Introduction to Android

Youngjin Cho

What is Android

An open-source software stack created for mobile phones and other devices

- includes an operating system, middleware and key applications

The Android Open Source Project (AOSP), led by Google

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Features

Dalvik virtual machine optimized for mobile devices

Integrated browser based on the open source *WebKit engine*

Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification**SQLite** for structured data storage

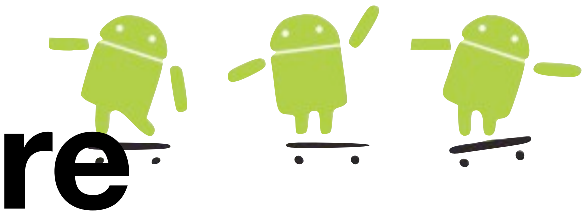
Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

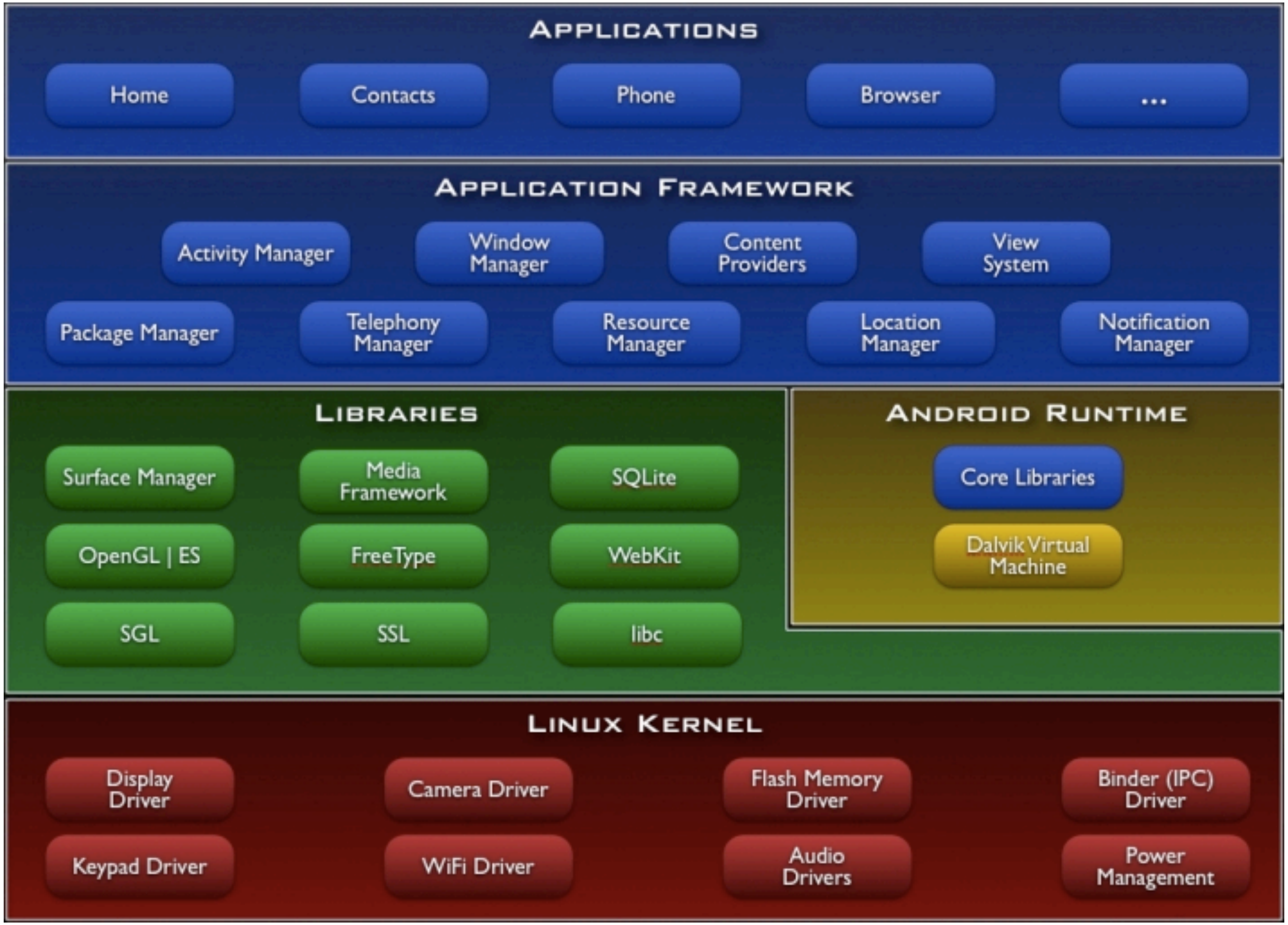
Bluetooth, EDGE, 3G, and WiFi

Camera, GPS, compass, and accelerometer

Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Android Architecture





Applications

Home, Contacts, Phone, Browser, ...

Application Framework

Activity Manager, Window Manager, Content Providers, View System

Package Manager, Telephony Manager, Resource Manager, Location Manager, Notification Manager

Libraries

Surface Manager, Media Framework, SQLite

OpenGL | ES, FreeType, WebKit

SGL, SSL, libc

Android Runtime

Core Libraries

Dalvik Virtual Machine

Linux Kernel

Display Driver, Camera Driver, Flash Memory Driver, Binder (IPC) Driver

Keypad Driver, WiFi Driver, Audio Drivers, Power Management

Applications

Android will ship with a set of core applications

- Email client
- SMS program
- Calendar
- Maps
- Browser
- Contacts, and others

All applications are written using the **Java** programming language.

Application Framework

Developers have full access to the same framework APIs used by the core applications.

Designed to simplify the reuse of components

Any application can publish its capabilities and any other application may then make use of those capabilities.

A rich and extensible set of Views

Content Providers

A Resource Manager

A Notification Manager

An Activity Manager

Libraries

Android includes a set of C/C++ libraries

- **System C library** - tuned for embedded Linux-based devices
- **Media Libraries** - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- **LibWebCore** - a modern web browser engine
- **SGL** - the underlying 2D graphics engine
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration
- **FreeType** - bitmap and vector font rendering
- **SQLite** - a powerful and lightweight relational database engine

Android Runtime

Android includes **a set of core libraries** that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the **Dalvik virtual machine**. Dalvik has been written so that a device can run multiple VMs efficiently.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Linux Kernel

Android relies on Linux version 2.6 for core system services

- Security
- Memory management
- Process management
- Network stack
- Driver model.

The kernel also acts as an **Hardware Abstraction Layer**

Android
Open Source Project

Youngjin Cho

Get Android Source Code

Setting up local work environment

How to use **Repo** and **Git** to get the Android files

How to build the files on your machine

To see snapshots and histories of the files available in the public Android repositories using GitWeb web interface.

The source is approximately 2.1GB in size.
You will need 6GB free to complete the build.

Setting up your machine

To set up your Linux development environment, make sure you have the following:

- Required Packages:
  - Git 1.5.4 or newer
  - JDK 5.0, update 12 or higher. Java 6 is not supported.
  - flex, bison, gperf, libsdl-dev, libesd0-dev, libwxgtk2.6-dev (optional), build-essential, zip, curl.
  - build-essential zip curl libncurses5-dev zlib1g-dev
  - You might also want Valgrind, a tool that will help you find memory leaks, stack corruption, array bounds overflows, etc.

Windows is not currently supported.

64-bit x86 Linux system has not been as well tested

Mission #1

Prepare the required packages

```
$sudo apt-get install git-core gnupg flex bison gperf
libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip
curl libncurses5-dev zlib1g-dev
$ sudo apt-get install valgrind
$ sudo apt-get install lib32readline5-dev
```

Download **Repo**

```
$cd ~
$mkdir bin
$echo $PATH
  #Download the repo script and make sure it is executable:
$curl http://android.git.kernel.org/repo >~/bin/repo
$chmod a+x ~/bin/repo

  #add ~/bin into $PATH
$source ~/.profile
$repo
```

Mission #1 (con’t)

Prepare JDK5 and JDK6

```
#java5
$sudo add-apt-repository "deb http://us.archive.ubuntu.com/ubuntu/ jaunty multiverse"
$sudo add-apt-repository "deb http://us.archive.ubuntu.com/ubuntu/ jaunty-updates multiverse"
$sudo apt-get update
$sudo apt-get install sun-java5-jdk

#java6
$sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"
$sudo apt-get update
$sudo apt-get install sun-java6-jdk

$sudo update-java-alternatives -l
java -version
$sudo update-java-alternatives -s java-1.5.0-sun
java -version
```

Mission #1 (con’t)

Create an empty directory to hold your working files:

```
$ mkdir mydroid
$ cd mydroid
```

Run "repo init" to bring down the latest version of Repo with all its most recent bug fixes. You must specify a URL for the manifest:

```
$ repo init -u git://android.git.kernel.org/platform/
manifest.git
```

If you would like to check out a branch other than "master", specify it with -b, like:

```
$ repo init -u git://android.git.kernel.org/platform/
manifest.git -b cupcake
```

To pull down files to your working directory from the repositories as specified in the default manifest, run

```
$ repo sync
```