내장형 자바가상기계를 위한 클래스 이미지 파일의 분석과 비교

김성수°, 김세영, 양희재

경성대학교 컴퓨터공학과

sskim@conet.ks.ac.kr, sykim@conet.ks.ac.kr, hjyang@star.ks.ac.kr

Comparison and Analysis of Class Image File for Embedded Java Virtual Machine

Sungsu Kim°, Seyoung Kim and Heejae Yang

Dept. of Computer Engineering, Kyungsung University

요 약

자바가상기계는 기계독립적인 바이트코드, 즉 자바 컴파일러가 자바 원천코드로 생성한 클래스 파일의 정보를 읽어 응용 프로그램을 실행한다. 클래스 파일의 내부정보는 동적인 클래스 적재를 지원하기 위한 각종 심볼명과 클래스, 상수, 필드, 메소드 등으로 구성되어 있으며 여러 가지 링크 정보와 디버깅 정보로 인해 메모리 낭비와 클래스 파일에 대한 정보를 접근하는데 비효율적인 요소가 많다. 이런 이유로 인해 메모리 사용에 제한을 받는 내장형 시스템 환경에서 동작하는 자바가상기계에서 클래스 파일을 그대로 이용하기에는 적합하지 않다. 따라서 클래스 파일을 다른 새로운 형태인 클래스 이미지 파일로 변환이 요구되는데 이를 지원하는 자바가상기계로 simpleRTJ와 leJOS등이 있다. 클래스 이미지 파일의 지원은 내장형 시스템에서 동작하는 자바가상기계의 메모리 절감과 함께 클래스 정보의 접근 효율성을 높이는데 있다. 본 논문에서는 simpleRTJ 및 leJOS의 클래스 이미지 파일을 분석한 후 자바가상기계의 메모리 절감과 접근의 효율을 높여주기 위한 목적에 의해 클래스 이미지 파일을 생성하는 응용 프로그램인 cls2bin 이라는 클래스 이미지 변환기로 생성한 이미지 파일과 비교하여 cls2bin이 생성한 클래스 이미지 파일이 얼마만큼의 효과를 가지는지에 대해 검증해 보고자 한다.

1. 서 론

현재 무선 휴대용 장치의 급속한 보급과 함께 정보가전에 대한 관심이 높아지고 있으며 이에 따라 내장형 시스템을 제어하기 위한 자바가상기계의 사용 또한 높아지고 있다. 자바가상기계가 내장형 시스템에서 큰 관심을 받는 이유는 플랫폼의 독립성일 것이다. 다양한 프로세스와 운영체제를 가지는 내장형 시스템에서 자바가상기계는 자바 컴파일러가 자바 원천코드로 생성한 기계 독립적인 바이트 코드인 클래스 파일의 정보를 읽어 실행하게 된다.

클래스 파일의 내부정보는 동적인 클래스 적재를 지원하기 위한 각종 심볼명과 클래스, 상수, 필드, 메소드 등으로 구성되어 있으며 여러 가지 링크 정보와 디버깅 정보로 인해 메모리 낭비와 클래스 파일에 대한 정보를 접근하는데 비효율적인 요소가 많다. 이런 이유로 메모리 사용에 제한을 받는 내장형 시스템에서 자바가상기계가 효율적인 동작을 수행하기 위한 방안으로 기존의 클래스 파일을 다른 새로운 형태인 클래스 이미지 파일 형식으로의 변환한 경우가 있으며, 이를 지원하는 자바가상기계로 simpleRTJ와 leJOS가 있다.

클래스 이미지 파일은 동적 클래스 적재에 사용하는 심볼명 대신 필요한 클래스들을 미리 정의하여 메모리 절감을 하고 클래스, 상수, 필드, 메소드 등의 내부정보들을 자바가상기계가 효율적으로 접근할 수 있도록 재배치하여 생성되어 진다. 따라서 제한된 메모리 사용을 요구하는 내장형 시스템에서 자바가상기계는 클래스 이미지 파일을 읽어 응용 프로그램을 실행하도록 하여 보다 효율적인 수행을 보장받게 된다.

본 논문에서는 자바가상기계의 메모리 절감과 접근의 효율을 높여주기 위한 목적에 의해 클래스 이미지 파일을 생성하는 응용 프로그램 cls2bin[1]이 생성한 클래스 이미지 파일을 simpleRTJ와 leJOS가 생성하는 클래스 이미지 파일들과 비교하여 얼마만큼의 효과를 가지는지 알아보았다.

본 논문의 구성은 다음과 같다. 2절에서는 관련연구로 simpleRTJ와 leJOS가 사용하는 클래스 이미지 파일 구조를 알아보고, 3절에서는 cls2bin이 생성한 클래스 이미지 파일의 구조에 대해 간략히 알아 본다. 4절에서는 cls2bin이 생성한 클래스 이미지 파일과 simpleRTJ와 leJOS의 클래스 이미지 파일을 비교 분석하였다. 5절에서는 cls2bin이 생성한 클래스 이미지 파일의 효과에 대한 분석과 함께 결론을 맺는다.

2. 관련연구

현재 클래스 이미지 파일을 지원하는 자바가상기계로 simpleRTJ[2]와 leJOS[3]가 있다. 두 가지 자바가상기계에 의해 사용되는 클래스 이미지 파일은 불필요한 심볼명의 제거로 인한 메모리 절감과 클래스 정보의 효율적인 접근이라는 동일한 목적을 가지고 있으나, 클래스 이미지 파일의 메모리 적재와 클래스 정보들의 접근 방식에 의해 클래스 이미지 파일은 서로 다른 구조를 가진다.

본 절에서는 관련연구로 simpleRTJ와 leJOS의 클래스 파일 구조와 함께 각 클래스 정보들이 어떻게 서로 링크되어 참조되는지를 알아보았다.

2.1 simpleRTJ의 클래스 이미지 파일 구조

simpleRTJ에서는 ClassLinker 라는 자바로 작성된 프로그램이 클래스 파일들을 참조하여 이 클래스의 변형된 형태의 모임인 클래스 이미지 파일, 즉 bin 파일을 생성한다.

bin 파일 내에는 checksum, 메인 메소드의 위치, 클래스의

개수, 프레임 및 인스턴스의 크기 등이 포함된 헤더 부분과 함께 다수개의 클래스들이 포함되어 있다. (그림 1)

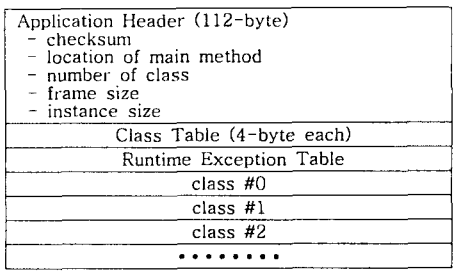

그림 1 simpleRTJ 클래스 이미지 파일 구조

ClassLinker가 생성한 bin 파일은 cls2bin이 생성한 bin 파일에서와 동일하게 클래스들간 링크를 위한 심볼명 대신 필요한 클래스들이 미리 정의되어져 포함하고 있으며, bin 파일의 클래스 내부에는 실행에 필요한 정보들인 클래스 자체 정보, 상수 풀 오프셋 정보, 상수 정보, 필드 정보, 메소드 정보 등으로 이루어져 있다.

simpleRTJ에서는 클래스 정보들을 얻어오기 위해 상수풀 오프셋 테이블(constant poll offset table)을 이용한다. 이것은 클래스, 상수, 필드, 메소드의 네 가지 정보에 공통적으로 사용되며, 각 정보를 요구하는 해당 바이트 코드의 인덱스 값으로 상수풀 오프셋 테이블을 읽어 상수 풀 오프셋 테이블이 가리키는 정보의 번지에서 해당 정보를 참조하게 되며, 그 결과 최소 두 번, 최대 세 번의 접근으로 클래스 정보를 접근이 가능하다.

3.2 leJOS의 클래스 이미지 파일 구조

leJOS에서는 TinyVm이라는 자바로 작성된 프로그램이 클래스 파일들을 참조하여 이 클래스의 변형된 형태의 모임인 클래스 이미지 파일, 즉 bin 파일을 생성한다.

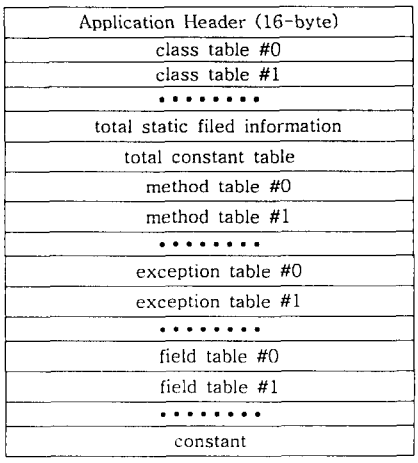

그림 2 leJOS 클래스 이미지 파일 구조

leJOS가 생성한 bin 파일은 simpleRTJ 클래스 이미지 파일 구조와는 다르게 구성되어 있다. 따라서 16바이트로 구성된 헤더부분 이후로, 각 클래스의 자체정보 모임, 전체 static 필드정보 모임, 전체 상수 정보 모임, 각 클래스당 메소드 정보 모임, 각 메소드당 예외 정보 모임, 각 클래스당 필드 정보 모임, 전체 상수 값의 모임으로 구성되어 있다. (그림 2)

클래스 정보의 접근은 ldc, invoke 계열, field의 바이트 코드의 인덱스를 참조하여 각 정보가 위치한 번지를 가지는 오프셋 테이블을 읽고, 오프셋 테이블에서 가리키는 번지로 정보를 접근하게 된다. 따라서 최소 두 번, 최대 세 번의 접근으로 각 정보의 접근이 가능하다.

본 절에서 simpleRTJ와 leJOS가 생성한 클래스 이미지 파일 구조가, 클래스 이미지 파일의 메모리 적재와 클래스 정보들의 접근 방식에 의해 서로 다른 구조를 가지고 있지만 링크에 필요한 심볼명 대신 필요한 클래스들을 미리 포함하여 메모리 절감을 이루고 각 클래스 정보들의 효율적인 접근을 지원한다는 것을 알 수 있다.

다음 절에서는 cls2bin이 생성한 클래스 이미지 파일의 구조와 클래스 정보의 접근에 대해서 알아보고, 4절에서는 본 절에서 관련연구로 알아본 simpleRTJ와 leJOS의 클래스 이미지 파일과 cls2bin이 생성한 클래스 이미지 파일의 성능 비교를 했다.

3. cls2bin이 생성한 클래스 이미지 파일의 구조

cls2bin은(클래스 이미지 변환기) 기존의 클래스 파일들을 읽어 심볼 테이블을 만든 후 클래스 정보의 효율적인 접근을 위해 ldc, invoke 계열, field의 바이트 코드의 인덱스를 변환하고 불필요한 심볼명을 제거한 후 클래스 이미지 파일 구조에 맞게 하나의 bin 파일을 생성한다.

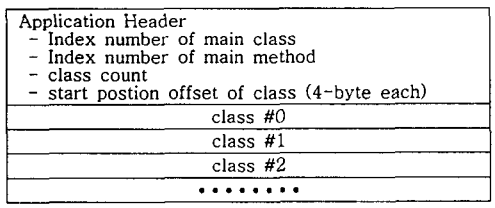

그림 3 cls2bin 클래스 이미지 파일 구조

bin 파일은 헤더부분에 메인 클래스 인덱스, 메인 메소드 인덱스, 클래스 개수, 각 클래스들의 시작 위치들이 포함되어 있으며, 그 외 다수개의 클래스들도 포함되어 있다. (그림 3)

cls2bin이 생성한 bin 파일은 두 가지의 큰 특징을 가진다. ① 정적인 링크 구조를 지원하기 때문에 클래스 간 링크에 관한 심볼명 대신 필요한 클래스들을 미리 정의를 하고 있다. 이것은 메모리 절감이 이루어지는 가장 큰 이유로 자바가상기계는 bin 파일 전체를 메모리에 적재가 가능하다. ② 클래스 구조는 클래스 자체에 대한 정보, 상수에 대한 정보, 필드에 대한 정보, 메소드에 대한 정보 등 실행에 필요한 정보로 구성되어지며 cls2bin은 각 정보들의 링크를 위해 바이트 코드의 인덱스 값을 클래스 정보가 존재하는 실제 위치 값으로 재정의해 놓았다. 따라서 이 값을 참조하여 최소 한 번 최대 두 번의 접

근으로 클래스 정보를 참조가 가능하다.

이 두 가지 특징을 위해 자바가상기계가 클래스 이미지 파일을 미리 분류하여 배열형태로 저장해야 하는 작업이 선행되어야 하지만 메모리의 절감과 클래스에 포함된 각 정보들에 고속의 접근이 가능하다는 장점을 가진다.

4. cls2bin이 생성한 클래스 이미지 파일의 성능평가

본 절에서는 simpleRTJ, leJOS, cls2bin이 생성하는 클래스 이미지 파일의 효율을 비교해 보았다.

우선 simpleRTJ와 leJOS의 응용 프로그램이 공통으로 사용하는 클래스 라이브러리를 조사해본 결과 ①클래스 라이브러리에 포함된 클래스의 절반 이상은 예외 및 오류처리를 위한 것이다. ②클래스 라이브러리 파일의 크기는 평균 902 바이트로 조사 되었다. ③각 클래스 라이브러리는 평균 1.58개의 필드와 7.28개의 메소드를 갖는다. ④각 클래스 라이브러리에서 심볼이 차지하는 비율은 최소 20%, 최대 75% 이며, 평균 53.9%이다[4].

각각의 가상 머신들이 기본적으로 사용하는 라이브러리들의 오차를 제거하기 위해 cls2bin에 leJOS에서 사용하는 기본적인 라이브러리들을 동일하게 적용해 보았다. 실험에 사용한 소스는 기본적인 자료형 연산과 상속관계만을 나타내는 것으로 한정하였다.

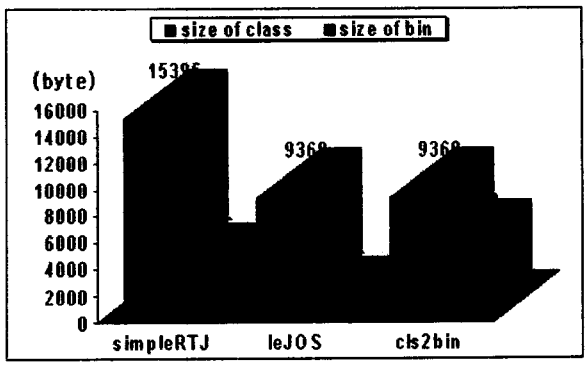

그림 4 클래스 이미지 파일 크기절감 비교

소스의 컴파일에 의해 생성된 클래스들의 크기는 1,280바이트이며, 각 클래스가 참조하는 라이브러리를 포함한 클래스의 전체 크기로 각각 생성한 클래스 이미지 파일 크기를 비교한 결과 simpleRTJ는 76%, leJOS는 87.8%, cls2bin은 42%의 크기절감이 있었다. (그림 4)

각 클래스 이미지 파일들이 동일한 목적과, 각 클래스 이미지 파일에 포함된 정보가 비슷한 형태로 구성되어 있어도 그림 4와 같이 클래스 이미지 파일의 크기절감이 서로 다른 이유는 심볼들의 삭제 이외에도 몇 가지 특징에 의해 차이가 난다.

① 클래스 파일에는 포함되어 있으나 실제 응용프로그램에서 사용되지 않는 메소드나 필드는 이미지 파일에 넣지 않는다. ② 메소드들에 대해서 동일한 크기의 스택, 즉 가장 많은 스택 공간을 요구하는 메소드와 동일하게 스택 공간을 할당하기 때문에 각각의 메소드 정보에서 스택 정보는 빠지게 된다. ③ 상수풀의 정보를 통합하여 관리하며, 동일한 상수 풀의 정보는 공통으로 사용한다.

cls2bin은 아직 위의 특징을 지원하지 않는다. 다만 내부적으로 사용되는 심볼 테이블의 삭제에 의해서 클래스 이미지 파일의 크기 절감 효과만을 가져온다. 하지만 그림 4와 같이 클래스 간 링크에 관한 심볼명 대신 필요한 클래스들을 미리 정의만 하더라도 실제 자바가상기계의 메모리 절감을 약 42% 정도 가져온다는 것을 알 수 있다.

cls2bin이 생성한 클래스 이미지 파일은 정적인 클래스 지원을 위해 상수 풀의 클래스 링크와 관련된 심볼명을 제거한 대신 필요한 클래스들은 미리 정의하고 있으며, 클래스 정보의 효율적인 접근을 위해 실제 클래스 정보의 위치 값을 바이트 코드의 인덱스에 재정의 하고 있다. 따라서 제한된 자원을 가지는 내장형 시스템에서 메모리를 절감하고, 프로세스로 하여금 클래스에 대한 정보를 한번 또는 두 번만으로, 고속의 접근이 가능하게 하는 효과를 가진다.

현재 cls2bin이 생성한 클래스 이미지 파일에는 클래스에서 사용되지 않는 메소드와 필드를 제외시키지 않고 그대로 포함하고 있다. 따라서 메모리 효율이 떨어질 가능성이 있다는 단점을 가진다. 그리고 클래스의 검증과정에 필요한 정보도 포함하지 않고 있어 잘못된 정보에 대해 검증을 하지 못해 자바가상기계가 동작하는 응용 프로그램이 치명적 수행오류를 발생할 수 있다는 단점을 가지고 있다[5].

5. 결 론

향후 cls2bin이 생성한 클래스 이미지 파일의 구조에는 최대한 메모리를 절감하기 위해 클래스에서 사용하지 않는 메소드와 필드를 제외시킬 것 이며, 중복된 바이트 코드들을 보다 작은 크기로 변환하여 생성시킬 것 이다. 또한 잘못된 정보에 의한 응용 프로그램의 치명적 수행오류를 제거하기 위해 검증과정에서 필요한 정보 또한 포함시킬 것 이다.

향후 연구는 클래스 이미지 파일을 자바가상기계에서 동작시키고, 클래스 정보에 대한 접근의 효율에 대해 simpleRTJ 및 leJOS와 정량적인 분석을 시행할 것이다. 그 결과를 토대로 메모리 사용면과 접근의 효율면에서 더욱 효과적인 클래스 이미지 파일 구조와 자원의 제약을 받는 내장형 시스템에서 동작하는 자바가상기계의 성능향상에 대해 연구할 계획이다.

참 고 문 헌

[1] 지정훈, 양희재, "내장형 자바가상기계를 위한 클래스 파일 변환기의 설계 및 구현", *한국정보과학회*, 제30회 춘계학술 발표회 발표예정, 2003.

[2] 양희재, "simpleRTJ 자바가상기계에서 클래스 파일의 메모리상 배치", *한국정보과학회*, 제29회 추계 학술대회, 2002. 10.

[3] R. Schiedermeier, An Introduction to leJOS, Munich University of Applied Sciences, http://www.informatik.fh-muenchen.de/~schieder/usinglejos/

[4] 양희재, "simpleRTJ 클래스 파일의 형식 분석", *한국해양정보통신 학술대회*, 6권, 2호, pp.373-377, 2002. 11.

[5] J. Engel, *Programming for the java Virtual Machine*, Addison-Wesley, 1999.

[6] T. Lindholm and F. Yellin, *The Java Virtual Machine Specification Second Edition*, Addison-Wesley, 1999.