International Journal of Engineering & Technology, 7 (2.7) (2018) 618-621

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Comparison of Scheduling Algorithms in The Design of Fault Tolerant Real Time Systems

Mary Swarna Latha Gade [1], K Sreenivasa Ravi [2]

[1] *Research Scholar, KL Education Foundation*
[2] *Professor of ECSC, KL Education Foundation*

Abstract

Most of the real time systems have the timing constraints. The main important timing constraints of any real time systems are to meet the deadlines of its application tasks. Not only satisfying the timing constraints of any real-time system, but also the functional correctness of application needs to be guaranteed. Meeting the dead line of application is no use if it deviates from its precise output. Timing constraints of the system can be satisfied by choosing proper task scheduling algorithms and the reliability of the system can be reached by providing fault-tolerance. In this paper, various fault scheduling algorithms like fixed priority, EDF(Earliest dead line first), LLF (Least Laxity First),Rate monotonic etc have been studied and compare the parameters like Worst-case execution times, response time, task missed deadlines, Number of preemption, number of context switches, Deadlock and processor utilization factor.

Keywords: *Dead line, real time system, scheduling algorithm, reliability ,fault tolerance.*

1. Introduction

A real time embedded system is a system which has to respond within some specified time period. Real time systems are classified as hard real time and soft real time systems. In the hard real time systems, the tasks are required to complete within the specified time. The real time task must be completed within the time limits and correctly or else the system can be considered as a total failure. Depending upon the particular application, the system failure could bring about disastrous harms and potentially death toll. In soft real time systems, the tasks are required to complete within the specified time only but not strict deadlines. Scheduling of tasks is a procedure by which computer system chooses which task and at what time should be executed. The goal of scheduling of real time tasks is to ensure the deadlines of real time tasks in the given real time system. To accomplish this objective, a lot of research has been conducted on real-time task scheduling. The two types of real-time scheduling are Static and Dynamic. In static scheduling, priorities can remain fixed and assigned during configuration time. In dynamic scheduling, priorities can't remain fixed and assigned during execution time as shown in fig 1.

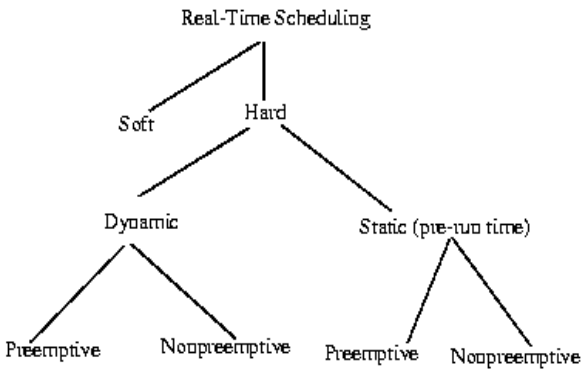


Fig 1. Classification of real time scheduling

The basic attributes of a real time task are release time ,start time, execution time, response time, completion time and deadline as shown in fig 2.

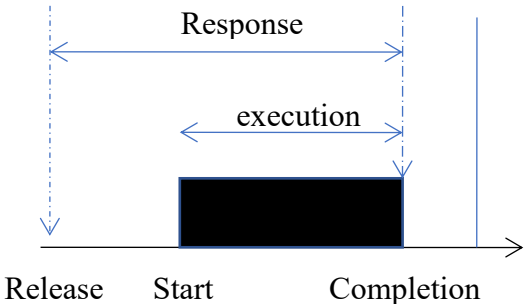


Fig 2. Basic attributes of a real time task

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Response time=completion time-release time
Execution time= completion time- start time

2. Scheduling Algorithms

2.1 Rate Monotonic

Rate monotonic scheduling algorithm is a static priority algorithm. It schedules periodic, independent and preemptible tasks on a single processor. It will assign the processor to a task according to its period. A task can be assigned with the highest priority if the task has the least period. Consider an example task set which consists of three tasks with execution time ,dead line and periods as shown in table. Since the task T1 has the least period, it is assigned as highest priority and the task T3 has longer period, it is assigned as the lest priority.
Consider a task set consists of 3 tasks T1,T2,T3.

Task	Start time	Execution time	Period	Dead line	Priority
1	0	2	6	6	1
2	0	2	9	9	2
3	0	3	12	12	3

2.2 Round Robin

Round robin is a scheduling algorithm in which each task is assigned a fixed time slice in a cyclic manner. It is very simple to implement compared to all the algorithms. But, the drawback of this algorithm is the number of context switching are more. Consider an example, a task set consists of five tasks A,B,C,D,E with service times 2,2,1,2,2 as shown in the following fig.

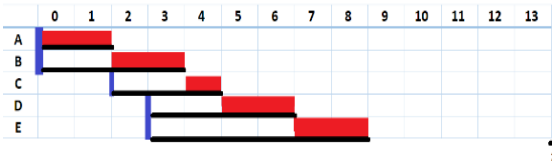


Fig 3: Scheduling of tasks

The scheduling with round robin technique is as shown in fig 4

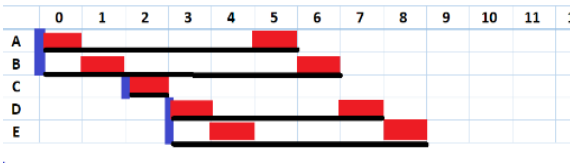


Fig 4: Scheduling of tasks with round robin

2.3 Least Laxity Protocol

Least laxity or least slack first is a scheduling algorithm which assigns priority depends upon the available slack time of a task. It is mostly used in embedded systems with multiple processors. Slack time is the time difference deadline, ready time and the remaining computation time.
S=D-T-C
S=slack time
D=dead line
T=ready time
C=remaining computation time

Consider an example as shown in fig.

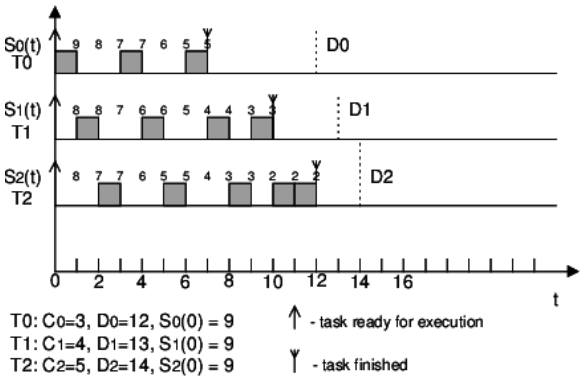


Fig 5. Scheduling with least slack time

2.4 Highest Priority First Algorithm

Highest priority first is a scheduling algorithm where the task with highest priority will be executed first. If two tasks are having the same priority, then the tasks are executed with first come first serve basis. Consider an example as shown in fig.

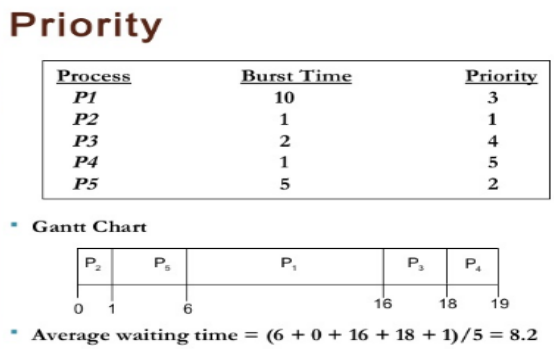


Fig 6. Scheduling with Highest priority first

2.5 EDF

Earliest dead line first is a scheduling algorithm where it chooses the task with earliest dead line first. This scheduling algorithm ignores the priority of the tasks. It can guarantee that the dead lines of all the tasks are met with better CPU utilization time. It is very difficult to implement this algorithm in hardware. Consider an example as shown in fig.

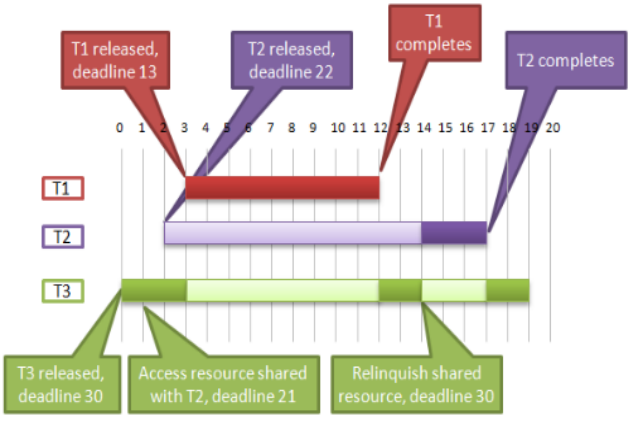


Fig. 7. scheduling with EDF

3. Comparison

The simulation results of different scheduling algorithms as shown in the following figures. The task response time, number of context switches, number of preemptions of different algorithms have been compared.

Rate monotonic

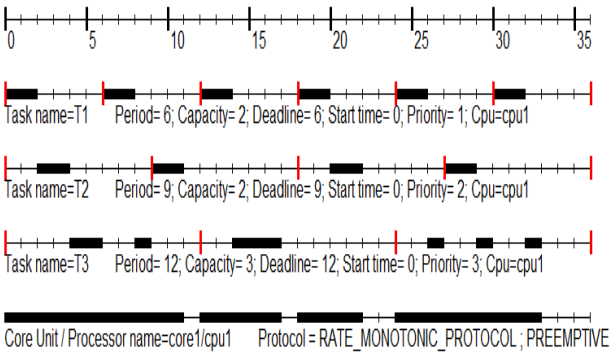


Fig 8. Scheduling using rate monotonic

Round robin protocol

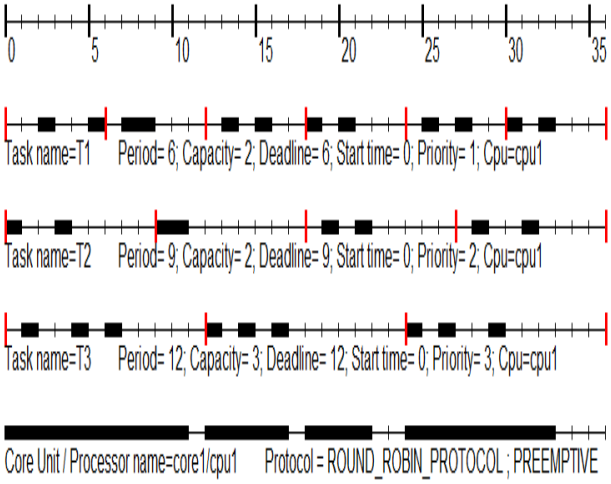


Fig 9. Scheduling using round robin

Least laxity protocol

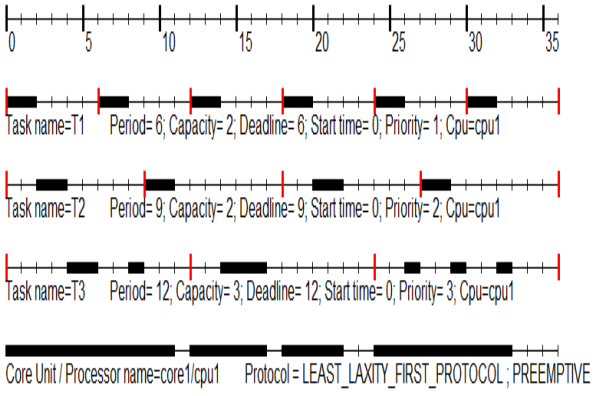


Fig 9. Scheduling using Least laxiry protocol

Highest priority first protocol

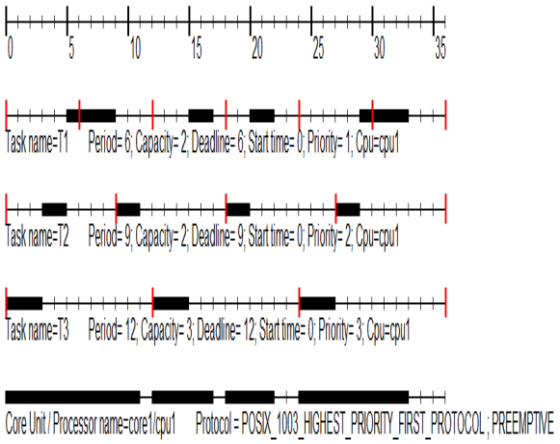


Fig 10. Scheduling using highest priority protocol

EDF

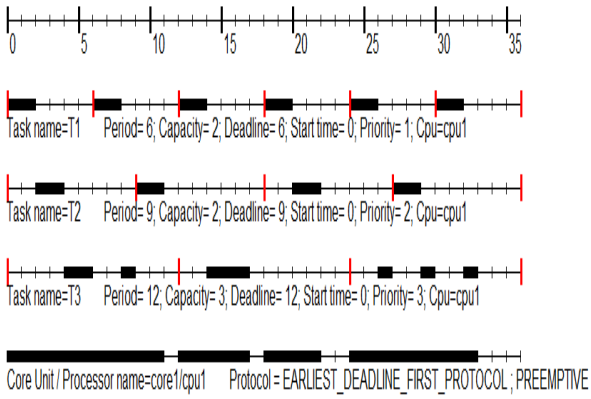


Fig 11. Scheduling using EDF

Scheduling	**context switches**	**preemptions**	**T1 response time**	**T2 response time**	**T3 response time**
Rate monotonic	15	3	2	4	9
Round robin	26	14	6	5	7
Least laxity protocol	15	3	2	4	9
POSIX Highest priority first protocol	10	0	7	5	3
EDF	**15**	**3**	**2**	**4**	**9**

4. Conclusion

The rightness of any real time system is most certainly not just relying upon the result of the completion time but also on the time when the result is occurred i.e. they have to meet their deadlines strictly. Scheduling in Real time systems is an important parameter to consider completing all the tasks within a given time. In this paper, We simulated the real time scheduling algorithms like rate monotonic, round robin, least laxity first, highest priority first protocol and Earliest dead line first and also compared their performance such as execution time, task response times, blocking time,

task deadlines, number of context switch, number of preemptions, utilization factor etc.

References

[1] Fengxiang Zhang and Alan Burns, September, 2009. Schedulability analysis for real-time systems with EDF scheduling. IEEE Transactions on computers, vol. 58, no. 9.

[2] Hamid Arabnejad and Jorge G. Barbosa, 2013. List scheduling algorithm for heterogeneous systems by an optimistic cost table. IEEE.

[3] J. Lehoczky, L. Sha, and Y. Ding, 1989 The rate monotonic scheduling algorithm: Exact characterization and average case behaviour. IEEE Real-Time Systems Symposium, 166-171.

[4] Peng Li and Binoy Ravindran September, 2004. Fast, Best-Effort Real-Time Scheduling Algorithms. IEEE Transactions on computers, vol. 53, no. 9.

[5] Luis Bu' rdalo, Andre´s Terrasa, Agustı´n Espinosa, and Ana Garcı´a-Fornes, December, 2012. Analyzing the Effect of Gain Time on Soft-Task Scheduling Policies in Real-Time Systems. IEEE Transactions on software engineering, vol. 38, no. 6.

[6] Zahereel Ishwar Abdul Khalib , Badlishah R. Ahmad and Ong Bi Lynn Ong, "High deadline meeting rate of nonpreemptive dynamic soft real timescheduling algorithm",296301,DOI:10.1109/ICCSCE.2012.648715, 2012 IEEE.

[7] Li, Q. & Ba, W, "A group priority earliest deadline first scheduling algorithm", Frontiers of Computer Science October 2012, Volume 6, Issue 5, pp 560–567.

[8] J.Leung and J. Whitehead, "On the complexity of fixedpriority schedulings of periodic, real-time tasks", Performance Evaluation 2(4):237-250 December 1982.

[9] 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.

[10] BALA DASTAGIRI, N. and HARI KISHORE, K., 2016. Analysis of low power low kickback noise dynamic comparators in pacemakers. Indian Journal of Science and Technology, 9(44),.

[11] BALA DASTAGIRI, N. and HARI KISHORE, K., 2016. Reduction of kickback noise in latched comparators for cardiac IMDs. Indian Journal of Science and Technology, 9(43),.

[12] HUSSAIN, S.N. and KISHORE, K.H., 2016. Computational Optimization of Placement and Routing using Genetic Algorithm. Indian Journal of Science and Technology, 9(47),.

[13] MUDAVATH, M. and HARIKISHORE, K., 2016. Design of CMOS RF front-end of low noise amplifier for LTE system applications. Asian Journal of Information Technology, 15(20), pp. 4040-4047.

[14] MURALI, A., KAKARLA, H.K. and VENKAT REDDY, D., 2016. Integrating FPGAs with trigger circuitry core system insertions for observability in debugging process. Journal of Engineering and Applied Sciences, 11(12), pp. 2643-2650.

[15] BALA GOPAL, P., HARI KISHORE, K., KALYANA VENKATESH, R.R. and HARINATH MANDALAPU, P., 2015. An FPGA implementation of onchip UART testing with BIST techniques. International Journal of Applied Engineering Research, 10(14), pp. 34047-34051.

[16] BHARADWAJ, M. and KISHORE, H., 2017. Enhanced launch-off-capture testing using BIST design. Journal of Engineering and Applied Sciences, 12(3), pp. 636-643.

[17] VUNDAVILLI, P.R., PARAPPAGOUDAR, M.B., KODALI, S.P. and BENGULURI, S., 2012. Fuzzy logic-based expert system for prediction of depth of cut in abrasive water jet machining process. Knowledge-Based Systems, 27, pp. 456-464.

[18] KILARU, S., HARIKISHORE, K., SRAVANI, T., ANVESH CHOWDARY, L. and BALAJI, T., 2014. Review and analysis of promising technologies with respect to Fifth generation networks, 1st International Conference on Networks and Soft Computing, ICNSC 2014 - Proceedings 2014, pp. 248-251.

[19] "Kolluru V R, Mahapatra K and Sudbudhi B, Real-Time Digital Simulation and Analysis of Sliding Mode and P&O MPPT Algorithms for a PV System, International Journal of Emerging Electric Power SystemsVolume 16, Issue 4, 1 August 2015, Pages 313-322."

[20] Kolluru V R, Mahapatra K and Sudbudhi B, Development and implementation of control algorithms for a photovoltaic system, Students Conference on Engineering and Systems, SCES 2013, Article number 6547525.

[21] Kolluru V R, Sarode S S, Patjoshi R K, Mahapatra K and Sudbudhi B, Design and implementation of an optimized sliding mode controller and compared with a conventional MPPT controller for a solar system, WSEAS Transactions on Systems and Control Volume 9, Issue 1, 2014, Pages 558-565