International Journal of Engineering & Technology, 7 (2.32) (2018) 275-278

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Efficient PIMRR Algorithm Based on Scheduling Measures for Improving Real Time Systems

Siva Nageswara Rao[1], Ramkumar Jayaraman[1] , Dr. S.V.N Srinivasu[2]

[1]Department of Computer Science and Engineering, KLEF, Vijayawada, India
[2]Department of Computer Science and Engineering, PNC and KR PG College, Guntur (dt), India
**Corresponding author E-mail: sivanags@kluniversity.in*

Abstract

Scheduling play an important role to perform a single (or) multiple process activities by considering scheduling criteria's such as, waiting time, turnaround time, CPU utilization and context switches. The scheduling criteria's mainly depends on the quantum time which is specific to real time systems. The challenges faced by the real time systems based on scheduling activities viz., higher waiting time, more context switches and high turnaround time. All the scheduling criteria's are integrated to achieve Quality of Service (QoS) like throughput and delay. To improve the scheduling criteria's like waiting time, context switches and turnaround time, PIMRR algorithm is proposed. The PIMRR algorithm is first integrated with modulo operation to provide priority to all the process. The average of all the processes burst time is equal to the quantum time. Performance analysis is done for PMIRR with the existing simple round robin, PRR, Priority based RR scheduling based on the scheduling criteria's. Our results demonstrates that the PIMRR is more efficient compared to the existing ones, in terms of waiting time and turnaround time versus quantum time.

Keywords: *Scheduling, waiting time, context switches, turnaround time, priority and quantum time.*

1. Introduction

In recent days, operating system's performance is mainly depending upon the CPU scheduling algorithms as to perform multitasking operations. Scheduling plays an important role to assign multiple resources for completing several scheduling activities. We have several traditional scheduling algorithms to make the processes to perform multiple tasks simultaneously [1] [2] [3]. The process in Operating System (OS) is nothing but the instance of programs that is being executed as it contains codes and activity. The traditional algorithms are First Come First Server (FCFS), Shortest Job First (SJF), Round Robin (RR), etc.

FCFS is the simplest scheduling algorithm where the process are inserted into the queue and served based on the first come and first server [4]. As FCFS is the simple and easy to compute, average waiting time and average turnaround time is very large. To overcome the problem of waiting time and turnaround time, SJF scheduling is proposed as it executes the small burst time first for all the processes [4]. SJF helps to minimize the waiting time and turnaround time but the prior knowledge of the burst time of all the process is the major issue. To overcome the problem of prior knowledge of burst time, RR algorithm is proposed as it is same as FCFS algorithm along with pre-emption [5]. The pre-emption is the temporary interruption of task but the task can be resumed later. To implement the RR scheduling algorithm, queue is maintained based on FCFS process.

The basic RR gives increased waiting time, turnaround time and more number of context switches. RR is mostly used in various scheduling criteria's by facing several problem like increase waiting time and turnaround time and context switches which depends upon the fixed size of quantum time. Depending on the size of the time quantum, new RR scheduling algorithms are proposed [6] [7] helps to improve the scheduling process like waiting time and turnaround time. The drawbacks of the existing algorithm are increased waiting time, low throughput, more number of context switches and increased turnaround time.

The main objective of the proposed PIMRR algorithm helps to improve the scheduling criteria's like waiting time, turnaround time and context switches. PIMRR uses modulo to prioritize the burst time and average all the burst value to quantify the quantum time.

2. Related Works

Scheduling in real time systems has several criteria's such as, waiting time, turnaround time, CPU utilization and context switches. Based on the criteria's, scheduling algorithms are proposed mainly to maximize the throughput in terms of reducing waiting time, context switches and turnaround time [8]. Traditional scheduling algorithms such as FCFS, SJF and RR allow the multiple users to share the multiple resources in terms of QoS.

Abbas Noon et al. proposed AN algorithm based on dynamic quantum time which uses the burst time of the waiting time in the ready queue. AN algorithm solves the problem of fixed quantum time to increase the performance of the processes used in the system. Bashir Alam et al. developed a fuzzy logic based algorithm to decide value such that throughput not decreased without depends on the context switch [9]. Based on the fuzzy logic, fixed quantum time problem is solved and calculated. Here, scheduling criteria's are not considered and analysis based on the fuzzy based algorithm. Mohanty et al. proposed Priority Based Dynamic Round Robin (PBDRR) for improving the waiting time, turnaround time and context switches by taking dynamic quantum time [10].

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

PBDRR lacks in considering the input processes which can also reduce the scheduling criteria further. Scheduling algorithm developed must be applied to both soft and hard real time systems.

Output: Gantt chart

Number of processes and their burst time
If (Ready queue is not empty)
{
Processes 'P_n' = Ready Queue
B → Burst time

Calculate 'p'
p = Average CPU burst time 'B' of all the processes

// Modulo 'M' each processes
Calculate 'M' for each process
M = Burst Time 'B' % p

Sort the processes 'P' based on the value 'M' in ascending order
Assign Quantum time 'q' = 'p'
Create Gantt chart based on all the processes with 'q'
For (p=1; p<=n;P++)
{
For (r=1;r<=n;r++)
{
If (p[r] <= q)
{
// Update the process table;
else
P[r] = B[r] – q;
// stored in the Gantt chart
}
}
}}

Suman Bhatia et al. consider the problem of fixed time slice to reduce the response time, waiting time and throughput. A modified round robin algorithm is proposed which depends on the variation of time slice to improve the system performance. Neha Mittal et al. proposed a modified round robin algorithm based on quantum time to reduce the time and space complexity with respect to the scheduling criteria's [11]. The modified round robin considered several factors such as, providing time slice for all the process, CPU utilization, degree of multiprogramming and usage of resources. It lacks with the suitably of hard real time systems by varying the quantum time.

Behera et al. consider the multitasking operating system based on CPU scheduling for real time systems. Proposed two process CPU scheduling algorithm which depends on the scheduling criteria viz., waiting time, turnaround time and context switches. Process gets dispatched to another process based on the CPU requirement and increase order of burst time [12] [13] [14]. Two processes CPU scheduling algorithm reduce the average waiting time and average turnaround time but context switch scheduling criteria is not considered [15] [16].

Above existing survey lacks in providing optimality regarding the scheduling criteria's in which the system performance can be improved further [17] [18] [19]. Propose a scheduling algorithm which is suitable for both soft and hard real time systems [20] [21] [22].

3. Proposed Model

A. PIMRR Algorithm:

In PIMRR algorithm, we have taken 'n' processes 'P' = {P_1,P_2,P_3,P_4,P_5, . . .P_n}along with the burst time 'B' to create gantt chart to validate the 'n' process response. The main objective of the algorithm to calculate the improve form of scheduling criteria's viz., waiting time, turnaround time and context switch. When the quantum time value depends on the processes availed during the particular time, gradual improvement in the system performance. So, PIMRR algorithm uses average CPU burst time of all the processes is taken as the quantum time for all the processes. The priority based processes are taken into consideration to reduce waiting time of the overall process. Hence, our algorithm uses modulo based method against burst time to prioritize process by process for a one complete cycle. Then for the next cycle, the above process gets repeated till the end of all the processes in the system. At the end, gantt chart is created for all the processes present in the system.

Input: Processes 'P' = {P_1,P_2,P_3, P_4,P_5, . . .P_n} and their burst time 'B'

B. Workflow Process:

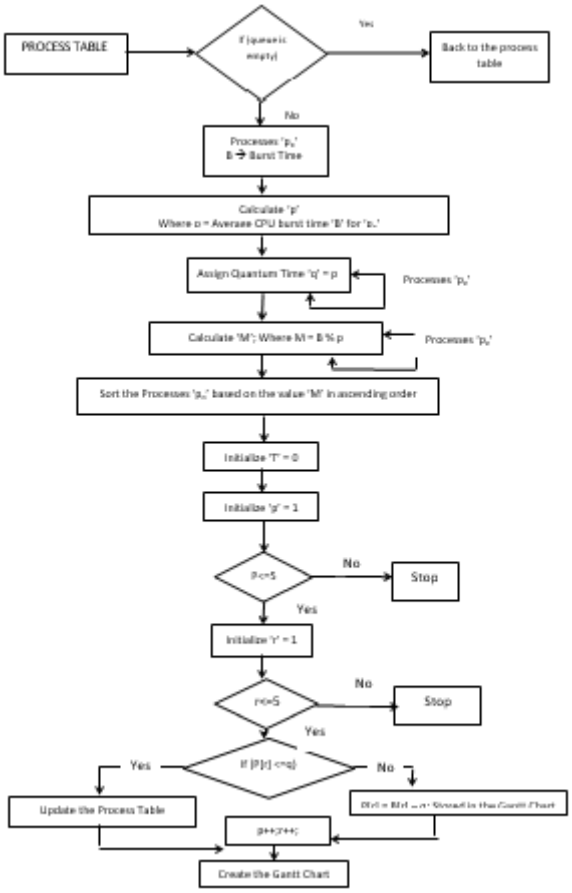


Fig.1 PIMRR algorithm

The workflow process of PIMRR algorithm is represented in Fig.1.

C. Mathematical Illustration:

For the mathematical illustration, we have taken '5' processes along with the burst time value.

Process	Burst Time
P1	22
P2	18
P3	09
P4	10
P5	04

Calculate 'M' = Average of the Burst Time

$$= \frac{22 + 18 + 9 + 10 + 4}{5} = \frac{63}{5}$$

$$= 12$$

Where, 'M' is equals to quantum time 'q' = 12

// To calculate and assign the priority value for each process by applying modulo

Process	Modulo Operation	Priority value
P1	22 mod 12	10
P2	18 mod 12	06
P3	9 mod 12	09
P4	10 mod 12	10
P5	4 mod 12	04

Based on the priority value assigned to each value, it is sorted in ascending order as below,

Process	Burst Time
P5	4
P2	18
P3	09
P1	10
P4	22

With Quantum time 'q' = 12, create Gantt Chart for all the process in the table below,

Process	Burst Time
P5	4
P2	12
P3	09
P1	12
P4	10

P5	P2	P3	P1	P4
0	4	16	37	47

Again apply quantum time 'q' = 12 for the remaining processes having burst time.

Process	Burst Time
P5	0
P2	06
P3	0
P1	10
P4	0

Again create the Gantt chart for the remaining processes

P5	P2	P3	P1	P4	P2	P1

0 4 16 25 37 47 53 63

Then calculate Context switch, Average waiting time and Average Turn Around time.

Number of Context Switches = 6

Average Waiting time = P1 → (0 – 0)

P2 → (4 – 1) + (47 – 4) = 46

P3 → (16 - 2) = 14

P4 → (25 – 3) + (63– 25)=38

P5 → (37 – 4) = 33

$$= \frac{46 + 14 + 60 + 33}{5}$$

= 30.6ms

Average turnaround Time = $4 + 53 + 25 + 63 + 47 / 5$

$= 192/5$

= 38.4ms

4. Performance Analysis

Based on the analysis, we have compared three algorithms viz., simple RR, PRR, Priority based RR CPU scheduling with our proposed PIMRR algorithm based on the scheduling criteria's such as, Quantum time, waiting time, turnaround time and context switch. Results show that the PIMRR algorithm is more efficient in all the scheduling criteria aspects considered above than the existing ones as represented in Table. 1.

Table. 1. Analysis based on various scheduling criteria's

Algorithm	Time Quantum	Average Waiting Time	Average Turn-around Time	Number of Context Switches
Simple Round Robin [9]	26	69.75ms	98.75 ms	07
Proposed RR [13]	29	69ms	98ms	05
Priority based Round Robin CPU scheduling [14]	12	32.5ms	44.5ms	13
PIMRR Algorithm	12	30.6ms	38.4ms	06

5. Conclusion

Based on the scheduling system aspects, PIMRR algorithm is proposed to improve the scheduling criteria viz., waiting time, turnaround time and context switches. Here, we have applied mod function to the burst time value to assign priority to the processes which are taken into consideration. The mod function is applied to reduce the average waiting time of all the processes in system. Quantum value is equal to the average of all processes burst time value. Based on the assigned quantum value, number of context switches gets decreased. Overall performance of PIMRR algorithm is analysed against existing simple RR, PRR and Priority based RR CPU scheduling algorithm with respect to several scheduling criteria like waiting time, turnaround time and context switch.

References

[1]. Sonia Zouaoui, LotfiBoussaid and AbdellatifMtibaa, "CPU scheduling algorithms: Case & comparative study," in Proc. of 17th International Conference onSciences and Techniques of Automatic Control and Computer Engineering (STA), 2016.

[2]. Bin Nie, Jianqiang Du, Guoliang Xu, Hongning Liu, Riyue Yu and Quan Wen, "A New Operating System Scheduling Algorithm," Advanced Research on Electronic Commerce, Web Application, and Communication, pp. 92-96, 2011.

[3]. Meierhofer J., Bernhard U.P., Balmelli P., and Bernasconi D., "Priority scheduling algorithm for ATM wireless network access," IEEE 6th International Conference onUniversal Personal Communications Record, Conference Record., 1997.

[4]. W. Zhao and J.A. Stankovic, "Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems," in Proc. of Real Time Systems Symposium, 1989, Doi: 10.1109/REAL.1989.63566.

[5]. Mokhtar A. Alworafi, AtyafDhari, Asma A. Al-Hashmi, A. BasitDarem and Suresha, "An improved SJF scheduling algorithm in cloud computing environment," In proc. of International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), 2016, Doi: 10.1109/ICEECCOT.2016.7955216.

[6]. HarshalBharatkumar Parekh and SheetalChaudhari, "Improved Round Robin CPU scheduling algorithm: Round Robin, Shortest Job First and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time," in Proc. International Conference on of Global Trends in Signal Processing, Information

Computing and Communication (ICGTSPICC), 2016,Doi: 10.1109/ICGTSPICC.2016.7955294.
[7]. Ahmed Alsheikhy, Reda Ammar and RaafatElfouly, "An improved dynamic Round Robin scheduling algorithm based on a variant quantum time," In proc. of 11th International Computer Engineering Conference (ICENCO), 2015, DOI: 10.1109/ICENCO.2015.7416332.
[8]. Bashir Alam, M. N. Doja and R. Biswas, "Finding Time Quantum of Round Robin CPU Scheduling Algorithm Using Fuzzy Logic," In proc. of International Conference on Computer and Electrical Engineering, 2008, Doi: 10.1109/ICCEE.2008.89.
[9]. Abbas Noon, Ali Kalakech and SeifedineKadry, "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue. 3, pp. 224-229, 2011.
[10]. Rakesh Mohanty, H. S. Behera, KhusbuPatwari, Monisha Dash and M. Lakshmi Prasanna, "Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time Systems," International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 2, No. 2, pp. 46-50, 2011.
[11]. Suman Bhatia and SupriyaRaheja, "Modified Round Robin Scheduling Algorithm Using Variable Time Slice," International Journal of Engineering Research and Development," Vol. 8, Issue. 3, pp. 57-61, 2013.
[12]. H. S. Behera, Rakesh Mohanty, SabyasachiSahu and Sourav Kumar Bhoi, "Comparative Performance Analysis Of Multi-Dynamic Time Quantum Round Robin (Mdtqrr) Algorithm WithArrival Time," Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2, No. 2, pp. 262-271, 2011.
[13]. Anukush Joshi and ShubhashishGoswami," Modified Round Robin Algorithm by using priority Scheduling," Advances in Computational Sciences and technology, Vol. 10, No. 6, pp. 1543-1549, 2017.
[14]. Ishwari Singh Rajput and Deepa Gupta, "A Priority based Round Robin CPUScheduling Algorithm for Real Time Systems," International Journal of Innovations in Engineering and Technology (IJIET), Vol. 1, Issue. 3, 2012.
[15]. Siva Nageswara Rao, G., Srinivasu, S.V.N., Srinivasu, N., Ramakoteswara Rao, G.Enhanced precedence scheduling algorithm with dynamic time quantum EPSADTQ)(2015) Research Journal of Applied Sciences, Engineering and Technology, 10 (8), pp. 938-941.
[16]. Siva Nageswara Rao, G., Srinivasu, N., Sagar, K.V.D., Sai Madhuri, P.Comparison of round robin CPU scheduling algorithm with various dynamic time quantum (2014) International Journal of Applied Engineering Research, 9 (18), pp. 4905-4916.
[17]. Siva Nageswara Rao, G., Srinivasu, S.V.N., Srinivasu, N., Naga Raju, O.A new proposed Dynamic dual processor based CPU scheduling algorithm(2015) Journal of Theoretical and Applied Information Technology, 73 (2), pp. 226-231.
[18]. Rao, G.S.N., Srinivasu, N., Girish Kumar, K., Abhishek, B.An enhanced dynamic Round Robin CPU scheduling algorithm(2014) International Journal of Applied Engineering Research, 9 (15), pp. 3085-3098.
[19]. Rao, G.S., Krishna, C.V.P., Rao, K.R.Extreme Programming for service-based application development architecture(2014) Proceedings of the 2014 Conference on IT in Business, Industry and Government: An International Conference by CSI on Big Data, CSIBIG 2014, art. no. 7056955,.
[20]. Rao, G.S., Krishna, C.V.P., Rao, K.R.Multi Objective Particle Swarm Optimization for Software Cost Estimation(2014) Advances in Intelligent Systems and Computing, 248 VOLUME I, pp. 125-132.
[21]. Rao, G.S., Krishna, C.V.P., Rao, K.R.Rational unified process for service oriented application in extreme programming (2013) 2013 4th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2013, art. no. 6726586, .
[22]. Daya Sagar, K.V., Sivanageswara Rao, G., Srikanth, T., Raghavendra, K.A relational analytic platform with Hadoop using the On Demand Integration (ODI) capability(2014) International Journal of Applied Engineering Research, 9 (13), pp. 2095-2102.