

第 1 章

Android 概述

说到智能手机，大家可能马上就能联想到 Android（中文翻译为“安卓”）、iOS 和 Windows Phone 等手机操作系统。确实，在当今，智能手机是如此的流行，以至于上至老年人，下至几岁孩童都在使用智能手机。市场研究公司 IDC 称，在 2018 年，Android 手机销量在全球市场上的占比达到 85.2%。在未来五年内，Android 手机的销量有望达到 2.4% 的复合年均增长率，到 2022 年将达到 14.1 亿部。

可以说 Android 占据了绝对优势。因此，作为应用开发人员，开发基于 Android 的手机应用是一个重要的方向。

1.1 Android 是什么

Android 是如此的流行，那么，Android 到底是什么？这个问题有点抽象，但是，作为即将踏入 Android 开发阵营的读者需要了解这个问题的答案。注意，只需要了解就可以了，即使在阅读这段文字时对有些概念不太理解，这也是正常的，不会影响将来开发出高水平的 Android 应用。

Android 是什么？Android 是一个平台，它包括基础系统、开发工具和完整的文档。Android 平台是一个通用的计算平台，它采用 Linux 为其支撑操作系统，以 Java 作为其开发环境，通过编程实现完整的电话、视频、网络、界面设计等基础功能。Android 平台的体系结构如图 1-1 所示。

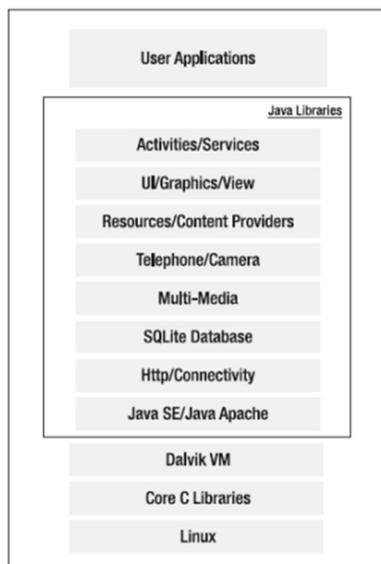


图 1-1 Android 平台的体系结构

我们采用从下往上的方式来介绍 Android 平台的体系结构的组成部分。

从图 1-1 可以看到，处在 Android 平台的底层的是 Linux 操作系统，在 Android 的平台体系结构下，Linux 提供了基础的支撑功能，包括设备管理、进程管理、文件管理等功能。按理说，Google 完全可以开发一套自己的操作系统而不采用 Linux 操作系统作为 Android 平台的支撑，可是，Linux 是如此成熟，应用又是如此广泛，并且还是开源的系统，为什么不用呢？为什么还要自己开发别人已经做得很好并且可以开源和自由使用的系统呢？这可能是 Google 使用 Linux 作为 Android 平台支撑的原因吧。

在图 1-1 中，从下往上，可以看到“Core C Libraries”，翻译为中文就是“核心 C 语言程序库”。为什么是这样呢？如前所述，Linux 是一个基础的支撑系统，可是在编写应用程序时如何使用 Linux 提供的功能呢？Linux 为了可以让应用程序使用它提供的功能而专门提供了一系列的 C 语言函数给应用程序调用，进而应用程序可以充分利用 Linux 操作系统提供的功能来达到应用程序的业务目标。作为 Android 应用开发者，除非将来要开发特定功能的应用程序，在一般情况下，读者不会直接使用到这些 C 语言函数库。当某一天要使用这个“Core C Libraries”开发 Android 应用程序时，恭喜你，你是 Android 应用开发的行家了。

图 1-1 再往上就是“Dalvik VM”，什么是 Dalvik VM？它没有那么神秘，Dalvik VM 其实就是一个变形的 Java 虚拟机（Java VM），也就是一个与普通 Java 虚拟机有一点区别的 Java 虚拟机而已。可是，Google 为什么要实现一个变形的 Java 虚拟机呢？原因很简单，因为普通的 Java 虚拟机在普通的桌面计算机上可以运行得很好，可是在小型的设备，如手机上就不一定了，为此，Google 实现了一个变形的 Java 虚拟机，并命名为 Dalvik VM。还有一个问题需要回答：为什么 Dalvik VM 需要出现在这里呢？Android 应用开发是基于 Java 语言进行的，Dalvik VM 就像我们开发桌面 Java 应用程序一样，我们需要一个 Java 虚拟机来运行开发的 Java 程序。

图 1-1 再往上就是一组用一个框框起来的“Java Libraries”，也就是一组 Java 类库。我们先来问第一个为什么：为什么 Java Libraries 会出现在这里？我们已经知道，Android 应用程序是基于 Java 语言开发的，为了开发 Java 应用程序，我们需要使用 Java 类库来完成某些基本的功能，因此，这里的“Java Libraries”就是在编写 Android 应用程序时可以使用的一组 Java 类库。第二个问题：这个 Java Libraries 中都包含了哪些 Java 类库？Java Libraries 中很多类库读者可能都不认识，也应该不认识，因为这些不认识的类库正是本书要介绍的内容，可是其中的一个读者一定认识：Java SE。是的，这个 Java SE 就是读者学习 Java 语言时的那个 Java SE，这个类库中包含像 String、Integer、File 等基础的类库，读者可以像使用 Java SE 类库中的类一样来使用其他类库中的类。

图 1-1 再往上就是“User Applications”，也就是用户应用程序，我们将来开发的所有 Android 应用程序都归属于这里。那么，如何来开发 Android 应用程序呢？这就是本书的目标。

我们从下往上介绍了 Android 平台的体系结构，现在，我们再从上往下将以上介绍的内容串起来，然后来看看如何开发一个 Android 应用程序，以及开发的 Android 应用程序是如何在 Android 设备上运行的：采用 Java 语言来开发一个 Android 应用程序（也就是一个 User Application），在开发应用程序时，可以使用“Java Libraries”中提供的 Java 类库来实现所要求的功能，当开发完成以后，可以在 Dalvik VM 上运行应用程序，Dalvik VM 会解释 Java 代码并在 Linux 操作系统上来执行代码，进而完成 Android 应用所要完成的业务功能。

1.2 Android 应用程序的组成

我们对 Android 体系有一个初步了解后，作为一个开发者，读者一定想知道 Android 是由哪些部分组成的。如下这些概念对于一个 Android 应用开发初学者来说，有些难于理解，但不要太纠结于这些概念，知道这些概念就好。我们会尽量采用通俗易懂的语言来介绍这些概念，使读者对 Android 程序的组成有一个初步的了解。

任何一个应用程序都会包括如下一些基本内容：应用程序的界面、业务功能的处理、部件之间的数据交互、数据存储。如此而已，Android 应用程序也不例外，只是被赋予了不同的名称而已。具体来说，一个 Android 应用程序包括如下的基本组成部分。

(1) Activity (窗体)。

在 Android 应用程序中，一个界面就是一个 Activity。这个名字有点与众不同，Google 经常会创造一些新名称，不是吗？既然 Activity 就是一个界面，因此，对每个 Activity，在进行设计时，都包括对界面的布局（Android 提供了丰富的 UI 组件来实现绚丽的界面）、对界面组件的点击会进行相应的事件处理等程序设计工作。可以将 Android 的 Activity 界面类比为 Internet 网页的一个页面。

(2) View (窗体组件)。

View 就是构建应用程序界面的基本组件，也就是说，Activity 界面是由一个或多个 View 构成的，例如，Button、Label、Text Field 等都是 View，View 是构建 Activity 的基本元素。

(3) Intent (窗体间或应用之间的通信组件)。

一般来说，一个 Android 应用程序会包括多个界面，用户在进行不同的操作时可能会进行不同界面之间的切换，就像在 Internet 的页面之间，当用户点击不同的页面链接时会进行不同页面之间的切换一样。在 Internet 页面之间的切换是通过链接来完成的，在 Android 的 Activity 之间实现不同的 Activity 界面之间的切换是通过称为 Intent 的对象来完成的。因此，可以这么说，Intent 是 Android 应用程序界面之间及功能部件之间实现信息交互的桥梁。

(4) Content Provider (应用之间数据交互方式)。

Content Provider，也就是内容提供者，是 Android 建议的应用程序之间进行数据交互的方式。举例来说，如果一个应用程序希望将自己的数据提供给其他应用程序使用，则该应用程序需要实现 Content Provider 接口，这样其他的应用程序便可以通过这个接口访问这个程序所提供的数据库。一个典型的实现了 Android 的 Content Provider 接口的程序是通讯录程序，任何需要使用通讯录数据的程序都可以通过该接口从电话通讯录程序中获得通讯录数据。

(5) Service (无窗口的在后台默默运行的程序)。

所谓 Service 就是运行于后台的程序。一般来说，Service 程序没有用户界面，它们运行于后台并为运行在前端的程序提供服务。Android 的 Service 程序在运行方式上类似于 Windows Phone 操作系统中的后台进程：它们在安静的运行，并在需要的时候为其他程序提供服务。

(6) 广播接收器 (信息广播方式)。

广播接收器，即 Broadcast Receiver。Android 平台中的程序在运行时会发生任何可能的事件，某个应用程序在运行时可能会将它的事件广播出来，其他的程序可以监听这样的事件，并对发生的事件进行必要的处理。举例来说，Android 的电池电量监视程序（这是 Android 的一个 Service）在随时监视着电池的电量，当电池的电量低于某个门限值时，该程序会广播一个消息，而其他的应用程序可以监听这个消息，并针对这个事件做出必要的处理，例如，一个正

在进行高耗电运算的程序监听到这个消息时，应该停止进行高耗电的运算，以便减少对电量的消耗。

(7) `AndroidManifest.xml` 文件（应用程序描述文件）。

`AndroidManifest.xml` 文件是 Android 应用程序的配置文件：它将构成 Android 应用程序的各个组件有效地装配起来从而构成一个完整的 Android 应用程序。每个应用程序一定会包含一个且只能包含一个配置文件。

1.3 Android 的发展历史

手机的常用操作系统有：Symbian OS、Microsoft Windows Mobile、Mobile Linux、iOS 等，这其中没有任何一个操作系统是事实上的标准，也没有任何一个操作系统是开源的。为此，Google 创立了 Android 移动平台。

2007 年，Google 牵头建立了开放手机联盟（Open Handset Alliance），到 2009 年，这个联盟成员包括 Sprint Nextel、T-Mobile、Motorola、Samsung、Sony Ericsson、Toshiba、Vodafone、Google、Intel、Texas Instruments 等 IT 巨头，到 2011 年，成员已近 80 家，Android 已经成为移动设备事实上的行业标准。到 2018 年 8 月，Android 平台的版本已从 1.0 发展到了 9.0。Android 是进化得如此之快，因此，读者在学习 Android 开发时，必须要学会使用 Android 的在线帮助文档。

1.4 Android 应用开发环境概述

Android 平台采用 Java 语言作为应用程序开发语言。Android 开发环境包括如下基本内容：Java 基本包、Android 基础组件、Android UI 组件、Android 服务组件、Android 电话和媒体服务组件、Android 仿真器（Android Virtual Device, AVD）、Android 调试器等。

笔者将在后续章节中，对这些内容进行详细介绍。首先从建立 Android 开发环境开始讲解。