

第 1 章

Android 概述

在移动开发社区的帮助与支持下，Android 操作系统已经成为移动操作系统中的全球领先者。移动设备用户已经显示出对 Android 的喜爱。开发 Android 应用是将移动用户作为目标并想留住用户的商业公司的一个主要方向。手机制造商和移动运营商已经在 Android 上投入巨资，用于给用户创造一种独特的体验。企业家和初创企业正努力为其服务提供 Android 应用的用户体验，这是在其他移动平台或其他平台(如桌面)上所见不到的情景。此外，新设备不断涌现，设备的创造者为这些设备采用 Android 操作系统充分支持。

在移动开发社区，Android 已逐渐成为一个改变游戏规则的平台。Android 是一个创新和开放的平台，随着持续扩展到手机和平板电脑之外的新型设备，以及向其他领域的进一步渗透，Android 正在满足不断增长的市场需求。本章将介绍 Android 是什么，该平台如何融入已建立的移动市场，以及该平台的运作方式。

1.1 Android 开源项目(AOSP)

Android 开源项目(Android Open Source Project, AOSP)由 Google 主导，旨在使 Android 操作系统的源代码可供所有人阅读、审查并可根据自己的喜好进行修改。只要愿意，也可以贡献自定义的代码供其他人使用。AOSP 的主要目标是提供一套相容性指导方针，以便 OEM 和设备制造商将 Android 移植到定制设备以及构建符合 Android 开放配件标准的附件，使 OEM 厂商和制造商能够提供标准体验。

虽然任何人都可以自由地创建 Android 操作系统的源代码分支，但是保持 OS 体验的一致性对 Android 生态系统非常重要，因为对体验进行根本性的改变将在市场上引入分裂，以及与 Android 分发形成竞争。要了解更多关于 AOSP 和审查 OS 源代码的信息，请参阅 <https://source.android.com/index.html>。

1.2 开放手机联盟

Google 一直致力于宣传它的愿景、品牌，推广它的搜索和创收平台以及针对移动市场的开发工具套件。Google 公司的商业模式已经在互联网上取得了巨大成功。从技术角度看，移动市场并没有什么不同。

1.2.1 Google 进入移动市场

Google 最初进入移动市场遇到了所能想象的所有问题。互联网用户享受的自由与使用旧式手机的用户完全不同，因为那时的移动操作系统是封闭的生态系统——不像 Android 是开放源代码的——所以能为这些处于封闭状态的手机操作系统开发应用的开发人员仅限于少数。

互联网用户可以选择一系列不同的电脑品牌、操作系统、互联网服务提供商和网络浏览器。几乎所有 Google 服务都是免费的，是由广告驱动的。Google 创建了许多应用，并直接与这些封闭生态系统的移动操作系统上可用的应用进行竞争。这些应用的范围从简单的日历和计算器到 Google 导航地图，更不用说其他服务，如 Gmail 和 YouTube。

然而，这种做法并没有产生预期的效果，Google 决定采用不同的方式：改造整个移动应用开发的基础系统，希望可以为用户和开发人员提供一个更开放的环境——互联网模式。互联网模式允许用户在免费软件、共享软件和付费软件之间选择，这允许不同服务之间的自由市场竞争。

到今天，Google 对 Android 的巨大投入已经成为人们关注的焦点。Google 的搜索引擎算法已被修改，从而对不兼容移动系统的网站实施惩罚。移动搜索的流量已经超过台式机搜索的流量，并且这还会持续增长。Google 移动至上的理念绝对非常重要。

1.2.2 开放手机联盟介绍

凭借以用户为中心、民主的设计理念，Google 将现存的、壁垒森严的移动市场转变为手机用户可以在不同运营商之间轻松切换，可以无限制地运行应用和服务的市场。凭借庞大的资源，Google 已经采取广泛的方法，研究移动市场的整个基础架构——从 FCC 的无线频谱政策，乃至手机制造商的需求、应用开发人员的需求以及移动运营商的期望。

多年前，Google 加入了具有相同理念的移动社区，并提出如下问题：如何制造更好的手机？开放手机联盟(Open Handset Alliance, OHA)这个成立于 2007 年 11 月的组织回答了这个问题。开发手机联盟是由这个星球上许多规模最大、最成功的手机厂商组成的联盟。它的成员包括芯片厂商、手机制造商、软件开发商和服务提供商。它们很好地代表了整个移动供应链。

Andy Rubin 被称为 Android 平台之父。他的公司 Android.Inc 于 2005 年被 Google 收购。OHA 成员(包括 Google)开始开发一个基于 Android.Inc 技术的开发式标准平台，旨在缓解阻碍移动社区的上述问题。这就产生了 Android 项目。

Google 在 Android 项目中的参与是如此广泛，以至于谁是 Android 平台的主导(OHA 还是 Google)并不清晰。Google 提供了 Android 开源项目的早期代码，并提供了在线 Android 文档、工具、论坛和软件开发工具包(Software Development Kit, SDK)，供开发人员使用。最重要的 Android 新闻来自 Google。Google 还举办了多项会议(Google I/O、全球移动通信大会和 CTIA 无线通信展览会)。一系列的竞赛用于鼓励开发人员编写 Android 平台的杀手级应用，优胜者可获取数百万美元的奖励。Google 不仅是组织者，更是平台后面的驱动力。

1.2.3 加入开放手机联盟

AOSP 提供了 Android 操作系统的完整源代码，以及为满足设备兼容性需求的指南，但这不包括许多 Google 私有应用套件的源代码。加入开放手机联盟的好处包括授予 Google 移动服务(Google Mobile Services, GMS)许可的能力，这包括 Google 私有的应用，如 Google Play、YouTube、Google 地图、Gmail 和其他许多 Google 自有品牌的应用和服务。GMS 不包括在 AOSP 中，必须从 Google 直接授权。成为 OHA 的成员，还可将 GMS 捆绑到 Android 兼容的设备。

1.2.4 制造商：设计 Android 设备

开发手机联盟里有一半的成员是设备制造商，例如 Samsung、Motorola、Dell、Sony Ericsson、HTC 和 LG，以及半导体公司，例如 Intel、Texas Instruments、ARM、NVIDIA 和 Qualcomm。

第一部搭载 Android 的手机 T-Mobile G1 由手机制造商 HTC 开发，由移动运营商 T-Mobile 提供服务，发布于 2008 年 10 月。许多其他的 Android 手机则于 2009 年和 2010 年早期发布。Android 平台发展势头迅猛，到了 2010 年第 4 季度，Android 开始统治智能手机市场，逐步取代了其他竞争的手机平台，例如 RIM 的黑莓、苹果公司的 iOS 以及 Windows Mobile。

Google 通常在每年的 Google I/O 会议和重要会议上宣布 Android 平台的统计数据，例如财务收入。到 2015 年 5 月，Android 设备销售到的国家和地区已超过 130 个，在过去 12 个月里，Google Play 有超过十亿的活跃用户，500 亿个应用被下载和安装。制造商和运营商支持的优势显得卓有成效。

制造商不断创造新一代的 Android 设备——从手机和配备高清显示器的平板电脑，到提高移动体验或管理健康水平的手表，再到专用的电子书阅读器，到全功能的电视机、上网本、与汽车集成，以及你能想象到的几乎其他所有“智能”设备。

1.2.5 移动运营商：提供 Android 体验

设备开发出来后，必须交付给用户使用。包括北美、南美和中美洲，以及欧洲、亚洲、印度、澳大利亚、非洲和中东地区的移动运营商都加入了 OHA，从而确保了 Android 的全

球市场地位。拥有近 10 亿用户的电信巨头——中国移动也是联盟的创始成员之一。

大部分 Android 设备的成功往往基于以下事实：许多 Android 设备不需要和传统手机一样加上价签——不少手机由运营商提供免费激活；而竞争对手，如苹果公司的 iPhone 则受困于无法在低端市场提供有竞争力的产品。这是第一次，一个普通人可以负担得起全功能的智能手机。我们听说过很多人，从待业人员到杂货店的店员，说到他们的生活在收到第一部 Android 手机后变得更好了，而这种现象只会日益提升 Android 的霸主地位。

制造厂商为 Android 的增长做出了巨大贡献。2015 年 7 月，据 IDC 公司统计 (<http://www.idc.com/getdoc.jsp?containerId=prUS25804315>)，2015 年第二季度，三星全球出货量 7320 万部智能手机，这些设备的大部分最有可能搭载的是 Android 系统。

Google 还创建了自己的 Android 品牌，称为 Nexus。现在的 Nexus 产品线有 6 款设备，分别是 Nexus 4、5、6、7、9 和 10，每款设备分别由手机制造商合作伙伴 LG(4、5)、摩托罗拉(6)、华硕(7)、HTC(9)和三星(10)制造。Nexus 设备提供了完整的、真正的 Google 所希望的 Android 体验。许多开发人员使用这些设备来创建和测试他们的应用，因为它们是世界上唯一能及时更新 Android 操作系统的设备。如果希望自己的应用能工作在最新的 Android 操作系统版本上，应该考虑购买其中一款或多款设备。

1.2.6 应用驱动设备的销售：开发 Android 应用

用户购买了 Android 后，他们需要杀手级应用，不是吗？

最初，Google 主导开发 Android 应用，其中有很多应用(例如电子邮件客户端和网页浏览器)是这个平台的核心功能。他们还开发了首个成功的第三方 Android 应用分发平台：Android 市场，也就是现在的 Google Play 商店。Google Play 商店仍然是用户下载应用的主要方式，但不再是唯一的 Android 应用分发平台。

截至 2015 年 5 月，在过去 12 个月内，从 Google Play 商店下载、安装的应用超过 500 亿。这些只考虑了应用在该市场分发，没有考虑其他应用单独出售或在其他市场分发的情况。这个数字也没有考虑到在 Android 平台上运行的 Web 应用。这些为 Android 用户提供了更多选择，也为 Android 开发人员提供了更多机会。

Google Play 商店一直在努力增加展示和销售游戏应用，并提供了 Play Game Services SDK。该 SDK 允许开发人员在游戏中增加实时社交功能，以及应用编程接口(Application Programming Interface, API)来实现排行榜和成就榜单，从而吸引新用户并鼓励老用户。Google 还在进行一项旨在帮助推动内容销售的工作。用户总是寻找新的音乐、电影、电视节目、书籍和杂志等。Google Play 专注于这些内容来满足用户对这方面服务的需求。

1.2.7 利用所有 Android 设备的优势

Android 的开放平台已经得到大量移动开发社区的支持——远远超过了 OHA 的成员。

随着 Android 设备和应用变得越来越容易获得，许多其他的移动运营商和设备制造商转而销售 Android 设备给他们的客户，特别是相对于其他专有平台的成本方面来考虑。

Android 平台的开放标准能为运营商减少许可和专利费用，所以我们看到了更多开放设备的迁移。市场已经完全敞开，新用户能够首次就考虑智能手机，而 Android 很好地填补了这一需求。

1.2.8 Android：我们现在取得的进展

Android 在各个方面(设备、开发人员和用户)持续增长。最近，焦点主要集中在以下这几个方面：

- **有竞争力的硬件和软件功能升级：**Android SDK 开发人员专注于提供竞争对手没有的功能 API，从而保持 Android 在市场上的领先地位。例如，最新发布的 Android SDK 版本显著改善了通知功能，在需要时为你提供重要信息。
- **扩展智能手机和平板电脑：**智能手表的用户与 Android 用户都呈上升趋势。目前市场上有许多新的 Android 穿戴设备，它们有不同的尺寸和外形。一些硬件制造商甚至将 Android 应用到游戏机、电视机和汽车仪表盘，以及许多其他需要操作系统的设备。Google 甚至已经宣布 Project Brillo，专为物联网(Internet of Things, IoT)设计的一个 Android 版本，以及一个用于连接这些设备的 IoT 协议 Weave。
- **提升面向用户功能：**Android 的开发团队将重点从功能的实现转到了提供面向用户的可用性升级和“多彩性”。投入巨资来创造更流畅、更快速、反应更灵敏的用户界面，并更新他们的设计文档，使其成为一流的教程供开发人员学习实践。遵循这些原则可以帮助所有的应用增加可用性。

注意



有些人可能会对移动市场中围绕着 Android 几乎所有成员的法律纠纷感到困惑。虽然大部分并没有直接影响开发人员，但其中一些(特别是涉及应用内购买的)则有影响。这种事情在任何主流的平台都会发生。这里并不能提供任何法律意见。我们能给出的建议是保持对法律纠纷的关注，希望一切都好，不只是在 Android 平台，还在其他受影响的平台。

1.3 Android 平台的独特性

Android 平台本身被誉为“第一个完整、开放和免费的移动平台”。

- **完整：**在开发 Android 平台的时候，设计者进行了全面的考虑。他们从一个安全的操作系统开始，在上面建立一个健壮的软件框架，从而允许在上面开发丰富的应用。
- **开放：**Android 平台通过开源许可协议来提供。开发人员开发应用时可以获得前所未有的访问设备功能的权限。

- **免费：**Android 应用可以免费开发。在该平台上开发不需要许可费用。没有加入开发成员的费用，没有测试费用，不需要签名或认证费用。Android 程序可以通过多种方法来分发和商业化。分发自己的应用是免费的，也有免费发布应用以供下载的应用商店。但是在 Google Play 商店上架则需要注册和支付一笔一次性的\$25 费用(免费意味着开发过程可能是有成本的，但这些都 Android 平台上不是强制要求的。这并不包括设计、开发、测试、市场和维护费用。如果提供了所有这些，可能不需要再付费，除了一项费用——\$25 的开发人员注册费，这项收费用于鼓励开发人员为 Google Play 创建高质量的应用)。

1.3.1 Android 的由来

Android 的吉祥物是一个绿色的小机器人，如图 1.1 所示，这个小机器人经常用来表示 Android 相关的内容。



图 1.1 Android 官方吉祥物

自从 Android 1.0 SDK 发布以来，Android 平台持续以快速的步伐前进。相当一段时间，每隔几个月时间就有一个新的 Android SDK 发布出来！在高科技行业，每个 Android SDK 版本都有一个独特的项目名称。在 Android 世界里，每一代 SDK 是按字母顺序命名的甜点。

1.3.2 自由和开放源代码

Android 是一个开源平台，不论是开发人员还是设备制造商都不需要为该平台开发支付专利费或许可费用。

Android 底层的操作系统基于 GNU 通用公共许可第二版(GPLv2)著作权许可，它要求任何第三方的修改必须继续保持开源许可协议的条款；而 Android 框架则基于 Apache 软件许可证(ASL/Apache2)发布，它允许发布开源或闭源的版本。平台的开发人员(尤其是设备制造商)可以选择增强 Android 功能而不需要将他们的改动提供给开源社区。相反，平台开

发人员可以从特定设备的改进工作中获利,并在他们想要的许可协议下重新发布工作成果。

Android 应用开发人员可以在他们喜爱的许可协议下发布他们的应用,也可以编写一个开源的自由软件或传统意义上的收费应用,或是介于两者之间的软件。

1.3.3 熟悉且廉价的开发工具

不像某些专有平台,需要开发人员缴纳注册费用、审批费用和购买昂贵的编译器,开发 Android 程序没有前期成本。

1. 免费提供的软件开发工具包(SDK)

Android SDK 和工具都可以免费得到。开发人员同意 Android SDK 许可协议后,就可以在 Android 网站下载 Android SDK。

2. 熟悉的编程语言,熟悉的开发环境

开发人员现在可以使用官方的集成开发环境(IDE)用于 Android 应用开发。Android Studio 集成了 Android SDK 工具、最新的 Android Platform 以及最新的带 Google API 的 Android 模拟器系统镜像。Android Studio 基于免费的由 JetBrains s.r.o 公司开发的 IntelliJ IDEA Community Edition。

在 Android Studio 成为 Android 开发的官方 IDE 之前,许多开发人员选择流行而且免费的 Eclipse IDE 来设计和开发 Android 应用。Eclipse 是最流行的 Android 集成开发环境之一。被称为 Android 开发人员工具(Android Developer Tools, ADT)的 Android Eclipse 的插件则可以辅助 Android 开发。

还可以选择从命令行以独立应用的方式使用 Android SDK 工具,而不将其整合到特定的 IDE 中,也可以运行命令行构建脚本。

Android Studio 是 Android 应用开发推荐的 IDE。Android 应用可以在以下操作系统中开发:

- Windows 2003、Vista、7 和 8 (32 位或 64 位)
- Mac OS X 10.8.5 至 10.9 的所有版本
- Linux GNOME 或 KDE 桌面(在 Ubuntu Linux 14.04 64 位上测试过)

1.3.4 合理的开发学习曲线

Android 应用使用著名的编程语言 Java 编写。Android 应用框架包含了传统的编程结构,如线程和进程,以及专门设计的数据结构来封装移动应用常用的对象。开发人员可以依靠熟悉的类库,例如 java.net 和 java.text。专业库的支持(例如图形和数据库的管理)则基于良好定义的开放标准,如 OpenGL 嵌入式系统(OpenGL ES)和 SQLite。

1.3.5 功能强大的应用开发支持

过去,设备制造商往往和信赖的第三方软件开发商(OEM、ODM)建立特殊关系。软件

开发商的精英们为之编写原生应用，如消息管理和 Web 浏览器，作为设备的核心功能集。为了设计这些应用，开发商需要给予开发人员得到内部软件框架和固件资料的权限。

而在 Android 平台上，原生应用和第三方应用之间并没有区别，从而可以保持开发人员之间的良性竞争。所有 Android 应用使用同一套 API，Android 应用可以访问底层硬件，允许开发人员编写更强大的应用。应用可以完全被扩展或替代。

1.3.6 丰富和安全的应用集成

Android 平台一个最引人注目和创新的功能是设计良好的应用集成。如果开发人员愿意，Android 可以允许开发人员编写一个应用，无缝地集成核心功能，如 Web 浏览器、联系人管理和短消息等。应用也可以作为内容提供程序并以安全的方式分享彼此的数据。

1.3.7 没有昂贵的开发费用

不像 iOS 等平台，Android 应用不需要昂贵和耗时的测试认证程序。创建 Android 应用，你所需要的仅是一台电脑、一个 Android 设备、一个好的想法和对 Java 的理解。

如果你想在 Google Play 商店发布应用，则需要支付一次性的低成本(\$25)的开发人员费用，但是你也可以选择一个不需要支付开发人员费用的应用商店来发布你的应用，或者你也可以自己为应用提供下载。

1.3.8 应用的“自由市场”

Android 开发人员可以自由选择任何一种他们想要的收入模式。他们可以开发免费软件、共享软件、试用软件或带广告的应用和收费应用。Android 的目的是从根本上颠覆移动应用的开发规则。在 Android 移动平台之前，开发人员面临着许多功能方面的限制，如：

- 软件市场对同一类特定类型应用数量的限制。
- 软件市场对价格、收费模式和专利费用的限制。
- 运营商不愿意为少数人提供应用。

在 Android 平台上，开发人员可以编写和成功发布他们想要的任何类型的应用。开发人员可以为少数人提供定制的应用，而不是基于移动运营商的要求只提供多数人的收费版本。垂直市场的应用可以部署到特定目标人群。

因为开发人员拥有多种应用分发机制的选择，他们可以选择一种方式而不需要强迫遵守别人的规则。Android 开发人员可以通过多种方式发布他们的应用：

- Google 开发的 Google Play 商店(原来的 Android 市场)——一个通用的收入共享的 Android 应用商店。Google Play 商店现在拥有一个 Web 商店用于在线浏览和购买应用。Google Play 同时也销售电影、音乐和书籍。因此，选择它，你的应用将出现在一个极好的商店里售卖。
- Amazon 在 2011 年上线了 Amazon AppStore，它包含了一系列令人兴奋的 Android 应用，并使用自己的收费和收入共享系统。

- 还有许多其他的第三方应用商店可供选择。有些比较小众，有些支持不同的移动平台。
- 开发人员还可以提供自己的支付/收费机制，例如在网站或企业内部分发。

移动运营商和手机开发商现在仍然可以免费地开发自己的应用商店并执行自定的规则，但这不再是开发人员分发他们应用的唯一方式。在这些平台分发你的应用之前，请一定仔细阅读应用商店的协议。

1.3.9 一个不断发展的平台

早期的 Android 开发人员必须面对新平台的典型困难：频繁修改的 SDK，缺乏良好的文档，市场的不确定性；移动运营商和设备制造商对 Android 的升级支持即使有，也很慢。这意味着 Android 开发人员常需面对不同的 SDK 版本以满足所有用户。幸运的是，不断发展的 Android 开发工具使其变得简单，现在 Android 已经是一个完善的平台，其中许多问题已经得到解决。Android 论坛社区十分活跃和友善，并非常倡导互相帮助解决困难。

Android SDK 每一次的版本更新都提供了一些平台的实质改善。在最新的版本中，Android 平台增加了很多人需要的“艳丽”用户界面，表现在视觉和性能上的改善。流行的设备，例如智能手表或互联网电视现在完全支持该平台，此外还支持新的类别，如智能汽车。

虽然大部分的升级和改善是受欢迎和必要的，但是新的 SDK 版本常会导致 Android 开发人员社区的混乱。一些已经发布的应用都需要重新测试和重新提交到 Google Play 商店来满足新的 SDK 的需求。这带来了 Android 设备的固件升级，使得一些旧的应用过时，有时甚至无法使用。

虽然这些成长中的阵痛可以预见，而且大部分开发人员已经容忍了这些，但记住，和 iOS 平台相比，Android 在移动市场是一个后来者。苹果的 APP Store 拥有许多应用，但用户希望他们的 Android 设备上也有相同的应用，开发商很少只为一个平台开发部署，他们必须能支持所有流行的平台。

1.4 Android 平台

Android 是一个操作系统，也是开发应用的软件平台。一些日常任务的核心组件，例如网页浏览和电子邮件应用都包含在 Android 设备里。

作为 OHA 的愿景——强大开源的移动开发环境，Android 是一个领先的移动开发平台。该平台旨在鼓励自由开放的市场，一个用户所希望的和开发人员渴望去开发的市场。到目前为止，该平台没有辜负这一期望。

1.4.1 Android 的底层架构

与其前辈相比，Android 平台被设计成具有更高容错能力的平台。设备运行在 Linux 操

作系统上, Android 应用在安全的方式下执行。每个 Android 应用运行在自己的应用沙箱中(见图 1.2)。Android 应用都是托管代码,因此,它们不太可能导致系统崩溃,进一步导致系统损坏(无法使用)的可能性更小。

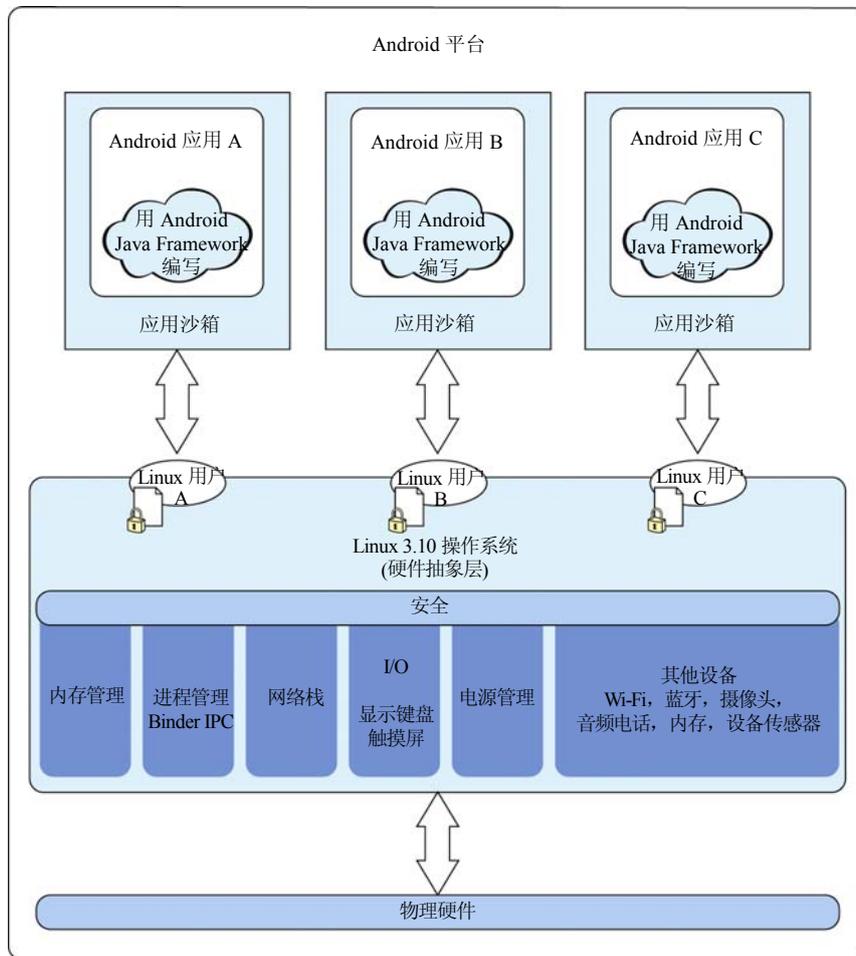


图 1.2 Android 平台架构图

1. Linux 操作系统

Linux 内核负责处理核心系统服务,并作为硬件抽象层(Hardware Abstraction Layer, HAL)介于物理硬件和 Android 软件栈之间。

内核处理的一些核心功能包括:

- 增强的应用权限和安全性。
- 低级别的内存管理。
- 进程管理和多线程。
- 网络栈。

- 显示、键盘输入、摄像头、Wi-Fi 无线、闪存、声音、Binder 进程间通信(IPC)和电源管理驱动力的访问。

2. Android 应用运行时环境

每个 Android 应用运行在单独的进程中，在它自己的应用沙箱中。Android 运行时(Android Runtime, ART)是 Dalvik 的运行时继任者。ART 引入的一个主要改进功能是预编译(Ahead-Of-Time Compilation, AOT), 而不是 Dalvik 的及时编译(Just-In-Time Compilation, JIT)。有了 ART, 应用在安装过程中编译。编译后的可执行文件存储在设备上, 而不必在启动应用之前编译可执行文件。而另一方面, Dalvik 在启动应用之前将编译应用文件。ART 在 Android 5.0 中正式推出, 并带来显著的性能增强, 这是 Dalvik 以前所不具备的。

1.4.2 安全和权限

Android 平台的完整性通过一系列安全措施来维护。这些措施确保用户数据安全, 设备不会遭受恶意软件和误操作的影响。

1. 应用作为操作系统的用户

当一个应用被安装后, 操作系统创建了一个和该应用相关联的新的用户配置文件。每个应用作为不同的用户来运行, 在文件系统中拥有私有的文件, 有独立的用户 ID, 独立的安全操作环境。

应用在操作系统中使用自己的用户 ID, 在自己的应用安全沙箱中运行自己的进程。

2. 安全增强型 Linux 内核模块

Android 4.3 推出了安全增强型 Linux(Security-Enhanced Linux, SELinux)内核模块的修改版本。此版本为 Android 操作系统提供了增强的安全性, 并进一步将应用限制在自己的沙盒, 同时也在所有进程上实施强制访问控制(MAC)。

3. 明确定义的应用权限

Android 应用需要注册所需要的特定权限来访问系统上的共享资源。有些权限允许应用使用设备的功能来拨打电话、访问网络、控制摄像头和其他硬件传感器。应用也需要权限来获取包含私人信息的共享数据, 例如用户偏好、用户位置和联系人信息。

应用也可以声明其他应用来使用它们自己的权限。一个应用可以声明任意数量的不同权限类型, 例如只读或读写权限, 从而更好地施加控制。

Android 6.0(API 等级 23)推出了精简的许可程序。不再要求用户在安装应用时授予应用需要的所有权限, 允许开发人员可在运行期间当应用实际上需要访问特定权限时请求权限。常规保护级别的权限在安装时进行授权, 而非常规的其他所有级别的权限必须在运行期间请求授权。

4. 应用签名

所有 Android 应用包都使用证书来签名，所以用户知道该应用是认证过的。证书的私钥由开发人员保存。这有助于建立开发人员和用户的信任关系。它也能使开发人员控制哪些应用能提供系统上其他应用的访问权限。没有哪家证书颁发机构是必需的，自签名也是可以接受的。

5. 多用户和限制配置文件

Android 4.2(API 级别 17)带来了可共享设备，例如平板电脑的多用户账户支持。随着 Android 4.3(API 级别 18)版本的发布，主设备用户现在可以创建限制配置文件，用于限制用户访问特定应用的权限。开发人员也可以利用他们应用中的限制配置文件的功能，从而使主设备用户拥有进一步限制特定设备用户访问特定应用内容的能力。

6. Google Play 开发人员注册

为在广受欢迎的 Google Play 商店发布应用，开发人员必须创建一个开发人员账户。Google Play 商店管理严格，并且不允许有恶意软件。

1.4.3 探索 Android 应用

Android SDK 提供了大量最新的、健壮的 API。Android 设备的核心服务公开给所有的应用来访问。只要授予了相应的权限，Android 应用可以相互共享数据，并能安全地访问系统上的共享资源。

1. Android 编程语言选择

Android 应用是用 Java 语言编写的。到目前为止，Java 语言是开发人员访问完整 Android SDK 的唯一选择。

提示



有一些猜测：其他的编程语言(例如 C++)可能会在 Android 未来版本中加入。如果你的应用必须依赖其他的编程语言(例如，C/C++)的本地代码，你可能需要考虑使用 Android Native Development Kit(NDK)。

也可以开发一个运行在 Android 设备上的移动 Web 应用。这些应用可以通过 Android 浏览器访问，也可以通过嵌入本地 Android 应用(仍然是用 Java 编写的)的 WebView 控件访问。本书专注于 Java 应用的开发。可以在 Android 开发人员网站找到更多关于开发 Web 应用的内容：<http://d.android.com/guide/webapps/index.html>。

想要部署到 Android 平台的 Flash 应用？请检查 Android 平台的 Adobe 的 AIR 支持情

况。用户从 Google Play 商店安装了 Adobe 的 AIR 应用之后，就可以用来加载兼容应用了。欲了解更多信息，请访问 Adobe 网站 http://adobe.com/devnet/air/air_for_android.html。

开发人员甚至可以选择使用某些脚本语言开发应用。目前有一个开源项目，它可以使用脚本语言，例如将 Python 等作为构建 Android 应用的选择，但是这个项目已经很长一段时间没有更新了。欲了解更多信息，请参阅 Android 脚本项目：<https://github.com/damonkohler/sl4a>。与 Web 应用、Adobe AIR 应用类似，开发 SL4A 应用不在本书的介绍范围之内。

2. 自带应用和第三方应用之间无差异

不像其他的应用开发平台，Android 平台上的自带应用和第三方应用之间没有区别。只要授予相应的权限，所有应用都能以相同的方式访问核心库以及底层接口。

Android 设备出厂的时候自带了一系列原生应用，例如 Web 浏览器和联系人管理器。第三方应用可以整合这些核心应用，并扩展它们以提供更丰富的用户体验，或者使用替代应用完全替代。这意味着：任何应用都是使用与第三方开发人员也能使用的完全相同的 API 构建的，从而营造了公平，或者尽可能接近公平的竞争环境。

值得注意的是，在较早时期 Google 公司自己在某些情况下使用了未文档化的 API。因为 Android 是开放的，没有私有的 API。Google 从来没有禁止访问这些未文档化的 API，但是警告了开发人员，使用这些私有 API 可能导致在未来的 SDK 版本中不兼容。参考博客 <http://android-developers.blogspot.com//2011/10/ics-and-non-public-apis.html>，有一些曾经未文档化的 API 成为公开 API 的示例。

3. 常用的包

在 Android 平台，移动开发人员不需要重新发明车轮。相反，开发人员可以使用 Android 的 Java 包内的类库来完成常见任务，包括图形、数据库访问、网络访问、安全通信和实用工具。Android 包提供了以下支持：

- 各种用户界面控件(按钮、下拉列表、文本输入)。
- 各种用户界面布局(表格、标签页、列表)。
- 整合功能(通知、窗口小部件)。
- 安全的网络和 Web 浏览功能(SSL、WebKit)。
- XML 支持(DOM、SAX、XML Pull 解析器)。
- 结构化存储和关系数据库(应用程序首选项、SQLite)。
- 强大的 2D 和 3D 图形库(包括 SGL、OpenGL ES 和 RenderScript)。
- 播放和录制单机或网络流的多媒体框架(MediaPlayer、JetPlayer、SoundPool 和 AudioManager)。
- 对许多音频和视频格式的广泛支持(MPEG4、H.264、MP3、AAC、AMR、JPG 和 PNG)。

- 可以访问可选的硬件，如基于位置的服务(Location-Based Services, LBS)、USB、无线网络、蓝牙、近场通信以及硬件传感器。

4. Android 应用框架

Android 应用框架提供了实现一般应用所需的一切东西。Android 应用的生命周期主要包含以下主要组件：

- Activity(活动)是应用执行的功能。
- Fragment(片段)是可重用和模块化的子 Activity。
- Loader(加载器)用于将数据异步加载到 Fragment 或 Activity 中。
- 视图的 Group(布局)用于定义应用的布局。
- Intent(意图)通知系统有关应用的计划。
- Service(服务)允许后台处理而不需要用户交互。
- Notification(通知)在一些有趣的事情发生时提醒用户。
- Content provider(内容提供程序)促进不同应用之间的数据传递。

5. Android 平台服务

Android 应用使用一系列管理器与操作系统以及底层硬件交互。每个管理器负责保持一些系统服务的状态。例如：

- LocationManager 用于和设备上的基于位置的服务进行交互。
- ViewManager 和 WindowManager 负责显示界面以及与设备相关的用户界面的基本组件。
- AccessibilityManager 负责辅助事件，为有物理损伤的用户提供支持。
- ClipboardManager 提供了访问设备全局剪切板的功能，可以剪切和复制内容。
- DownloadManager 作为系统服务，负责 HTTP 的后台下载。
- FragmentManager 管理 Activity 的 Fragment。
- AudioManager 提供对音频和振铃控制的访问。

6. Google 服务

Google 提供了 API 用于整合许多不同的 Google 服务。这些服务被添加之前，开发人员需要等待移动运营商和设备制造商更新 Android 设备，才可以使用许多功能，例如地图和基于位置的服务。现在，开发人员可通过在应用的项目中加入所需的 SDK 来整合这些最新最好的服务更新。Google 服务包括：

- 地图(Maps)
- 基于位置的服务(Places)
- 游戏服务(Play Game service)
- Google 账户登录(Google Sign-In)
- 应用内收费和订阅(In-app Billing and Subscription)

- Google 云消息(Google Cloud Messaging)
- 移动应用分析 SDK(Mobile App Analytics SDK)
- AdMob 广告服务(AdMob Ads)

1.5 OHA 和 GMS 之外的 Android 版本

设备制造商加入 OHA 的一个主要好处是可以授权使用 Google 品牌应用的 GMS 套件, 如 Google Play。GMS 提供了很多特性和功能。也就是说, 有一些与 OHA 没有关联的流行 Android 版本, 所以如果设备售出后用户不自己安装 GMS 的话, 这些版本的 Android 不能访问 GMS。基于非 OHA 的 Android 版本的设备, 不包含 GMS 或 Google Play, 这并不意味着你应该忽视为这些设备提供应用支持。下面介绍一些有趣的自定义 Android 版本。

1.5.1 Amazon Fire OS

亚马逊(Amazon)创建了自己的 Android 版本, 名为 Fire OS。Fire OS 是 AOSP 的一个分支, Fire OS 安装在所有的亚马逊 Fire 品牌设备上, 如 Fire Phone、Fire Tablet 和 Fire TV。最近, 亚马逊发布了基于 Android Lollipop 的 Fire OS 5 开发人员预览版。

根据 Strategy Analytics 公司的一份报告, 亚马逊 Fire TV 自开卖以来, 出货量接近 450 万台 (<http://www.prnewswire.com/news-releases/amazonfires-to-the-top-of-the-us-digital-media-streamer-market-says-strategy-analytics-300094475.html>)。有几百万台的设备, 所以在亚马逊 Fire OS 上支持你的 Android 应用是绝对值得考虑的。

要进一步了解亚马逊的 Fire OS 版本的 Android, 请参考 <https://developer.amazon.com/public/solutions/platforms/android-fireos>。

1.5.2 Cyanogen OS 和 CyanogenMod

另一个值得关注的 Android 版本是 Cyanogen OS。Cyanogen OS 基于 CyanogenMode 项目, 该项目是 Android OS 的一个社区驱动的分支版本, 没有 GMS, 不过用户社区提供了指南和工具用于安装 Google Play 等应用。Cyanogen 公司的博客(<https://cyngn.com/blog/an-open-future>)号称分布在全球 190 个国家的超过 5000 万用户在运行不同版本的 CyanogenMod。CyanogenMod 定位为一个可替代的固件, 为了替换购买时设备内置的系统, 需要用户手动安装。另一方面, Cyanogen OS 是一个可预先安装在 Android 设备上的固件。

Cyanogen 是 Cyanogen OS 背后的公司, 正在致力于创建一个像 Google 一样有竞争力的 Android 生态系统。Cyanogen 公司最近融到一笔 8000 万美元的风险投资, 来自 Qualcomm Incorporated、Twitter Ventures、Rupert Murdoch、Andreesen Horowitz 以及腾讯等投资人。

欲了解更多有关 CyanogenMod 的信息, 请参考 <http://www.cyanogenmod.org>。欲了解更多有关 Cyanogen OS 的信息, 请参考 <https://cyngn.com>。

1.5.3 Maker Movement 和开源硬件

另一个值得关注的领域是“Maker Movement”，这是一个 DIY 技术爱好者社区，通常也称为“Makers”（创客）。这一社区的亚文化潜心于基于开源硬件的项目。类似于开源软件运动的早期，硬件工业正经历类似开源的趋势——主要在电子和印刷电路板(PCB)设计领域。进入设计复杂的电子设备(例如电脑、笔记本电脑、平板电脑或 IoT 设备)的壁垒似乎只局限于人们的想象力和创新的欲望。

主要的硬件组件公司历来谨慎保护电子和 PCB 设计，现在逐渐意识到某些开源设计的创新潜力。处理器制造商，例如 Intel，以及授权和制造基于 ARM 处理组件的其他公司，已发布开源 PCB 设计并提供完整的 PCB 电路图，列出了完成电路设计所需的组件。这对组件制造商提供可工作的 PCB 设计以帮助推动这些组件的销售是一种激励。

相当多生产 ARM 处理器的公司已经为平板电脑设备开发了开源 PCB，使用 Android 作为操作系统。这使得设计复杂的设备(运行 Android 的平板电脑)对有 PCB 设计能力的人来说能轻松完成。PCB 软件设计工具(例如，Altium Designer)被用于 PCB 设计。

强大的工具与开源 PCB 设计以及 AOSP 相结合，可能带来我们现在还无法想象的新一代设备。Android 应用开发的未来一定是光明的，为 Android 开发创新应用的可能性几乎是无限的。

1.5.4 保持警觉

虽然本书是关于开发 Android 应用的，但我们希望提供整个 Android 生态系统的背景知识。我们认为随着不断扩张的 Android 生态系统，对这些事情产生的影响保持警觉总是一个好主意，因为这些事情将影响每个参与者。今天，Android 有许多令人兴奋的事件值得关注，也希望未来能带给我们更多惊喜。

1.6 本章小结

Android 软件开发在过去几年发展迅猛。Android 已经成为移动开发平台的领头羊，它借鉴了其他平台过去的成功经验，吸取了过去其他平台的失败教训。Android 设计为鼓励开发人员编写创新型应用。该平台是开源的，没有前期费用，相对于其他的竞争平台，开发人员可以享受很多好处。Android 生态系统在不断地努力进军一些有前途的新领域。现在是深入研究 Android 平台，以便可以评估 Android 平台能给你带来什么的时候了。

1.7 小测验

1. 首字母缩写 AOSP 指的是什么？

2. 判断题：加入开放手机联盟后，就允许设备制造商绑定 Google Mobile Service。
3. Google 购买了哪家公司，并在 Android 操作系统中使用和发展了它的技术？
4. 第一款 Android 设备是什么？哪家设备制造商开发了它？在哪家移动运营商销售？
5. 基于 Android 的 Amazon OS 的名称是什么？

1.8 练习题

1. 描述 Android 作为开源系统的好处。
2. 用你自己的语言，阐述 Android 的底层架构。
3. 要熟悉 Android 的文档，可以通过下面的网址找到：<http://d.android.com/index.html>。

1.9 参考资料和更多信息

Android 开发人员：

<http://d.android.com/index.html>

Android 开源项目：

<https://source.android.com/index.html>

开放手机联盟：

<http://openhandsalliance.com>

官方的 Android 开发人员博客：

<http://android-developers.blogspot.com>

本书的博客：

<http://introductiontoandroid.blogspot.com>

Intel 开放源码：Android 在 Intel 平台上：

<https://01.org/android-IA>

ARM Connected Community: Android Community:

<http://community.arm.com/groups/android-community>

Altium Designer:

<http://www.altium.com/altium-designer/overview>

Wikipedia: Maker Culture:

https://en.wikipedia.org/wiki/Maker_culture