Name= Muhammad Mubbashir

Abstract

New Operating Systems are converted into multitask environment from single task environment. Now today operating system demand is to utilize max number of computing. There are many scheduling algorithms are using today. This algorithm is not using in normal life just because of high context switching. There are four main objective which is use to achieve a better performance in operating system High Response time, large average wait, less throughput and large turnaround time. We use this objective to achieve a new scheduling algorithm. The objective of this paper is to develop a new approach for primitive shortest job first which help to improve the efficiency of computing in real time and also time-sharing environment. In the result portion we propose algorithm to improve the system performance by decreasing context switching.

Introduction:

The modern operating system uses a multiprogramming paradigm to give an illusion of running multiple processes simultaneously. This phenomenon can be achieved by many process scheduling algorithms, one such algorithm is Shortest Job First algorithm which runs on a principle of non-preemption i-e: Executing the process with the lowest burst time. In contrast to this, the “Short Remaining Time First Schedule” was proposed to handle the preemptive mode of process scheduling in which a process runs for a specific time quantum and then upon having multiple processes in the ready queue, the process with the minimum burst time is then executed for the same time quantum.

The idea of this algorithm is that when it first starts, it chooses the job with the least burst time and starts the process. It is a preemptive scheduling algorithm, which means that when a new job comes in the ready queue, it chooses whether or not to preempt the running process. Context switching, as we all know, is an overhead that reduces system performance. It also reduces the amount of time a process takes to complete.

The main drawback of SJF is to high time context switching and also more waiting time of a process. The process had higher wait time and higher context switching which was an overhead and this led to compromised system performance. For high context switching we come up with new proposed algorithm which give us low context switching and low waiting time of a process.

Literature review

CPU scheduling is the main and basic operation of operating system. There are many operating system all of them are use CPU scheduling concept. This concept shows in many operating systems books.

A few concepts use in operating system which is flow shop and job shop. it is use to manage difficulties scheduling. In Job shop use commonly in general purpose resources and it is very highly flexible. In Flow shop use commonly special resources and it use fixed path. It is needing to edit the actual working program for every application area. To start first quickly explain about the old existing algorithm which is no best for today computers. In this we have ROUNDROBIN, SJF, SRTF, PRIORITY, FIFO and many more. Many different model and simulation are used to identify the existing algorithm. Many Researchers are publishing their paper on the topic of operating system scheduling algorithm. For example, in priority base algorithm the process which is high priority are selected first and then it performs execution. In all algorithm when the routine completes its operation under their execution time then the process is terminated and then it removes from process list. The new process is selected for ready queue to execute next process. When process is successfully executed it remove from waiting process list. In SJF the shortest burst time are placed in front of ready queue.

Methodology:

The new proposed algorithm is a hybrid scheduling algorithm. In this proposed algorithm we combine two scheduling algorithms to improve operating system performance, the first one is shortest job first (SJF) and other one is the constraint of remaining burst time of the running process. Hybrid scheduling based on preemptive scheduling algorithm. The concept of this algorithm is that first select the shortest job first and then executes of that process.as we declare that it is preemptive scheduling so when job is arrived in ready queue so first its check that weather is it preemptive process or not. If the remaining execution of process is remaining and which is less than and equal to newly process execution time so the it will not preempt the process. In this new hybrid model, there are two possibilities **1) "If half of the remaining execution time of the running process is less than the burst time of the new process, then the running process will not preempt." 2) "If half of the remaining execution time of the running process is greater than the burst time of the newly arrived process, then we will preempt the running process and assign the processor to the newly arrived process.".** in hybrid scheduling algorithm it maximizes the system performance by minimizing the context switching .as we know that it minimizes the context switching and it also minimize the waiting time of a process. The second name of this algorithm is conditional preemptive algorithm when the condition is decided that the process is preemptive or not.

PROCESS	TIME OF ARRIVAL	BURST DURATION
P1	0	4.0
P2	2	7.0
P3	5	5.0
P4	6	8.0
P5	8	9.0

PROPOSED ALGORITHM:

P1	P2	P3	P4	P5

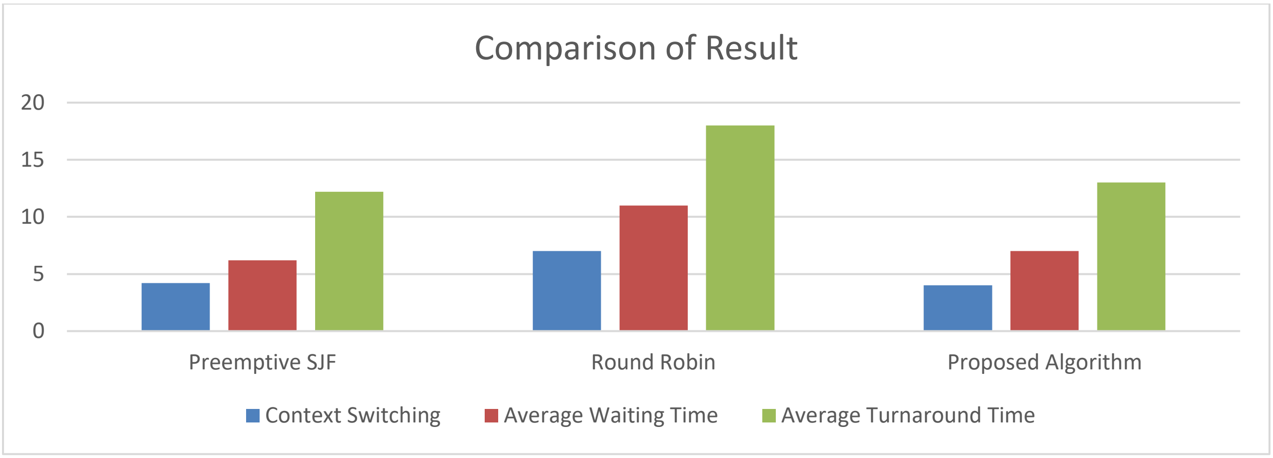
No of context switches = 4

Average Waiting time = 6.8ms

Average turnaround time = 13.4ms

RESULTS:

To test the effectiveness of the newly developed editing algorithm, we provide the same set processes in different editing algorithms. The comparative result shows that the proposed method has a limited content change as a comparison with the round robin and the original SJF. As we know the context switching is overhead too reduces system performance. The performance of the proposed method can adequately enhance the system working by minimizing context switching. Performance test of the proposed algorithm in comparison a round robin and a preemptive SJF are shown in the chart provided below,



Conclusion

As we see past results this shows that when we use First Come First Serve its calculation The overhead is small, but we realize that it produces less because of its poor performance. One The result of FCFS is that it has a high waiting time. The Shortest Job First is an ideal algorithm for editing to reduce the waiting time limit for a given set of available processes. The SJF concept offers a short average waiting time in every available process but has an extra waiting time for the process that requires more time to complete to do when comparing it with the FCFS algorithm. In the event of a short process to arrive continuously then hunger because a long process will take place. Round robin is an algorithm that provides good CPU sharing for all available procedures. In a round robin the quantum correction is given throughout the process in a straight line. Procedures with a short burst time can complete their execution with a one-time quantum, indicating the best response time. In the case of systems with longer explosion they offer longer switching time compared to their waiting time again increase. There is also the possibility of processes with a relatively short quantum process that can wait a long time to become their chance. Speaking of the pre-established system, we see that hunger also appears here. The process most importantly it will be done first so that the priority process may not get a chance to be executed, this creates a hunger for less important processes. Setting is much better than other algorithms available. The proposed approach provides sufficient reduction in context change with minimal change in median waiting time compared to SJF. But this change in median waiting time is very small. So, by doing a few themes you change these algorithms increase system performance. The main Drawback of Hybrid scheduling algorithm is that the Average waiting time of this algorithm is equal to the average waiting time of SJF. In hybrid scheduling we cannot see the improvement in average waiting time.