Volume 1, No. 2, September 2010
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

Novel Scheduling Algorithm for Uni-Processor Operating System

Saudagar Barade[*1] and Nikhil Khande[2]
*[1]CSIS, BITS,Pilani/ Pilani, Rajasthan, India
h2009@bits-pilani.ac.in[1]
[2]CSE, IITK/IIT,Kanpur/Kanpur, Uttar Pradesh, India
niksk@iitk.ac.in [2]

Abstract: One of the fundamental function of an operating system is scheduling. There are 2 types of uni-processor operating system in general.Those are uni-programming and multi-programming. Uni-programming operating system execute only single job at a time while multi-programming operating system is capable of executing multiple jobs concurrently. Resource utilization is the basic aim of multi-programming operating system. There are many scheduling algorithms available for multi-programming operating system. But our work focuses on design and development aspect of new and novel scheduling algorithm for multi-programming operating system in the view of optimisation. We developed a tool which gives output in the form of experimental results with respect to some standard scheduling algorithms e.g. First come first serve, shortest job first, round robin etc.
Keywords: Operating system; uni-processor; scheduling; Resource utilization; uni-programming; multi-programming

INTRODUCTION

Scheduling is the heart of any computer system since it contain decision of giving resources between possible processes. Sharing of computer resources between multiple processes is also called as scheduling.

The CPU is one of the primary computer resources, so its scheduling is essential to an operating system's design[1]. Efficient resource utilization is achieved by sharing system resources amongst multiple users and system processes[2].Optimum resource sharing depends on the efficient scheduling of competing users and system processes for the processor, which renders process scheduling an important aspect of a multiprogramming operating system. As the processor is the most important resource, process scheduling, which is called CPU scheduling, becomes all the more important in achieving the above mentioned objectives[1].

Part of the reason for using multiprogramming is that the operating system itself is implemented as one or more processes, so there must be a way for the operating system and application processes to share the CPU. Another main reason is the need for processes to perform I/O operations in the normal course of computation. Since I/O operations ordinarily require orders of magnitude more time to complete than do CPU instructions, multiprograming systems allocate the CPU to another process whenever a process invokes an I/O operation[3].

Goals for Scheduling

- Utilization/Efficiency: keep the CPU busy 100% of the time with useful work
- Throughput: maximize the number of jobs processed per hour.
- Turnaround time: from the time of submission to the time of completion, minimize the time batch users must wait for output
- Waiting time: Sum of times spent in ready queue - Minimize this
- Response Time: time from submission till the first response is produced, minimize response time for interactive users
- Fairness: make sure each process gets a fair share of the CPU

SHEDULING ALGORITHMMS

We will start with five commonly used scheduling algorithms [1,2,4,5]

1.First Come First Served Scheduling Algorithm (FCFS)
FCFS is the simplest scheduling algorithm. For this algorithm the ready queue is maintained as a FIFO queue. PCB (Process Control Block) of a process submitted to the system is linked to the tail of the queue. The algorithm dispatches processes from the head of the ready queue for execution by the CPU. When a process has completed its task it terminates and is deleted from the system. The next process is then dispatched from the head of the ready queue.

2.Shortest Job First Scheduling Algorithm (SJF)
For this algorithm the ready queue is maintained in order of CPU burst length, with the shortest burst length at the head of the queue. A PCB of a process submitted to the system is linked to the queue in accordance with its CPU burst length. The algorithm dispatches processes from the head of the ready queue for execution by the CPU. When a process has completed its task it terminates and is deleted from the system. The next process is then dispatched from the head of the ready queue.

3.Shortest Remaining Time First Scheduling Algorithm (SRTF)
For this algorithm the ready queue is maintained in order of CPU burst length, with the shortest burst length at the head of the queue. A PCB of a process submitted to the system has its CPU burst length compared with the remaining time of the PCB being executed. If the new process requires less time than that remaining of the active' process then preemption occurs and it becomes the new PCB's turn for execution, otherwise it

is linked to the queue in accordance with its CPU burst length. The algorithm dispatches processes from the head of the ready queue for execution by the CPU. When a process has completed its task it terminates and is deleted from the system. The next process is then dispatched from the head of the ready queue.

*4.**Round Robin Scheduling Algorithm (RR)***
For this algorithm the ready queue is maintained as a FIFO queue. A PCB of a process submitted to the system is linked to the tail of the queue. The algorithm dispatches processes from the head of the ready queue for execution by the CPU. Processes being executed are preempted on expiry of a time quantum, which is a systemdefined variable. A preempted process's PCB is linked to the tail of the ready queue. When a process has completed its task, i.e. before the expiry of the time quantum, it terminates and is deleted from the system. The next process is then dispatched from the head of the ready queue.

5.***Highest Response Ratio Next(HRRN)***
For this algorithm the ready queue is maintained in the order of response ratio. A PCB of a process submitted to the system is linked to the last PCB in the queue and once particular running process executes completely the processes currently present in ready queue are scheduled based on the response ratio which is given as (burst time+ waiting time) / burst time.

PROPOSED SCHEDULING ALGORITHM

RR have problem of high avg. WT(Waiting Time), SRTF gives starvation to longer jobs and though HHRN(Highest Response Ratio Next) is useful in avoiding problem of RR and SRTF, it fails in terms of responsiveness due to its non pre-emptive mode.
So we proposed this algorithm which will try to minimize avg. WT and starvation to longer jobs . And also increases the responsiveness due to its pre-emptive nature.
For 1st job time given(TG) is 2*TQ.It will be useful since many processes may come in that time, which is useful for making effective decision ahead as we have many processes to choose from.

Let,

TQ(time quantum)=2;

NTQ(new)=2*TQ;

NBT(new burst time)= |_3*Avg.BT_|;

Initially,VNTQ(varying)=NTQ;

VNTQ (increased or decreased by 1 accordingly) used to give appropriate time as per process's WT req.

T=Threshold=3*TQ; //to compare WT(waiting time)
TT=total time;
PC=process completes;
P=pre-emption;
RQ=ready queue

When process completes, rearrange RQ as per incremental order of remaining BT.
When process pre-empts then put it to last position in RQ and no changes in other process's order.

Algorithm in form of Pseudo code:

```
if( P && WT<NBT)
{
        No change in TG ; //TG=TQ
        ++VNTQ;
}

else    //to avoid starvation , try to give maximum
//time to that process
{
        TG=max(--VNTQ,TQ);
}


if( PC && TT<NBT)
{
        TG=max(--VNTQ,TQ);  //there is no
                        // starvation yet so try to
                // give less time
        VNTQ=NTQ;
}
else
{
        if( WT< T)
        TG=min(max(--VNTQ,TQ),TQ);

        else
        TG=NTQ;  //means process waiting from
                // long time so give NTQ to him
}
```

Take an example in which we have some processes with arrival time (AT) and burst time (BT).We have to form a schedule which will try to minimize avg.WT(waiting time) and also try to avoid starvation to longer jobs.

Example

Process	AT	BT
1	0	4
2	1	5
3	2	2
4	3	3
5	4	6
6	5	1

ANALYSIS OF ALGORITHM BASED ON SOME SCHEDULING FACTORS
There are many factors like waiting time,turn around time(TAT) etc on which we can check the performance of the scheduling algorithm.
To test the performance we will use two parameters such as avg. WT, avg. TAT
Average WT is mentioned in queueing theory as follows:
Queueing theory is the mathematical study of waiting lines, or queues. The theory enables mathematical analysis of several related processes, including arriving at the (back of the) queue, waiting in the queue (essentially a storage process), and being served at the front of the queue. The theory permits the derivation and calculation of several performance measures including the average waiting time in the queue or the system, the expected number waiting or receiving service, and the probability of encountering the system in certain states, such as empty, full, having an available server or having to wait a certain time to be served[7].
Average TAT is nothing but the total time between submission of a process and its completion[6].

GANTT CHART FOR ROUND ROBIN ALGORITHM

A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project [8].

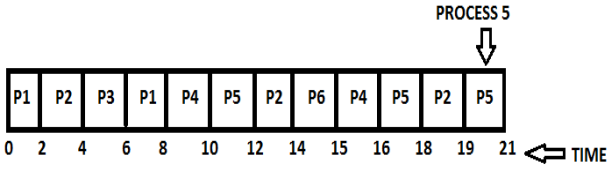

Figure 1:Gantt chart

Process	AT	BT	CT	TAT	WT
1	0	4	8	8	4
2	1	5	19	18	13
3	2	2	6	4	2
4	3	3	16	13	10
5	4	6	21	17	11
6	5	1	15	10	9
				---	---
			TOTAL	70	49

Average TAT for RR =70/6
=11.67

Average WT for RR =49/6
=8.16

Average TAT is nothing but sum of the all TAT of all the processes divided by number of processes.

Average WT is nothing but sum of the all WT of all the processes divided by number of processes.

These are some statistics based on avg. WT

FCFS : 7.12
SJF : 4.50
SRTF : 4.50 //prefers shorter jobs so starvation
//to longer jobs
RR : 8.16
HRRN : 4.50 //sometimes give pref. to longer jobs
//but have less responsiveness to some
//jobs due to its NP mode

Proposed: 5.00 //give pref. to longer jobs
//and also have high responsiveness to
//jobs due to its NP mode

Figure 1 shows the Gantt chart for the round robin schedule. Gantt chart is useful to show the process with its completion time and very useful tool to decide how many time one particular process is pre-empted. FIGURE 2 shows some of the experimental results obtained from the tool, which we created, based on the proposed novel algorithm.

SCHEDULING ALGORITHM	AVERAGE WATING TIME	AVERAGE TURN-AROUND TIME
FCFS	7.12	10.67
SJF	4.5	8.0
SRTF	4.5	8.0
RR	8.16	11.67
HRRN	4.5	7.66
PROPOSED	**5.0**	**8.5**

Figure 2.: Experimental Results

CONCLUSION

Our proposed novel scheduling algorithm (FCFS+RR) for uni-processor system gives good responsiveness and minimizes starvation for longer jobs, it also minimizes average waiting time of all processes. Our next focus of research is new load balancing algorithm for distributed computing system.

REFERENCES

[1] Hybrid Scheduling and Dual Queue Scheduling Syed Nasir Shah,Ahmad Mahmood,Alan Oxley 2009-IEEE 978-1-4244-4520-2/09 Conference

[2] Operating System concepts 7th Edition,By Galvin,Silberschatz,Gange John Wiley & Sons, 2005

[3] http://www1bpt.bridgeport.edu/sed/projects/cs503/Spring_2001/kode/os/scheduling.htm

[4] Milan Milenkovic, "Operting System Concepts and Design", Second Edition McGraw Hill International, 1992

[5] Leland L. Beck, "System Software", 3rd Ed., Addison Wesley, 1997

[6] http://en.wikipedia.org/wiki/Turnaround

[7] http://en.wikipedia.org/wiki/Queueing_theory

[8] http://en.wikipedia.org/wiki/Gantt_chart

AUTHOR

Saudagar Barade is student of Master of Engineering in Software Systems at BITS,Pilani,India.His research interest lies in the field of Operating System, Distributed computing and Software Engineering.

Nikhil Khande is student of Master of Technology in Computer Science at IIT,Kanpur,India.His research interest lies in field of Operating System, Computer Networks and Theory of Computation.