CHAPTER 11

Mass-Storage Structure

Practice Exercises

11.1 Is disk scheduling, other than FCFS scheduling, useful in a single-user environment? Explain your answer.

Answer:
In a single-user environment, the I/O queue usually is empty. Requests generally arrive from a single process for one block or for a sequence of consecutive blocks. In these cases, FCFS is an economical method of disk scheduling. But LOOK is nearly as easy to program and will give much better performance when multiple processes are performing concurrent I/O, such as when a Web browser retrieves data in the background while the operating system is paging and another application is active in the foreground.

11.2 Explain why SSTF scheduling tends to favor middle cylinders over the innermost and outermost cylinders.

Answer:
The center of the disk is the location having the smallest average distance to all other tracks. Thus, the disk head tends to move away from the edges of the disk. Here is another way to think of it. The current location of the head divides the cylinders into two groups. If the head is not in the center of the disk and a new request arrives, the new request is more likely to be in the group that includes the center of the disk; thus, the head is more likely to move in that direction.

11.3 Why is rotational latency usually not considered in disk scheduling? How would you modify SSTF, SCAN, and C-SCAN to include latency optimization?

Answer:
Most disks do not export their rotational position information to the host. Even if they did, the time for this information to reach the scheduler would be subject to imprecision, and the time consumed by the scheduler is variable, so the rotational position information would become incorrect. Further, the disk requests are usually given in terms

of logical block numbers, and the mapping between logical blocks and physical locations is very complex.

11.4 Why is it important to balance file-system I/O among the disks and controllers on a system in a multitasking environment?

Answer:
A system can perform only at the speed of its slowest bottleneck. Disks or disk controllers are frequently the bottlenecks in modern systems, as their individual performance cannot keep up with that of the CPU and system bus. When I/O is balanced among disks and controllers, neither an individual disk nor a controller is overwhelmed, so that bottleneck is avoided.

11.5 What are the tradeoffs involved in rereading code pages from the file system versus using swap space to store them?

Answer:
If code pages are stored in swap space, they can be transferred more quickly to main memory (because swap-space allocation is tuned for faster performance than general file-system allocation). Using swap space can require startup time if the pages are copied there at process invocation rather than just being paged out to swap space on demand. Also, more swap space must be allocated if it is used for both code and data pages.

11.6 Is there any way to implement truly stable storage? Explain your answer.

Answer:
Truly stable storage would never lose data. The fundamental technique for stable storage is to maintain multiple copies of the data, so that if one copy is destroyed, some other copy is still available for use. But for any scheme, we can imagine a large enough disaster that all copies are destroyed.

11.7 It is sometimes said that tape is a sequential-access medium, whereas a hard disk is a random-access medium. In fact, the suitability of a storage device for random access depends on the transfer size. The term ***streaming transfer rate*** denotes the rate for a data transfer that is underway, excluding the effect of access latency. In contrast, the ***effective transfer rate*** is the ratio of total bytes to total seconds, including overhead time such as access latency.

Suppose we have a computer with the following characteristics: the level-2 cache has an access latency of 8 nanoseconds and a streaming transfer rate of 800 megabytes per second, the main memory has an access latency of 60 nanoseconds and a streaming transfer rate of 80 megabytes per second, the hard disk has an access latency of 15 milliseconds and a streaming transfer rate of 5 megabytes per second, and a tape drive has an access latency of 60 seconds and a streaming transfer rate of 2 megabytes per second.

a. Random access causes the effective transfer rate of a device to decrease, because no data are transferred during the access time.

For the disk described, what is the effective transfer rate if an average access is followed by a streaming transfer of (1) 512 bytes, (2) 8 kilobytes, (3) 1 megabyte, and (4) 16 megabytes?

b. The utilization of a device is the ratio of effective transfer rate to streaming transfer rate. Calculate the utilization of the disk drive for each of the four transfer sizes given in part a.

c. Suppose that a utilization of 25 percent (or higher) is considered acceptable. Using the performance figures given, compute the smallest transfer size for a disk that gives acceptable utilization.

d. Complete the following sentence: A disk is a random-access device for transfers larger than ______ bytes and is a sequential-access device for smaller transfers.

e. Compute the minimum transfer sizes that give acceptable utilization for cache, memory, and tape.

f. When is a tape a random-access device, and when is it a sequential-access device?

Answer:

a. For 512 bytes, the effective transfer rate is calculated as follows.
ETR = transfer size/transfer time.
If X is transfer size, then transfer time is ((X/STR) + latency).
Transfer time is 15ms + (512B/5MB per second) = 15.0097ms.
Effective transfer rate is therefore 512B/15.0097ms = 33.12 KB/sec.
ETR for 8KB = .47MB/sec.
ETR for 1MB = 4.65MB/sec.
ETR for 16MB = 4.98MB/sec.

b. Utilization of the device for 512B = 33.12 KB/sec / 5MB/sec = .0064 = .64
For 8KB = 9.4%.
For 1MB = 93%.
For 16MB = 99.6%.

c. Calculate .25 = ETR/STR, solving for transfer size X.
STR = 5MB, so 1.25MB/S = ETR.
1.25MB/S * ((X/5) + .015) = X.
.25X + .01875 = X.
X = .025MB.

d. A disk is a random-access device for transfers larger than K bytes (where K > disk block size), and is a sequential-access device for smaller transfers.

e. Calculate minimum transfer size for acceptable utilization of cache memory:
STR = 800MB, ETR = 200, latency = $8 * 10^{-9}$.
200 (XMB/800 + $8 \text{ X } 10^{-9}$) = XMB.
.25XMB + $1600 * 10^{-9}$ = XMB.
X = 2.24 bytes.

Calculate for memory:
STR = 80MB, ETR = 20, L = 60 * 10^{-9}.
20 (XMB/80 + 60 * 10^{-9}) = XMB.
.25XMB + 1200 * 10^{-9} = XMB.
X = 1.68 bytes.
Calculate for tape:
STR = 2MB, ETR = .5, L = 60s.
.5 (XMB/2 + 60) = XMB.
.25XMB + 30 = XMB.
X = 40MB.

f. It depends on how it is being used. Assume we are using the tape to restore a backup. In this instance, the tape acts as a sequential-access device where we are sequentially reading the contents of the tape. As another example, assume we are using the tape to access a variety of records stored on the tape. In this instance, access to the tape is arbitrary and hence considered random.

11.8 Could a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organization (with nonredundant striping of data)? If so, how?

Answer:
Yes, a RAID level 1 organization could achieve better performance for read requests. When a read operation is performed, a RAID level 1 system can decide which of the two copies of the block should be accessed to satisfy the request. It could base this choice on the current location of the disk head and could therefore optimize performance by choosing the disk head that is closer to the target data.

11.9 Give three reasons to use HDDs as secondary storage.

Answer:
HDDs are still the most common secondary storage device.

a. They are the largest random-access storage device for the price, providing terabytes of storage at a low cost.

b. Many devices, from external chassis to storage arrays, are designed to use HDDs, providing a wide variety of ways to use HDDs.

c. They maintain the same read and write performance over their lifetime, unlike NVM storage devices, which lose write performance as they get full and as they age.

11.10 Give three reasons to use NVM devices as secondary storage.

Answer:
NVM devices are increasing in size and decreasing in price faster than HDDs.

a. High-speed NVM devices (including SSDs and usually not including USB drives) are much faster than HDDs. Secondary storage speed has a large impact on overall system performance.

b. NVM devices use less power than HDDs, making them very useful for laptops and other portable, battery-powered devices. NVM devices can also be much smaller than HDDs and thus can, for example, be surface-mounted to motherboards in devices like smartphones.

c. Because NVM devices have no moving parts, they tend to be much more reliable than HDDs.

Exercises

11.11 None of the disk-scheduling disciplines, except FCFS, is truly fair (starvation may occur).

a. Explain why this assertion is true.

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

c. Explain why fairness is an important goal in a multi-user systems.

d. Give three or more examples of circumstances in which it is important that the operating system be unfair in serving I/O requests.

Answer:

a. New requests for the track over which the head currently resides can theoretically arrive as quickly as these requests are being serviced.

b. 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.

c. Fairness is important to prevent unusually long response times.

d. Paging and swapping should take priority over user requests. It may be desirable for other kernel-initiated I/O, such as the writing of file-system metadata, to take precedence over user I/O. If the kernel supports real-time process priorities, the I/O requests of those processes should be favored.

11.12 Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is:

2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

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?

a. FCFS

b. SCAN

c. C-SCAN

Answer:

a. The FCFS schedule is 2,150; 2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681 The total seek distance is 13,011.

b. The SCAN schedule is 2,150; 2,296; 2,800; 3,681; 4,965; 2,069; 1,618; 1,523; 1,212; 544, 356. The total seek distance is 7,492.

c. The C-SCAN schedule is 2,150; 2,296; 2,800; 3,681; 4,965; 356; 544; 1,212; 1,523; 1,618; 2,069. The total seek distance is 9,917.

11.13 Compare and contrast HDDs and NVM devices. What are the best applications for each type?

Answer:
HDD characteristics: large capacity, low cost, consistent read and write speed over their lifespan, many storage devices designed to integrate them. Useful when large capacities are needed and when cost is more important than performance.
NVM characteristics: smaller capacity, more expensive, much faster, more reliable, varying write performance, smaller, use less power. Useful when performance, battery life, or reliability is of most importance.

11.14 Compare the performance of C-SCAN and SCAN scheduling, assuming a uniform distribution of requests. Consider the average response time (the time between the arrival of a request and the completion of that request's service), the variation in response time, and the effective bandwidth. How does performance depend on the relative sizes of seek time and rotational latency?

Answer:
There is no simple analytical argument to answer the first part of this question. It would make a good small simulation experiment for students. The answer can be found in Figure 2 of Worthington et al. [1994]. (Worthington et al. studied the LOOK algorithm, but similar results obtain for SCAN.) The figure shows that C-LOOK has an average response time just a few percent higher than LOOK but that C-LOOK has a significantly lower variance in response time for medium and heavy workloads. The intuitive reason for the difference in variance is that LOOK (like SCAN) tends to favor requests near the middle cylinders, whereas the C-versions do not have this imbalance. The intuitive reason for the slower response time of C-LOOK is the "circular" seek from one end of the disk to the farthest request at the other end. This seek satisfies no requests. It causes only a small performance degradation because the square-root dependency of seek time on distance implies that a

long seek isn't terribly expensive by comparison with moderate-length seeks.

For the second part of the question, we observe that these algorithms do not schedule to improve rotational latency; therefore, as seek times decrease relative to rotational latency, the performance differences between the algorithms will decrease.

11.15 Consider a RAID level 5 organization comprising five disks, with the parity for sets of four blocks on four disks stored on the fifth disk. How many blocks are accessed in order to perform the following?

a. A write of one block of data

b. A write of seven continuous blocks of data

Answer:

a. A write of one block of data requires the following: read of the parity block, read of the old data stored in the target block, computation of the new parity based on the difference between the new and old contents of the target block, and write of the parity block and the target block.

b. Assume that the seven contiguous blocks begin at a four-block boundary. A write of seven contiguous blocks of data could be performed by writing the seven contiguous blocks, writing the parity block of the first four blocks, reading the eighth block, computing the parity for the next set of four blocks, and writing the corresponding parity block onto disk.

11.16 Assume that you have a mixed configuration comprising disks organized as RAID level 1 and RAID level 5 disks. Assume that the system has flexibility in deciding which disk organization to use for storing a particular file. Which files should be stored in the RAID level 1 disks and which in the RAID level 5 disks in order to optimize performance?

Answer:

Frequently updated data need to be stored on RAID level 1 disks, while data that are more frequently read as opposed to being written should be stored in RAID level 5 disks.

11.17 The reliability of a storage device is typically described in terms of mean time between failures (MTBF). Although this quantity is called a "time," the MTBF actually is measured in drive-hours per failure.

a. If a system contains 1,000 disk drives, each of which has a 750,000-hour MTBF, 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?

b. Mortality statistics indicate that, on the average, a U.S. resident has about 1 chance in 1,000 of dying between the ages of 20 and 21. Deduce the MTBF hours for 20-year-olds. Convert this figure from hours to years. What does this MTBF tell you about the expected lifetime of a 20-year-old?

c. The manufacturer guarantees a 1-million-hour MTBF for a certain model of disk drive. What can you conclude about the number of years for which one of these drives is under warranty?

Answer:

a. 750,000 drive-hours per failure divided by 1,000 drives gives 750 hours per failure—about 31 days, or once per month.

b. We can calculate the human hours per failure as 8,760 (hours in a year) divided by 0.001 failure, giving a value of 8,760,000 “hours” for the MTBF; 8,760,000 hours equals 1,000 years. This tells us nothing about the expected lifetime of a person of age 20.

c. The MTBF tells nothing about the expected lifetime. Hard disk drives are generally designed to have a lifetime of five years. If such a drive truly has a million-hour MTBF, it is very unlikely that the drive will fail during its expected lifetime.

11.18 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?

Answer:

While allocating blocks for a file, the operating system could allocate blocks that are geometrically nearby 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.