Title	
Author(s)	,
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4345
Rights	
Description	Supervisor: , ,

Research on Implementation of Embedded Real-Time Database Library

Kouhei Hayashi (0610072)

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

February 7, 2008

1 Introduction

Recently, many embedded systems demand large-scaled and complicated data management. Such system has mechanisms for efficient data manipulation on RTOS. The mechanisms are called Data Base Management System (DBMS). DBMS has an ability of database-management, and can retrieve and update a large amount of data efficiently. Generally, the DBMS is implemented in the system as a software component. DBMS is required to be fast software which can handle data with efficient algorithm in the limited memory space[1]. In this research, the DBMS is implemented as a software component. And, I propose the technique of automatically generating an executable binary code that is suitable for an embedded system. This technique reduces memory usage while keeping in real-time properties[2].

2 Target Real-Time System

Recently, ARM[3] is widely used as an embedded processor. The Thumb instruction set consists of 16-bit instructions that are a subset of the 32-bit

instructions of the standard ARM. The system developer can optimize the performance and the code size of each routine by using ARM and Thumb, and can make an optimized binary code[4]. μITRON[5], a real-time operating system specification for embedded systems, introduces static priorities to control the processing order of tasks. Therefore, using feature of ARM processor and the static priorities of tasks, the system developer can implement a system where the performance and the code size are well-balanced.

3 Implementation of DBMS Library

In this research, the DBMS is implemented as a software component[6]. Query functions included in DBMS functions are provided by both ARM and Thumb versions in this research. I propose a new configuration tool which automatically selects the binary codes of query functions according to static priorities of tasks when the binaries are linked together with. For this reason, the tool can reduce the binary size of DBMS automatically while it can holds real-time execution. Moreover, the tool can reduce system developer's burden.

4 Evaluation

The DBMS was implemented as a software component that conforms to μITRON4.0 Standard Profile. A proposed technique which reduces binary size and holds real-time execution is implemented in the configurator of DBMS. To evaluate the proposal technique, the ARM simulator was developed. RTOS for the ARM processor was developed by using the kernel library of μITRON4.0 Standard Profile[7]. It was confirmed that the size of the DBMS functions decreased when a task-set with the DBMS library was applied. Moreover, the task-set using the proposed technique was evaluated by the simulation, and it was confirmed that the technique achieved average response time almost equal to the ARM mode.

5 Conclusion

In this research, I propose the technique for reducing the execution binary size using static priorities of the ITRON's tasks. It was shown to be able to reduce the binary size of DBMS by using the technique. Moreover, it was indicated that it achieved average response time equal with ARM in the evaluation by the simulation. It was shown to keep real-time properties while reducing the binary size by using the proposal technique.

References

[1]   .“
     ”,CQ   2006.

[2] G. C. Buttazzo, "Hard Real-Time Computing Systems : Predictable Scheduling Algorithums And Applications," Springer, 2004

[3] ARM Limited."ARM Architecture Reference Manual",Pearson Education, 1996-2000.

[4] SLOSS, Symes, Wright, "ARM System Developer's Guide", MORGAN KAUFMANN PUBLISHERS, 2004.

[5] ITRON Committee, TRON ASSOCIATION."μITRON4.0 Specification Ver.4.00.00".

[6] Empress Software   ."http://www.empress-jp.com/".

[7] Kiyofumi Tanaka,"Real-Time Adaptive Task Scheduling", Proc. of International Conference on Embedded Systems and Applications (ESA'05), pp. 24 - 30, 2005