

## 第一章 关键技术讨论

### 1.1 Android 平台

#### 1.1.1 Android 起源及发展状况

*Android* 一词最早出现于法国作家利尔亚当在 1886 年发表的科幻小说《未来夏娃》中。他将外表像人的机器起名为 *Android*。2010 年 2 月 3 日, Linux 内核开发者 Greg Kroah-Hartman 将 *Android* 的驱动程序从 Linux 内核“状态树”(“staging tree”)上除去, 从此, *Android* 与 Linux 核心开发分道扬镳。

*Android* 是 Google 开发的基于 Linux 平台的开源手机操作系统。它该平台由操作系统、中间件、用户界面和应用软件组成, 是一个全面整合的软件栈, 号称是首个为移动终端打造的真正开放和完整的移动软件平台。you 谷歌与开放手机联盟合作开发了 *Android*, 这个联盟由包括中国移动、摩托罗拉、高通、宏达和 T-Mobile 在内的 30 多家技术和无线应用的领军企业组成。通过与运营商、设备制造商、开发商和其他有关各方结成深层次的合作伙伴关系, 我们希望借助建立标准化、开放式的移动电话软件平台, 在移动产业内形成一个开放式的生态系统。我们认为此举必将推进更好、更快的创新, 为移动用户提供不可预知的应用和服务。

*Android* 作为谷歌企业战略的重要组成部分, 将进一步推进“随时随地为每个人提供信息”这一企业目标的实现。由数据表明, 全球为数众多的移动电话用户从未使用过任何基于 *Android* 的电话。谷歌的目标是让移动通讯不依赖于设备甚至平台。出于这个目的, *Android* 将会将其补充, 而不会替代谷歌长期以来奉行的移动发展战略: 通过与全球各地的手机制造商和移动运营商结成合作伙伴, 开发既有用又有吸引力的移动服务, 并推广这些产品。开放手机联盟的成礼和 *Android* 的推出是对现状的重大改变, 在带来初步效益之前, 还需要不小的耐心和高昂的投入。但是, 我们认为全球移动用户从中能获得的潜在利益是值得付出这些努力的。如果你也是一个开发者, 并对我们的想法感兴趣, 就请再给我们一星期的时间, 届时谷歌便能提供 SDK 了。

Google 与开放手机联盟 (Open Handset Alliance) 合作开发了 *Android* 移动开

发平台，这个联盟由摩托罗拉、高通、宏达电和 T-Mobile、中国移动等在内的 30 多家移动通讯领域的领军企业组成。Google 与运营商、设备制造商、开发商和其他第三方结成了深层次的合作伙伴关系，希望通过建立标准化、开放式的移动电话软件平台，在移动产业内形成一个开放式的生态系统。

Android 作为 Google 企业战略的重要组成部分，将进一步推进“随时随地为每个人提供信息”这一企业目标的实现。全球为数众多的移动电话用户从未使用过任何基于 Android 的移动通讯设备，Google 的目标是让移动通讯不依赖于设备甚至平台。出于这个目的，Android 将补充而不会代替 Google 长期以来奉行的移动发展战略：通过与全球各地的手机制造商和移动运营商结成合作伙伴，开发即有用又有吸引力的移动服务，并推广这些产品。

### 1.1.2 Android 平台及其优点

Android 是开源的，开源是一种的完全开放的开发模式，有着很多移动终端平台的优点，主要有以下几点：

#### 1. 简单性

开源软件解决方案很容易找到和很容易实施，许多架构师和开发人员都熟悉这个技术的架构。开源软件团体推动开源软件开发人员提供使用方便的框架和平台。开源软件解决方案还能够让企业迅速创建一些解决方案以提供有形的和可衡量的好处。

#### 2. 开放性

开源软件本身的灵活性允许比专有软件产品更大的自由和个性化。这就意味着一个机构能够从开源软件的安装中看到与自己的业务关系更密切的更大的价值。

#### 3. 价格负担低

Android 是一款基于 Linux 平台的开源操作系统，从而避开了阻碍市场发展的专利壁垒，是一款完全免费的智能手机平台。而 Windows Mobile 高达 20 多美元的单台授权费相比，采用 Android 系统的终端可以有效的降低产品成本；Android 系统对第三方软件开发商也是完全开放和免费的。Android 系统承载着 Google 帝国的梦想。

### 1.1.3 Android 系统架构原理

对操作系统而言,必须做到设计合理、层次分明,同时还需考虑整个系统的结构要聚耦适当,Android 系统是基于 Linux 内核的,因此还必须具备开源的特性,以符合开源人员共同工作。

从系统的组成要件来讲,Android 平台架构包括硬件设备、板级支持包、驱动程序、操作系统内核、程序运行库,运行框架,应用程序等,它们的有机结合和协同工作共同完成了整个系统的正常运行和对事务的处理。

依据 Google 开源资料可知,整个系统由 Linux 内核、程序库、Android Runtime、应用程序框架和应用程序等 5 部分组成,系统架构如图1.1所示。

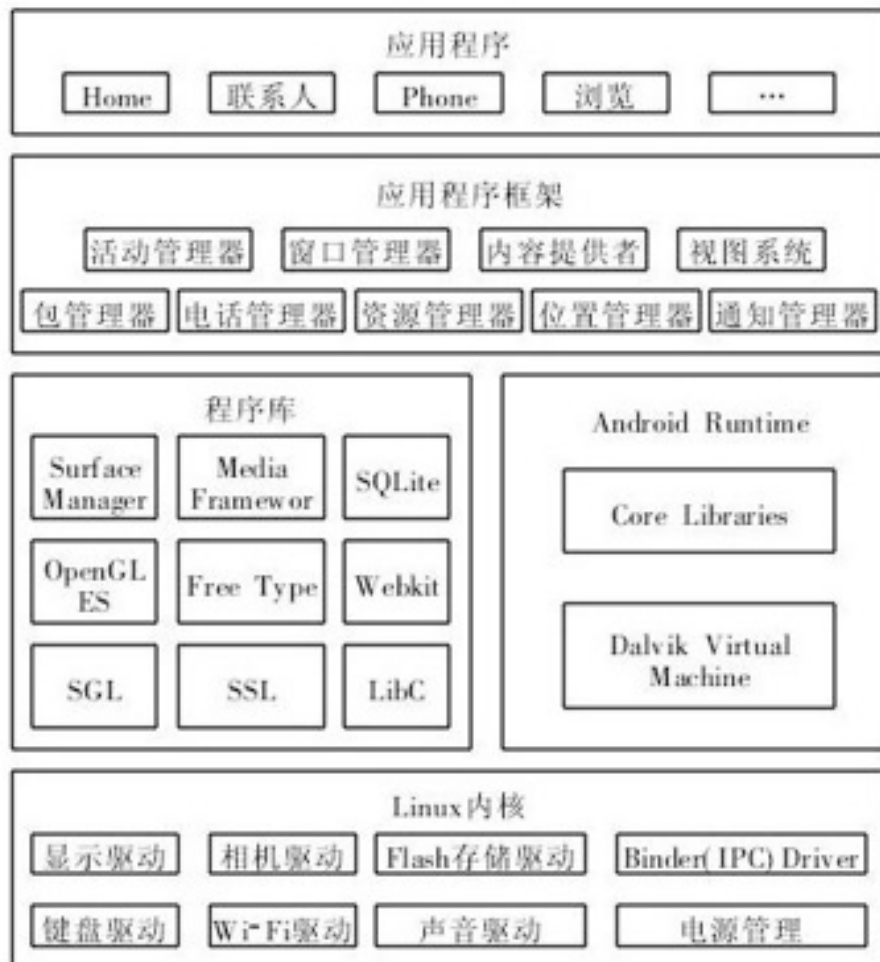


图 1.1 Android 系统架构图

参照图1.1,由上而下对组成系统各部分的主要组件作以下描述。

#### 1. Linux 内核

Android 基于 Linux 2.6 内核, 但并非完全照搬内核, 而是对内核作了部分增删和修改, 在 Linux 2.6 内核的基础上, Android 核心系统实现了安全性、内存管理、进程管理、网络协议栈和驱动模型等功能, Linux 内核也同时作为硬件和软件栈之间的抽象层。

硬件驱动程序: 完成与各种硬件的通信, Linux 内核提供了大部分设备的驱动程序, 如显示屏, 摄像头, 内存, 键盘, 无线网络, 音频设备, 电源等组件。

系统内存管理: 对所有可用的内存进行统一编码管理, 定义一整套内存定位, 使用与回收的策略。

系统进程管理: 内核管理进程的创建与销毁, 管理进程间的通信, 以及采取必要的措施避免死锁等内容。

网络管理系统: 无线网络设备工作原理, 内核掌控如何读取网络设备中的缓存数据。

## 2. 程序库

程序库是指可供使用的各种标准程序、子程序、文件以及它们的目录等信息的有序集合, Android 包含一些 C/C++ 库, Android 系统中不同的组件通过应用程序框架可以使用这些库, 以下是一些核心库:

**Surface Manager:** 管理显示子系统, 并且为多个应用程序提供 2D 和 3D 图层的无缝融合。

**Media Framework:** 基于 OpenCORE 的多媒体框架, 支持多种常用的音频、视频格式文件的回放和录制, 同时支持静态图像文件。

**SQLite:** 一个对于所有应用程序可用, 功能强劲的轻型关系型数据库引擎。

**OpenGL ES:** 3D 图形库, 用于 3D 图形渲染, 该库可以使用 3D 硬件加速。

**FreeType:** 位图 (Bitmap) 和矢量 (Vector) 字体显示。

**WebKit:** 支持 Android 浏览器和一个可嵌入的 Web 视图。

**SGL:** 2D 图形库, 用于 2D 图形渲染。

**LibC:** 一个从 BSD 继承的标准 C 系统函数库, 它是专门为基于嵌入式 Linux 设备定制的。

### 3. Android 运行库 (Android Runtime)

Android 运行库包括两部分：一是核心库，二是自身的虚拟机。

核心库提供 Java 编程语言核心库的大多数功能。Dalvik 虚拟机是 Google 专为 Android 开发的，比 SunJava 虚拟机的效率更高，功能也更为复杂，以更好的支撑 Android 平台，并拥有独立的版权。每一个 Android 应用程序都在自己的进程中运行，都拥有一个独立的 Dalvik 虚拟机实例，Dalvik 虚拟机执行 .dex 的可执行文件，该格式文件针对小内存的使用进行了优化，同时虚拟机是基于寄存器实现的，所有的类由 Java 编译器编译，然后通过 SDK 中的相应工具转化成 .dex 格式，最后由虚拟机执行。

### 4. 应用程序框架

应用程序框架是指定义了一个应用程序运行所必须的全部功能组件，开发者也可以访问核心应用程序所使用的 API 框架。该应用程序的架构设计简化了组件的重用；任何一个应用程序都可以发布它的功能块，并且任何其他的应用程序都可以使用其所发布的功能块（应该遵循框架的安全性限制）。同样，该应用程序的重用机制也使用户可以方便地替换程序组件。

### 5. 应用程序

Android 系统发布时，会同一系列核心应用程序和常用程序一起发布，如常用的手机功能程序，包括语音电话、通讯录、短信收发、照相、话机设置等；数据应用程序，包括邮件工具，日程表，浏览器，地图导航等，以及 Android Market 上的各种应用程序；所有的应用程序都是使用 Java 语言编写。

## 1.2 NDK 简介

NDK 即 Native Development Kit，NDK 是 Google 发布的一系列开发工具集合，这个工具可以帮助 Android 应用程序开发者编译来自 C/C++ 语言编写的源代码，然后嵌入到他们自己的应用包中，供应用程序调用。Android 的虚拟机在应用程序的源代码中通过 JNI 调用方法实现本地代码。在内核层面，这意味着你的应用程序源代码可以用关键字 `native` 方式声明一个或多个方法，用于指出这些方法是通过本地代码方式实现的；另外你必须提供一个本地共享库包含上述用关键字

native 声明的方法的具体实现代码。这个库是要被打包到应用程序的 apk 文件中，而且库的命名也要跟 Unix 的规则，即 lib<something>.so，将包含一个标准的 JNI 入口点。

NDK 集成了交叉编译器，并提供了相应的 MK 文件隔离 CPU、平台、ABI 等差异。MK 文件相当于配置文件，开发人员在 MK 文件中设置一些参数、欲编译的文件及编译特性的要求等，这样编译器就可以根据 MK 文件创建出动态库 so 文件。为了更好的与 C/C++ 开发相结合，也在不断完善开发中，目前 NDK 提供了一些稳定 API 头文件声明，但是在功能上还是相当有限，包括 C 标准库 (libc)、标准数学库 (libm)、压缩库 (libz)、Log 库 (liblog)。随着版本的不断升级，功能将越来越强大。

### 1.3 OpenCV 简介

OpenCV 是近年来推出的开源、免费的计算机视觉库，利用其所包含的函数可以很方便地实现数字图像和视频处理。同时利用面向对象的 VC++ 6.0 编程工具，用 C++ 语言进行程序编写，大大提高了计算机的运行速度。本文首先阐述了 OpenCV 的特点以及结构，然后以平滑处理、图像形态学为例介绍了 OpenCV 在数字图像处理中的典型应用。OpenCV 算法库为 VC++ 编程处理数字图像提供了很大的方便，其必将成为图像视频处理领域的强有力的工具。

利用 OpenCV 视觉库，科研开发人员只需添加自己的编写程序，直接调用 OpenCV 中的函数即可实现，这样不仅降低了开发程序的难度，而且缩短了相关程序的开发周期。