

A Study on the Platform Independent SIL Code based Compiler for Smart Virtual Machine

Yunsik Son¹, Jaehyun Kim², Yangsun Lee^{2*}

¹Dept. of Computer Engineering, Dongguk University
26 3-Ga Phil-Dong, Jung-Gu, Seoul 100-715, KOREA
sonbug@dongguk.edu

²Dept. of Computer Engineering, Seokyeong University
16-1 Jungneung-Dong, Sungbuk-Ku, Seoul 136-704, KOREA
statsr@skuniv.ac.kr, *Corresponding Author : yslee@skuniv.ac.kr

Abstract. The SVM (Smart Virtual Machine) is a stack based platform independent virtual machine solution, loaded on smart devices, which allows dynamic application programs to be downloaded and run platform independently. The SIL (Smart Intermediate Language) has been designed as a standard model of virtual machine codes for ordinary smart phones and embedded systems. It is a set of stack based commands which has the characteristics of language independence, hardware independence and platform independence. In this paper, we present the platform independent virtual machine code based the compilers for the SVM. This makes the existing mobile contents to be ported quickly, and to be executed to a different platform.

Keywords: Compiler, Virtual Machine Code, Smart Device, SVM (Smart Virtual Machine), SIL (Smart Intermediate Language), Platform Independence

1 Introduction

The previous development environments for smart phone contents are needed to generate specific target code depending on target devices or platforms, and each platform has its own developing language. Therefore, even if the same contents are to be used, it must be redeveloped depending on the target machine and a compiler for that specific machine is needed, making the contents development process very inefficient [1-3]. The SVM (Smart Virtual Machine) is a virtual machine solution which aims to resolve such problems, and it uses the SIL (Smart Intermediate Language) code as an input [4]. In this study, we present the platform independent code based the compilers on a SVM. This makes the existing mobile contents to be ported quickly, and to be executed to a different platform.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(No.20110006884).

2 Related Researches

2.1 Smart Virtual Machine

The SVM (Smart Virtual Machine) is a stack based virtual machine solution, loaded on smart devices, which allows dynamic application programs to be downloaded and run platform independently. The SVM is designed to use an intermediary language, SIL, which is capable of accommodating both procedural and object-oriented languages. It has the advantage of accommodating languages such as C/C++, and Java which are currently used by a majority of developers [1-3]. Figure 1 shows a system configuration of SVM system.

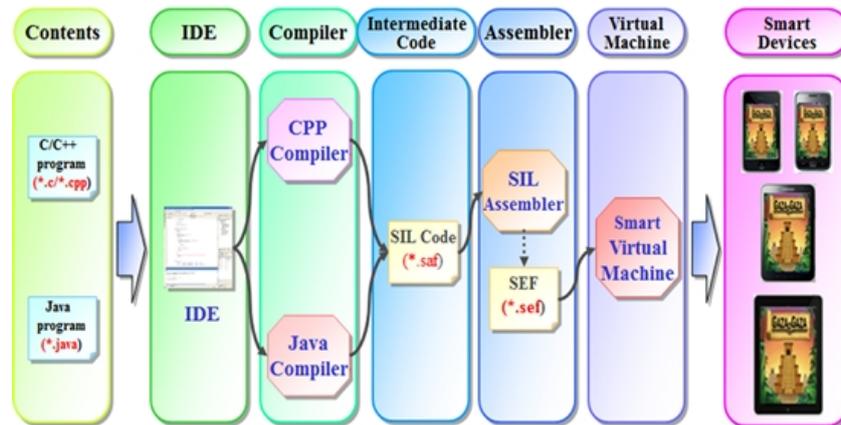


Fig. 1. System Configuration of SVM

The SVM system consists of three parts; a compiler which compiles application programs to create a **SAF** (Smart Assembly Format) form file made from SIL code, an assembler which converts the SAF file into the execution formation **SEF** (Smart Executable Format), and a virtual machine which receives the SEF form file and runs the program.

SIL (Smart Intermediate Language) and SEF are each intermediate language and execution format for SVM. As a standardized model of virtual machine codes for general smart phones and embedded systems, it is designed for language, hardware and platform independency. It has a set of stack based operation code. Programs composed of SIL code are converted to SEF by the assembler. SEF, an execution file format in SVM, is composed of the header domain which expresses the SEF file's composition, program segment and debugging segment which expresses debugging information [1-3].

3 The Platform Independent SIL Code based Compiler

Existing compilers are designed to translate source programs into target codes and then execute them. Reusability and portability of codes will also decrease because target codes have to be different for each platform they are run under. To solve these problems, the SVM (Smart Virtual Machine) was designed in a virtual machine solution. Also, the virtual machine based compilers and SIL, an intermediate language, was used to cover the conventional object-oriented programming languages like C/C++, and Java.

3.1 The C/C++ Compiler

The C++ compiler supports the contents written in C/C++ language on SVM (Smart Virtual Machine) which generates platform independently stack-based SIL (Smart Intermediate Language) code as target code. The C++ to SIL compiler embodies the characteristics of the C++ language and therefore was designed with six different parts; preprocessor, lexical analysis, syntax analysis, symbol table, semantic analysis and code generation [5]. Figure 2 shows a system configuration of the C++ compiler.

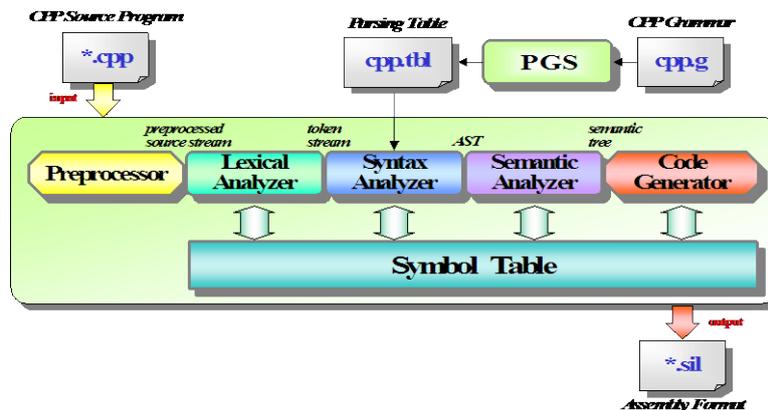


Fig. 2. System Configuration of the C++ Compiler

3.2 The Java Compiler

The Java compiler supports the contents written in Java language on SVM which generates platform independently stack-based SIL code as target code. The Java to SIL compiler embodies the characteristics of the Java language and therefore was designed with five different parts; syntax analysis, class file loader, symbol information collection module, semantic analysis and code generation [6]. Figure 3 shows a system configuration of the Java compiler.

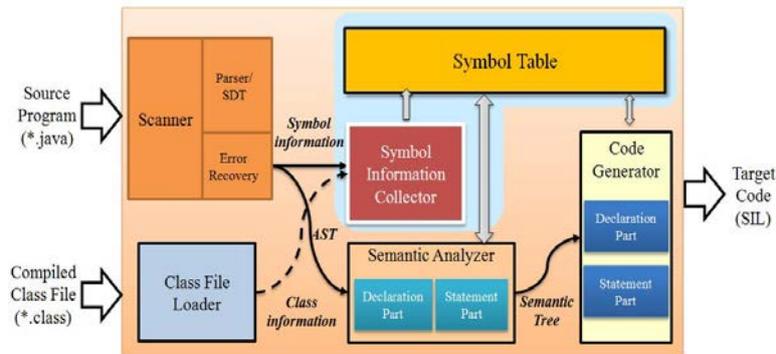


Fig. 3. System Configuration of the Java Compiler

4 Conclusion

Virtual machines refer to a means of using the same application programs over a variety of different processors or operating systems. In this study, platform independent SIL code based compilers for smart virtual machine were designed and implemented to run a program originally created for another platform in order to enable its use on an SVM (Smart Virtual Machine). Programs developed for use as Android or iOS content could be run on an SVM using this compiler, thus reducing the costs involved in producing such content.

References

1. Lee, Y.S, Son, Y.: A Study on the Smart Virtual Machine for Smart Devices. Information-an International Interdisciplinary Journal, International Information Institute, Vol.16, No.2, pp.1465-1472 (2013)
2. Han, S.M., Son, Y., Lee, Y.S.: "Design and Implementation of the Smart Virtual Machine for Smart Cross Platform," Journal of Korea Multimedia Society, Vol.16, No.2, pp.190-197 (2013)
3. Son, Y., Lee, Y.S.: "A Study on the Smart Virtual Machine for Executing Virtual Machine Codes on Smart Platforms." International Journal of Smart Home, Vol.6, No.4, pp.93-106 (2012)
4. James E. Smith, Ravi Nair, Virtual Machines, Elsevier (2005)
5. Lee, Y.S, Son, Y.: "A Study on Verification and Analysis of Symbol Tables for Development of the C++ Compiler," International Journal of Multimedia and Ubiquitous Engineering, SERSC, Vol.7, No.4, pp.175-186 (2012)
6. Son, Y., Lee, Y.S.: "A Study on the Java Compiler for the Smart Virtual Machine Platform," Multimedia, Computer Graphics and Broadcasting, CCIS, Springer, Vol.353, pp.135-140 (2012)