Java & Android & NDK

Lee, Joon Won
삼성전자(소프트웨어센터)

목차

1) My Story of Java

2) Android Tip

3) NDK Basic on Android

소개

이준원

- OOO SI 업체
- OO 전자
- OOO 인터넷 포털업체
- 현 삼성전자 소프트웨어 센터
  - 웹 관련 부서

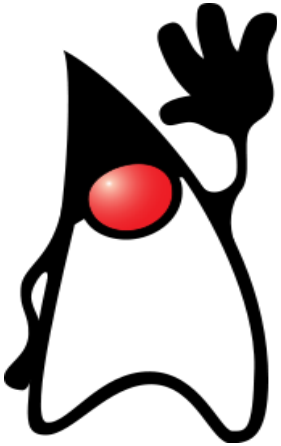
My Story about Java

1997~1999

Java has no pointer!!

What is NullPointerException??

My Story about Java



Duke, the Java mascot

James Gosling, the creator of Java

JDK 1.0 (January 21, 1996)

JDK 1.1 (February 19, 1997)

J2SE 1.2 (December 8, 1998)

J2SE 1.3 (May 8, 2000)

J2SE 1.4 (February 6, 2002)

J2SE 5.0 (September 30, 2004)

Java SE 6 (December 11, 2006)

Java SE 7 (July 28, 2011)

Sun microsystems

ORACLE

My Story about Java

2000~2002

Stock Boom!!!

HTS

Applet

Java Application

MS Java Virtural Machine

It was the **fastest Windows-based implementation** of a Java virtual machine for the first two years after its release.

Sun Microsystems, the creator of Java, **sued** Microsoft in October **1997** for **incompletely implementing** the **Java 1.1** standard

Microsoft settled the lawsuit with Sun and **discontinued its Java implementation in 2011.**

From Wikipedia

MS Java Virtural Machine

성능이슈 + preload된 MSJVM의 편의성 ➔
MS VM compatibility를 유지할 수 있도록 구현 ➔ Java 1.1 (AWT로만 UI 구현)

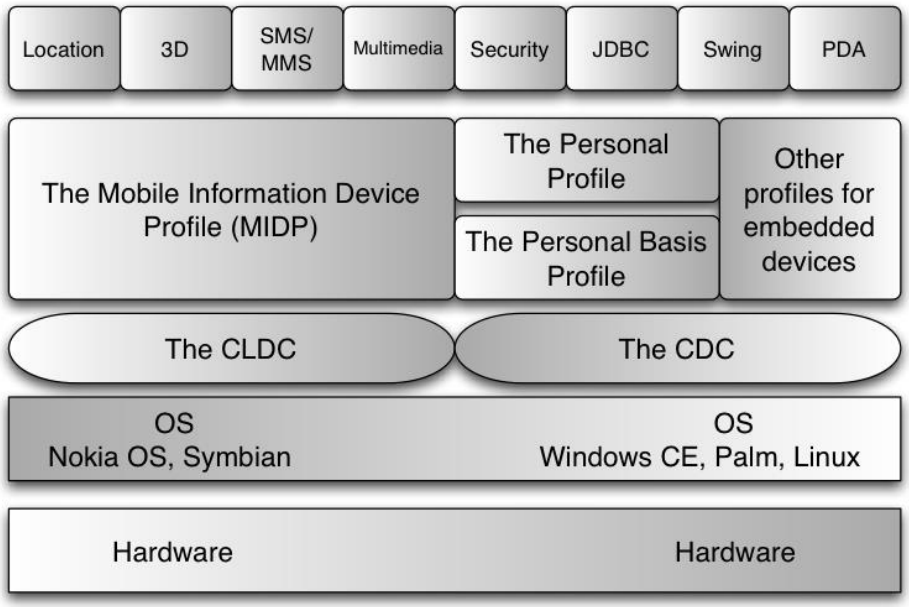
Window XP에서는 MSJVM의 preload 미지원 ➔ 다운로드 링크 제공

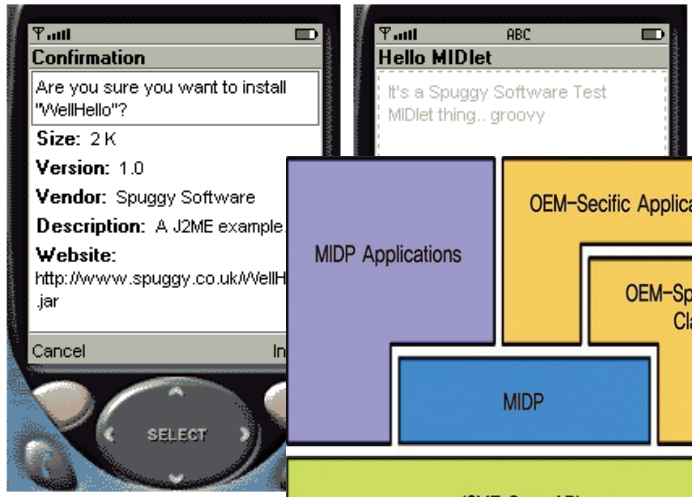
MS에서 .net framework 발표
Java 개발자를 위한 J# 언어 제공
ActiveX

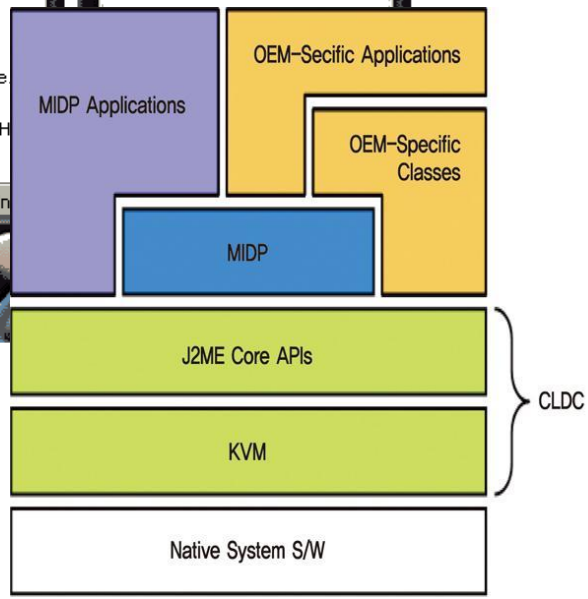
My Story about Java

2003~2006

Handset Mobile Java







My Story about Java

J2SE, J2ME, J2EE

KVM, MIDP, CLDC, MIDlet, AMS

MVM

JSR

TCK, JDTS

Esmotec, Qalcomm, Sun

My Story about Java

http://jcp.org/en/jsr/overview

MIDP 2.0 (**JSR 118**)

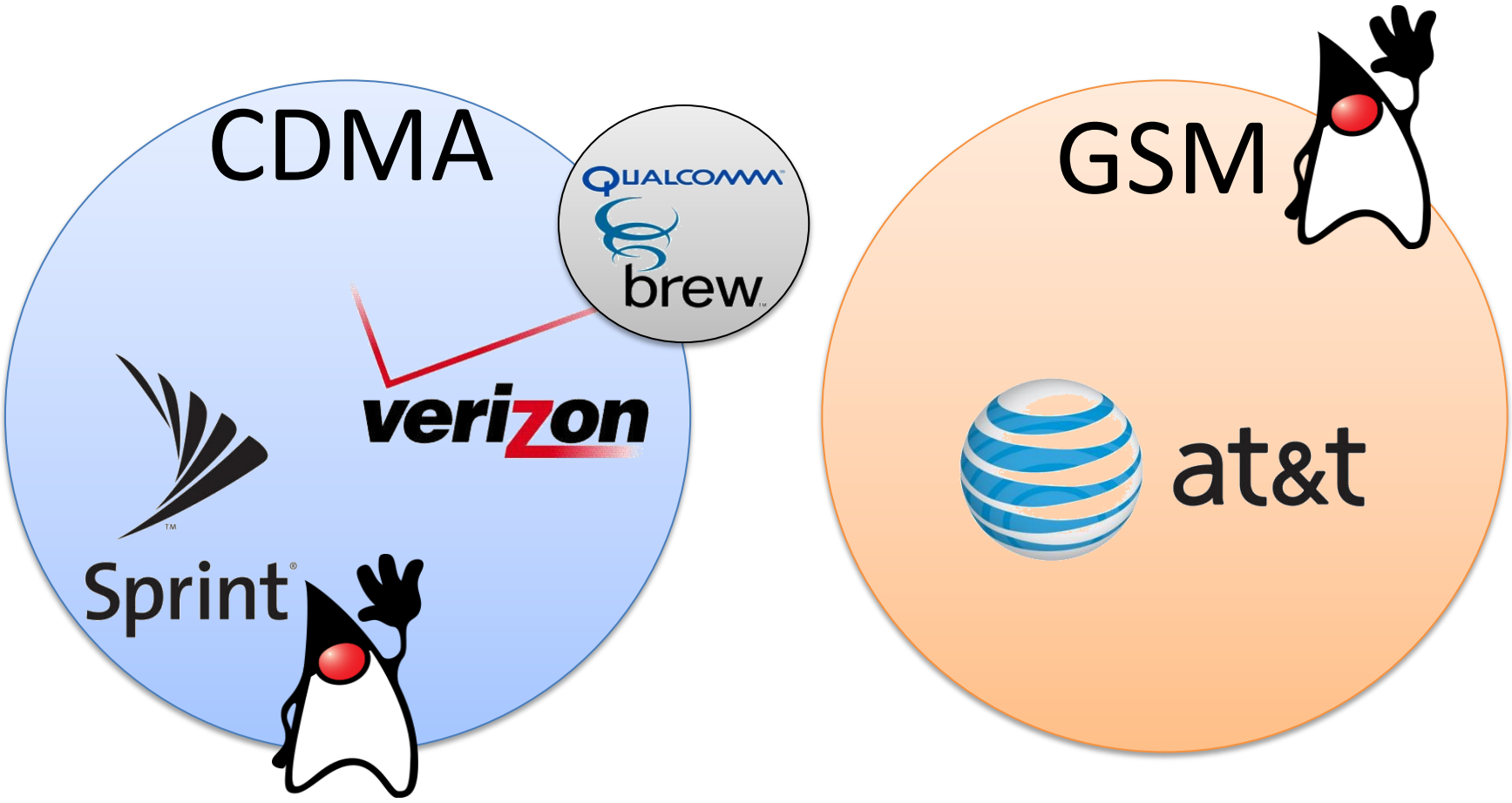
CLDC 1.1 (**JSR 139**)

Each Java technology has an **API specification**, a **reference implementation (RI)**, and a **technology compatibility kit (TCK)** associated to it.

Java Hotspot : JIT comilation

My Story about Java

미국이통사



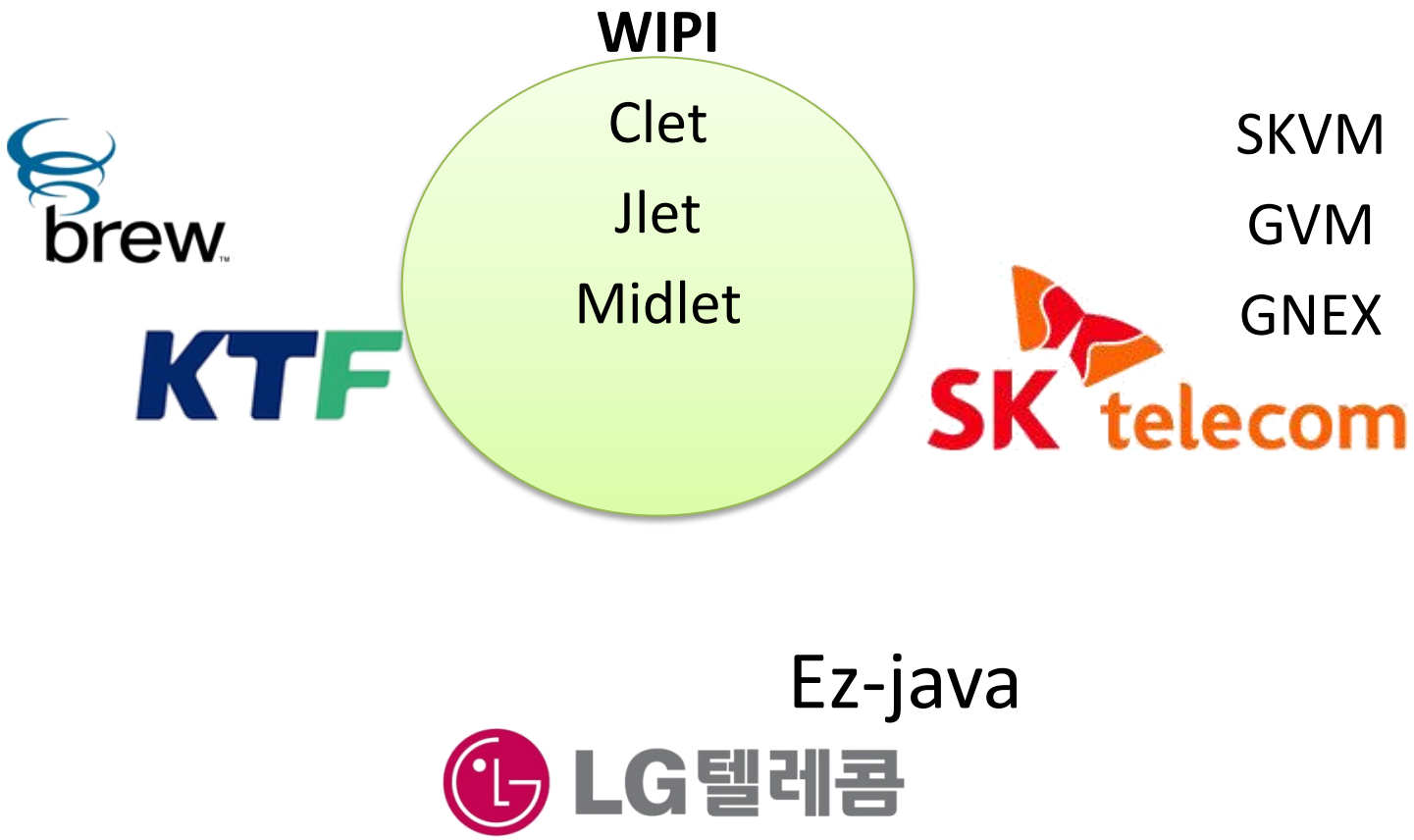
My Story about Java

2007~2010

- 국내 이통사
- 인터넷포탈
- WIPI
- WAP

My Story about Java

한국이통사



My Story about Java

이통사들의 노력

SKT 1mm

LGT 오늘은

KTF 팝업

My Story about Java

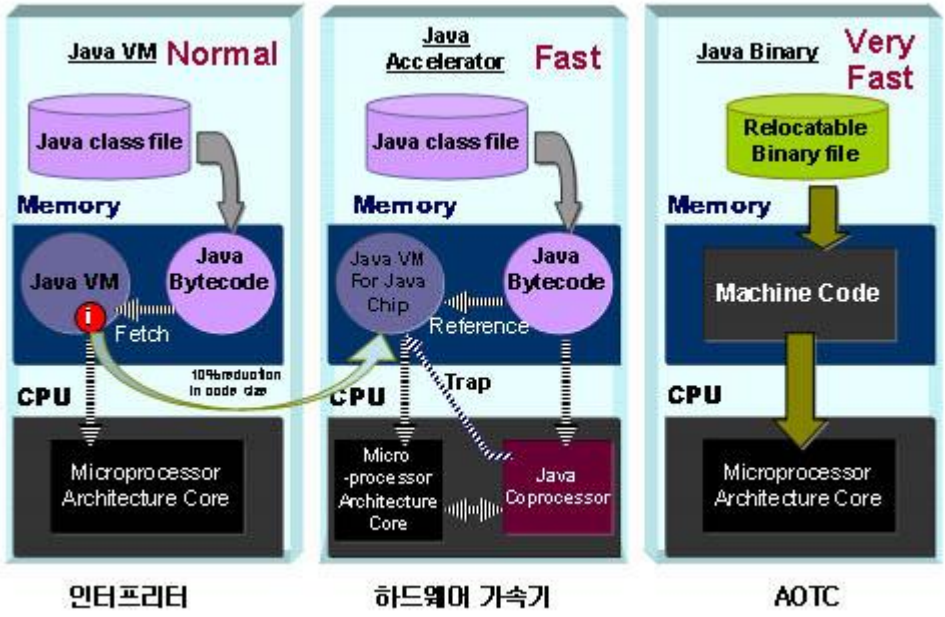
포털들의 노력

모바일369(WAP)

KTF 네이버팝업

My Story about Java

자바VM 가속



인터프리터 방식

자바 바이트코드를 소프트웨어 인터프리터로 수행하는 방식으로, 초기의 JVM(Java Virtual Machine)들에서 사용되었으며 속도가 현저히 느리다. 인터프리터 부분을 CPU칩에 최적화하도록 엔지니어링을 하면 어느 정도 빠른 속도를 낼 수 있지만, 바이너리 플랫폼과의 격차를 해소하기에는 역부족이다

하드웨어 가속기 사용 방식

소프트웨어 인터프리터를 하드웨어 가속기로 대체하는 방식이다. 하드웨어 가속기가 자바 가상머신의 모든 기능을 수행하는 것이 아니라, 자주 사용되는 일부 인스트럭션 외의 복잡한 인스트럭션은 trap이 발생되어 소프트웨어가 수행하는 방식이다. 상당한 속도 개선이 있으나, trap으로 인한 수행상의 손실이 발생하고, 부가적인 하드웨어로 인한 비용과 전원 소모를 고려하여야 한다.

ARM 칩 >> Jazelle

My Story about Java

VM 가속

Just In Time Compile(JITC) 방식

수행 중에 바이트 코드를 컴파일 해서 CPU에 맞는 바이너리 코드를 생성하는 방식이다. 컴파일 속도가 중요하기 때문에 최소한의 최적화만 수행한다. 메모리 요구량이 커서 메모리 제약이 많은 단말기에서 적절한 방식이 아니다. 최근 썬마이크로시스템즈사에서 몬티라는 이름으로 J2SE에서 사용하던 Hot-spot기술을 최적화하여 스마트폰과 같은 하이엔드 J2ME에 적용하려는 시도를 하고 있으며, 어느 정도 기술적 호응을 얻고 있다. 하지만 많은 리소스를 필요로 하는데 대한 부담과 이 기술을 사용하려면 추가 로열티를 지불해야 하는 부담이 존재한다.

Ahead Of Time Compile(AOTC) 방식

바이트 코드 형태로 되어 있는 자바 어플리케이션을 수행되기 전에 미리 컴파일 해서 단말기의 CPU의 최적화된 바이너리 코드를 생성하는 방식이다. JIT에 비해 충분한 최적화 시간을 가질 수 있다. 이 기술은 J2SE에서는 많은 연구 프로젝트가 있었으나 하드웨어 성능의 빠른 속도로 발전해 감에 따라 적절한 시장을 찾지 못하였다. 또한 자바의 바이트코드를 바이너리화 하면서 코드 크기에 커지게 되는데 이러한 문제도 해결해야할 과제이다.

My Story about Java

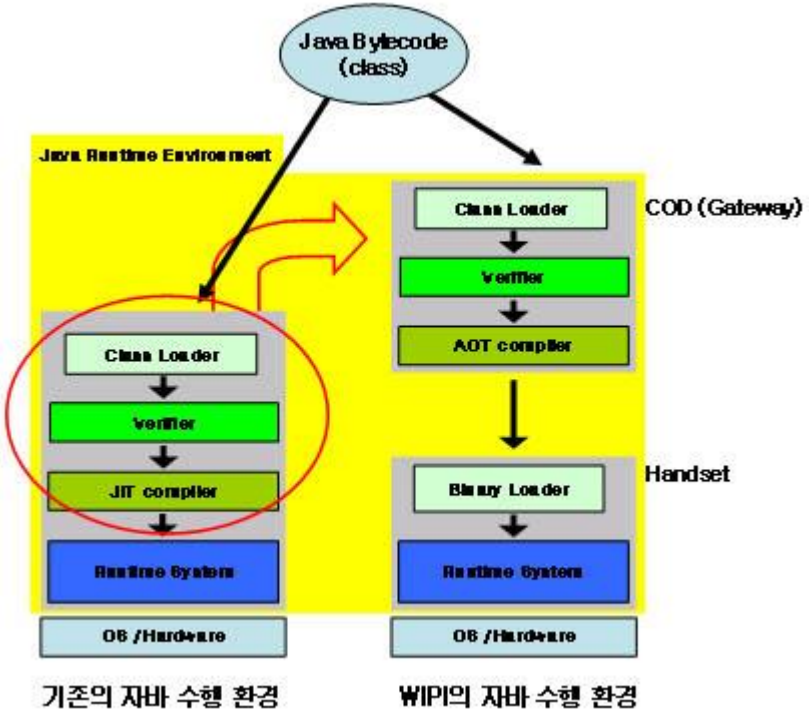
WIPI 규격(Jlet)

Ahead Of Time Compile(AOTC) 방식 채택

기존의 인터프리터 방식의 플랫폼에 비해 10배 이상의 향상

COD(Compile On Demand)

자바의 Verification 과정을 서버에서 진행
단말기가 여러 종류의 CPU칩으로 서비스 될 경우에 각각의 CPU에 맞도록 서버단에서 컴파일되어 진다는 의미를 함축하고 있으며, WIPI에서는 자바로 작성된 어플리케이션에만 적용되는 기술



Android Tip

안드로이드 자바 소스 얻기

이클립스의 소스 링크 기능 활용을 위한 sdk 자바 소스 얻기

1. Android 소스코드를 받는다.
2. 해당 폴더에서 옆의 파이썬 코드를 돌린다.
3. sources.zip 생성되며, android 자바코드가 package에 맞는 폴더로 압축되어 있다.
4. Sdk의 platforms 밑의 사용하는 sdk 버전 폴더 안으로 sources.zip을 이동한다.(ICS의 경우 android-14폴더)
5. 압축을 푼다. (sources 폴더로 풀린다.)
6. 이클립스를 실행시킨다.
7. Sdk가 ics로 되어 있는 상태에서 소스내의 소스를 보고 싶은 클래스를 선택하고 F3키를 누르면 android source를 볼수 있음.

http://blog.michael-forster.de/2008/12/view-android-source-code-in-eclipse.html

```
from __future__ import with_statement  # for Python < 2.6
import os
import re
import zipfile
# open a zip file
DST_FILE = 'sources.zip'
if os.path.exists(DST_FILE):
  print DST_FILE, "already exists"
  exit(1)
zip = zipfile.ZipFile(DST_FILE, 'w', zipfile.ZIP_DEFLATED)
# some files are duplicated, copy them only once
written = {}
# iterate over all Java files
for dir, subdirs, files in os.walk('.'):
  if dir.find('test') == -1:
     for file in files:
          if file.endswith('.java'):
            # search package name
            path = os.path.join(dir, file)
            with open(path) as f:
              for line in f:
                match = re.match(r'\s*package\s+([a-zA-Z0-9\._]+);', line)
                if match:
                  # copy source into the zip file using the package as path
                  zippath = match.group(1).replace('.', '/') + '/' + file
                  if zippath not in written:
                    written[zippath] = 1
                    zip.write(path, zippath)
                  break;

zip.close()
```

Android Tip

App에서 루트 권한 얻기

루팅후 (su 파일 변경 – Superuser.apk)

App에서 아래와 같은 코드로 실행

```
            // 앱개발시 코드 내에서 su를 통해 shell 명령으로 root권한으로 실행 가능

void checkPermission(String rootCommand){
    try {
        Process process = Runtime.getRuntime().exec("su");
        DataOutputStream outputStream = new DataOutputStream(process.getOutputStream());
        DataInputStream inputStream = new DataInputStream(process.getInputStream());
        outputStream.writeBytes(command + "\n");
        outputStream.flush();
        outputStream.writeBytes("exit\n");
        outputStream.flush();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

Android Tip

자바 decompile

Class 파일 ➔ java 파일로 변경

대표적인 툴 : JAD

이를 활용한 많은 GUI 툴 존재

추천 툴

- jd-gui-0.3.3.windows

  jar파일의 drag & drop만으로 decompile해서 보여줌.

  소스상에서 클래스 클릭시 해당 소스로 이동 기능

  http://java.decompiler.free.fr/

DEX decompile

안드로이드는 class가 아닌 dex 형태로 구성됨(apk 압축 해제시 classes.dex가 들어 있음)

이를 decompile하기 위해서는 dex ➔ jar 로 변환해야함

추천 툴

- dex2jar-0.0.7.9

  “dex2jar dex파일” 실행시 classes.dex.dex2jar.jar 생성

  이를 자바 decompile 툴로 decomile 가능

Android Tip

Xml decoder

안드로이드에서는 xml 파일을 binary 형태로 변경하여 apk 압축을 풀더라도 xml 파일을 text editor로 확인할 수 없음

Xml decoder 툴 : AXMLPrinter2.jar

Java –jar AXMLPrinter2.jar AndroidManifest.xml > result.txt

Redirection을 통해 result.xml 생성 ➔ text editor에서 확인 가능

APK 분석하기

1. 분석이 필요한 apk를 구한다.
2. 확장자를 zip으로 변경후 압축을 푼다.
3. AXMLPrinter2.jar를 이용해서 AndroidManaifest.xml을 decoding한 후 사용하는 intent나 receiver를 분석한다.
4. classes.dex를 dex2jar-0.0.7.9툴을 이용하여 jar로 변경한다.
5. Jar 파일을 jd-gui에 드레그앤드랍하여 클래스 구조 및 소스를 살펴본다.
6. Jd-gui의 “save all sources” 메뉴를 통해 소스로 뽑아낸다.(decomile해서 zip으로 묶어준다.)
7. 압축을 풀어서 source insight 등의 툴을 이용해서 분석한다.

Android Tip

Obfuscator

Decompile을 하더라도 소스코드를 알아보기 어렵게 만들어주는 툴

패키지, 클래스, 메서드, 변수 명을 a, b, c … 등으로 단순화 시킴

좋은 툴의 경우 한번 더 최적화를 해주며, 사용하지 않는 메서드나 클래스는 삭제시켜 준다.

장점 :

- 소스 코드를 알아보기 어렵게 만듦으로써 decompile을 방지한다.
- 이름을 줄임으로써 전체 바이너리 사이즈가 줄어든다.
- 최적화를 통해 사이즈가 줄어든다.

유의점 :

- 이름이 변경됨으로 exception 발생시 디버깅이 어렵다. (이름 변경 mapping table을 따로 유지하고 있어야 한다.)
- Reflection 이나 다른 모듈에 의해 사용되는 클래스나 메서드 명은 변경되지 않도록 해야 한다.

Android SDK에서 지원하는 Obfuscator 툴

- Proguard를 사용함
- 무료 툴이며, 다양한 기능을 제공한다.
- Apk build시 build.xml 수정을 통해 적용할 수 있다.

Android Tip

ADB port forward

adb forward <local> <remote>

Http 접속

adb forward tcp:8080 tcp:8080

브라우저 상에서 127.0.0.1:8080 으로 안드로이드 내의 web 서버 접속 가능

Ftp 접속

Ftp 서버를 안드로이드에 설치한후 passive 모드로 실행

Passive mode port range 설정후 설정된 만큼 port forward 설정

예) range 2121:2124 인 경우

adb forward tcp:21 tcp:21

adb forward tcp:2121 tcp:2121

adb forward tcp:2122 tcp:2122

adb forward tcp:2123 tcp:2123

adb forward tcp:2124 tcp:2124

추천 안드로이드 ftp server app : FTPDroid

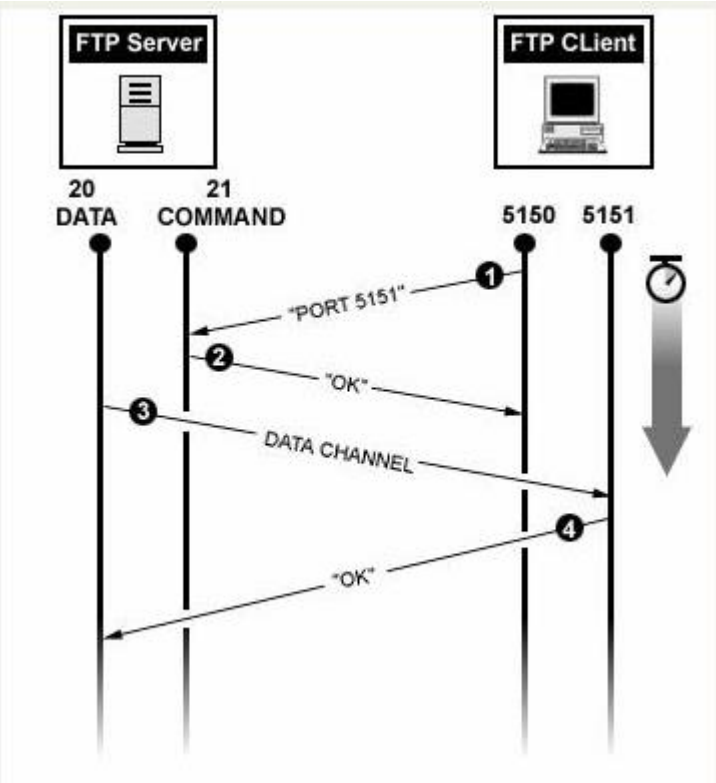
1024까지의 포트를 사용하기 위해서는 root권한을 필요로 한다. 따라서 ftp의 21번 포트는 루트 권한이 있다는 가정에서 가능하며, 아닌 경우 다른 포트로 설정하면 된다.

Ftp는 파일을 이동할 수 있기 때문에 루트권한이 있는 경우에만 시스템 폴더에 파일을 쓸수 있다.

Android Tip

FTP의 이해

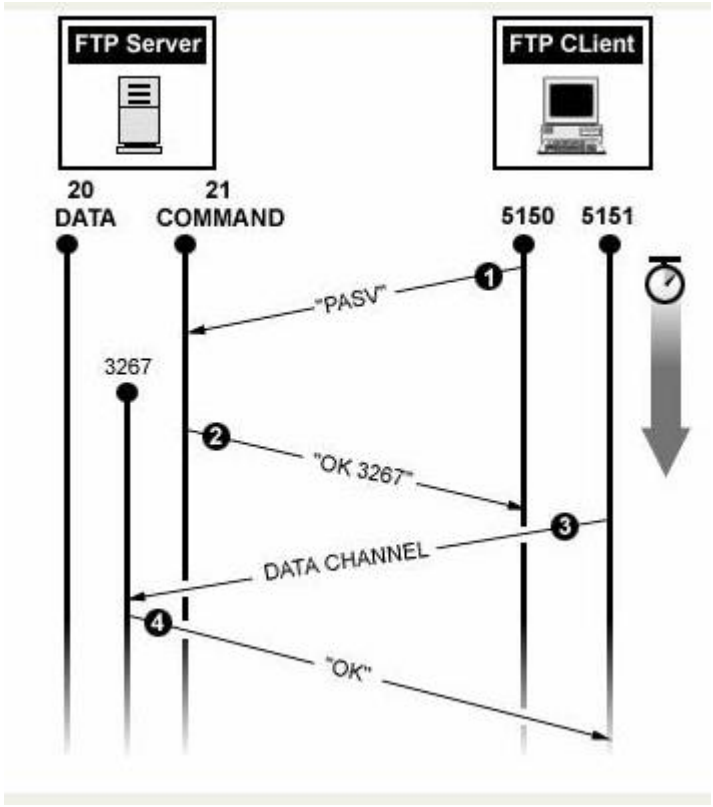
Active mode



1) 클라이언트에서 서버가 접속할 포트를 알려줌
2) Ack 신호 보냄
3) 서버는 20번 포트에서 알려준 포트로 접속 시도
4) 접속 성공

* 서버에서 클라이언트 접속을 시도함으로 방화벽으로 막혀있는 경우 문제 발생

Passive mode



1) 클라이언트가 command 포트로 접속 시도
2) 서버에서 서버가 사용할 포트를 알려줌
3) 클라이언트는 다른 포트를 열어 서버가 알려주 포트 접속
4) 접속 성공

* Port forward의 경우도 한쪽 방향으로만 지원함으로 adb forward를 위해서는 passive 모드만 가능함

Android Tip

벤더 제공 **API & Hidden API** 사용하기

Libs/LibAllShareInterface.zip

Eclipse > project properties > libraries > LibAllShareInterface.zip 추가

Ant 빌드시 기본 설정만으로 빌드됨

Runtime시 같은 이름의 클래스가 존재하는 경우 System class, Dex안의 class 중 어떤 것이 loading 될까?

Obfuscater 툴까지 돌린 경우 어떤 것이 loading 될까?

bootClassPathOut="project.target.class.path“

- Javac시 system class 설정 값

Androidjar = project.target.android.jar

- Javac시 class 설정값

Android Tip

벤더 제공 **API & Hidden API** 사용하기

Build.xml

```
<scriptdef name="resetpath" language="javascript">
   <attribute name="sep"/>
   <attribute name="name"/>
   <attribute name="exclude"/>
   <element name="path" type="path"/>
      <![CDATA[
    importClass(org.apache.tools.ant.types.Path);

    path = elements.get("path");
    separate = attributes.get("sep");
    name = attributes.get("name");
    exclude = attributes.get("exclude");
    length = path.size();
    strPath = null;
    for (i = 0; i < length; i++){
     items = "" + path.get(i);
     item = items.split(separate);
     for (j = 0; j < item.length; j++){
      if (exclude && item[j].indexOf(exclude) > 0) continue;
      if (strPath == null)
        strPath = item[j];
      else
        strPath += separate + item[j];
     }
    }
    if (strPath != null){
     newPath = new Path(project);
     newPath.setPath(strPath);
     project.addReference(name, newPath);
   }
      ]]>
</scriptdef>
```

```
<target name="-pre-compile">
<resetpath sep="${path.separator}" name="project.target.class.path">
<path>
<fileset dir="${jar.libs.absolute.dir}" includes="*.zip" />
<path refid="project.target.class.path" />
</path>
</resetpath>
</target>
```

NDK basic on Android

Android NDK

Android NDK

- Android NDK는 C, C++ 언어로 작성된 코드를 Java 언어에서 JNI (Java Native Interface)방법으로 호출가능하도록 컴파일하여 *.so 포맷의 라이브러리를 생성
- 생성된 라이브러리는 Android 플랫폼의 Java언어에서 로드할 수 있고 로드된 라이브러리 안에 존재하는 C, C++ 함수를 호출
- Android SDK에 의해 사용되는 것이 아니라, Cygwin을 설치/실행하여 Command line 상에서 Android NDK가 사용되어 C, C++코드가 컴파일되고 공유 라이브러리 파일(*.so)이 생성된다.

JNI (Java Native Interface)

- 자바가 다른 언어로 만들어진 어플리케이션과 상호 작용할 수 있는 인터페이스를 제공한다.
- 자바가상머신(JVM)이 원시 메소드(native method)를 적재(locate)하고 수행(invoke)할 수 있도록 한다
- JNI가 자바가상머신내에 포함됨으로써, 자바가상머신이 호스트 운영체제상의 입출력, 그래픽스, 네트워킹, 그리고 스레드와 같은 기능들을 작동하기 위한 로컬시스템호출(local system calls)을 수행할 수 있도록 한다.

NDK basic on Android

Android NDK

참고 강의자료

http://www.slideshare.net/pianoon/android-ndk-jni-15707268

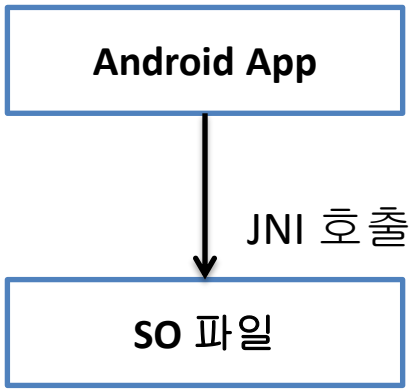
NDK basic on Android

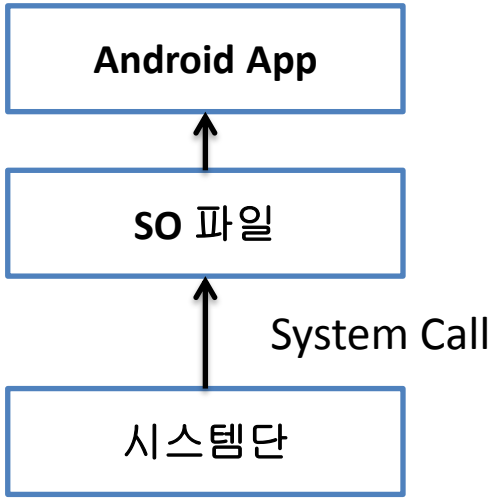
외부 소스 **Porting**

유의점

- 안드로이드는 리눅스 커널을 사용
- 안드로이드는 Bionic libc 사용 – POSIX function 일부 미지원
- 리눅스용 소스를 적용하기 위해서는 일부 수정필요

호출방식





NDK basic on Android

JNI – call back을 통한 자바 호출

```
class JVMScope
{
private:
    static JavaVM *_gJavaVM;
          …
public:
    JVMScope();
    ~JVMScope();

static void SetJVM(JavaVM *gJavaVM)
    {
        _gJavaVM = gJavaVM;
    }
}
```

1) Java VM을 static으로 선언
2) Jni library loading시 vm을 설정
3) 설정된 vm값을 이용 자바 콜

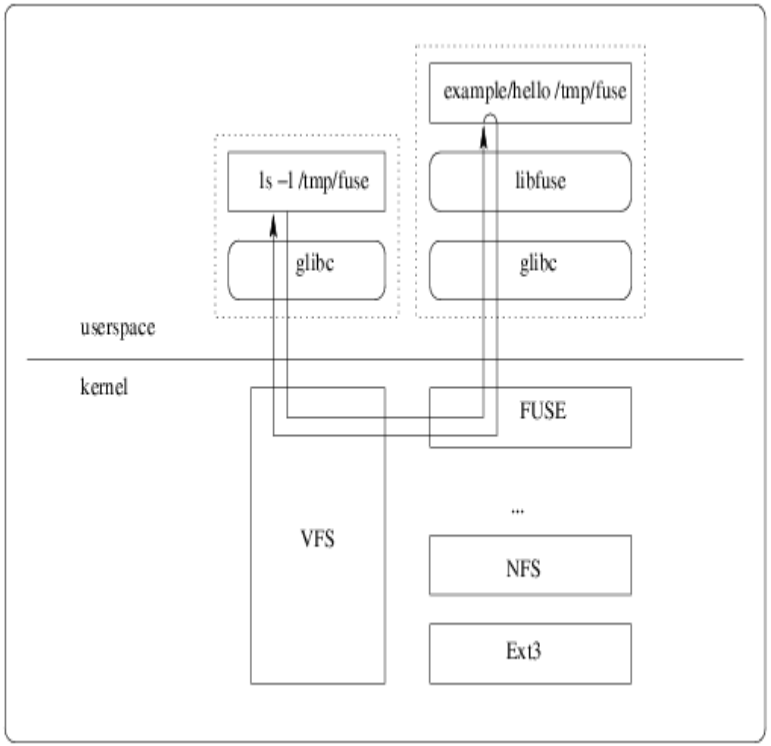
```
JNIEXPORT jint JNICALL JNI_OnLoad(JavaVM* vm, void * reserved)
{
    JNIEnv * env = 0;
    JVMScope::SetJVM(vm);
    return JNI_VERSION_1_4;
}
```

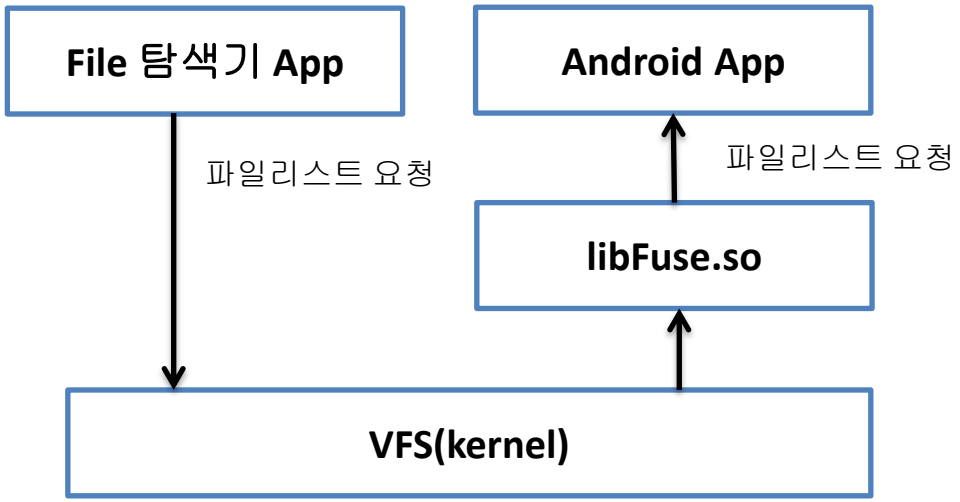
```
void PLATFORM_CLASS::HandlerVoid(FUNC_TYPE functype, int * integerVar, char** stringVar)
{
    int rtn =0;
    JVMScope jscope;
    JNIEnv *env = jscope.env();
    LOG_DEBUG("ContentPluginHandlerVoid functype %d", functype);

}
```

NDK basic on Android

FUSE (Filesystem in Userspace)





https://github.com/seth-hg/fuse-android

활용예

User 영역 메모리를 시스템영역 메모리와 합쳐서 app 설치영역 확장(안드로이드3.0이하)
웹스토리지 영역을 파일시스템으로 인식

NDK basic on Android

1. 커널 빌드시 옵션 변경

커널 빌드 옵션파일인 .config 파일에 fuse enable 옵션을 추가하여 빌드 (enable 되어 있지 않은 경우)

2. Fuse 포팅

3. 환경설정

```
./adb wait-for-device
./adb remount
./adb push fusermount /system/xbin
./adb shell chmod 4755 /system/xbin/fusermount
./adb shell chmod 666 /dev/fuse
```

- fusermount를 복사후 권한을 4755로 변경
- /dev/fuse의 권한 변경

Setuid bit

```
$ chmod 4755 myfile
$ ls -ld myfile

-rwsr-xr-x   root  root  96 Dec 8 12:53 myfile
```

Myfile는 누가 실행시키거나 상관없이 myfile own권한으로 실행시킬 수 있다.

Q & A