A Novel CPU Scheduling with Variable Time Quantumbased on Mean Difference of Burst Time

Sushil KumarSaroj
CSEDepartment
GCET,Gr. Noida

Aravendra Kumar Sharma
CSEDepartment
GalgotiasUniversity,Gr. Noida

Sanjeev Kumar Chauhan
Intellicus Technologies
Private Limited

Abstract-**CPU scheduling has significant contribution in efficient utilization of computer resources and increases thesystem performance by switchingthe CPU among the various processes.However, it also introduces some problems such as starvation, largeaverage waiting time, turnaround time andits practical implementation.Many CPU scheduling algorithmsare given to resolve these problems but they are lacked in some ways. Most of the given algorithms tried to resolve one problem but lead to others.To remove these problems, we introduce an approach that usesbothaverage and variable time quantum. In this approach, some processes are served with average time quantum and others with variable time quantum. This approach not only provides the minimum average waiting time and turnaround time but also try to prevent the starvation problem.**

***Keywords*:CPU scheduling, time quantum, waiting time, turnaround time, process, sorted queue, burst time.**

I. INTRODUCTION

CPU Scheduling is a basic and desirable works of any operating system. Since,before use,almostall computer resources are scheduled first. CPU scheduling [1][2][6][8] aretwo types, preemptive and non-preemptive scheduling. Pre-emption isan ability of the operating system to preempt or stop a currentlyscheduled task in favour of a higher priority task. Non-preempt ability arises,for instance, when handling an interrupt. In this case, scheduling is avoided until theinterruptis handled. Making a scheduler pre-emptible has the advantage of better systemresponsiveness and scalability.

Like the memory, CPU is the primary computer resource. Thus,to design operating System,CPU schedulingalgorithm'srole is crucial.Operating system must decide which one is to run first when there aremany runnable processes are available.Scheduler that is the part of the operating system,take care of it and the algorithm it uses is called scheduling algorithm. There are three distinct types of schedulers:long-term, mid-term and short-term scheduler.

Long-term Scheduler:
When a process enters in the system, it is put in the job queue. The long-term scheduler also called admission scheduler decides which processes are to be broughtinto the ready queue from the job queue for execution [1][3].

Mid-term Scheduler:
The mid-term scheduler is useful in time sharing system where processes are moving from main memory to secondary memory and vice versa. Which processes are to be moved is decided by mid-term scheduler [1][3].

Short-term Scheduler:
The short-term scheduler is also called CPU scheduler. It allocates CPU to the processin the ready queue. Means whichprocess execute next is decide by short term scheduler. It takes scheduling decision much faster than mid-term and long term scheduler. It may preemptive and non-preemptive [1][3].

Dispatcher:
A dispatcher is the module that gives control of the CPU tothe process selectedby the CPU scheduler. The dispatcher invoked during every processswitch.So it should be as fast as possible.High quality scheduling algorithm design affects the success of a CPU scheduler.High-quality CPU scheduling algorithms depends on scheduling criteria such as throughput, turnaround time,CPUutilization rate, time responseandwaiting time. Thus, themain focus of the work is to invent ageneralized, high quality,optimal CPU schedulingalgorithm appropriate for any types of processes [1][3][4].

Context Switching:
Context switching is a desirable feature of time sharing systems. A time sharingsystem is one in which multiple processes are running on a single CPU seemingparallel and without interrupting with each other. This illusion of parallelism isachieved by means of context switching that are occurring in rapid succession. These context-switches happen as a result of processes themselves release the CPU or as a result of the scheduler making the switchwhen a process has used up its time quantum. Context switches happen only in kernelmode. Kernel mode is a supervised mode of the CPU and provides access to all memory locations and all other system resources.Context switching is generally computational overhead. That is, itrequires CPU time which is substantial [10][11][12].

II. RELATED WORK

This section describes some of the research works done related to CPU scheduling algorithms. The basic CPU scheduling algorithms and their characteristics are discussed in thissection.

ISBN:978-1-5090-1666-2/16/$31.00 ©2016 IEEE

2.1 Algorithms and Its Characteristics

2.1.1 First Come First Serve

The most simple and fair approach is to allow the first process to executefirst. This scheme is called as first-come, first-served (FCFS) scheduling [1][2][3][9].

Algorithm:

Step 1: Allocate CPU to first requested process.

Step 2: The new processes are put at the tail of the ready queue.

Step 3: When the process finishes execution, fetch the next process from head of ready queueand allocate CPU to it.

Characteristics:

1: Absence of prioritization does allow every process to eventually finish execution, hence nostarvation problem.

2: Waiting time, turnaround time and response time is large.

3: Longest burst time process can monopolize CPU, even if burst times of other processes are too short. Hence, throughput is very low.

2.1.2 Shortest Job First

In SJF[1][2][3][9], processes are arranged in such way so that least burst time process at the head and longest burst time process at the tail of ready queue.

Algorithm:

Step 1: Here, CPU is allocated to the process that has the shortest burst time.

Step 2: Allocate the CPU to the processesaccording to the FCFS basis if, two or more than two processes have equal burst time.

Step 3: When a process completes execution, fetch the next process from head of ready queueand execute it.

Step 4: Partially executed process are swapped out and put in tail of ready queue.

Characteristics:

1: The actual difficulty with the SJF algorithm isto have prior knowledge about burst time of next CPU request.

2:SJF has the minimal average waiting timebecause it services smaller processesbefore larger ones.

3:It reduces average waiting and turnaround time but it may starvea process when a larger burst time process has made request but there are too many incoming shorter burst time processes.

2.1.3 Priority Scheduling

In priority scheduling, lower priority process always get interruptedby incoming higher priority processes [5][7][9].

Algorithm:

Step 1: A priority is assignedto each process in the ready queue.

Step 2: Assign the CPU to higher priority process first for time quantum and so on.

Step 3: Assign the CPU to the processes according to the FCFS basis if, two or more than two processes have same priority.

Characteristics:

1:Low priority processmay get starve.

2:For the equal priority processes, the waiting time gradually increases.

3:Here, waiting and response time for higher priority processes have smaller.

2.1.4 Round Robin

Round Robin algorithm [1][2][3][9] uses time quantum that is slice of time.

Algorithm:

Step 1: Select the time quantum and allocateCPU to each process of ready queue.

Step 2: Assign the CPU to each process according to the FCFS basis.

Step 3: If burst time of the process is less than or equal to time quantum, assign the CPU to that process still it finishes the execution. Else the process will hold the CPU till time quantum and isadded to the tail of the queue for the second round of execution.

Step 4: New processes are added to the tail of the ready queue.

Characteristics:

1: Selecting the time quantum too small causes too many context switches and decreases the CPUthroughput and efficiency.

2: Selecting the time quantum too large may cause poor response time.

3:Deadlines are rarely achieved in a pure Round Robin system because of largeaverage waiting times.

2.1.5 SRV

SRV [3] uses the working methods of basicCPU scheduling algorithms. The algorithm works as follows:

Algorithm:

Step 1: Calculate the time quantum

$$\text{Time quantum} = \left\{\sum_{i=1}^{n} P_i\right\}/n$$

Where, P_i is the burst time of i^{th}Process, n is the number of processes.

Step 2: Allocate this Time Quantum to each process of the ready queue.

Step 3: Arrange the processes in such way so that least burst time process at the head and longest burst time process at the tail of ready queue.

Step 4: Assign the CPU to the processes according to FCFS basis if, two or more processes have same burst time.

Step 5: If burst time of a process is less than Time Quantum, assign the CPU to that process still it terminates. Elsethe process will hold the CPU till the Time Quantum and isput to the tail of the ready queue for the second round of execution.

Characteristics:

1: Here,starvation problem of a longer process can be removed by assigningCPU for a time quantum to each.

2: No process can monopolize CPU.

3: Waiting time,response time, turnaround time and CPU utilization and are average.

This section describes various CPU scheduling algorithms such as FCFS, SJF, Round Robin, Priority scheduling and SRV. Here, their characteristics,advantages and various issues are discussed. After counseling all scheduling algorithms,it is found that there is still need a CPU scheduling algorithm that can remove all these problems. The proposed CPU scheduling algorithm tries to solve these problems.

III. MODEL AND ASSUMPTIONS

The proposed CPU scheduling model contains two queue, sortedqueue(readyqueue) andwaiting queue as shown in fig.1. Ready queue is called sorted queue here because all the processesfor fixed interval are sorted in ready queue. Processesare sorted in increasing order of their burst time. So, ready queue contain smallest bursttime at head. Scheduler selects the process from head of ready queue and assigns the CPU.Now, if process has burst time less than or equal to time quantum then it is executed in singleCPU assignment and relinquish the CPU voluntarily otherwise partially executed processis swapped out and put it to tail of sorted queue for second round of execution.

Now, CPU is assigned to next process in sorted queue. If, a process is waiting for I/Oduring execution then that process is swapped out immediatelyand put it to waiting queue and later enter thetail of sorted queue on its turn. Here, it is assumedthat no new process is adding in tail of ready queue. Scheduling is done among processesthat are already taken. For second round of execution of remaining processes, same procedure is fallowed.

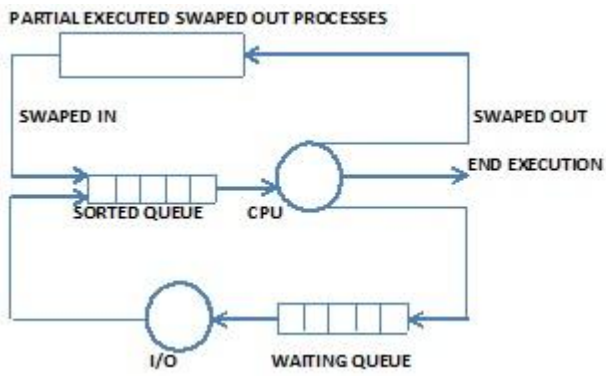

Fig. 1Proposed Scheduling Model

IV. PROPOSED SCHEME

There are some concepts that we shouldknow to understand the proposed scheme better.

Context Switching:

A major focus in the design of proposed CPU scheduling algorithm is to avoid unnecessary context switching as possible. The proposed scheduling algorithmprovides asolution to context switching. The suitable time quantum is assigned to all theprocesses in the queue for execution and numbers of processes get completed in first or just few round of assigned time quantum. The significant decrease innumber of processes will lead to substantial reduction in the number of context switch, which show high overhead on the operating system in several cases.

The number of context switches can be calculated mathematically as follows [5]:

$$\text{Context Switching Time} = \left\{\sum_{i=1}^{n} P_i\right\}-1$$

Where, P_i is i^{th}process's burst time and n is total no of processes.

.

Waiting Time:

The total amount of time that a process has spent in sorted queue (ready queue) is called waiting time [15]. The average waiting time is calculated using the following equation:

$$\text{Average Waiting Time } (AW_T) = \left\{\sum_{i=1}^{n} W_{Ti}\right\}/n$$

Where,$W_{Ti}$is the waiting time ofi^{th} process and n is total no of processes.

Turnaround Time:

It is the time interval from submission of a process and completion of execution of that process. The average turnaround time is calculatedusing following equation:

$$\text{Average turnaround Time } (AT_T) = \left\{\sum_{i=1}^{n} T_{Ti}\right\}/n$$

Where,$T_{Ti}$is theturnaround time of i^{th}processand n is total no of processes.

Time Quantum:

Time quantum or time slice is basically a small unit of time. In CPU scheduling, timequantum is generally in milliseconds.

Average Time Quantum:

Time quantum can be taken randomly but here time quantum is calculated as follows:

$$\text{Average Time quantum (TQ)} = \left\{\sum_{i=1}^{n} P_i\right\}/n$$

Where, P_i is the i^{th} process's burst time and n is total no of processes.

Mean Difference:

The mean difference is a measure of statistical dispersion equal to the average of consecutivedifference of two or more individual values.

Example:

Process ID	Burst Time
P1	X3
P2	X1
P3	X5
P4	X2
P5	X4

Table 1: Processes and their Burst Time

Arrange the processes in increasing order of their burst time. Suppose increasing order of their burst time as follows

X1, X2, X3, X4, X5

Take consecutive differences of burst time of processes and add them.

(X2-X1)+(X3-X2)+(X4-X3)+(X5-X4)

Take the Mean Difference (C) of them.

Mean Difference (C) =(X2-X1)+(X3-X2)+(X4-X3)+(X5-X4)/no of terms (means total no of processes -1)

Variable Time Quantum:

Variable time quantum is computedas follows:
Variable Time Quantum (VTQ) for a process:
Variable Time Quantum (VTQ) = Average Time Quantum (TQ) + no of turn of the process(turn of the process is basically position of that process among those processeshaving burst time greater than average time quantum (TQ) each) * C

Example:

Time quantum for first process (first in those who have burst time greater than averagetime quantum each)

VTQ_1= TQ+1*C

Time quantum for second process

VTQ_2 = TQ + 2*C

Time quantum for third process

VTQ_3 = TQ + 3*C

Time quantum for forth process

VTQ_4 = TQ + 4*C

- .
- .
- .
- .
- .
- .

Time quantum for i^{th} process

VTQ_i = TQ + i*C

Where, i is an integer variable.

Proposed Scheduling Algorithm:

Round Robin and SRV are CPU scheduling algorithms based on time quantum. Theirperformance solely depends on the time quantum. If, time quantum is too large then RoundRobin degenerates to FCFS. And if time quantum is too small then context switchingoccur more which is computational overhead. SRV CPU scheduling algorithm is sameas Round Robin but in place of FCFS as in Round Robin here processes are served inSJF fashion, having its own calculated time quantum. But performance of SRV depends on its calculated time quantum. So, selection of timequantum is crucial. The proposed approach is same as SRV scheduling algorithm but itworks with two types of time quantum.

In the proposed algorithm, first calculate the average time quantum (TQ). Then arrange all the processes in increasing order of their burst time. After that, calculate the Mean Difference (C) and Variable time quantum (VTQ). Processes havingtheirburst time less than or equal to average time quantum (TQ) each are served with averagetime quantum (TQ) and all these processes will be finished in just single time quantum and rest of the processes willbe served with variable time quantum (VTQ) but this VTQ is not same for all rest of the processes. VTQ is different for different processes. It depends on process's turn. Actually, VTQ is TQ where for each incoming process (processes having burst timegreater than average time quantum (TQ)); value equal to mean difference (C) is added toprevious time quantum.Variable time quantum (VTQ) is calculated on TQ, mean difference (C) and process's turn.

Proposed Scheduling Algorithm:

Step1. Calculate the Average Time Quantum (TQ)

Average Time quantum (TQ) = $\{\sum_{i=1}^{n} P_i\}/n$

Where, P_i is the burst time of i^{th} process and n is total no of processes.

Step2. Arrange all the processes in increasing order of their bursttime.

Step3. Calculate the Mean Difference (C) of burst times ofall the processes.

Step4. Calculate the Variable Time Quantum (VTQ_i)

$VTQ_i = TQ + i*C$
Where, i is an integer variable.

Step5.If two or more processes have equal burst time
{
Assign the CPU to the processes according to FCFS basis.
}

Step6. If (burst time of a process <= Average Time Quantum (TQ))
{
Assign the CPU to that process till it is finished. And that process will be served with timequantum that is equal to Average Time Quantum (TQ).
}
Else
{
Each incoming process P_i is allocated CPU for different time quantum called Variable Time quantum (VTQ_i). The process will hold the CPU till its VTQ_i and added to tail of sorted queue for next round of execution if, it is not finished within given time quantum VTQ_i.
}

V. PERFORMANCE ANALYSIS AND SIMULATION

There are given several algorithms on CPUscheduling and they have different characteristics. The selection of a specific algorithm may be betterfor a type processes over others. One algorithm may be better over others for some requirements. If, we want optimal average waiting time and turnaround time then SJF is the best. But it suffers with starvation and also has problem to implement it practically. Now, if we want a fair and simplest algorithm then FCFS is good but it has large averagewaiting time, turnaround time. It also suffers with starvation problem. Priority scheduling algorithm has large averagewaiting time, turnaround time. It also suffers with starvation problem.Round Robin and SRV are CPU scheduling algorithms based on time quantum. Theirperformance solely depends on selection of time quantum. If, time quantum is too large then RoundRobin degenerates to FCFS. And if time quantum is too small then context switching occurs more which is computational overhead. Same as with SRV algorithm, if time quantum is too large then SRV degenerates to SJF. And if time quantum is too small then context switching occurs more. SRV uses the time quantum that is the average of burst time of processes.Round Robin and SRV both use the singleand same time quantum for all the processes. Their performances are average in terms of waiting and turnaround time. Actually, as we try to move towards optimal average waiting time and turnaround time, starvation problem increases. Because optimal average waiting time and turnaround time achieve when we try to execute shortest job first, means we are setting a type of priority among them. Where priority applies, starvation problem arises.So, all algorithms try to achieve minimum average waiting time and turnaround time may have starvation problem at some level.

In the proposed scheme, all processes having burst time less than or equal to average time quantum (TQ), are finished executionin single round and rest of the processes are finished in single orin few rounds.The proposed scheme shows optimal result when allprocesses have equal burst time or their burst time differences are multiple of mean difference of their burst time because at that situation each process will finish in single round. But in general, proposed scheme shows minimumaverage waiting time and turnaround time compare to others except SJF. It also removes the starvation problemat certain level because each process can hold CPU tilltime quantum assigned to it. In the proposed algorithm, processes are fixed for a time interval unlike SJF and the algorithm is simple, so it is easy to implement. The proposed algorithm is basically a new and efficient approach where we want minimum turnaround timeand waiting time.

The proposed scheme is implemented and simulated with C programming. It may be a new approach in case of CPU scheduling and can be implemented easily in the operating system. Here, we can see the comparative results of some CPU scheduling algorithms. There are three samples of data over which algorithms are analyzed.

Ex.1

Process:	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Burst Time:	5	2	8	10	4	14	20	17	30	27
Priority	3	6	2	10	5	8	1	7	4	9

Ex.2

Process:	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Burst Time:	10	1	2	1	5	7	13	4	8	16
Priority	3	1	4	6	2	5	7	9	10	8

Ex.3

Process:	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Burst Time:	7	3	11	2	5	16	20	1	6	9
Priority	3	2	9	5	1	8	4	7	6	10

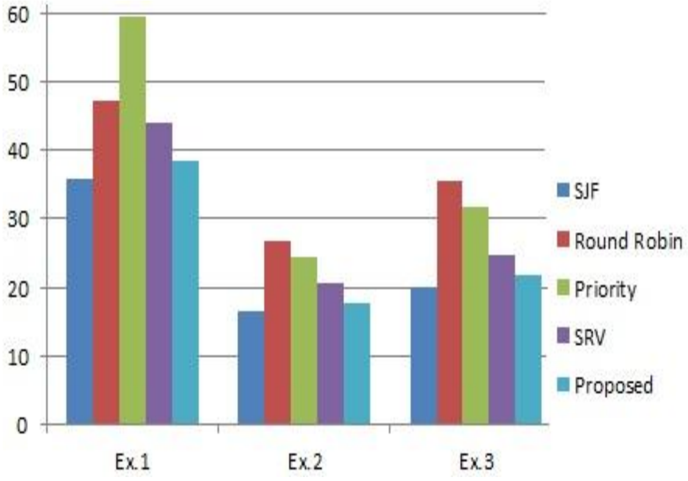

Fig. 2Average Waiting Time

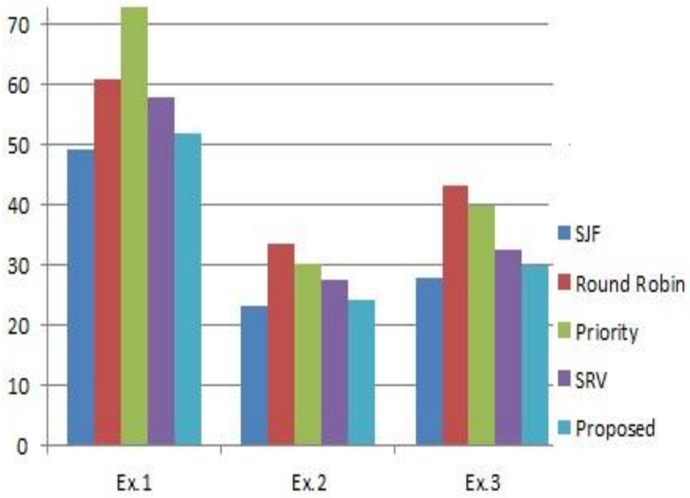

Fig. 3Average Turnaround Time

V. Conclusion

There are many criteria such as waitingtime, turnaround time, starvation and implementationupon which we can evaluate the CPU scheduling algorithms. Many scheduling algorithms have given but they have lacked in some ways. For example, SJF has optimal average waiting and turnaround time but suffer with starvation and its practical implementation problem.The proposed algorithm has removed the starvation problem and also easy to implement by employing two types of time quantum. In general, it has also minimum average turnaround time and waiting time compare to others except SJF.

References

[1] A. Silberschatz, P. Galvin, G. Gagne, "*Applied Operating System Concepts,*"First edition john Wiley Sons Inc., 2000.

[2] S. A. Tanenbaum, A. Woodhull, "*Operating System, Design,"* PrenticeHall; 3rd editions, 2006.

[3] M. Sindhu, R. Rajkamal, P. Vigneshwaran,"*An Optimum Multilevel CPU SchedulingAlgorithm,"* ACE, IEEEInternational Conference, 2010.

[4] S. Shah, A. Mahmood, A. Oxley,*"Hybrid Scheduling and Dual Queue Scheduling,"*Computer Science and Information Technology, 2009.

[5] R. Matarnesh, "*Self Adjustment Time Quantum in Round Robin Algorithm Dependingon Burst Time Of the Now Running Processes,*" American Journal of AppliedScience,2009.

[6] P.Kokkinsis, "*A Software Tool for Process Scheduling*," report, 2007.

[7] J.Lakma, "*Fortifying the Operating System CPU Scheduler,*" A project reportsubmitted to csed of Makerere Uni., 2005.

[8] Congnizant Handout, "*Fundamentals of Computer Technology,*" Version:FCT/Handout/0307/7.1, 2007.

[9] Md. A. F. AlHusainy,"*Best -Job-First CPU SchedulingAlgorithm,*" InformationTechnology Journal, 2007.

[10]Dr. R. Sakellariou,"*Operating System Module, Process Scheduling,*" 2nd year lecture notes, 2005/6.

[11] Md. M. Rashid, Md. N.Adhar, "*A New Multilevel CPU SchedulingAlgorithm,*" Journal of Applied Science, 2009.

[12] S.Suranautwarat, "A *CPU Scheduling Algorithm Simulator,*" 37th ASEE/IEEE Frontiers in Education Conference, 2009.