

확장된 마이크로윈도우즈 기반의 모바일 플랫폼 설계 및 에뮬레이터 구현에 관한 연구

윤지훈[○] 채영훈 문승진

수원대학교 IT대학 컴퓨터학과
{jhyoon01[○], yhchae, sjmoon}@suwon.ac.kr

A Study on a Mobile Platform Design and Implementation of an Emulator Based on the Extended Microwindows

Ji-hoon Yun[○], Young-hoon Chae, Seung-jin Moon

Dept. of Computer Science, The University of Suwon

요 약

현재 대중화 되어 있는 모바일 폰의 가격을 보면 Java Virtual Machine이 기반이 되는 플랫폼의 사용으로 인한 License 비용으로 많은 돈을 지불하고 있는 실정이다. 또한 Java는 C/C++로 컴파일 된 Binary File에 비해 속도가 떨어지고 이를 위해 프로세서 또한 고성능의 프로세서를 필요로 하기 때문에 가격이 비싸질 수밖에 없다. Native Binary를 사용하는 Microwindows Graphic Engine은 저 사양 프로세서에서 사용가능한 모바일 플랫폼으로써 Java Virtual Machine 보다 빠른 속도를 구현할 수 있고 GPL License를 따르기 때문에 생산단가도 절약할 수 있어 저가형 핸드폰의 대량 생산으로 인해 인도·브라질·중국 등 신흥시장에서도 보다 경쟁력을 가질 수 있을 것으로 기대된다. 또한 Device Independence의 특성을 가지고 있어서 확장이 자유롭다. 본 논문에서는 기존의 모바일 플랫폼과 Microwindows를 비교해보고 모바일 플랫폼으로써의 개발 방향에 대해 논해 보려 한다.

1. 서 론

현재 국내 모바일 서비스 시장은 개인 휴대단말기 보급률의 빠른 증가와 기존의 단순 음성, 텍스트 기반의 모바일 서비스에서 멀티미디어 중심의 서비스로 변화가 맞물려서 지속 적인 발전을 이루고 있다.

1세대 2세대를 지나 제 3세대로 불리는 모바일 기술의 등장에 따라 모바일 시장이 커져 가는 가운데 [1] BREW, GVM, KVM 등 플랫폼의 다양화로 인해 독자적인 콘텐츠 개발로 인한 고가의 이용요금, 다양한 콘텐츠의 부족, 인터페이스 호환성 문제, 운용의 불편함등이 야기 되었고 2001년부터 국책사업으로 개발되어 온 WIPI도 SUN사의 의존도가 높고 로열티가 존재하고 각 이동통신사별로 별도의 확장기술을 채택하는 등 WIPI도 대안으로 완벽하다고 볼 수 없게 되었다.

그 대안으로 Microwindows를 이용한 플랫폼을 제시해 보고자 한다.

본 논문의 2장에서는 현재 사용되고 있는 모바일 플랫폼들과 Microwindows를 비교해 보고 장점과 단점에 대해 논해보고 3장에서는 Microwindows의 플랫폼 개발 방향과 구현에 대해 논하고 4장에서는 이에 따른 결론과 개발 과정의 문제점을 서술하고자 한다.

2. 관련 연구

2.1 Mobile Platform

Platform이란 컴퓨터 용어로 어떤 작업 또는 기술 구현이 이루어질 수 있는 환경을 의미 한다.

우리가 컴퓨터를 사용하려고 하드웨어 장비를 움직이는데는 운영체제가 없으면 하드웨어 장비는 단순히 전기적인 장치일 뿐이고 우리가 사용하지 못할 것이다. 이렇게 하드웨어 장비에 정상적인 사용을 위해 만들어지는 환경을 의미한다.

모바일 플랫폼은 각 이동통신 사업자의 입맛에 맞추어 상당히 많은 종류의 새로운 구조들로 태어났다. 이러한 폭발적인 성장에 의해 실행 플랫폼인 VM(Virtual Machine)에 대한 관심이 고조되었으며, 국내외를 막론하고 그 발전 속도는 점점 심화되어 가고 있다.

VM 기술은 프로그래밍 언어에 따른 개발 기준으로 크게 JAVA와 C계열의 두 가지로 구분할 수 있는데, LG텔레콤의 KVM, XCE의 XVM은 JAVA 계열로, 신지소프트의 GVM(General Virtual Machine), 모빌탑의 MAP(Mobile Application S/W Plug-In), 퀄컴의 BREW(Binary Run-time Environment for Wireless)등은 C계열로 개발하고 있다. [1]

2.2 Mobile Platform의 종류

2.2.1 BREW

브루(BREW : Binary Runtime Environment for Wireless)는 CDMA용 무선 장치들을 위한 미국 퀄컴사의

응용프로그램 개발용 플랫폼으로 원시코드가 개방되어 있다. BREW를 이용하면 CDMA 칩셋이 장착된 어떠한 핸드폰 상에서도 동작이 가능한 응용프로그램들을 만들 수 있다.

BREW는 응용프로그램과 칩의 운영체제 사이에서 동작하므로, 응용프로그램이 시스템 인터페이스를 코딩하지 않는 것은 물론, 심지어 무선 응용프로그램에 대한 아무런 이해 없이도 그 장치의 기능들을 사용할 수 있다.

이렇게 많은 장점에도 불구하고 많은 로열티가 해외로 나가서 CDMA 이동 전화의 원천 기술을 보유하고 있는 퀄컴사에 대한 로열티 지급은 2004년 한해 6,500억원(단말기 대당 5% 선)에 달할 것으로 보인다. [2][3]

2.2.2 GVM

GVM(General Virtual Machine)은 순수 국산 기술로 만들어 졌으며 이동 단말기용으로 개발된 프로그램 스크립트를 시스템으로 다운로드 받아 사용하는 방식이다. (주)신시소프트가 Mobile C를 이용하여 개발하였다. Mobile C로 작성된 애플리케이션은 단말기에 최적화되고 그래픽 처리를 간소화하여 실행속도 면에 있어서 상당히 빠르다. 아울러 120여개의 내장 라이브러리를 제공하며, 확장 U 코드 채택을 통해 운영체제에 독립적인데다 CDMA, GSM, GPRS, PDC 등 모든 무선 전송 규격을 지원하여 강력한 멀티미디어 기능을 제공한다.

하지만 GVM을 이용해 서비스를 하는 콘텐츠 사업자는 매출의 4.5%를 로열티로 내야하며 자체 저장영역이 작아서 GVM에서 구현되는 게임은 그런점에서 조금 떨어진다. [2][3]

2.2.3 KVM

셀룰러폰, 페이지, 모바일 인터넷 디바이스, POS(Point-Of-Sale) 터미널, 정보가전 단말기처럼 소형의 자원이 제약된 디바이스들을 목표로 하는 이식성이 뛰어난 Java Virtual Machine이다. KVM이 지향하는 것은 Java 프로그래밍 언어의 핵심 능력을 보유하면서 수십 킬로바이트(Kilobytes)의 메모리를 가진 디바이스에서도 효율적인 VM을 개발하는데 있다.

하지만 일찍 시장에 선보인 것치고는 탑재돼 있는 단말기 대수가 적고 가장 적은 용량, 적은 컬러를 담아내서 모바일 게임 같은 경우 퀄리티가 떨어지는 단점이 있다. [2][3]

2.2.4 WIPI

WIPI(Wireless Internet Platform for Interoperability)란 이동통신 단말기에 탑재되어 응용 프로그램에 실행환경을 제공하는 모바일 플랫폼에 대한 표준 규격이다. 각 사업자의 별도 플랫폼에 따른 개발의 중복성, 사용자 선택권 제한 등을 해결하고자 플랫폼의 표준화 필요성이 높아져 2001년 7월 한국무선인터넷 표준화포럼 내에 모바일 플랫폼 특별분과가 신설되어 표준화가 추진된 이래 2002년 2월 WIPI V1.0 규격이, 2004년 2월 V2.0이 발표됐다.

이 플랫폼의 장점으로는 표준규격을 정하여 각 사업자들의 개발 환경을 복잡하지 않고 간결하게 해주었으며

멀티태스킹이라는 강력한 기능을 제공한다. Java Virtual Machine을 올려서 사용할 수 있도록 해주어 Native Binary와 Java 모두 개발 언어로 사용할 수 있도록 해주었다.

하지만 Sun 과의 저작권 분쟁으로 MIDP 규격이 필수로 포함되게 되어 결국 Sun에 주는 로열티는 피할 수 없게 되었다. 다른 플랫폼에 비해 아직 관련 개발 인력이 마흡한데다 추진 및 책임 주체도 상대적으로 느슨한 상태이고 이동통신사별로 별도의 확장기술을 채택할 경우 콘텐츠의 자유로운 호환은 보장 받기 어려울 것이다. [4][5]

2.2.5 Microwindows

Microwindows는 Open Source Project의 일환으로 MS Windows 나 X-Windows 같은 Graphical Windowing System을 지향하고 있다. 하지만 위에 열거한 Windowing System 보다 적은 용량과 적은 메모리를 필요로 하기 때문에 Embedded 환경을 위한 플랫폼으로 현재에는 포켓PC 등의 플랫폼으로 사용되고 있다.

장점으로는 GPL 라이선스를 따르는 무료 Open Source 플랫폼이며 독자적인 Graphic Engine을 통한 Device Independent 특징을 가지고 있어 제작 및 어플리케이션의 확장이 자유롭다. 또한 서버/클라이언트 방식을 지원하여 멀티태스킹 기능이 가능하고 컴파일 시 수백 KB의 작은 용량만을 필요로 하기 때문에 메모리의 활용도가 더욱 높아지게 된다. Microwindows는 Nano-X API와 함께 Win32 API와도 유사한 API를 지원함으로써 X-Window 프로그래밍에 익숙한 개발자와 Win32 API 프로그래밍에 익숙한 개발자 모두가 쉽게 개발 할 수 있는 환경을 제공한다. 또한 Native Binary형식의 파일을 지원하고 있어 Java Virtual Machine에서의 구현속도보다 빠른 실행을 할 수 있게 된다.

하지만 아직 핸드폰용 API가 지원되지 않고 있으며 모바일 폰용 Emulator도 개발중에 있다. [6]

3. Mobile Platform 개발

3.1 Mobile Platform 개발을 위한 API 개발

기본적으로 Microwindows API의 함수를 사용하여 Mobile Phone 전용 API의 제작이 필요하다.

제작해야 할 분야는 크게 두 가지로 나눌 수 있다.

첫 번째는 Handset Hardware의 제어를 API로써 HAL을 통해 모바일 폰의 사운드, 진동 등의 제어를 위한 API 작성이 필요하다.

두 번째는 그래픽 부분에 대한 API의 작성이다. Microwindows에서 제공하는 Widget은 없지만 FLNX [7] TinyWidget [8] 은 일반 윈도우용 Application에 적합하게 구성되어 있으므로 모바일 폰 전용 Widget의 제작이 필요하다.

3.2 Emulator를 이용한 Windows 개발환경 구축

리눅스 환경에서의 개발자 많지 않은 상태이기 때문에 Microsoft Windows에서 모바일 폰이 존재 하지 않는 상태에서 실제로 모바일 폰과 동등한 상태에서 테스트 할

수 있는 Emulator의 개발이 필요하다.



그림 1 모바일 폰의 계층도

이를 위해서 선행되어야 할 과제는 HAL(Handset Adaptation Layer)의 제작이다. 그림 1에서 나타난 것처럼 HAL은 Handset을 추상화하기 위한 계층으로 단말기 제조사들이 구현해야 할 API 계층이다. [9]HAL이 Target Machine에 Porting되게 되면 Hardware를 제어하기 위한 기능을 하게 되고 Microsoft Windows에 Porting되게 되면 Emulator를 위한 기능을 수행한다.

Emulator를 만드는 순서는 우선 Microwindows를 라이브러리화 하고 실제 핸드폰의 모양의 어플리케이션을 제작한다.

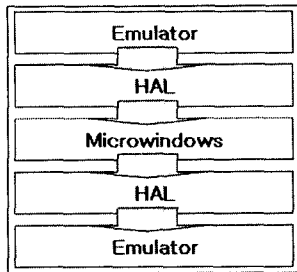


그림 2 Emulator 처리 순서

Emulator는 그림 2와 같이 Emulator에서 받은 입력값은 HAL을 거쳐서 Microwindows에서 처리후 다시 HAL을 거쳐서 Emulator에서 출력하는 방법을 사용한다.

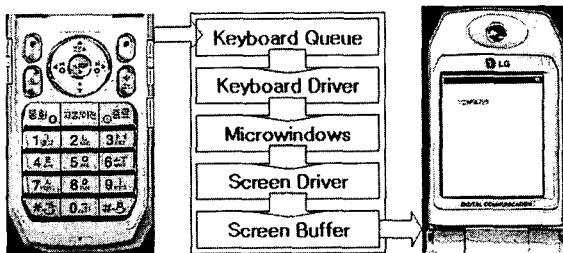


그림 3 실제 Key 입력 시 화면에 출력되는 처리순서

그림 3에 보여지는 것처럼 Emulator 상의 Key값은 입력시 HAL의 Queue에 저장된다. 이는 MW의 Driver를 거

쳐서 다시 Emulator상의 Screen Buffer에 출력하게 된다.

3.3 Emulator를 이용한 실제 응용 어플리케이션 개발
실제 Emulator 이용하여 Key 입력과 Screen 출력을 이용하여서 모바일 폰의 기본적인 기능인 알람, 계산기, 주소록등을 구현 할수 있다.

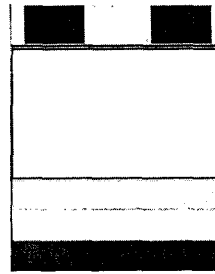


그림 4 주소록

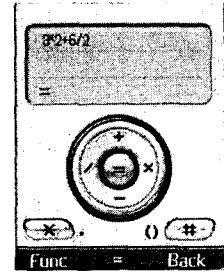


그림 5 계산기

4. 결론 및 향후 연구과제

Microwindows는 현재까지는 모바일 폰을 위한 플랫폼으로써의 개발 환경이 갖추어 지지는 않았지만 Open Source로서 무한한 발전 가능성을 내재하고 있고 GPL License를 통한 무료 플랫폼 이라는 메리트를 가지고 있다. 또한 작은 용량만을 필요로하여 메모리의 효율도 커지게 되어 모바일 폰 생산 비용의 절감효과가 기대된다. 또한 Win32 API 개발자와 X-Windows 개발자 모두를 위해 익숙한 개발환경을 제공하고 있어 미래에 많은 개발자들이 모일 것이라는 기대를 하고 있다.

하지만 그 전에 Microwindows가 Mobile 플랫폼으로써의 가능성을 보여주기 위해서 3장에서 설명한 API Set의 구현을 계획하고 있고 Microsoft Windows용 Emulator를 개발중에 있다.

참 고 문 헌

- [1] 한국 경제 2004년 3월 11일
http://www.roxia.co.kr/prroom/press_view.php?idx=3&page=1
- [2] 강상원, 임석진, 심양섭 : 위피 프로그래밍
- [3] 월간 디지털 콘텐츠 10월호
<http://exit7.co.kr/blog/?no=175>
- [4] 권기경 : WIPI 플랫폼 개발 기술 요소 및 서비스 환경
- [5] 김정훈 : 표준 무선인터넷 플랫폼 WIPI 기술 및 표준화 동향
- [6] Microwindows Architecture :
http://www.microwindows.org/microwindows_architecture.html
- [7] <http://www.fltk.org/>
- [8] <http://tinywidgets.sourceforge.net/>
- [9] 한국 무선 인터넷 표준화 포럼 제 1편 규격 구조 및 기능 Page .4