International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 4 | May-Jun 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

Simulation of Process Scheduling Algorithms

Daw Khin Po

Lecturer, University of Computer Studies, Mandalay, Myanmar

***How to cite this paper**:* Daw Khin Po "Simulation of Process Scheduling Algorithms" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-4, June 2019, pp.1629-1632, URL: https://www.ijtsrd.com/papers/ijtsrd25124.pdf

IJTSRD25124

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. CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. By switching the CPU among processor the operating system can make the computer more productive. Scheduling affectes the performance of the system because it determines which processes will wait and which will progress. In this paper, simulation of various scheduling algorithm First-Come-First-Served (FCFS), Round-Robin (RR), Shortest Process Next (SPN) and Shortest Remaining Time (SRT) is done over C#

***Keywords:** Simulation, Scheduling algorithm, Multiprogramming, I/O operation*

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. A computer is a set of resources for the movement, storage and processing of data and for the control of these functions. The operating system is responsible for managing these resources.

A key responsibility of the operating system is to manage the various resources available to it and to schedule their use by the various active process. Scheduling is a fundamental and most important OS function which is essential to an operating system's design. Multiprogramming, Multiuser, Multitasking, Multiprocessing and Multi threading are more attractive features for OS designer to improve the performance.

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.

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. [4] presented a Linear Data Model Based Study of Improved Round Robin CPU Scheduling algorithm with features of Shortest Job Firs scheduling with varying time quanum.Lulla et al. [6] 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 we 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. [9]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. [10] 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. [8] 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 [11] used simulator in operating system to learn CPU scheduling algorithms in an easier and a more effective way. Sindhu et al. [12] proposed an algorithm which can handle all types of process with optimum scheduling criteria.

Terry Regner & Craig Lacey [13] 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. [7] 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 [5] 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. BASIC CONCEPTS

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. Then objective of multiprogramming is to have some processes running at all time, to maximize CPU utilization. The idea is relatively simple. A processor is executed until it must wait typically for the completion of some I/O request. In a simple computer system, the CPU then just sits idle. All this waiting time is wasted no useful work is accomplished. With multi programming, we try use this time productively.

Several processes are kept in memory at one time. When one process has to wait, the operating system takes the CPU away from that process and gives the CPU to another process. Scheduling can be defined as a mechanism or a tool to control the execution of number of processes performed by a computer. The basic idea is to keep the CPU busy as much as possible by executing a process and then switch to another process. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating system design.

3.1 Types of CPU 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. The short term also known as the dispate. This section has discussed the scheduling oplicies in short term scheduling.

Figure 1 relates the scheduling functions to the process state transition diagram.

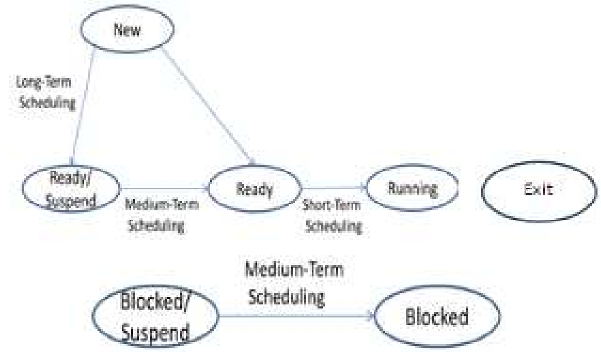

Figure 1.Scheduling and Process State Transitions

3.2 Dispatcher

The dispatchers is the module that gives control of the CPU to the process selected by the short term schduler; this involves.

- Switching context
- Switching to user mode
- Jumping to the proper location in the user progress to restart that program.

3.3 CPU Scheduler

Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out the short term scheduler or CPU scheduler. The CPU scheduler selects a process from the processes in memory that are ready to execute and allocates the CPU to that process. All the processes in the ready queue are lined up waiting for a chance to run on the CPU. The records in the queues are generally process control blocks (PCB) of the processes.

3.4 Characteristics of Various Scheduling Policies

- Selction Function: determines which process, among ready processes is selected next for execution.
- Decision mode: specifies the instants in time at which the selection function is exercised. There are two general categories: Nonpreemptive and preemptive.
- Throughput: the schduling policy should attempt to maximize the number of processes completed per unit of time.
- Response time: For a interactive processes, this is the time from the submission of a request until the response begins to be received.
- Starvation: A condition in which a process is in definitely delayed because other processes are always given performance.

3.5 Basic CPU Scheduling Algorithm

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.

3.5.1 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 [14].

3.5.2 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 techniques is also known as time slicing, because each process is given a slice of time before being preempted [14].

3.5.3 Shortest Process Next

Another approach to reducing the basis in favor of long processes inherent in FCFS is the shortes process next (SPN) policy. This is a non reemptive 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 [14].

3.5.4 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 [14].

4. IMPLEMENTATION

The simulator is designed using C#. The simulator analysis four CPU Scheduling policies. CPU scheduling policies is most important function and also critical part of an operating system. There are several policies of process allocation such as FCFS, RR, SRT and SPN. Simulator is designed to evaluate the process schedule strategies by considering user choose.User can can enter process time and arrival time of processor via textboxes of simulator.

Figure 2: Comparison Waiting Time and Turnaroud Time for Four scheduling policies in Simulation 1.

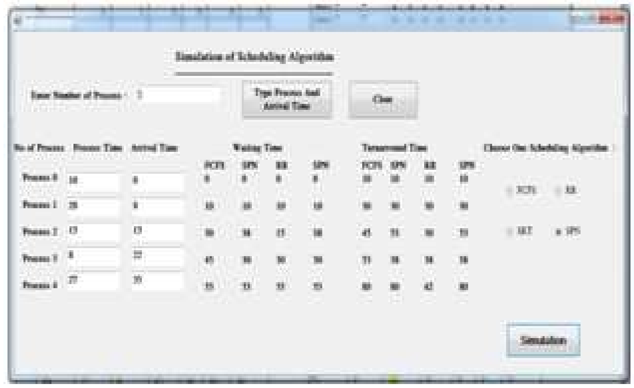

Figure 3: Comparison Waiting Time and Turnaroud Time for Four scheduling policies in Simulation 2.

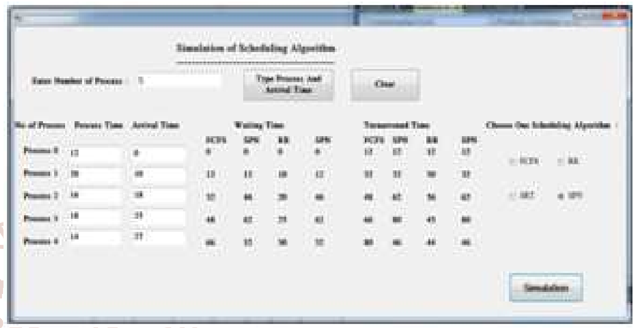

Figure 4: Comparison Waiting Time and Turnaroud Time for Four scheduling policies in Simulation 3.

Figure 5: Comparison Waiting Time and Turnaroud Time for Four scheduling policies in Simulation 4.

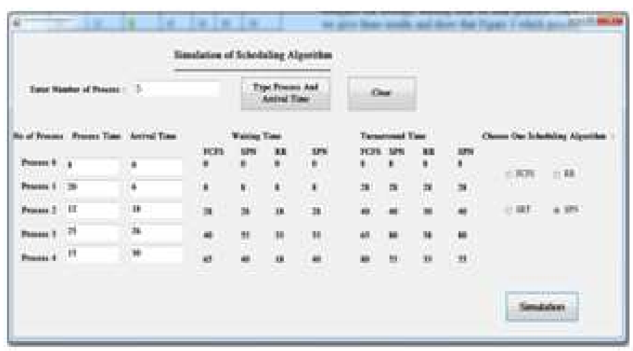

Figure 6: Comparison Waiting Time and Turnaroud Time for Four scheduling policies in Simulation 5.

Simulator can calculate waiting time and turnnaround time for four scheduling policies.

Turnaround time =service time +waiting time

Now, the average waiting time of four policies are compared. Then, these results are given and showed that Figure 7 which process gives the minimum average waiting time out of simulation 5 times.

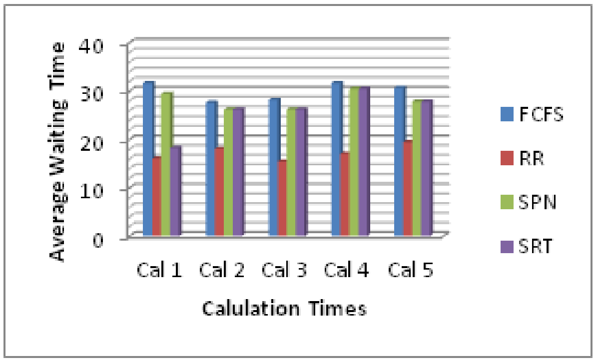

Figure7. Average Waiting Time Comparison between Four Scheduling Algorithms

Figure 8 shows that these four policies according to the average Turnaround time.

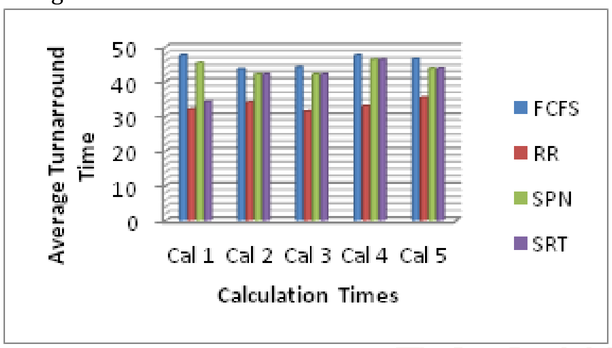

Figure8. Average Waiting Time Comparison between Four Scheduling Algorithms

5. CONCLUSION

The present paper shows analysison simulation of four CPU scheduling policies via FCFS, RR, SPN, SRT. The comparative analysis was made using simulation with C#. By this experimental setup, statistical analysis of the performance of all the four basic scheduling algorithms have able to do, 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.

6. 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] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

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