A Case Study: Analysis of Different Scheduling Algorithm of CPU Scheduling

Palak Bhavsar[1] Mrs. Shilpa Serasiya[2]
[1]M.E. Student [2]Professor
[1,2]Department of Computer Science & Engineering
[1,2]KITRC College, Kalol, Gujarat, India

Abstract— CPU is heart of the computer system and to manage that component scheduling is used. Scheduling gives proper resources to the CPU for execution, so you can say that CPU Scheduling is one of the primary part of the CPU, which allocate CPU time to the Processes. Multiprogramming operating system can execute multiple processes concurrently and for that different algorithm are available for this kind of system in which there are FCFS, SJF, Priority and RR included. In this paper we will discuss those entire algorithms with their state diagram and also perform them with some bunch of processes. Moreover that we mention some work related to those algorithms in recent years those work are done on the idea of basic scheduling algorithms. So, our goal is to analyze and compare all those algorithm which fulfills different Scheduling goals.

Key words: CPU Scheduling, Scheduling Algorithm

I. Introduction

Operating system is interface between user and computer hardware system from that all of us are familiar now a days. In operating system there are basically two type of Uni-processor, one is Uni-programming which executes only single job at a time, while another one is Multi-programming which can handle multiple jobs concurrently, therefore all jobs are scheduled before use [1][6].

Multiprogramming can executes many processes simultaneously all those active processes needs CPU time for execution. So, operating system is responsible to apply a strategy which can meet the demand of various CPU processes [4].

The Scheduling of processes put very much impact on CPU's efficiency and performance there for it should be done fairly and correctly, so that each and every process gets chance to execute on CPU. The objective of this task is to maximize CPU utilization its efficiency hence the performance [5]. Now a day CPU Scheduling fundaments are useful in each and every technology like Bluetooth, WSN, Cloud, etc. in order to improve the performance of application.

II. Theory

A. CPU Scheduling:

We can consider a process is more than a program code, so as a process executes, it changes state. In Figure 2.1, different states of a process are shown [3]. For process execution OS feature three different types of scheduler, first one is Long term Scheduler which is also known as admission scheduler, admits a job or process to the ready queue. Second one is medium-term or med-term scheduler, which removes process from main memory to secondary memory, this scheme is called as Swapping, so, the process is Swapped Out and later in can be Swapped In. and the third one is Short-term Scheduler, which selects the process from the ready queue to be executed.

In Scheduling a queue where all processes are ready to run and waiting for CPU is called Ready Queue. Any process who wants to execute need CPU time and for that it must enter into Ready Queue [4].

B. Criteria for Scheduling:

1) CPU Utilization:

To keep the CPU as busy as possible by multiprogramming and its utilization rate should be 100%.

2) Fairness:

Make sure each Process get a fair share of the CPU time.

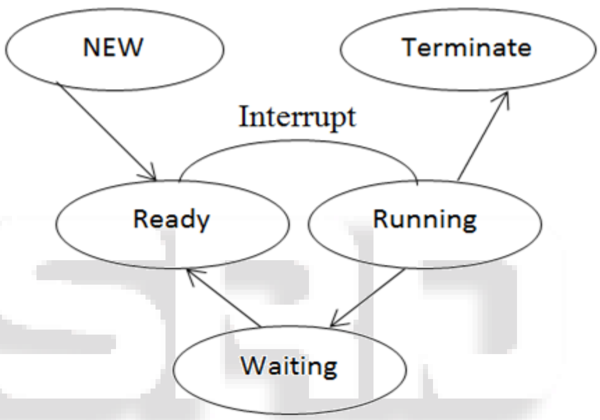

Fig. 2.1: Different States of Process

3) Throughput:

No. of processes that are completed per unit of time called Throughput.

4) Response Time:

After submission of a request until first response is produced from CPU

5) Turnaround Time:

The time interval between submission of a process and completion of a process

6) Waiting Time:

Sum of time periods which process spent waiting in Ready Queue

C. Scheduling Algorithms:

CPU Scheduler can work properly if the CPU have high quality design algorithm, which have criteria like improving CPU Utilization, Throughput, Turnaround time, waiting time and response time. Below State diagram of different algorithm [2] gives idea about working of them.

1) First Come First Serve:

FCFS in fig. 2.2 is the simplest scheduling algorithm from all, in which ready queue is maintain as FIFO queue [6]. In FCFS CPU allocation is done on the basis of which process request for it first, and new processes enter to the tail of the queue [1]. Any process does not release the CPU until its execution gets completed.

2) Shortest Job First:

SJF algorithm in fig. 2.3 associates with the length of particular process, means which process has shorter burst time executes first. SJF could be Pre-emptive or Non-preemptive [1].

3) Priority Scheduling:

When more than one process is at runnable state, the operating system has to decide which one runs first. At that time Scheduler takes that decision and give them priority for CPU time this algorithm is called Priority Scheduling, in fig. 2.4 which highest priority from 0 to 9 allowed to run first [3].

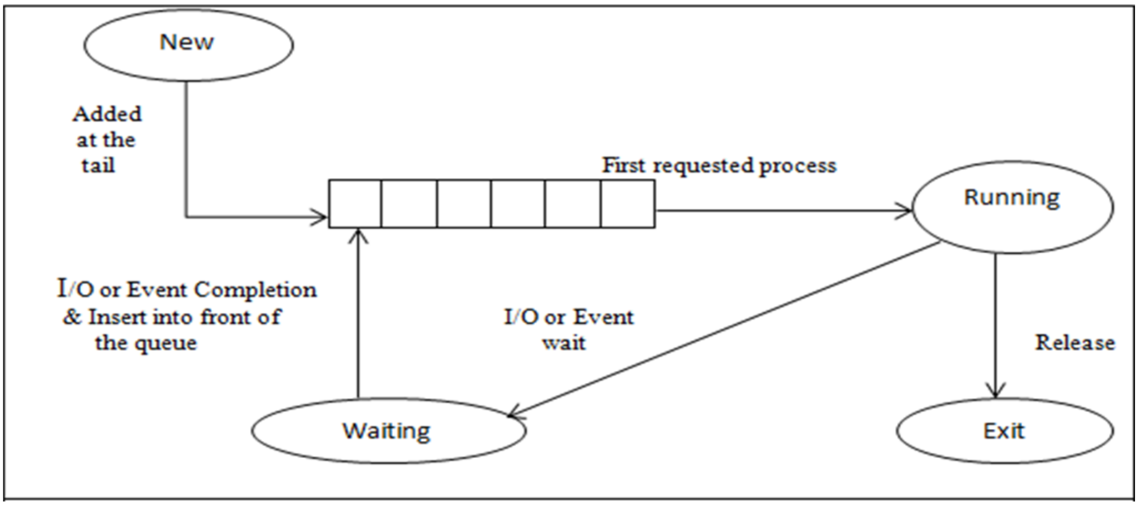

Fig. 2.2: First Come First Serve State Diagram

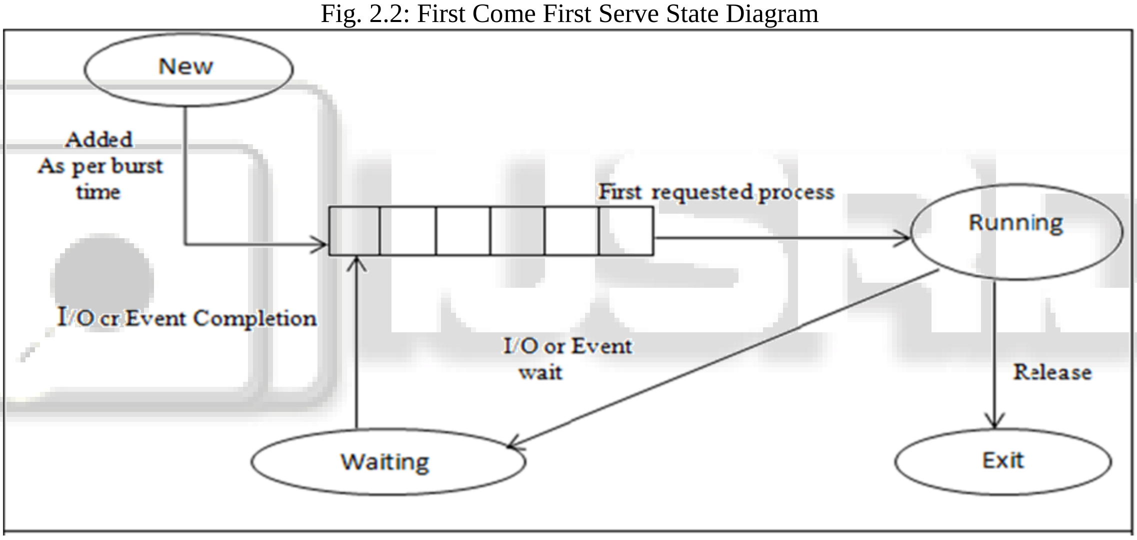

Fig. 2.3: Shortest Job First State Diagram

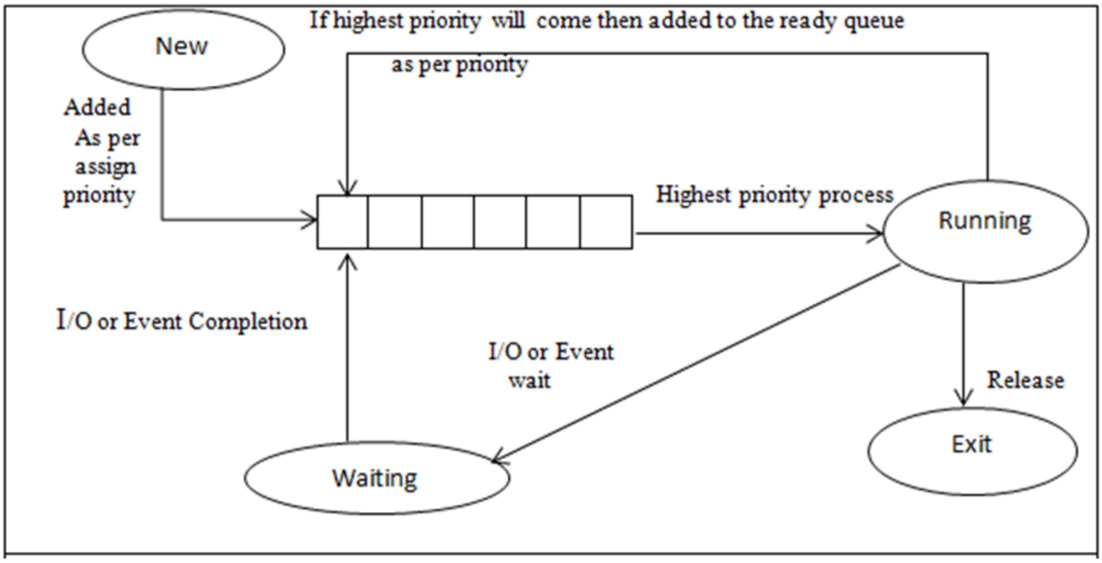

Fig. 2.4: Priority Scheduling State Diagram

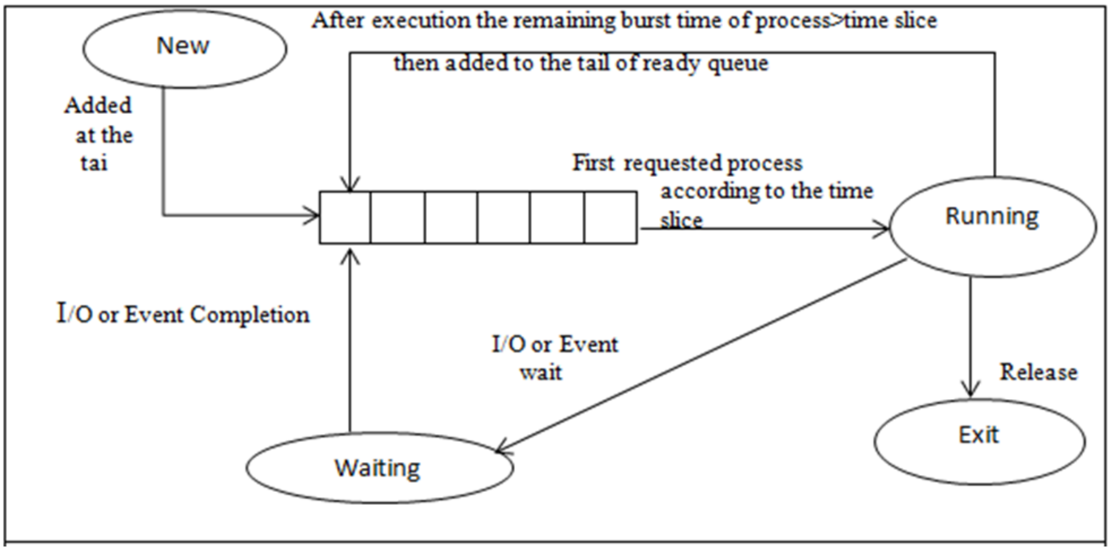

Fig. 2.5: Round Robin State Diagram

4) Round Robin:

Round robin algorithm fig. 2.5 designed especially for time-sharing system, which is simplest, fairest and most widely used one [3]. Round Robin is similar to the FCFS with little twist, here a small unit of time called Time Slice or Time Quantum is define. RR will allow first process for execution till the time slice, if that process have some more burst time remain at that time that process have to wait till next turn, then CPU will choose next process from ready queue for execution. Note that ready queue in RR is circular queue [1][4].

III. RELATED WORK

MS. Tabassum A. Maktum in [5] combines FCFS and SJF with Genetic approach and produce new algorithm for Process Scheduling. Sukumar in [6], take two new algorithm called Anovel CPU Scheduling algorithm Pre-emptive, Non-preemptive and Efficient CPU Scheduling algorithm, and compare them with FCFS, SFJ, Round Robin and Priority Scheduling. Neeraj Kumar in [7] tries to improve performance of their algorithm with respect to turnaround and waiting time factor. Average Max Round Robin Scheduling Algorithm (AMRR) in [8], An Effective Round Robin Algorithm using Min-Max Dispersion Measure [9], Ascending Quantum and Minimum-Maximum Bust Time Algorithm [10], use Dynamic Time Quantum or Time Slice to reduce the value of Average Turnaround time and Average Waiting time and also compare them with simple Round Robin Algorithm

IV. PERFORMANCE EALUATION

Consider the following set of processes Shown in Table 4.1 form [1], with Process No., Burst Time and Priority, where burst time is given in millisecond and processes are assumed to have arrived in order of P1, P2, P3, P4 and P5 at 0 time and Time Quantum is 5 millisecond, and small priority number implies high priority.

According to these processes Gantt Charts are shown below in fig 4.1, 4.2, 4.3, 4.4for particular algorithm and calculate their waiting time and turnaround time in Table 4.2, 4.3 respectively for each process with Average Turnaround time (ATT) and Average Waiting Time (AWT).

Process No.	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Table 4.1: List of Process No., Burst Time, Priority

1) First Come First Sever:

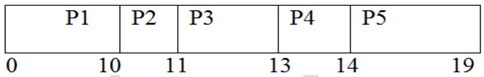

Fig. 4.1: Gantt chart for FCFS

2) Shortest Job First:

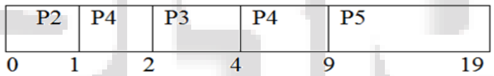

Fig. 4.2: Gantt chart for SJF

3) Priority Scheduling

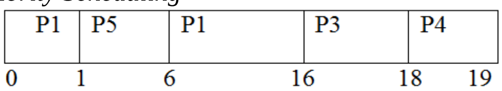

Fig. 4.3: Gantt chart for Priority

4) Round Robin (Time Quantum = 5)

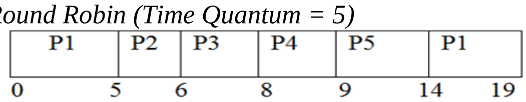

Fig. 4.4: Gantt chart for RR

Process No.	Turnaround Time			
	FCFS	SJF	Priority	RR
P1	10	19	16	19
P2	11	1	1	6
P3	13	4	18	8
P4	14	2	19	9
P5	19	9	6	14
ATT	13.4	7.0	12.0	11.2

Table 4.2: Turnaround Time of all Algorithms

Process No.	Waiting Time			
	FCFS	SJF	Priority	RR
P1	0	9	6	9
P2	10	0	0	5
P3	11	2	16	6
P4	13	1	18	8
P5	14	4	1	9
AWT	9.6	3.2	8.2	7.4

Table 4.3: Waiting Time of all Algorithms

V. CONCLUSION

By analyzing those basic algorithms of CPU Scheduling we conclude that SJF performed best in terms of Waiting Time and Turnaround Time, but drawback of SJF is, it occurs starvation during long processes. At the another hand RR gives the second result in terms of Waiting time, Turunaround time, and it is best for Time-sharing system. So, in future we will Proposed a new method which is based on RR and add some essence of SJF Algorithm, with which we could improve the result of AWT and ATT.

REFERENCES

[1] Silberschartz, Galvin and Gagne, Operating System concept, 8th Edition, Wiley, 2009.

[2] Neetu Goel, Dr. R. B. Garg, " A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics & Image Processing, Vol 2, November 2012, pg. no. 245-251.

[3] Ashok Kumar, Dr. Harish Rohil, Suraj Arya, "Analysis of CPU Scheduling Policies through Simulation", Volume 3, Issue 5, May 2013.

[4] Neetu Jain and P. V. Suresh, "A Java Based Visual Tool to Learn CPU Scheduling Algorithms", 978-93-80544-12-0/14/$31.00_c 2014 IEEE.

[5] Ms. Tabassum A. Maktum, Ms. Rashmi A. Dhumal, Dr. Lata Ragha, "A Genetic Approach for Processor Scheduling", 978-1-4799-4040-0/14/$31.00 ©2014 IEEE.

[6] Sukumar Babu B., Neelima Priyanka N. & Dr. P. Suresh Varma, "Optimized Round Robin CPU Scheduling Algorithm", Volume 12 , Issue 11, Version 1.0, Year 2012.

[7] Neeraj Kumar, Nirvikar "Performance Improvement Using CPU Scheduling Algorithm-SRT", Volume 2, ISSN 2278-6856, March – April 2013, pg.no. 110-113.

[8] Pallab Banerjee, Probal Banerjee, Shweta Sonali Dhal, "Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum", Volume-1, Issue-3, August 2012.

[9] Sanjaya Kumar Panda, Sourav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure", Vol. 4, January 2012.

[10] Ali Jbaeer Dawood, "Improving Efficiency of Round Robin Scheduling Using Ascending Quantum And Minumim-Maxumum Burst Time", Vol.6:NO.2, 2012

