JAIST Repository
https://dspace.jaist.ac.jp/

Title	Verification of automotive operating systems for multi-core processors
Author(s)	マリーヌアン, パッターウット
Citation	
Issue Date	2015-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12933
Rights	
Description	Supervisor:Toshiaki Aoki, School of Information Science, Master

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Verification of automotive operating systems for multi-core processors

Pattaravut Maleehuan (1310205)

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

August 6, 2015

Keywords: formal verification, multi-core processor, memory consistency model.

Currently, the automotive systems provide many functionalities for automobile. Although those functionalities are able to support our lives, the violation of those functionalities may be serious issue that we have to concern. Obviously, the automotive systems require high-reliability of the system to ensure safeness of our lives. In this research, we specifically focus on *operating systems* of automotive systems. The operating system is the basis component which provide services to serve application software. Thus, the correctness of the operating systems is the serious condition to correctly implement the operating system kernel.

In addition, since the demand of usage is increasing, the *multi-core processors* are adopt in the automotive systems for performance improving. The multi-core system is the system that have multi-processors with shared-memory. In the shared-memory systems, according to[1], the results of programs might be not same as the *sequential execution*, which is the execution which follow the program order specified by programs. As programmer's point-of-view, these results maybe unexpected results. These results are affected by *memory consistency models* which are define the behaviors of memory in shared-memory systems.

In the shared-memory systems, the systems allow each processor to access the memory locations simultaneously. Moreover, each processor are able to issue the memory accesses out-of-order because of optimization mechanisms. Since each processor is independent to each other, the access order of memory access might be different. Hence, the memory maybe not consistence among processors. The specifications of memory consistency models ensure the execution order of memory accessing to shared-memory locations. Unfortunately, these behaviors are not appeared explicitly. It might be difficult to consider the behaviors of program execution.

Our research aims to verify the automotive operating systems for multi-core systems. Since the behaviors of program execution are not appeared explicitly, the verification might be difficult. In addition, the behaviors of program executions are affected by the hardware architecture. That means we cannot verify the programs on multi-core systems by themselves. Hence, this research will provide the verification which take the

[1]Kourosh Gharachorloo (1995). "Memory Consistency Models for shared-memory multiprocessors". PhD thesis. Stanford University.

hardware behaviors into account for ensure the correctness. Moreover, the verification of the whole operating systems will be difficult because operating system is a complex system. Therefore, the scope of this research considers only multiprocessor programs to provide the verification for multi-core systems.

In software development, there are many approaches to ensure the correctness of the software. Due to the automotive operating systems require high-reliability, the *formal verification* is adopt in this research. Since the behaviors of program execution in multi-core systems might be complicated, we apply the *theorem proving* approach instead of model checking. Therefore, due to the behaviors of hardware is significant issue, this research provides the formal model which represent such behaviors. Then, the verification method is proposed to provide the proofs based on our formal model.