Elevator algorithm

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(November 2007)*

The **elevator algorithm** (also **SCAN**) is a disk scheduling algorithm to determine the motion of the disk's arm and head in servicing read and write requests.

This algorithm is named after the behavior of a building elevator, where the elevator continues to travel in its current direction (up or down) until empty, stopping only to let individuals off or to pick up new individuals heading in the same direction.

From an implementation perspective, the drive maintains a buffer of pending read/write requests, along with the associated cylinder number of the request. Lower cylinder numbers indicate that the cylinder is closest to the spindle, and higher numbers indicate the cylinder is farther away.

Contents [hide]

Description [edit]

When a new request arrives while the drive is idle, the initial arm/head movement will be in the direction of the cylinder where the data is stored, either *in* or *out*. As additional requests arrive, requests are serviced only in the current direction of arm movement until the arm reaches the edge of the disk. When this happens, the direction of the arm reverses, and the requests that were remaining in the opposite direction are serviced, and so on.[1]

Variations [edit]

One variation of this method ensures all requests are serviced in only one direction, that is, once the head has arrived at the outer edge of the disk, it returns to the beginning and services the new requests in this one direction only (or vice versa). This is known as the "Circular Elevator Algorithm" or C-SCAN. This results in more equal performance for all head positions, as the expected distance from the head is always half the maximum distance, unlike in the standard elevator algorithm where cylinders in the middle will be serviced as much as twice as often as the innermost or outermost cylinders.

Other variations include:

- FSCAN
- LOOK (and **C-LOOK**)
- N-Step-SCAN

Example [edit]

The following is an example of how to calculate average disk seek times for both the SCAN and C-SCAN algorithms.

- Example list of pending disk requests (listed by track number): 100, 50, 10, 20, 75.
- The starting track number for the examples will be 35.
- The list will need to be sorted in ascending order: 10, 20, 50, 75, 100.

Both SCAN and C-SCAN behave in the same manner until they reach the last track queued. For the sake of this

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Ελληνικά
فارسی
हिन्दी
Italiano
Magyar
Српски / srpski
Edit links

example let us assume that the SCAN algorithm is currently going from a lower track number to a higher track number (like the C-SCAN is doing). For both methods, one takes the difference in magnitude (i.e. absolute value) between the next track request and the current track.

- **Seek 1:** 50 − 35 = 15
- **Seek 2:** 75 − 50 = 25
- **Seek 3:** 100 − 75 = 25

At this point both have reached the highest (end) track request. SCAN will just reverse direction and service the next closest disk request (in this example, 20) and C-SCAN will always go back to track 0 and start going to higher track requests.

- **Seek 4 (SCAN):** 20 − 100 = 80
- **Seek 5 (SCAN):** 10 − 20 = 10
- **Total (SCAN):** 155
- **Average (SCAN):** 155 ÷ 5 = 31
- **Seek 4 (C-SCAN):** 0 − 100 = 100 (C-SCAN always goes back to the first track)
- **Seek 5 (C-SCAN):** 10 − 0 = 10
- **Seek 6 (C-SCAN):** 20 − 10 = 10
- **Total (C-SCAN):** 185
- **Average (C-SCAN):** 185 ÷ 5 = 37

Note: Even though six seeks were performed using the C-SCAN algorithm, only five I/Os were actually done.

Definition of C-SCAN: C-SCAN moves the head from one end of the Disk to the other end, Servicing requests along the way.The head on reaching the other end,however immediately returns to the beginning of the Disk but without servicing any requests on the return.

Analysis [edit]

The arm movement is thus always less than twice the number of total cylinders then, for both versions of the elevator algorithm. The variation has the advantage to have a smaller variance in response time. The algorithm is also relatively simple.

However, the elevator algorithm is not always better than shortest seek first, which is slightly closer to optimal, but can result in high variance in response time and even in starvation when new requests continually get serviced prior to existing requests.

Anti-starvation techniques can be applied to the shortest seek time first algorithm to guarantee an optimum response time.

See also [edit]

- FCFS
- Shortest seek time first

References [edit]

1. ^ "Disk scheduling". Archived from the original on 2008-06-06. Retrieved 2008-01-21.

Categories: Disk scheduling algorithms | Sorting algorithms