

C502 – Operating Systems Tutorial *

Disk Management – Solutions

Note: The solution notes below only briefly list (some of) the key points that should be included in an answer. They are by no means complete. In an exam, you are expected to spell out the solution more fully and include a detailed explanation of your reasoning.

1. Briefly describe each of the following terms with respect to disk management:

- (a) Low-level format
- (b) Seek Time
- (c) SCAN scheduling
- (d) C-SCAN scheduling

(Exam question in 2016-17).

Answer:

- (a) *Low-level format* - Done to create the disk sector layout of the hard drive. Usually done by the manufacturer.
- (b) *Seek Time* - Time taken for the necessary track to appear under the read/write head of the hard drive.
- (c) *SCAN scheduling* - Process requests that result in the shortest seek time in preferred direction. The direction is only changed after reaching the outermost/innermost cylinder (or no further requests in preferred direction).
- (d) *C-SCAN scheduling* - Service requests in one direction only and change jump to outermost request once you have processed the innermost request.

2. Suppose that the current position of the disk arm is over cylinder 200. The disk request queue contains requests for sectors on the following cylinders: 400, 20, 19, 74, 899

In which order will the requests be handled under:

- (a) the FCFS disk head scheduling policy?
- (b) the SSTF policy?
- (c) the SCAN policy?
- (d) the C-SCAN policy?

Briefly describe the policies and their respective trade-offs.

Answer:

- (a) *FCFS (first come first served):* 400, 20, 19, 74, 899. FCFS services disk requests in the order they arrive. It is simple, but it performs well only if the average queue length is close to one.
- (b) *SSTF (shortest seek time first):* 74, 20, 19, 400, 899. SSTF services whichever requests in the queue are closest to the current position of the disk head (it doesn't matter how ties are broken). It is only a bit more complex than FCFS, has good throughput, but tends to starve requests near the edge of the disk if the system is loaded.
- (c) *SCAN (the elevator algorithm):* 400, 899, 74, 20, 19 (assuming the initial direction is up). SCAN maintains a current seek direction and two queues: one for requests ahead of the disk head and those behind. It services the nearest request in the current direction; when no such remain, it switches direction. It has almost as good throughput as SSTF and doesn't starve any process.

*with thanks to Morris Sloman

- (d) *CSCAN: 400,899,19,20,74* (assuming the initial direction is up). *CSCAN* also maintains two queues, but when the current one becomes empty it seeks to serve the request closest to the farthest edge of the disk and resumes seeking in the same direction from there. It is slightly slower than *SCAN* but does not discriminate in favour of blocks at the middle of the disk.

3. Consider the following parameters describing a disk:

Parameter	Description
C	Number of cylinders
T	Number of tracks per cylinder (number of platters)
S	Number of sectors per track
R	Rotational velocity (rotations per second)
B	Number of bytes per sector

In terms of these parameters, how many bytes of data are on each disk cylinder?

Assume that you are designing a disk drive, and you hope to reduce the expected rotational latency for requests from the disk. Which of the parameters above would you attempt to change, and in what way would you change them?

Suppose you wanted to reduce the disk's data transfer time – which parameters would you attempt to change?

Answer:

$$\text{bytes/cylinder} = T * S * B$$

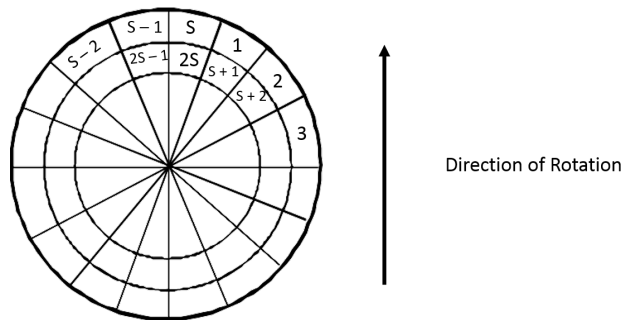
In order to reduce rotational latency, increase the rotational velocity.

*In order to reduce transfer time, increase the rotational velocity, or increase the number of bytes per track ($S * B$).*

4. A disk drive has S sectors per track and C cylinders. For simplicity, we will assume that the disk has only one, single-sided platter, i.e., the number of tracks per cylinder is one. The platter spins at R rotations per millisecond. The following function gives the relationship between seek distance d , in cylinders, and seek time, t_{seek} , in milliseconds:

$$t_{seek} = \begin{cases} 0 & \text{if } d = 0 \\ 5 + 0.05 * d & \text{if } 0 < d \leq C \end{cases}$$

The sectors are laid out and numbered sequentially, starting with the outer cylinder, as shown in the diagram below.



- (a) Suppose the disk read/write head is located over cylinder 10. The disk receives a request to read sector S . What is the expected service time for this request?

- (b) Exactly d milliseconds after completing the request for S , the disk receives a request for sector $S + 1$. What is the expected service time for this request?

Answer:

- (a) $\text{seek time} = 5 + 0.05 * 10 = 5.5 \text{ ms}$ $\text{rotational latency} = (1 / 2) * (1 / R)$
 $\text{transfer time} = (1 / S) (1 / R)$
 $\text{service time} = \text{seek time} + \text{rotational latency} + \text{transfer time}.$
- (b) $\text{seek time} = 5 + 0.05 * 1 = 5.05 \text{ ms}$
 $\text{rotational latency} = (1 / R) - (d + t_{\text{seek}})$
 $\text{transfer time} = (1 / S) * (1 / R)$
 $\text{service time} = \text{seek time} + \text{rotational latency} + \text{transfer time}$