CS433 Operating Systems

SCI2 302 M.W. 16:00-17:15

Homework #4

Part IV –Storage Management

1. None of the disk-scheduling disciplines, except FCFS, is truly fair (starvation may occur). (6 pts)
   1. Explain why this assertion is true.

This assertion is true because new requests for the track over which the head currently resides can theoretically arrive as quickly as these requests are being serviced.

* 1. Describe a way to modify algorithms such as SCAN to ensure fairness.

A way to modify algorithms like SCAN to ensure fairness is all requests older than some predetermined age could be “forced” to the top of the queue, and an associated bit for each could be set to indicate that no new request could be moved ahead of these requests. For SSTF, the rest of the queue would have to be reorganized with respect to the last of these “old” requests.

* 1. Explain why fairness is an important goal in a time-sharing

Fairness is an important goal in a time-sharing because it prevents extremely long response times.

1. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is:

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? (10 pts)

1. FCFS

The FCFS schedule is 143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. The total seek distance is 7081.

1. SSTF

The SSTF schedule is 143, 130, 86, 913, 948, 1022, 1470, 1509, 1750, 1774. The total seek distance is 1745.

1. SCAN

The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 130, 86. The total seek distance is 9769.

1. LOOK

The LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 130, 86. The total seek distance is 3319.

1. C-SCAN

The C-SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 86, 130. The total seek distance is 9813.

1. The reliability of a hard-disk drive is typically described in terms of a quantity called *mean time between failures* (*MTBF*). Although this quantity is called a “time,” the MTBF actually is measured in drive-hours per failure. (3 pts)
   1. If a system contains 1000 drives, each of which has a 750,000-hourMTBF, which of the following best describes how often a drive failure will occur in that disk farm: once per thousand years, once per century, once per decade, once per year, once per month, once per week, once per day, once per hour, once per minute, or once per second?

750,000 drive-hours per failure divided by 1000 drives gives 750hours per failure—about 31 days or once per month

1. Discuss the reasons why the operating system might require accurate information on how blocks are stored on a disk. How could the operating system improve file system performance with this knowledge? (4 pts)

While allocating blocks for a file, the operating system could allocate blocks that are geometrically close by on the disk if it had more information regarding the physical location of the blocks on the disk. In particular, it could allocate a block of data and then allocate the second block of data in the same cylinder but on a different surface at a rotationally optimal place so that the access to the next block could be made with minimal cost.

1. Provide examples of applications that typically access files according to sequential and random access methods. (4 pts)

Random Access Method:

An example could be an pdf reader. The book can be flipped open to any arbitrary page instead of reading it sequentially. This requires random access of file.

Sequential Access Method:

An example could be a notepad. Notepad opens file using sequential access and reads contents of file sequentially character by character

.

1. Give an example of an application that could benefit from operating system support for random access to indexed files. (3 pts)

A database could benefit from operating system support for random access to index file. This is because a file in the database will be searched for the file by a file pointer; if the file is found, it's contents will be fetched.

1. Some systems provide file sharing by maintaining a single copy of a file; other systems maintain several copies, one for each of the users sharing the file. Discuss the relative merits of each approach. (4 pts)

The merits for each approach are the following: automatic opening and closing of files relieves the user from the invocation of these functions. This makes it more convenient to the user. The drawback is it requires more overhead than the case where explicit opening and closing is required.

1. Consider a system where free space is kept in a free-space list. (4 pts)
   1. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.

The system can reconstruct the free-space list. The process to reconstruct the free list would be: it would be necessary to perform “garbage collection.” This would entail searching the entire directory structure to determine which pages are already allocated to jobs. Those remaining unallocated pages could be relinked as the free-space list.

* 1. Consider a file system similar to the one used by UNIX with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at */a/b/c*? Assume that none of the disk blocks is currently being cached.

Reading the contents of the small local file /a/b/c involves 4 separate disk operations: Reading in the disk block containing the root directory /, reading in the disk block containing the directories b and c, and reading in the disk block containing the file c.

1. Some file systems allow disk storage to be allocated at different levels of granularity. For instance, a file system could allocate 4 KB of disk space as a single 4-KB block or as eight 512-byte blocks. How could we take advantage of this flexibility to improve performance? What modifications would have to be made to the free-space management scheme in order to support this feature? (4 pts)

Such a scheme would decrease internal fragmentation. If a file is 5 KB, then it could be allocated a 4 KB block and two contiguous 512-byte blocks. In addition to maintaining a bitmap of free blocks, one would also have to maintain extra state regarding which of the subblocks are currently being used inside a block. The allocator would then have to examine this extra state to allocate subblocks and coalesce the subblocks to obtain the larger block when all of the subblocks become free.

1. Consider a file system that uses inodes to represent files. Disk blocks are 8-KB in size and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, plus single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system? (3 pts)

(12 \* 8 /KB/) + (2048 \* 8 /KB) + (2048 \* 2048 \* 8 /KB/) + (2048 \* 2048 \* 2048 \* 8 /KB) **= 64 terabytes**

1. Discuss how performance optimizations for file systems might result in difficulties in maintaining the consistency of the systems in the event of computer crashes. (4 pts)

The primary difficulty that might arise is due to delayed updates of data and metadata. Updates could be delayed in the hope that the same data might be updated in the future or that the updated data might be temporary and might be deleted in the near future. However, if the system were to crash without having committed the delayed updates, then the consistency of the file system is destroyed.