



A Survey of Scheduling Algorithms in Real Time Systems

Wamika Basu*, Subho Chaudhuri

Department of Computer Science, Birla Institute of Technology, Mesra,
Kolkata Campus, India

Abstract— A system is said to be a real time system if it produces logically correct results within a specified time called deadline. Real time systems may be Hard or Soft. A hard real time system must always meet the deadline to avoid a catastrophe. For example, if an antimissile system is not able to detect an incoming missile within few seconds, it may lead to loss of human life. If a soft real time system does not meet the deadline, it merely results in poor performance of the system. For example, if during online video streaming, the audio data packets arrive late and are not in synchronization with the video, one will simply have a bad viewing experience. Scheduling in real time systems determines the order in which real time tasks should be executed such that they meet their deadlines. This paper reviews the work done by various researchers in the field of real time scheduling. This paper also presents an analytical study of a few real time scheduling algorithms.

Keywords— Real time system, Deadline, Hard, Soft, Scheduling

I. INTRODUCTION

The proper functioning of a real time system depends on both the timeliness and correctness of the outputs [1], [2] defines the terms real and time separately. According to [2], the word real in real time implies that the system must respond to external events as an when they crop up. The word time implies that the effectiveness of the system depends on the time at which the results are produced in addition to being correct. [3] states that a real time system must respond to external inputs within a time limit to avoid failure. The application of real time systems can be found in Robotics, Pacemakers, Chemical Plants, Antimissile Systems, and Embedded Systems etc. to name a few [4].

II. THE REAL TIME TASK MODEL

In general, a task is said to be a program in execution [5]. A real time task has the following parameters [2,4]-

- *Release time* - This is the time at which a task is ready to be executed. It is also known as arrival time.
- *Start time* - The time at which the task begins to execute is known as start time.
- *Finish time* - The time at which the task completes its execution is called finish time.
- *Worst-case Computation time* - This is the time taken by the processor to complete the execution of a task without any intervention.
- *Absolute deadline* - It is the instant of time within which a task must finish its execution.
- *Relative deadline* - It is the time interval between release time and absolute deadline.

III. TYPES OF REAL TIME TASKS

There are three types of real time tasks-

- *Periodic* - A periodic task occurs after fixed intervals of time. Most real time tasks are periodic.
- *Sporadic* - A sporadic task occurs at random instants of time.
- *Aperiodic* - Aperiodic tasks occur at random instants of time with a minimum inter-arrival time between any two consecutive instances of the task.

IV. SCHEDULING

In real time systems, process scheduling refers to the order of execution of real time tasks so that they do not miss their deadlines. During the early 1970's, several real time scheduling algorithms were extensively studied. This paper analyzes a few widely used existing real time scheduling algorithms.

A few important terms related to real time scheduling are [6]-

- *Utilization (Processor utilization)* - It refers to the fraction of time spent by the CPU in executing a set of tasks.
- *Schedulability Test* - It is a test to determine whether the tasks can meet their deadlines.
- *Optimal Scheduler* - If a set of tasks cannot be scheduled by an optimal scheduler, then it cannot be feasibly scheduled by any other scheduler. Tasks are said to be feasibly scheduled if all of them meet their deadlines.

V. HARD REAL TIME SCHEDULING ALGORITHMS

Hard real time scheduling algorithms can be of two types [2]-

- *Fixed Priority* - In fixed priority scheduling algorithms, the tasks are assigned certain fixed parameters and scheduling is done on the basis of that parameter. Rate Monotonic (RM) and Deadline Monotonic (DM) are fixed priority algorithms.
- *Dynamic Priority* - In dynamic priority scheduling algorithms, scheduling is done on the basis of dynamic parameters that change with time as the system evolves. Earliest Deadline First (EDF) is a dynamic priority algorithm.

A. Fixed Priority Scheduling Algorithms- RM and DM

1) *Rate Monotonic (RM)* [5, 7, 8, 9, 10,11]: The Rate Monotonic algorithm is an optimal static priority algorithm with preemption. It schedules periodic tasks with deadline equal to period. Each task is assigned a fixed priority inversely proportional to period. This implies that a task with a shorter period has a higher priority. A set of n tasks will always meet their deadlines when scheduled according to the rate monotonic approach if,

$$\text{Utilization (U)} = \sum_{i=1}^n C_i / P_i \leq n(2^{1/n} - 1) \quad \text{where } C_i = \text{worst-case computation time, } P_i = \text{period}$$

2) *Deadline Monotonic (DM)* [12,13]: The Deadline Monotonic algorithm is an optimal static priority algorithm with preemption. It schedules periodic tasks with deadline less than or equal to period. Each task is assigned a fixed priority inversely proportional to relative deadline. This implies that a task with a shorter relative deadline has a higher priority. The rate monotonic algorithm is a special case of the deadline monotonic approach when deadline is equal to the period.

B. Dynamic Priority Scheduling Algorithm- EDF

1) *Earliest Deadline First (EDF)* [5,8]: The Earliest Deadline First algorithm is an optimal dynamic priority algorithm with preemption. It schedules periodic tasks with deadline equal to period. Each task is assigned a fixed priority inversely proportional to absolute deadline. This implies that a task with a shorter absolute deadline has a higher priority. A set of n tasks will always meet their deadlines when scheduled according to the earliest deadline first approach if,

$$\text{Utilization (U)} = \sum_{i=1}^n C_i / P_i \leq 1 \quad \text{where } C_i = \text{worst-case computation time, } P_i = \text{period}$$

VI. LITERATURE SURVEY

The following is a comparative survey of hard real time scheduling algorithms with respect to –

A. Advantages and Disadvantages (TABLE I)

B. Various parameters like priority assignment, scheduling criterion, constraint, CPU utilization, optimality and type of task scheduled (TABLE II)

TABLE I ADVANTAGES AND DISADVANTAGES OF HARD REAL TIME SCHEDULING ALGORITHMS (SURVEYED)

Algorithm	Advantages	Disadvantages
Rate Monotonic (RM)	<ul style="list-style-type: none"> • Simple to implement • Commonly used algorithm 	<ul style="list-style-type: none"> • Difficult to schedule aperiodic and sporadic tasks • Not optimal when deadline ≠ period • When DM fails, RM will always fail
Deadline Monotonic (DM)	<ul style="list-style-type: none"> • When RM fails, DM will produce a feasible solution • Can schedule both periodic and sporadic tasks 	<ul style="list-style-type: none"> • Does not guarantee full processor utilization
Earliest Deadline First (EDF)	<ul style="list-style-type: none"> • Less context switches • Full processor utilization 	<ul style="list-style-type: none"> • During overload condition cannot predict which tasks will miss their deadlines

TABLE III HARD REAL TIME SCHEDULING ALGORITHMS SURVEY

Reference	Algorithm	Priority Assignment	Scheduling Criterion	Constraint	CPU Utilization	Optimality	Type of task scheduled
[12,13]	Deadline Monotonic	Fixed	Relative deadline	Deadline less than equal to period	More than Rate Monotonic	Optimal static priority algorithm when deadline less than or equal to period	Periodic & Sporadic
[10,11]	Rate Monotonic	Fixed	Task Period	Deadline equal to period	Less	Optimal static priority algorithm when deadline equal to period	Periodic
[8,14]	Earliest Deadline First	Dynamic	Absolute Deadline	Deadline equal to period	High	Optimal dynamic priority algorithm when deadline equal to period	Periodic

VII. CONCLUSIONS

The study and survey of some existing hard real time scheduling algorithms has been done in this paper. It has been observed that deadline is the most important concept in real time systems and a yardstick to measure the performance of real time scheduling algorithms. A few important hard real time scheduling algorithms have been studied and a comparison between them has been presented.

Based on the survey, it can be concluded that *dynamic priority scheduling algorithms* result in 100 percent CPU utilization. *Fixed priority scheduling algorithms* are easier to implement and hence widely used.

Rate Monotonic is an optimal static priority algorithm whereas *Earliest Deadline First* is an optimal dynamic priority algorithm. *Deadline Monotonic* is an optimal static priority algorithm when deadline is less than or equal to period.

REFERENCES

- [1] P. A. Laplante, *Real-Time Systems Design and Analysis*, 3rd ed., IEEE Press, 2004.
- [2] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 3rd ed., Springer, 2005.
- [3] R. L. Panigrahi and M .K. Senapaty, "Real Time System for Software Engineering: An Overview", *Global Journal for Research Analysis*, Vol. 3, Issue 1, pp. 25-27, January 2014.
- [4] J. W. S. Liu, *Real-Time Systems*, Prentice Hall, 2000.
- [5] A. Silberschatz, P. B. Galvin and G. Gagne, *Operating System Concepts*, 8th ed., Wiley, 2012.
- [6] R. Mall, *Real-Time Systems: Theory and Practice*, Pearson, 2008.
- [7] I. Khera and A. Kakkar, "Comparative Study of Scheduling Algorithms for Real Time Environment", *International Journal of Computer Applications*, Vol. 44, Issue 2, pp. 5-8, April 2012.
- [8] D. G. Harkut and A. M. Agrawal, "Comparison of Different Task Scheduling Algorithms in RTOS: A Survey", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, Issue 7, pp. 1236-1240, July 2014.
- [9] V. Singh, K. Vidhani, P. Ganeshpurkar and S. Khatri, "Analytical Study on Scheduling Algorithms for Real Time Static System", *Proceedings of the International Conference on Recent Advances in Communication, VLSI & Embedded Systems*, pp. 179-182, December 2014.
- [10] J. Lehoczky, L. Sha and Y. Ding, "The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behaviour", *Proceedings of the 10th IEEE Real-Time Systems Symposium*, pp. 166-171, December 1989.
- [11] A. Burns, N.C. Audsley, R.I. Davis, K.W. Tindell and A.J. Wellings, "Fixed Priority Pre-emptive Scheduling: An Historical Perspective", *Real-Time Systems*, Vol. 8, Issue 2, pp. 173-198, March 1995.
- [12] N. C. Audsley, A. Burns, M. F. Richardson, and A. J. Wellings, "Hard Real-Time Scheduling: The Deadline Monotonic Approach", *Proceedings of the 8th IEEE Workshop on Real-Time Operating Systems and Software*, pp. 133-137, May 1991.
- [13] N.C. Audsley, "Deadline Monotonic Scheduling", Technical Report, Department of Computer Science, University of York, September 1990.
- [14] C. L. Liu, and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment," *Journal of the Association for Computing Machinery*, Vol. 20, Issue 1, pp. 44-61, January 1973.