Journal of the Nigerian Association of Mathematical Physics
***Volume 60* (April – June *2021 Issue*), pp25 –30**
© J. of NAMP

COMPARATIVE ANALYSIS OF ROUND ROBIN, HIGHEST RESPONSE RATIO NEXT AND DYNAMIC QUANTUM BASED ROUND ROBIN SCHEDULING ALGORITHM

Suleiman T.[1], Iliyasu U.[2] and [3]Hassan K.A.

[1,2,3] Department of Computer Science and IT, Federal University Dutsinma

Abstract

This work intends to mitigate the difficulty in the choice of scheduling algorithm by operating system developers as the performance of system is primarily based on CPU schedulinsg algorithms. Operating system provides an environment where various processes are handled to maximize the CPU utilization. The ultimate objective of CPU scheduling is to make turnaround time and average waiting time minimal, in order to allow as many potential running processes as possible to make best use of CPU. The aim of this research work is to present appraisal on three (3) CPU scheduling algorithms: Round Robin (RR), Highest Response Ratio Next (HRRN) and Dynamic Quantum Based CPU Scheduling Algorithm. These algorithms were simulated using java programming to ascertain which amongst the algorithms utilize the CPU/ hardware resources in an optimal or efficient manner. This is achieved by using a performance metrics such as waiting and turnaround time of the processes to be executed. The experiment is conducted in a uniprocessor environment and accomplished by taking six (6) processes; i.e. Po, P1, P2…………P6. The data used in the simulation is generated using normal distribution and the processes arrival times using exponential function.
The simulator to run the experiment is designed using java programming and was run on windows 8 operating system.

***Keywords*:** CPU, Scheduling, OperatingSystem, Process, Algorithm.

INTRODUCTION

A process can be thought of as a program in execution. A process will need certain resources-such as CPU time, memory, files, and 1/0 devices to accomplish its task. These resources are allocated to the process either when it is created or while it is executing [1].The prime objective of OS is to provide an environment where various processes are handled to maximize the CPU utilization.
Scheduling is a core function of an operating system, as the main idea is to share computer resources among various processes. Almost each computer resource is scheduled before use,
Central Processing Unit (CPU) is one of the primary computer resources, so its scheduling is essential to an operating system's design [2].
Process scheduling is important because it plays an important role in effective resource utilization and the overall performance of the system [3].
The CPU scheduling algorithms focus on maximizing CPU utilization by minimizing waiting time, turnaround time and number of context switches for a set of processes [4].

Corresponding Author**:** Suleiman T., Email: tsuleiman@fudutsinma.edu.ng, Tel: +2348030672786

MATERIALS AND METHOD

This research work will use the following tools to be described below. The tools needed for this research work are hardware and software.

Hardware description

The system properties of the machine used to run the experiment is described as follows:

Processor-dual core @ 2.4GHZ (core i5)

RAM – 8GB

Hard drive – 320GB 5400 RPM hard drive

Software Description

Java virtual machine retrieved from url:https://en.wikipedia.org/wiki/Java_virtual_machine JVM: A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to java byte code. The JVM is detailed by a specification that formally describes what is required of a JVM implementation. Having a specification ensures interoperability of Java programs across different implementations so that program authors using the Java Development Kit (JDK) need not worry about idiosyncrasies of the underlying hardware platform.

SYSTEM DESIGN

The experiment is conducted in a uniprocessor environment and accomplished by taking six (6) processes; i.e. Po, P1, P2…………P6. The data to be used in the simulation is generated using normal distribution and the processes arrival times using exponential function.

The simulator to run the experiment is designed using java programming and was run on windows 8 operating system. The size of the machine's installed memory used is 4.0GB.

The time quantum in case of RR is determined statically by the user/designer before the processes are executed.

When the program was run the average waiting and turnaround time of the processes were automatically computed.

How RR, HRRN and DTQRR work

Table 1 below contains six processes with their associated burst times (BT) and arrival time (AT) measured in millisecond(ms), to illustrate how RR, HRRN and DTQRR work, prior to simulating all the algorithms in java as stated in the preceding pages.

Table 1: data generated

Process Name(P)	Arrival Time (AT)	CPU Burst Time (ms)
Po	0	4
P1	1	5
P2	2	2
P3	3	1
P4	4	6
P5	6	3

Table 2, 3 and 4 below have shown the Gantt charts of the three (3) CPU scheduling algorithm to be examined.

Table 2 Round Robin

P_0	P_1	P_2	Po	P_3	P_4	P_1	P_5	P_4	P_1	P_5	P_4	
0	2	4	6	8	9	11	13	15	17	18	19	21

Waiting time of the processes (WT):

Po = (0-0) + (6-2) =4, P1= (2-1) + (11-4) + (17-13) =12 P2= (4-2) =2

P3= (8-3) =5, P4= (9-4) + (15-7) + (19-17) =11 , P5= (13-6) + (18-15) =10
AWT= (4+12+2+5+11+10)/6=44/6=7.33ms
Turn Around Time of the Process (TAT):
Po=4+4=8, P1=12+5=17, P2=2+2=4, P3=5+1=6, P4=11+6=17, P5=10+3=13
ATAT= (8+17+4+6+17+13)/=65/6=10.83ms

Table 3: HRRN

Po	P2	P3	P1	P5	P4

0 4 6 7 12 15 21

RR1=(S+W)/S= [(4-1) +5]/5= (3+5)/5=1.6, RR2= [2+ (4-2)]/2= (2+2)/2= 2
RR3= (1+ (4-3))/1=2RR4= (6+4-4)/6=6/6=1 RR1 = (5+5)/5=2RR3= (1+3)/1=4
RR4= (6+2)/6=1.33 RR5 = (3+0)/3=1 RR1 = (5+7-1)/5=11/5=2.2RR4= (6+7-4)/6=9/6=1.5
RR5= (3+7-6)/3= (3+1)/3=1.33RR4= (6+12-4)/6=14/6=2.33 RR5= (3+12-6)/3=9/3=3
WT: Po=0, P1:7-1=6, P2:4-2=2 P3:6-3=3 P4:15-4=11 P5:12-6=6
AWT= (0+6+2+3+11+6)/6=28/6=4.67
TAT:
Po=0+4=4, P1=6++5=11, P2=2+2=4 P3=3+1=4 P4=11+6=17 P5=6+3=9
ATAT= (4+11+4+4+17+9)/6=49/6=8.17

Table 3: DTQRR

Po	P1	P1	P2	P3	P4	P4	P5

0 4 8 9 11 12 16 18 21

Wt: Po=0, P1=4-1=0, P2=9-2=7, P3=11-3=8, P4=12-4=8, P5=18-6=12
AWT= (0+3+7+8+8+12)/6=30/6=5.00ms
TAT: Po=0+4=4 P1=3+5=8, P2=7+2=9,, P3=8+1=9, P4=8+6=14, P5=12+3=15
ATAT=59/6=9.83ms

RESULTS AND ANALYSIS

Assumptions:

All experiments are assumed to be performed in uniprocessor environment and all the Processes are independent from each other. Attributes like burst time(BT), arrival time(AT) and time quantum are known prior to submission of process. All processes are CPU bound. No process is I/O bound.

This describes the implementation of the three CPU scheduling algorithm. The result of this implementation were analysed using SPSS.

Table 5: Generated Arrival time and CPU burst time

Process Name (P)	Arrival Time (AT)	CPU Burst Time (ms)
Po	0	4
P1	1	5
P2	2	2
P3	3	1
P4	4	6
P5	6	3

Table 5 contains the data generated for the experiment:.

As stated in the preceding page, the average waiting and turnaround time are the comparison metrics for the three scheduling algorithms. These are compared based on average waiting time (AWT) and average turnaround time.

Implementation

Figure 4.1, 4.2 and 4.3 below present the result of the simulation.

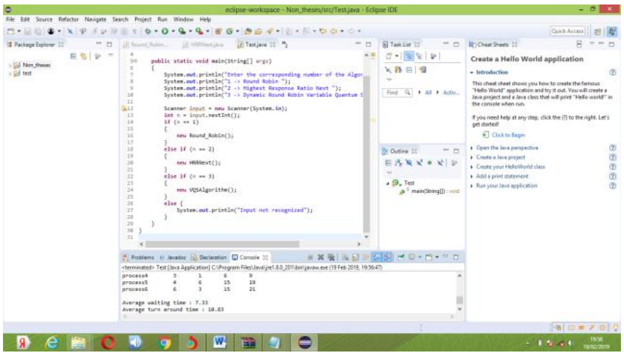

Figure 4-1: Round Robin (RR)

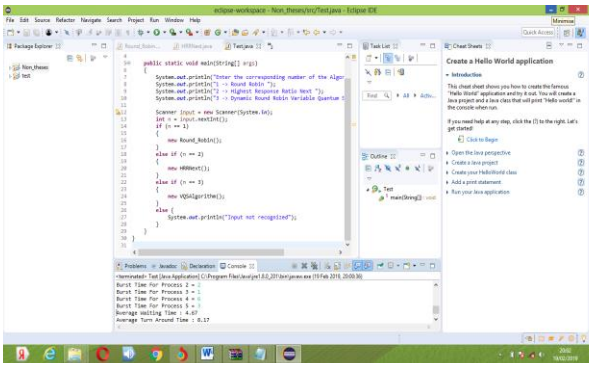

Figure 4-2Highest Response Ratio Next (HRRN)

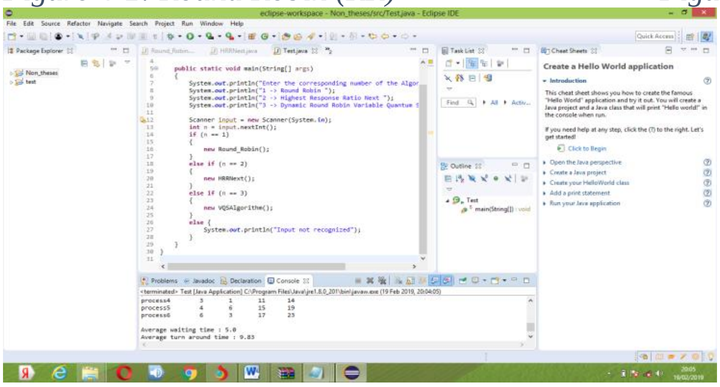

Figure 4-3Dynamic quantum based round robin

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	28.583ᵃ	2	14.292	.706	.497
Intercept	2211.125	1	2211.125	109.188	.000
PROCESSES	28.583	2	14.292	.706	.497
Error	1397.292	69	20.251		
Total	3637.000	72			
Corrected Total	1425.875	71			

Figure 4.4: R Squared = .020 (Adjusted R Squared = .008)

Figure 4.4 shows that, the adjusted coefficient of determination R-Square is 0.026 indicating that there is weak linear relationship between the processes and the Adjusted R Squared is indicating 0.8% of the variation is explained among the Processes. The P-Value (0.49) of the model is less than 0.05 we therefore conclude that there is no significant difference among the samples.

Sample	N	Means
Highest Response Ratio Next	24	4.7500
Dynamic Time Quantum Based Round Robin	24	5.5833
Round Robin	24	6.2917
Sig.		0.465

Figure 4.5: Means of the Scheduling Algorithms

Figure 4.5 compares the differences between the Processes, Highest Response Ratio Next have a mean of (4.7500) which is less than the means of the other processes, we therefore conclude that the Highest Response Ratio Next is more preferable than the others.

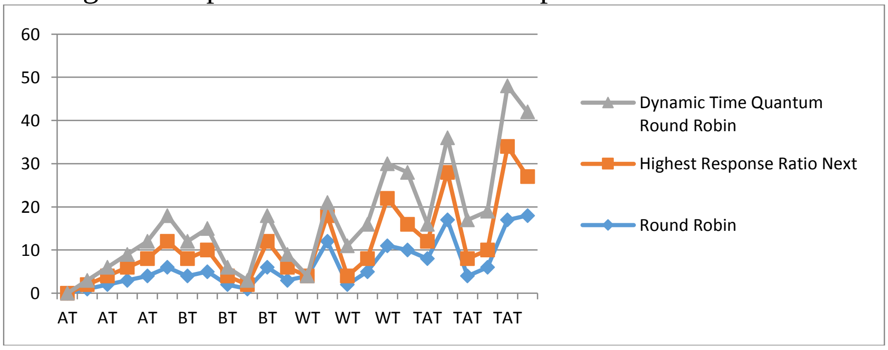

Figure 4-6graphical representations of the Processes against the Period

Figure 4.6 shows a graphical representation of the Processes against the Period.

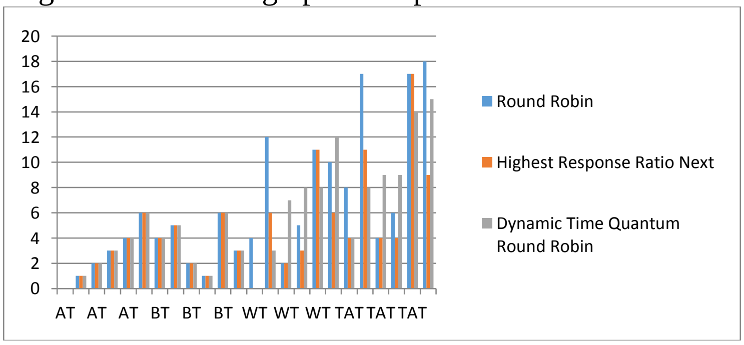

Figure 4-7Process against period

Figure 4.7 shows a Bar Chart representation of the Processes against their Period.

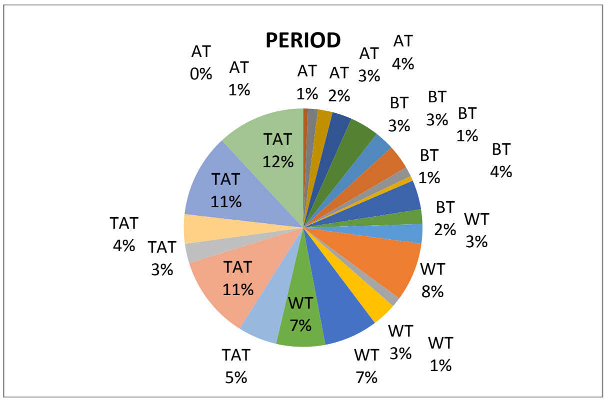

Figure 4-8 Pie Chart representation of the Period

Figure 4.8 shows a Pie Chart representation of the Period, where it illustrates that TAT is the most frequent period among the other periods because it has the highest percentage.

CONCLUSION

The simulation results have shown that the highest response ratio next scheduling algorithm (that is, HRRN was the best scheduling algorithm in terms of minimizing AWT and ATAT, and it was followed by DTQRR, and RR respectively.

RECOMMENDATION

Based on the result of the analysis been conducted this algorithm (that is Highest response ratio next) should be preferred over the other two algorithms, since it produces the minimal waiting and turnaround time.

References

[1] Berhanu, Y., Alemu, A., & Kumar, M. (2017). Dynamic Time Quantum based Round Robin CPU Scheduling Algorithm. *International Journal of Computer Applications*, *167*(13), 48–55. https://doi.org/10.5120/ijca2017914569.

[2] Patel, J., & Solanki, A. K. (2015). Performance Enhancement of CPU Scheduling by Hybrid Algorithms Using Genetic Approach, (May).

[3] Saeidi, S., & Baktash, H. A. (2012). Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method. *International Journal of Information Technology and Computer Science*, *4*(10), 67–73. https://doi.org/10.5815/ijitcs.2012.10.08

[4] Shidali, G. A., Junaidu, S. B., & Abdullahi, S. E. (2015). A New Hybrid Process Scheduling Algorithm (Pre-Emptive Modified Highest Response Ratio Next). *Computer Science and Engineering*, *5*(1), 1–7. https://doi.org/10.5923/j.computer.20150501.01

[5] Singh, A., Goyal, P., & Batra, S. (2010). An optimized round robin scheduling algorithm for CPU scheduling. *International Journal on Computer Science and Engineering*, *02*(07), 2383–2385.