

EVM 파일 포맷을 위한 시각화 브라우저

지정환^o, 오세만
동국대학교 컴퓨터공학과
e-mail : {jhjee, smoh}@dongguk.edu

Visualized Browser for EVM File Format

Jung-Hwan Jee^o, Se-Man Oh
Dept. of Computer Engineering, Dongguk University

요 약

임베디드 시스템을 위한 가상 기계 기술은 모바일 디바이스와 디지털 TV 등에 탑재할 수 있는 핵심 기술로 다운로드 솔루션에서는 꼭 필요한 소프트웨어 기술이다. 현재 EVM(Embedded Virtual Machine)이라 명명되어진 임베디드 시스템을 위한 가상 기계에 대한 연구가 진행 중이며, 임베디드 시스템을 위한 실행파일 포맷인 EVM 파일 포맷이 설계 되었다. EVM 파일 포맷은 이진 스트림 형태로 구성되어 있기 때문에, EVM 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계를 가지고 있다.

본 논문에서는 클래스 파일과 PE 파일 등 기존의 가상 기계를 위한 파일들에 대한 시각화 브라우저들의 분석을 기반으로 하여 EVM 파일 포맷을 위한 시각화 브라우저를 설계하고 구현한다. EVM 파일 포맷을 위한 시각화 브라우저를 통해서 EVM 파일에 대한 분석을 보다 편리하게 진행하고 EVM 파일이 갖는 정보에 대한 접근을 용이하게 할 수 있다.

1. 서론

가상 기계는 하드웨어가 아닌 소프트웨어로 제작되어 논리적인 시스템 구성을 갖는 개념적인 프로세서이다. 따라서 가상 기계는 실행 환경인 하드웨어와 운영체제에 종속적이지 않고 플랫폼 독립을 가능하게 한다. 대표적인 가상 기계로는 자바 클래스 파일을 입력으로 받아 실행하는 JVM(Java Virtual Machine)이 있다. 최근에는 모바일 장치에 이식 가능한 GVM, KVM 등의 가상 기계들이 개발 되면서 그 중요성이 더욱 부각되고 있다.

임베디드 시스템이란 전용 동작을 수행하거나 또는 특정 임베디드 소프트웨어 응용 프로그램과 함께 사용되도록 디자인된 특정 컴퓨터 시스템 또는 컴퓨팅 장치를 말한다. 임베디드 시스템을 위한 가상 기계 기술은 모바일 디바이스와 디지털 TV 등에 탑재할 수 있는 핵심 기술로 다운로드 솔루션에서는 꼭 필요한 소프트웨어 기술이다.

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

이러한 가상 기계는 입력으로 받는 실행 파일이 있다. JVM의 클래스 파일, .NET CLR의 PE 파일, GVM의 SGS 파일, EVM의 EVM 파일 등이 있다.

현재 Microsoft의 C# 언어와 SUN사의 자바 언어 등을 모두 수용할 수 있는 가상 기계에 대한 연구가 진행중이다. EVM(Embedded Virtual Machine)이라 명명된 이 가상기계 솔루션은 C#과 자바 등 객체지향 언어뿐만 아니라 C 언어와 같이 순차적인 언어로 작성된 프로그램들을 가상 기계를 위한 코드인 *.sil로 변환한다. 그리고 *.sil을 EVM 파일 포맷으로 변환하여 임베디드 시스템에 탑재된 가상기계에서 실행할 수 있도록 한다.

이진 스트림 형태로 구성된 EVM 파일은 EVM에서 올바른 수행을 위한 많은 정보를 가지고 있다. 하지만 EVM 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계를 가지고 있다.

본 논문에서는 클래스 파일과 PE 파일 등 기존의 가상 기계를 위한 파일들에 대한 시각화 브라우저들의 분석을 기반으로 하여 EVM 파일 포맷을 위한 시각화 브라우저를 설계하고 구현한다.

본 논문의 2 장에서는 기존의 가상 기계를 위한 파

일들에 대한 시각화 브라우저들과 EVM 에 대한 관련 연구를 소개한다. 3 장에서는 EVM 파일 포맷을 위한 시각화 브라우저의 설계 과정 및 실험 결과를 기술한다. 마지막으로 4 장에서는 본 연구의 결론과 향후 연구 과제에 대해서 기술한다.

2. 관련 연구

2.1 시각화 브라우저

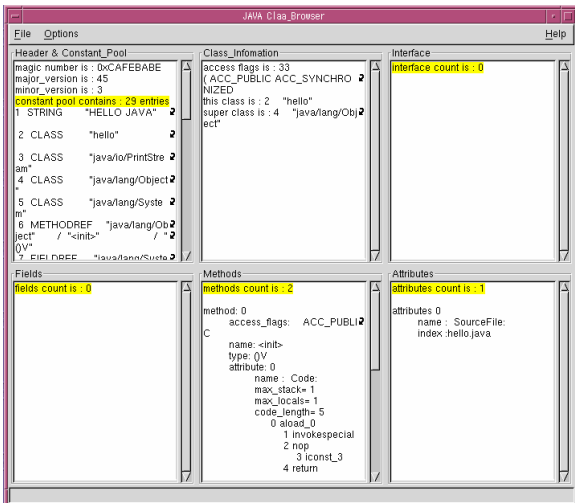
가상 기계가 입력으로 받는 실행 파일은 이진 스트림 형태로 되어있기 때문에 실행 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계가 있다. 시각화 브라우저는 이 실행파일을 입력으로 받아 이에 대한 정보를 사용자가 보다 편리하게 접근 할 수 있도록 시각화 된 형태로 보여준다.

기존의 실행 파일을 위한 시각화 브라우저로는 클래스파일을 위한 시각화 브라우저와 PE 파일을 위한 시각화 브라우저 등이 있다.

2.1.1 클래스 파일 시각화 브라우저

클래스 파일은 자바 소스파일을 입력으로 받은 자바 컴파일러에 의해 생성된다. 확장자가 *.class 이며 자바 가상 기계의 입력이 되어 실행된다. 클래스 파일은 8 비트 단위의 스트림으로 이루어져 있다. 16 비트, 32 비트, 64 비트 크기를 가진 데이터들은 8 비트 단위로 나누어져 높은 비트가 먼저 나오는 빅 엔디언 (Big-Endian)의 순서로 저장된다.

[그림 1]은 클래스 파일에 대한 시각화 브라우저의 실행 화면이다.



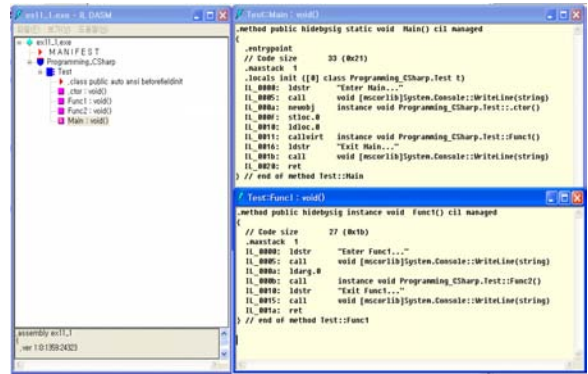
[그림 1] 클래스 파일에 대한 시각화 브라우저

클래스 파일에 대한 시각화 브라우저는 *.class 파일을 입력으로 받아 순차적으로 Constant_Pool, Class_File_Information, Interfaces, Fields, Methods,

Attributes 와 같이 총 6 개의 부분으로 분리하여 각각에 대한 정보를 출력한다.

2.1.2 .NET PE 파일 시각화 브라우저

.NET 실행 파일은 마이크로소프트 PE(Portable Executable)와 COFF(Common Object File Format)의 기준을 토대로 확장한 형태로써, CLR(Common Language Runtime) 환경에서 동작하기 위한 메타데이터와 IL 을 포함하고 있다. .NET 실행 파일은 기존에 실행 파일과 동일하게 확장자가 *.exe 또는 *.dll 이며 스몰 엔디언 (Small-Endian)의 순서로 저장된다. [그림 2]는 Visual Studio .NET 에서 제공하는 IL Disassembler 의 실행 화면이다.

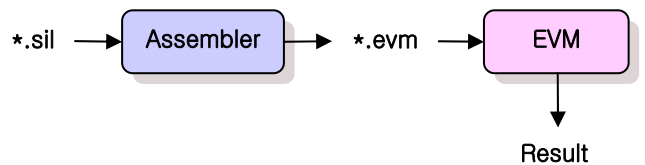


[그림 2] .NET ILdasm

.NET ILdasm 은 메타데이터의 일종인 목록과 클래스, 메소드, 필드 등의 정보를 트리 형태로 보여준다. 그리고 각각에 대해 원하는 정보를 새로운 창에 출력한다.

2.2 EVM 파일 포맷

EVM 을 위한 실행 파일 형식인 EVM 파일 포맷 (*.evm)은 가상 기계를 위한 표준 중간언어(*.sil)를 입력으로 받는 assembler 의 출력이다. 이는 다시 EVM 에 입력이 되어 실행 결과를 만들어 낸다. [그림 3]은 *.sil 이 assembler 를 통해서 *.evm 을 생성하고, *.evm 이 가상기계를 거쳐서 결과를 출력하는 과정을 보여준다.



[그림 3] EVM 구조도

EVM 파일 포맷은 가상 기계인 EVM 을 위한 실행 파일 형식이다. EVM 파일 포맷은 언어간에 통합을 기반으로 설계 되었으며 구조가 간결하고 확장이 용이하다. 또한 메타데이터와 중간언어(SIL)가 분리되어

있어 파일 분석이 쉽고 타입 체크가 편리한 구조이다.
 [표 1] EVM 파일의 구조를 C 언어의 구조체 형태로 나타낸 것이다.

[표 1] EVM 파일의 구조

```

EvmFile {
    u4 magic;
    u2 majorVersion;
    u2 minorVersion;
    u2 module;
    u2 language;
    u2 entryPoint;
    MetadataInfo MDT[];
    SilInfo SIL[];
}
    
```

EVM 파일은 Header, Metadata Table 그리고 SIL 의 세 부분으로 이루어져 있다.

Header 는 파일에 대한 기본 정보를 가진다. [표 1] 에서 magic, majorVersion, minorVersion, module, language, entrypoint 가 Header 부분이다. EVM 파일인지를 식별하기 위한 magic 은 0x0E054DFF(4 바이트)의 값을 가진다. majorVersion 과 minorVersion 은 EVM 파일의 버전을 나타낸다. Module 은 소스 프로그램의 이름을 나타내고, Metadata Table index 로 표현되며 참조 테이블은 string table 이다. language 는 소스 프로그램의 언어가 무엇인지를 나타내고, entryPoint 는 프로그램의 시작 메소드를 나타낸다.

Metadata Table 은 클래스와 클래스 멤버들의 속성, 관계들을 나타낸다. 구성은 tag 와 각 tag 의 속성들로 이루어져있다. [표 2]는 Metadata Table 의 tag 의 종류와 값을 나타낸 것이다.

[표 2] Metadata Table Tag 의 종류

Tag	값	Tag	값
MDTRefClass	0x01	MDTString	0x40
MDTDefClass	0x03	MDTUserString	0x45
MDTNestedClass	0x05	MDTDescriptor	0x30
MDTInterface	0x06	MDTInteger	0x53
MDTField	0x10	MDTFloat	0x54
MDTMethod	0x20	MDTLong	0x55
MDTException	0xe0	MDTDouble	0x56

MDTRefClass 는 참조하는 클래스의 정보를 저장하며, MDTDefClass 는 정의된 클래스의 정보를 저장한다. MDTNestedClass 는 중첩되는 외부와 내부 클래스를 저장한다. MDTInterface 는 인터페이스와 이를 상속하는 클래스 또는 인터페이스를 저장한다. MDTField 와 MDTMethod 는 필드와 메소드의 정보를 저장한다. MDTException 은 예외 처리에 관한 정보를 저장한다. MDTString 은 스트링을 저장하고, MDTUserString 은 사용자가 정의하는 스트링을 저장한다. MDTDescriptor 는 필드의 타입과 메소드의 시그네춰 등을 저장한다. MDTInteger, MDTFloat, MDTLong, 그리고 MDTDouble 은 각 타입에 대한 값을 저장한다.

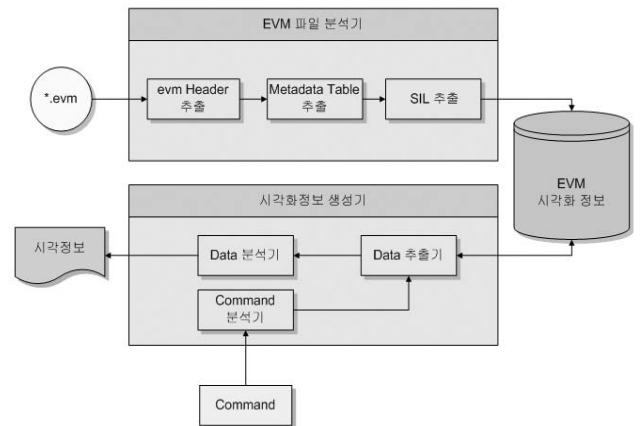
마지막으로 SIL 은 실행을 위한 명령어들의 집합이

다. 구성은 SIL 임을 나타내는 tag 와 SIL 의 레코드 개수인 silsCount, 그리고 silsCount 개수 만큼의 MethodTableIndex 와 해당 바이트로 이루어져있다.

3. EVM 시각화 브라우저

3.1 시스템 구성

EVM 시각화 브라우저는 확장자가 *.evm 인 파일을 입력으로 받아서 사용자가 원하는 정보를 시각화하여 출력해준다. [그림 4]는 EVM 시각화 브라우저를 위한 시스템 구성도이다.

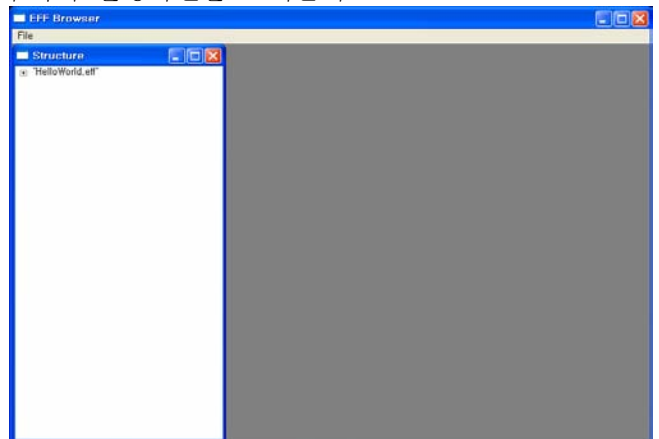


[그림 4] 시스템 구성도

*.evm 을 입력으로 받아서 evm Header, Metadata Table, SIL 에 대한 정보를 각각 추출하여 내부적으로 정의된 자료 구조에 저장한다. 사용자의 command 를 입력으로 받으면 command 를 분석하고, 그에 해당하는 data 를 추출한다. 그리고 추출된 data 를 분석하여 이에 해당하는 시각화 정보를 출력해준다.

3.2 인터페이스

EVM 시각화 브라우저는 윈도우 환경에서 동작하도록 프로그램 되었다. [그림 5]는 EVM 시각화 브라우저의 실행화면을 보여준다.



[그림 5] EVM 시각화 브라우저 실행화면

위의 그림에서와 같이 시각화 브라우저는 크게 두 부분으로 나누어진다. 왼쪽에 위치한 **Structure** 창은 프로그램의 정보, 클래스, 인터페이스, 클래스의 멤버 등을 트리 구조 형태로 보여준다. 오른쪽에 위치한 창은 트리 구조 형태로 나타난 각각의 요소에 대해서 보다 자세한 정보를 보여준다.

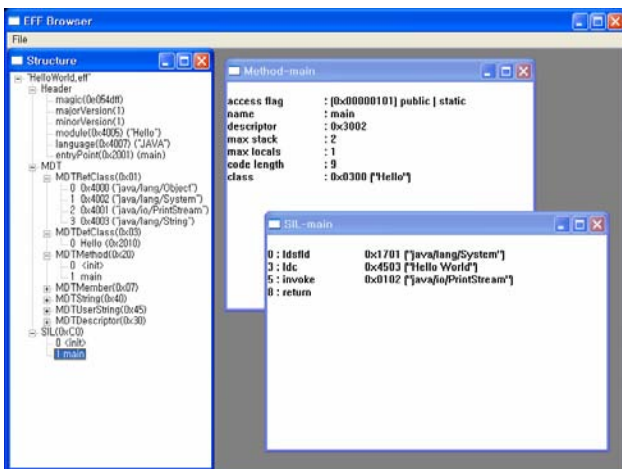
3.3 실험 및 분석

자바 언어로 작성된 프로그램을 *.evm 으로 변환하여 EVM 브라우저를 실행하였다. [표 3]은 실험을 위한 자바 예제 프로그램을 나타낸다.

[표 3] 자바 예제 프로그램

```
public class Hello {
    public static void main(String args[]) {
        System.Out.println( "Hello World" );
    }
}
```

위의 예제 프로그램에 대한 *.evm 을 입력으로 실행된 EVM 시각화 브라우저의 결과 화면은 아래의 [그림 6]과 같다.



[그림 6] 결과 화면

자바 예제 프로그램에 대한 EVM 파일을 읽어 들여 EVM 파일의 구조를 분석한 결과를 Structure 창에 트리 구조로 나타내었다. 전체 구조는 크게 evm Header, Metadata Table, SIL 의 세 부분으로 나누어진다. 트리 구조의 항목을 선택하면, 선택된 항목에 대한 상세 정보를 새로운 창을 생성하여 출력해준다.

위의 결과 화면에서 첫 번째 창은 Metadata Table 에 있는 main 메소드의 정보를 출력해준다. 여기에서는 access flag, name, descriptor, max stack, max locals, code length, class 에 대한 정보를 출력해준다. 두 번째 창에서는 SIL 에서 main 메소드에 대한 SIL 코드를 출력해준다.

4. 결론 및 향후 연구

가상 기계 기술은 실행 환경인 하드웨어와 운영체제에 종속적이지 않고 플랫폼 독립을 가능하게 해주는 기술이다. 특히, 임베디드 시스템을 위한 가상기계 기술은 모바일 디바이스와 디지털 TV 등에 탑재되는 다운로드 솔루션에 꼭 필요한 소프트웨어 기술이다. 이러한 임베디드 시스템을 위한 실행 파일 포맷인 EVM 파일 포맷은 EVM 에서 올바른 수행을 위한 많은 정보를 가지고 있다. 하지만 EVM 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계를 가지고 있다.

본 논문에서는 클래스 파일과 PE 파일 등 기존의 가상 기계를 위한 파일들에 대한 시각화 브라우저들의 분석을 기반으로 하여 EVM 파일 포맷을 위한 시각화 브라우저를 설계하고 구현하였다. EVM 시각화 브라우저는 EVM 파일로부터 사용자가 원하는 정보를 추출하여 시각적으로 출력해준다. 사용자는 이러한 시각화된 정보를 통해서 EVM 파일에 대한 접근을 보다 편리하게 하고, 효과적인 분석이 가능하다.

향후 연구 과제로는 EVM 파일을 보다 시각적으로 표현하기 위한 보완 연구와 EVM 시각화 브라우저를 통한 EVM 파일 포맷 검증에 대한 연구가 필요하다.

참고문헌

- [1] David Platt, *Introducing Microsoft .NET 2nd Edition*, Microsoft Press, 2002.
- [2] Joshua Engel, *Programming for the Java Virtual Machine*, Addison Wesley, 2000.
- [3] Kevin Burton, *.NET Common Language Runtime*, SAMS, 2002.
- [4] Thai & lam, *.NET Framework Essentials*, O'REILLY, 2002.
- [5] 오세만, *컴파일러 입문 개정판*, 정익사, 2000.
- [6] 윤영수, 박상필, 고평만, "자바 클래스 파일에 대한 시각화 브라우저의 설계 및 구현", 한국정보과학회 춘계학술발표논문집, 제 30 권 제 2 호, pp.157~159, 2003.
- [7] 정한중, 윤성림, 오세만, "가상기계를 위한 실행 파일 포맷", 한국정보처리학회 추계학술발표대회 논문집 (중), 제 10 권 제 2 호, pp.647~650, 2003.