Analysis of CPU Scheduling Policies

Khin Po
Faulty of Computer Science
University of Computer Studies, Pinlon
dawkhinpopo@gmail.com

Abstract

In a multiprogramming system, multiple processes exist concurrently in main memory. Each process alternates between using a processor and waiting for some event to occur, such as the completion of an I/O operation. The processor or processors are kept busy by executing one process while the others wait. The key to multiprogramming is scheduling. The aim of processor scheduling is to assign to execute by the processors over time, in a way that meets system objectives, such as response time, throughput and processor efficiency. This paper analysis the four scheduling policies First-Come-First-Served (FCFS), Round-Robin (RR), Shortest Process Next (SPN) and Shortest Remaining Time (SRT) and gives the result which policy is the best among them. A comparative analysis is made on the basis of data generated through the simulation using exponentially generated random numbers.

Keywords: Scheduling policies, I/O operation, Throughputs, Response Time, Multi-pro gramming.

1. Introduction

Operating system (OS) is system software which acts as an interface between a user and the computer hardware. OS is also known as resource manager because its prime responsibility is to manage the resources of the computer system. Scheduling is a fundamental and most important OS Function which is essential to an operating system's design. Scheduling refers to set of rules, policies and mechanism that govern the order in which resource is allocated to the various processes and the work is to be done.

The scheduling is a methodology of managing multiple queues of processes in order to minimize delay and to optimize performance of the system. A scheduler is an OS module that implements the scheduling policies. Its primary objective is to optimize the system performance according to the criteria set by the system designers. It selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them. Multiprogramming, Multiuser, Multitasking, Multi-processing, and Multithreading are some attractive features for OS designer to improve the performance. In a single-processor system, only one process can run at a time; any others must wait until the CPU is free and can be rescheduled. In the multiprocessing, is to have some process running at all times, to maximize CPU utilization. Almost all computer resources are scheduled before use. In multiuser and multi-programming environment also has multiple users queuing processes simultaneously to provide best performance to all users. So Controlling and managing CPU for all process requests.

Different scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. Consider the properties of the various algorithms to choosing in a particular situation. Many criteria have been suggested for comparing scheduling algorithms judging to be best. CPU Scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. It is the act of selecting the next process for the CPU to "service" once the current process leaves the CPU idle that decisions may take place among the processes. Many algorithms for making this selection from these seven classical kind of scheduling algorithms like First Come-First Served (FCFS), Shortest Process Next (SPN), Shortest Remaining Time (SRT), Round-Robin (RR).

2. Related Work

Jain et al. [5] presented a Linear Data Model Based Study of Improved Round Robin CPU Scheduling algorithm with features of Shortest Job First scheduling with varying time quantum. **Lulla** et al. **[7]** developed a new approach for round robin CPU scheduling algorithm which improves the performance of CPU using Dynamic Time Quantum. **Banerjee et al. [3]** proposed a new algorithm called Optimized Performance Round Robin (OPRR) in which focused on dynamic time quantum which give result as a very less context switching as well as average waiting time and average turnaround time and also reduces the overhead of the CPU by adjusting the time quantum according to the highest burst time of the processes in the ready queue.

Rao et al. [10] proposed a new algorithm which is a logical extension of the popular Round Robin CPU scheduling algorithm suggests that a priority be assigned to each process based on balanced precedence factor using mean average as a time quantum conducting experiments to measure the effectiveness of this novel method that showed EPSADTQ is superior to RR and PSMTQ and its variants. **Abdulrahim et al. [1]** proposed algorithm compared with the other algorithms, produces minimal average waiting time (AWT), average turnaround time (ATAT), and number of context switches that adopt RR CPU scheduling.

Sukhija et al. [11] proposed a new-fangled CPU scheduling algorithm called MIN-MAX which behaves as both preemptive and non-preemptive algorithm basis on the burst time to improve the CPU efficiency in multiprogramming OS and also trims down the starvation problem among processes and focused on the comparative study of the existing algorithms on basis of various scheduling parameters.

Panda et al. [9] considered different time quantum for a group of processes and reduced context switches as well as enhancing the performance of RR algorithm, calculated time quantum using min-max dispersion measure and showed experimental analysis that Group Based Time Quantum (GBTQ) RR algorithm performance better than existing RR algorithm.

Suranauwarat [12] used simulator in operating system to learn CPU scheduling algorithms in an easier and a more effective way. **Sindhu et al. [13]** proposed an algorithm which can handle all types of process with optimum scheduling criteria. **Terry Regner & Craig Lacey [14]** has introduced the concepts and fundamentals of the structure and functionality of operating systems. The purpose of this article was to analyze different scheduling algorithms in a simulated system. This article has the implementation of three different scheduling algorithms: shortest process first, round robin, and priority sequence. Comparing the three algorithms they find that the CPU utilization values indicate that the shortest process first has the highest throughput values with CPU utilization times comparable to those of the round robin.

Nazleeni Samiha Haron et.al. [8] has analyzed distributed systems, process scheduling plays a vital role in determining the efficiency of the system. Process scheduling algorithms are used to ensure that the components of the system would be able to maximize its utilization and able to complete all the processes assigned in a specified period of time.

Ajit Singh [2] has developed a new approach for round robin scheduling which help to improve the CPU efficiency in real time and time sharing operating system. **Lalit [6]** discussed about various types of scheduling. A comparison of various types of algorithms is also shown with practical implementation using MATLAB. By this experimental setup he has been able to do statistical analysis of the performance of all the four basic scheduling algorithms.

3. CPU Scheduling

Scheduling can be defined as a mechanism or a tool to control the execution of number of processes performed by a computer. CPU is the most important of all the resources available in a computer system that are scheduled before use; Multiprogramming is attained by efficient scheduling of the CPU. The basic idea is to keep the CPU busy as much as possible by executing a process, and then switch to another process. The aim of processor scheduling is to assign processes to be executed by the processor or processors over time, in a way that meets system obectives, suchas response time, throughput and processor efficiency.

3.1 Goals for CPU Scheduling

To make sure that scheduling strategy is good enough with the following criteria:

- Utilization/Efficiency: keeps the CPU busy 100% of the time with useful work.
- Throughput: maximizes the number of jobs processed per hour.
- Turnaround time: from the time of submission to the time of completion and minimize the time batch users must wait for output.
- Waiting time: Sum of times spent in ready queue.
- Response Time: time from submission till the first response is produced and minimize response time for interactive users.
- Fairness: make sure each

3.2 Types of Scheduling

(1) Long Term Scheduling: The decision as to add to the pool of processes to be executed.

(2) Medium-Term Scheduling: The decision as to add to the numbers of process that are partially or fully in main memory.

(3) Short-Term Scheduling: The decision as to which avaliable process will be executed by the processor.

(4) I/O Scheduling: The decision as to which process's pending I/O request shall be handled by an available I/O devices. Figure 1 relates the scheduling functions to the process state transition diagram.

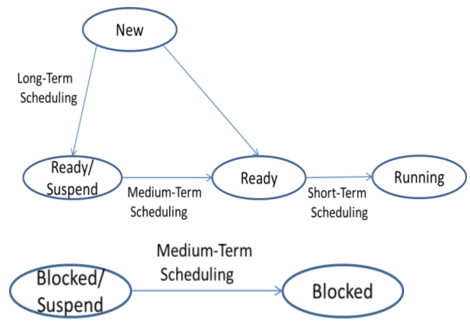

Figure 1.Scheduling and Process State Transitions

Scheduling affects the performance of the system because it determines which processes will wait and which will progress. This point of view is presented in Figure 2, which shows the queues involved in the state transitions of a process.

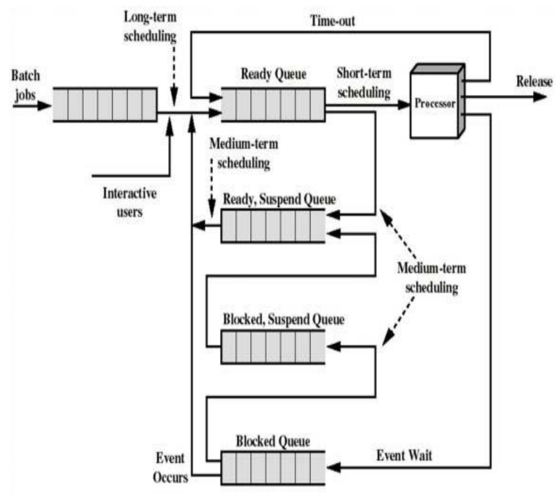

Figure 2. Queuing Diagram for Scheduling

This section has discussed the scheduling policies in short-term scheduling. The short term scheduler also known as the dispatcher executes most frequently and makes the fine-grained decision of which process to execute next. The short term scheduler is involved whether an event occurs that may lead to the blocking of the current process or that may provide an opportunity to preempt a currently running process in favour of another. Examples of such events include

- Clock interrupts

- I/O interrupts
- Operating system calls
- Signal (eg. Semaphores)

3.3 Basic CPU scheduling Algorithms

CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. The Basic CPU Scheduling algorithms are First-Come, First-Served, Shortest Process Next, Round Robin and Shortest Remaining Time.

3.3.1 First-Come-First-Served

The simplest scheduling policy is first come first served (FCFS), also known as first in first out (FIFO) or a strict queuing scheme. As each process becomes ready, it joins the ready queue. When the currently running process causes to execute, the process that has been in the ready queue the longest is selected for running [15].

3.3.2 Round Robin

A straightforward way to reduce the penalty that short jobs suffer with FCFS is to use preemptive based on a clock. The simplest such policy is round robin. A clock interrupt is generated at periodic intervals. When the interrupt occurs, the currently running process is placed in the ready queue and the next ready job is selected on a FCFS basis. This technique is also known as time slicing, because each process is given a slice of time before being preempted [15].

3.3.3 Shortest Process Next

Another approach to reducing the basis in favour of long processes inherent in FCFS is the shortest process next (SPN) policy. This is a non preemptive policy in which the process with the t expected processing time is selected next. Thus, a short process will jump to the head of the queue past longer jobs. One difficulty with the SPN policy is the need to know or at least estimate the required processing time of each process [15].

3.3.4 Shortest Remaining Time

The shortest remaining time (SRT) policy is a preemptive version of SPN. In this case, the scheduler always chooses the process that has the shortest expected remaining processing time. When a new process joins the ready queue, it may in fact have shorter remaining time than the currently running process. According the scheduler may preempt the current processes when a new process becomes ready. As with SPN, the scheduler must have an estimate of processing time to perform the selection function and then is a risk of starvation of longer processes [15].

4. Experimental Setup

How do select a CPU Scheduling algorithm for a particular system? Since different scheduling algorithm have its own parameter selection can be difficult. Therefore, simulation over many CPU scheduling algorithms use in the most real system. Simulation is the imitation of the operation of a real-world process or system over time. Whether done by hand or on a computer, simulation involves the generation of an artificial history of a system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system [4]. To get a more accurate evaluation of scheduling algorithms, this paper can use simulation by hand.

Let's take 5 processes that arrive at random time in the below given order and analysis their performance in various scheduling (namely FCFS, SPN, RR & SRT) algorithms, with given service time and time quantum of 5ms. The simulation by hand is calculation 5 times. Each times a burst time and arrival time is generated by random method. Table (1) to (5) show waiting times for four scheduling policies over 5 processes with randomly arrival and service time.

Table 1. Comparison Waiting Time for Four scheduling policies in Calculation 1.

Pro-cess	Arrival Time	Service Time	WT FCFS	WT RR	WT SPN	WT SRT
P1	0	5	0	0	0	0
P2	4	20	5	5	5	5
P3	15	30	25	10	25	25
P4	26	18	55	15	62	26
P5	35	7	73	50	55	35

Table 2. Comparison Waiting Time for Four scheduling policies in Calculation 2.

Pro-cess	Arrival Time	Service Time	FCFS	RR	SPN	SRT
P1	0	10	0	0	0	0
P2	8	20	10	10	10	10
P3	15	15	30	15	38	38
P4	22	8	45	30	30	30
P5	35	27	53	35	53	53

Table 3. Comparison Waiting Time for Four scheduling policies in Calculation 3.

Pro-cess	Arrival Time	Service Time	FCFS	RR	SPN	SRT
P1	0	8	0	0	0	0
P2	6	20	8	8	8	8
P3	18	12	28	18	28	28
P4	26	25	40	33	55	55
P5	30	15	65	18	40	40

Table 4. Comparison Waiting Time for Four scheduling policies in Calculation 4.

Pro-cess	Arrival Time	Service Time	FCFS	RR	SPN	SRT
P1	0	12	0	0	0	0
P2	10	20	12	10	12	12
P3	18	16	32	20	46	46
P4	25	18	48	25	62	62
P5	27	14	66	30	32	32

Table 5. Comparison Waiting Time for Four scheduling policies in Calculation 5.

Process	Arrival Time	Service Time	FCFS	RR	SPN	SRT
P1	0	8	0	0	0	0
P2	7	20	8	8	8	8
P3	15	22	28	18	58	58
P4	25	17	50	33	28	28
P5	33	13	67	38	45	45

Turnaround time =service time +waiting time

The module has been executed; average waiting time is calculated by each policies are produced. Firstly, average waiting times of four policies will compare. Then, these results show Figure 3 which process gives the minimum average waiting time out of 5 runs.

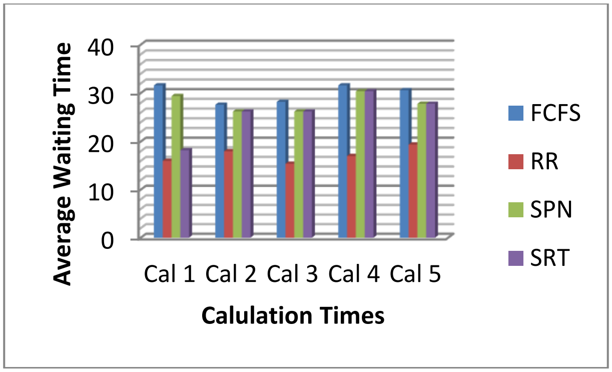

Figure 3. Average Waiting Time Comparison between Four Scheduling Algorithms

Figure 4 shows that these four policies according to the average Turnaround time.

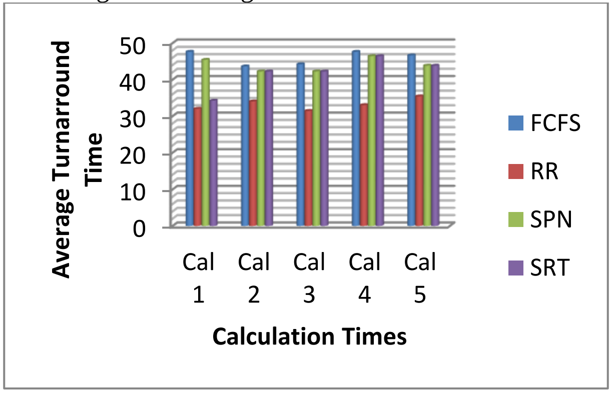

Figure 4. Average Turnaround Time Comparison between Four Scheduling Algorithms

5. Conclusion

The present paper analysis the four CPU scheduling policies via FCFS, RR, SPN, SRT. The comparative analysis was made using simulation by hand. By this experimental setup, statistical analysis of the performance of all the four basic scheduling algorithms has been performed, as stated above. After running and comparing the waiting times and turnaround times, and average waiting time of each scheduling algorithm (FCFS, RR, SPN and SRT), RR result has noticed minimal average waiting time, though encountered maximum waiting times in generating index numbers for FCFS. But, calculation time by hand for RR, SPN and SRT take 5 min per calculation. However, FCFS takes 3min per calculation.

References

[1] Abdulrahim, A., Abdullahi, S., E. and Sahalu J., B. A New Improved Round Robin (NIRR) CPU Scheduling. International Journal of computer Application. March 2014, Vol. 4, No.90, pp. 27-33.

[2] Ajit Singh, Priyanka Goyal ; An Optimized Round Robin Scheduling Algorithm for CPU Scheduling; 2010.

[3] Banerjee, P., Kumari, A. and Jha, P. Comparative performance analysis of optimized performance round robin scheduling algorithm (OPRR) with an based round robin scheduling algorithm using dynamic time quantum in real time system with arrival time. International Journal Computer Science and Engineering (IJCSE), May 2015,Vol. 03, No. 05, pp. 309-316.

[4] Carson J.S., Bank J., and Nelson B.L., (1999) “Discrete-event simulation,” 3rd ed, Prentice Hall international.

[5] Jain, S., Shukla, D. and Jain, R. Linear Data Model Based Study of Improved Round Robin CPU Scheduling algorithm. International Journal of Advanced Research in Computer and Communication Engineering, June 2015, Vol. 4, No. 6, pp.562-564.

[6] Lalit Kishor, Dinesh Goyal; Comparative Analysis of Various Scheduling Algorithms; April 2013.

[7] Lulla, D., Tayade, J. and Mankar, V. Priority based Round Robin CPU Scheduling using Dynamic Time Quantum. International Journal on Emerging Trends in Technology, September 2015,Vol. 2, No. 2, pp. 358-363.

[8] Nazleeni Samiha Haron, Anang Hudaya Muhamad Amin; Time Comparative Simulator for Distributed Process Scheduling Algorithms;2006.

[9] Panda, S., K., Dash, D., and Rout, J., K. A Group based Time Quantum Round Robin Algorithm using Min-Max Spread Measure. International Journal of Computer Applications, February 2013,Vol. 64, No.10, pp.1-7.

[10] Rao, G., S., N., Srinivasu, and S., V., N., Srinivasu, N. and Rao G. R. Enhanced Precedence Scheduling Algorithm with Dynamic Time Quantum (EPSADTQ). Research Journal of Applied Sciences, Engineering and Technology© Maxwell Scientific Organization, 2015, Vol. 10, No.8, pp.938-941.

[11] Sukhija, K., Aggarwal, N. and Jindal, M. An Optimized Approach to CPU Scheduling Algorithm: Min-Max. Journal of Emerging Technologies in Web Intelligence © 2014 Academy Publisher, NOV.2014, Vol. . 6, No. 4, pp. 420-428.

[12] Suranauwarat, S. A CPU scheduling algorithm simulator. IEEE Proceedings (Frontiers in Education Conference - Global Engineering: Knowledge without Borders, Opportunities without Passports, 2007. FIE '07. 37th Annual), pp. 19 – 24.

[13] Sindhu, M., Rajkamal, R. and Vigneshwaran, P. An Optimum Multilevel CPU Scheduling Algorithm. IEEE (International Conference on Advances in Computer Engineering (ACE)), 2010, pp. 90 – 94.

[14] Terry Regner, craig lacy; An Introductory Study of Scheduling Algorithms; Feb 2005.

[15] William Stallings, Operating Systems Internals and Design Principles, 5th Edition.