**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 5, May 2014, pg.954 – 960***

**RESEARCH ARTICLE**

# A HEURISTIC APPROACH FOR MIXED CRITICALITY JOBS SCHEDULING IN REAL TIME SYSTEMS

**Naveenkumar Koregol[1], Prof. M. V. Panduranga Rao[2]**

[1]Dept of Computer Science and Engineering, BTL Institute of Technology, Bangalore, India
[2]Dept of Computer Science and Engineering, BTL Institute of Technology, Bangalore, India
[1] Naveen08cs@gmail.com, [2] raomvp@yahoo.com

***Abstract:*** ***In this paper, we spotlight on scheduling and schedulability analysis problem for mixed criticality real - time system on a uni-processor platform. Here, a novel approach OLBA is proposed to maximize the utilization of the processor time in mixed criticality real time system.***

***Key words: WCET, Criticality level, scheduling, deadline, jobs***

## I. INTRODUCTION

A mixed criticality system is a system comprising computer hardware and software that can run various applications of distinct criticality, such as safety-critical and non-safety critical. In the real-time operating system, the selection of process scheduling algorithms to allot CPU resources can have a very crucial affect on the system execution as well as on the reliability and maintainability of the system [3].

The use of computers to handle real-time functions has increased rapidly. As a consequence, real-time systems — computer systems where the accuracy of a computation is reliant on both the logical results of the computation and the time at which these outcomes are displayed have become the focus of study.

 

The notion of "time" is of such significance in real-time application systems, and since these systems usually engross the allotment of one or more resources among an assortment of contending processes, the idea of scheduling is integral to real-time system design and analysis. Scheduling refers to a finite set of requests for resources[10], However, requests in real-time environments are often of a repeating character, Such systems are typically modelled as predetermined accumulations of simple, highly repetitive tasks, each of which produces jobs in a very predictable manner. These jobs have higher bounds upon their WCET requirements, and consorted deadlines.

Real-time scheduling theory has usually centered upon the improvement of algorithms for probability examination (finding out whether all jobs can finish execution by their specified deadlines) and run-time scheduling of such systems[6]. In hard-real-time systems, there are certain fundamental units of work known as jobs, which required to be executed by the system. Each such job has an associated deadline, and it is very important for the correctness of the system that all such jobs finish by their deadlines[7]. We limit our work to a preemptive model of scheduling – a job under execution on the processor may be disrupted, and its execution restarted at a later point in time.

## II. LITERATURE SURVEY

Strategic significance of mixed criticality certification is widely recognized and has been the subject of multiple workshops and working group meetings, some of the findings of which are highlighted in a white paper [4]. To our knowledge, the scheduling problem that arises from multiple certification requirements, at different criticality levels, was first identified and formalized by Vestal in [5], in the context of the fixed-priority preemptive uniprocessor scheduling of recurrent task systems.

Although many other real-time scheduling papers deal with mixed-criticality systems, they do not really deal with scheduling for certification. Lakshmanan and colleagues [8], [11], [12] deal with a different aspect of mixed-criticality systems from the one we focus on here, in that they do not directly address the certification issue. Nevertheless, [8], [11], [12] contain very interesting and novel ideas that merit mention. This body of work observes that the complete inter criticality isolation offered by the reservations approach may cause criticality inversion: preventing a higher criticality job from meeting its deadline while allowing lower criticality jobs to complete[9]. On the other hand, assigning priorities according to criticality may result in very poor processor utilization. An innovative slack-aware approach is proposed that builds a top priority-based scheduling (with priorities not necessarily assigned according to criticality), to allow for asymmetric protection of reservations thereby helping to lessen criticality inversion while retaining reasonable resource utilization[13].

 

The schedulability guarantee that their proposed algorithm can make may be stated as follows in the context of systems with only two criticality levels: all deadlines of high-criticality tasks are guaranteed to be met regardless of the runtime behavior of the lower criticality tasks, provided the execution of at most one higher criticality job overruns its WCET estimated at the lower criticality level (to no more than its WCET estimated at a level of assurance consistent with the higher criticality level)[1],[2]. Although this represents a significant improvement over prior approaches, it is far removed from what would pass certification: for that, we would need to guarantee that all higher criticality tasks complete by their deadlines even if they all execute for up to their WCET requirements specified at the higher level of assurance.

Pellizzoni et al. [14], use a reservations-based approach to ensure strong isolation among subsystems of different criticalities; this paper proposes innovative design and architectural techniques for preserving such isolation despite some necessary interaction (e.g., in the sharing of additional non preemptable resources) between jobs of different criticalities. The focus differs from ours in that the goal is not to optimize resource utilization, but to ensure isolation among the different criticalities.

## III. PROPOSED APPROACH

The proposed approach, Optimized Load Based Approach, maximizes the number of successfully executed jobs. The jobs are executed based on their arrival time, Criticality level and deadline. If the jobs arrival time is same for more than one job, then the job with high criticality level is executed. In case, If the jobs with same arrival time exists then the job having low criticality level and early deadline and high criticality job having sufficient execution time such that the low criticality job and high criticality job get executed before the deadline of the high criticality job then the low criticality job will be executed first and then the high criticality job.

The low criticality jobs are preferred for execution even though, the high criticality jobs are available at some time instant because, the low criticality job should not miss its deadline because of high criticality job.

The low criticality jobs are preferred over high criticality jobs only when, the low criticality job misses its deadline, and both the jobs, the low critical and high critical jobs get executed within high criticality job deadline, without missing the low criticality job deadline.

 

# IV. IMPLEMENTATION

Consider the below example,

| Job-id | Job arrival time | Deadline | criticality level | WCET | |
|---|---|---|---|---|---|
| | | | | LOW | HIGH |
| J1 | 3 | 4 | LOW | 1 | 1 |
| J2 | 8 | 10 | LOW | 1 | 1 |
| J3 | 3 | 7 | HIGH | 1 | 2 |
| J4 | 0 | 4 | HIGH | 1 | 3 |

Figure 1 : Input example.

In the above example(Figure 1), the job arrival time depicts the arrival of job and jobs will be scheduled for execution. The criticality level shows the critical level of job, If the critical level is low then, the WCET(worst case execution time) is considered from LOW column, If the critical level is high then, the WCET is considered from HIGH column.

The figure 2 shows the Input window, where the text file is given as input which consists of collection of jobs and its parameters such as jobs arrival time, deadline, WCET, criticality of the job and algorithm is selected for these set of jobs.

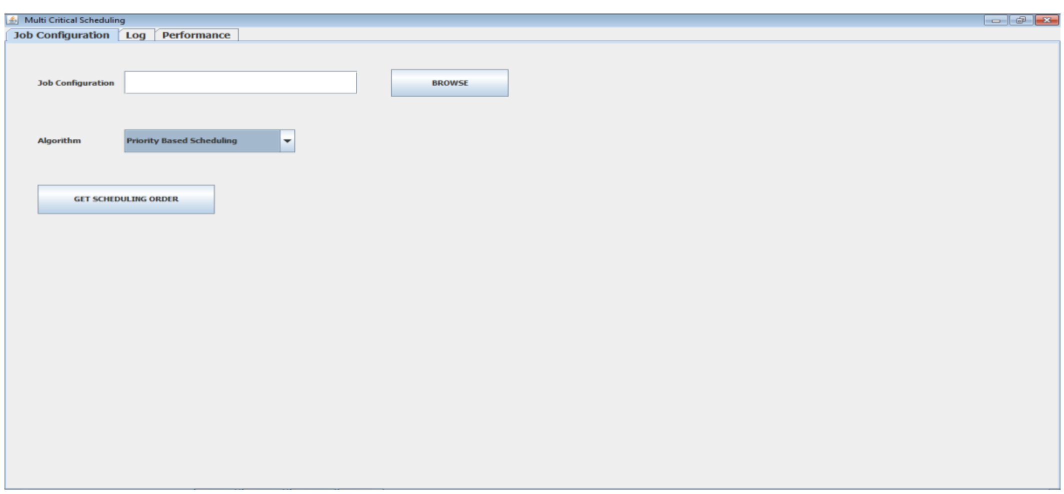


Figure 2 : Input window

If the jobs are executed based on arrival time ,deadline and criticality level, the jobs will be executed as shown in figure 3. The Job J4 executes from 0 to 3ms. The jobs J1 & J3 arrives at time instant 3 ms, The Job J3 has high criticality level then J1 so, J3 gets executed first and job J1 misses its deadline .

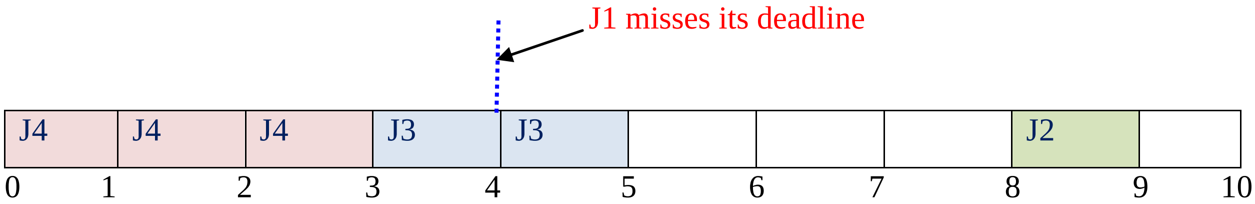


Figure 3 : Jobs scheduled using priority based approach

The figure 4 shows the output for priority based approach

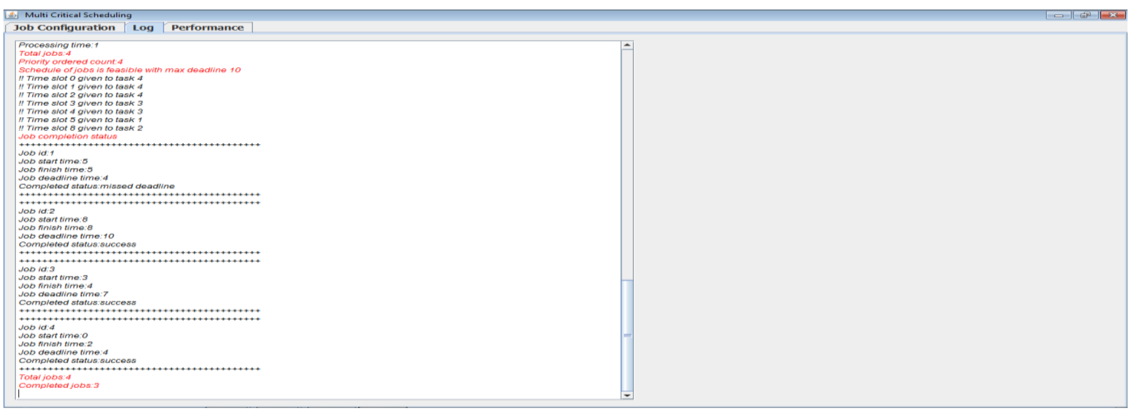


Figure 4 : shows output for priority based approach

If the jobs are executed according to the proposed approach, the jobs will be scheduled as shown in figure 5. The job J4 executes from 0 to 3 ms, and The jobs J1 & J3 arrives at time instant 3 ms, the job J1 get executed first even it has low priority as J1 has deadline 4ms and J3 executes after J1, J3 has deadline 7ms, which can be executed within its deadline.

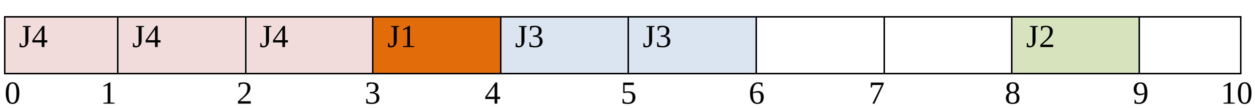


Figure 5 : Jobs scheduled using OLBA.

The figure 6 shows the output for the proposed, optimized load based approach

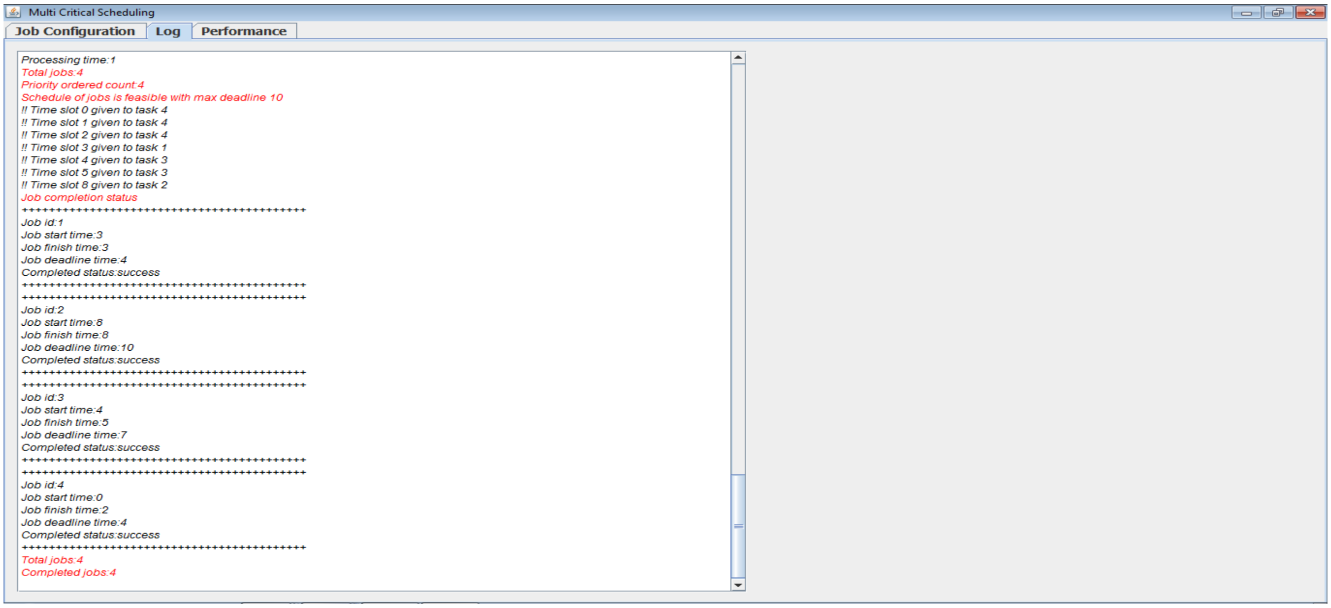


Figure 6 : Shows output using OLBA.

## V. CONCLUSION

We Propose a novel approach known as optimized load based approach to maximize the utilization of the processor time in mixed criticality real time system, as an extension for representing a simple kind of mixed-criticality system—those that can be represented as collections of independent jobs executing upon a single pre-emptable processor.

## REFERENCES

[1] Scheduling Real-Time Mixed-Criticality Jobs Sanjoy Baruah, Vincenzo Bonifaci, Gianlorenzo D'angelo, Haohan Li, Alberto Marchetti-Spaccamela, Nicole Megow, And Leen Stougie, Ieee Transactions On Computers, Vol. 61, No. 8, August 2012.

[2] S. Baruah, H. Li, and L. Stougie, "Towards the Design of Certifiable Mixed-Criticality Systems," Proc. the IEEE Real-Time Technology and Applications Symp. (RTAS), Apr. 2010.

[3] A Time-Driven Scheduling Model for Real-Time Operating Systems E. Douglas Jensen, C. Douglass Locke, Hideyuki Tokuda Computer Science Department Carnegie-Me//on University, Pittsburgh, PA 15213.

 

[4] J. Barhorst, T. Belote, P. Binns, J. Hoffman, J. Paunicka, P. Sarathy, J.S.P. Stanfill, D. Stuart, and R. Urzi, "White Paper: A Research Agenda for Mixed-Criticality Systems," http://www.cse.wustl.edu/~cdgill/CPSWEEK09_MCAR, Apr. 2009.

[5] S. Vestal, "Preemptive Scheduling of Multi-Criticality Systems with Varying Degrees of Execution Time Assurance," Proc. The Real-Time Systems Symp., pp. 239-243, Dec. 2007.

[6] ARINC, ARINC 653-1 Avionics Application Software Standard Interface, Oct. 2003.

[7] N.C. Audsley, "Optimal Priority Assignment and Feasibility of Static Priority Tasks with Arbitrary Start Times," technical report, The Univ. of York, England, 1991.

[8] K. Lakshmanan, D. de Niz, R.R. Rajkumar, and G. Moreno, "Resource Allocation in Distributed Mixed-Criticality Cyber- Physical Systems," Proc. the 30th Int'l Conf. Distributed Computing Systems, 2010.

[9] N.C. Audsley, "Flexible Scheduling in Hard-Real-Time Systems," PhD thesis, Dept. of Computer Science, Univ. of York, 1993.

[10] http://en.wikipedia.org/wiki/Mixed_criticality

[11] D. de Niz, K. Lakshmanan, and R.R. Rajkumar, "On the Scheduling of Mixed-Criticality Real-Time Task Sets," Proc. The Real-Time Systems Symp., pp. 291-300, 2009.

[12] J. Windsor and K. Hjortnaes, "Time and Space Partitioning in Spacecraft Avionics," Proc. IEEE Int'l Conf. Space Mission Challenges for Information Technology, pp. 13-20, 2009.

[13] S.K. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti- Spaccamela, N. Megow, and L. Stougie, "Scheduling Real-Time Mixed-Criticality Jobs," Proc. the 35th Int'l Symp. the Math.

[14] R. Pellizzoni, P. Meredith, M.Y. Nam, M. Sun, M. Caccamo, and L. Sha, "Handling Mixed Criticality in SoC-Based Real-Time Embedded Systems," Proc. the Int'l Conf. Embedded Software (EMSOFT), 2009.

## Authors Bibliography

**Naveenkumar Koregol** received the DIPLOMA in Computer Science and Engineering from Government polytechnic Bijapur and received the B.E. degree in Computer Science and Engineering from PES institute of technology and management, Shivamogga. At present persuing the Master of Technology in Computer Science and Engineering Department at BTL institute of Technology, Bangalore, Karnataka, India.

**M.V.Panduranga Rao** is a research scholar at National Institute ofTechnology Karnataka, Mangalore, India. His research interests are in the field of Real time and Embedded systems on Linux platform and Security. He has published various research papers across India and in IEEE international conference in Okinawa, Japan. He has also authored two reference books on Linux Internals. He is the Life member of Indian Society for Technical Education and IAENG.