RTOS implementation that cooperates with speed-up mechanism of embedded processor

Nobuyuki Shimada (310050)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 10, 2005

Keywords: Adaptive Dynamic Priority Method,Casablanca,Real Time Operating System,Scheduler.

1 Introduction

Recently, qeneral-purpose processor of which functions are reduced in order to cut costs and electric power consumption, is being used as control CPU for development of embedded system development. And μITRON, which is widely used as realtime operating system, lays schedule according to static priorities which are given to each task. But, recognizing possibility that tasks with static low priority cannot be executed, adaptive dynamic priority method[1], which will be able to increase tasks that are executed in time, by scheduling dynamically using necessary execution time estimated from tasks' time factor, are proposed.

This study aims, based on low-cost processor architecture[2] which specializes in the embedded system field, to equip realtime operating system, which can make the most of speed-up mechanism such as the speed-up interruption response mechanism of this architecture and cache control mechanism. Considering development efficiency of application in the field of embedded system, interface will follow the standardized μITRON4.0[3]. And changing the existing task scheduler, we will adopt static priority method as a base and adaptive dynamic priority method partially.

2 Composition of scheduler

scheduler to be equipped will have static priority method and adaptive dynamic priority method as function, and will use basically static priority method and partially adaptive priority method. The existing ready queue will be equipped without changing its composition, which will enable plural scheduler to be easily changed.

In order to reduce "Overhead" during calculating static priority method, we propose the following methods.

1. The scheduler can be called with plural service calls but adaptive dynamic priority method will be used only to finish own tasks(ext_tsk).
2. "Overhead" from calculating priority will be reduced through, not calculating priority of all tasks but calculating "n"-piece tasks which have higher priority and are connected with ready queue.
3. Such composition will be adopted that only adaptive dynamic priority method will be used as for ready queue with 1 priority[2]. And when connecting tasks to ready queue with 1 priority, connecting will be executed according to priority of static priority, not with FIFO method.

3 Packaging method

This study has made CPU simulator and equipped real-time-operating system with μITRON4.0 specification to evaluate performance of proposed composition. Composition of scheduler has both static priority method and adaptive dynamic priority method.

4 Evaluation

Our trial shows that it is possible to calculate according to dynamic priority method with very little "Overhead". It also shows reduction of number of task's deadline over. And it shows that High-speed interrupt response mechanism and data cache control mechanism of embedded CPU.

[2]highest priority.

5 Summary

The composition proposed in this study can calculate according to dynamic priority method with very little "Overhead". And it can reduce the number of task's deadline over.

References

[1] Adaptive scheduling method in Kazumichi Kuritani, Kiyofumi Tanaka, and realtime Operating System, electronic telecommunication society technology research reports CPSY, Vol.102, No.478, and pp.127–132,2002.

[2] Kiyofumi Tanaka, Takashi Matsumoto, "Casablanca: A Real-Time RISC Core for Embedded Systems ", International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the INternet, 2001, CD-ROM.

[3] μITRON4.0 specific Ver. 4.00.00
http://222.sakamura-lab.org/TRON/ITRON/