Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 12; October-December 2014 pp. 137-141

http://www.krishisanskriti.org/jbaer.html

Non-Preemptive Real-Time Scheduling of Earliest Deadline Approach through Periodic and Sporadic Tasks

S. Eswaraiah Naidu[1], Abdul Rahim.B[2] and Fahimuddin Shaik[3]

[1,2,3]Department of E.C.E Annamacharya Institute of Technology & Sciences, Rajampet, A.P-516126, India
E-mail: [1]eswarnaidun27@gmail.com, [2]prof.abraheem@hotmail.com, [3]fahim6285@gmail.com

Abstract: *Scheduling is the process which assigns the resources to Different tasks depending on its deadline. Scheduling algorithms are individual for each of the task priority (or) deadline. Different approaches for scheduling are like Rate monotonic, Deadline monotonic, and Earliest deadline first, least slack time, Round-robin, weighted Round-robin etc, can be utilized. The problem of inconsistencies occur in these algorithms such as optimality, transient overload condition, resource sharing problem, CPU utilization ratio and task collision. Earliest Deadline First (EDF) is an optimal scheduling algorithm for uniprocessor real-time systems. When non-pre-emptive EDF shares a common stack, leading to vastly reduced memory requirements and needed to solve resource sharing and task collision. According to the EDF approach, it is one of the methods to improve processor utilization ratio by evaluating number of computations on task period. The performance of the EDF is calculated in terms of Success Ratio and Effective CPU Utilization. Time constraint is the main factor in real-time operating system. Task scheduling will be used to allocate some time frames and used to schedule the task based on priorities of the task and its execution. Our goal here is to verify for which scheduling strategy is best suited for real-time system applications. Although these systems are multiprocessor based heterogeneous, the task scheduling will be very difficult and a challenging one. We have used MATLAB for simulation and analysis.*

Keywords: *Real-Time System, Task Scheduling, Earliest Deadline First, Rate Monotonic, Deadline Monotonic, Round Robin.*

1. INTRODUCTION

This paper describes about non-preemptive scheduling of recurring (periodic / sporadic) task models, with applications to resource-constrained, single-processor real-time and embedded systems. In particular, it is concerned with scheduler architectures for use with such systems, consisting of a small amount of hardware (typically a timer / and interrupt controller) and software.

Real time scheduling technique is divided into two types: static and priority driven as shown in fig1. In static algorithm priorities are assigned at design time. No one can change the priorities because they are fixed. In priority driven approach priorities are assigned at the run time.

Real-time systems use scheduling algorithms to decide an order of execution of the tasks and an amount of time assigned for each task in the system so that no task (for hard real-time systems) or a minimum number of tasks (for soft real-time systems) misses their deadlines.

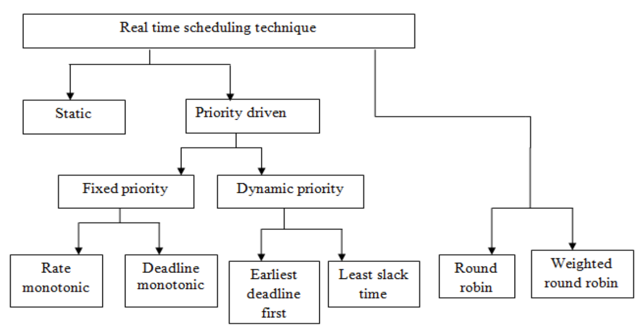

Fig. 1: Real time scheduling technique

There are some parameters that scheduling algorithms aim to fulfill. These includes; throughput, turnaround, response time, and waiting time. There are many scheduling algorithm such as round robin, first come first serve, shortest job first, least laxity first, rate monotonic, deadline monotonic, earliest deadline first[2].

1.1 Round Robin (RR)

RR gives every process with the same priority set share of time before making a context task, when all tasks have got their time share. The first task gets back into the CPU for its next processing. The advantages of RR algorithm is simple, low overhead, good interactivity. The disadvantage of the algorithm is, it include the absence of priority system. If quantum is too small, too much time wasted in context switching [2].

1.2 First-Come-First-Served (FCFS)

FCFS algorithm selects the task with the earliest arrival time. If system contains periodic tasks, their release time will be considered. The advantages of FCFS algorithm is simple and low overhead. The disadvantages are inappropriate for interactive systems, maximum turnaround time, and low throughput [8].

1.3 Least Laxity First (LLF)

LLF algorithm selects the task that has the lowest laxity among all the ready ones whenever processor become idle, and executes it to completion. The advantages of LLF are to decide which task should execute next at the schedule time and assigning fixed priorities to the task at development time. The disadvantage of the algorithm is it gives the poor runtime behavior [2].

1.4 Rate Monotonic (RM)

RM algorithm sets priority level for each task in order of their period length; tasks with short periods will get a high priority while task with long periods gets low priority. High priority task then take precedence before lower priority tasks. The disadvantage of RM algorithm is it may not give a feasible schedule even if processor is idle at some points [7].

1.5 Deadline Monotonic (DM)

DM assigns priorities to tasks according their relative deadlines. The shorter the relative deadline, higher the priority. The disadvantage of DM is not optimal for fixed priority non-preemptive scheduling [2].

1.6 Earliest Deadline First (EDF)

EDF schedules the task with the earliest deadline. EDF is the dynamic scheduling algorithm which places the task in a priority queue. The priority is assign at run time depending upon the deadline. A Task is schedulable under EDF, if and only if it satisfies the condition that the total processor utilization due to task is less than [1].

When EDF algorithm implemented in hard real time system there aren't any overloading conditions. But when EDF algorithm worked in soft real time system overload condition occurs. EDF algorithm decreases the performance when system overloaded. The first problem, it is difficult to predict which task will miss their deadlines during overloads. Next problem is a late job which has already missed its deadline has a higher priority than job whose deadline is still in the future. Each task in an EDF scheduler is assigned with a deadline. Every time a task is inserted in the system, the system looks for the other task which is present in the queue and which has the next nearest deadline and select it for execution [7]. In order to ensure that the scheduling application a scheduler must evaluate if each new incoming task doesn't overload the system and slow down the execution.

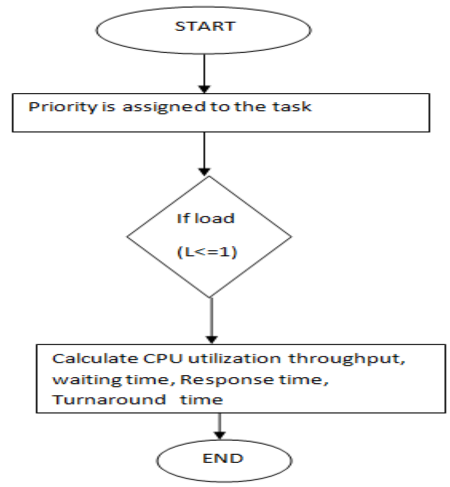

Fig. 2: Basic working of EDF

In Earliest Deadline First scheduling, at every scheduling point the task having the shortest deadline is taken up for scheduling. The problem of scheduling of such periodic task system on multiprocessor there are identical processors available which jobs may be generated by the periodic task ready for execution.

$$\sum_{i=0}^{n} \frac{ei}{\min(pi,di)} \le 1$$

Where **ei** is the execution time, **pi** is the priority of task and However, if **pi <di ,** it is possible that a set of tasks is EDF schedulable , even when the task set fail to meet according to expression number of task in set. When Earliest Feasible Deadline First is used to schedule a set of real-time tasks, unacceptable high overheads might have to be incurred to support resource sharing among the tasks without making tasks to miss their respective deadlines, due to this it will take again more time.

1. When according to the Non-Preemptive real-time scheduling, group-EDF techniques are used. In the group-EDF is based on grouping of tasks with deadlines that are very close to each other, and using Shortest Job First (SJF) technique to schedule tasks within the group. Whenever grouping of tasks with similar deadlines due to that task collision will occur and also under over load condition the computational complexity of group-EDF is more.

2. The non-preemptive EDF algorithm is universal for both periodic and sporadic tasks. We examined the problem of scheduling on a concrete set of periodic or sporadic tasks. Recall that a concrete task set consists of a task set together with release times of the tasks. However, for concrete periodic tasks the situation is more complex.

2. REAL-TIME SCHEDULING ALGORITHM

Many researchers have work in the field of real-time scheduling and it has been observed that, Different steps for scheduling are

- To check whether a system performs schedulability analysis.
- If it does then find it is done statistically or dynamically.
- Whether the result of the analysis itself produces a schedule or plan according to which task are dispatched at run time.
- Scheduling real time task in static priority scheduling algorithms for maximum CPU utilization but it can be increased more using dynamic priorities.

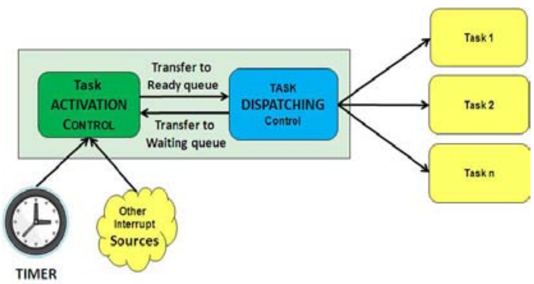

Fig. 3: Aspects of real-time embedded scheduling

In this context the two main aspects of a scheduler can be stated as follows,

2.1 Task activation

This is the process of deciding at which points in time a task becomes ready for execution (is activated). Periodic tasks are normally activated via a timer; event driven (sporadic) tasks can be either directly activated by interrupts.

2.2 Task dispatching

Real-time systems are required to perform specific processing in a timely fashion; when multiple tasks are simultaneously active, then some form of scheduling algorithm is normally required to process the events in an appropriate order.

Various researches have carried out their research in recent past for efficient real time task scheduling. For analyzing algorithms they used various terms like task types, task parameters, task priorities, scheduling categories, etc.

3. OBSERVATIONS

The comparison shows that the approach is different from EDF with the other methods based on the schedling criteria, time difference,task response, and processor performance

criteria	Rate monotonic (RMA)	Deadline monotonic (DM)	Earliest Dead line First(EDF)	Least Laxity First(LLF)	Round Robin(RR)	weighted Round Robin
Task importance	Equal importance to all tasks	Equal importance to all tasks	Task with earlier deadline is assigned higher priority	Which assigns the priority Based on slack time	Equal importance to all tasks	Equal importance to all tasks
Scheduling criteria	Task period	Task deadline	Earlier deadline	Task periodic	Ready queue can treated as a circular	Weight assigned to Cyclic scheduling
Task scheduling	Tasks are arranged in ascending order	Tasks are arranged in ascending order	No arrangement in any order	Tasks are arranged in ascending order	It is a first in first out process	It is a FCFS process
Time difference	Task period not vary with time	Task deadline does not vary with time	Deadline vary with time	Laxity time is the time difference b/w deadline, ready time &run time	Time period vary with completion of the task	time slice is varied for each individual job based on the weight
Type of scheduling	Static scheduling	Static scheduling	Dynamic scheduling	Dynamic scheduling	Queue based scheduling	Queue based scheduling
Task Response time	shorter	shorter	longer	Laxity time will be low	longer	Time slice will be longer
CPU utilization ratio	poor	poor	good	poor	good	good

Fig. 4: Comparison Table

ADVANTAGES

1. Non-preemptive EDF scheduling algorithms are easier to implement than their preemptive counter parts and can exhibit lower runtime overheads.
2. Non-preemptive scheduling naturally guarantees exclusive access to resources eliminating the need for complex resource access protocols.
3. Preemptive systems require individual task stacks whereas non-preemptive tasks can share a common stack, leading to vastly reduced memory requirements.

From above related work it can be seen that EDF algorithm have some limitations. EDF is not optimal for non-periodic and for multiprocessor scheduling. EDF behaves poorly under overload condition. EDF needs priority queue for storing deadlines. EDF is also less predictable. EDF have less control over the execution process. Implementation of EDF is relatively complicated [2]. Hence overcoming of limitations of EDF scheduling algorithm is a challenging problem.

In proposed work, we implemented EDF algorithm. EDF can be implemented by maintaining all ready tasks in sorted priority queue depending on its deadline. Whenever processor becomes free then by using EDF scheduling algorithm, according to priority queue shortest deadline task will be assigned to the processor. At the same time longer task waits in a ready queue. In that case, any new task arrives; its deadline will be checked with the deadline of currently executing task, if deadline of newly arrived task is closer to the current time, it will get chance for scheduling. And old task will be preempted and placed at the end of the queue [2].

4. RESULTS AND ANALYSIS

Task is schedulable under EDF, if and only if it satisfies the condition that the total processor utilization due to task is less than 1. As per the budget analysis time slot is varied with particular deadline, it will be assigned to the task with highest priority. Due to the number of computations, the processor utilization ratio will be more as much as possible, and the Simulation results are taken as follows.

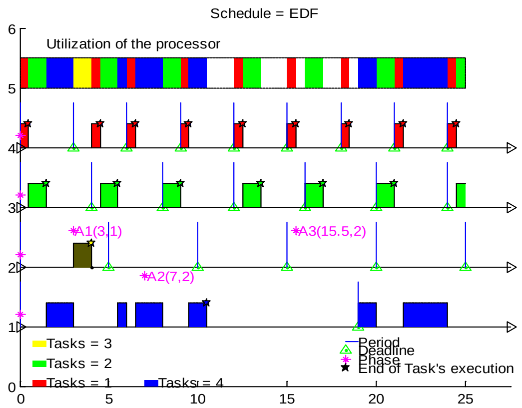

Fig. 5: Schedule analysis for EDF

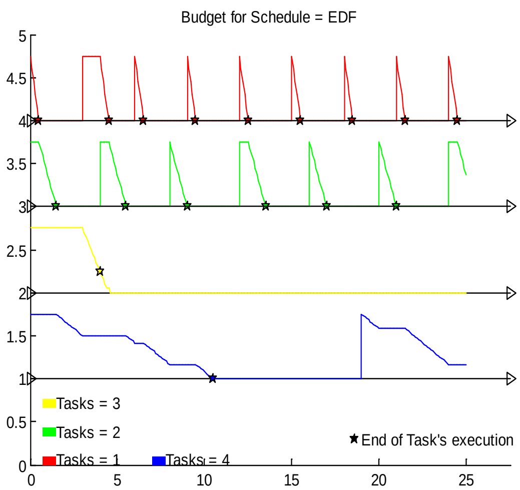

Fig. 6: Budget schedule for EDF

5. CONCLUSION AND DISCUSSIONS

Earliest Deadline First algorithms are presented the least complexity according to their performance many of the supposed 'problems' that have been attributed to this type of scheduling technique. It is clear that earliest deadline first is the efficient scheduling algorithm if CPU utilization is not more than 100% but does scale well when the system is overloaded. In the experimental environment EDF scheduling algorithm can meet the needs of real-time applications.

- Real-time scheduling techniques have some problems according to their task priority bases and over load conditions.
- When compare to rate monotonic and dead line EDF finds the solutions to the problem of task inconsistence.
- Different scheduling procedures compare to each other EDF perform only on single processor. It doesn't perform on multiprocessor platform in dynamic scheduling this is the major problem.

6. ACKNOWLEDGMENTS

Author would also like to thank Annamacharya institute of technology and sciences, Rajampet, India. for providing support and facilities to carry out this work.

REFERENCES

[1] Jag beer Singh, Satyendra Prasad Singh "An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 2, No.2, ISSN: 2156-5570, pp 31-35 February 2011

[2] Jashweeni Nandanwar "An Adaptive Real Time Task Scheduler", International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 6, No 1, pp 335-339 November 2012 ISSN (Online): 1694-0814

[3] Rina V. Bhuyar, D. G. Harkut, "Adaptive Neuro Fuzzy Scheduler for Real Time Task", International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 2, pp 393-396 February 2014 ISSN: 2277 128XI.

[4] D. G. Harkut, Anuj M. Agrawal, "Comparison of Different Task Scheduling Algorithms in RTOS", International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, pp 1236-1239, Issue 7, July 2014.

[5] Michael Short "The Case For Non-Preemptive, Deadline-driven Scheduling in Real-time Embedded Systems", Proceedings of the World Congress on Engineering 2010 Vol I WCE 2010, pp June 30 -July 2, 2010, London, U.K.

[6] Umm-I-aiman & Sher afzal khan, "review of different approaches for optimal performance of multi-processors", August, 2013 VFAST Transactions on 2013 Volume 1, Number 2, pp 7-11,July-August 2013.

[7] Sanjoy baruah, "Priority-driven Scheduling of periodic task systems on multiprocessors", Real-Time Systems, pp 188-201, Kluwer Academic Publishers, 2003.

[8] Hamza Gharsellaoui, "New Optimal Solutions for Real-Time Reconfigurable Periodic Asynchronous OS Tasks with Minimizations of Response Times", pp 1 -18.

[9] R. Kalpana, S. Keerthika "An Efficient Non-Preemptive Algorithm for Soft Real-Time Systems using Domain Cluster–Group EDF", International Journal of Computer Applications (0975–8887) Volume 93–No.20, pp 1-7 May 2014.

[10] Ishan Khera, Ajay Kakkar, "Comparative Study of Scheduling Algorithms for Real Time Environment", International journal of computer applications, Vol 44, No. 2 pp15-22, April 2012.

AUTHORS PROFILE

S. Eswaraiah Naidu born in pulivendula, A.P, India in 1992. He received B.Tech Degree in Electronics & Communication Engineering from JNT University, Anantapur, India. Presently he is pursuing M.Tech (Digital Electronics and Communication Systems) from Annamacharya Institute of Technology & Sciences, Rajampet, A.P., India. His research interests include embedded systems and image processing

B Abdul Rahim born in Guntakal, A.P, India in 1969. He received Bachelor of Engineering in Electronics & Communication Engineering from Gulbarga University in 1990. M.Tech (Digital Systems & Computer Electronics) from Jawaharlal Nehru Technological University in 2004. He is currently working towards Ph.D. degree from JNT University, Anantapur. Presently he is with Annamacharya Institute of technology & sciences, Rajampet as Head, Department of Electronics & communication engineering. He has published papers in international journals and conferences. He is a member of professional bodies like IEEE, EIE, ISTE, IACSIT, IAENG etc. His research interests include Fault Tolerant Systems, Embedded Systems and parallel processing. He recieved "Best Teacher Award" for his services by Lions Club, Rajampet India.

Fahimuddin Shaik did his B.Tech in Electronics & Communication Engineering and M.Tech (DECS) from JNT University, Hyderabad, India. He is currently working Towards a PhD in biomedical image processing at Rayalaseema University, Kurnool, and India. He is an assistant professor in the Department of ECE at the Annamacharya Institute of Technology & Sciences (an Autonomous Institute), in Rajampet, Andhra Pradesh. His research interests include signal processing, time series analysis, and biomedical image processing. He has presented many research papers at national and international conferences. He has authored a book "MEDICAL IMAGING IN DIABETES, VOL 1-A Technical Approach", Cinnamonteal Publishing, December 2011.He is a member in no. of professional bodies like, ISTE, IEI, BMI, IACSIT etc.