

스마트폰 OS를 위한 최적화 어플리케이션 개발에 관한 연구

이소행

청암대학 IT학과

A Study on optimizing application development for SmartPhone OS

Dept. of IT, Cheongam College

so-haeng, Lee

imac@scjc.ac.kr

요 약

개인용 휴대전화는 다양한 형태로 발전 되어 왔다. 그 동안 국내 통신 시장은 피쳐폰이라고 불리는 단순 전화, 문자 위주의 서비스만을 제공하는 하드웨어 기반의 서비스가 대부분이었다. 몇 년 전부터 PDA와 전화 기능을 기반으로 한 하드웨어가 출시되기 시작하였으나 사용 및 운영의 불편함으로 인해 수요의 폭을 넓히지 못 하였고, 2007년 Apple사의 iPhone 출시를 기점으로 전 세계적으로 다양한 형태의 스마트 폰이 출시되기 시작하였다. 또한 휴대폰 공급 업체들이 제공하는 일방적인 서비스에 만족하지 않고 사용자가 원하는 방향에 필요한 콘텐츠를 다양하게 제공할 수 있는 소프트웨어 마켓의 등장으로 스마트 폰은 그 사용자 층이 매년 큰 폭으로 확대되고 있다. 최근 스마트폰은 글로벌 경기 침체 속에서도 13%대의 성장률(2009년 3/4분기)을 보이며 세계 휴대폰 시장의 중심으로 자리 잡고 있다. 또 사용자의 요구를 충족 시키기 위해서는 다양한 플랫폼의 개발자가 양성 되고 있으며 다양한 어플리케이션을 제공하기 위해 노력하고 있다. 본 논문에서는 다양한 플랫폼 기반의 개발 환경을 소개하고 향후 개발자들이 접목할 수 있는 개발 도구 및 응용 방법을 비교 분석함으로써 스마트폰을 기반으로 한 최적화 어플리케이션의 개발을 촉진 시키고자 한다.

키워드

SmartPhone, Symbian, WinCE, iPhoneOS, Android

1. 서론

스마트폰(Smartphone)은 PDA 또는 유사한 개인용 전자 보조 기능을 하는 하드웨어의 특성과 휴대폰의 기능을 결합한 제품을 지칭한다. PDA를 개발하던 업체와 이동통신 단말기를 개발하던 업체가 PDA에 이동통신 단말기를 추가로 결합하여 PDA 폰이라고 부르기 시작했다.

PDA폰은 PDA 고유의 특성과 핸드폰을 결합한 형태여서 휴대성 및 사용성 면에서 열악했지만 많은 메니아들 사이에 차세대 IT기기로서의 가능성을 보여준 예라 할 수 있다. 하지만, 제한된 하드웨어의 성능과 불안정한 OS, 검증되지 않은 Application 으로 인해 PDA를 기반으로 한 폰은 사용자층을 넓히지 못하고 매우 적은 수의 메니아들의 전유물로 남게 되었다.

핸드폰을 제조하는 메이커에서는 사용자 입장의 다양한 콘텐츠를 제공하려는 시도가 점점 커지고 있었고, 제조사와 이동통신 서비스 업체의 연계로 제한된 형태의 콘텐츠들이 등장하고 있었

지만 그 대상이 청소년과 20대를 주 대상으로 개발되어진 것들 이었다. 콘텐츠를 이용하는 방법은 2G, 3G 통신망을 이용하는 형태로 기존 전화망을 활용하는 방법 이었지만 이로 인해 과다한 통신 비용이 청구되어 일부 청소년들의 사회 문제로 대두되기 시작했고 많은 사용자들이 좀 더 저렴한 통신요금, 남녀노소를 가리지 않는 다양한 콘텐츠의 제공, 빠른 속도, 안정성을 요구하는 목소리가 커졌다. 하지만 국내의 폐쇄적인 통신 서비스와 국가 정책으로 말미암아 사용자들이 품질 좋은 서비스를 기대하기엔 어려움이 많았다.

2007년 Apple사에서 풀 터치스크린과 혁신적인 UI(User Interface)를 장착하고 앱 스토어를 통한 어플리케이션 획득을 쉽고 빠르게 제공하는 스마트폰(iPhone)이 출시되자 스마트폰에 대한 관심은 폭발적으로 늘어나게 되었다. PDA에 대한 고정관념을 탈피한 스마트폰은 개인 일정관리, 인터넷, 이메일, 음성통화, MP3, 동영상 재생등을 원활하게 활용할 수 있는 소프트웨어를 탑재하고 획기적으로 빨라진 하드웨어의 발전으로 PC와의

격차를 줄여가기 시작했다.

본 연구에서는 국내에서 활발히 사용자 층이 넓어지고 있는 윈도우모바일6.5 기반의 개발환경, Apple iPhone 기반의 개발환경, 최근 가장 주목 받고 있는 안드로이드 기반의 개발환경을 비교 분석하여 스마트폰 어플리케이션 개발자들이 다양하게 활용할 수 있는 기반을 마련하고자 한다.

2.스마트폰 개발환경

스마트 폰의 개발환경을 살펴보기 전에 스마트폰의 분류, 플랫폼의 종류, 스마트폰 OS의 현황 및 특성을 명확히 알아야 한다. 특히 향후 5년간 스마트폰 시장은 시장을 선점하기 위해 극심한 경쟁이 예상되고 있으며 이 경쟁에서의 승자는 어느 누구도 장담할 수 없기 때문에 스마트폰 어플리케이션 개발자들은 다양한 플랫폼에 적응 할 수 있어야 한다.

iPhone 3GS (Apple, 09.06)	옵니아 2 (Samsung, 09.07)	Droid (Motorola, 09.11)	XpressMusic (Nokia, 09.04)	Mororoi (Motorola, 10.02)
				

그림 1. 최근 출시된 스마트 폰

2.1 스마트폰의 분류

스마트폰의 중요한 특징은 PDA기능, 무선인터넷 서비스(2G,3G,wi-fi) 기능이다. 또한 스마트폰은 일반폰보다 뛰어난 하드웨어를 장착하고 있다. 하지만 최근에는 일반폰의 사양이 스마트폰과 거의 차이가 없을 정도로 뛰어난 성능을 가지고 있어서 하드웨어의 성능만으로 일반폰과 스마트폰과의 차이를 말하기는 어렵다.

스마트폰과 일반폰을 구분할 수 있는 가장 큰 특징은 개방성이다. 스마트폰은 범용 운영체제를 사용하고 표준화 된 개발 환경을 제공하므로 개방화된 운영체제를 통해 개발자들이 자유롭게 어플리케이션을 개발할 수 있는 환경을 제공하고 있다.

따라서 사용자들은 개방형 운영체제를 기반으로 다양한 어플리케이션의 공유가 가능하기 때문에 손쉬운 정보 교환이 가능하다. 스마트폰은 사용자, 사업자, 제조사 측면에서 일반 휴대폰에 비해 다양한 이점을 제공한다.

스마트폰과 일반 휴대폰과의 차이는 보이스 중심의 서비스와 다양한 정보 제공과 양방향 정보 교환이 가능한 멀티미디어 적인 요소를 포함한 서비스의 차이라고 하겠다. 이러한 차이를 정리하면 다음과 같다.

스마트 폰	일반 휴대폰
.다양한 범용 OS	.보이스 중심 서비스
.멀티 태스킹/데이터 중심	.WiPi 기반 호스트만 접속
.대용량 메모리 장치	.카메라, MP3 및 멀티미디어 기능
.Wi-Fi, Bluetooth 지원	.SMS/MMS 위주
.풀 브라우징 서비스	.3rd party 어플리케이션 설치 불가
.3rd party 어플리케이션 설치/사용가능	
.서비스 오퍼레이터 인증필요	

표1. 스마트폰과 일반 휴대폰과의 차이점

3 스마트폰 OS의 현황 및 특성

스마트폰에서 볼 수 있는 운영체제는 심비안 OS, 아이폰 OS, 블랙베리 OS, 팜 웹OS, 윈도우 모바일, 구글 안드로이드, 마에모 등이 있다. 안드로이드와 마에모는 리눅스를 기반으로 작성되었으며 아이폰 OS는 모두 유닉스와 관련되어 있는 BSD와 넥스트스텝 운영 체제에서 비롯 되었다.

최근 오픈 소스 문화가 스마트폰 시장에 파고 들기 시작했다. 스마트폰의 하드웨어와 소프트웨어 둘 다 오픈 소스화 하려는 시도가 점차 증가하고 있다.

오픈 하드웨어 개발에서 가장 눈에 띄는 프로젝트는 오픈모코가 개발한 네오 프리러너 스마트폰이다. 뒤에 구글 안드로이드 OS가 대중적인 오픈 소스 모바일 운영체제가 되었다. 노키아 또한 2010년에 모든 심비안 스마트폰 코드를 심비안에 오픈 소스화 할 초안을 가지고 있다.

3.1 Symbian OS

EPOC이 가지고 있는 GUI인 'EIKON'의 특징은 화려하지 않지만, 통일된 구조로 되어있는 사용자 편의를 많이 배려한 Interface에 있다고 생각된다. EPOC에서 기본적으로 제공하는 Application에 대한 button이 하단에 정의되어 있고, 왼쪽에는 메뉴, Edit(윈도우에서 사용하는 것과 동일한 기능), infrared send/receive(무선 적외선 포트인 IrDA를 이용한 파일전송), 화면 확대/축소 등에 대한 버튼이 있다. EPOC이 가지는 Component 들 EPOC에서 사용하고 있는 각 Application들과 각 resource들이 여러 형태의 component 들로 구성되어 있는데, 각 process boundary는 그림 2와 같다.

Kernel은 PDA의 Resource들을 관리하고, 각 component가 각 resource 들을 접근하고 사용할 수 있도록 해준다. Kernel은 해당 resource를 얻기 위해 hardware supported privilege를 사용하는데, 이것은 CPU가 kernel을 위해 특정 privilege를 제공해 주는것을 말한다. 따라서, user mode에서 사용자에게 의해 실행되는 application들은 특정한 privilege가 없이도 kernel API를 통해 각 resource 들에 접근 할 수 있게 되는 것이다.

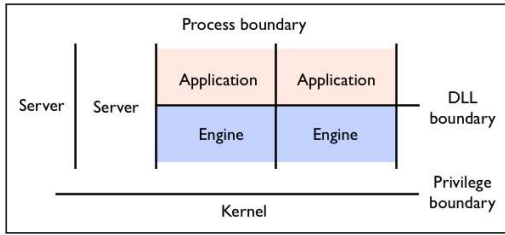


그림 2. EPOC의 각 Component 별 Boundary Type

Application은 각기 구분된 영역을 확보하고 구동 되게 되는데, 각 Application들은 서로 독립적인 주소 공간을 가지며, 다른 Application 영역의 data를 overwrite 하거나 access 할 수 없는데, 이는 각 application 사이의 process boundary 때문이다. Server는 User Interface 가 빠진 Application 과 같은 개념이다. server는 하나 이상의 resource들을 관리하게 되는데, 이는 EPOC 이 Client Server Frame work 으로 되어 있기 때문이다. 즉, 우리가 알고있는 server-client 프로 그래밍에서 볼 수 있듯이 EPOC에서도 내부적으로 하나의 resource를 여러개의 Application이 사용 할 수 있도록 하기 위해, 이러한 개념을 사용 하게 된 것이다.

3.2 WinCE OS

WinCE는 소형 컴퓨터나 PDA등에 사용되는 마이크로소프트 윈도우 제품으로, Windows 3.11 을 기반으로 시작해서 새롭게 만들어지 소형 OS 이다. 인텔 X86 계열, MIPS, ARM, 히타치 SuperH 프로세서를 지원한다. CE커널은 자체만으로 1메가바이트 이하의 메모리에서도 동작이 가능하도록 설계되어 있다. 장치들은 디스크 저장 장치를 사용하지 않고 설정할 수 있으며, 사용자가 추가하지 못하도록 만들 수도 있다. 윈도우 Ce는 실시간 운영체제를 표방하고 있으며 256단계의 우선순위 정도를 가지고 있다. 유닉스 계열 운영체제와는 달리 실행 파일의 기초 단위는 스탠드이다.

WinCE는 소스코드 형태로 제공된다. 따라서 제조사의 설정에 맞게 커스터 마이징이 가능하기 때문에 다양한 업체들이 사용하고 있다. WinCE 운영체제 이미지 작성 및 사용자 정의 도구인 Platform Builder같은 제품이 소스코드를 포함한 형태로 공개되지만 사용하기가 상당히 어렵고 문서화가 되어 있지 않다는 단점이 있다.

5.X의 가상 메모리 구조는 2GB의 커널, 2GB의 프로세서를 사용할 수 있는 구조이며 최대 32개의 프로세스 최대 32MB의 메모리를 사용하는 구조로 되어 있다. 이에 비해 6.X 버전은 프로세서 당 2GB의 가상메모리를 지원하고 최대 32000 프로세서를 지원하고 있다.

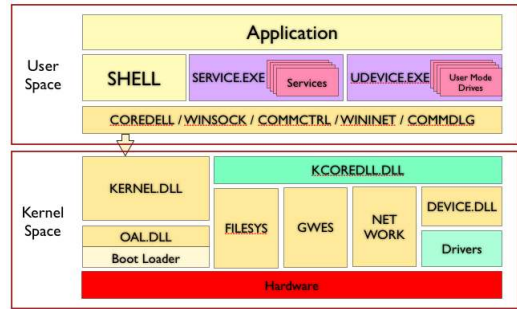


그림 3. WinCE 6.X의 구조

WinCE의 kernel은 "NK.EXE"이라는 컴포넌트가 된다. kernel의 주요 역할은 Memory관리, 스케줄링, Win32 APIwldnjs를 수행하며 Micro-kernel 구조를 통해 PSL(Protected Server Library)를 구현 한다. 마이크로소프트는 개발자 커널의 포팅을 최소화 하기위해 주요 커널의 기능을 별도의 라이브러리 파일에 담아 준비 하였으며, 하드웨어 또는 CPU에 관련된 포팅만을 개발자에게 부담하도록 하였다.

하지만 비약적인 메모리의 지원과 프로세서의 지원에도 불구하고 WinCE를 기반으로 하는 PDA, 스마트폰은 많은 문제점으로 인해 점점 그 사용자 층의 감소와 개발자의 감소로 이어지고 있는 추세이다. 특히 WinCE 자체 기술적인 진보가 거의 없는 상태였으므로 마이크로 소프트는 WinCE를 기반으로한 모바일/임베디드 기기분야에서 수억 달러의 손실을 보게 되었다.

3.3 iPhone OS

아이폰 OS는 애플의 스마트폰인 아이폰과 디지털 미디어 재생기인 아이팟 터치에 내장되어 있는 운영체제이다. iPhone OS는 Mac OS 10.5를 기반으로 한다. iPhone OS의 요소인 코코아, 코어 애니메이션 등의 어플리케이션 프레임워크를 포함하고 있고 여기에 멀티터치를 비롯하여 종래의 휴대전화 및 스마트폰에는 없었던 특징적인 유저 인터페이스를 구현하고 있다.

iPhone OS 의 내부 구조는 Mac OS X 와 유사하다. iPhone OS의 커널은 OS X 에서 사용된 기본적인 마하커널(Mach Kernel)의 변형된 구조이다. 이커널의 상부에 자리잡고 있는 서비스 계층은 이 플랫폼의 응용프로그램 구현에 사용된다. 그림은 이러한 계층을 보여주고 있다. 이런 계층 구조는 당신이 코드를 작성할때 선택을 할 수 있게 한다. 예를 들어 Core OS와 Core서비스 계층은 iPhone OS의 기초가 되는 인터페이스인 파일 액세스, 로우레벨 데이터, Bonjour(자동네트워킹)서비스, 네트워크 소켓, 등등을 포함하고 있다.

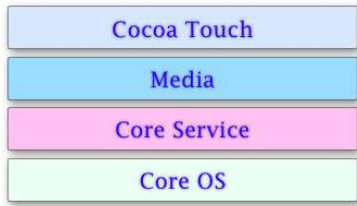


그림 4. iPhone OS의 계층구조

Cocoa 터치 계층은 대부분 오브젝트-C 기술을 사용하고 있다. 이 계층의 프레임워크는 당시의 응용 프로그램에 사용할 수 있는 기반이 되는 인프라를 제공한다. 예를 들어, 기반 프레임워크는 collection, 파일관리, 네트워킹 및 기타에 대해 오브젝트 지향적인 지원을 제공한다. UIKit 프레임워크는 개발되는 응용 프로그램에게 비주얼 인프라를 제공하는데, window, view, control class 등이 있으며 이러한 오브젝트에 대한 Controller들도 제공한다. 이 계층에서 다른 프레임워크들은 사용자의 전화번호부나 사진정보 등에 대한 접근방법과 가속도계 또는 기타 디바이스에 대한 하드웨어적 정보를 얻을 수 있게 해준다. 모든 신규 프로젝트의 시작점은 Cocoa 터치 계층이며 몇몇 경우 UIKit 프레임워크가 될 수 있다.

3.4 Android OS

모바일 장치에 올려지는 OS, 미들웨어, 키 어플리케이션들이 포함된 소프트웨어를 안드로이드라고 통상 말하고 있는데 리눅스 커널 기반으로 개발한 구글에서 만든 오픈소스 형태의 모바일 OS라고 말하는 것이 더 정확한 표현이다.

안드로이드는 시스템의 다양한 컴포넌트에서 사용된 C/C++ 라이브러리가 포함되어 있다. 이 기능들은 안드로이드 응용 프로그램 프레임워크를 통해 개발자들에게 공개되어 있는데 코어 라이브러리는 다음과 같다.

시스템 C 라이브러리 : 임베디드 리눅스 기반 기기용으로 조정된 표준 C 시스템 라이브러리의 BSD에서 파생된 구현이라 할 수 있다.

Surface Manager : 화면 하부 시스템과 다중 응용 프로그램에서 결합없는 2D/3D 그래픽 계층의 복합체에 대한 접근을 관리한다.



그림 5. Android OS의 구조

[<http://developer.android.com/guide/basics/what-is-android.html>]

Media library : PacketVideo의 OpenCORE에 기반을 둔다. 본 라이브러리는 MPEG4, H.264, mp3, AAC, AMR, JPG 및 PNG를 포함하고 다양한 대중적인 오디오/비디오 포맷과 정적 이미지 파일들의 재생과 레코딩을 지원한다. SGL : 기반 2D 그래픽 엔진을 의미한다. 3D library : OpenGL ES 1.0 API 기반으로 구현되어 있다. 하드웨어 가속, 최적화된 내장 3D 소프트웨어 레스터라이저 둘 모두를 사용할 수 있는 특징을 가지고 있다. 모든 안드로이드 응용 프로그램들은 자체 프로세스에서 달빅 가상 기기(Dalvik Virtual Machine)의 자체 인터페이스를 갖고 구동된다. Dalvik이 구현되어 있어 기기의 다중 VM을 효과적으로 구동할 수 있다. Dalvik VM은 최소한도로 메모리를 차지하도록 최적화된 달빅 실행 형식에서 파일을 실행한다. VM은 레지스터 기반이며, 자바 언어 컴파일러를 이용하여 컴파일되고 포함된 'dex' 두고를 이용하여 .dex 형식으로 변환된 클래스들을 구동한다.

4. 결론

한국 IT산업에서 HW와 SW의 불균형은 생각보다 심각하다. 국내 IT 총생산액 중 HW가 대부분(73%)을 차지하는 반면, SW는 8%(24.4조원)에 불과하다. 이는 세계시장의 흐름을 역행하는 것이라는 점에 심각성이 크다. 2008년 기준으로 세계 HW시장은 전체 IT산업의 22.4%(0.8조달러) 뿐이고, SW가 30%(1조달러)를 차지하고 있다. 2002년 이후 SW시장 규모가 HW를 상회하며 가파른 성장세를 보이고 있다.

모바일 기기의 하드웨어 성능은 비약적이라 할 수 있지만 대부분의 개발자들은 고성능의 PC에 개발 방법을 적용 운영하고 있어서 모바일용 어플리케이션 개발을 위한 최적화 기법을 적용할 필요가 있다. 이 논문에서는 다양한 종류의 스마트폰 OS를 비교 분석하여 최적화된 어플리케이션 제작을 위해 활용하여야 할 방법을 제시하였다. 앞으로 다양한 방법의 최적화 기법과 운영체제 구조를 분석함으로써 향상된 어플리케이션 개발에 힘써야 할 것으로 판단된다.

참고문헌

- [1] Programming in Object-C, Stephen G.Kochan, Developer's Library
- [2] Object-C 오기하라타케시, 한빛미디어
- [3] 아이폰 프로그래밍, 데이브마크외, 위키북스
- [4] 배성호, 김우생, "안드로이드 기반 모바일 정보공유시스템", 전자공학회논문지