Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduling Algorithm

Assist. Professor Ahmed Mohamed Malallah Al-Suffar

Computer Engineering Department, Al-Farabi University College, Iraq

Abstract

According to a high gap between the bounded developments in disk scheduling algorithms in contrast to processor speed and memory capacity developments, so successive improving disk scheduling algorithms is almost needed to reduce the overall time consuming of system performance.

To enhance both traditional SCAN and C-SCAN algorithms is the goal of this research, so, rather than the best Seek-Time obtained from traditional (SCAN and C-SCAN) algorithms, the idea behind this proposal "Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduling Algorithm" is to enhance the traditional algorithms and gaining the optimal performance time by having the optimal Seek-Time.

It is worthily mentioned that the evaluation of the performance of the new strategy in this algorithm was tested by scientific criteria to ensure gaining the optimal algorithm compering with the traditional strategy.

Keywords- SCAN, C-SCAN, Scheduler, algorithm, Seek-Time, increasing throughput, response time.

1. Introduction

The circular flat shape of disk has two surfaces, which are covered with a magnetic material. The high-speed device motor spins the disk when it is in use and the disk controller orders the disk drive to carry out the instruction from the CPU, so, information is recorded on the surfaces. The disk surface is divided into circles called *tracks*. Each one of them is divided into *sectors* where the information is stored. [1]

There are two basic types of head disk in hard drives: [2]

1- Fixed head disks, which are expensive, but require no head movement to serve requests in each track on the disk.

2. Movable head disks, which are cheaper system with a single head driven by rapid motor to move and position the head over desired track on the disk.

It is worthily to mention that there are three times to access a block on the disk, the system firstly moves the head to the appropriate track by seek movement, within *seek time*. Once *latency time* as the head must wait until the desired block rotates under the read/write head. Finally, by the *transfer time, the data* transfers from the disk to main memory. Thus, the total time consumes to service all requests is the sum of the *seek time, latency time* and *transfer time*. [5, 6-B]

http://doi.org/10.33193/IJSER.1.1.2022.41

As a trade-off between increasing *throughput* and maximizing *response time* has driven to maximize the number of serving I/O requests and minimizing the movement of the head but causing no requests having to wait a long time. [4, 6-A]

Finally, there are many disk scheduler algorithms used to select which request is going to be satisfied first but no particular algorithm is absolutely the best algorithm.[3]

SCAN and C-SCAN schedulers are one of these types of algorithms, the disk head in SCAN and C-SCAN schedulers is usually serving the requests toward either innermost requests or outermost requests direction, start from the current head position at latest achieving set of requests, going thoroughly forward to the next request and so on serving extreme request at that direction without moving the head toward inverse direction until extremely finished. [1, 3]

This research's goal is to establish the optimal algorithm by enhancing the strategy of SCAN and C-SCAN algorithms to gain the optimal Seek-Time rather than traditional SCAN and C-SCAN algorithms. [3]

2. Experimental and Implementation Work

Enhancement both traditional SCAN and C-SCAN algorithms is the goal of this research. The proposed algorithm " Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduler " by this research has a new feature as to rapidly pre-reset the head position from current head position, at achieving the last set of requests, automatically to the track zero as soon as the system begins working to access and achieve the new requests.

To prove that the goal of the proposed Strategy "Pre-Reset the R/W Head to Zero Track in SCAN /C-SCAN Disk Scheduling Algorithm" is the best idea in competing the traditional scheduler algorithm of both SCAN/C-SCAN, the comparison among them becomes needed by explaining the features of both strategies through assessment of the minimum Seek-Time needed for the proposed algorithm using the same set of requests included in science literatures to prevent tendency to selective set of requests.

To settle the conditions of the implementation, suppose:

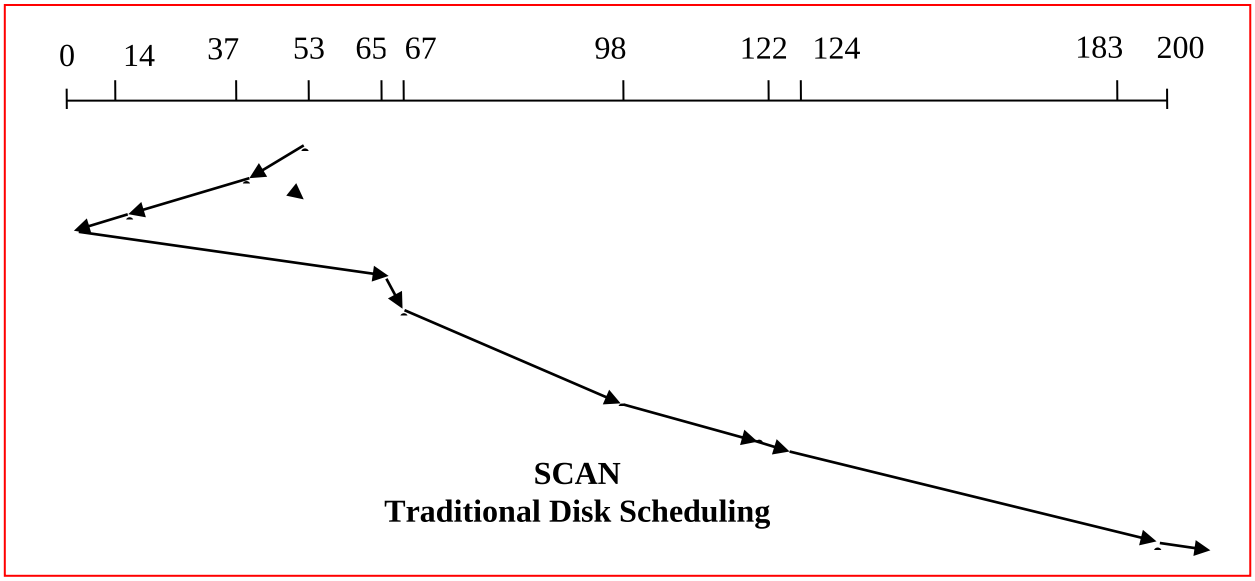
1- the disk has 200 tracks, numbered (0 to 200).

2- the certain requests are: (98, 183, 37, 122, 14, 124, 65, 67).

3- the current position of disk head is at track (53).

In Traditional SCAN Algorithm:

A known in this algorithm, the drive head moves across all requests toward one direction before return back to inverse direction. In this example the movement toward the innermost cylinders so the head services track (37) and (14) requests and goes toward track (0) before returning back changing direction to the outermost cylinders to serve the requests at tracks (65, 67, 98, …. and so on). So, it prevents starvation but it doesn't have a good locality. [6-C]



SCAN
Traditional Disk Scheduling

The order of requests is: 53, 37, 14, 65, 67, 98, 122, 124, 183.

The number of seek time from track (53) to (0) is:

[(53-37)+(37-14)+(14-0)]=53 time

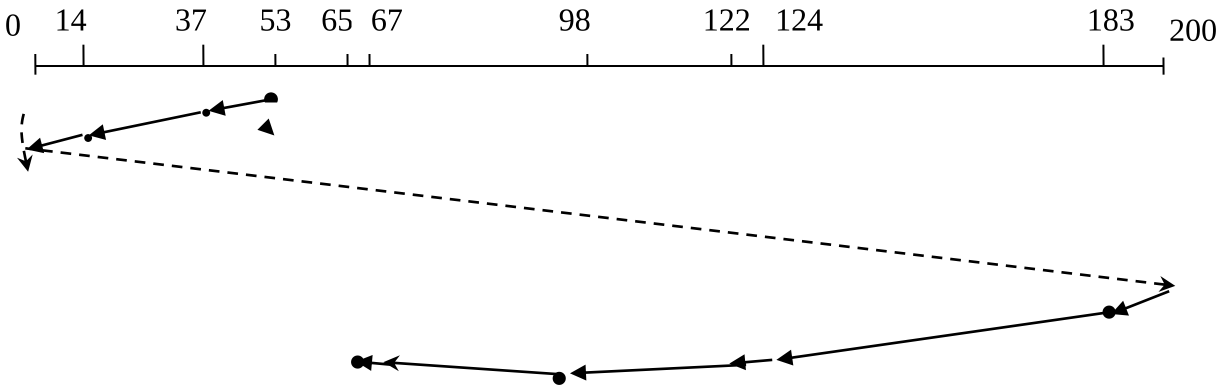
The number of seek time from track (65) to (183) is:

(65-0)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)] = 183 time

So, total seek time to R/W Head to achieve this set of requests = (53+183)=236 time

In Traditional C-SCAN Algorithm

Circular scan, has a restricted movement policy to only one direction; from innermost to outermost or vice versa. When the head reaches the last request at the exact direction the arm will move back automatically and rapidly to the other end of the disk. So, it reduces the delay for the new request when compared with the SCAN policy in addition to it is a good algorithm for a highly loaded system but it doesn't has a good locality. [1, 6-C]



C-SCAN
Traditional Disk Scheduling

The order is: 53, 65, 67, 98, 122, 124, 183, 14, 37.

To calculate the total seek time by the R/W head movement by seeking to most inner:

1- The number of seek time from track (53) to (0) is.
[(53-37)+(37-14)+(14-0)]=53 time

2- The number of seek time from track (200) to (65) is.
[(200-183)+(183-124)+(124-122)+(122-98)+(98-67)+(67-65)]=135 time

So, total seek time to R/W Head to achieve this set of requests = (53+135)=188 time

Proposed Algorithm

The algorithm in this proposal; ***Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduling Algorithm,*** once the system's r/w head finished the latest set of requests it will be rapidly reset automatically by an efficient motor to track zero.

Really, this proposal looks like ***traditional SCAN and C-SCAN Disk Scheduling Algorithm*** in all steps with additional features. That is the R/W head in this proposal never starts from current position yet but just from innermost track zero position going toward outermost cylinder serving requests successively until last request at that direction.

So, when any request arrives before begging in closed location at any extreme of cylinder, the new algorithm does guarantee to access it through strategy of pre-reset the head to extreme innermost of the disk to start scanning from there. This algorithm prevents local starvation in any extreme cylinder and has more uniform response time.

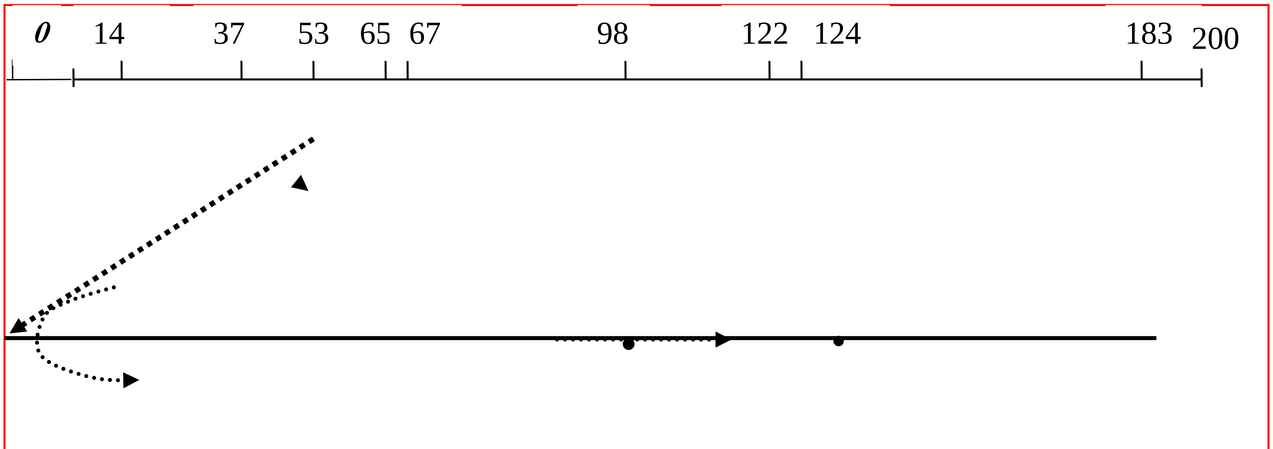
Experimental test for "***Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduling Algorithm"***:

The request's order is: (53, 65, 67, 37, 14, 183, 124, 122, 98).

1- Automatically reset the R/W head to the innermost cylinder track (0).

2- Going toward the outermost cylinder to R/W requests.

Total Seek time is: 14+23+16+12+2+31+24+2+59 = <u>183</u> time.



Pre-Reset the R/W Head to Zero Track in SCAN and C-SCAN Disk Scheduling Algorithm

3. Conclusion

Evaluation of the implementation of the proposed algorithm is depend on scientific criteria. So, it observed to yield a range of improvements:

1- consumes less time to access all requests in this tested set by minimizing Seek-Time to (183 t.). So, it has better scale for seek-time than traditional SCAN and C-SCAN algorithms with (253, 188) t. successively.

2- increased throughput.

3- produces more uniform response time.

4- The algorithm improved performance and fulfilled the basic idea behind the project's goal by achieving optimal R/W head-movement to treat all I/O requests equally with fairness response time.

References

[1] Ahmed Mohamed Mal-Allah Al-Suffar, Assist. Professor, "The Side of Shortest Remote Request First (SSRRF) as a Strategy to Enhance LOOK Algorithm", College of Engineering, AL-Iraqia University, Journal of Advanced Computer Science and Technology Research, Vol.4 No.2, June 2014, 40-49.

[2] James L. Peterson & Abraham Silberschatz, " Operating System Concepts ", Second Edition, University of Texas at Austin, Addison-Wesley, Inc. Copyright © 1985.

[3] A. M. Lister, " Fundamentals Of Operating Systems ", Third Edition, University of Queensland, Higher and Further Education Division, Copyright © 1984.

[4] DAVID BARRON, Professor of Computer Studies, University of Southampton, "Computer Operating Systems For micros, minis and mainframes", Second Edition, USA by Chapman and Hall, Copyright © 1984.

[5] Avi Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts Essentials", Ninth Edition, John Wiley & Sons, Inc, ISBN 978-1-118-06333-0, © December, 7- 2012.

[6] INTERNET

A ▪ http://www2.cs.uregina.ca/~hamilton/courses/330/notes/io/node7.html

B ▪ http://www.eecs.harvard.edu/margo/papers/usenix90

C▪http://search.tb.ask.com/search/GGmain.jhtml?searchfor=Disk+Scheduler+Circular+LOOK+%28C-