早稲田大学大学院 基幹理工学研究科

博 士 論 文 概 要

論 文 題 目

Analyzing Real-Time Performance Problems in Embedded Linux

組込み Linux におけるカーネルのリアルタイム性能に関する問題の分析

申 請 者

権	奇 徳
Ki Duk	Kwon

情報理工学専攻 分散システム研究

2010 年 5 月

These days embedded systems are used in various fields. Embedded systems, which are usually designed to perform specific purpose, are significantly developing with embedded software technology. For example, there are many devices and systems based on embedded systems such as elevators, TV, MP3 players, set-top boxes, digital cameras, PDA, smart phones, mobile phones, engine control of automobiles and many medical devices. As one can see, they are closely related to our daily life, and they have a lot of applications. For these reasons, the demand of embedded systems and software for the market is increasing rapidly.

In recent years, among embedded software technologies, embedded operating systems have changed innovatively. Especially, an embedded Linux has progressed critically compared to RTOS (VxWorks, pSOS, QNX, VRTX) which are mainly used in home appliances and other embedded applications. The biggest reason is that hardware for embedded systems have significantly improved quantitatively and qualitatively. Thanks to remarkable progress of the hardware, embedded applications can be more active on desktop environments, and an embedded Linux could be much faster than any other embedded operating systems. In addition, an embedded Linux fulfills not only high-speed but also low-cost development since it is based on open source development.

Generally speaking, software is grouped into Operating System (OS), Middleware and Application. Because each classified has different features and functions, every group is important as itself. However, the OS is remarkably important since it controls and manages all of hardware and systems to keep running applications and middleware safely with respect to resources such as memory or CPU or I/O devices. Therefore, it could be the fundamental of Computer. Many OS such as Unix, Linux and Windows are currently used in personal computers. Actually, Many OS for embedded systems are also used a lot in various fields. For example, there are Android from Google, the iPhone OS from Apple, embedded Windows from Microsoft and embedded Linux. Especially, embedded Linux has been widely used because it retains many powerful features of general Linux such as multi-tasking, various network environments, different types of file systems, system scalabilities and is served at no cost. On the other hand, embedded systems compared to general systems have a disadvantage. There are not enough skilled embedded system developers in

addition to many restrictions on hardware. However, market needs a fast development cycle but it is hard because of these disadvantages. Today embedded systems have become one of the most important parts on all of industries. Therefore, debugging a various problems and improving the performance of embedded systems also is becoming highly important.

While developing embedded systems, problems usually can be categorized into two groups – user level and kernel level. In user level, it is not that hard to fix since there are a lot of tools for developing and debugging. On the other hands, when a problem occurs at kernel level, it is much more difficult to fix than in user level. Tools for kernel development usually provide minimum functions and the functions are not limited to fix up problems. Moreover, even though an embedded system is currently operating without any problems, there is a possibility for an unexpected error to occur. There is no one hundred percent perfect system. Even currently well working embedded systems are also likely to have hidden problems. These problems usually do not show up during testing and debugging. Even after a system has been released as a product on the markets, however, it still has a possibility to contain errors or problem. Furthermore, this unexpected small errors cause inconvenience for many users and can possibly become life-threatening catastrophic problem.

An embedded system's project is usually complex and requires developers with a high level of comprehension about hardware and software, compared to general software's project. In addition, an embedded system's projects have a lot of constraints and limitations. Due to these characteristics, it is not easy to propose a solution for problems which occur during project development. These days various solutions for these problems are proposed; however, unfortunately the structure of embedded systems is becoming more complex quickly. So, skilled technicians and their experience are highly evaluated. However, there is a limitation to rely on the technicians' experience and skill to handle problems. Therefore, in order to analyze and solve problems in systems, a framework which makes performance measurement and analysis is urgently needed.

In this paper, we propose a tool called Kernel Analysis System (KAS) which can resolve

problems quickly in the kernel. First, the KAS uses LTTng to log events. The event information from logs is analyzed by the KAS. The KAS has three main layers. First, is the detection layer. In this layer, the KAS finds out problems by checking all events that occurred in the kernel and counting there number. Then, it sends all information of problems to the next layer. Second, is the separation layer. In this layer, the KAS separates only the events related to problems from all executed events. Thanks to this layer, developers can work effectively since they do not need to look up all event logs. Third, is the analysis layer. In this layer, the KAS analyzes the problems by calculating all the events' running time and the number of error occurrences so as to figure out the cause of the problems. If there are unexpected occurrences or time consumed, developer can easily find out the problem. The analysis result from the KAS is very efficient to analyze and fix up problems quickly compared to just analyze all of layer.

The rest of the paper is organized as follows. In Section 1, we introduce challenge and contribution. In the challenge, we explain the reason why event log analysis is required, and how we could control problems in kernel. In the contribution, we describe the benefit of the framework we suggest for system development. In Section 2, we explain all the background knowledge which is related to this paper in order to situate our work. We explain about embedded systems, Linux kernel and embedded Linux which is an embedded operating system. In addition, we describe system monitoring, event log and problem analysis to suggest solution for problems. In Section 3, we attend related work, reviews and discussions. We explain the tools for system monitoring and event log analysis. In Section 4, we propose a new system framework. A system framework has three main layer – Detection Layer, Separation Layer and Analysis Layer. We explain each layer in detail. In Section 5, we analyze the kernel timer by using the kernel analysis system which we suggest. We analyze the kernel timer and delay problem. Moreover, we find the reason of the problem after high resolution timer experiment at overloaded system. Finally, in Section 6, we make a conclusion and suggest possible future directions. We explain about the system framework and remained problems of the KAS to be solved.

早稲田大学　博士（工学）　学位申請　研究業績書

氏名　　権　奇　徳　　印

（ 2010　年　5月　現在 ）

種 類 別	題名、　発表・発行掲載誌名、　発表・発行年月、　連名者（申請者含む）
論文誌	1. *Ki Duk Kwon, Midori Sugaya and Tatsuo Nakajima*. “KTAS: Analysis of Timer Latency for Embedded Linux Kernel”, International Journal of Advanced Science and Technology (IJAST), vol.18, May, 2010 (Accepted)
国際会議	1. *Kiduk Kwon, Midori Sugaya, Tatsuo Nakajima*. "Analysis of Embedded Linux Using Kernel Analysis System," The 6th IEEE International Conferences on Embedded Software and Systems (ICESS), pp.417-422, May 2009 2. *Kiduk Kwon, Midori Sugaya, Tatsuo Nakajima*. "Analysis of High Resolution Timer Latency Using Kernel Analysis System in Embedded System”, 12th IEEE Symposium on Object/component/service-oriented Real-time distributed Computing Co-located with First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009),pp.122-126, March 2009. 3. *Tatsuo Nakajima, Hiroo Ishikawa, Yuki Kinebuchi, Midori Sugaya, Lei Sun, Alexandre Courbot, Andrej van der Zee, Aleksi Aalto, and Kwon Ki Duk.* “An Operating System Architecture for Future Information Appliances. ”The 6th IFIP WG 10.2 International Workshop, SEUS 2008, October 2008. Lecture Notes in Computer Science (LNCS), Vol. 5287 / 2008, pp. 292-303.

早稲田大学　博士（工学）　学位申請　研究業績書

種 類 別	題名、　発表・発行掲載誌名、　発表・発行年月、　連名者（申請者含む）
国内会議	1. *Kwon Ki Duk, Sugaya Midori, Ohno Yuuki and Nakajima Tatsuo*. “Performance analysis of information explosion by using LTTng”, Information Processing Society of Japan (IPSJ), 5-299, March 2008. 2. *Ohno Yuuki, Sugaya Midori and Kwon Ki-Duk.* “Performance analysis of distributed applications in the information explosion era”, Information Processing Society of Japan (IPSJ), 5-147, March 2008. 3. *Ohno Yuuki, Sugaya Midori and Kwon Ki-Duk, Nakajima Tatsuo* “**リソースモニタリングによる異常検出システム**”, The 6th Dependablity System Workshop (DSW’08 Summer), pp.71-76, 2008.