International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 06 | Sep-2015 www.irjet.net p-ISSN: 2395-0072

LAST COME FIRST SERVE ALGORITHM (LCFS) FOR DISK SCHEDULING

G Vijaya Lakshmi

Assistant Professor, Department of Computer Science, Vikrama Simhapuri University, Nellore, Andhra Pradesh, India

---

Abstract - ***In multiprogramming system, most of the jobs heavily depends on disk for program loading and I/O files. When the disk queue has several pending requests, the operating system chooses which request has to be served next. But due to increase in processor speed and memory capacity than disk speed there may have many pending request in queue and as a result there is degradation in the performance of disk scheduling. In operating system, there exists various disk scheduling algorithms like FCFS, SSTF, SCAN, (C-SCAN), LOOK, C-LOOK to enhance the performance of disk I/O operation. Therefore In this paper, to enhance the efficiency of disk performance LCFS algorithm is proposed which decreases the total number of head movements and average seek time than exiting algorithms.***

Key Words: *Average seek time, total head movement, disk scheduling, sorting*.

1. INTRODUCTION

Data is recorded on a series of magnetic disks[5,8] or platters, connected to a spindle that rotates at high speed .A read-write head lies just above the surface of each platter. The heads are attached to a disk arm [1] . The surface of the platter is divided into tracks, which are subdivided into sectors [3]. The set of tracks that are at one arm position called cylinder [2]. Once the head in a position, the read or write operation is then performed as the sector moves under the head [4].The main aim is to allocate service to the pending request in queue by reducing the number of head movements and seek time, because Increase in seek time will degrade the system performance. . Seek Time [8] is the time for the disk arm to move the heads to the cylinder containing the desired sector. There are various disk scheduling algorithm like First come first serve (FCFS), shortest seek time first(SSTF), SCAN which is also called Elevator algorithm, Circular scan(C-SCAN) [6], LOOK and C-LOOK[7] algorithm to optimize the disk I/O performance.

2. PREVIOUS WORK

In the recent years many researchers have proposed various algorithm to enhance the disk performance by reducing the total number of head movement and to minimize the seek time .Sourav Kumar Bhoi, Sanjaya Kumar panda, Imran Hossain Farak, proposed a ODSA algorithm which increases efficiency of the disk Performance by reducing seek time and transfer time [6]. Hu Ming, proposed a method based on the idea of disk arm and rotational position and showed increase in the disk rotation leads to higher data transfer time [9].

III.EXISTING DISK SCHEDULING ALGORITHMS

FCFS**:** this algorithm simply serves the request coming first. It is simple to implement. But the drawback is it does not provide fastest service. The average head movement in the algorithm is too high [10]. Shortest Seek Time Next (SSTF) selects the request with minimum seek time from the current head position. It gives substantial improvement in comparison to FCFS. Scan algorithm is called elevator algorithm. In this the disk arm moves from one end of the disk and move towards other end, while in mean time all requests are servicing until it gets other end of the disk. Comparing with FCFS and SSTF it gives better performance. C-Scan scheduling algorithm is called Circular scan[4] The head moves from one end to other end of the disk, servicing the request along the way. The waiting time increases in the algorithm. Look scheduling the disk arm moves across the full width of the disk. The arm goes as far as the final request in each direction and reverses immediately. So these are some of the disk scheduling algorithms to serve the requests'-Look is the variant of C-SCAN. Disk arm only travels as far as the last request in each direction, then immediately reverses direction, without first going all the way to the end of the disk..

IV.RESULTS AND ANALYSIS

Suppose a disk drive has 200 cylinders, numbered 0 to 199. Consider a disk queue with requests for i/o to blocks on cylinder: 46,190,130,20,25,50,198,160,178,190,198.

Assume that disk head is currently at cylinder 140. Fig1 to Fig 6 show the representation of FCFS, SSTF, SSTF, SCAN, C-SCAN, LOOK and C-LOOK disk scheduling algorithm respectively.

FCFS

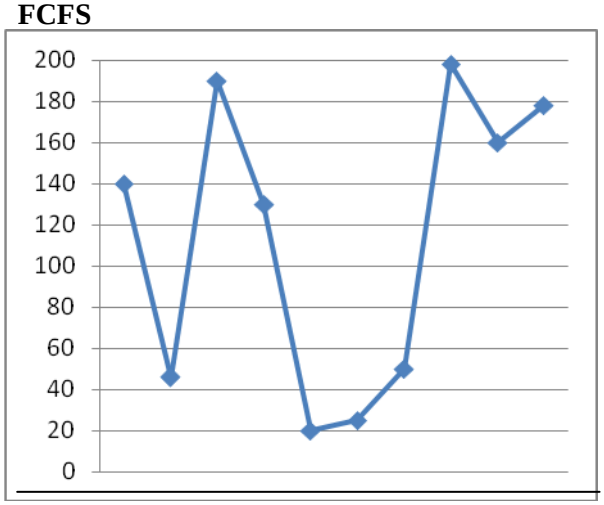

Fig1. Representation of FCFS

Total No. of Head movements: (140-46)+(46-190)+(190-130) +(130-20)+(20-25) +(25-50) +(50-198)+ (198-160) +(160-178)=642
Average Seek Time: 642/9=71.33

SSTF

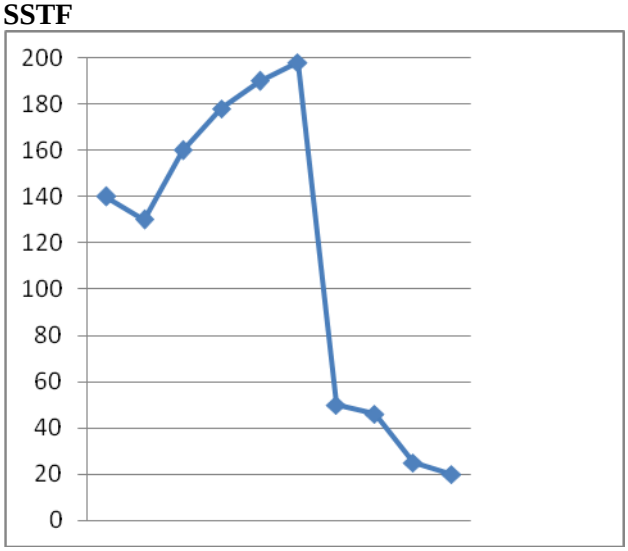

Fig2. Representation of SSTF

Total No. Of Head movements :(140-130) +(160-130)+ (178-160) +(190-178)+ (198-190) +(198-50)+ (50-46)+ (46-25)+(25-20)= 256
Average Seek Time: 28.44

SCAN

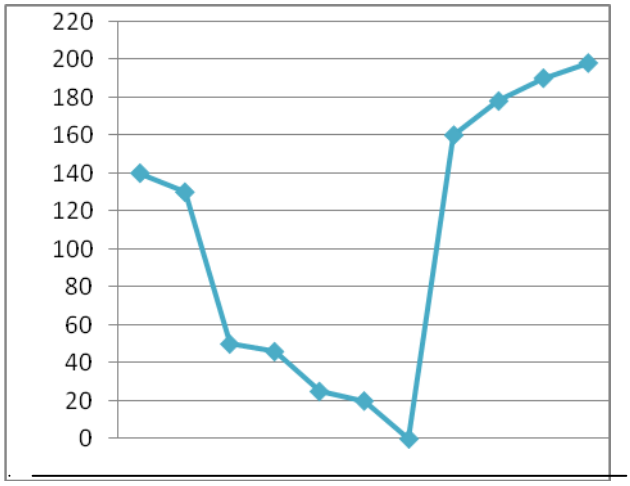

Fig3. Representation of SCAN

Total No. of Head movements:(140-130)+(130-50)+ (50-46)+ (46-25)+ (25-20)+ (20-0)+ (160-0)+ (178-160)+ (190-178)+ (198-190)=338
Average Seek Time: 37.55.

C-SCAN

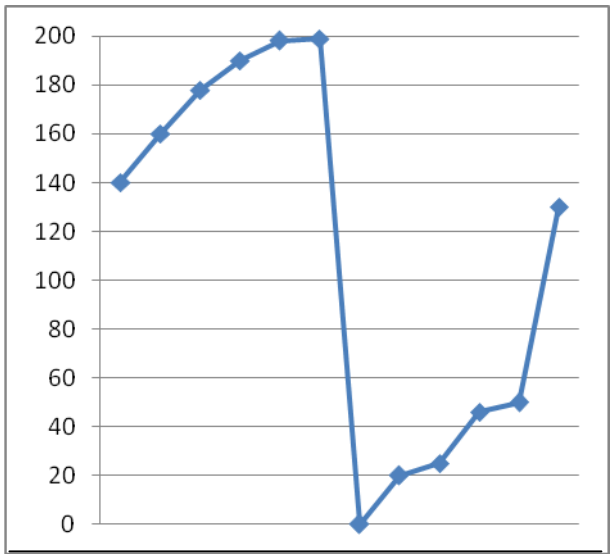

Fig4. Representation of CSCAN

Total No. Of head movements:(140-160)+(160-178)+(178-190)+(190-198)+(198-199)+(199-0)+(20-0)+(25-20)+(46-25)+(50-46)+(130-50)=388
Average Seek Time: 42.88

LOOK

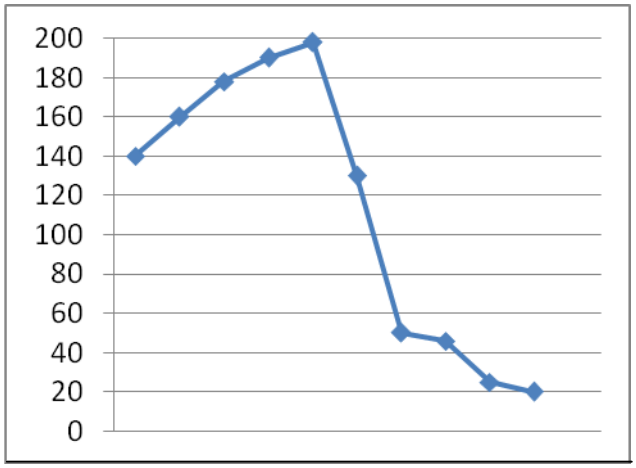

Fig5. Representation of LOOK

Total No. Of Head movements:(160-140)+(178-60)+(190-178)+(190-198)+(198-130)+(130-50)+(50-46)+(46-25)+(25-20)=236
Average Seek Time: 26.22

C- LOOK

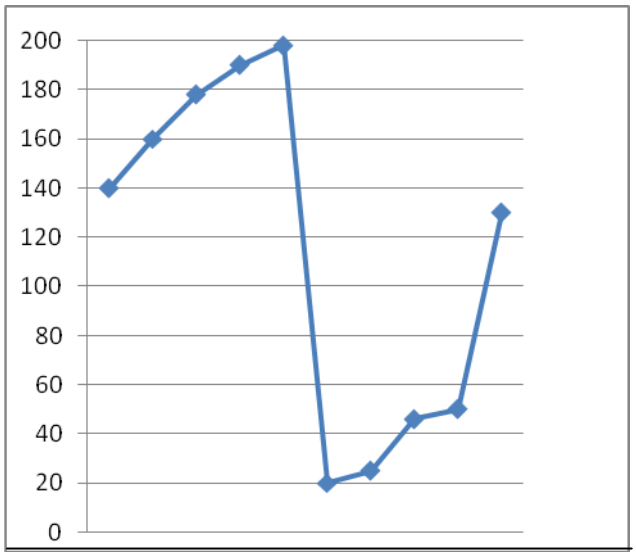

Fig 6. Representation of C -LOOK

Total No .of Head movements :(160-140)+ (178-160)+ (190-178)+ (198-190)+ (198-20)+ (25-20)+ (46-25)+ (50-46)+ (130-50)=346
Average Seek Time: 38.44

V. PROPOSED ALGORITHM (LCFS)

The main aim of the proposed work is to improve disk performance by reducing seek time and total No. of head movements. Here we assume that the disk controller and disk drive are busy. The request that cannot be serviced by the hardware will form a queue. Assuming the requests are in the random order. Now by applying any sorting techniques, sort the requests in the ascending order to the track request. Now the current head pointer first scans all request in the queue and then starts servicing from the last request i.e in descending order, until it reaches the first request in the sorted lit. Finally, we calculate the total head movement and average seek time.

1.PSEUDOCODE FOR THE PROPOSED DISK SCHEDULING ALGORITHM (LCFS):

1. Initialization
K[ ]: List of pending requests which are waiting in queue to be serviced.
N: Total no .of Requests waiting in k[ ].
CHP: Current Head position.
THM: Total Head Movement
AST: Average Seek Time
2. By applying any sorting technique, sort all the pending requests which are in queue in ascending order.
3. Read the CHP.
4. Scanning will start from last request k [i-1] to initial request k[1] from the CHP.
5. Calculate the total no of head movements
THM=CHP+ | K [i-1] + k [1] | .
6. Calculate Average Seek time
AST= THM/N
7. Stop the algorithm.

LCFS (PROPOSED ALGORITHM)

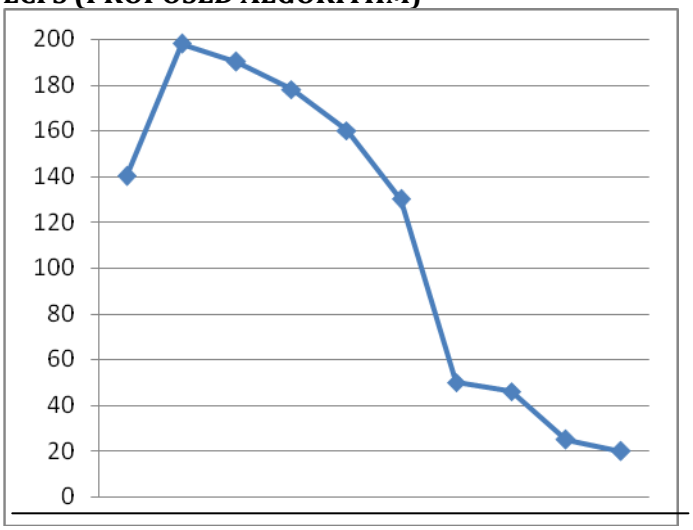

Fig7. Representation of LCFS (Proposed)

Total No of. Head movements:(198-140)+(198-190)+(190-178)+(178-160)+(160-130)+(130-50)+(50-46)+(46-25)+(25-20)=232
Average Seek Time: 25.77.

Table 1 and Fig 8 shows the comparisons of all algorithms and it is been observed that total number of head movements and average seek time decreases when compared to all the algorithm (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK) with our proposed algorithm(LCFS) .

TABLE 1: Comparison of All Algorithms

Algorithms	Total No. Head Movement	Avg seek time
FCFS	642	71.33
SSTF	256	28.44
SCAN	338	37.55
C-SCAN	386	42.88
LOOK	236	26.22
C-LOOK	346	38.44
LCFS(proposed)	**232**	**25.77**

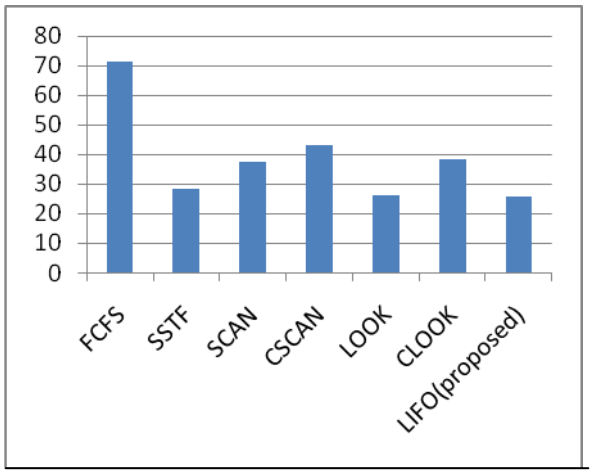

Fig 8. Comparison of average seek time

VI.CONCLUSION

From the above results and analysis , it is been observed that the proposed new scheduling algorithm LCFS gives better performance than existing algorithms like FCFS,SSTF,SCAN,C-CAN,LOOK,C-L0OK. The average seek time and total number of head movements have been improvised in proposed scheduling algorithm which increases the efficiency of the disk performance. In future we can implement this algorithm in real time systems.

REFERENCES

[1] Silberchatz, Galvin and Gagne, operating systems concepts, 8th edition, John Wiley and Sons, 2012

[2] A Comprehesive Review for Disk Scheduling Algorithm",International journal of computer science issues,vol.11,issue 1,No.1,January 2014.

[3] Dietel, Dietel and Choffnes, Operating Systems, 3rd edition, Pearson education, 2009

[4] William Stallings, "Operating Systems: Internal and Design Principles", seventh edition, prentice hall, 2012.

[5] An Improved Approach to Maximize the Performance of Disk Scheduling Algorithm by Minimizing the Head Movement and Seek Time using Sort Mid Current Comparison (SMCC) Algorithm

[6] Sourav Kumar Bhoi, Sanjaya Kumar Panda, and Imran Hossain Faruk, "Design and Performance Evaluation of an Optimized Disk Schedduling Algorithm (ODSA)", International Journal of Computer Applications, Vol. 40, No. 11, Feb 2012, pp. 28-35.

[7] A Comprehensive Review for Disk Scheduling Algorithms", IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 1, No 1, January 2014

[8] New Disk Scheduling Algorithm for Higher Performance International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)", Volume 3, Issue 1, January – February 2014.

[9] Hu. Ming, "Improved Disk Scheduling Algorithms Based on Rotational Position," *Journal of Shanghai University*, Vol. 9, No. 5, 2005,pp. 411-414.

[10] A.S. Tanenbaum, "Modern Operating Systems", Prentice-Hall, 1996.

Dr. G.Vijaya Lakshmi received her B.Tech Degree in Computer Science & Engineering from Intel College of Engineering, Affiliated to JNTU, Anantapur, India, in 2002. M.Tech in Software Engineering in JNTU College of Engineering, Anantapur, India, in 2005.Received Ph.D degree in Computer Science & Engineering from JawaharLal Nehru Technological University, Anantapur, A.P., India in February 2012. At Present she is Assistant Professor, Computer Science Dept in Vikrama Simhapuri University, Nellore, INDIA . Her current Research Interest includes computer networks, wireless communication network, cloud computing.