실시간 운영체제에서 iRTOS에서의 CVM GUI 설계 및 구현

Design and Implementation of GUI in CVM on Real-Time Operating System, iRTOS

최찬우, 이철훈
충남대학교 컴퓨터공학과

Choi chan-woo, Lee cheol-hoon
Dept. of Computer Engineering, Chungnam National Univ.

요약

셋탑박스나 스마트폰과 같은 임베디드 장비는 GUI (Graphical User Interface)기능 제공 및 플랫폼 독립성 (Platform Independance) 유지를 위해 자바 환경을 사용한다. 이러한 장비에 자바 환경을 적용하기 위해 SUN에서 제공하는 임베디드용 자바가상머신의 하나인 CVM(Classic Virtual Machine)을 탑재하여 사용하게 된다. 자바의 GUI를 제공하기 위해서는 CVM에 포함된 PBP(Personal Basis Profile)에서 명세하고 있는 GUI 표준 API를 사용해야 한다. GUI 표준 API를 구현 할 때에 자바의 네이티브 메서드와 운영체제의 네이티브 함수간에 상호 연동이 될 수 있도록 JNI(Java Native Interface)를 사용하여 구현한다. 이에 본 논문에서는 실시간 운영체제 iRTOS를 기반으로 CVM GUI를 구현하기 위해 그래픽 윈도우 시스템과 GUI 표준 API와의 상호 연동을 하는 네이티브 함수와 각종 이벤트 처리에 대해 설계 및 구현한 내용을 기술한다.

Abstract

JVM(Java Virtual Machine) has GUI(Graphical User Interface) facility and Platform Independance and is used on the embedded device such as set-top box and smart phone. This needs JVM to execute Java application in the embedded device. CVM(Classic Virtual Machine) which is the kind of JVM is designed for embedded device to have limited resources. To support GUI facility of JAVA uses PBP(Personal Basis Profile) which is included on CVM. The PBP defines the GUI Stardard API to support GUI facility. When the GUI Stardard API is implemented, JNI(Java Native Interface) is used to connect between Java Native Method and Native function in Operating System. In this pater, PBP which is defined by CVM has designed and implemented on the Real-Time Operating System, iRTOS.

I. 서 론

IT 산업이 발전하면서 리소스가 제한된 임베디드 디바이스는 빠르게 발전하고 다양한 기능들을 요구한다. 임베디드 디바이스는 일반 데스크탑 PC가 아닌 리소스가 제한된 디바이스이다. 이러한 디바이스에 플랫폼 독립성(Platform Independency), 보안성(Security), 네트워크 이동성(Network Mobility)등의 장점을 가진 자바를 지원하기 위해 클래식 가상 머신(CVM, Classic Virtual Machine)의 탑재가 증가하는 추세에 있다. CVM은 이러한 제품을 사용하는 사람들에게 편리한 그래픽 환경을 지원하기 위해 GUI(Graphical User Interface)를 제공하고 있다.

편리한 그래픽 환경을 제공하기 위해서 SUN에서 제공하는 J2ME(Java 2 Micro Edition)의 프로파일인 PBP(Personal Basis Profile)는 자바의 GUI를 위한 표준 API를 java.awt 패키지로 명세하고 있으며, 이는 운영체제의 그래픽 윈도우 시스템과 연계되어 동작하게 된다. 본 논문에서는 실시간 운영체제인 iRTOS 상에서 그래픽 윈도우 시스템과 PBP의 GUI API와의 상호 동작을 위한 네이티브(Native) 함수(기본 도형 그리기, 색 지정, 텍스트 처리) 및 이벤트 처리에 대해 설계 및 구현한 내용을 기술한다.

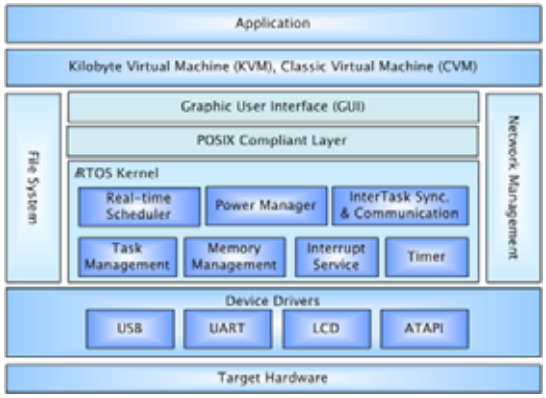
본 논문의 구성은, 2장에서는 관련 연구로서 CVM의 기반 운영체제인 실시간 운영체제 iRTOS와 PBP에 대한 개념을 기술하며, 3장에서는 CVM의 GUI와 그래픽 윈도우 시스템의 상호 동작을 위한 네이티브 함수와 이벤트 처리에 대한 설계 및 구현을, 4장에서는 테스트 환경 및 결과를 기술한다. 마지막으로, 5장에서는 결론 및 향후 연구과제에 대해서 기술한다.

II. 관련연구

1. 실시간 운영체제 iRTOS

본 논문에서 CVM을 구현하기 위해 기반으로 사용한 iRTOS는 우선순위 기반 선점형 멀티 쓰레드(Multi-thread) 실시간 운영체제이다. 즉 실시간 운영체제 커널과 응용 프로그

램이 통합되어 하나의 큰 프로그램으로 동작하는 구조로써 공통의 메모리 영역을 자유롭게 접근할 수 있고 크기는 약 24KByte 정도이다.



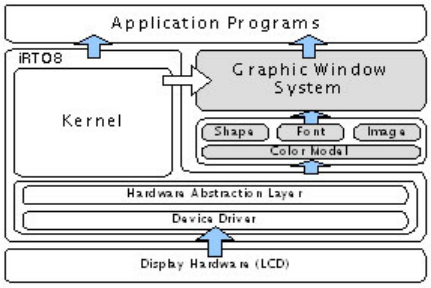
▶▶ 그림 1. iRTOS의 기능 블록 다이어그램

그림 1. 은 iRTOS의 기능을 블록 다이어그램으로 나타 낸 것으로, 운영체제에서 제공되는 기능은 태스크 관리, 태스크간 동기화/통신, 동적 메모리 관리 등이 있다.

시스템 메모리의 힙(Heap) 영역을 관리하기 위해 힙 스토리지 매니저를 사용하며, 힙 영역에서 동적 메모리의 할당은 MK_GetMemory()를 통해 이루어진다[1].

2. iRTOS의 그래픽 윈도우 시스템

그림 2.는 현재 iRTOS상에서 구현된 그래픽 윈도우 시스템으로 타겟 플랫폼이 바뀌었을 경우 쉽게 이식시키기 위해 Layered Architecture Design 방식으로 설계되어 있다.



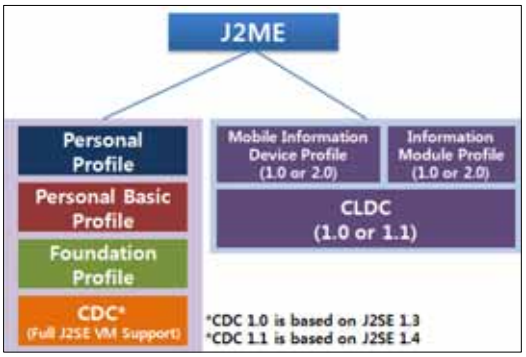
▶▶ 그림 2. iRTOS의 그래픽 윈도우 시스템

3. PBP (Personal Basis Profile)

PBP는 그래픽 유저 인터페이스 프레임워크를 지원하여 리소스가 제한된 디바이스에서 Java API를 지원한다. PBP는 다음과 같은 특징을 가진다.

- 경량화(lightweight) 컴포넌트 툴킷을 빌딩하기 위한 그래픽 유저 인터페이스 프레임워크
- Xlet 애플리케이션 프로그래밍 모델을 지원
- FP(Foundation Profile)에 포함된 애플리케이션 지원 API를 모두 포함

이러한 PBP의 그래픽 유저 인터페이스 API는 java.awt, javax.microedition.xlet 패키지를 통해 제공되고 있다[2,3].

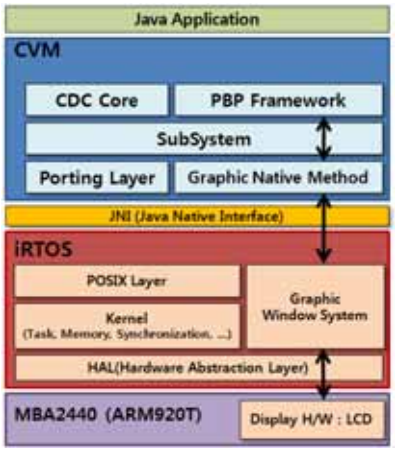


▶▶ 그림 3. J2ME 구조

Ⅲ. CVM의 GUI 및 이벤트 처리의 설계 및 구현

1. 설계시 고려사항

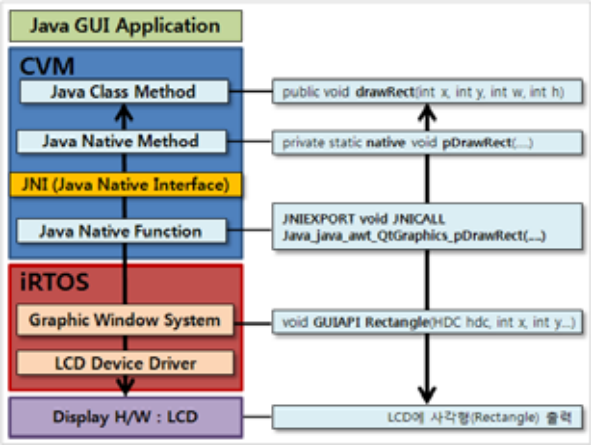
CVM의 GUI 기능을 지원하는 PBP를 구현하기 위해서는 이전에 CVM에서 정의하고 있는 CDC(Connected Device Configuration)가 구현되어 있어야 한다. CDC는 클래스 로더 부분(Class Loader)과 클래스 파일을 얻어 와서 가상머신의 실행상태로 만들어주는 클래스 런타임 데이터 영역(Runtime Data Areas)과 바이트 코드의 인스트럭션을 수행하는 실행 엔진(Execution Engine)으로 세분화된다. 그림 4.에서 이러한 기능을 수행하는 SubSystem과 실시간 운영체제 iRTOS와 연결되어 태스크 및 메모리 관리를 지원하는 CDC 코어는 이미 설계 및 구현되어 있음을 명시한다[5].



▶▶ 그림 4. iRTOS와 CVM 구조

2. GUI의 네이티브(Native) 함수 구현

본 논문에서는 iRTOS의 그래픽 윈도우 시스템과 PBP에서 지원하는 GUI API의 상호 동작을 위한 네이티브 함수를 구현하였다.



▶▶ 그림 5. CVM의 GUI 동작 메커니즘

그림 5.는 GUI 기능을 제공하는 Java Class Method를 호출해서 iRTOS상의 그래픽 윈도우 시스템의 함수를 실행시켜 최종적으로 LCD 하드웨어에 사격형(Rectangle)을 출력하는 동작 메커니즘을 나타낸 그림이다.

Java Class Method는 GUI의 동작을 위해 Java Native Method를 호출한다. Java Native Method는 실제적인 작업을 수행하는 Java Native Function과 연결시켜주는 역할을 담당하며, 이때 JNI(Java Native Interface)를 사용해서 이를 연결하였다. 실제 화면에 출력되기 위한 Java Native Function의 구현은 iRTOS에서 제공하는 그래픽 윈도우 시스템에 포함된 사각형 그리기 함수인 "void GUIAPI Rectangle ()"를 호출하여 구현하였다. 이와 같은 동작 과정을 통해서 기본 도형, 색 지정, 텍스트 처리에 관련된 API를 구현하였다. 표 1.은 구현한 GUI API의 목록이다.

[표 1] 구현 GUI API 목록

함수명	기능
fillRect	배경색을 포함한 사각형 그리기
drawRect	배경색이 없는 사각형 그리기
drawLine	선 그리기
drawString	문자열 그리기
setColor	색깔 지정하기
drawImage	이미지 그리기

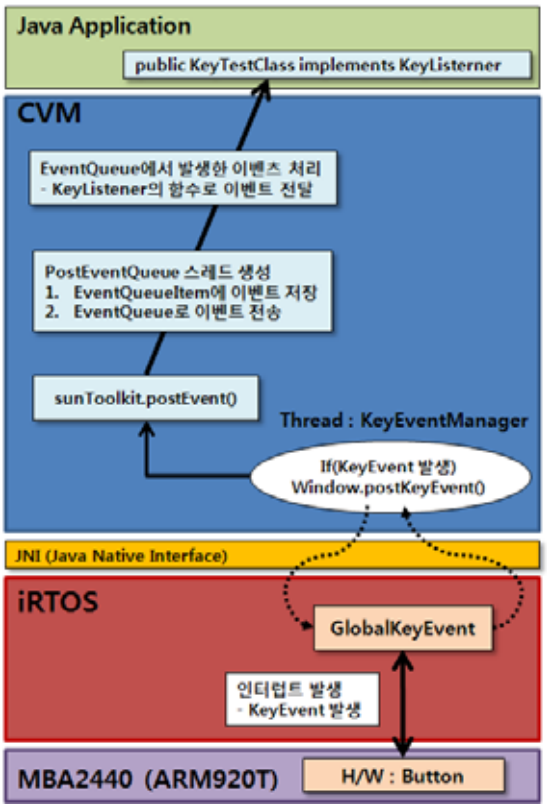
3. GUI의 이벤트 처리 구현

CVM 내부에는 키 이벤트(Key Event)를 처리하는 Key Listener 인터페이스와 알고리즘이 구현되어 있다. 이러한 이벤트 처리 메커니즘과 실시간 운영체제 iRTOS와의 연동을 위해 실제 하드웨어 버튼에서 발생한 키 인터럽트를 처리해 주는 루틴을 구현하였다.

그림 6.은 하드웨어의 버튼 클릭으로 발생한 인터럽트가 운영체제를 지나 CVM의 최상위 자바 애플리케이션까지 이벤트가 전달되는 과정을 나타낸 그림이다.

하드웨어 버튼 클릭시 발생하는 인터럽트와 해당 버튼 이벤트를 처리해주는 인터럽트 처리 함수를 실시간 운영체제 iRTOS에서 제공하는 MK_IRQInstall() 함수를 사용하여 등록하였고, 버튼 클릭으로 인한 인터럽트 발생시 GlobalKey Event 전역변수에 발생한 인터럽트에 대한 값을 저장하였다. 발생한 이벤트를 CVM에서 감지하기 위해서 키 이벤트만을 처리하는 KeyEventManager 스레드를 생성하였고, Key EventManager는 pGetEventOccurred() 네이티브 함수를 사용하여 GlobalKeyEvent의 값을 체크하여 키 이벤트의 발생을 확인하였다.

키 이벤트 발생 시 발생한 이벤트 정보를 CVM의 내부 클래스인 Window, SunToolkit, PostEventQueue, EventQueue를 사용하여 자바 애플리케이션의 KeyListener 인터페이스를 구현한 함수를 호출하도록 구현하였다.



▶▶ 그림 6. CVM의 이벤트 처리 메커니즘

Ⅳ. 테스트 환경 및 결과

본 논문에서는 ARM920T 기반의 S3C2440 32-bit RISC Micro Processor를 사용한 MBA2440 보드 상에 ADS v1.2(ARM Developer Suite ver1.2) 개발환경을 사용하여 실시간 운영체제 iRTOS에서 CVM GUI를 설계 및 구현하여 테스트하였다.

그림 7.은 기본 도형, 색 지정, 텍스트 처리에 관련된 그래픽 함수를 사용하여 구현한 테스트 프로그램의 실행 화면이다.

▶▶ 그림 7. PBP를 이용한 그래픽 함수 실행 화면

그림 8.은 GUI API 함수인 drawImage()함수와 키 이벤트 관리자(KeyEvent Manager)를 사용하여, 핸드폰과 같은 임베디드 기기에서 실행되는 PushPush 게임을 구현하여 테스트한 실행 화면이다.

▶▶ 그림 8. PBP를 이용한 PushPush 게임 실행 화면

Ⅴ. 결론 및 향후 연구 과제

본 논문에서는 리소스가 제한된 디바이스에 자바 응용 프로그램을 실행시키기 위해서, 실시간 운영체제 iRTOS상에서 CVM GUI를 설계 및 구현하였다.

향후 연구과제로는 CVM의 다양한 기능을 지원하기 위해 네트워크 기능을 지원하는 프로파일(Profile)을 구현해야 하며, 애플릿(Applet)과 고급 GUI 기능을 지원하는 PP (Personal Profile)을 구현해야 한다. 또한 보안(Security) 기능을 위한 JSSE, 암호화(Cryptography) 기능 지원을 위하 JCE, 인증(Authentication) 기능 지원을 위한 JAAS와 같은 프로파일을 구현하여 실시간 운영체제 iRTOS상에서 더욱더 다양한 자바의 기능을 지원할 수 있도록 하여야 한다.

▌참 고 문 헌▐

[1] Aijisystems, "Embedded hardware & software solution, UbiFOS(Ubiquitous Flexible Real-Time Operating Systems) & MCU platforms," http://www.aijisystem.com
[2] Sun Microsystems, "CDC (Connected Device Configuration) Runtime guide"
[3] Sun Microsystems, "CDC : JAVA Platform Technology For Connected Device, Java Platform, Micro Edition, White Paper", June 2005.
[4] Sun Microsystems, "Inside the JAVA2 Virtual Machine second edition".
[5] 최찬우, 이철훈, "실시간 운영체제 UbiFOS™상에서 CVM 설계 및 구현", 한국컨텐츠학회, pp.812-816, 2007.11
[6] 손필창, 강희성, 정명조, 이철훈, "실시간 운영체제 UbiFOS™상에서 KVM GUI의 설계 및 구현", 한국정보과학회, Vol. 33, No. 1(A), pp.358-360, 2006.06