

嵌入式笔记 - 第九章

这章讲讲Android

Android应用组件

Android开发四大组件分别是：

活动 (Activity)：用于表现功能。

服务 (Service)：后台运行服务，不提供界面呈现。

广播接收器 (BroadcastReceiver)：用于接收广播。

内容提供商 (Content Provider)：支持在多个应用中存储和读取数据，相当于数据库。

Activity

Android 中，Activity是所有程序的根本，所有程序的流程都运行在Activity 之中，Activity可以算是开发者遇到的最频繁，也是Android 当中最基本的模块之一。在Android的程序当中，Activity 一般代表手机屏幕的一屏。如果把手机比作一个浏览器，那么Activity就相当于一个网页。在Activity 当中可以添加一些 Button、Check box 等控件。可以看到Activity 概念和网页的概念相当类似

Service

Service不能自己运行，只能后台运行，并且可以和其他组件进行交互。Service 是没有界面的长生命周期的代码。

BroadcastReceiver

在Android 中，Broadcast是一种广泛运用的在应用程序之间传输信息的机制。而BroadcastReceiver 是对发送出来的Broadcast进行过滤接受并响应的一类组件

Content Provider

Content Provider 是Android提供的第三方应用数据的访问方案。主要是数据库

Android安全权限机制

Android安全架构的核心设计思想是，在默认设置下，所有应用都没有权限对其他应用、系统或用户进行较大影响的操作。这其中包括读写用户隐私数据（联系人或电子邮件），读写其他应用文件，访问网络或阻止设备待机。

Android 应用程序权限申请：每个应用程序的 APK 包里面都包含有一个 AndroidManifest.xml 文件，该文件除了罗列应用程序运行时库、运行依赖关系等之外，还会详细地罗列出该应用程序所需的系统访问。程序员在进行应用软件开发时，需要通过设置该文件的 uses-permission 字段来显式地向 Android 系统申请访问权限。

用户安装的时候可以分析一款应用所需权限，从而简单判定这款应用是否安全。

Android开发环境

基础：JDK(Java Development Kit)

IDE：Eclipse/Android Studio

ADT (Android Development Toolkit)

Android SDK

Android系统特性

源代码完全开放，比其他系统更吸引设备制造商采用。

手机、PDA、GPS、平板电脑、MP4、智能家电、智能手表等众多设备均可以安装该系统,

内置Google独特业务、工具（搜索，导航，Gmail，Google Talk，地图，语音搜索）

未来Google云计算技术

更好的硬件层调用（GPS模块，Wifi模块，重力感应，3D加速）

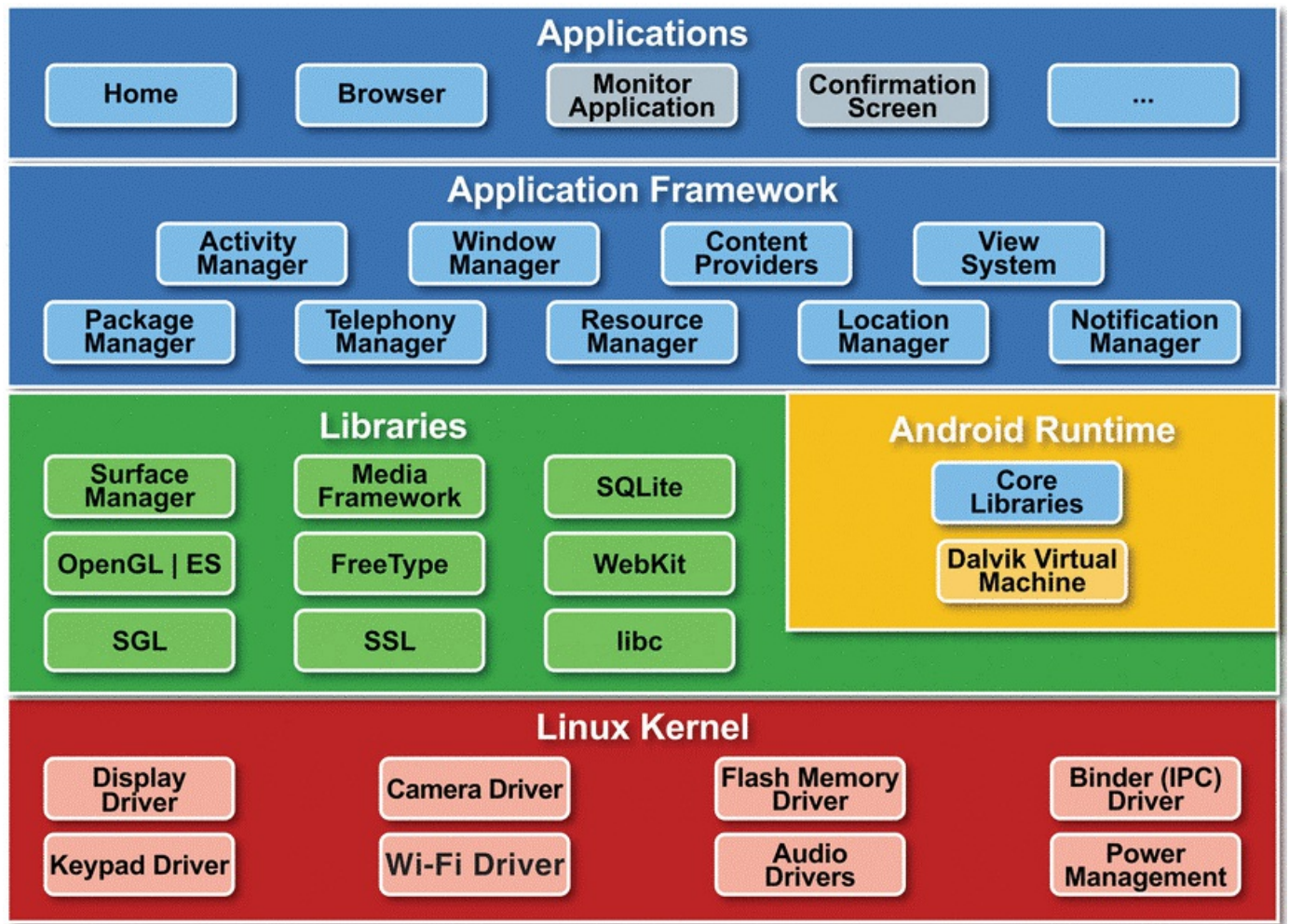
可以使用JAVA基于Android SDK开发上层应用、游戏

也可以使用C++基于Android NDK开发底层和对效率要求较高的核心算法

iOS VS Android

| 比较 | IOS | Android |
|--------|----------------------|---|
| 开发平台 | Apple Mac OS | 不局限于操作系统 |
| 开发工具 | Xcode | Eclipse |
| 开发语言 | Objective - C | Java |
| 兼容性 | 封闭操作系统，由Apple制定，兼容性高 | Google规定出硬件标准，由不同的厂商进行手机的研发，由于厂商众多，所以兼容性低 |
| UI交互界面 | 主要依靠触屏完成 | 需要触屏和按键同时操作 |
| 显示风格 | 统一的视觉规范和分辨率 | 视觉规范由厂商决定，屏幕分辨率繁多 |

Android系统架构



APPLICATIONS(应用程序)

Android平台默认包含了主要的应用程序，包括电子邮件、短信、日历、地图、浏览器、联系人等，这些程序都是用Java语言来编写的，当然也可以用自己编写的软件来替代Android提供的程序。

APPLICATION FRAMEWORK(应用程序框架)

Android应用程序框架是开发人员进行开发的基础。首先了解一下应用程序框架所包含的内容。它包括9大部分：视图系统、内容提供器、窗口管理器、活动管理器、通知管理器、位置管理器、资源管理器、电话管理器和包管理器。下面逐一介绍各

部分的功能。

- 视图系统 (View System) : 用来构建应用程序的基本组件, 包括文本框、按钮、列表等, 甚至内嵌的网页浏览器。
- 内容提供器 (Content Provider) : 提供了程序之间数据的共享机制。
- 窗口管理器 (Window Manager) : 管理所有的窗口程序。通过窗口管理器提供的接口我们可以向窗口中添加、删除View。
- 活动管理器 (Activity Manager) : 管理应用程序生命周期, 并提供导航回退功能。
- 通知管理器 (Notification Manager) : 使所有的程序能够在状态栏显示自定义的警告。需要注意的是, 在手机上状态栏位于屏幕的顶部。
- 位置管理器 (Location Manager) : 用来提供位置服务。
其中包括两种技术: GPS定位技术和网络定位技术。
- 资源管理器: (Resource Manager) : 提供各种资源让应用程序使用, 如布局文件、图片、音频文件等非代码资源。
- 电话管理器 (Telephone Manager) : 管理所有的移动设备功能。
- 包管理器 (Package Manager) : 主要用于系统内的程序管理。

在Android平台中, 开发人员可以完全访问核心的应用程序框架所包含的API, 可以自由地利用设备硬件优势, 访问位置信息、运行后台服务、设置闹钟、向状态栏添加通知等来开发出更多实用和新颖的程序。同时, Android平台在设计时就考虑到了组件的重用, 基于这种机制, 用户就可以方便地替换平台本身所提供的各种应用组件。开发人员在开发Android平台上的应用程序时, 也可以使用新的软件组件, 并将该软件组建放入Android的应用程序框架中。

LIBRARIES(系统库)

应用程序框架是贴近于应用程序的软件组件服务, 而更底层则是Android的函数

库。

- Surface Manager：在同时执行多个应用程序时，Surface Manager会负责管理显示与存取操作之间的互动，并且为应用程序提供2D和3D图层的无缝融合。
- 媒体函数库：基于 PacketVideo
- OpenCORE；该库支持录放，并且可以录制许多流行的音频、视频格式，还有静态影像文件，包括MPEG4, MP3, AAC, AMR, JPG, PNG。
- SQLite：它是一个对于所有应用程序可用，轻量级的关系型数据库引擎。
- Free Type：提供点阵字和向量字的描绘显示。
- Web Kit：一个最新的web浏览器引擎用来支持Android浏览器和一个可嵌入的web视图。WebKit是一个开源的项目，许多浏览器也都是用WebKit引擎所开发成的。如诺基亚S60手机内的浏览器。
- SGL：提供Android在2D绘图方面的绘图引擎。
- Lib C：一个从 BSD 继承来的标准 C 系统函数库（libc），专门为基于嵌入式Linux的设备定制。
- OpenGL ES：提供OpenGL ES 1.0 APIs实现；该库可以使用硬件3D加速（如果可用）或者使用高度优化的3D软加速。

ANDROID RUNTIME(Android运行时)

Android虽然采用Java语言来编写应用程序，但是它并不使用来执行Java程序，而是采用Android自用的Android运行时。Android运行时包括**核心库**和**Dalvik虚拟机**两部分。

核心库(Core Libraries)：核心库已经包含两部分内容，一部分为绝大多数Java程序语言所需要调用的功能函数，如java.io等；

另外一部分是Android的核心库，如android.os,android.net等。

Dalvik虚拟机：Dalvik虚拟机是Google公司自己设计的用于Android平台的Java虚拟机。它是专门为移动设备而设计的，在开发的时候就考虑到了用最少的内存资源来执行。在设计的时候Dalvik虚拟机很多地方参考了Java虚拟机的设计，但是它并不支持Java虚拟机所执行的Java字节码，也不直接执行java的类文件。它可以支持已转换为.dex（即Dalvik Executable）格式的Java应用程序的运行，.dex格式是专为Dalvik设计的一种压缩格式，适合内存和处理器速度有限的系统。

Dalvik虚拟机与Java虚拟机的最大不同在于Java虚拟机是基于栈（stack-based），而Dalvik基于寄存器（register-based）。基于寄存器的虚拟机的其中一个优点是所需要的资源相对较少，在硬件上也会比较容易实现。

LINUX KERNEL(Linux内核)

Android平台中的操作系统采用了Linux2.6版本的内核，它包括了显示驱动、摄像头驱动、Flash内存驱动、Binder(IPC)驱动、键盘驱动、WIFI驱动、Audio驱动以及电源管理部分。它作为硬件和软件应用之间的硬件抽象层，使得应用程序开发人员不需关心硬件细节。但是对于硬件开发商而言，如果想要Android平台运行到自己的硬件平台上就必须对Linux内核进行修改，为自己的硬件编写驱动程序。