

はじめようAndroid

株式会社センシンソフト

目次

- ◆概要
- ◆開発環境の構築
- ◆アーキテクチャ概要
- ◆アプリケーション フレームワーク層
- ◆ライブラリ層
- ◆ランタイム層
- ◆カーネル層
- ◆ライセンス規約
- ◆スキルアップしよう

概要

- ◆ Androidとは、米グーグル社（Google Inc.）が、モバイル向けプラットフォームとして発表した、無償で誰にでも提供されるオープンソースのオペレーションシステム、ミドルウェア、キーモバイルアプリケーションを含むモバイル端末のソフトウェアスタックです。
- ◆ Androidは、オープンソースのソフトウェアでAndroid上で作成したアプリケーションはAndroidが搭載されている機器ならば、どこでも動作します。
- ◆ Android用のアプリケーションは、基本的にはDalvik仮想マシン（VM）と呼ばれるAndroid OS上で動作し、プレ・インストール・アプリケーションと、後からインストールするアプリケーションを、公平に扱うのが特徴です。
- ◆ Androidにはローカライズ機能がすでに入っているため開発時にローカライズを強く意識することなく開発する事が出来ます。

概要

◆ Androidは携帯電話以外のネットワークやハードウェアにも対応しています。
主な機能は以下の物があります。

- ・ Bluetooth
- ・ 3G
- ・ WiFi
- ・ カメラ
- ・ GPS
- ・ コンパス
- ・ 加速度計

上記の機能を組み合わせることで、現実環境にコンピュータを用いて情報を付加提示するAR (Augmented Reality : 現実拡張) 機能を持つアプリケーションの作成が可能です。

開発環境の構築

◆ Androidアプリケーションを作成するためには以下の開発ツールを使用します。

- ・ **JDK** (Java Development Kit)

 - <http://java.sun.com/javase/ja/6/download.html>

- ・ **ADT** (Android Development Tools)

 - <http://developer.android.com/intl/ja/sdk/eclipse-adt.html>

- ・ **Android SDK**

 - <http://developer.android.com/intl/ja/sdk/index.html>

- ・ **Eclipse**

 - <http://www.eclipse.org/downloads/>

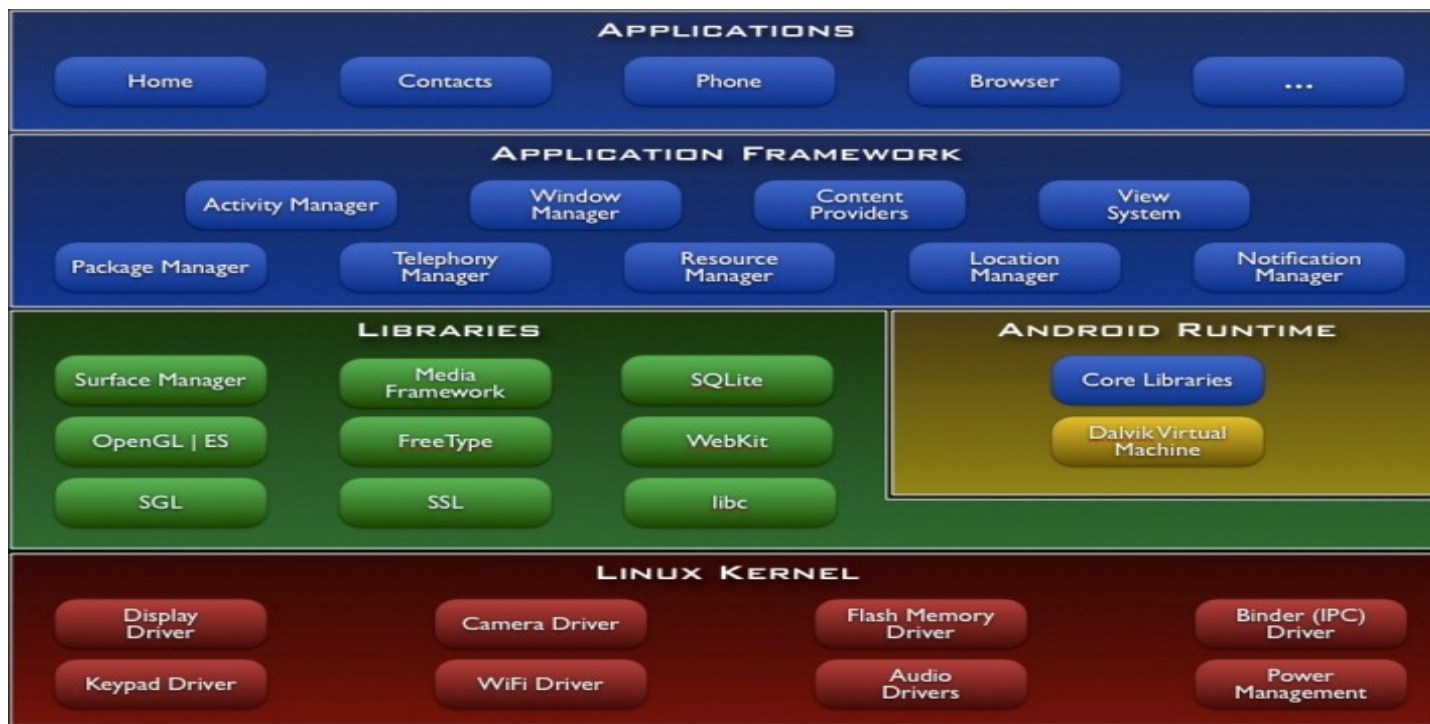
基本的には、JDKを入れてある環境でEclipseにADTを入れAndroid SDKを設定するだけで開発環境となります。

◆ またEclipseを日本語化する際には別途ツールを使用します。
例としては、以下のようなものが挙げられます。

- ・ **Babel Project**

アーキテクチャ概要

- ◆ Androidは、オペレーティングシステム、ミドルウェア、主要アプリケーションを含むモバイルデバイス用のソフトウェアの集まりであり次の図は、オペレーティングシステムの主要コンポーネントを示しており、概念的には4階層構造で構成されています。



※ <http://developer.android.com/guide/basics/what-is-android.html> より図を抜粋.

アーキテクチャ概要

- ◆ 一般的なAndroidアプリケーションを開発する場合のアプリケーション自体は、最上位層のApplication層に当たり、ApplicationFramework層のAPI群、Libraries層のミドルウェア群を使用した開発手法となります。
- ◆ AndroidRuntime層はAndroidのVirtualMachineアーキテクトであり、Java仮想マシンで言うところのVMに当たります（実際にはJavaVMとDalvikVMは異なるアーキテクトである）。
- ◆ LinuxKernel層はAndroidの最も下位に位置するものであり、Linuxカーネルをベースに作成されています。
- ◆ 一般的なアプリケーション開発で、AndroidRuntime層、LinuxKernel層を開発者が意識することは殆どありません、尚、Androidには、電子メールクライアント、SMSプログラム、カレンダー、地図、ブラウザ、連絡先、およびその他を含むコアアプリケーションがApplicationFramework層としてセットで出荷されています。

アプリケーション フレームワーク層



※ <http://developer.android.com/guide/basics/what-is-android.html> より図を抜粋.

- ◆ この層では主に、アプリケーションの起動から終了までの流れ、すなわちライフサイクルの管理を行っています。
- ◆ ユーザーインターフェースの表示やユーザーによる操作など携帯端末で起こるさまざまな状態の変化を各アプリケーションに伝える手段も提供しています。

アプリケーション フレームワーク層

- **Activity Manager**

アプリケーションのライフサイクルを管理し、共通のナビゲーションスタック（画面遷移時の引き渡し情報など）を提供します。

- **Window Manager**

ウィンドウを管理するためのものであり、それ自体だけでは何もできないが、他のインスタンスと連携してAndroidのウィンドウ全般に対する管理を行います。

- **Content Providers**

独自のデータを共有したり、特定のアプリケーションが保持するデータに対して他のアプリケーションからアクセスできるようにしたものです。

- **View System**

アプリケーションを構築するために使用できるリッチで拡張可能なビューのセットであり、リスト、グリッド、テキストボックス、ボタン、組み込み可能なウェブブラウザなどがそれです。

アプリケーション フレームワーク層

- **Package Manager**

システムにロードされたアプリケーションの情報を保存・変更・削除という一連の管理機能を有し、アプリケーションに提供します。

- **Telephony Manager**

電話機能の管理を提供しています (Radioデバイス (GSM、EDGE、3G) など)。

- **Resource Manager**

画面に使うデザインや画像データ、文字データを「リソース」としてプログラムから分離し、アプリケーションから動的に参照する機能を提供するものであります。

ローカライズなどもこの分野に入ります。

アプリケーション フレームワーク層

- **Location Manager**

GPSを使用し位置情報の管理を提供します。

- **Notification Manager**

有効なすべてのアプリケーションはステータスバーでカスタムアラートを表示することができる。

ライブラリ層



※ <http://developer.android.com/guide/basics/what-is-android.html> より図を抜粋.

- ◆ Androidにはシステム内で使用されるCまたはC++で開発されたライブラリを標準で含んでいます。これらのライブラリの機能は、アプリケーションの開発過程で使うが、直接操作することではなく、一般的には各アプリケーションのフレームワークを経由して利用します。

ライブラリ層

● *SurfaceManager*

「SurfaceManager」は、グラフィックレイヤーを合成して一つの画面に表示する仕組みをアプリケーションに提供する描画ライブラリであります。グラフィックレイヤーを使うと、アプリケーションを経由しないで画面表示が可能のため、高速に描画できる。アプリケーションからは、主にカメラやGPU (GraphicsProcessingUnit) といったハードウェアからのデータを高速に描画するために利用します。

● *MediaFramework*

画像の表示や音声・動画のエンコードとデコードの処理を実行するライブラリ。米PacketVideo社のメディア再生用ライブラリ「OpenCORE」をベースに開発されている。対応するフォーマットとして、音声では「mp3」や「aac」などがあり、動画では「h.263」や「h.264」などがあります。

ライブラリ層

● SQLite

高速なリレーショナルデータベースエンジンでありデータ管理の機能を提供します。格納する電話番号を比較するなど、Android用に文字列を処理する機能が追加されています。

アプリケーションからは、フレームワークを介してリレーショナル・データベースとして利用します。フレームワークにある「ContentProvider」のようなアプリケーション間のデータ共有の仕組みの中でも利用されています。

● OpenGL/ES

ハードウェア・アクセラレータを使った高速な3D描画を実現する3Dグラフィックライブラリ。

OpenGL ES 1.0 APIに準拠しており、ハードウェアがそれに対応している場合、ライブラリは高度に最適化された3Dアクセラレーション、3Dソフトウェアラスタライザのいずれかを使用できます。

ライブラリ層

● *FreeType*

FreeTypeとはフォントエンジンを実装したライブラリであります。Androidの標準フォントには、FreeTypeを使用しており、他のフォント形式もサポートしています。

● WebKit

HTMLまたはXHTMLの描画を担当するライブラリ。内部に高速なJavaScriptエンジンである「SquirrelFish」が搭載されている。開発者がアプリケーションの画面にWebページを表示する場合に利用する画面部品（ウィジェット）である「WebView」にも機能を提供している。

● *SGL*

Androidでのグラフィックライブラリには、2Dグラフィックライブラリ、3Dグラフィックライブラリが存在するが、そのうちの2Dグラフィックライブラリ部を総称してSGLと呼んでいます。

ライブラリ層

● libc (bionic)

組込みLinuxベースのデバイス用にチューニングされたAndroid用のCライブラリであります。BSDのCライブラリを基に開発されており、x86とARMに対応している。標準規格に沿ったlibcと比べ、ロケール（各国語対応）やマルチバイト文字に対応していないことやC++の例外に対応していないなど、最低限の実装のみサポートしている。携帯端末上での動作を最適化するために、「機能を限定してライブラリが肥大化してしまうことを防止する」という方針で開発されています。

ランタイム層



※ <http://developer.android.com/guide/basics/what-is-android.html> より図を抜粋.

- ◆ ランタイム層は基本的なAPIを提供するコアライブラリと仮想マシン「Dalvik VM」で構成されています。
- ◆ 一般的なアプリケーション開発で、AndroidRuntime層を開発者が意識することは殆どありません。

カーネル層



※ <http://developer.android.com/guide/basics/what-is-android.html> より図を抜粋.

- ◆ LinuxKernel層はAndroidの最も下位に位置するものであり、Linuxカーネルをベースに作成されています。
- ◆ 一般的なアプリケーション開発で、LinuxKernel層を開発者が意識することは殆どありません。

ライセンス規約

- ◆ Androidは、オープンソースのソフトウェアですが、オープンソースには数多くのライセンスが存在します。
全てのライセンスに共通しているのが以下の2点です。

1. 著作権表示の保持。原作者のクレジットを表記することにより、利用する第三者が作ったように偽ることを防ぐ。
2. 無保証であること。原作者は、葉生物を含め、その動作・品質を保証しない。

また、次の点がライセンスにより異なり、

1. ソースコードを開示する範囲。
2. 改変・複製したものを再配布する際、どのようなライセンスを適用させるか。

その異なる内容により、以下の3つに分類されます。

- ・ GPL系
- ・ MPL系
- ・ BSD系

スキルアップしよう

- ◆ 世の中のソフトウェア製品にそれぞれの認定試験制度があるように、Androidにも「Android技術者認定試験制度」という独自の認定試験が存在します。

Androidの知識を取得したら・・・又は知識を取得することを目的として認定試験を受験するのもよいかもしれません。

※2012年2月現在、以下の構成で提供されています。

Android 技術者認定試験制度 (<http://www.oesf.jp>)

アプリケーション技術者試験

- Android プログラミング
- Java プログラミングの知識
- スクリプト言語
- オブジェクト指向
- Android 基礎知識 (アプリ視点)

プラットフォーム技術者試験 (開発中)

Linuxの知識
ライブラリ概要
Bionic
HAL
起動シーケンス
Android 基礎知識 (プラットフォーム視点)

共通知識

開発環境 (Eclipse、SDK)
NDK、JNI
デバッグ技術
デバッグ手法