

가상기계 코드의 커버링 문제

남동근^o, 오세만
동국대학교 컴퓨터 공학과
{capsrom^o, smoh}@dongguk.edu

Covering Problem for Virtual Machine Code

Dongkeun Nam^o, Seman Oh
Dept. of Computer Engineering, Dongguk University

요 약

임베디드 시스템을 위한 가상기계 기술은 모바일 디바이스와 디지털 TV 등의 다운로드 솔루션에 꼭 필요한 소프트웨어 기술이다. 현재 EVM(Embedded Virtual Machine)이라 명명되어진 임베디드 시스템을 위한 가상 기계에 대한 연구가 진행 중이며, EVM의 중간 언어로 SIL(Standard Intermediate Language)이 설계되었다.

본 논문에서는 임베디드 시스템을 위한 가상기계의 표준 중간 언어로서 SIL의 완벽성을 증명한다. 기존에 이미 널리 사용되고 있는 가상기계 코드를 SIL 카테고리에 매핑시킴으로써 구조적으로 SIL의 완벽성을 증명한다. SIL은 표준 중간 언어로서의 요구 사항을 갖추고 있으며 6개의 메인 카테고리 와 16개의 서브 카테고리로 나누어 진다. Oolong과 .NET IL 등 기존의 가상기계 코드들의 카테고리는 SIL 카테고리에 매핑된다.

1. 서 론

가상기계란 하드웨어로 이루어진 물리적 시스템과는 달리 소프트웨어로 제작되어 논리적인 시스템 구성을 갖는 개념적인 컴퓨터이다. 가상기계 기술을 이용하면 응용 프로그램 실행 환경인 프로세서나 운영 체제가 바뀌더라도 응용 프로그램을 수정하지 않고 사용할 수 있다.

최근에는 GVM, KVM 등 모바일 단말기를 위한 가상기계들이 개발되면서 그 중요성이 더욱 부각되고 있다. 특히, 임베디드 시스템을 위한 가상기계 기술은 모바일 디바이스와 디지털 TV 등에 탑재할 수 있는 핵심 기술로서 다운로드 솔루션에서는 꼭 필요한 소프트웨어 기술이다.

현재 Microsoft의 C#언어와 SUN사의 Java언어 등을 모두 수용할 수 있는 가상기계에 대한 연구가 진행 중이다. EVM(Embedded Virtual Machine)이라 명명되어진 이 가상기계 솔루션은 C#, Java 등 객체지향 언어 뿐만 아니라 C언어와 같이 순차적인 언어로 작성된 프로그램을 어셈블리 언어 형태인 *.sil을 거쳐서 *.evm 파일 형태로 변환하여 임베디드 시스템에 탑재된 가상기계에서 실행할 수 있도록 한다.

SIL(Standard Intermediate Language)은 EVM의 가상기계 코드로써 바이트 코드, .NET IL 등을 입력으로 받는 번역기의 목적 코드이다.

SIL은 다양한 프로그래밍 언어를 수용하기 위해서 기존의 가상기계 어셈블리 언어들의 분석을 토대로 정의

본 연구는 한국과학재단 목적기초연구(R01-2002-000-00041-0)지원으로 수행되었음.

되었으며, 객체지향 언어와 순차적 언어를 모두 수용하기 위한 연산 코드 집합을 갖고 있다.

본 논문에서는 SIL 설계 과정에서 나타난 임베디드 시스템을 위한 표준 중간 언어로서 SIL의 완벽성 증명 문제를 해결하고자 한다.

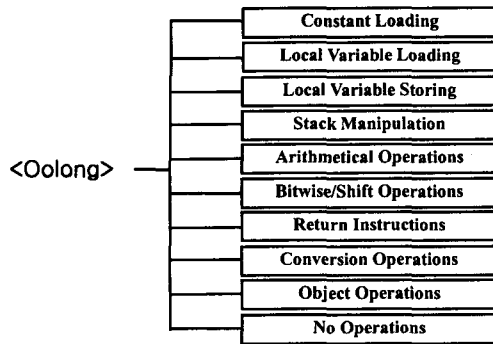
SIL의 설계 요건이 표준 중간 언어로서의 필요 조건임을 논하고, 기존에 널리 사용되고 있는 가상기계 코드를 SIL 카테고리에 매핑시킴으로써 SIL의 완벽성을 구조적으로 증명한다. 매핑 대상은 JVM의 가상기계 코드인 Oolong과 .NET 플랫폼의 가상기계 코드인 .NET IL 이다.

2. 관련 연구

2.1 Oolong

Oolong은 클래스 파일을 표현하기 위한 JVM의 가상기계 코드이다. JVM이 실행하는 클래스 파일 포맷은 프로그래머가 분석하거나 수정하기가 매우 힘들기 때문에 Oolong 언어가 만들어졌다. Oolong은 Gnoo 디스어셈블러를 사용하여 클래스 파일로부터 추출해 낼 수 있으며, 클래스 파일 포맷과 거의 동일하지만 클래스 파일에 비해서 읽고 쓰기가 훨씬 쉽다.

Oolong은 John Meyer의 Jasmin언어를 기반으로 만들어졌으며 Jasmin의 니모닉을 그대로 사용하고 있다. Oolong의 연산 코드는 228개이고 10개의 카테고리로 분류할 수 있다. [그림1]은 Joshua Engel이 분류한 Oolong 코드의 카테고리이다[5].



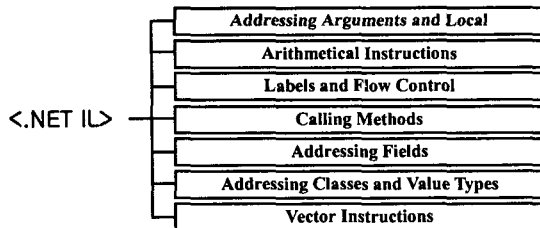
[그림1] Oolong 연산 코드 카테고리

2.2 .NET IL(MSIL)

.NET IL은 .NET 플랫폼에서 사용되는 가상기계 코드이며, 컴파일러나 다른 도구에 의해 소스 코드로부터 쉽게 생성될 수 있도록 설계된 스택 기반의 명령어 집합이다.

Oolong과는 다르게 처음부터 언어 독립적으로 설계되어 포괄적인 프로그래밍을 목적으로 하기 때문에 프로그램의 기능 및 구조의 변화에 잘 적응하는 언어이다.

.NET에서 제공하는 Ildasm 디어셈블러를 이용하여 *.dll이나 *.exe에서 .NET IL을 추출할 수 있다. .NET IL의 연산 코드는 236개이며 7개의 카테고리로 분류할 수 있다. [그림2]는 Serge Lidin이 분류한 .NET IL의 카테고리이다[7].



[그림2].NET IL 연산 코드 카테고리

2.3 EVM

EVM은 모바일 디바이스, 셋톱 박스, 디지털 TV 등에 탑재되어 동적 응용 프로그램을 다운로드하여 실행할 수 있는 가상기계 솔루션이다.

EVM은 크게 번역기, 어셈블러, 가상기계의 세 부분으로 나뉘어진다. EVM의 가상기계 코드는 SIL이며 어셈블러를 통해 EVM의 실행 포맷인 *.evm으로 변환된다. EVM의 가상기계 부분은 계층적인 구조로 설계되어 리타겟팅 과정의 부담을 최소화 한다.

3. 표준 중간 언어(Standard Intermediate Language)

3.1 가상기계 코드의 요구 사항

EVM의 가상기계 코드인 SIL은 일반적인 임베디드 시스템을 위한 가상기계 코드의 표준화 모델로 설계되었다.

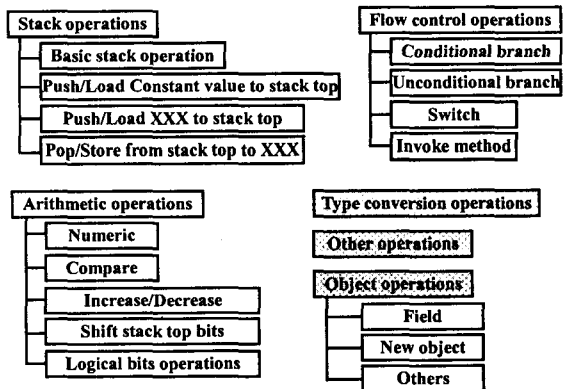
SIL은 스택 기반의 명령어 집합이며 언어 독립성과 하드웨어 및 플랫폼 독립성을 갖고 있다. SIL은 다양한 프로그래밍 언어를 수용하기 위해서 바이트 코드, .NET IL 등 기존의 가상기계 코드들의 분석을 토대로 정의 되었으며, 객체 지향 언어와 순차적 언어를 모두 수용하기 위한 연산 코드 집합을 갖고 있다.

SIL은 클래스 선언 등 특정 작업의 수행을 나타내는 의사 코드와 실제 명령어에 대응되는 연산 코드로 이루어져 있다. 연산 코드는 특정 하드웨어나 소스 언어에 종속되지 않는 추상적인 형태를 지니며, 어셈블리 언어 수준의 디버깅을 용이하게 하기 위해 일관성 있는 이름 규칙을 적용하여 가독성 높은 니모닉으로 정의되어 있다. 또한 최적화를 위한 short form 연산 코드를 갖고 있다.

3.2 가상기계 코드 카테고리

SIL은 6개의 카테고리로 분류할 수 있으며 각각의 카테고리는 서브 카테고리를 갖는다.

Stack operations 카테고리는 기본적인 스택 제어 및 저장 장소에 따른 push/load 와 pop/store 연산 코드를 포함한다. Arithmetic operations 카테고리는 수학적 연산과 비교 연산, 비트 연산, 그리고 논리 연산에 관련된 연산 코드를 포함한다. Flow control operations 카테고리는 branch와 메소드 호출에 관한 연산 코드를 포함한다. Type conversion operations 카테고리는 스택 탑에 있는 데이터의 타입을 변환하는 연산 코드를 포함한다. Object operations 카테고리는 객체 생성과 필드에 관련된 연산 코드를 포함한다. 그 이외의 연산 코드는 Other operations 카테고리에 포함된다[그림3].

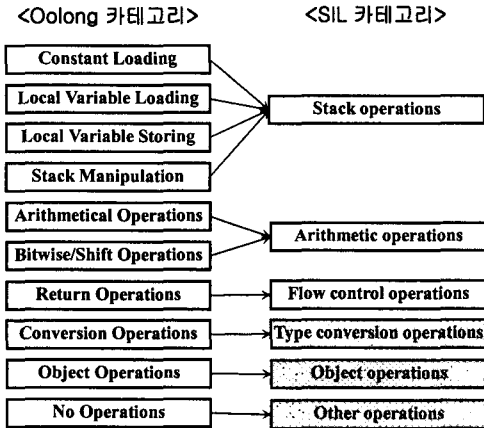


[그림3] SIL 연산 코드 카테고리

4. SIL 카테고리 매핑

4.1 SIL 카테고리에 대한 Oolong 카테고리 매핑

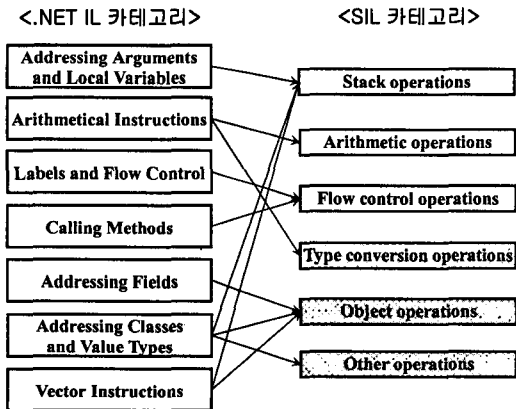
Oolong 의 카테고리는 메인 카테고리 수준에서 SIL 카테고리에 명백하게 매핑이 이루어진다. 스택에서 값을 가져오거나 스택 탑에 특정 값을 저장하는 코드들의 카테고리는 SIL의 Stack operations 카테고리에 매핑된다. 수학적 연산 및 비트 연산, 논리 연산 등의 카테고리는 SIL의 Arithmetic operations 카테고리에 매핑된다. 그 이외의 카테고리는 모두 각각 해당하는 SIL 카테고리에 1대1 매핑이 이루어진다.



[그림4] Oolong에서 SIL로의 매핑

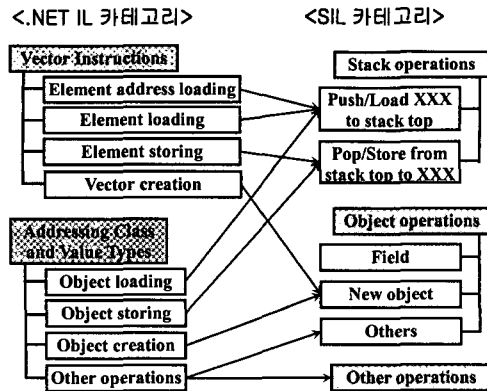
4.2 SIL 카테고리에 대한 .NET IL 카테고리 매핑

.NET IL 카테고리에서 SIL 카테고리의 매핑은 Oolong의 경우 보다 좀더 복잡하다. 메인 카테고리 수준에서의 매핑은 1대 1 매핑뿐만 아니라 1대 다의 매핑이 발생한다[그림5]. 따라서, 서브 카테고리 수준의 매핑을 통한 설명이 필요하다.



[그림5] .NET IL에서 SIL로의 매핑

서브 카테고리 수준에서는 .NET IL의 카테고리가 SIL 카테고리에 명백하게 매핑된다. 메인 카테고리 매핑에서 가장 복잡한 매핑을 나타낸 두 카테고리의 서브 카테고리 매핑을 [그림6]에 나타내었다.



[그림6] .NET IL에서 SIL로의 서브 카테고리 매핑

5. 결론 및 향후 연구

SIL은 EVM의 가상기계 코드로서 일반적인 임베디드 시스템을 위한 가상기계의 표준 중간 언어로 설계되었다. 따라서, SIL이 가상기계 코드로서 완벽성을 갖추고 있는 지에 대한 설명이 필요하다.

본 논문에서는 SIL이 임베디드 시스템을 위한 가상기계의 표준 중간 언어가 되기 위해 충족하고 있는 특징을 논하고, 기존의 가상기계 코드들의 카테고리를 SIL 카테고리에 매핑시킴으로써 SIL의 완벽성을 구조적으로 증명하였다. 매핑 대상은 이미 널리 사용되고 있는 가상기계 코드인 Oolong과 .NET IL로 하였다.

SIL은 스택 기반의 명령어 집합이며, 객체지향 언어와 순차적인 언어를 모두 수용하기 위한 연산 코드를 갖고 있다. 또한 언어 및 하드웨어 독립성 등 임베디드 시스템을 위한 가상기계의 표준 중간 언어로서의 요건을 갖추고 있다.

SIL은 6개의 메인 카테고리 및 16개의 서브 카테고리로 분류되며 Oolong과 .NET IL의 카테고리는 각각 해당하는 SIL 카테고리에 매핑된다. 본 논문에서는 부분적인 설명만을 보여주고 있으나 연구 과정에서 이미 Oolong과 .NET IL 전체에 대한 설명을 수행하였다. 따라서, SIL은 임베디드 시스템을 위한 가상기계의 표준 중간 언어로서 충분히 그 역할을 할 수 있다고 판단된다.

향후 과제로는 SIL 연산 코드의 보완 및 SIL코드의 실행 파일 포맷에 대한 연구가 진행되어야 하고 또한, SIL 코드를 실행할 수 있는 가상기계 구현에 대한 연구가 필요하다.

◆참고 문헌

- [1] Bill Venners, Inside the JAVA Virtual Machine Second Edition, McGraw-Hill, Dec. 1998.
- [2] Common Language Infrastructure, Microsoft Corporation, Dec. 2001.
- [3] C# Language Specification, Microsoft Corporation, Nov. 2000.
- [4] John Gough, Compiling for the .NET Common Language Runtime(CLR), Prentice Hall, 2002.
- [5] Joshua Engel, Programming for the Java Virtual Machine, Addison Wesley, Jan. 1999.
- [6] MSIL Instruction Set Specification, Microsoft Corporation, Nov. 2000.
- [7] Serge Lidin, Inside Microsoft .NET IL Assembler, Microsoft Press, 2002.
- [8] 남동근, 윤성림, 오세만, "가상기계의 어셈블리 언어", 정보처리학회 춘계학술발표논문집(중), 제 10 권 제 1 호, pp.783-786, 2003.
- [9] 민정현, 매핑 테이블을 이용한 자바 바이트코드에서 MSIL로의 번역기, 동국대학교 석사학위 논문, 2001.