153

Operating Systems CSL373: Major

May 8, 2007 Max Marks 66

Write your assumption, if any, in the answer scripts. Answer all 7 questions.

Synchronization

(Q1) Due to the Dussehra rush, you have been hired by the Vasant Kunj Mall to provide synchronization for their revolving door. The door is pictured below. Your job is to keep people flowing in both directions through the door. Here are the constraints: [10]

1. The door has four (4) compartments. Each compartment can hold, at most, one (1) person. The door only revolves in one direction.
2. People wait in line on either side of the door until an empty compartment is available. A person must exit from a compartment before the next person can enter it.
3. If people are waiting both inside and outside, then one person from each direction should move through the door at each turn.
4. You have available to you (already written) a procedure called **TurnDoor()**, which rotates the door by 90 degrees. You have no idea how long this procedure takes to execute. If you rotate the door *while* a person is getting in or out, you will squash them.
5. You also have available to you (already written) procedures called **EnterDoor()** and **LeaveDoor()**, which moves a person calling this routine into and out of the revolving door. You have no idea how long these procedures take to execute.
6. When your program starts, assume that the door is completely empty and that open compartments are facing the inside and outside. Also, assume that no one is waiting in line.
7. You are to use **C++, and use monitors** with condition variables and wait() and wakeup() calls) as your synchronization mechanism.

You are to write the code for **EnterStore()** and **LeaveStore()** procedures. These procedures are called by a person (process) when they (it) want to enter or leave the store. These procedures will return when the person (process) has safely entered or exited the store. You also have to write the code for any procedure that **EnterStore()** and **LeaveStore()** may need to call (except **TurnDoor()**).

Memory Management

(Q2) Consider a demand paging system. We measure the various resource utilizations and see: [7]

CPU utilization 20%

Paging disk 99%

Which of these (if any) should improve the CPU utilization? Why?

a) Get a faster CPU.
b) Get a bigger disk
c) Get a faster disk.
d) Increase the degree of multiprogramming.
e) Decrease the degree of multiprogramming.

I/O Systems

(Q3) Assume that a disk drive manufacturer has invented an atomic disk drive. [9]
It is identical to a normal disk drive except it can take groups of write commands and perform them atomically. Once the drive has accepted the group-write request all the writes will be written to the disk even if the power fails in the middle.

(a) Describe how a new atomic disk could be used by a file system to solve some of the problems of crash recovery. Give examples of how the file system would use it to implement the creation of a file.

(b) There is a disagreement among the file system designers of what the atomic disk means for the write ahead logging technique. Some say it is still a valuable technique even with an atomic disk while others claim it is pointless now. Which is right? Justify.
(c) Describe how the disk drive manufacturers could make group writes of the atomic disk go faster than if it was given the write requests one at a time.

(Q4) In class we described three file descriptor structures: [7]

(i) Indexed files. [eg inodes]
(ii) Linked files. [eg fat]
(iii) Contiguous (extent-based) allocation.

Each of the structures has its advantages and disadvantages depending on the goals for the file system and the expected file access pattern. For each of the following situations, rank the three structures in order of preference. Be sure to include the justification for your rankings.

(a) You have a file system where the most important criterion is the performance of sequential access to very large files.
(b) You have a file system where the most important criterion is the performance of random access to very large files.
(c) You have a file system where the most important criterion is the utilization of the disk capacity (i.e. getting the most file bytes on the disk).

Unix File System

(Q5) Hard links in Unix are tracked using a reference count in the inode data structure. Describe what would happen to a file system if an error caused a reference count that was

(a) too large?
(b) too small?

Describe how the problem would likely manifest itself to the user of the system. [6]

Protection and Security

(Q6)

i) Assume that you are given a file system that used capabilities to protect access to files. Describe an algorithm you could use to convert it into using access control lists. The access control lists should give the users the same access to files as with capabilities. [3]

ii) Your project partner adds a weird protection system to your project's file system. The protection system requires that in order to open a file a process must pass the *message digest* of the file as an additional argument to the open system call. Unfortunately your partuer leaves town early leaving you to write up the design document for the project.

(a) Would you classify this protection system as an access control list or a capability based system? Justify your answer. [3]
(b) Describe a file system workload in which this kind of protection system would have large performance overhead. [3]

154

True/False Objective Part TO be answercd here itself. Name:_____

Roll No_____

(Q7) Please indicate whether the statement is **true (T)** or **false [F]** **[18]**

1) (1 point) Context switch time on modern hardware is small enough to be ignored entirely when designing a CPU scheduler.
2) (1 point) Setting the "base" and "limit" registers are privileged operations.
3) (1 point) Shortest-job-first scheduling is not suitable for a general-purpose computer system.
4) (1 point) DMA (Direct Memory Aecess) requires only one interrupt per data block.
5) (1 point) In a monolithic kernel, most operating system components rcside and execute outside the kernel.
6) (1 point) Threads bclonging to one process have separate address spaces.
7) (1 point) There is always a deadlock when the mutual exclusion, hold and wait, and no preemption conditions hold at thc same time.
8) (1 point) An operating system can always disallow the mutual exclusion to prevent deadlocks.
9) (1 point) Page faults cannot be handled by software because the overhead is too large.
10) (1 point) A "write through" policy is used for virtual memory systems.

11) (2 points) Coherence cache misses cannot occur with a memory system attached to a single processor, even if I/O peripherals are able to read and write the memory system independent of the proccssor. If false, explain your answer.

12) (2 points) In a memory system with a multi-level data cache (small/fast L1 data cachc, larger/slower L2 data cache), the miss rate of the L1 cache is usually larger than the miss ratc of the L2 cache. If falsc, explain your answer

13) Clearly point out the similarities and differences between two concepts/terms.
(a) temporal and spatial locality (in the context of memory management) [2]

(b) semaphores and monitors [2]