

# 인터넷 가전용 iRTOS™ 실시간 운영체제와 임베디드 용 자바 가상 머신의 통합 \*

°유지영, 전문광, 이철춘

충남대학교 컴퓨터공학과

{jyyoo, mgjeon, chlee}@ce.cun.ac.kr

## Integration of the Embedded Java Virtual Machine and the iRTOS™ Real-time Operating System for Internet appliance

Ji-Young Yoo°, Mun-Gwang Jeon, Cheol-Hoon Lee

Dept. of Computer Engineering, Chungnam National Univ.

### 요 약

본 논문에서는 임베디드 자바 가상 머신(Embedded Java Virtual Machine)과 임베디드 실시간 운영체제(Embedded Real-Time Operating System)인 iRTOS™를 통합하기 위한, 설계 및 구현 내용을 기술하고자 한다. 인터넷 가전을 위한 실시간 운영체제 기반의 통합 환경은 각 가정에서 사용하는 정보 가전 제품들을 인터넷에 연결하여 사용함으로써 사용자에게 좀 더 효율적이며, 편리한 서비스를 제공할 수 있다. 또한 어떠한 플랫폼에도 구애 받지 않고 동작 할 수 있는 자바를 실행하기 위한 내장형 JVM 의 추가로, 기존의 플랫폼 의존적인 방법을 벗어나 어떠한 가전 기기상에서도 자바로 개발된 모든 기능들을 동일한 조건으로 동작시킬 수 있다는 장점을 갖는다. 본 논문에서는 통합 설계서의 세부 내용과 통합 방법에 대해 언급하고자 한다.

### 1. 서 론

연결성이 있으며 지능적이고 개인화 되어 진 정보 가전 기기들은 점점 더 우리의 생활과 비즈니스에서 중요시 되고 있다. 2000 년 전세계 인터넷 이용자가 3 억 명에 이르고, 2005 년에는 10 억 명으로 증가할 것으로 전망되는 지금, 현재까지의 가전 기기들과는 다르게 앞으로는 모든 가정 용 정보 기기들이 유무선 인터넷으로 연결되는 시대가 도래할 것이다.

인터넷 가전기기들은 과거에 알려져 왔던 일반적인 목적을 가진 컴퓨팅 기계들과는 달리 특정한 목적을 갖는 제한된 기능의 장치들로 가는 경향이 있다. 과거의 홈 가전 기기들은 주로 기계의 제어에 초점을 맞추고 있었으나, 점차 기기들을 기능화 할 필요가 생기면서 가전 기기에 여러 서비스 추가하는 방향으로 변화하고 있다[4].

이들 인터넷 가전 기기들은 매우 다양한 플랫폼을 지니고 있어, 온라인 서비스를 제공할 때, 서비스 제공자는 각각의 플랫폼에 맞는 어플리케이션을 작성해

야 한다. 이 때, 같은 기능을 하는 어플리케이션이라 하더라도 다양한 플랫폼으로 인해, 여러 플랫폼에 해당하는 같은 기능의 어플리케이션들을 작성해야 하는 어려움이 생긴다. 이것을 보완해 줄 수 있는 기술이 바로 플랫폼에 독립적으로 실행되는 자바이다. 자바가 수행되어 지기 위해서는, 자바 수행 환경인 자바 가상 머신(Java Virtual Machine : 이하 JVM)을 인터넷 가전 기기의 운영체제에 추가해야만 한다.

이와 같이 본 연구는 인터넷 정보 가전기기들을 Internet 에 연결시켜 유용하게 사용하도록 할 수 있는 핵심 기술인 자바에 실행 환경을 제공하는 JVM 과, 가정용 기기들을 사용할 수 있도록 해주는 임베디드 시스템용 실시간 운영체제를 통합하려 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 iRTOS™의 특징과 JVM 을 통합하는 이유와 임베디드 자바 가상 머신에 대해 기술하고, 3 장에서는 두 가지를 통합하기 위한 전체적인 설계도와 통합 과정을 설명하였다. 마지막으로, 4 장에서는 결론과 향후 연구과제를 기술

\* 이 논문은 BK21 대전, 충남 정보통신인력양성 사업단의 RA 연구비 지원에 의한 것임

하였다.

## 2. 관련 연구

### 2.1 iRTOS™

임베디드 시스템 용 실시간 운영체제인 iRTOS™는 멀티 태스킹이 가능하며 각 태스크는 우선순위를 지니고 있어, 우선순위가 높은 태스크가 먼저 수행하는 선점형 방식을 채택하고 있다. 여러 개의 태스크를 정적, 동적으로 생성하거나 삭제 할 수 있으며, 같은 우선순위를 갖는 여러 개의 태스크를 위해 Round-Robin 을 지원하는 스케줄링 정책을 가지고 있다.

일정 시간 후 특정 루틴을 수행시키기 위한 타이머 관련 기능과, 중요도에 따른 차별적인 인터럽트 처리 루틴, 그리고 태스크간 통신 ( ITC:Inter Task Communication )을 위하여 세마포어, 메일 박스 그리고 메시지 큐 등을 제공한다[1. 6].

### 2.2 Java Virtual Machine

#### 2.2.1 자바 기술

자바 기술은 가정에서 사용되는 일종의 대화형 제어기 개발 프로젝트 진행에서 탄생된 기술로 1995 년 Sun Microsystems 에서 개발되었다. 이것은 인터넷의 빠른 확산과 더불어 차세대 컴퓨팅 플랫폼으로 인식되고 있는 기술로, 아래 그림 1 은 자바 실행 과정과 JVM 의 위치를 보여준다.

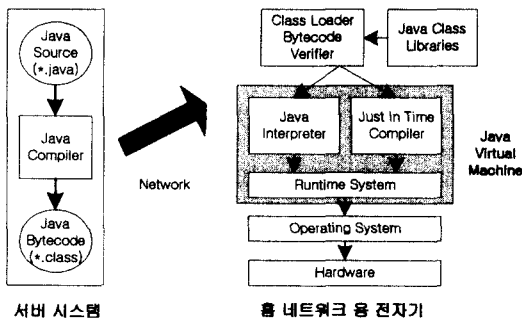


그림 1. 자바 실행 과정과 JVM 의 위치

서버 시스템은 자바 소스 코드를 컴파일 한 bytecode 를 네트워크를 통하여, 가정용 홈 네트워크 전자 기기들에 다운로드 하며, 홈 네트워크 기기들은 다운로드 된 클래스와 프로그램들을 실행 할 때 필요한 기본적인 라이브러리 클래스들을 클래스 로더로 로딩하여 자바 인터프리터로 해석하면서 실행한다. 위 과정의 최대 장점은, 플랫폼에 독립적으로 실행되어지는 자바 기술을 사용함으로써, 다양한 하드웨어 플랫폼이 존재하는 홈 네트워크 가전기기에서 플랫폼과는 무관하게 일관된 어플리케이션 실행환경을 제공할 수 있다는 점이다. 이는 사용자와 개발자에게 모두 편리함을 주는 특성이다.

#### 2.2.2 자바 플랫폼과 JVM

자바 코어 플랫폼은 자바 프로그램의 핵심적인 클래스들인 자바 코어 클래스들과 JVM, 그리고 JVM 과 여러 운영체제 및 브라우저 중간에서 연결 역할을 하는 Porting Interface 와 코어 자바 클래스들 위해서 자바의 활용을 확장 시켜주는 역할을 하는 자바 표준/확장 클래스들로 구성된다.

이 중 JVM 은 일종의 소프트웨어 CPU 로, 자바 번역기라고도 불린다. 이것은 컴파일 된 자바 바이트 코드와 컴퓨터 운영 시스템간의 번역기 역할을 하며 자바로 작성된 응용 프로그램을 윈도우나 매킨토시 등 플랫폼에 관계없이 사용할 수 있도록 해주는 소프트웨어이다. 즉, OS 에 의존적인 Porting Interface 를 맞춰준 자바 가상 기계가 홈 네트워크용 가전 기기에 내장되어 있다면, 그 가전 기기의 플랫폼에 무관하게 자바 프로그램을 다운로드 받아서 실행시킬 수 있다는 것이다[5].

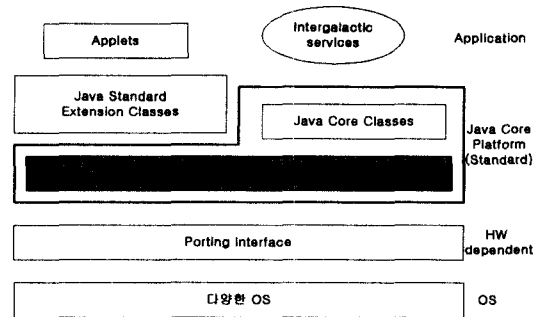


그림 2. 자바 플랫폼

#### 2.2.3 임베디드 자바 가상 머신

빠르게 발전하는 임베디드 시장에서 인터넷 정보 가전을 위한 요구를 충족하기 위해, Sun Microsystems 에서는 임베디드 시스템을 위한 기술로, PersonalJava(이하 pjava)와 EmbeddedJava(이하 ejava)를 소개했다. 임베디드 용 JVM 은 전체적인 자바 기술을 필요로 하지 않는 임베디드 시스템에 탑재되도록 고안된 것으로, ejava 는 산업계에서 쓰이는 제어기나, 프린터, 라우터 등에 사용되는 것으로, 총 1MB(롬+램)를 차지하며, pjava 는 엔터프라이즈 자바가 제공하는 수준의 기능을 필요로 하지 않는 단말기 또는 셋탑 박스 등에 쓰이며 전체 약 2MB(롬+램)를 차지하는 것이다. 본 논문에서는 pjava 를 사용하였다 [2. 3].

## 3. 통합 과정

본 논문에서 구현 하고자 하는 인터넷 가전용 실시간 운영체제 인 iRTOS™와 임베디드 자바의 통합은 타겟 보드는 IBM PC 이고 호스트는 IBM PC 상의 Visual Studio 컴파일러를 사용한 개발 환경이다. iRTOS™를 기반으로 임베디드 자바 가상 머신을 탑재하여 임베디드 자바 가상 머신인 pjava 를 iRTOS™ 상에서 동작 가능하도록 하는 것이다.

통합 과정은 설계와 구현 및 수정과정을 거친다.

### 3.1 설계

통합 과정의 첫 단계로 두 개의 모듈을 통합하기 위하여 먼저 수정 및 구현이 필요한 부분과 필요 없는 부분, 즉 플랫폼에 독립적인 부분과 의존적인 부분으로 크게 나누어서 설계도를 작성하였다.

iRTOS™에 탑재하기 위한 pjava의 전체적인 설계도는 다음 그림 3과 같다.

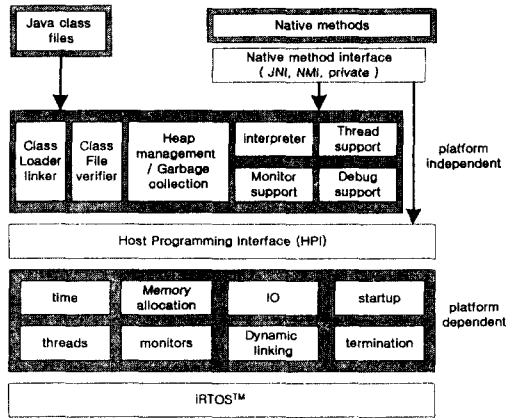


그림 3. iRTOS™ 상의 pjava

먼저 Native Method layer는 JVM에서 제공해주어야 하는 부분과 운영체제에서 제공해주어야 하는 부분으로 나뉘며, platform independent layer는 JVM과 관련된 일반적인 서비스들을 제공하는 곳으로, iRTOS™에 통합 시 수정될 필요가 없는 부분이고, platform-dependent layer는 HPI(Host Programming Interface)를 구현하는 함수들을 갖는 것으로 운영체제와 밀접한 연관성을 갖는다.

### 3.2 구현 및 수정

JVM에서 사용한 프로그램 언어는 자바, C 그리고 어셈블리 언어이다. 아래 그림 4는, 각 언어별로 그림 3의 계층을 나누어 본 도식으로, 음영이 넣어져 있는 부분이 C와 어셈블리 언어를 사용한 것이고, 음영이 넣어져 있지 않은 부분은 자바코드로 작성된 것이다. 또한 이 그림에서는 각 layer 사이의 구현 방법들이 기술되어 있다.

JVM과 운영체제와의 통합은 크게 3분류로 나뉘어져 있다. 첫번째는 운영체제와 JVM이고, 두 번째는 JVM과 클래스 라이브러리들, 마지막으로 운영체제와 클래스 라이브러리이다.

클래스 라이브러리에서는 자바 이외의 코드를 사용하기 위하여 native 함수들을 사용한다. 이 중에는 JVM과의 인터페이스를 맞추어 주어야 하는 것과, 운영체제에서 제공하는 함수들이 있으며, 본 연구에서는 이미 pjava에 맞추어져 있는 클래스 라이브러리를 사용하였으므로, JVM과의 인터페이스가 아닌 운영체제와의 인터페이스에 중점을 두어 수정하였다. 그리고 기존 iRTOS™에서 제공하지 않던 몇몇 함수들은 추가 구현 하였다.

본 작업에서 가장 중요한 JVM과 운영체제와의 인터페이스를 맞추는 작업은 먼저 파일 시스템이 없는 iRTOS™에 통합을 하기 위하여, pjava를 ROMIZE 형태로 사용하기로 하였으며, 태스크 기반으로 작동하는 iRTOS™에 멀티 스레드를 제공하는 JVM을 통합하기 위하여, pjava에서 제공하는 user-level thread인 green thread를 사용하여, iRTOS™의 태스크에 일대일 매핑을 하였다. 또한 스레드의 문맥 교환(context switching)에 사용하는 어셈블리 함수들 또한 HAL(Hardware Abstraction Layer)에 맞추어서 수정하였다.

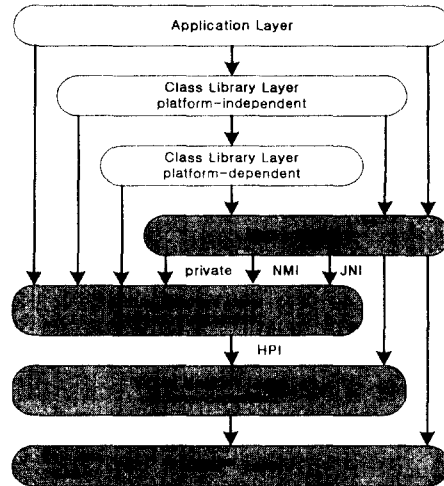


그림 4. 언어 별 통합 그래프

### 4. 결론 및 향후 연구과제

iRTOS™상에 pjava를 탑재 하는 것은 임베디드 시스템의 제한적인 상황을 보완 할 수 있는 좋은 방법으로 확인되었다. 구현 과정에서 green thread와 iRTOS™의 태스크사이의 함수 매핑에서 문제가 발생하였는데, 향후 추가적으로 연구할 방향은 이 문제를 해결하는 것과, 좀더 나아가 Sun사의 JVM이 아닌 독자적인 JVM과 클래스 라이브러리를 개발하는 것이다

### 5. 참고문헌

- [1] <http://www.inestech.com>
- [2] <http://java.sun.com/products/personaljava/>
- [3] <http://java.sun.com/products/embeddedjava/>
- [4] <http://java.sun.com/j2me/>
- [5] <http://java.sun.com/docs/books/vmspec/html/VMSpecTOC.doc.html>
- [6] S. Ryu, H.-B. Pyun, J.-H. Lee, K.-H. Yoon, J.-K. Lee, C.-H. Lee, "Design and Implementation of a Real-Time Operating System for the CDMA Systems," Proc. 26th KISS Fall Conf., pp.140-142. Oct. 1999.