

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

a. Explain why this assertion is true.

FCFS 对于每个 request 的优先级相同, 其他都不是, 因此都有可能出现饥饿现象;

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

对每个 request 可以用简单的一两位记录一下时间戳, 每个一段时间扫描整个列表并更新时间戳, 当发现一个请求等待时间过长的时候就优先完成它;

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

多用户系统中, 当一个用户提交了大量请求并被优先处理的时候, 就可能导致其他用户提交的请求出现饥饿现象, 响应时间过长;

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

单纯磁盘读文件的请求和操作系统自身的内存交换请求;
一些内核的文件信息比起用户文件在需要的时候要被优先访问;
前台任务比起后台的优先级较高;

11.13 Suppose that a disk drive has 5000 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 226 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?

a. FCFS

序列如下: 2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681
总距离是 13011

b. SCAN

序列如下: 2150, 2296, 2800, 3681, 4965, (4999), 2069, 1618, 1523, 1212, 544, 356

总距离是 7492

c. C-SCAN

序列如下: 2150, 2296, 2800, 3681, 4965, (4999), (0), 356, 544, 1212, 1523, 1618, 2069

总距离是 9917

11.14 Elementary physics states that when an object is subjected to a constant acceleration a , the relationship between distance d and time t is given by $d = \frac{1}{2}at^2$. Suppose that, during a seek, the disk in Exercise 11.13 accelerates the disk arm at a constant rate for the first half of the seek, then decelerates the disk arm at the same rate for the second half of the seek. Assume that the disk can perform a seek to an adjacent cylinder in 1 millisecond and a full-stroke seek over all 5,000 cylinders in 18 milliseconds.

a. The distance of a seek is the number of cylinders over which the head moves. Explain why the seek time is proportional to the square root of the seek distance.

设它开始匀加速, 后来匀减速, 距离和加速度相同:

$$\frac{1}{2}at'^2 = d/2$$

$$t' = \sqrt{d/a}$$

$$t = 2 * t' = 2\sqrt{d/a}$$

b. Write an equation for the seek time as a function of the seek distance. This equation should be of the form $t = x + y\sqrt{L}$, where t is the time in milliseconds and L is the seek distance in cylinders.

由:

$$1 = x + y\sqrt{1}$$

$$18 = x + y\sqrt{4999}$$

得:

$$t = 7561 + 2439\sqrt{L}$$

c. Calculate the total seek time for each of the schedules in Exercise 11.13. Determine which schedule is the fastest (has the smallest total seek time).

- FCFS 90.02ms
- SCAN 68.85ms
- C-SCAN 78.25ms

d. The percentage speedup is the time saved divided by the original time.

What is the percentage speedup of the fastest schedule over FCFS

23.52%

11.15 Suppose that the disk in Exercise 11.14 rotates at 7200 RPM.

a. What is the average rotational latency of this disk drive?

$$7200 \text{ RPM} = 120 \text{ r/s}$$

$$\text{average rotational latency} = \text{半圈} = 1/240\text{s} = 4.167\text{ms}$$

b. What seek distance can be covered in the time that you found for part a?

$$t = 4.167\text{ms} = 0.7561 + 0.2439 * \text{sqrt}(L)$$

$$L = 195.5 \text{ tracks}$$

13.9 Consider a file system in which a file can be deleted and its disk space reclaimed while links to that file still exist. What problems may occur if a new file is created in the same storage area or with the same absolute path name? How can these problems be avoided?

这个旧的 link 会指向新的文件，也不会有报错；

方案：可以在一个表中保留所有的 link，当文件被删除的时候让所有 link 失效；也可以通过 hash 文件，然后在 link 里面保留 hash 值的方式让 link 只能指向这个文件，不能指向其他同名同位置的不同文件；

13.13 Some systems automatically open a file when it is referenced for the first time and close the file when the job terminates. Discuss the advantages and disadvantages of this scheme compared with the more traditional one, where the user has to open and close the file explicitly.

advantages：对于用户来说编程比较简单，不需要进行额外的资源控制；

disadvantages：会占用更多的系统资源，降低速度；

13.16 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

- 只保留一份拷贝：

节省空间，但不对同步做任何控制，多个用户同时写入的话可能会出现不可预料的结果；

- 每个用户一份：

保证了每个用户在同时写文件的时候不会出现问题，也能提高性能，但会占用大量空间；

14.4 One problem with contiguous allocation is that the user must preallocate enough space for each file. If the file grows to be larger than the space allocated for it, special actions must be taken. One solution to this problem is to define a file structure consisting of an initial contiguous area of a specified size. If this area is filled, the operating system automatically defines an overflow area that is linked to the initial contiguous area. If the overflow area is filled, another overflow area is allocated. Compare this implementation of a file with the standard contiguous and linked implementations.

- 比起标准的 contiguous allocation 形式：

解决了外部碎片、文件占用空间增长的问题，但也可能会带来内部碎片，同时速度会相对降低；

- 比起标准的 linked implementations：

链接的数量下降了很多，性能会更好；

14.14 Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:

a. How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)

设起始块号为Z

- contiguous: 物理块号 = $Z + \text{逻辑块号} / 512$; offset = $\text{逻辑块号} \% 512$;
- linked: 设 pointer 是 1 byte; 需要读取 $\text{逻辑块号} / (512 - 1) + 1$ 个 block 才能知道这个块在哪; offset = $\text{逻辑块号} \% (512 - 1) + 1$
- indexed: 直接读 index 获取块号; offset = $\text{逻辑块号} \% 512$;

b. If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

- contiguous: 1
- linked: 4
- indexed: 2