I.J. Mathematical Sciences and Computing, **2020, 2, 71-78**
Published Online April 2020 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijmsc.2020.02.04

Available online at http://www.mecs-press.net/ijmsc

Simulation of CPU Scheduling Algorithms using Poisson Distribution

Amit Mishra[a], Abdullahi Ofujeh Ahmed[b]

[a] *Baze University, Abuja, Nigeria*
[b] *Nigerian Nile Turkish University, Abuja, Nigeria*

Received: 02 November 2019; Accepted: 28 December 2019; Published: 08 April 2020

Abstract

Numerous scheduling algorithms have been developed and implemented in a bid to optimized CPU utilization. However, selecting a scheduling algorithm for real system is still very challenging as most of these algorithms have their peculiarities. In this paper, a comparative analysis of three CPU scheduling algorithms Shortest Job First Non-Preemptive, Dynamic Round-Robin even-odd number quantum Scheduling algorithm and Highest Response-Ratio-Next (HRRN) was carried out using dataset generated using Poisson Distribution. The performance of these algorithms was evaluated in the context of Average Waiting Time (AWT), Average Turnaround Time (ATT). Experimental results showed that Shortest Job First Non-Pre-emptive resulted in minimal AWT and ATT when compared with two other algorithms.

Index Terms: Poisson distribution, Shortest Job First Non-Preemptive, Dynamic Round-Robin even-odd number quantum Scheduling algorithm, Highest Response-Ratio-Next.

* Corresponding author.
E-mail address:

1. Introduction

One of the most expensive resources the operating system has to manage in a computer system is the Central Processing Unit (CPU). In a multi-programming and single processor environment, the single processor is shared among these processes following some CPU scheduling algorithm. CPU scheduling is the process where by one process is allowed to use the CPU while the execution of another process is on hold (waiting for an I/O or an interrupt) thereby making full utilization of the CPU [5]. The goal of CPU scheduling is to maximize CPU utilization by reducing process response time, waiting time, turnaround time and number of context switches. In a bid to find the best way of scheduling processes for execution, numerous CPU scheduling algorithms have been developed and implemented some of which are preemptive, while others are non-preemptive. In a preemptive scheduling, a process with a higher priority can block the currently running process, while in a non-preemptive scheduling algorithm, once the CPU is allocated to a process, the process must terminate before the next process will have chance to run.

Most of these scheduling algorithms have their pros and cons, consequently, choosing a scheduling algorithm for a real system becomes a problem. To overcome this problem, numerous algorithm evaluation techniques have been proposed. In view of the above, this study is employing the simulation technique of algorithm evaluation to investigate the performance of three scheduling algorithms namely Shortest Job First Non-Preemptive, Dynamic Round-Robin even-odd number quantum Scheduling algorithm and Highest Response-Ratio-Next (HRRN).

1.1 CPU Scheduling

In a multiprogramming environment, the CPU must be shared among the multiple processes following some scheduling algorithm. The task of determining when the processor should be assigned and to which process and for how long is called CPU scheduling. The goal of CPU scheduling is to maximize CPU utilization (keep the CPU busy all the time) so as to maximize system throughput. And by doing so, reducing process response time, waiting time, turnaround time and number of context switches.

1.2 CPU Scheduling Algorithms

Numerous Scheduling algorithm have been developed and implemented. Some of these include:

1.2.1 Shortest Job First Non-Preemptive Scheduling algorithm

Shortest job first (SJF) or shortest job next, is a planning arrangement that chooses the holding up procedure with the littlest execution time to execute next. SJN is a non-preemptive algorithm. Non-preemptive: We can't expel a procedure until it finishes it execution.

- Shortest Job first has the benefit of having least normal waiting time among all scheduling algorithms.
- It is a Greedy Algorithm.
- It may cause starvation if shorter procedures continue coming. This issue can be tackled utilizing the idea of aging.
- It is for all intents and purposes infeasible as Operating System may not realize burst time and therefore may not sort them. While it is beyond the realm of imagination to expect to foresee execution time, a few techniques can be utilized to appraise the execution time for a job, for example, a weighted normal of past execution times. SJF can be utilized in particular conditions where precise evaluations of running time are accessible.

1.2.2 Highest Response-Ratio-Next (HRRN) Scheduling algorithm

Round Robin scheduling is a preemptive type of FCFS. In RR scheduling, forms are likewise dispatched in FIFO succession however each procedure is permitted to run uniquely for a constrained measure of time.

In this Dynamic Round-Robin even-odd number quantum Scheduling algorithm, two-time quantum's TQ1 and TQ2 is determined. Where TQ1 is the burst time of the considerable number of procedures at even places in the prepared line and TQ2 is the normal burst time of the considerable number of processes at odd places in the prepared line [6]. At that point we analyze the two time quantum's, to get the more prominent Time Quantum. Take more prominent time as a period quantum and apply to each procedure.

```
If(TQ1 > TQ2 ){
TQ= TQ1                                   (1)
}else{
TQ= TQ2                                   (2)
}
```

Where tq1 is the average of even place process's burst time and tq2 is the average of odd place process's burst time.

1.2.3 Highest Response Ratio Next (HRNN) Scheduling Algorithm

Highest Response Ratio Next (HRNN) is one of the most optimal scheduling algorithms. This is a non-pre-emptive algorithm in which, the scheduling is done on the basis of an extra parameter called Response Ratio. A Response Ratio is calculated for each of the available jobs and the Job with the highest response ratio is given priority over the others.

Response Ratio is calculated by the given formula.

$$\textit{Response Ratio} = (W+S)/S \qquad (3)$$

Where,

W → Waiting Time

S → Service Time or Burst Time

1.2.4 Algorithm Evaluation Techniques

Different scheduling algorithms have been developed and implemented each of which has their peculiarities. Therefore, selecting a CPU scheduling algorithm for a given system becomes a problem. One way to select a scheduling algorithm for a particular system is by evaluating their performance on predefined tasks based on CPU scheduling criteria (metrics used to major CPU scheduling algorithm) using CPU algorithm evaluation techniques. Few of these algorithm evaluation techniques are discussed below:

- **Simulation**

Simulation is the way toward mimicking the task of a true procedure or framework. It includes the amalgamation of a counterfeit history of a framework and the perception of the fake history to make deduction about the working attributes of a genuine framework [4]. Although simulation gives accurate result, it usually involves programing a model of the computer system [5].

- **Implementation**

A more accurate but expensive technique of evaluating the performance of CPU scheduling algorithm is the implementation technique. Implementation involves implementing the algorithms in real system and observing their performance under real working conditions. This technique has overheads of coding and modifying the OS to support the new algorithms and sometimes users may not like working in an environment where the OS is constantly changing.

Other CPU algorithm evaluation techniques that can be found in the literature include; Queuing Model and Deterministic modelling.

2. Related Work

In recent time, many studies have been proposed to compare the performance CPU Scheduling algorithms. In 2011, Abdullahi, S. et al. carried out a similar study in which they used datasets generated using exponential distribution to compare the performance of three scheduling algorithms (SJF, RR and FCFS.). Their results show that RR resulted in minimal AWT and ATT [3].

In 2013, Goel, N., & Garg, R. B also carried out a similar studies in their paper “Comparative study of CPU scheduling algorithm” they compared four scheduling algorithm (FCFS, RR, Priority and SJF) and concluded that SJF resulted in minimal AWT and ATT among the others [4].

In 2014, Viral D Sanghvi1, Jignesh N Solanki. Focused on the RR scheduling techniques. There are two kinds of RR scheduling procedure, one is RR with static quantum and other is RR with dynamic quantum. In this paper we look at the aftereffect of various RR algorithm procedures those having dynamic quantum and we demonstrate that even-odd RR scheduling is the best scheduling procedures contrast with simple RR, average-max RR and average mid-max RR. [6].

3. Simulation of CPU Scheduling Algorithms Using Poisson Probability Distribution

Simulation is one of the best technique of evaluating the performance of a scheduling algorithms as it always gives more accurate results when compared to other techniques used in evaluating the performance scheduling algorithm [4]. In this paper, the simulation technique is employed in evaluating the performance of three scheduling algorithms (SJF, HRRN and RR) using datasets generated from binomial probability distribution function. The simulation model was implemented in Java programming language on a system using windows 10 with a processor of 2.70 Ghz. and 8 GB Ram.

Poisson, is a discrete probability distribution that states the probability of a given number of events happening in a fixed interval of time or space if these events happen with a known constant rate and freely of the time since the last event.

The Poisson distribution is well known for modelling the times an event happens in an interval of time or space.

An event can happen 0, 1, 2 … times in an interval. The normal number of events in an interval is assigned (lambda). Lambda is the occasion rate, likewise called the rate parameter. The likelihood of observing k events in an interval is given by the condition

$$P(k \text{ events in interval}) = e^{-\lambda}\frac{\lambda^k}{k!} \tag{4}$$

Where

- is the average number of events per interval
- e is the number 2.71828... (Euler's number) the base of the natural logarithms
- k takes values 0, 1, 2, …
- k! = k × (k − 1) × (k − 2) × … × 2 × 1 is the factorial of k.

3.1 Evaluation Criteria

To compare the performance of the three scheduling algorithms, we employed the following CPU scheduling criteria:

- Turnaround Time – the time interval from when a process was submitted to when it finish execution
- Waiting Time – the time a process spent waiting in the ready queue.
- Context Switch – the process of storing and restoring context (state) of a process, so that execution can be resumed at the same point at a later time.

4 Experiments and Results

In order to investigate the performance of the scheduling algorithms used in our simulation, an experiment was performed. The CPU burst time was synthesized using the Poisson distribution function using the following parameters, n = 19, p = 0.5 while the arrival time of the processes was generated using the following Poisson distribution parameters n= 3 and p =0.4.

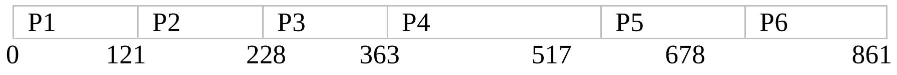
Table 1: Process Burst Time (Case I)

Process	Arrival Time	Burst Time
P1	0	121
P2	2	107
P3	10	135
P4	6	161
P5	4	154
P6	3	183

The simulation results of SJF, RR and HRRN are shown in table 2, 3, and 4 respectively. Similarly, fig 1, 2 and 3 show the Gantt chart representation of SJF, RR and HRRN respectively.

Table 2: Result of Shortest Job First Non-Preemptive Scheduling algorithm

Process	**Arrival Time**	**Burst Time**	**Waiting Time**	**Turnaround Time**
P1	0	121	O	121
P2	2	107	119	226
P3	10	135	218	353
P4	6	161	511	672
P5	4	154	359	513
P6	3	183	675	858
Average			**313.67**	**457.17**



CS = 0

Fig 1: Gantt chart for Shortest Job First Non-Preemptive Scheduling algorithm

Table 3: Result of Dynamic Round-Robin variable quantum Scheduling algorithm

Process	Arrival Time	Burst Time	Waiting Time	Turnaround Time
P1	0	121	O	121
P2	2	107	119	226
P3	10	135	218	353
P4	6	161	653	814
P5	4	154	668	822
P6	3	183	675	858
Average			**388.83**	**532.33**

P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P4**	**P5**	**P6**	**P6**	
0	121	228	363	511	659	807	820	826	857	861

CS = 4

Fig 2: Gantt chart for Dynamic Round-Robin even-odd number quantum

Table 4: Highest Response-Ratio-Next (HRRN) scheduling algorithm

Process	Arrival Time	Burst Time	Waiting Time	Turnaround Time
P1	0	121	O	121
P2	2	107	119	226
P3	10	135	518	690
P4	6	161	694	855
P5	4	154	407	561
P6	3	183	225	408
Average			**333.33**	**476.83**

P1	P2	P6	P5	P3	P1	
0	121	228	411	565	700	864

CS = 0

Fig 3: Gantt chart for HRRN

4.1 Result Analysis

In this section, the comparative analysis of our simulation results is presented. First we present the comparative results for both cases in tabular form as shown in table x and y and finally, we made use of bar charts to further compare the results obtained as can be seen in figure x , y and z for case I and figure x, y, and z respectively. The comparative results clearly show that SJFPS algorithm is given better results in terms of reducing AWT, ATT and number of context switches than FCFS and RR scheduling algorithms for both cases.

Table 5: comparison of Algorithms performance

Algorithms	Average Turnaround Time	Average Waiting Time
SJF	**457.17**	**313.67**
RR	**532.83**	**388.83**
HRRN	**476.83**	**333.33**

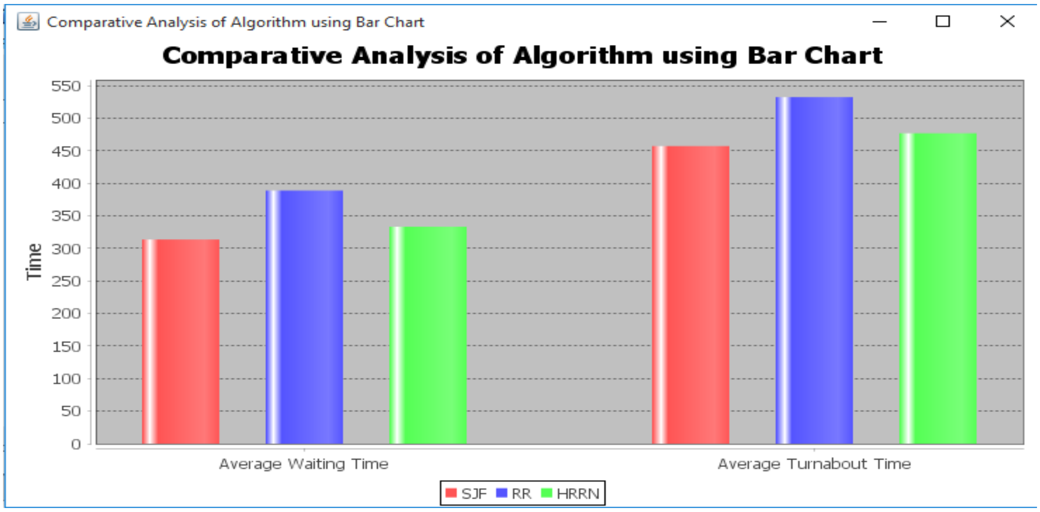
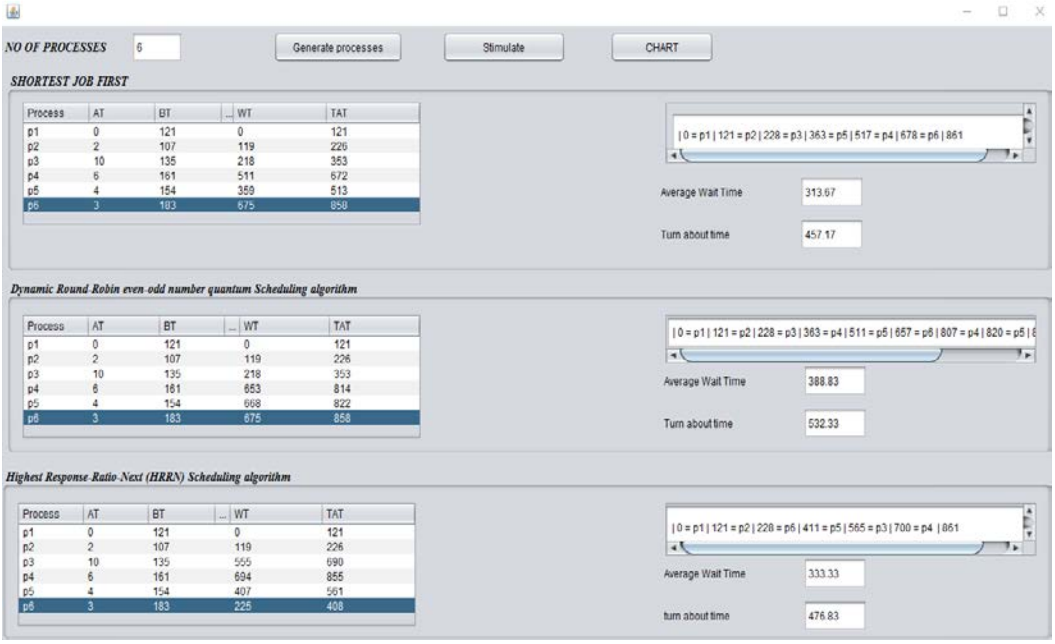


Fig 4: Bar chart representation of AWT, ATT



NO OF PROCESSES 6

SHORTEST JOB FIRST

Process	AT	BT	...	WT	TAT
p1	0	121		0	121
p2	2	107		119	226
p3	10	135		218	353
p4	6	161		511	672
p5	4	154		359	513
p6	3	183		675	858

| 0 = p1 | 121 = p2 | 228 = p3 | 363 = p5 | 517 = p4 | 678 = p6 | 861

Average Wait Time 313.67

Turn about time 457.17

Dynamic Round-Robin even-odd number quantum Scheduling algorithm

Process	AT	BT	...	WT	TAT
p1	0	121		0	121
p2	2	107		119	226
p3	10	135		218	353
p4	6	161		653	814
p5	4	154		668	822
p6	3	183		675	858

| 0 = p1 | 121 = p2 | 228 = p3 | 363 = p4 | 511 = p5 | 657 = p6 | 807 = p4 | 820 = p5 | 8

Average Wait Time 388.83

Turn about time 532.33

Highest Response-Ratio-Next (HRRN) Scheduling algorithm

Process	AT	BT	...	WT	TAT
p1	0	121		0	121
p2	2	107		119	226
p3	10	135		555	690
p4	6	161		694	855
p5	4	154		407	561
p6	3	183		225	408

| 0 = p1 | 121 = p2 | 228 = p6 | 411 = p5 | 565 = p3 | 700 = p4 | 861

Average Wait Time 333.33

turn about time 476.83

Fig 5: CPU Stimulation environment built with java

5 Conclusion

In this paper, the simulation of three CPU scheduling algorithms Shortest Job First Non-Preemptive, Highest Response-Ratio-Next (HRRN) and Dynamic Round-Robin even-odd number quantum. Using Poisson probability distribution function was implemented and their results were compared based on three scheduling criteria, Average waiting time, Average turnaround time and number of context switch. The experimental results demonstrated that Shortest Job First Non-Preemptive Scheduling algorithm perform better than Highest Response-Ratio-Next (HRRN) Scheduling algorithm and Dynamic Round-Robin even-odd number quantum Scheduling algorithm.

References

[1] Muraleedharan, A., Antony, N., & Nandakumar, R. (2016). Dynamic time slice round robin scheduling algorithm with unknown burst time. Indian Journal of Science and Technology, 9(8).

[2] Simon, A., Abdullahi, S., & Junaidu, S. (2014). Dynamic Round Robin with Controlled Preemption (DRRCP). International Journal of Computer Science Issues (IJCSI), 11(3), 109.

[3] Goel, N., & Garg, R. B. (2013). A comparative study of cpu scheduling algorithms. arXiv preprint arXiv:1307.4165.

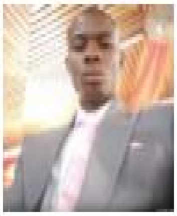
[4]Abur, M., Mohammed, A., Danjuma, S., & Abdullahi, S. (2011). A critical simulation of cpu scheduling algorithm using exponential distribution. IJCSI International Journal of Computer Science Issues, 8(6).

[5] Silberschatz A, Galvin P.B. and Gagne .G, 2009, Operating Systems Concepts, (7th ed), John Wiley & Sons, USA.

[6] Pallab Banerjee , Probal Banerjee , Shweta Sonali Dhal ,Comparative Performance Analysis of Even Odd Round Robin scheduling algorithm (EORR) using Dynamic Time Quantum with Round Robin scheduling algorithm using static Time Quantum , International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 9, September 2012

Authors' Profiles

Dr. **Amit Mishra** is an Assistant Professor in the Department of Computer Science, at Baze University, Abuja, Nigeria. His current research interests include: Software Engineering, Reverse Engineering, Software maintenance, web development, mobile app. He has been publishing in the above-mentioned fields in various national and international journals. He is also Microsoft Certified professional in Win2K and Exchange Server.



Abdullahi Ofujeh Ahmed is graduate student of computer science, Nile University Abuja, Nigeria. His current area of interest includes: web development, mobile app development, desktop app development and Software engineering. He also has key knowledge of machine learning and programming languages such as java, python, C, C++ and database management such as Oracle and Mysql.

How to cite this paper: Amit Mishra, Abdullahi Ofujeh Ahmed, "Simulation of CPU Scheduling Algorithms using Poisson Distribution", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.6, No.2, pp.71-78, 2020. DOI: 10.5815/ijmsc.2020.02.04