Study on Binary Construction According to Static Priority of Tasks

Takashi Sasayama (0610040)

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

February 5, 2009

Keywords: ARM Processor, μITRON4.0 specification OS, Static Priority, Configurator .

1 Introduction

Recently many embedded systems have limited amount of memory, due to the fact that the devices are limited in hardware size and budget. Therefore the size of the code and memory is required to be reduced. To reduce the code size, processors for embedded systems equip different Instruction Set Architecture. ARM[1] is one the embedded processor that represents the feature. An ARM processor is a 32-bit RISC processor and widely used by embedded systems. The ARM processor equips with a normal 32-bit ARM mode and a Thumb mode, which is a 16-bit compressed code.In this research, I implement a μITRON4.0 specification OS[2] for an ARM processor. The proposed method automatically switches between ARM and Thumb mode according to the ITRON's static priority of task. The execution binary code is optimized and the execution binary is built in real-time properties[3], thus reducing the developer's load.

2 ARM and μITRON4.0 Specification OS

The ARM processor is a 32-bit RISC processor widely used in embedded systems. It is equipped with 32-bit ARM mode and 16-bit Thumb mode.

Thumb mode in comparison with ARM mode, has an increase in execution code length resulting longer execution time. Nevertheless, the Thumb mode has it's advantage of reduced code size. It is possible to mix the two modes into one execution binary.Recent developments in embedded system have a wide opportunity in using Embedded OS. Improved real-time factors and reduction in load of application development are achievable due to the use of the OS. But Embedded OS in comparison with common OS, Real-time factors are great importance to Embedded OS, thus Embedded OS's require dead-line miss and average response time information. In this research, I use the μITRON4.0 specification OS, one of the OS for the Embedded system. The μITRON4.0 specification OS employs scheduling system in accordance to the static priority of task.

3 Proposal Technique

In this research I propose a execution binary optimized system. This will be done by implementing and running the object on a OS targeted to run on a μITRON4.0 specification application that runs on an ARM processor, and also by designing a simulator. The execution binary optimized system automatically switches between the ARM and Thumb mode according to the ITRON's static priority of task. This aim is to reduce the developer's manual operation of execution binary optimization, thus reducing the develop load. The Algorithm of the execution binary optimize system is set by the programable threshold. Next, the threshold is compared with the static priority.When the priority is high, the mode is switched to ARM mode. Lower priority switches the mode into Thumb mode. The simulator measures the average response time of task and dead-line misses. Real-time property of the execution binary is aimed to support development. The simulator targets simulating the ARM7TDMI environment.

4 Evaluation

Optimized system's execution binary was implemented on a μITRON4.0 specification application targeted to run on an ARM processor. And the execution binary was built according to static priority of task automaticlly

using the optimized system. To evaluate real-time property of execution binary the ARM simulator was implemented. An ARM simulator was built to evaluate the real-time property of execution binary. Result of evaluation, I propose method reduce execution binary size. The results of the evaluation confirmed that the size of the execution binary code was reduced when the proposed method is used. Also, the average response time and dead-line misses were measurable by the simulator.

5 Conclusion

In this research, I have implemented a μITRON4.0 specification OS that runs on an ARM processor. The results show that the size of the execution binary code was able to be reduced by developing a tool that is able to configure the execution binary code according to the static priority of task. In addition an ARM simulator capable of evaluating the real-time property of execution binary was implemented.

References

[1] ARM Limited.“ ARM Architecture Reference Manual ”,Pearson Education, 1996-2000.

[2] トロン協会.“ μITRON4.0仕様 Ver.4.03.00 ”.

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