Android 源码分析实录

基于当今市面主流版本
全面讲解Android系统的整体架构
细致剖析14大核心系统
底层、HAL层和应用层学习循序渐进

李忠良 编著

讲解详尽，深入底层

本书详细讲解了Android源码分析的每一个知识点，为了更加透彻地说明原理，从深入底层着手，到顶层Java应用结束，即使菜鸟也能够看懂并掌握。

大话模式，趣味性强

全书采用诙谐、生动的**大话模式讲解实例**，在逼真的生活场景中学习编程，将复杂的高深专业知识，以趣味性的语言讲解出来。

实例典型，提示丰富

书中的实例典型，融合了技术中所有的**经典范例**，以加深读者对知识的掌握。

高深内容详细剖析，做到一应俱全

作为某项专业技术，用最合理的篇幅**详细剖析了每个知识点**，内容涉及了领域内的方方面面，可直接作为此领域的权威书籍。

清华大学出版社

Android 源码分析实录

李忠良　编著

清华大学出版社
北京

内 容 简 介

Android 是一款服务于智能手机和平板电脑等设备的操作系统，截止作者撰写此书时为止，Android 在智能手机操作系统市场中已经占有 75%的份额。为了让广大读者充分了解这款神奇的操作系统的架构原理，本书循序渐进地分析了 Android 系统核心源码的基本知识。

本书共分为 15 章，主要内容包括走进 Android 世界、硬件抽象层详解、分析 JNI(Java 本地接口)层、Android 内存系统分析、Android 虚拟机系统详解、IPC 通信机制详解、Zygote 进程/System 进程和应用程序进程、分析 Activity 组件、Content Provider 数据存储、Broadcast(广播)系统详解、多媒体系统详解、电源管理系统详解、输入系统驱动应用、蓝牙系统详解、网络系统详解等。

本书几乎涵盖了 Android 源码中的所有核心系统的内容，全书内容通俗易懂，适合 Android 初学者、Android 爱好者、Android 底层开发人员、Android 应用开发人员阅读和学习，也可以作为相关培训学校和大专院校相关专业的教学用书。

图书在版编目(CIP)数据

Android 源码分析实录/李忠良编著. --北京：清华大学出版社，2015
ISBN 978-7-302-39329-0

Ⅰ. ①A…　Ⅱ. ①李…　Ⅲ. ①移动终端—应用程序—程序设计　Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2015)第 024952 号

责任编辑：杨作梅
封面设计：杨玉兰
责任校对：马素伟
责任印制：沈　露

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
　　社 总 机：010-62770175　　**邮　　购**：010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：三河市君旺印务有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：46.25　　**字　　数**：1124 千字
版　　次：2015 年 4 月第 1 版　　**印　　次**：2015 年 4 月第1 次印刷
印　　数：1～3000
定　　价：89.00 元

产品编号：054375-01

前　言

Android(中文译名为安卓)是 IT 界巨头 Google(谷歌)公司于 2007 年 11 月 5 日推出的一款智能操作系统，最初被应用于智能手机，后来随着版本的更新和发展，也被广泛应用于平板电脑、智能电视、可穿戴设备和健康设备中。Android 是一款基于 Linux 平台的开源操作系统的名称，根据国际数据公司(IDC)公布的数据，Android 在智能手机操作系统中的市场占有率已经达到 75%。

高份额的市场占有率使得更多的开发人员把目光投入这款神奇的系统，很多初学者也纷纷涌入 Android 的学习行列中，配合这些需求，Android 的各种应用类图书不断涌现并广受欢迎。但美中不足的是，深入源码分析的书籍屈指可数。而源码分析正是通往 Android 殿堂、跻身为高手的阶梯。

为了让广大初学者可以对 Android 系统实现“灵与肉”的感知，而不是停留在抽象的原理和概念上，本书对 Android 系统的源码进行细致的分析，这样做的目的，是提炼出 Android 系统埋藏于深处的本质和精华的东西，以展示这款神奇的系统究竟是怎样实现的。

1．本书内容

Android 系统升级较快，有些代码变动很大。系统自 2007 年发布第一个版本 1.1 以来，截至 2013 年 7 月发布版本 4.3，中间一共存在十多个版本。但据官方统计，到 2013 年 5 月 5 日，占据前三位的版本分别是 Android 4.2，Android 4.1 和 Android 4.3，其实这三个版本的区别并不是很大，只是在某领域的细节上进行了更新。因此，在本书中，我们选择本书最初写作时的最新版本 Android 4.3 系统的实现。

本书共分 15 章，依次为走进 Android 世界、硬件抽象层详解、分析 JNI(Java 本地接口)层、Android 内存系统分析、Android 虚拟机系统详解、IPC 通信机制详解、Zygote 进程/System 进程和应用程序进程、分析 Activity 组件、Content Provider 数据存储、Broadcast(广播)系统详解、多媒体系统详解、电源管理系统详解、输入系统驱动应用、蓝牙系统详解、网络系统详解。

本书几乎涵盖 Android 源码中的所有核心系统的内容，全书通俗易懂，特别有利于初学者学习和消化。

2．本书特色

本书内容十分丰富，分析细致、全面。我们的目标是通过一本图书，提供多本图书的价值，读者可以根据自己的需要，有选择地阅读。

在内容的编写上，本书具有以下特色。

(1) 结构合理

从用户的实际需要出发，科学安排知识结构。全书详细地讲解与 Android 应用开发有关的源码，内容循序渐进，由浅入深。

(2) 易学易懂

本书条理清晰、语言简洁，可帮助读者快速掌握每个知识点，使读者既可以按照本书编排

的章节顺序进行学习，也可以根据自己的需求，对某一章节进行有针对性的学习。

(3) 实用性强

本书彻底摒弃枯燥的理论知识罗列，注重实用性和可操作性，通过细腻的笔法，逐步讲解各个知识点的基本知识。

(4) 内容全面

本书是如今市面上“内容最全的 Android 源码分析书”，无论是获取源码，还是各个常用、常见的模块系统，在本书中您都能找到解决问题的答案。

3. 读者对象

本书适合下列人员阅读和学习：

- 初学 Android 编程的自学者。
- Android 源码分析人员。
- Android 底层开发人员。
- Android 系统开发人员。
- 相关培训机构的教师和学员。
- 从事 Android 开发的程序员。

4. 作者支持

在编写此书的过程中，得到了清华大学出版社工作人员的大力支持，正是由于各位编辑的求实态度、耐心的工作和奉献精神，才使得本书能够快速出版。

另外也十分感谢我的家人在我写作的时候给予的巨大支持。

由于作者水平有限，本书的疏漏之处在所难免，恳请读者提出意见或建议，以便再版时修订并使之更臻完善。我们提供了售后支持 QQ(号码为 1727069718)，读者如有疑问可以通过 QQ 提出，将会得到满意的答复。

编　者

目　录

第 1 章
走进 Android 世界

Android 系统于 2007 年诞生，是一款建立在 Linux 内核之上的智能设备系统，是一款经典的手机、平板电脑等移动设备的软件解决方案。从 2011 年下半年开始到现在，Android 系统在全球智能手机操作系统中的占有率一直位居第一。

本章将简单介绍 Android 系统的发展历程和背景，让读者了解 Android 系统的发展之路，充分体验这款无与伦比的操作系统。

1.1 Android 系统的优势

为什么 Google 的 Android(安卓)系统能够在短短 4 年内超越了 Symbian(塞班)、Blackberry(黑莓)、iOS 等前辈，从一名后起之秀变为移动智能设备市场占有率的大佬？这需要从 Android 系统的优势谈起，在本节的内容中，将为读者展示这些优势。

1.1.1 开源

Google 的 Android 出身于 Linux 世家，是一款开源的手机操作系统。正因为如此，在 Android 崭露头角之后，各大手机厂商和电信部门纷纷加入到了 Android 联盟中。这个联盟由业界内的公认大佬组成，主要成员包括 Google、中国移动、摩托罗拉、高通和 T-Mobile 等在内的 30 多家技术和无线应用的领军企业。Android 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，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。

开源意味着对开发人员和手机厂商来说，Android 是完全无偿免费使用的。正是因为源代码公开的原因，所以吸引了全世界各地无数程序员的热情。于是很多手机厂商都纷纷采用 Android 作为自己产品的系统，这当然也包括很多山寨厂商。因为免费，所以降低了成本，因而提高了利润。而对于开发人员来说，因为 Android 被众多移动设备产品所采用，所以这方面的人才也变得愈发抢手。于是有一些在别的系统上干得还可以的程序员也改行做 Android 开发，纷纷加入到 Android 开发大军中来，原因是待遇更好；另外，也有很多混得不尽如人意的程序员更是纷纷改行做 Android 手机开发，目的是想寻找自己程序员生涯的转机。

而像本书作者这样遇到发展瓶颈的程序员，后来也决定做 Android 开发，因为这样可以学习一门新的技术，使自己的未来更加有保障。

1.1.2 强大的开发团队的支持

Android 的研发队伍阵容强大，包括 Google、摩托罗拉、HTC(宏达电子)、Philips、T-Mobile、高通、魅族、三星、LG 以及中国移动在内的 34 家企业，这些企业都基于 Android 平台开发手机的新型业务，并使应用之间的通用性和互联性在最大程度上得到保持。从硬件到软件开发机构，再到电信服务商，Android 从一开始便成为业界内的宠儿，被当作新秀而重点培养，在强大的开发团队的培育和呵护下，顺利地功成名就，成为一方霸主。

1.1.3 开发人员的支持

Google 一直视程序员为前进的动力和源泉，为了提高程序员们的开发积极性，不但为开发人员提供了一流的开发装备和软件服务，而且还提出了振奋人心的奖励机制。

具体的开发人员支持主要体现在如下三个方面。

(1) 可以迅速步入 Android 应用开发。在 Android 平台上，程序员可以开发出各式各样的应用。Android 应用程序是通过 Java 语言开发的，只要具备 Java 开发基础，就能很快地上手并掌握。对于单独的 Android 应用开发来说，并没有很高的 Java 编程门槛，即使没有编程经验的

门外汉，也可以在突击学习 Java 之后很快掌握 Android 编程。另外，Android 完全支持 2D、3D 和数据库，并且与浏览器实现了集成。所以，通过 Android 平台，程序员可以迅速、高效地开发出绚丽多彩的应用，例如常见的工具、管理程序、互联网程序和游戏程序等。

(2) 可以参加奖金丰厚的 Android 大赛。为了吸引更多的用户使用 Android 开发，Google 定期举办奖金为数千万美元的开发者竞赛，鼓励开发人员做出创意十足的软件。这种大赛对于开发人员来说，不但能磨练自己的开发水平，并且高额奖金本身也是吸引学习的动力。

(3) 可以加入自由经营的贸易市场。为了能让 Android 平台吸引更多的关注，Google 提供了一个专门下载 Android 应用的门店：Android Market，网址是 https://play.google.com/store。在这个门店里面，允许开发人员发布应用程序，也允许 Android 用户下载自己喜欢的程序。作为开发者，需要申请开发者账号，申请后才能将自己的程序上传到 Android Market，并且可以对自己的软件进行定价。只要你的软件程序足够吸引人，就可以获得很好的回报。学习和赚钱两不误，我们何乐而不为呢？

1.2　Android 系统架构介绍

Android 是一个移动设备的开发平台，其软件层次结构大体上包括操作系统(OS)、中间件(Middleware)和应用程序(Application)。

Android 操作系统的组件结构如图 1-1 所示。

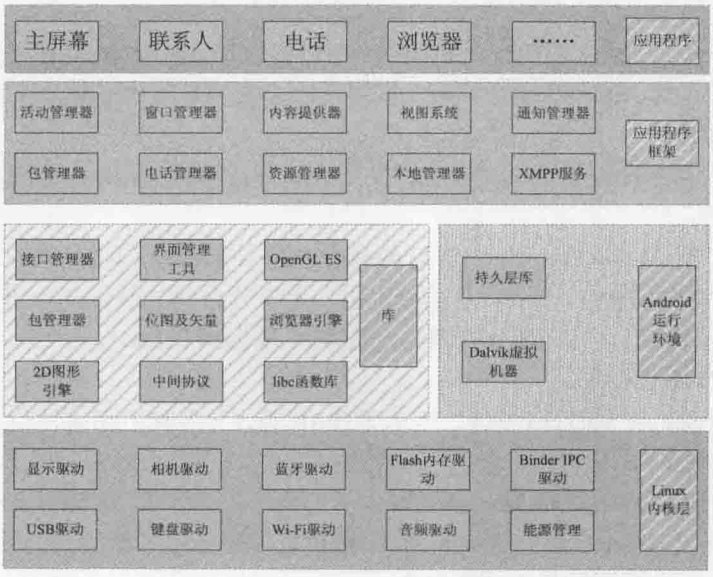

图 1-1　Android 操作系统的组件结构

根据 Android 操作系统的组件结构框图可知，其软件层次结构自下而上分为 4 层。

(1) 操作系统层(即 Linux 内核层)。

(2) 各种库(Libraries)和 Android 运行环境(Runtime)。

(3) 应用程序框架(Application Framework)。

(4) 应用程序(Application)。

1.2.1 底层操作系统层(Linux 内核层)

因为 Android 源于 Linux，使用了 Linux 内核，所以 Android 使用 Linux 2.6 作为操作系统。Linux 2.6 是一种标准的技术，Linux 也是一个开放的操作系统。Android 对操作系统的使用包括核心和驱动程序两部分，Android 的 Linux 核心为标准的 Linux 2.6 内核，Android 更多的是需要一些与移动设备相关的驱动程序。主要的驱动程序如下所示。

- 显示驱动(Display Driver)：是常用的基于 Linux 的帧缓冲(Frame Buffer)驱动程序。
- Flash 内存驱动(Flash Memory Driver)：是基于 MTD 的 Flash 驱动程序。
- 照相机驱动(Camera Driver)：常用基于 Linux 的 V4L(Video for Linux)驱动程序。
- 音频驱动(Audio Driver)：常用基于 ALSA(Advanced Linux Sound Architecture，高级 Linux 声音体系)的驱动程序。
- Wi-Fi 驱动(Wi-Fi Driver)：基于 IEEE 802.11 标准的驱动程序。
- 键盘驱动(Keyboard Driver)：作为输入设备的键盘驱动程序。
- 蓝牙驱动(Bluetooth Driver)：基于 IEEE 802.15.1 标准的无线传输技术驱动程序。
- Binder IPC 驱动：具有单独的设备节点，提供进程间通信功能的特殊驱动程序。
- Power Management(电源管理)：用于管理电池电量等信息的驱动程序。

1.2.2 库(Libraries)和运行环境(Runtime)

本层次对应一般的嵌入式系统，相当于中间件层次。Android 的本层次分成两个部分，一个是各种库，另一个是 Android 运行环境。本层的内容大多是使用 C 语言实现的，其中包含了如下所示的各种库。

- C 库：C 语言的标准库，也是系统中一个最为底层的库，C 库是通过 Linux 的系统调用实现的。
- 多媒体框架(Media Framework)：这部分内容是 Android 多媒体的核心部分，基于 Packet Video(即 PV)的 Open core，从功能上看，本库分为两大部分，一部分是音频、视频的回放(Play Back)，另一部分是则是音视频的记录(Recorder)。
- SGL：2D 图像引擎。
- SSL：即 Secure Socket Layer，位于 TCP/IP 协议与各种应用层协议之间，为数据通信提供安全支持。
- OpenGL ES：提供了对 3D 的支持。
- 界面管理工具(Surface Management)：提供管理显示子系统等功能。
- SQLite：一个通用的嵌入式数据库。
- WebKit：网络浏览器的核心。
- FreeType：位图和矢量字体的功能。

在一般情况下，Android 的各种库是以系统中间件的形式提供的，它们的显著特点是与移动设备平台的应用密切相关。

另外，Android 的运行环境主要是基于 Dalvik(虚拟机)技术的。而 Dalvik 与一般的 Java 虚拟机(Java Virtual Machine，JVM)是有如下区别的。

- Java 虚拟机：执行的是 Java 标准的字节码(Bytecode)。
- Dalvik：执行的是 Dalvik 可执行格式(.dex)的文件。在执行的过程中，每一个应用程序即一个进程(Linux 的一个 Process)。

二者最大的区别在于，JVM 是基于栈的虚拟机(Stack-based)，而 Dalvik 是基于寄存器的虚拟机(Register-based)。显然，后者最大的好处在于可以依据硬件实现更大的优化，这更适合移动设备的特点。

1.2.3　应用程序框架(Application Framework)

在整个 Android 系统中，与应用开发最相关的是 Application Framework，在这一层，Android 为应用程序层的开发者提供了各种功能强大的 APIs，这实际上是一个应用程序的框架。由于上层的应用程序是以 Java 构建的。在本层提供了程序中所需要的各种控件，例如：Views(视图组件)、List(列表)、Grid(栅格)、Text Box(文本框)、Button(按钮)，甚至还有一个嵌入式的 Web 浏览器。

一个基本的 Android 应用程序可以利用应用程序框架中的以下 5 个部分。

- Activity：活动。
- Broadcast Intent Receiver：广播意图接收者。
- Service：服务。
- Content Provider：内容提供者。
- Intent and Intent Filter：意图和意图过滤器。

1.2.4　顶层应用程序(Application)

Android 的应用程序主要是用户界面(User Interface)方面的，本层通常使用 Java 语言编写，其中还可以包含各种被放置在“res”目录中的资源文件。Java 程序和相关资源在经过编译后，会生成一个 APK 包。Android 本身提供了主屏幕(Home)、联系人(Contact)、电话(Phone)、浏览器(Browsers)等众多的核心应用。同时，应用程序的开发者还可以使用应用程序框架层的 API 实现自己的程序。这也是 Android 开源的巨大潜力的体现。

注意： 我们目的是分析 Android 系统的源码，因为涉及的知识面涵盖了所有上述 4 个层次，所以学习过程任重而道远。

1.3　核 心 组 件

一个典型的 Android 程序通常由 5 个组件组成，这 5 个组件构成了 Android 的核心功能。在分析 Android 4.3 源码之前，本节将详细讲解 Android 应用程序的核心组件的基本知识。

1.3.1　Activity 的界面表现

Activities 是这 5 个组件中最常用的一个组件。程序中 Activity 通常的表现形式是一个单独

的界面(screen)。每个 Activity 都是一个单独的类，它扩展实现了 Activity 基础类。这个类显示为一个由 Views 组成的用户界面，并响应事件。大多数程序有多个 Activity。例如，一个文本信息程序有这么几个界面：显示联系人列表界面、写信息界面、查看信息界面或者设置界面等。每个界面都是一个 Activity。切换到另一个界面就是载入一个新的 Activity。某些情况下，一个 Activity 可能会给前一个 Activity 返回值——例如，一个让用户选择相片的 Activity 会把选择到的相片返回给其调用者。

打开一个新界面后，前一个界面就被暂停，并放入历史栈中(界面切换历史栈)。使用者可以回溯前面已经打开的存放在历史栈中的界面。也可以从历史栈中删除没有界面价值的界面。Android 在历史栈中保留程序运行产生的所有界面：从第一个界面，到最后一个。

1.3.2 Intent 和 IntentFilters 界面切换

Android 通过一个专门的 Intent 类来进行界面的切换。Intent 描述了程序想做什么(Intent 意为意图，目的，意向)。Intent 类还有一个相关类 IntentFilter。Intent 是一个请求，用来明确做什么事情，IntentFilter 则描述了一个 Activity(或下文的 IntentReceiver)能处理什么意图。显示某人联系信息的 Activity 使用了一个 IntentFilter，就是说，它知道如何处理应用到此人数据的 VIEW 操作。Activities 在文件 AndroidManifest.xml 中使用 IntentFilters。

通过解析 Intents 可以实现 Activity 的切换，我们可以使用 startActivity(myIntent)启用新的 Activity。系统会考察所有安装程序的 IntentFilters，然后找到与 myIntent 匹配最好的 IntentFilters 所对应的 Activity。这个新 Activity 能够接收 Intent 传来的消息，并因此被启用。解析 Intents 的过程发生在 startActivity 被实时调用时，这样做有如下两个好处：

- Activities 仅发出一个 Intent 请求，便能重用其他组件的功能。
- Activities 可以随时被替换为有等价 IntentFilter 的新 Activity。

1.3.3 Service 服务

Service 是一个没有用户界面(UI)且长驻系统的代码，最常见的例子是媒体播放器从播放列表中播放歌曲。在媒体播放器程序中，可能有一个或多个 Activities 让用户选择播放的歌曲。然而在后台播放歌曲时无需 Activity 干涉，因为用户希望在音乐播放的同时能够切换到其他界面。既然这样，媒体播放器 Activity 需要通过 Context.startService()启动一个 Service，这个 Service 在后台运行以保持继续播放音乐。在媒体播放器被关闭之前，系统会保持音乐后台播放 Service 的正常运行。可以用 Context.bindService()方法连接到一个 Service 上(如果 Service 未运行的话，连接后还会启动它)，连接后就可以通过一个 Service 提供的接口与 Service 进行通话。对音乐 Service 来说，提供了暂停和重放等功能。

Android 系统将会尝试保留那些启动了的或者绑定了的服务进程，具体说明如下。

(1) 如果该服务正在进程的 onCreate()、onStart()或者 onDestroy()这些方法中执行，那么主进程将会成为一个前台进程，以确保此代码不会被停止。

(2) 如果服务已经开始，那么它的主进程的重要性会低于所有的可见进程，但是会高于不可见进程。由于只有少数几个进程是用户可见的，所以，只要不是内存特别低，该服务就不会停止。

(3) 如果有多个客户端绑定了服务，只要客户端中的一个对于用户是可见的，就可以认为该服务可见。

1.3.4　用 Broadcast IntentReceiver 广播

当要执行一些与外部事件相关的代码时，比如来电响铃时，或者到半夜时，就可能用到 IntentReceiver。尽管 IntentReceivers 使用 NotificationManager 来通知用户一些好玩事情的发生，但没有用户界面。

IntentReceivers 可以在文件 AndroidManifest.xml 中声明，也可以用 Context.registerReceiver() 来声明。当一个 IntentReceiver 被触发时，如果需要，系统自然会自动启动程序。程序也可以通过 Context.broadcastIntent()来发送自己的 Intent 广播给其他程序。

1.3.5　用 Content Provider 存储

在 Android 系统中，应用程序将数据存放在一个 SQLite 数据库格式的文件里，或者存放在其他有效设备里。如果想让其他程序能够使用我们程序中的数据，此时 Content Provider 就很有用了。Content Provider 是一个实现了一系列标准方法的类，这个类使得其他程序能存储、读取某种 Content Provider 可处理的数据。

1.4　进程和线程

在 Android 系统中，进程和线程用于完成某个 Android 任务。当第一次运行某个组件的时候，Android 会启动一个进程。在默认情况下，所有的组件和程序运行在这个进程和线程中，也可以安排组件在其他的进程或者线程中运行。在本节的内容中，将简要介绍 Android 系统中进程和线程的基本知识。

1.4.1　什么是进程

组件运行的进程是由 manifest 文件控制的。组件的节点一般都包含一个 process 属性，例如<activity>、<service>、<receiver>和<provider>节点。

属性 process 可以设置组件运行的进程，可以配置组件在一个独立进程中运行，或者多个组件在同一个进程中运行，甚至可以让多个程序在一个进程中运行，当然前提是这些程序共享一个 User ID 并给定同样的权限。另外，<application>节点也包含了 process 属性，用来设置程序中所有组件的默认进程。

当更加常用的进程无法获取足够的内存时，Android 会智能地关闭不常用的进程。当下次启动程序的时候，会重新启动这些进程。

当决定哪个进程需要被关闭的时候，Android 会考虑进程对用户是否还有用。例如 Android 会倾向于关闭一个长期不显示在界面的进程来支持一个经常显示在界面的进程。是否关闭一个进程，决定于组件在进程中的状态。