size of a single level page table if the the virtual address space is 32-bits and size of a page is 4K. What will be the size of an inverted page table for this system. Assume that each page table entry takes one byte.

3. How can sharing be implemented on a system that supports pure paging scheme. What operations does a page daemon need to do before replacing a shared page. What data structure is used by the page daemon in Unix to determine the processes that share a page? How is this different in Windows/NT?

4. What are the factors that determine page size? To capitalize on the advantages of large and small page sizes, some architectures support multiple page sizes (e.g., R4000) such as small, medium, and large. Given these choices, what page size would you use for a text, data, and stack segment.

5. Assume that the access time for reading a memory location on a virtual memory system without TLB is x seconds. As soon as a TLB is added, the access time becomes $x/2$ seconds. What kind of virtual memory implementation does this system use (paging, segmentation, or combined paging and segmentation)? Justify your answer.

6. Argue for and against large TLB (Translation Lookaside Buffer) sizes. What kind of a TLB system would you design for a micro kernel and why?

4. "File systems" (20 points, 40 minutes)

1. Assume that we have a stateful server which informs clients as soon as a file is updated at the server (becomes invalid). Why would it still be necessary for the client to periodically check with the server about the validity of the cached files?

2. Explain how by use of a file allocation table, random access can be provided for a file system that uses linked allocation. The contents of the file allocation table are x, x, 10, 11, 7, x, 3, 2, x, x, 12, 14, eof, x, eof, x. Where eof implies end-of-file and x does not matter for our purposes. What is the size of the file in blocks if File A starts at block 4? What is the size of File B that starts at block 6. Assume blocks are numbered starting from 1.