Improved Dynamic Time Slice Priority Scheduling Algorithm with Unknown Burst Time

A. Sankar[1], G. Ananthnath[2].
[1]*Student, Department of Computer Applications, KMM Institute of Postgraduate Studies, Tirupathi*
[2]*Assistant Professor, Department of Computer Applications, KMM Institute of Postgraduate Studies, Tirupathi*

***Abstract*- Operating System (OS) is the brain of computer system which can manages the resources available the system. OS controls the execution of many other application programs and acts as an interface between computer hardware and software. It has some attractive features like multiprogramming, multitasking and allows multi-users. Priority scheduling is one of the algorithms mostly used in time sharing system and in network scheduling. Time slices are assigned to each process in equal portions and it handling all processes in circular order without priority. Priority scheduling algorithm requires some parameter such as arrival time, burst time and quantum time which enables the scheduler to predict the behavior of possible processes. Prior to the execution of a process, the burst time is not known. This paper proposed a more improvement in the Priority scheduling algorithm improving the of Anju et.al algorithm. Where burst time is determined using an initial time quantum. However, the improved algorithm determines burst time using instruction counting each of the process and by experimental analysis. This proposed algorithm performs better than Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time in terms of minimizing average waiting time, average turnaround time and number of context switches.**

***Index Terms*- CPU scheduling algorithm, Average Waiting time, Average Turnaround Time, Number of Context Switches.**

I. INTRODUCTION

Operating System (OS) is the brain of computer system which can manages the resources available the system. OS controls the execution of many other application programs and acts as an interface between computer hardware and software. It has some attractive features like multiprogramming, multitasking and allows multi-users. The most important tasks of OS need to perform the job scheduling where many processing requests coming from multiple channels to a ready queue and system manage all requests to achieve high efficiency.

Time Slice:
Each procedure is permitted to keep running for just a constrained measure of time. This time interim is known as a period cut or quantum. Or then again the timeframe for which a procedure is permitted to run un-interrupted in a pre-emptive multitasking working framework.

Burst Time:
While booking, each procedure gets the opportunity to utilize the CPU for its slice. The cut that it gets is known as the CPU burst. In basic terms, the length for which a procedure gains power of the CPU is the CPU burst time, and the idea of picking up control of the CPU is the CPU burst.

II. CPU SCHEDULING ALGORITHM

CPU scheduling is the basis of multiprogramming operating systems, by switching the CPU among processes. The operating system can make the computer more effective, whenever the CPU becomes idle; the operating system must select one of the processes in the ready queue for execution. There are different CPU scheduling algorithms having different properties, and the choice particular algorithm is based on favor of one class of processes over another. If we considered one section of algorithm for a particular situation, properties of different algorithms must be considered. A good scheduling algorithm consisting of following characteristics such as context switch, throughput, CPU utilization, turnaround time, waiting time and response time.

CPU scheduling algorithm:- First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling (PS) and Round Robin (RR) . FCFS is the simplest form of CPU scheduling algorithm. Here we allocate CPU to the processes based on their arrival time to the ready queue. Based on that arriving time we allocate the CPU to the process. After completion of that process, the process must be removed from the ready queue. SJF, here the time scheduler is based shortest burst time in the ready queue, so that the process with least burst time is executed first. If two processes have equal burst time then FCFS is applied. The processes which have long burst time it takes more time to execute that's why the CPU gives first preference to shortest burst time.

PS is associated with the priority number. The CPU is allocated to the processes based on the priority. The process which has highest priority number then the CPU will give first preference to that process. If there are multiple processes with same priority, then FCFS will be applied. Lower priority processes can wait in the queue, because the CPU may have a steady supply of higher priority process. Round Robin (RR) scheduling algorithm is designed especially for time-sharing systems. It is a similar version of first-come, first-served scheduling. Processes are stayed in a first-in-first-out sequence but each process is allowed to run for only a limited amount of time. This time interval is known as a time-slice or quantum. It is similar to FIFO scheduling but preemption added to switches between processes. In existing system. Prior to the execution of a process, the burst time is not known. This paper proposed a priority scheduling algorithm improving the of Anju et.al algorithm. Where burst time is determined using an initial time quantum. However, the improved algorithm determines burst time using instruction counting each of the process and by experimental analysis. This proposed algorithm performs better than Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time in terms of minimizing average waiting time, average turnaround time and number of context switches. Disadvantages of RR algorithm is larger waiting time and Response time, Context Switches and Low throughput.

III. PROPOSED ALGORITHM

Need planning is a non-preemptive calculation and a standout amongst the most well-known booking calculations in cluster frameworks. Each procedure is allotted a need. Process with the most noteworthy need is to be executed first et cetera. Procedures with a similar need are executed on first start things out served premise. Need can be chosen in view of memory prerequisites, time necessities or some other asset requirement. Implementation: 1-First information the procedures with their burst time and priority.2-Sort the procedures, burst time and need as indicated by the priority.3-Now essentially apply FCFS algorithm.

Proposed Improved Dynamic Time Slice need CPU planning calculation with Unknown Burst Time

Stage 1: Allocate CPU to each procedure in as per the given need, for given time quantum (say k units) just for one time.

Stage 2: After finishing of initial step following advances are performed:

a) Processors are orchestrated in expanding request or their outstanding CPU burst time in the prepared line. New needs are allotted as indicated by the rest of the CPU blasts of procedures; the procedure with most brief residual CPU burst is doled out with most elevated need.

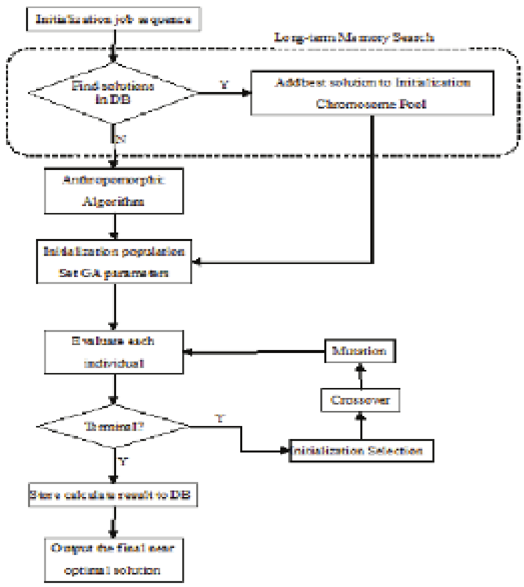
b) The procedures are executed by the new needs in view of the rest of the CPU blasts, and each procedure gains the power of the CPU until the point when they completed their execution.

Flow Chart

Flow diagram of the proposed need planning calculation. The proposed calculation will be executed in two stages which will limits various execution parameters, for example, setting

Switches, normal holding up time and normal turnaround time flow Chart

Flow chart of the proposed priority scheduling algorithm. The proposed algorithm will be executed in two steps which will helps to minimize a number of performance parameters such as context switches, average waiting time and average turnaround time.



IV. CONCLUSION

The proposed algorithm was used to enhance dynamic time slice priority scheduling with unknown burst time scheduling algorithm. The algorithm determines burst time using instruction count in each of the process by experimental analysis. The simulation results showed that this approach minimized average waiting Time, average turnaround time and number of context switches. Future research should focus on investigating other method of determining burst time of processes to improve efficiency of scheduling techniques.

REFERENCES

[1] NeetuGoel, Dr. R. B. Garg, "A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics & Image Processing, Vol 2, issue 4, November 2012.

[2] Sukumar Babu Bandarupalli, Neelima Priyanka Nutulapati, Prof. Dr. P.Suresh Varma, "A Novel CPU Scheduling Algorithm–Preemptive & Non-Preemptive", International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.6, pp-4484-4490 ISSN: 2249-6645, Nov-Dec. 2012.

[3] Pushpraj Singh, Vinod Singh, Anjani Pandey, "Analysis and Comparison of CPU Scheduling Algorithms", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 1, ISSN 2250- 2459, ISO 9001:2008, January 2014. 640 650 660 670 680 FCFS based Priority Scheduling Algorithm SJF based Priority Scheduling Algorithm Average Turnaround Time Average Turnaround Time Rukhsar Khan et al, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.9, September- 2015, pg. 324-331 © 2015, IJCSMC All Rights Reserved 331

[4] E.O. Oyetunji and A. E. Oluleye, "Performance Assessment of Some CPU Scheduling Algorithms", Res. J. Inform. Technol., 1(1): 22-26, 2009, ISSN: 2041-3114, Aug 2009.

[5] Jyotirmay Patel and A.K.Solanki, "CPU Scheduling: A Comparative Study", Proceedings of the 5th atonal Conference, INDIACom, ISSN 0973-7529, ISBN 978-93-80544-00-7, 2011.

[6] Taqwa Flayyih Hasan, "CPU scheduling Visualization", Diyala Journal of Engineering Sciences, Vol. 07, No. 01, pp. 16-29, ISSN 1999-8716, March 2014

[7] Mehdi Neshat, Mehdi Sargolzaei, Adel Najaran, Ali Adeli, "The New Method Of Adaptive CPU Scheduling Using Fonseca And Fleming's Genetic Algorithm", Journal of Theoretical and Applied Information Technology, Vol. 37 No.1, ISSN: 1992-8645, 15th March 2012.

[8] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", 7th Edition, John Wiley & Sons, ISBN: 978-81-265-2051-0, 2005.

[9] William Stallings, "Operating System Internals And Design Principles", 7th Edition, Prentice Hall, ISBN 10: 0-13-230998-X, ISBN 13: 978-0-13-230998-1, 2009.

[10] Ashok Kumar, Dr. Harish Rohil, Suraj Arya, "Analysis of CPU Scheduling Policies through Simulation", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, ISSN: 2277 128X, May 2013.

[11] Siddharth Tyagi, Sudheer Choudhary, Akshant Poonia," Enhanced Priority Scheduling Algorithm to minimize Process Starvation", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 10, October 2012.