Optimized Scheduling Algorithm

Lalit Kishor
Dinesh Goyal
Suresh Gyan Vihar University, Jaipur, Rajasthan, India

Rajendra Singh
Suresh Gyan Vihar University, Jaipur, Rajasthan, India

Praveen Sharma
Suresh Gyan Vihar University, Jaipur, Rajasthan, India

ABSTRACT
Scheduling means a set of policies and mechanisms to control the order of work to be performed by a computer system. The basic idea behind scheduling is to keep the CPU busy as much as possible by executing a process until it must wait for an event and then switch to anther process. In this paper, we discuss & analyze various types of Scheduling & try to obtain a better solution to get more throughputs with less CPU utilization. First Come First Served (FCFS) is a non-preemptive, simplest scheduling. FCFS performs better for long job. Shortest Job First (SJF) scheduling selects that job first which has shortest processing time. Round Robin (RR) scheduling removes the drawbacks of FCFS by preempting running jobs periodically. But if the length of quantum is too short then more time will waste in context switching. In Priority Based scheduling each process is assigned a priority.

Keywords
Scheduling, FCFS Scheduling, Shortest Job First Scheduling, Round Robin Scheduling, Priority Based Scheduling.

1. INTRODUCTION TO SCHEDULING
"The scheduler is a component of an operating system that determine which process should be run and when."

Scheduling refers to a set of policies and mechanisms to control the order of work to be performed by a computer system. Of all the resources in a computer system that are scheduled before use, the CPU is by far the most important. Multiprogramming is the (efficient) scheduling of the CPU. The basic idea is to keep the CPU busy as much as possible by executing a (user) process until it must wait for an event, and then switch to another process.

The Scheduling criteria are:

- **CPU utilization**:-keep the CPU as busy as possible
- **Throughput:-** number of processes that complete there execution per unit time
- **Turn around time:-**amount of time to execute a particular process.
- **Waiting time:** - amount of time a process has been waiting in the ready queue.
- **Response time:-**amount of time it takes from when a request was submitted until the first response is produced not output.

Scheduling can be divided into two categories:

1.1 Non Preemptive
A non preemptive scheduling algorithm picks a process to run and then just lets its run until it blocks or until it voluntarily released by CPU. E.g.: FCFS, SJF.

1.2 Preemptive
In this type of scheduling execution of process is preempted before the completion of the burst time of process and any other process may starts its execution whose priority is higher than the first arrived process in system. E.g.: Round Robin, Priority Driven.

2. FIRST COME FIRST SERVE (FCFS)
This is a non preemptive scheduling algorithm. FIFO strategy assigns priority to processes in the order they request the processor. The process that requests the CPU first is allocated the CPU first. FCFS, also known as first-In- First-Out (FIFO), is the simplest scheduling policy. Arriving jobs are inserted into the tail (rear) of the ready queue and the process to be executed next is removed from the head (front) of the queue. FCFS performs better for long jobs. Relative importance of jobs measured only by arrival time (poor choice).

Let's take three process that arrive at same time in this order

Process	CPU burst time(ms)
P1	24
P2	3
P3	3

Gantt chart

P1	P2	P3

P1's waiting time = 0

P2's waiting time = 24

P3's waiting time = 27

Average waiting time= (0+24+27)/3=17milliseconds

Turn around time =burst time +waiting time

P1's turn around time=24+0=24

P2's turn around time=3+24=27

P3's turn around time=3+27=30

Average turn around time = (24+27+30)/3=27milliseconds

Limitation: - the average waiting under a purely first in first out system is going to often be poor if one task is significantly longer than others. This is not fit for time-sharing system, where each user needs to get a share of the CPU at regular intervals.

In a multi-task system, several processes are kept in the main memory and processor is kept active to run a process while the others are waiting. The key to Multi- Programming is scheduling. The Multi-Layer Queue (MLQ) scheduling partitions the queue into several separate queues. Each queue has its own scheduling algorithm. Each process can be easily classified into groups based on several properties of the process and permanently assigned to one queue. In the MLFQ scheduling, the processes can be dynamically moved in different queues. So processes that need a large amount of CPU time are sent to the low priority queues and process requiring I/O bound or related to interactive processes are sent to high priority queues. The MLFQ scheduling organizes the queues to minimize the queuing delay and optimize the queuing environment efficiency. [01]

3. SHORTEST JOB FIRST (SJF)

In SJF waiting job or process with smallest estimated run time to completion is run next. The main advantage is that waiting time is minimal. The SJF is especially appropriate for the batch jobs for which the run time are known in advance. SJF can classified in two schemes non preemptive and preemptive.

Non preemptive SJF:-Once CPU is given to the process it cannot be preempted until completes its CPU burst.

Let four processes that arrive at same time in this order.

Process	CPU time needed(ms)
P1	6
P2	8
P3	7
P4	3

Grant Chart:-

P4	P1	P3	P2

0 3 9 16 24

Average turnaround time = (9+24+16+3)/4=13 milliseconds

Preemptive (SRTF):-

If a new process arrives with CPU burst length less than remaining time of current executing process a preemptive SJF will preempt the currently executing time of current executing process. This scheme is known as the shortest remaining time first (SRTF).

Let we have four processes arrive at different time as shown below

process	Arrival time	Burst time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

process	Burst time(ms)
P1	53
P2	17
P3	68
P4	24

Gantt chart:-

P1	P2	P4	P1	P3

P1's waiting time = 0+ (10-1) =9ms

P2's waiting time = 0 ms

P3's waiting time = 17-2=15ms

P4's waiting time = 5-3=2 ms

Average waiting time= (9+0+17+2)/4=6.5milliseconds

Turn around time =burst time +waiting time

P1's turn around time=8+9=17ms

P2's turn around time=4+0=4ms

P3's turn around time=9+15=24ms

P4's turn around time=5+2=7ms

Average turn around time = (17+4+24+7)/4=13 milliseconds

Limitations :-The difficulty with algorithm is to know which incoming process is indeed shorter than another. There is no way to know the length of the next CPU burst so this type of algorithm is largely impossible. Also, long running jobs may starve, because the CPU has a steady supply of short jobs.

There are two general policies often used for the scheduler: longest-job-first (LJF) and shortest-job- first (SJF). The LJF is a priority scheme and always provides an advantage of balancing the loads in data channels, but is known to result in relatively poor average packet delay. The SJF, on the contrary, is also a priority scheme that can reduce the average packet delay by scarifying the feature of load balancing. Schemes of LJF and SJF may be the tradeoff in the performance of loading balance and average delay. At the first glass, it seems to be independent of the effect between the two performance requirements. In fact, they are causal in each other for a long term. The inferior balance scheduling will reduce the bandwidth utilization and cause larger average packet delay in each node. Therefore, a good scheduler should consider the effects of loading balance and average delay. In this paper, based on SJF and LJF, two different scheduling algorithms will be proposed to achieve low access delay and load balancing. [02]

4. ROUND ROBIN (RR)

In this approach a time slot is defined which is a particular small unit of time. In each time slice the CPU executes the current process until the end of time slice. If that process is done it is discarded and the next one in queue is deal with. However if the process is not done it is halted and put at the back of the queue and then the next process in line is addressed during the next time slice. RR reduces the penalty that short jobs suffer with FCFS by preempting running jobs periodically. The main advantage is that every process gets the CPU and thus there is no starvation.

Let's take four processes that arrive at the same time in this order and time quantum is 20

Gantt chart:-

P1	P2	P3	P4	P1	P3	P4	P1	P3	P3

0 20 37 57 77 97 117 121 134 154 162

P1's waiting time = 0+57+24=81

P2's waiting time = 20

P3's waiting time = 37+40+17=94

p4's waiting time = 57+40=97

Average waiting time= (81+20+94+97)/4=73milliseconds

Turn around time =burst time +waiting time

P1's turn around time=53+81=134

P2's turn around time=17+20=37

P3's turn around time=68+94=162

P4's turn around time=24+97=121

Average turn around time = (134+37+162+121)/4=113.5 milliseconds

Limitation:-this method vastly slow down short processes because they have to share the CPU time with other processes instead of just finishing up quickly. So the critical issue with the RR policy is the length of the quantum. If it is too short, then the CPU will be spending more time on context switching. Otherwise, interactive processes will suffer.

A new scheduling algorithm named DRRA (Deficit Round Robin Alternated) that achieves an improvement in the uniform distribution of the output traffic respect to the existent DRR (Deficit Round Robin) algorithm. This work is embraced into a line of research that intends to provide Ethernet users with the quality of service of ATM. To achieve this, an architecture called "Cells in Frame" (CIF) is used, which allows ATM cells

to be carried in Ethernet frames. An implementation of this architecture has been made using the DRRA algorithm. Some other algorithms have also been deployed in order to contrast them with the proposed one. [03]

Airtime Deficit Round Robin (ADRR), a novel scheduling discipline aiming at providing intra-cell airtime fairness as opposed to the bandwidth fairness provided by traditional scheduling policy, i.e. Fair Queuing or, in case of equally sized data packets, Round- Robin. ADRR enhances the Deficit Round Robin (DRR) scheduling discipline by taking into account the channel quality experienced by the transmitting node. The devised algorithm addresses the "IEEE 802.11 performance anomaly", preventing a node affected by high packet losses from monopolizing the wireless channels lowering the performance of the whole system. [04]

5. PRIORITY BASED SCHEDULING

In this scheduling algorithm each process has a priority associated with it and as each process hits the queue it is sorted in based on its priority so that process with higher priority are dealt with first. It should be noted that equal priority process are scheduled in FCFS order. The main advantage is the important jobs can be finished earlier as much as possible.

For example consider the following set of processes with given priorities and burst time assumed to arrive at time 0

Process	Priority	Burst time
P1	3	8
P2	2	2
P3	1	14
P4	4	6

Gantt chart:-

P3	P2	P1	P4

0 14 16 24 32

P1's waiting time = 16

P2's waiting time = 14

P3's waiting time = 0

p4's waiting time = 24

Average waiting time= (16+14+0+24)/4=13.5milliseconds

Turn around time =burst time +waiting time

P1's turn around time=6+16=22

P2's turn around time=2+14=16

P3's turn around time=14+0=14

P4's turn around time=6+24=30

Average turn around time = (22+16+14+30)/4=20.5 milliseconds

Limitation :-The problem occur when operating system gives a particular task very low priority so it sits in queue for a larger amount of time ,not being dealt with by the CPU.

Mr. T. Funkhouser and Mr. P. Shilane in their paper introduces a priority-driven algorithm for searching all objects in a database at once. The algorithm is given a query object and a database of target objects, all represented by sets of local shape features, and its goal is to produce a ranked list of the best target objects sorted by how well any subset of k features on the query match features on the target object. To achieve this goal, the system maintains a priority queue of potential sets of feature correspondences (partial matches) sorted by a cost function accounting for both feature dissimilarity and geometric deformation. Initially, all pair wise correspondences between the features of the query and features of target objects are loaded onto the priority queue. [05]

6. COMPARISION OF SCHEDULING

The given table gives a comparison between various types of scheduling according to given parameters.

Scheduling algorithm	CPU Utilization	Throughput	Turnaround time
FCFS	Low	Low	High
SJF	Medium	High	Medium
RR	High	Medium	Medium
PRIORITY	Medium	Low	High

7. PROPOSED MODEL

We are arrange the Process according to the short burst time in ascending order then we Calculate time Quantum for ROUND ROBIN by this Formula.

TIME QUANTUM=Average Burst Time.

For example consider the following set of processes:

process	Burst time(ms)
P1	10
P2	15
P3	02
P4	08

We arrange all Process according to shortest job First

process	Burst time(ms)
P3	02
P4	08
P1	10
P2	15

Then we Calculate Time Quantum by Formula

TIME QUANTUM= 35/4=08

First we Solve by Round Robin Algorithm with Time Quantum=8

Gantt chart:-

P1	P2	P3	P4	P1	P2

0 8 16 18 26 28 35

Average Waiting Time= (18+20+16+18)/4=18ms

Average turnaround time = (28+35+18+26)/4=26 ms

Solve By Round Robin with SJF:-

Gantt chart:-

P3	P4	P1	P2	P1	P2

0 2 10 18 26 28 35

Average Waiting Time= (0+2+18+20)/4=10ms

Average turnaround time = (2+10+28+35)/4=18 ms

8. CONCLUSION

In this paper first of all we have discussed about scheduling and then various types of scheduling with the examples. A comparison of various types of algorithms is also shown. By examining the advantages and drawbacks of various algorithms we can develop a new scheduling algorithm which can solve drawbacks of discussed algorithms

We Increase the Performance, Throughput and decrease the Turnaround Time by solves Round Robin with SJF.

9. ADVANTAGES

1. Shortest job are completed in one time.
2. Efficiency:-medium.
3. Turn around time:-Low (comparison to Round Robin Algorithm)

10. REFERENCES

[1] MohammadReza EffatParvar, Karim Faez, Mehdi EffatParvar, Mehdi Zarei, Saeed Safari. An Intelligent MLFQ Scheduling Algorithm (IMLFQ) with Fault Tolerant Mechanism

[2] Shiann-Tsong Sheu, Yue-Ru Chuang and Hsuen-Wen Tseng. A Pipelined Scheduling Strategy for Supporting Variable-Length Packets in WDM Networks.

[3] J. M. Arco D. Meziat B. Alarcos. Deficit Round Robin Alternated: A New Scheduling Algorithm [04]. Roberto Riggio, Daniele Miorandi, and Imrich Chlamtac. Airtime Deficit Round Robin (ADRR) Packet Scheduling Algorithm_

[4] T. Funkhouser and P. Shilane. Partial Matching of 3D Shapes with Priority-Driven Search