UACEE International Journal of Advances in Computer Science and its Applications – IJCSIA
Volume 3 : Issue 2 [ISSN 2250 – 3765]
Publication Date : 05 June 2013

An Evolution of Static Priority Scheduling in Real Time System

Jaya Gupta[1] Saroj Hiranwal[2]
Sri Balaji College of Engineering & Technology
Affiliated by Rajasthan Technical University
Benad Road, Near Dadi ka Phatak,
Jaipur-302012, India
Jaya.friend@gmail.com[1] hod.it@sbss.in[2]

***Abstract*- This paper reviews scheduling algorithm available in RTOS to schedule the tasks during execution in the Real Time System considering the static priority and evolution in the static priority scheduling in last decade. Real Time Operating System (RTOS) success depends on completion of task in given deadline. Real Time Operating System (RTOS) used, where we require specific application that meet there deadline with keeping logically correct results as an important constraint. Scheduling can be done in static or dynamic manner. According to priority, scheduling can also be classified as static priority and dynamic priority in dynamic scheduling. This paper concentrates on the static priority scheduling algorithm and the changes made or suggested by researchers and the proposed way to overcome from this problem.**

Keywords- RTOS, Scheduling Algorithm, Deadline, Dynamic, Static Priority.

I. INTRODUCTION

Task scheduling refers to identifying the order in which tasks should execute in a system. Embedded system design involves how to assign tasks to hardware and software components. The process of assigning a task, their execution order is known as scheduling. Since most of the tasks in a real-time embedded system are periodic in nature, the real-time task scheduling algorithms mostly concentrate on periodic tasks.[18]
Scheduling algorithm can be classified in following categories:

- Clock Driven Scheduling
- Event Driven Scheduling

A clock driven scheduling, the scheduling points are the interrupts received from a periodic clock. A basic problem with clock driven scheduling strategies is their inability to handle a large number of tasks.

In event driven scheduling, scheduler responds to different events in the system. The scheduling points are the events like arrival and completion of tasks. Scheduling also can be done on the basis of Priority. Priority is assigned to the tasks which may be static or dynamic, on the basis of priority we decide the sequence of task to be executed. In static priority we have two main algorithms: 1) Rate Monotonic (RM) and 2) Deadline Monotonic. The most successful algorithm in static priority is Rate Monotonic. In enduring paper we will discuss about the Rate Monotonic and the changes suggested by the researchers in the standard algorithm suggested by Liu and Layland. Section 2 of this paper describes overview of scheduling algorithms available in static priority with their respective disadvantages. Survey of related literature is given in Section 3. Proposed Algorithm is given in Section 4, Conclusion and future work is given in Section 5.

II.SCHEDULINGALGORITHM'S-OVERVIEW

2.1 Rate Monotonic Algorithm

Rate-monotonic scheduling is a scheduling algorithm in which static-priority is used for scheduling.

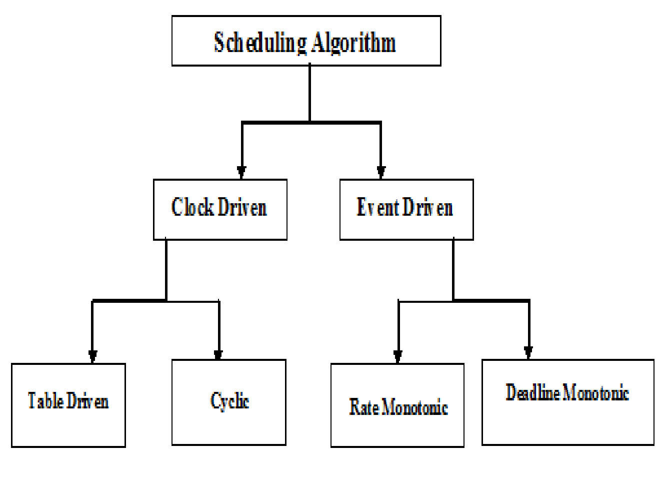

Fig 1: Classification of Scheduling algorithm.

The static priorities are assigned according to the period of the job, the shortest period job get the highest job priority and vice-versa.

These operating systems are generally preemptive and have deterministic guarantees with regard to response times. [1, 3]

A simple version of rate-monotonic analysis assumes that threads have the following properties:

1. All processes run on a single CPU i.e. no parallel task execution.
2. Deterministic deadlines are exactly equal to periods
3. Static priorities (the task with the highest static priority that is run able immediately preempts all other tasks)
4. Tasks are independent from each other.
5. Context switch times have no impact on the performance.

Liu & Layland (1973) proved that for a set of *n* periodic tasks with unique periods, a feasible schedule that will always meet deadlines exists if the CPU utilization is below a specific bound (depending on the number of tasks). [21]

The schedulability test for RMS is:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$$

Where C_i is the computation time, T_i is the release period (with deadline one period later), and n is the number of processes to be scheduled.[15,18,20]

Drawbacks:

- It is difficult to handle aperiodic and sporadic tasks.
- It is not optimal if deadline and period of the task are differing.

2.2 Deadline Monotonic

Deadline-monotonic scheduling is a scheduling algorithm in which static-priority is used for scheduling. The static priorities are assigned according to the deadline of the job, the shortest deadline job get the highest job priority and vice-versa.

This priority policy is optimal for a set of periodic or sporadic tasks with following conditions:

1. All tasks deadlines should less than or equal to their minimum inter-arrival times (or periods).
2. All tasks have worst-case execution times (WCET) that are less than or equal to their deadlines.
3. All tasks are independent and they do not affect the working of other by blocking.
4. No task voluntarily suspends itself.
5. During execution there could be an instant occur when all the task goes ready to execute that instant is known as critical instant.
6. Scheduling overheads (switching from one task to another) are zero.
7. All tasks have zero release jitter (the time from the task arriving to it becoming ready to execute).[16]

Drawback:

- It is less predictable as task depends upon the deadlines.
- It is less controllable.

III. RELATED WORK DONE

Since 1973, when Liu and Layland proved that for a set of *n* periodic tasks with unique periods, a feasible schedule that will always meet deadlines exists if the CPU utilization is below a specific bound. In field of static scheduling algorithm designing lot of research have been done like some worked on finding the optimal scheduling algorithm, do the comparison of scheduling algorithm, try to create algorithm better for all type of task or suggest the changes in existing algorithm for improvement. Maximum research done in static priority scheduling is related to Rate Monotonic, as it is the optimal algorithm available in this scheduling technique. Few of them are concluded in the given below table with their respective years:

Year	Proposed Algorithm/ Work Title	Proposer/Author	Conclusion
1993	Modified Rate-Monotonic Algorithm for Scheduling Periodic Jobs with Deferred Deadlines	Wei Kuan Shih, Jane W. S. Liu and C. L. Liu	Author proposed semi-static priority-driven algorithm; each job is assigned two fixed priorities, the higher priority for its previous job request and the lower priority for recent job request. When the amounts of deference are sufficiently large, jobs can be feasibly scheduled using this algorithm with condition that their total utilization is equal to or less than one. [2]
1995	A Reservation-Based Algorithm for Scheduling Both Periodic and Aperiodic Real-Time Tasks	Kang G. Shin and Yi-Chieh Chang	Proposed a new algorithm to schedule both periodic and aperiodic real-time tasks. Periodic tasks are scheduled according to the RMPA. Aperiodic tasks are scheduled by using the reserved and unused CPU time in each unit cycle.[5]
2002	A Modified Version of Rate-Monotonic Scheduling Algorithm and its Efficiency Assessment	Mahmoud Naghibzadeh	Author identifies special case of systems execution using RMA, for this special case it was shown that least upper bound to load factor is improved. Author named delayed rate-monotonic algorithm. There are few set of task that can run safely with DRM and behave unsafe with RM.[6]
2002	The Space of Rate Monotonic Schedulability	Enrico Bini, Giorgio C. Buttazzo	Approach to analyze the schedulability of periodic tasks under the Rate Monotonic priority approach, such an approach allowed us to accurately decide the feasibility region in the space of task computation times and able to derive a tunable guarantee test. Tunability property is important in special cases where performance of a polynomial time test is not adequate for achieving high processor utilization, and the overhead introduced by exact tests is too high for an on-line admission control. [10]
2006	Rate Monotonic Schedulability Conditions Using Relative Period Ratios	Wan-Chen Lu, Hsin-Wen Wei, Kwei-Jay Lin	This paper presents a special case of RM schedulability bound as a function of *z*1 and *z*2, the ratios of the smallest and the largest virtual periods to the largest period. We can reduce the difference between the smallest and largest job period and a periodic task system can be achieve a higher Rate-Montonic

			schedulability.[11]
2008	Response Time Analysis of Asynchronous Periodic and Sporadic Tasks Scheduled by a Fixed-Priority Preemptive Algorithm	Manuel Coutinho, Jos´e Rufino and Carlos Almeida	This paper propose an algorithm to analyze the schedulability of asynchronous periodic and sporadic tasks scheduled by a fixed-priority preemptive algorithm.[7]
2008	Global Rate-Monotonic Scheduling with Priority Promotion	Shinpei Kato, Akira Takeda and Nobuyuki Yamasaki	Rate-Monotonic until Zero Laxity (RMZL) algorithm, use the laxity-driven priority promotion strategy as compare with the global Rate-Monotonic algorithm. According to author, the RMZL algorithm is capable to execute more tasks as compare with Rate-Monotonic and RMUS algorithms by schedulability test.[12]
2009	A Modified Rate-Monotonic Algorithm for Scheduling Periodic Tasks with Different Importance in Embedded System	Jiwen Dong , Yang Zhang	Author proposes a new scheduling algorithm based on RM. It improves the scheduling of long period tasks which are important by adding two parameters, importance and laxity.[8]
2010	A New Scheduling Algorithm for Real Time System	Yaashuwanth .C and Dr.R. Ramesh	Author purpose a algorithm which eliminates the drawbacks of considering a single component which could be period or deadline to assign the priority. This algorithm can be implemented in real time system with condition where period and deadline doesn't vary with time.[17]

Table 1: Summary of different proposed algorithms for task scheduling in real time system

3.1 Limitations:

In above created table we read about many algorithms and suggestions in existing algorithm patterns. In "Modified Rate-Monotonic Algorithm for Scheduling Periodic Jobs with Deferred Deadlines", Author proposed semi-static priority-driven algorithm; each job is assigned two fixed priorities, the higher priority for its previous job request and the lower priority for recent job request. When the amounts of deference are sufficiently large, jobs can be feasibly scheduled using this algorithm with condition that their total utilization is equal to or less than one. Limitation of this algorithm is, in a special case modified rate-monotonic algorithm is not optimal for this case when the period ratio is larger than two.[2]

In "A Reservation-Based Algorithm for Scheduling Both Periodic and Aperiodic Real-Time Tasks" author conclude that however there approach is good but not best as there some issues could arise like: if the task switching time is not negligible, this overhead in the RB algorithm may affect the derivation. A good scheduling algorithm (EDF, MUF) may improve the probability of guaranteeing aperiodic tasks. [5, 13]

In "Modified Version of Rate-Monotonic Scheduling Algorithm and its Efficiency Assessment" Author identifies special case of systems execution using RMA, for this special case it was shown that least upper bound to load factor is improved. Author named delayed rate-monotonic algorithm. There are few set of task that can run safely with DRM and behave unsafe with RM. Due to importance of safety verification of real-time systems prior to their being operational there is still need to make it more efficient.[6]

In "Space of Rate Monotonic Schedulability" Approach to analyze the schedulability of periodic tasks under the Rate Monotonic priority approach, such an approach allowed us to accurately decide the feasibility region in the space of task computation times and able to derive a tunable guarantee test. Tunability property is important in special cases where performance of a polynomial time test is not adequate for achieving high processor utilization, and the overhead introduced by exact tests is too high for an on-line admission control. In the proposed algorithm we can introduce different cases like when task computation times are considered as random variables with known probability distribution. [10]

In "Rate Monotonic Schedulability Conditions Using Relative Period Ratios", This paper presents a special case of RM schedulability bound as a function of $z1$ and $z2$, the ratios of the smallest and the largest virtual periods to the largest period. We can reduce the difference between the smallest and largest job period and a periodic task system can be achieve a higher Rate-Monotonic schedulability. Limitation of this approach is that this provide good result in uniprocessor but in multiprocessor environment it still need to be execute.[11]

In "Response Time Analysis of Asynchronous Periodic and Sporadic Tasks Scheduled by a Fixed-Priority Preemptive Algorithm" [7]. This algorithm can be used to study the schedulability of asynchronous periodic and sporadic tasks scheduled by a fixed-priority preemptive algorithm. It can be used to:

• Diminish the pessimism induced by the critical instant
• Allow offset relationships
• Determine the worst response time of each job
– analyze response jitter
– account for missed deadlines
• Calculate the *asynchronous critical instant*
– integrate sporadic tasks
– increase analysis speed

The calculation of the response times of all jobs under a particular condition is necessary. Therefore, the proposed methods do not scale well against the number of periodic tasks, especially task sets using co-prime periods.

In "Modified Rate-Monotonic Algorithm for Scheduling Periodic Tasks with Different Importance in Embedded System", Author proposes a new scheduling algorithm based on RM. It improves the scheduling of long period tasks which are important by adding two parameters, importance and laxity. Author suggests if importance is higher when task laxity is zero could preempt the running task. Authors also suggest that the algorithm could be decrease the deadline-missing ratio of the tasks and the CPU resource could be used more efficiently. Limitation of this algorithm is it can use only in limited area where priority does not depends on deadline or on period.[8, 15]

In "Global Rate-Monotonic Scheduling with Priority Promotion" In this paper, author presents the Rate-Monotonic until Zero Laxity (RMZL) algorithm, which applies the laxity-driven priority promotion strategy to the global Rate-Monotonic algorithm. Authors execute the schedulability test and the tardiness bound in the algorithm. According to author RMZL algorithm is able to execute more tasks than the Rate-Monotonic and the RMUS. As per paper the RMZL algorithm is competitive with and is even better for a larger number of CPU cores than the EDZL algorithm. Limitation of this paper is that they present schedulability test is pessimistic, while it outperforms the existing fixed-priority algorithms. There is one more issue is there how to detect the zero-laxity condition. [12, 14]

In last algorithm "A New Scheduling Algorithm for Real Time System" author purpose a algorithm which eliminates the drawbacks of considering a single component which could be period or deadline to assign the priority. This algorithm can be implemented in real time system with condition where period and deadline doesn't vary with time. In rate monotonic algorithm all available task consider as equal important and priority decide by the period of the task, due to which some unimportant task could be scheduled before an important task. Deadline monotonic algorithm where all the tasks consider as equal important and they scheduled on the basis of deadlines same problem could occur in this algorithm also. Authors propose architecture to overcome with this drawback by combining the advantages of period and deadline. Percentage of weightage is provided to all the parameter and based on their weight, a priority component calculated on the basis of which we schedule the task. Limitation of this proposed algorithm is that, it can be implemented in real time system with condition where period and deadline doesn't vary with time. [17,19]

IV.PROPOSED ALGORITHM- ARCHITETURE

After doing the literature survey, we are now in condition to purpose an architecture using which we can calculate a selective component. Selective component will take 30% weightage from period, 20% from deadline and 50% weightage will be taken from user defined priority. On the basis of this selective component now task scheduling we be performed, in case any two or more task calculate the same selective component then period will be check and short period task will execute first.

For eg. Let's consider a task whose period is 45, deadline is 55 and the user defined priority is 4. Now the selective component will be calculated 30% of 45 + 20% of 55 + 50% of 4= 26. Now the sorting of task will be done on the basis of this selective component.

V. CONCLUSION & FUTURE WORK

This paper presents a categorized summary of work done in the area of static priority based task scheduling in RTOS, types of scheduling algorithm like clock driven & event driven and there subclass algorithms for e.g. cyclic, table driven, rate monotonic, deadline monotonic and their advantages and disadvantages . My review concludes that there is no single algorithm available which optimal for any type of tasks, every algorithm has some finite limitation.

The major challenge is, no algorithm has been suggested such as to use one or more than one, important parameters to make an equation such that the created algorithm could select a set of task and do there scheduling as no task miss the deadline or make the throughput time delay minimum. In last section we proposed an algorithm architecture using which we can reduce the deadline missing time.

REFERENCES:

[1] John Lehoczky , Lui Sha and Ye Ding "The Rate Monotonic Scheduling Algorithm: Exact Characterization And Average Case Behavior" CH2803-5/89/0000/0166,1989 IEEE.

[2] Wei Kuan Shih, Jane W. S. Liu and *C.* L. Liu "Modified Rate-Monotonic Algorithm for Scheduling Periodic Jobs with Deferred Deadlines" IEEE transactions on sofiware engineering, vol. 19, no. 12, december 1993

[3] Krithi Ramamritham and John A. Stankovic " Scheduling Algorithms and Operating Systems Support for Real-Time Systems" proceedings of the IEEE, vol. *82.* no. I , january 1994.

[4] John A. Stankovic, Marco Spuri hlarco Di Natale and Giorgio C Buttazzo" Implications of Classical Scheduling Results for Real-Time Systems". 0018-9162/95,1995 IEEE.

[5] Kang G. Shin and Yi-Chieh Chang " A Reservation-Based Algorithm for Scheduling Both Periodic and Aperiodic Real-Time Tasks". IEEE Transactions on Computers, vol. 44, no. 12, december 1995.

[6] Mahmoud Naghibzadeh " A Modified Version of Rate-Monotonic Scheduling Algorithm and its Efficiency Assessment" 1530-1443/02, 2002 IEEE.

[7] Manuel Coutinho, Jos´e Rufino and Carlos Almeida" Response Time Analysis of Asynchronous Periodic and Sporadic Tasks Scheduled by a Fixed-Priority Preemptive Algorithm" 1068-3070/08, 2008 IEEE.

[8] Jiwen Dong ,Yang Zhang "A Modified Rate-Monotonic Algorithm for Scheduling Periodic Tasks with Different Importance in Embedded System" ICEMI'2009.

[9] Li Jie, Guo Ruifengand Shao Zhixiang "The Research of Scheduling Algorithms in Real-time System" 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering.

[10] Enrico Bini, Giorgio C. Buttazzo "The Space of Rate Monotonic Schedulability" Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS'02), 1052-8725/02 2002 IEEE

[11] Wan-Chen Lu, Hsin-Wen Wei, Kwei-Jay Lin "Rate Monotonic Schedulability Conditions Using Relative Period Ratios" Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06) 2006

[12] Shinpei Kato, Akira Takeda and Nobuyuki Yamasaki" Global Rate-Monotonic Scheduling with Priority Promotion" IPSJ Transactions on Advanced Computing System (ACS), Vol. 2, No. 1, pp. 64–74, March 2008.

[13] Nasro Min-Allah · Samee Ullah Khan · Nasir Ghani · Juan Li · LizheWang ·"A comparative study of rate monotonic schedulability tests" Springer Science+Business Media, LLC 2011.

[14] M.Kaladevi and Dr.S.Sathiyabama," A Comparative Study of Scheduling Algorithms for Real Time Task" International Journal of Advances in Science and Technology, Vol. 1, No. 4, 2010.

[15] Liu, C. L.; Layland, J. (1973), "Scheduling algorithms for multiprogramming in a hard real-time environment", *Journal of the ACM* 20 (1): 46–61.

[16] Alan Burns and Andy Wellings (2001). Real-Time Systems and Programming Languages (3rd Edition). Addison Wesley.

[17] Yaashuwanth .C and Dr.R. Ramesh" A New Scheduling Algorithm for Real Time System" International Journal of Computer and Electrical Engineering, Vol.2, No.6, December, 2010 1793-8163

[18] Santanu Chattopadhyay , Embedded System Design PHI, ch-5, pg- 53-79.

[19] Enrico Bini and Giorgio C. Buttazzo, "Schedulability Analysis of Periodic Fixed Priority Systems" in IEEE Transactions on Computers,Vol. 53, No. 11, pp. 1462-1473, Nov. 2004.

[20] Lui Sha, Ragunathan Rajkumar and Shirish s. sathaye "Generalized Rate-Monotonic Scheduling Theory: A Framework for Developing Real-Time System" , Proceeding of the IEEE, Vol 82, No.1, January 1994.

[21] Alan A. Bertossi , Andrea Fusiello "Rate-Monotonic scheduling for hard-real-time systems" 1997 Elsevier Science.