

REN an Efficient Algorithm for Process Scheduling to Optimize CPU Blocking with the Time Slice

Renu Tiwari*
Student (M.Tech)
Department of Information Communication
Suresh Gyan Vihar University
Jaipur, Rajasthan, India

Dr. Naveen Hemrajani
Professor
Department of CS/IT
Suresh Gyan Vihar University
Jaipur, Rajasthan, India

Abstract— Scheduling is the main concept used efficiently in Operating System. It helps in selecting the processes for execution. Round Robin (RR) is one of the most widely used CPU scheduling algorithm. This algorithm is designed especially for time sharing systems. It is similar to FCFS scheduling algorithm, but preemption is added to switch between processes. But, its performance decreases with respect to context switching, which is a challenge and it happens during each scheduling. Overall performance of the system depends on choice of an optimal time quantum, so that context switching can be reduced. [1]

In this paper we propose an efficient scheduling algorithm by using operating system simulation which is based on time quantum between the communication of two processes at the same time and to reduce the number of blocking by the scheduler.

The operating system simulation produces simulated result based on real scheduling perform by an operating system scheduler in the form of graphical results, as we know any simulator is the virtual implementation of the real time environment. This algorithm for scheduling provides an efficient approach of scheduling the incoming processes.

Keywords- Software Simulation; Time-Quantum; RR; FCFS

I. INTRODUCTION

Scheduling is the process in which all coming processes are scheduled by scheduler, first of all we discuss about all scheduling algorithms. In the First-Come-First-Serve (FCFS) algorithm, process which comes or arrive first is immediately allocated to the CPU based on FIFO criterion. In Shortest Job First (SJF) algorithm, process having shortest CPU burst time will execute first. If two processes having same burst time and arrive simultaneously, then FCFS procedure is applied.[3] Priority scheduling algorithm, provides priority (internally or externally) to each process and selects the highest priority process from the ready queue. In case of Round Robin (RR) algorithm, time interval of one time quantum is given to each

process present in the circular queue emphasizing on the fairness factor.[1]

CPU Scheduling consist of the following steps:

- 1- Approx all processes deals with CPU scheduling number waiting for I/O task. (taking from memory takes more time as compared to CPU speed)

- 2- In a general system running a single process, the time taken for waiting I/O is wasted, and those CPU cycles are lost forever.
- 3- A scheduling system allows one process to use the CPU while another is waiting for I/O, thereby making full use of otherwise lost CPU cycles.
- 4- The challenge is to make the overall system as "efficient" and "fair" as possible, subject to varying and often dynamic conditions, and where "efficient" and "fair" are somewhat subjective terms, often subject to shifting priority policies CPU-I/O Burst Cycle Almost all processes alternate between two states in a continuing cycle[2].

The purpose of this scheduling software simulation is that it describes the graphically display of process scheduling and takes the input files to perform operation on it and produces output file as result. This software simulation provides a graphical user interface and step by step procedure of program execution which display the number of blocks of a process occurred by other process by giving suitable quantum of time to each process.

In this paper we will describe Introduction in Section 1; Section 2 describes Scheduling used for simulation; Section 3 tells about the modified and efficient REN algorithm for scheduling of the CPU processes and Section 4 is for result and analysis

II. SCHEDULING USED FOR SIMULATION

Round-robin scheduling each task have equal CPU execution time. Real round-robin scheduling doesn't perform real-time tasks because in real-time systems, task performs in distinct measures of scheduling.[4] But, preemptive scheduling and priority scheduling integrated with round-robin scheduling by the use of time-quantum or time-slice to give equally distributed CPU time as shown in Figure.

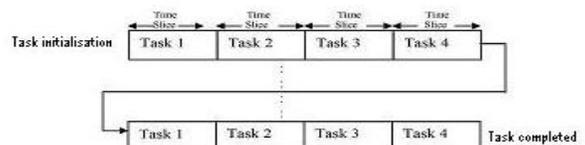


Fig: Round-Robin Scheduling [4]

Basically this type of scheduling consists of the two initial steps:

1) Queue is created for the tasks 2) Fetching the most require task to run for one quantum. Then another task is fetched by the scheduler to run. [6]

According to the hardware time quantum is denoted by time units of constant size. According to this criterion scheduler can achieve sharing by the two ways: by adjusting the frequency chose by the user or by adjusting the length of the time quantum so that user is allowed to run longer for desired allocations. According to the fair-share scheduling perfect fairness is a state in which the amount of tasks that each user receives is precisely proportional to its share and the resource can be used at the same time by all users.[7] But, when scheduling is fixed for a time-dependent resource and discrete time quantum, then the fairness of the scheduling is affected and is not ideally fair in all time intervals to all users. Even though scheduling algorithms that can achieve the ideal fairness do not exist, some specifications come close to the result of ideal allocation.[8] Its allocation scheme is compared with the allocation scheme of the perfectly fair-share scheduling.

III. THE MODIFIED AND EFFICIENT REN ALGORITHM FOR SCHEDULING OF THE CPU PROCESSES

Step 1: Create a configuration file Fc.

Step 2: Number of parameters used to initialize a file Fc:

- (a) PNUM: Number of Processes which are requires scheduling
- (b) MDEV: The average time period in milliseconds required by a process before completion.
- (c) SDEV: Take standard deviations by default zero from the average time period in milliseconds required by a process before completion.
- (d) PTQ: Specify number of processes with time quantum or I/O blocking.
- (e) FSC: Give fair-share CPU run time to all processes. (FSC= P0+P1+P2+.....+Pn) where FSC is Fair-share CPU run time, P0.....Pn is run time for each process respectively.

Step 3: Simulation performed by the following command
\$Java Scheduling File_Name

- (a) File_Name specifies the name of the file created in step 1
- (b) Where Scheduling is a prefix keyword which helps to run the scheduling simulation.

Step 4: The Scheduling simulation is depend on following

- (a) Process number
- (b) Mean Deviation
- (c) Standard Deviation=0
- (d) Time quantum
- (e) Run Time of All processes

Step 5: Completion of Simulation produces the two result files as

- 1) SUM_RSUL: The Summary Result file specifies the Process Number, CPU Time, I/O Blocking Time, CPU Completion Time and Number of CPU Blocked
- 2) SUM_PROC : The Summary Process file Specifies a table of the events occur during simulation process. Each line in the table is of the following form:

Process: PNUM PStat... (FSC PTQ AC_TIME FINAL_AC_TIME)

Where Process is a keyword, PNUM specifies the Process Number, PSTAT defines the three states of Process (Registered, I/O Blocked and Completed). FSC time is equally distributed among all processes, PTQ is the process time quantum, AC_Time is initial accumulated time and FINAL_AC_TIME is the final accumulated time of a process.[9].

IV. RESULT AND ANALYSIS

the overall simulation process we take two cases with the different configuration files with different parameters and compare the results of both files on the basis of number of CPU blocks occurred in the scheduling for the processes. Each case consist three steps for the scheduling process.

CASE 1: Taken according to the time quantum given by the users:

Step 1: Creating the configuration file Fc.

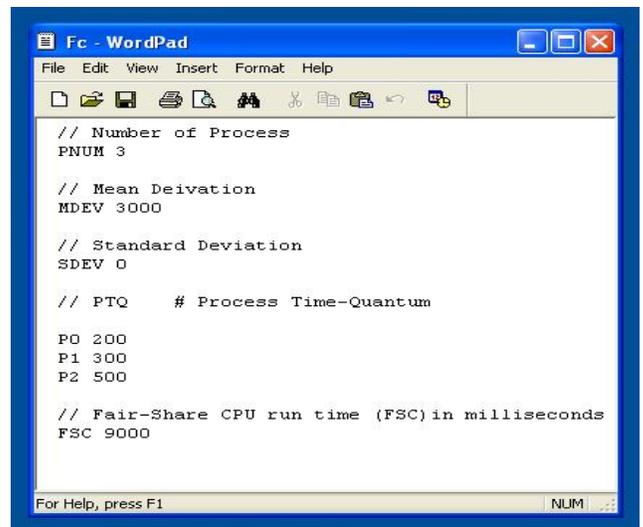


Figure 1: Fc File

In this configuration file Fc we declare some parameters with the constraints. As shown in fig.1 the first one is number of processes (PNUM) which is 3 (P0, P1 and P2 with PTQ 200, 300 and 500 respectively), Mean Deviation (MDEV) with value of 3000 milliseconds, Standard Deviation (SDEV) by default zero and Fair-Share CPU run time (FSC) is 9000 milliseconds.

Step 2: Initialization of Simulation Process:

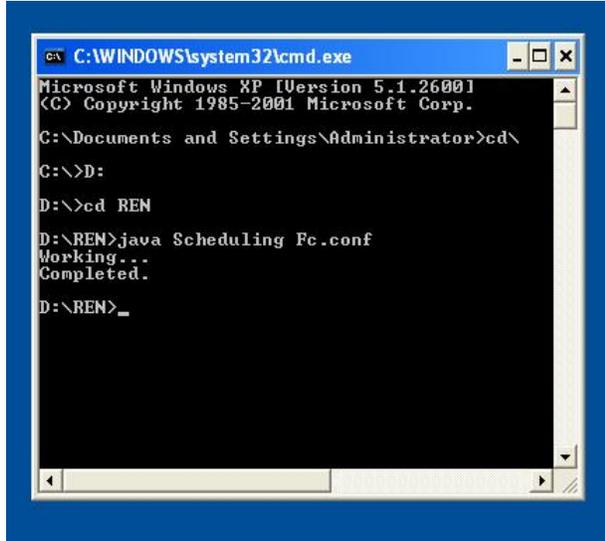


Figure 2: Command execution

As Figure 1 represents the initialization process for scheduling using the command. We use the command “Scheduling” as prefix of conf file to schedule the Fc.conf file. After the command scheduling process completed with working completed prompt as shown in figure 2. The execution process generates the two result files which name as SUM_PROC and SUM_RSUL.

Step 3: Two files generated after running the Fc file as
 (a) SUM_RSUL

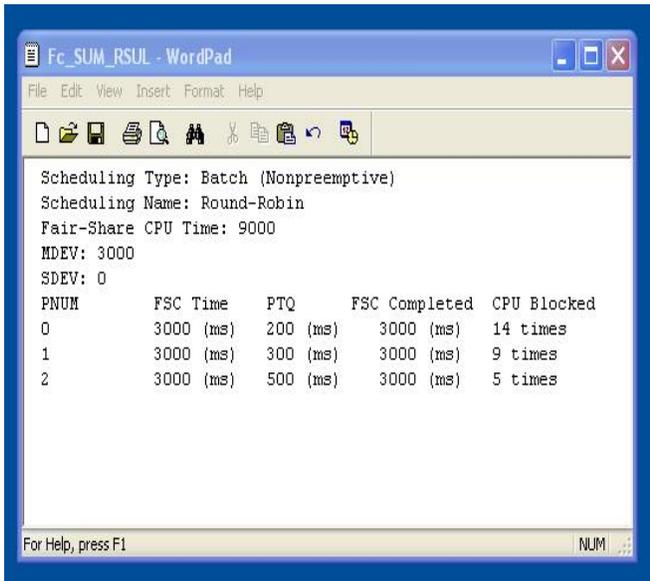


Figure 3: Fc_SUM_RSUL File

As the above figure 3 shows the five entries generated as result (PNUM, FSC Time, PTQ, FSC Completed and CPU Blocked), FSC time given as 9000 milliseconds which equally distributed among the three processes as 3000 milliseconds per process, during this time CPU blocked 14 times for P0, 9 times for P1 and 5 times for P2 with PTQ 200, 300 and 500 milliseconds for the process P0, P1, P2 respectively.[9]

(b) SUM_PROC

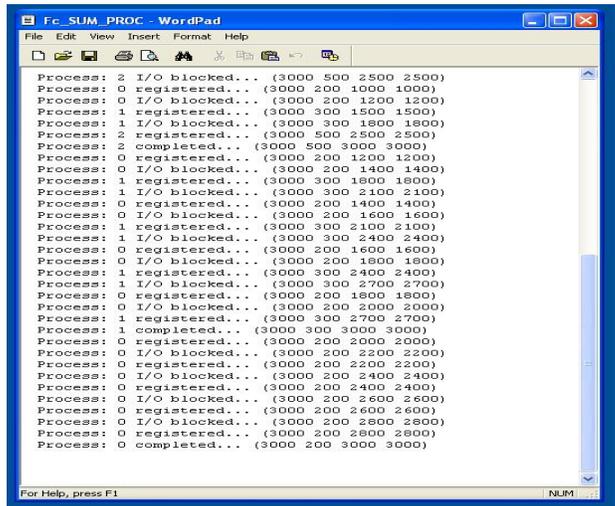
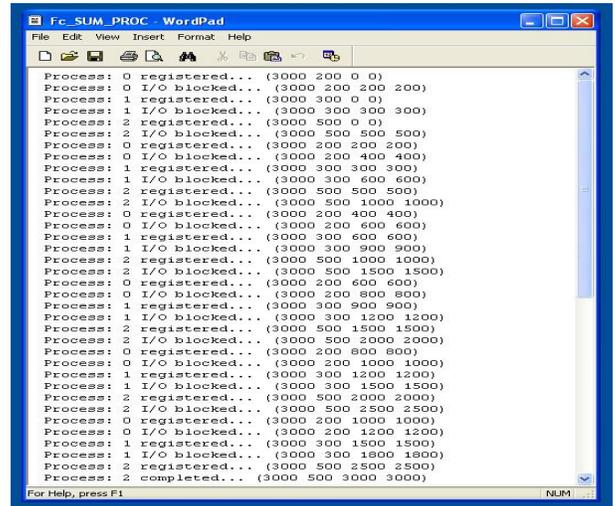


Figure 4: Fc_SUM_PROC File

As shown in Fig 4, there are 3-states for each process during scheduling as registered, I/O blocked and completed, each process work according to given quantum.

CASE 2: Taken maximum Time-Quantum among given time-quantum for each process and proceed scheduling with the following steps:

Step 1: Creating the configuration file Fc1.

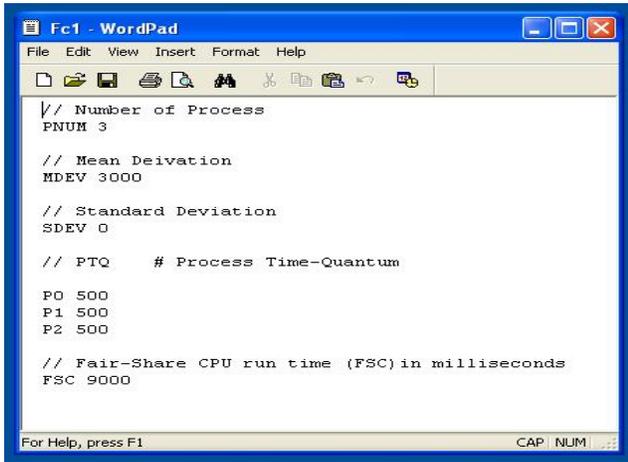


Figure 5: Fc1 File

In this modified configuration file Fc1 we again declare some parameters with the constraints. As shown in fig 5 the first one is number of processes (PNUM) which is 3 (P0, P1 and P2 with PTQ 500, 500 and 500 respectively as maximum quantum), Mean Deviation (MDEV) with value of 3000 milliseconds, Standard Deviation (SDEV) by default zero and Fair-Share CPU run time (FSC) is 9000 milliseconds.

Step 2: Initialization of Simulation Process:

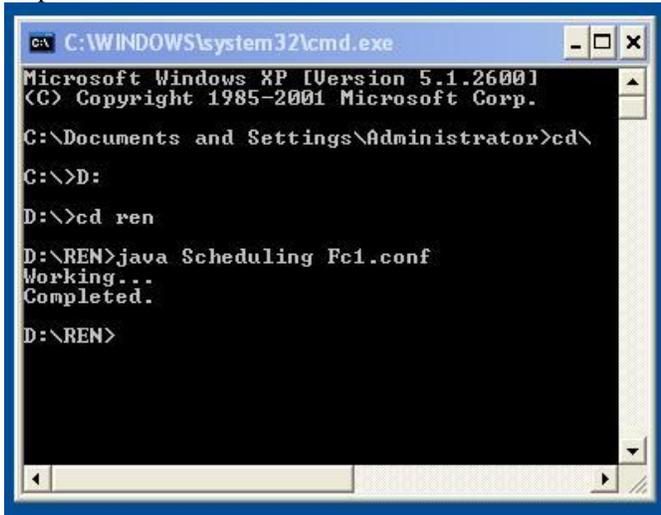


Figure 6: Command execution

As figure 5 represents the initialization process for scheduling using the command. We use again the command "Scheduling" as prefix of conf file to schedule the Fc.conf file. After the command scheduling process completed with working completed prompt as shown in figure 6. The execution process generates the two result files which name as SUM_PROC and SUM_RSUL.

Step 3: Two files generated after running the Fc1 file as

(a) Fc1_SUM_RSUL

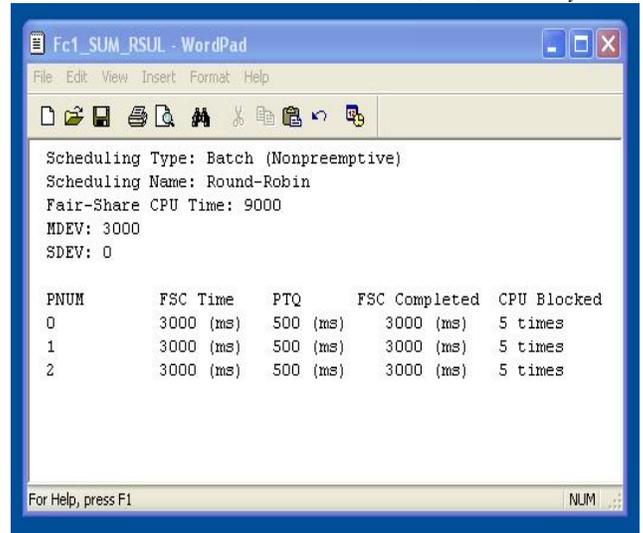


Figure 7: Fc1_SUM_RSUL File

As the above figure 7 shows the five entries generated as result (PNUM, FSC Time, PTQ, FSC Completed and CPU Blocked), FSC time given as 9000 milliseconds which equally distributed among the three processes as 3000 milliseconds per process, during this time CPU blocked 5 times for P0, 5 times for P1 and 5 times for P2 with PTQ 500, 500 and 500 milliseconds for the process P0, P1, P2 respectively which is our desired result with less CPU blocking.

(b) Fc_SUM_PROC

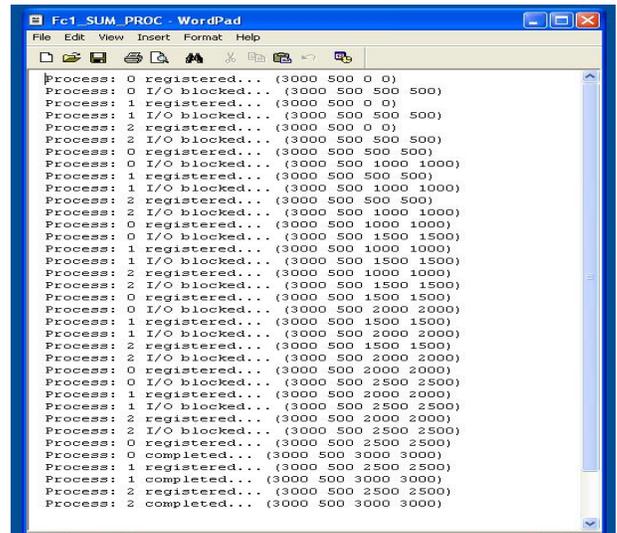


Figure 8: Fc1_SUM_PROC File

As shown in Fig, 8 there are 3-states for each process during scheduling as registered, I/O blocked and completed, each process work according to given quantum.

CONCLUSION

The benefit which is provided by this simulation is reduced hardware cost and also reduced cost for working on it, as it is fully dependent on software for simulation that is why it gives accurate results as compared to hardware based system, this simulation is for reducing CPU blocking and helps in full utilization of CPU.

The algorithm proposed in this paper describes the full round-robin scheduling process in an efficient manner. And also tell how we can improve the round-robin scheduling by reducing CPU blocking.

ACKNOWLEDGMENT

The authors are thankful to the Department of Information and communication, Suresh Gyan Vihar university, Jaipur, India for providing research facilities and their faculty for being the constant source of inspiration. The authors would like to thank CIITM, Jaipur for his valuable support during the preparation of this paper.

REFERENCES

- [1]. A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis
- [2]. http://www.cs.uic.edu/~jbell/CourseNotes/Operating Systems/5_CPU_Scheduling.html
- [3]. Rami J. Matarneh, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes", American Journal of Applied Sciences, Vol 6, No. 10, 2009.
- [4]. Silberschatz, Galvin and Gagne, Operating systems concepts, 8th edition, Wiley, 2009.
- [5]. "Tanebaun, A.S., 2008" Modern Operating Systems. 3rd Edn. Prentice Hall, ISBN: 13: 9780136006633, pp: 1104.
- [6]. "Tong, W. and J. Zhao, 2007" Quantum varying deficit round robin scheduling over priority queues. Proceedings of the International Conference on Computational Intelligence and Security, Dec. 15- 19, Computer Society, Harbin, China, pp: 252-256.
- [7]. Richard roehi (1995) "Designing a fairer round robin scheduling algorithm" SIGCOMM '95 Cambridge, MA USA 0 ACM. www.cmg.org/proceedings/2005/5056.pdf.
- [8]. <http://mat.gsia.cmu.edu/trick/survey.pdf>
- [9]. <http://www.ontko.com/moss/#sched>

AUTHORS PROFILE



Renu Tiwari pursuing M.Tech (Information Communication) from Suresh Gyan Vihar University, Jaipur (Raj), India and completed her Masters in Computer Application from Rajasthan University, India in the year 2008. Her area of interests lies in the field of Operating System, Database

Management, Information System Securities and research areas. She is working as a Lecturer in MCA Department of CIITM College, Jaipur (Raj), India. She has three years Academic/Research experience.



Prof. (Dr.) Naveen Hemrajani, Principal (Engg.),SGVU and Chairman CSI(Jaipur Chapter) received his B.E degree in Computer Science & Engineering from Shivaji University in the year 1992 and M.Tech (CSE) in 2004. His Research Topic for PhD was Admission Control for

Video Transmission over IP Networks. He possesses 20 years of Teaching and research experience. He has published three books and many research papers in International and National Journals of repute. He has also presented several papers in International and National conferences. He is also Editorial Board member of many international Journals of repute. He is also working on DST (Department of Science & Tech.) sanctioned project.