

第1章

Android 系统概述

本章概述 Android 系统方面的内容，包括智能手机平台的发展历程、Android 系统的特点、Android 的三种开发工作，以及 Android 的两种不同的开发方式。

在学习本章的过程中，读者应重点关注以下内容：

- 作为智能手机平台，Android 系统的特点
- 移植开发、应用开发、系统开发三种工作的适应场合
- Android 的 SDK 开发和源代码开发各自需要的知识
- Android 的相关资源（市场、开发者、源代码）

1.1 基础知识

1.1.1 Android 开发系统的由来

OHA（Open Handset Alliance，开放手机联盟）是 Google 与 33 家公司联手为 Android 移动平台系统的发展而组建的一个组织。

Android 是目前最流行的手机开发平台，依靠 Google 的强大开发和媒体资源，Android 成为众多手机厂商竞相追逐的对象。

Android 是一个包括操作系统、中间件和关键应用的移动设备软件堆。Android 系统的 Logo 是一个机器人，如图 1-1 所示。

Android 系统的网站为：<http://www.android.com/>。基于 Android 系统的第一款移动电话 G1 如图 1-2 所示。



图 1-2 Android 系统移动电话

1.1.2 移动电话系统开发模式

移动电话（手机）作为一种消费类电子产品，走进普通民众的生活，已经有 20 年左右的历史了。移动电话的开发经历了以下几个模式：

- 传统移动电话的开发；
- 半开放式移动电话的开发；
- 全开放式移动电话的开发。

从移动电话开发的历史来看，传统的移动电话开发模式一直是一种主要的形式，其开发模式乃至商业模式一般只有一种，这就是：厂商制作移动电话出售。在这种开发模式中，移动电话的制造厂商制作移动电话，然后以移动电话为产品出售给消费者，这是一种比较简单、常见的商品制造和销售模式。从开发过程来看，移动电话厂商可以使用自己的研发机构，也可以依靠其他公司提供的解决方案（例如 MTK 的解决方案）来完成移动电话的开发工作。

这种传统的移动电话开发模式目前依然是移动电话开发的主流，其特点是：这样开发出来的移动电话是一个封闭的系统，和普通的产品没有太大的差别。用户只能使用移动电话厂商提供的固定功能，而不能根据需求开发和使用个性化的功能。

随着用户对移动电话系统的自定义需求的增加，移动电话系统的开发走向了开放模式。这种模式事实上是一种半开放式的移动电话开发，它把移动电话的开发过程分成了两个阶段：

- 厂商制作移动电话出售；
- 第三方开发移动电话应用。

在这种半开放式的移动电话开发中，厂商依然依靠制造移动电话出售作为主要的盈利模式，但是增加了第三方开发移动电话程序的机会。例如，很多现有的移动电话系统开始支持让用户自己下载和安装程序。例如，J2ME 技术以及苹果公司的 iPhone 可以开发第三方应用，并可以在 AppStore（iPhone/iPodtouch 的网上软件商店）进行交易。

在这种模式中，需要移动电话厂商支持开放的接口，为第三方应用程序提供在它们的移动电话上运行的可能。这样，移动电话厂商和第三方应用的制作者就可以形成一个互相促进的关系：一方面，第三方应用的制作者可以像开发通常软件一样，开发移动电话应用，获得一定的利润；另一方面，由于具有了众多第三方应用，移动电话也具有了更强大的竞争力，这也促使移动电话的制作厂商倾向于支持第三方应用的移动电话。

上面所说的移动电话的开放模式，还只能说是一种半开放的开发模式，因为在这种开放模式中，第三方应用的接口是开放的，但是移动电话系统本身的开发并不是开放的。

Android 的出现，构建了一种新型的移动电话开发模式，这就是一种全开放的移动电话开发模式。Android 移动电话不但可以使用第三方应用，而且移动电话系统本身也是开放的，在这种开发模式下：

- 各厂商在统一开放的平台上开发移动电话；
- 第三方开发移动电话应用。

Google 的 Android 系统是一个完全开放的系统，也是一个完整的生态系统，它分成了三个有机的组成部分：

- Android 源代码工程（Android Open Source Project）；
- Android 开发者（Android Developer）；
- Android 市场（Android Market）。

Google 的 Android 网站也是由以上三个部分组成的，如图 1-3 所示。

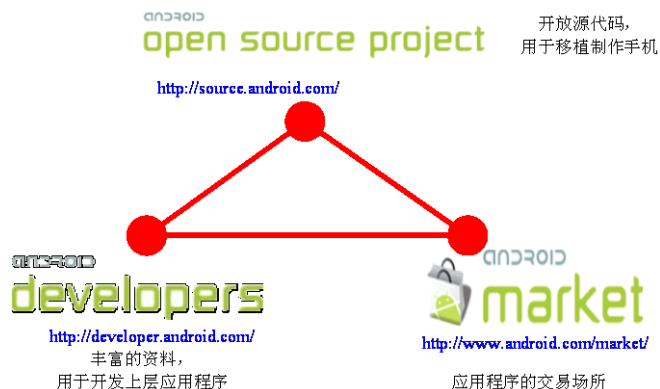


图 1-3 Android 源代码、开发者和市场三者的关系

Android 源代码工程包含了 Android 系统的全部源代码, 用于开发移动电话系统。Android 开发者包含了 Android 应用层的接口和开发文档, 提供给 Android 应用程序的开发者。Android 市场 (Market) 与苹果公司的 AppStore 类似, 是 Google 为 Android 的程序交易提供的场所, 其网址为: <http://www.android.com/market/>。

Android 的 Market 标志如图 1-4 所示。

在 Android 的 Market 中, 开发者上传自己开发的 Android 应用程序, 而使用者在市场上下载这些程序使用, 丰富自己的移动电话功能。

以上三个部分是相互依存的, 从移动电话的系统软件开发, 到移动应用程序的开发, 再到应用程序的交易, Android 系统是完全开放的。

1.1.3 未来移动电话的功能及 Android 的优势

移动电话 (Mobile Phone) 的原本含义是和固定电话对应的, 是一个通信工具, 其主要功能是通过移动网络进行通信。然而经过 20 年的发展, 移动电话的功能已经远远超过了语音通信工具的功能。目前, 随着 3G 技术的发展, 移动电话正向着智能化的方向迈进, 未来的移动电话将成为多种工具的功能载体。

● 通信工具

移动电话的最基本功能是通信, 因此使用运营商提供的通信网络 (GSM 或者 CDMA 等) 进行语音通话依然是移动电话的最基本功能。除了传统的语音通话功能外, 移动电话的短消息 (SMS) 功能是其较早出现的功能, 这个功能也是使用通信网络实现的。作为通信工具的终端, 移动电话通常具有个人信息系统管理 (PIM) 方面的功能, 例如电话本等。

● 网络工具

随着数字业务使用的普遍化, 移动电话通常可以作为网络工具使用, 这样移动电话可以完成笔记本电脑的部分功能。移动电话在网络方面的主要功能包括浏览器、IM (即时信息)、邮件等, 基本包含了网络方面的大部分功能。

● 媒体播放器

随着多媒体技术的应用, 在移动电话上进行音频和视频播放已经成为经常使用的功能。目前用户对于移动电话的媒体播放功能的需求日益增加, 例如支持更多的音/视频格式, 支持更高分辨率的视频、更流畅的播放, 以及和网络结合的流媒体方面的功能。

● 媒体获取设备

移动电话与媒体获取设备的集成日益增强, 利用手机这个贴身设备记录丰富多彩的外部世界的确是一个很有吸引力的功能, 目前的移动电话通常都提供了照相机、录音机、摄像机等功能。其中移动电话中照相机的最高分辨率已经成了移动电话最重要的性能指标之一。

● 多类型的连接设备

移动电话体积小, 但是却可以具有多种连接方式, 例如 USB、红外、无线局域网、蓝牙、GPS 等。具有了这些连接方式, 可以让移动电话以多种方式和外部交互。

● 信息感知终端

随着传感器（Sensor）技术的发展，传感器已经可以嵌入到移动电话中。移动电话可以利用速度-加速度、温度、方向等传感器感知外部的信息。根据从外部获取的信息，移动电话同时可以作为集成化的信息感知终端使用，并可以结合移动电话的界面来创建更多的效果。

- 视频电话

视频电话是综合了多媒体和通信的综合技术，也是 3G 技术的重要内容。视频电话对系统多方面性能的要求较高，也是 3G 手机的技术难点。

- 具有友好和绚丽用户界面的设备

移动电话不仅需要具有实质性的功能，用户界面也是其重要的指标。目前的移动电话界面一方面需要友好，让用户容易学习和操作；另一方面又需要绚丽，具有良好的视觉效果。移动电话的用户界面（GUI）方面的设计，也是其竞争力的重要组成部分。

- 可以个性化定制的平台

随着移动电话功能的发展，用户的个性化需求也越来越强烈，从比较简单的自定义界面的颜色，到全面自定义手机的功能，同一款移动电话也越来越具有更多的自定义功能用以满足不同用户的需求，增加其竞争力。

- 概念性产品

随着技术的发展，移动电话用户越来越需要更有品质的产品，未来的移动电话将是一个具有品牌的概念性产品。

Android 是一个智能手机的平台，其技术上的优势主要包含以下一些方面：

- 全开放智能移动电话平台；
- 多硬件平台的支持；
- 使用众多的标准化技术；
- 核心技术完整、统一；
- 完善的 SDK 和文档；
- 完善的辅助开发工具。

由此可见，Android 系统各方面的特点和未来智能手机的方向是密切相关的，它也代表并引领目前的技术潮流。


1.2 Android 的开发工作

Android 是一个开放的系统，这个系统的体积非常庞大，不同的开发者在开发过程中并不需要掌握整个 Android 系统，只需要进行其中某一个部分的开发。从功能上来区分，Android 的开发分成 3 种类型：

- 第一，移植开发移动电话系统；
- 第二，Android 应用程序开发；
- 第三，Android 系统开发。

从商业模式的角度来看，第一种类型的开发和第二种类型的开发是 Android 开发的主流。事实上，移动电话的制造者主要进行第一种类型的开发，其产品是 Android 手机；公司、个人和团体都可以进行第二种类型的开发，其产品是不同的 Android 应用程序。

在 Android 的开发过程中，每一种类型的开发都只涉及整个 Android 系统的一个子集。在 Android 系统中，有着众多的开发点，这些开发点相互独立，又有内在联系。在开发的过程中，需要重点掌握目前开发点涉及的部分。

 **提示：**Android 是一个比较庞大的软件系统，在开发过程中，应该重点掌握开发点，系统中还有很多比较固定的部分，一般不需要进行开发。

第一种类型和第二种类型的开发如图 1-5 所示。

第一种开发是构建硬件系统，并且移植 Android 的软件系统，最终形成手机产品。第二种开发是开发出各种 Android 应用程序，然后将这些应用程序投入 Android 市场，进行交易。

1.2.1 Android 移植开发

Android 移植开发的最终目的是为了开发手机产品，从开发者的角度来看，这种类型的开发以具有硬件系统为前提，在硬件系统的基础上构建 Android 软件系统。这种类型的开发工作在 Android 系统的底层。在软件系统方面，主要的工作集中在以下两个方面：

- Linux 中的相关设备驱动程序

驱动程序是硬件和上层软件的接口，在 Android 手机系统中，需要基本的屏幕、触摸屏、键盘等驱动程序，以及音频、摄像头、电话的 Modem、Wifi、蓝牙等多种设备驱动程序。

- Android 本地框架中的硬件抽象层

在 Android 中，硬件抽象层工作在用户空间，介于驱动程序和 Android 系统之间。Android 系统对硬件抽象层通常都有标准的接口定义，在开发过程中，实现这些接口也就给 Android 系统提供了硬件抽象层。

Android 移植开发的结构如图 1-6 所示。

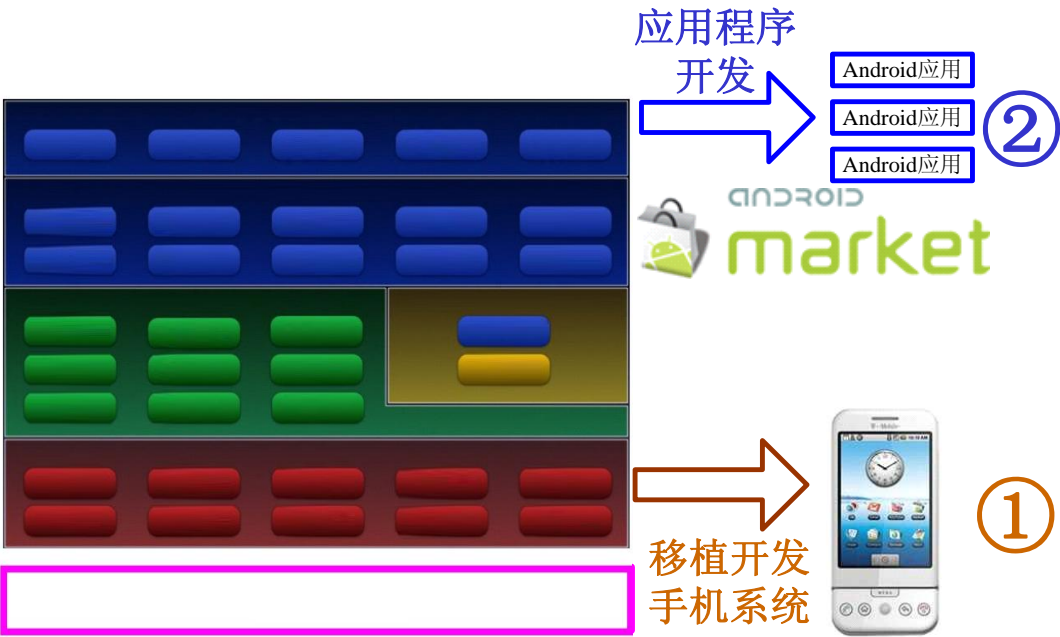


图 1-5 Android 的移植开发手机和应用程序开发

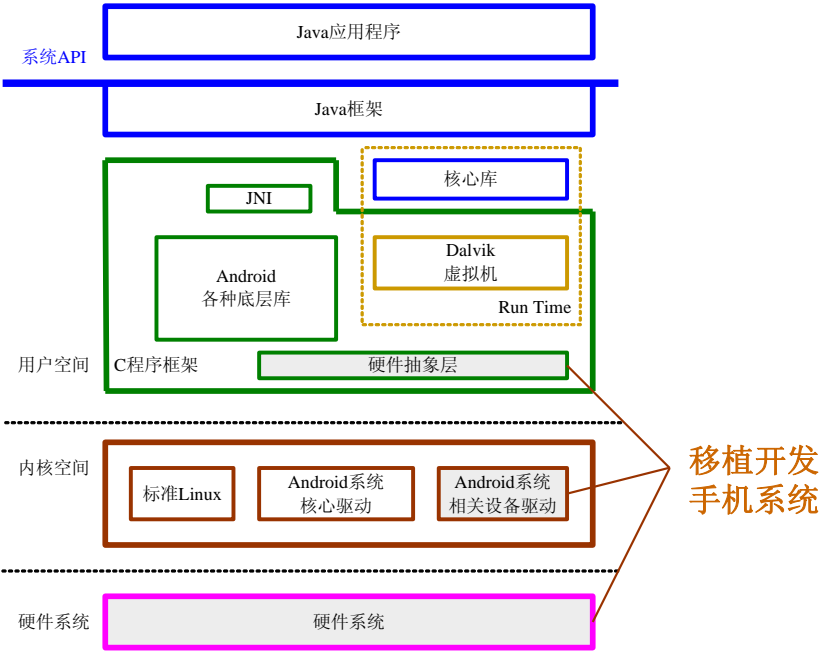


图 1-6 Android 移植开发的结构

上面的两个部分综合起来，即完成了 Android 系统的软件移植。移植成功与否取决于驱动程序的品质和对 Android 硬件抽象层接口的理解程度。

1.2.2 Android 应用开发

Android 的应用程序开发是 Android 开发的另一个方面，从开发的角度来看，这种形式的开发可以基于某个硬件系统，在没有硬件系统的情况下也可以基于 Linux 或者 Windows 下的 Android 模拟器来开发。这种类型的开发工作在 Android 系统的上层。

事实上，在 Android 软件系统中，第 3 个层次（Java 框架）和第 4 个层次（Java 应用）之间的接口也就是 Android 的系统接口（系统 API）。这个层次是标准的接口，所有的 Android 应用程序都是基于这个层次的接口开发出来的。Android 系统的第 4 个层次就是一组内置的 Android 应用程序。

作为 Android 应用程序开发者,其开发的应用程序其实和 Android 系统的第 4 个层次的应用程序是一个层次的内容。例如,Android 系统提供了基本的桌面程序,开发者可以根据 Android 的系统接口,实现另外一个桌面程序,提供给用户安装使用;根据 Android 系统的接口开发游戏,也是 Android 应用程序开发的一个重要方向。

Android 应用程序开发的结构如图 1-7 所示。

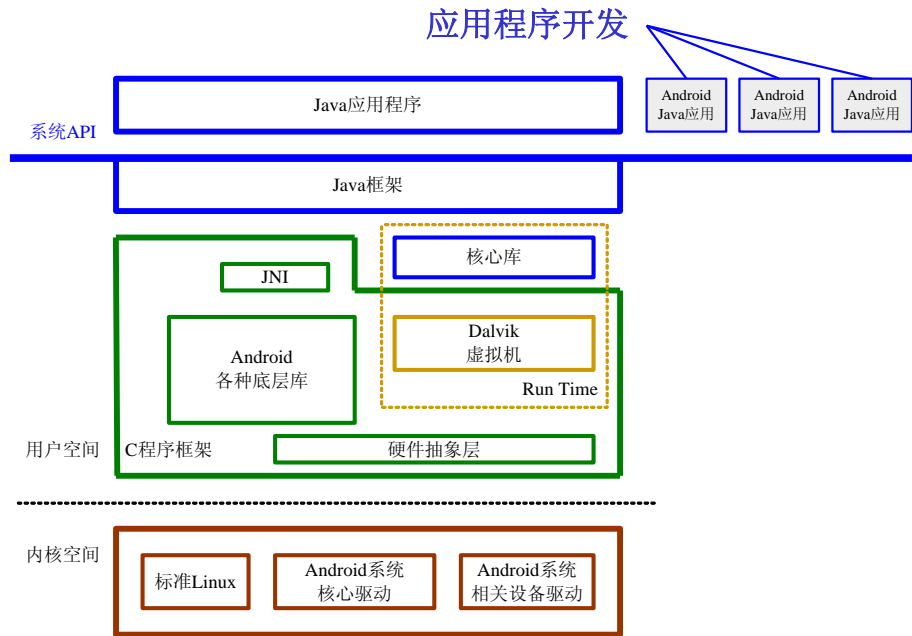


图 1-7 Android 应用程序开发的结构

1.2.3 Android 系统开发

Android 系统本身的功能也在增加和完善的过程中，因此 Android 系统的开发也是一个重要的方面。这种类型的开发涉及 Android 软件系统的各个层次。

Android 系统开发的一个比较典型的示例就是当系统需要某种功能时，为了给 Java 层次的应用程序提供调用的接口，需要从底层到上层的整体开发，步骤如下所示：

- ① 增加 C 或者 C++本地库；
- ② 定义 Java 层所需要的类（系统 API）；
- ③ 将所需要的代码封装成 JNI；
- ④ 结合 Java 类和 JNI；
- ⑤ 应用程序调用 Java 类。

如果涉及 Android 系统 API 的改动，则一定要慎重，因为系统 API 的改动可能涉及 Android 应用程序的不兼容。在更多的时候，Android 系统开发只是在不改变系统 API 的情况下修正系统的缺陷，增加系统的稳定性。

Android 系统开发的结构如图 1-8 所示。

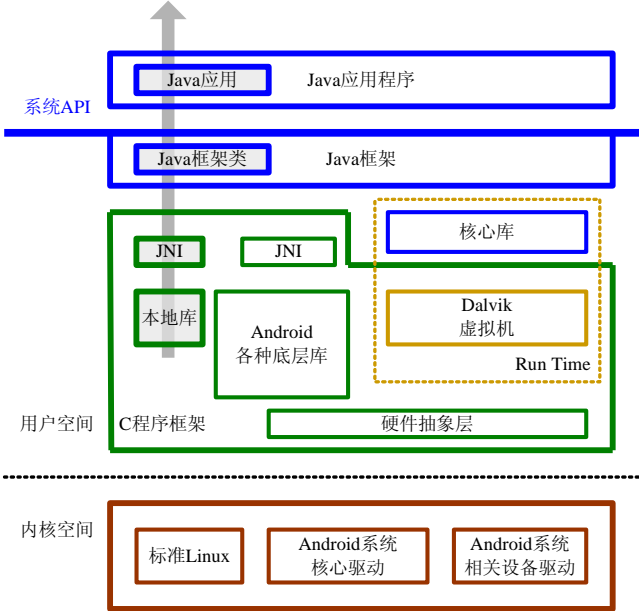



图 1-8 Android 系统开发的结构

 **提示:** Android 系统 API 的接口是 Java 语言, 在通常情况下, 更改接口函数的格式(参数、返回值)、常量的值等内容属于更改系统 API。事实上, 在不改变函数格式的情况下, 更改了函数的行为, 这其实也属于更改了系统 API 的情况, 可能造成已有程序的不兼容。

1.3 Android 的 SDK 与源代码

Android 开发系统主要集中在两种环境中, 分别是基于 Android SDK 的开发和基于 Android 源代码的开发。前者可以在 Linux 或者 Windows 两种环境中使用 IDE 完成, 后者需要在 Linux 环境中进行开发。

1.3.1 基于 SDK 的 Android 开发

基于 SDK 的 Android 开发, 使用 Google 提供的 SDK。Android 的 SDK 中包含 Android 系统下层的二进制映像、模拟器及相关的工具, 在 Linux 和 Windows 系统中使用 IDE 环境进行 Android 应用程序的开发。

不同版本的 Android 的 SDK 如下:

- android-sdk-linux_x86-1.0: Linux 环境中 Android 1.0 的 SDK;
- android-sdk-windows_x86-1.0: Windows 环境中 Android 1.0 的 SDK;
- android-sdk-linux_x86-1.5_r3: Linux 环境中 Android 1.5 R3 的 SDK;
- android-sdk-windows_x86-1.5_r3: Windows 环境中 Android 1.5 R3 的 SDK;
- android-sdk-linux_x86-1.6_r1: Linux 环境中 Android 1.6 R1 的 SDK;
- android-sdk-windows_x86-1.6_r1: Windows 环境中 Android 1.6 R1 的 SDK。

基于 SDK 的 Android 开发所对应的网站是 Google 的 Android 开发者 (Android Developers), 网址为: <http://developer.android.com/>。

Android 开发者的标志如图 1-9 所示。



图 1-9 Android 开发者的标志

基于 SDK 的 Android 开发仅限于开发 Android 的应用程序, 其开发成果即 Android 最上层的 Java 应用程序包。这种开发环境中的开发主要使用 Java 语言, 不需要使用硬件, 不需要涉及 Android 系统的底层, 只需要了解 Android 系统的 API。由于可以使用主机的仿真环境, 因此有各种成熟的工具可以使用。

基于 SDK 的 Android 开发, 所涉及的知识体系包含以下一些内容:

- Java 语言知识 (通用知识);
- Android 应用程序架构;
- Android GUI 设计基础知识;
- Android 中各种视图的使用;
- Android 中 2D/3D 图形 API;
- Android 中应用程序的设计思想。

 **提示:** Android 的应用程序层虽然需要设计很多 Android 特定的架构和接口, 但是通用的 GUI 设计思想依然适用。

1.3.2 基于源代码 SDK Android 开发

基于源代码 Android 的开发在获取所有源代码的基础上进行，可以开发 Android 应用程序，进行系统移植或者开发 Android 系统本身。

Android 源代码工程的网址为：<http://source.android.com/>。

Android 源代码工程的标志如图 1-10 所示。



图 1-10 Android 源代码工程的标志

Android 源代码可以从其 git 的源代码仓库中获得：android.git.kernel.org。

 **提示：**这个网络地址可以使用 git 和 repo 工具操作，也可以使用浏览器按照网页的方式（HTTP 协议）来访问。

基于源代码的 Android 开发，所涉及的知识体系包含以下一些内容：

- Linux 操作系统的基础知识；
- Linux 内核知识（C 语言）；
- Linux 驱动程序知识（C 语言）；
- Android 底层库（C 语言、C++）；
- Android 的 Java 运行环境（C++、Java）；
- Android GUI 系统（C++、Java）；
- 音频、视频和多媒体（C 语言、C++、Java）；
- 电话部分（C 语言、C++、Java）；
- 蓝牙、Wifi、定位系统（C 语言、C++、Java）；
- 传感器系统（C 语言、C++、Java）。