

## EVM 파일 포맷을 위한 시각화 브라우저의 구현

지정환<sup>0</sup>, 오세만  
동국대학교 컴퓨터 공학과  
{jhjee<sup>0</sup>, smoh}@dongguk.edu

### Implementation of a Visualized Browser for EVM File Format

Junghwan Jee<sup>0</sup>, Seman Oh  
Dept. of Computer Engineering, Dongguk University

#### 요 약

임베디드 시스템을 위한 가상기계 기술은 모바일 디바이스와 디지털 TV 등에 탑재되어 다운로드 솔루션에 꼭 필요한 소프트웨어 기술이다. 현재 EVM(Embedded Virtual Machine)이라 명명되어진 임베디드 시스템을 위한 가상기계에 대한 연구가 진행 중이며, 임베디드 시스템을 위한 실행 파일 포맷인 EVM 파일 포맷이 설계되었다. EVM 파일 포맷은 이진 스트림 형태로 구성되어 있기 때문에, EVM 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계를 가지고 있다.

본 논문에서는 기존의 클래스 파일에 대한 시각화 브라우저와 클래스 파일 실행 분석기의 분석을 기반으로 하여 EVM 파일 포맷을 위한 시각화 브라우저를 설계하고 구현한다. 기존의 텍스트 중심의 브라우저와는 다르게 정보를 이해하기 쉽게 시각화하여 표현하였다. 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. 서 론

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 개발되면서 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.

이러한 가상 기계는 입력으로 받는 실행 파일이 있으며 JVM의 클래스 파일, .NET CLR의 PE 파일, EVM의 EVM 파일 등이 이에 해당한다.

현재 Microsoft의 C# 언어와 SUN 사의 자바 언어 등을 모두 수용할 수 있는 가상 기계에 대한 연구가 진행 중이다[4]. EVM(Embedded Virtual Machine)이라 명명된 이 가상기계 솔루션은 C#과 자바 등 객체지향 언어뿐만 아니라 C언어와 같이 순차적인 언어로 작성된

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

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.

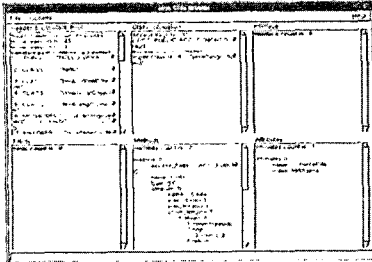
본 논문에서는 기존의 클래스 파일에 대한 시각화 브라우저와 클래스 파일 실행 분석기의 분석을 기반으로 하여 EVM 파일 포맷을 위한 시각화 브라우저를 설계하고 구현한다.

#### 2. 배경 연구

##### 2.1 클래스 파일 시각화 브라우저

클래스 파일 시각화 브라우저는 자바의 실행파일인 클래스 파일을 입력으로 받아 이에 대한 정보를 사용자가 편리하게 접근할 수 있도록 시각화된 형태로 보여준다.

클래스파일은 자바 소스파일을 입력으로 받은 컴파일러에 의해 생성된다. 확장자가 \*.class이며 자바 가상기계의 입력이 되어 실행된다. 클래스 파일은 8비트 단위의 스트림으로 이루어져 있다. 16비트, 32비트, 64비트 크기를 가진 데이터들은 8비트 단위로 나누어져 높은 비트가 먼저 나오는 빅 엔디언(Big-Endian)의 순서로 저장된다. [그림 1]은 클래스 파일에 대한 시각화 브라우저의 실행 화면이다.

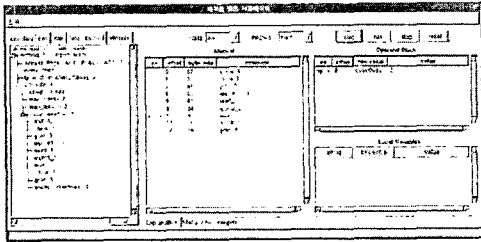


[그림 1] 클래스 파일에 대한 시각화 브라우저

클래스 파일에 대한 시각화 브라우저는 \*.class 파일을 입력으로 받아 순차적으로 Constant Pool, Class File Information, Interfaces, Fields, Methods, Attributes와 같이 총 6개의 부분으로 분리하여 각각에 대한 정보를 출력한다[6].

### 2.2 클래스 파일 실행 분석기

클래스 파일 실행 분석기는 클래스 파일에 대한 분석 및 실제 실행 과정을 보다 시각적으로 표현한다. [그림 2]는 클래스파일 실행 분석기의 실행 화면이다.



[그림 2] 클래스 파일 실행 분석기

\*.class 파일을 입력으로 받으면 Constant Pool, Class file info, Interface, Field, Method, Attribute의 6개의 단계로 나누어서 차례대로 그 내용을 분석한다. 그리고 분석된 내용 중에서 자바 가상기계의 동작 과정 분석을 위해서 필요한 Method 부분을 따로 읽어 들여 Local Variable과 Stack 영역으로 나누어 Local Variable 영역 정보와 Stack 영역 정보에 어떠한 내용들이 어떻게 변화되는지에 대해 분석하게 된다[3].

### 2.3 EVM 파일 포맷

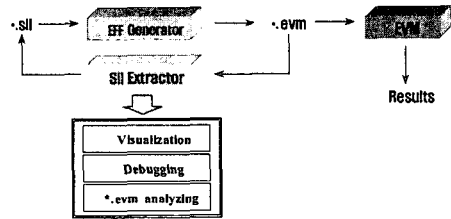
EVM 파일 포맷은 가상 기계인 EVM을 위한 실행 파일 형식이다. EVM 파일 포맷은 언어간의 통합을 기반으로 설계되었으며 구조가 간결하고 확장이 용이하다. 또한 메타데이터와 중간언어(SIL)가 분리되어있어 파일 분석이 용이하고 타입 체크가 편리한 구조이다.

EVM 파일은 Header, Metadata Table 그리고 SIL(Standard Intermediate Language)의 세 부분으로 이루어져있다. Header는 파일에 대한 기본 정보를 나타내며, Metadata Table은 클래스와 클래스 멤버들의 속성, 관계들을 나타낸다. 그리고 SIL은 실행을 위한 명령어들의 집합이다[7].

## 3. 시각화 브라우저의 구현

### 3.1 EVM 시각화 브라우저

EVM을 위한 실행 파일 형식인 EVM 파일(\*.evm)은 가상 기계를 위한 표준 중간언어(\*.sil)를 입력으로 받는 EFF Generator의 출력이다. 이는 다시 EVM의 입력이 되어 실행 결과를 만들어낸다. [그림 3]은 \*.sil이 EFF Generator를 통해서 EVM 파일을 생성하고, EVM 파일이 가상기계를 거쳐서 결과를 출력하는 과정을 보여준다[4].

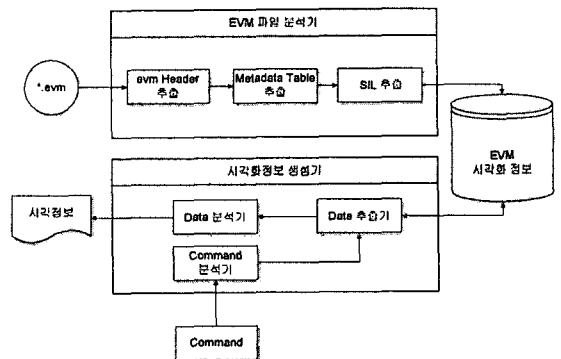


[그림 3] EVM 구조도

[그림 3]에 도시된 Sil Extractor는 EVM 파일 (\*.evm)에서 표준 중간 언어(\*.sil)을 추출하는 역할을 담당한다. 이 과정에서 EVM 파일을 분석하여 시각화하고, 또한 디버깅 과정도 함께 수행한다.

### 3.2 시스템 구성

EVM 파일 포맷을 위한 시각화 브라우저는 확장자가 \*.evm인 EVM 파일을 입력으로 받아서 사용자가 원하는 정보를 시각화하여 출력해 준다. [그림 4]는 EVM 파일 포맷을 위한 시각화 브라우저의 시스템 구성도이다.



[그림 4] 시스템 구성도

EVM 파일 포맷을 위한 시각화 브라우저는 크게 EVM 파일 분석기와 시각화 정보 생성기 두 부분으로 나뉘어진다.

### 3.3 EVM 파일 분석기

EVM 파일은 evm Header, Metadata Table 그리고 SIL의 세 부분으로 이루어져 있다. EVM 파일 분석기는 EVM

파일(\*.evm)을 입력으로 받아서 evm Header, Metadata Table, SIL에 대한 정보를 각각 추출한다. 그리고 추출된 시각화 정보들을 내부적으로 정의된 자료 구조에 저장한다.

### 3.4 시각화 정보 생성기

시각화 정보 생성기는 사용자의 명령을 입력으로 받아서 사용자가 원하는 정보를 시각화하여 출력한다. 사용자의 명령은 명령어 분석기에 의해서 분석되고, 데이터 추출기는 필요한 데이터를 저장된 EVM 시각화 정보로부터 추출한다. 데이터 분석기는 추출된 데이터를 분석하여 최종적으로 사용자가 원하는 시각정보를 출력한다.

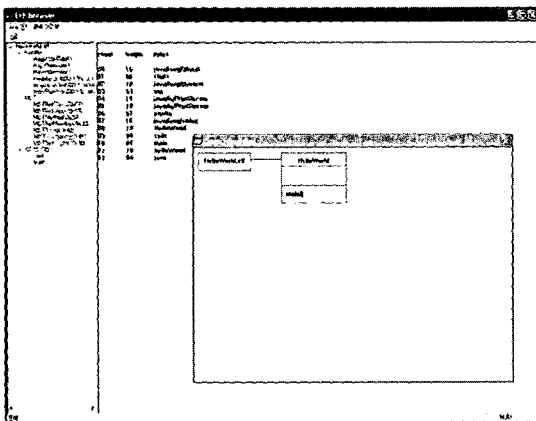
### 3.5 실험 및 분석

자바 언어로 작성된 프로그램을 EFF Generator를 통해서 EVM 파일(\*.evm)을 생성하고, EVM 파일 포맷을 위한 시각화 브라우저를 실행하였다. EVM 시각화 브라우저는 윈도우 환경에서 동작하도록 프로그램을 작성하였다. [표 1]은 실험을 위한 자바 예제 프로그램으로 문자열을 출력하는 프로그램이다.

[표 1] 자바 예제 프로그램

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

위의 자바 예제 프로그램에 대한 시각화 브라우저의 결과 화면은 [그림 5]와 같다.



[그림 5] 시각화 브라우저의 실행 화면

[그림 5]의 시각화 브라우저는 예제 프로그램에 대한 EVM 파일(\*.evm)을 읽어 들어서 구조를 분석하고 저장한다. 그리고 저장된 EVM 파일의 구조를 시각화 브라우저의 왼쪽 부분에 사용자가 편리하게 볼 수 있는 트리 구조 형태로 출력한다. 트리의 각 항목을 클릭하면

선택된 항목의 보다 자세한 정보가 브라우저의 오른쪽 부분에 출력된다. EVM 파일 포맷의 구조를 트리 형태로 표현함으로써 사용자는 생성된 EVM 파일에 보다 편리하게 접근할 수 있다.

클래스 정보 메뉴 내의 Hierarchy 메뉴를 선택하면 EVM 파일 내에 존재하는 클래스들 간의 관계를 보여주는 새로운 차일드 윈도우가 생성된다. 이 차일드 윈도우는 EVM 파일 내의 모든 클래스 상속 관계를 그래프 형태로 출력한다. 이 그래프는 생성된 EVM 파일명을 루트로 하여 EVM 파일 내에 존재하는 모든 클래스들의 상속 관계를 표현한다. 사용자는 현재 EVM 파일 내에 어떤 클래스가 존재하는가를 클래스 상속 그래프를 통하여 알 수 있으며 클래스 간의 상속 관계도 일목요연하게 파악할 수 있다.

## 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 EVM에서 올바른 수행을 위한 많은 정보를 가지고 있다. 하지만 EVM 파일의 형태를 분석하고 각각의 정보를 보다 용이하게 접근하는 데는 한계를 가지고 있다.

본 논문에서는 기존의 클래스 파일에 대한 시각화 브라우저와 클래스 파일 실행 분석기의 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 검증용 연구가 필요하다. 그리고 EVM 파일의 실행 과정을 시각적으로 보여주는 EVM 파일 실행 분석기에 관한 연구가 진행되어야 하겠다.

### ◆참고 문헌

- [1] Joshua Engel, Programming for the Java Virtual Machine, Addison Wesley, 2000.
- [2] Tim Lindholm, The Java Virtual Machine Specification, SUN, 1999.
- [3] 박상필, 고광만, “자바 클래스 파일 실행 분석기”, 한국정보과학회 추계학술발표논문집, 제30권 제2호, pp.262~264, 2003.
- [4] 오세만 외 2인, 임베디드 시스템을 위한 가상기계의 설계 및 구현, 한국과학재단 특정 기초연구, 2차 중간보고서, 2004.6.
- [5] 오세만, 컴파일러 입문 개정판, 정익사, 2004.
- [6] 윤영수, 박상필, 고광만, “자바 클래스 파일에 대한 시각화 브라우저의 설계 및 구현”, 한국정보과학회 추계학술발표논문집, 제30권 제2호, pp.157~159, 2003.
- [7] 정한중, 윤성림, 오세만, “가상기계를 위한 실행 파일 포맷”, 한국정보처리학회 추계학술발표대회 논문집(중), 제 10권 제2호, pp.647~650, 2003.