ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

Priority Based Disk Scheduling Algorithm

C.Mallikarjuna[1], K. Sudheer Babu[2], Dr. P. Chitti Babu[3]

Assistant Professor, Department of Master of Computer Applications, APGCCS, Rajampet, Kadapa, AndhraPradesh, India.[1]

Assistant Professor, Department of Master of Computer Applications, APGCCS, Rajampet, Kadapa, AndhraPradesh, India [2]

Professor, Department of Master of Computer Applications, APGCCS, Rajampet, Kadapa, AndhraPradesh, India [3]

ABSTRACT: Processor speed and memory size are increasing day by day several times faster than disk speed. The difference between these two identifies that disk I/O performance could become a very important factor concerning system performance. In operating systems, seek time is very crucial. Since all device I/O requests are linked in queues, if the seek time increases the system performance gets degraded. So we need certain algorithms for serving disk requests more efficiently. These algorithms are used for scheduling the various requests regarding the disks with less time and finally the system shows the better performance. In this paper we proposed a new algorithm for disk scheduling which is quite new when compared with the existing disk scheduling algorithms.

KEYWORDS: Seek Time, Scheduling, Disk Scheduling, Priority.

I. INTRODUCTION

For a multiprogramming system with many processes, the disk queue may often have several pending requests. Thus, when one request is completed, the operating system chooses which pending request to service next. All these things should be performed by incorporating one disk scheduling algorithm. In operating system disk scheduling is major concern to access the hard disk. By using disk scheduling we can access or speed up the accessing process of hard disk. Disk scheduling is the problem of deciding which particular request for data by your computer from your hard drive should be served first.

II. RELATED WORK

In this section we describe the related work regarding proposed disk scheduling algorithm. According to the proposed Disk Scheduling Algorithm, threads are scheduled to run based on their **scheduling priority.** We have also explained here how to categorize various disk requests in to many types and which priority values can be assigned to them. Each thread is assigned a scheduling priority. The priority levels range is from zero (lowest priority) to 31 (highest priority). The priority class and priority level are combined to form the **base priority** of a thread. The priority of each thread is determined by the following criteria.

- The priority class of its process
- The priority level of the thread within the priority class of its process

Priority Class

Each process belongs to one of the following priority classes.

- IDLE_PRIORITY_CLASS
- BELOW_NORMAL_PRIORITY_CLASS
- NORMAL_PRIORITY_CLASS
- ABOVE_NORMAL_PRIORITY_CLASS
- HIGH_PRIORITY_CLASS
- REALTIME_PRIORITY_CLASS

DOI: 10.15680/IJIRSET.2014.030924

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

Priority Level

The following are priority levels within each priority class.

- THREAD_PRIORITY_IDLE
- THREAD_PRIORITY_LOWEST
- THREAD_PRIORITY_BELOW_NORMAL
- THREAD_PRIORITY_NORMAL
- THREAD_PRIORITY_ABOVE_NORMAL
- THREAD_PRIORITY_HIGHEST
- THREAD_PRIORITY_TIME_CRITICAL

Base Priority

The process priority class and thread priority level are combined to form the base priority of each thread. The following table shows the base priority for combinations of process priority class and thread priority value.

TABLE 1: Priorities Table

Process priority class	Thread priority level	Base priority
IDLE_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	2
	THREAD_PRIORITY_BELOW_NORMAL	3
	THREAD_PRIORITY_NORMAL	4
	THREAD_PRIORITY_ABOVE_NORMAL	5
	THREAD_PRIORITY_HIGHEST	6
	THREAD_PRIORITY_TIME_CRITICAL	15
BELOW_NORMAL_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	4
	THREAD_PRIORITY_BELOW_NORMAL	5
	THREAD_PRIORITY_NORMAL	6
	THREAD_PRIORITY_ABOVE_NORMAL	7
	THREAD_PRIORITY_HIGHEST	8
	THREAD_PRIORITY_TIME_CRITICAL	15
NORMAL_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	6
	THREAD_PRIORITY_BELOW_NORMAL	7
	THREAD_PRIORITY_NORMAL	8
	THREAD_PRIORITY_ABOVE_NORMAL	9
	THREAD_PRIORITY_HIGHEST	10
	THREAD_PRIORITY_TIME_CRITICAL	15
ABOVE_NORMAL_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	8
	THREAD_PRIORITY_BELOW_NORMAL	9
	THREAD_PRIORITY_NORMAL	10
	THREAD_PRIORITY_ABOVE_NORMAL	11
	THREAD_PRIORITY_HIGHEST	12
	THREAD_PRIORITY_TIME_CRITICAL	15
HIGH_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	11

DOI: 10.15680/IJIRSET.2014.030924

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

	THREAD_PRIORITY_BELOW_NORMAL	12
	THREAD_PRIORITY_NORMAL	13
	THREAD_PRIORITY_ABOVE_NORMAL	14
	THREAD_PRIORITY_HIGHEST	15
	THREAD_PRIORITY_TIME_CRITICAL	15
HIGH_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	1
	THREAD_PRIORITY_LOWEST	11
	THREAD_PRIORITY_BELOW_NORMAL	12
	THREAD_PRIORITY_NORMAL	13
	THREAD_PRIORITY_ABOVE_NORMAL	14
	THREAD_PRIORITY_HIGHEST	15
	THREAD_PRIORITY_TIME_CRITICAL	15
REALTIME_PRIORITY_CLASS	THREAD_PRIORITY_IDLE	16
	THREAD_PRIORITY_LOWEST	22
	THREAD_PRIORITY_BELOW_NORMAL	23
	THREAD_PRIORITY_NORMAL	24
	THREAD_PRIORITY_ABOVE_NORMAL	25
	THREAD_PRIORITY_HIGHEST	26
\	THREAD_PRIORITY_TIME_CRITICAL	31

Table 1 describes about the various process priority classes and their respective priority values.

III. PROPOSED DISK SCHEDULING ALGORITHM

In this section the proposed Disk Scheduling Algorithm is described. The principle of this algorithm is –"The disk requests which has highest priority will be served first". So before applying this algorithm we sort disk requests based on their priority values. According to the priority values each disk request will be served and the head movement is calculated. Like that all the disk requests will be served which are waiting in a disk queue. Finally the head movement is calculated for each served disk requests.

Assumptions

In this algorithm we made following assumptions.

- Each Read/Write request must be assign with some priority.
- The disk request which has assigned highest priority that can be serviced first.
- The disk head is not moving till end of the disk and is moved further towards the request.
- The disk requests can be processed in both directions i.e. either in forward direction and backward direction.

Algorithm

Priority Disk scheduling algorithm (n, p, dr, hc, count, diff, hm)

1. Start
2. //n is the number of disk requests which are arrived at the disk queue.
3. //p [] is an array which stores the priority values of various disk requests.
4. //dr [] is an array which stores the cylinder numbers of different disk requests.
5. //hc is a variable which currently holds the value of current head position.
6. //various disk requests are sorting based on priority value.
7. // diff is a variable which calculates the difference between current head position and high priority cylinder number.
8. for i=0 to n-1do
   8.1 Diff=│hc-dr[i]│
   8.2 hm=hm+diff;
   8.3//End for loop;

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

9. Display hm value;
10. Stop.

IV. EXPERIMENTAL RESULTS

In this section we will discuss about experimental results of the proposed Disk Scheduling Algorithm with an example. And also, we have mentioned the comparison results of the proposed Disk Scheduling Algorithm with the existing Disk Scheduling Algorithms like FCFS, SSTF, SCAN, CSCAN, LOOK, and CLOOK.
Table-2 consists of the information about various disk requests on various cylinder numbers and their priority values which are currently waited in a disk queue. And the current head position is 53.

TABLE 2: Disk Requests.

Cylinder No	98	183	37	122	14	124	65	67
Priority	6	8	1	5	2	7	3	4

Table-3 consists of the information about the sorted disk requests based on their priority values.

TABLE 3: Sorting Disk Requests Based on Priority.

Cylinder No	37	14	65	67	122	98	124	183
priority	1	2	3	4	5	6	7	8

The Figure-1 shows, graphical representation of various disk scheduling algorithms for the example which is mentioned in table-2. The Figure 1-(a),(b),(c),(d),(e),(f) shows the order of servicing disk requests according to the FCFS,SSTF,SCAN,C-SCAN,LOOK,C-LOOK respectively.

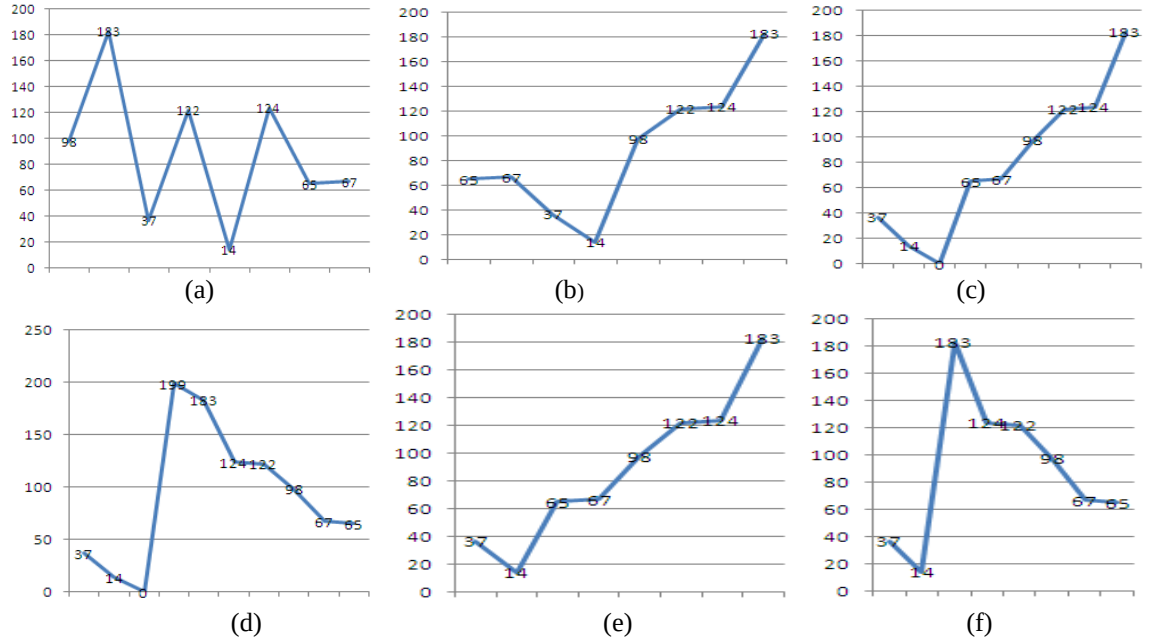

Figure 1: Graphical Representation of various disk scheduling algorithms.

Figure-2 shows the graphical representation of proposed disk scheduling Algorithm i.e. "Priority Based Disk Scheduling Algorithm" and the order of disk requests which are get serviced.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

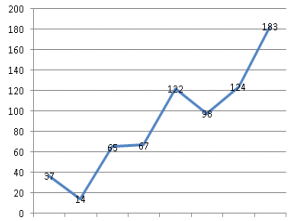

Figure 2: Priority Based Disk Scheduling.

Table-4 shows the details about head movements of different disk scheduling algorithms including "Priority disk scheduling algorithm".

Table 4: Comparison of Disk scheduling Algorithms

Name of the Disk Scheduling Algorithm	Total Head Movements
FCFS	640
SSTF	216
SCAN	236
CSCAN	386
LOOK	208
CLOOK	346
PRIORITY	300

Limitations

A major problem with Priority Disk Scheduling algorithm is "indefinite blocking or starvation". That is a request which has lowest priority that can be waited for indefinite time period to make process. Any way we can overcome this problem of indefinite blockage of low priority request with aging. Aging is a technique of gradually increasing the priority of disk requests which is waiting for a long time.

V. CONCLUSIONS

In this paper we have proposed a new Disk Scheduling Algorithm named as "Priority Based Disk Scheduling Algorithm" which works purely on the basis of priority. According to this, a disk request which is having highest priority can be served first. So in this way all the high priority disk requests can be served before, when compared with the low priority disk requests. Perhaps the disk request which is having highest priority is not waiting for so much time to do service. It is proved very easily that this scheduling algorithm is better than some of the existing algorithms.

REFERENCES

[1] W. Stallings, "Operating Systems", 4th Edition. , Pearson Education Pvt. Ltd., 2007, ISBN 81-7808-503-8.
[2] Robert Geist, Stephen Daniel, "A Continuum of Disk Scheduling Algorithms" ACM Transactions on Computer Systems" Vol. 5, No. 1, Feb 1987, pp. 77-92
[3] A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating System Concepts", 7th Edition. , John Wiley and Sons Inc, 2005, ISBN 0-471-69466-5.
[4] L.F. Bic and A.C. Shaw," Principles of Operating Systems", Prentice-Hall/Pearson Education, 2003.
[5] A.S. Tanenbaum, "Modern Operating Systems", Prentice-Hall, 1996.
[6] M.Y.Javed,"simulation and performance of four disk scheduling algorithms", IEEE TENCON-2000, Vol- 2, pages10-15, ISBN 0-7803-6355-8.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2014

BIOGRAPHY

C. Mallikarjuna[1] holds two master degrees, first one in Computer Applications, and the other in Computer Science Engineering, currently working as Assistant Professor in the Department of M.C.A, Annamacharya P.G college of Computer Studies, Rajampeta, Kadapa Dist., A.P.

K. Sudheer Babu[2] has completed master degrees in both Information Technology, and Computer Science Engineering. At present working as Assistant Professor in the Department of M.C.A, Annamacharya P.G college of Computer Studies, Rajampeta, Kadapa Dist., A.P.

Dr. P. Chitti Babu is currently working as Professor & Principal, Annamacharya PG College of Computer Studies, Rajampet, AndhraPradesh. He received his Master's Degree in Computer Applications (MCA) from SV University, Tirupathi. He has completed M.Tech in Computer Science &Engineering from Acharya Nagarjuna Univeristy, Guntur, India. He has successfully received his Ph.D(ComputerScience) from S.V. University. He has a total of 14 years ofexperience. He has published 20 papers in International and National Journals. He is a life member of CSI, IACSIT, IAENG and ISTE. Presently he is guiding 2 students for research under Ph.D programme.