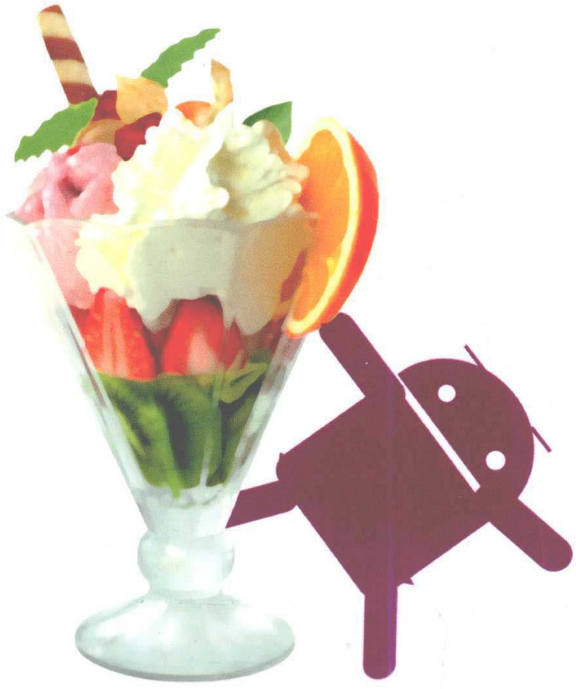
HZ BOOKS
华章科技

从源代码角度，系统、深入、透彻剖析Android系统框架层（Framework）的设计思想和实现原理，为Android应用工程师和系统工程师解决各种难题提供了原理性的指导！

资深Android内核专家亲自执笔，Lordhong等数位资深专家和安卓巴士等专业社区联袂推荐

移动开发



杨云君◎著

Design and Ir

Android : Volume I

Android的设计与实现

卷I

机械工业出版社
China Machine Press

移动开发

Design and Implementation of Android : Volume I

Android的设计与实现

卷I

杨云君◎著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Android的设计与实现：卷Ⅰ/杨云君著. —北京：机械工业出版社，2013.4

ISBN 978-7-111-41713-2

Ⅰ.A… Ⅱ.杨… Ⅲ.移动终端－应用程序－程序设计 Ⅳ.TN929.53

中国版本图书馆CIP数据核字（2013）第042844号

本书是Android应用开发工程师和Android系统工程师进阶修炼的必读之作。它由资深Android内核专家亲自执笔，从源代码角度，系统、深入、透彻剖析Android系统框架层（Framework）的设计思想和实现原理，为Android应用工程师和系统工程师解决实际工作中的各种难题提供了原理性的指导。为了降低读者的阅读成本，本书使用了大量简单的UML类图和序列图来展示类的层次结构和方法的调用流程，使读者能迅速读完本书并领会其精髓！

“Android的设计与实现”系列丛书主要围绕Android系统的四层结构展开，通过源代码来分析各层的设计思想与实现原理，卷I则主要是针对Framework（框架层）的。全书共12章，分为六个部分：基础篇（第1～2章）详细讲解了Android的体系结构、源代码阅读和调试环境的搭建，以及整个框架的基础；启动篇（第3～4章）深入分析了Android启动过程的机制和实现原理，能帮助读者全面理解框架层系统服务的运行基础；Binder篇（第5～6章）着重分析了Binder在Native框架层和Java框架层的机制和实现，能让读者深入理解进程间的通信模型；消息通信篇（第7章）重点分析了Android的消息驱动和异步处理机制，能让读者深入理解线程间的通信模型；Package Manager篇（第8～9章）主要讲解了Package Manager的机制与实现，以及APK的安装方法与过程；Activity Manager篇（第10～12章）深入阐述了ActivityManagerService的运行机制、应用程序和进程的启动流程，以及进程管理机制。

本书适合中高级的Android应用开发工程师、Android系统开发工程师、Android系统架构师，以及负责对Android系统进行调试和优化的工程师们阅读。

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：白 宇
北京市荣盛彩色印刷有限公司印刷
2013年5月第1版第1次印刷
186mm×240mm·27印张
标准书号：ISBN 978-7-111-41713-2
定价：79.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

为什么要写本书

Android 从 2007 年问世至今，不仅在各个应用领域发展得如火如荼，其图书市场也是一片“兴旺”，各个层次、各种类型的 Android 图书的需求都比较旺盛。目前市场上已经有的图书主要分为以下三类：

- 针对 Android SDK API 使用的描述
- 针对 Android 系统架构各部分的描述
- 针对 Kernel 移植的描述

其中鲜有针对 Android 四层架构中某一层进行深入挖掘的图书，这让读者有一种只能窥其全貌，却不能独得一隅的遗憾。

框架层是整个 Android 系统的灵魂，这一层起着承上启下的作用，是理解整个 Android 的关键，也是解决 Android 应用层 Bug 的关键。要开发一款精品手机，就必须深入理解这一层。

国际知名的手机厂商对手机品质有着近乎苛刻的要求，手机必须在严格的测试环境下运行数百小时无问题方可上市销售。这期间出现的稳定性（ANR、Crash、Watchdog）、内存（OOM）、性能等问题都让人十分头痛。这些问题主要来自于应用程序、Framework、Dalvik 虚拟机、Linux Kernel、Driver 以及 Modem，其中相当大一部分问题源自对 Framework 的错误理解和使用。举例如下：

- 解决 KeyDispatchTimeout 类型的 ANR，需要熟悉 Activity Manager、Input 消息处理系统的机制。
- 解决应用程序 IDLE 状态时发生的 ANR，需要熟悉 Activity Manager、Binder 的运行机制。
- 解决框架层的 Watchdog 问题，需要熟悉 Android 启动阶段开启的系统服务和 Watchdog 的运行机制。

- 解决应用程序的性能问题，同样需要理解框架层的运行和调度机制。

上述问题只是冰山一角，仅仅停留在使用 SDK API 的层次是不可能解决上述问题的。因此，非常需要一本能深入挖掘框架层的专著。

针对以上问题，编写“Android 的设计与实现”系列丛书，对 Android 核心模块和主要问题进行深入分析。其中卷 I 的主题是启动和通信，主要分析 Android 运行环境、Package Manager、Activity Manager、Binder 和消息机制等核心模块。卷 II 的主题是资源和 UI，主要分析 Content Provider、Resource、View System、Window Manager、SurfaceFlinger 等核心模块。

读者对象

本书主要分析了 Android 框架层主要部分的体系结构和实现原理，让读者对 Framework 有一个清晰的理解，并以此增强解决实际开发中遇到的 Bug 的能力。

本书适合以下几类读者：

- 中高级 Android 应用开发者
- Android 系统开发者
- Android 系统架构师
- 负责 Android 调试与优化的工作者

如何阅读本书

本书分为六大部分：

第一部分为基础篇（第 1 ~ 2 章），简要分析 Android 的体系结构、开发和调试环境、框架基础。

第二部分为启动篇（第 3 ~ 4 章），着重分析 Android 启动过程的机制和实现，让读者对框架层系统服务的运行基础有一个全面的理解。

第三部分为 Binder 篇（第 5 ~ 6 章），着重分析 Binder 在 Native 框架层和 Java 框架层的机制和实现，让读者掌握进程间通信模型。

第四部分为消息通信篇（第 7 章），着重分析 Android 的消息驱动和异步处理机制，帮助读者掌握线程间通信模型。

第五部分为 Package Manager 篇（第 8 ~ 9 章），着重分析 Android 应用程序的解析和安装流程。

第六部分为 Activity Manager 篇（第 10 ~ 12 章），着重分析 Activity Manager Service 的运行机制、应用程序和进程的启动流程、进程管理机制。

此外，本书在分析类的层次结构和方法的调用流程上，分别使用了 UML 的类图和序列图。为了不增加读者的阅读负担，本书只使用了 UML 中比较简单的图形。

类图如图 1 所示。

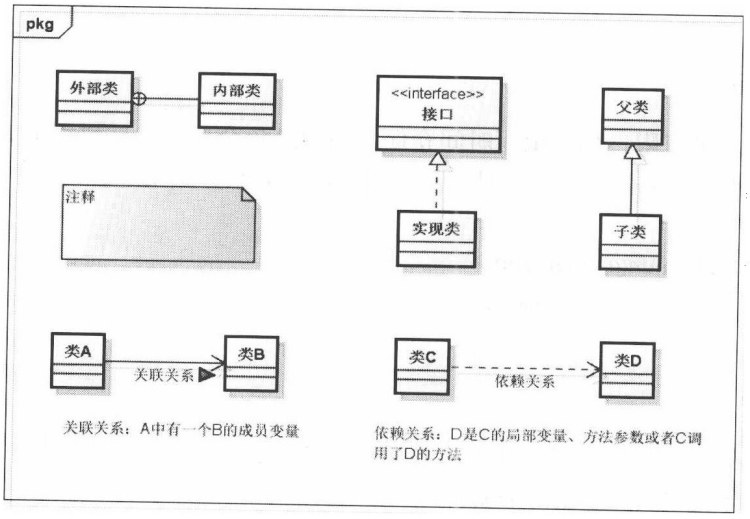


图 1　类图

图中展示了内部类和外部类的关系、接口和实现类的关系、父类和子类的关系、注释的表示法、类的关联关系和依赖关系。

为了简单鲜明，本书在绘制类图时，会省略部分类图中的方法参数和返回值，读者可参照源码查看方法的具体返回值和参数。

序列图如图 2 所示。

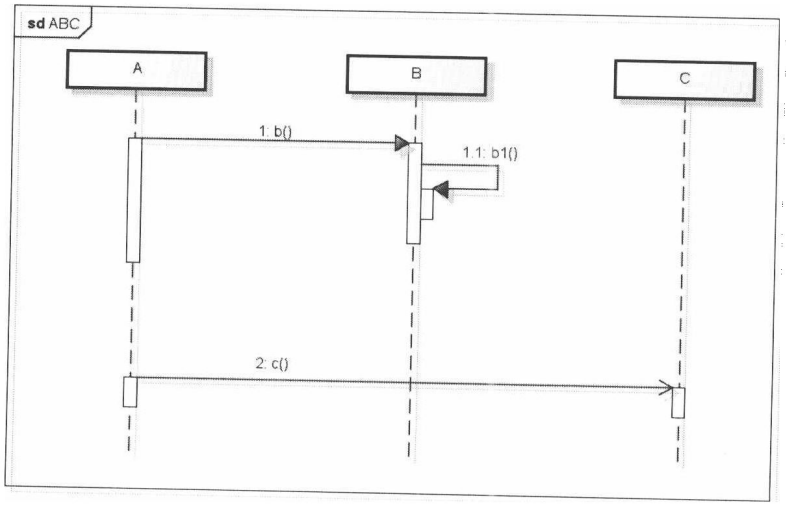


图 2　序列图

图中表示 A 调用了 B 的 b 方法，在 b 方法中又调用了 B 的 b1 方法，这两次调用都是同步调用，用实心箭头表示；A 调用 C 的 c 方法，这次调用是异步调用，用箭头表示。

勘误和支持

由于作者的水平有限，加之编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如果你有宝贵意见，欢迎通过以下联系方式与笔者讨论，期待能够得到你们的真挚反馈。

- 邮箱：yangyunjun@gmail.com
- 博客：blog.csdn.net/allongriver

致谢

感谢本书的策划编辑杨福川，他不仅对本书的写作思路提出了宝贵意见，而且在写作过程中一直鼓励和引导我顺利完成全部书稿。

感谢编辑白宇，她不仅将我的初稿修订成格式工整的书稿，还对篇章的布局提出了很好的思路。

感谢亲朋好友的无私帮助和大力支持。

感谢花费时间和精力阅读本书的读者朋友，你们的批评和指正是我进步的动力。

杨云君

2013 年 1 月于北京

第一部分

基 础 篇

本部分内容

- 第 1 章　Android 体系结构及源代码阅读环境搭建
- 第 2 章　框架基础 JNI

第 1 章　Android 体系结构及源代码阅读环境搭建

Android 是 Google 于 2007 年 11 月 5 日发布的基于 Linux 内核的开源移动设备软件平台，该平台由操作系统、虚拟机、运行库、框架、应用软件以及开发工具组成。

1.1　Android 发展过程

Android 自 Android 1.5 开始，以甜点作为平台版本代号，并且各代号首字母以 C D E F G H I J 顺序排列。Android 自发布以来，经过数个版本的更新和完善，已成为最大的智能手机平台之一。Android 的发展过程如表 1-1 所示。

表 1-1　Android 发展过程情况

版本代号	平台版本	API 级别	发布时间	适用设备
——	1.0	1	2008 年 9 月	智能手机
——	1.1	2	2009 年 2 月	智能手机
Cupcake（纸杯蛋糕）	1.5	3（NDK 1）	2009 年 4 月	智能手机
Donut（甜甜圈）	1.6	4（NDK 2）	2009 年 9 月	智能手机
Eclair（松饼）	2.0	5	2009 年 10 月	智能手机
Eclair（松饼）	2.0.1	6	2009 年 12 月	智能手机
Eclair（松饼）	2.1	7（NDK 3）	2010 年 1 月	智能手机
Froyo（冻酸奶）	2.2.x	8（NDK 4）	2010 年 5 月	智能手机
Gingerbread（姜饼）	2.3-2.3.2	9（NDK 5）	2010 年 12 月	智能手机
Gingerbread（姜饼）	2.3.3- 2.3.7	10	2011 年 2 月	智能手机
Honeycomb（蜂巢）	3.0	11	2011 年 2 月	平板电脑
Honeycomb（蜂巢）	3.1	12（NDK 6）	2011 年 5 月	平板电脑
Honeycomb（蜂巢）	3.2.x	13	2011 年 6 月	平板电脑
Ice Cream Sandwich（冰激凌三明治）	4.0.1- 4.0.2	14（NDK 7）	2011 年 10 月	智能手机 / 平板电脑
Ice Cream Sandwich（冰激凌三明治）	4.0.3	15	2011 年 12 月	智能手机 / 平板电脑
Jelly Bean（果冻豆）	4.1.x	16（NDK 8）	2012 年 7 月	智能手机 / 平板电脑
Jelly Bean（果冻豆）	4.2	——	2012 年 10 月	智能手机 / 平板电脑

截至2012年年底，Android最新的版本代号是Jelly Bean，其对应的平台版本有4.1.x和4.2。Google官网称Android 4.2为A new flavor of Jelly Bean，即一种新口味的果冻豆。因此可以将4.2版视为Jelly Bean的一个升级版本，与4.1相比变化不大。

注意 Android官网是每个Android学习者的宝库，读者可以直接从官网了解Android的最新进展。

Android官网主页地址：www.android.com

Android源码主页地址：http://source.android.com

Android开发者主页地址：http://developer.android.com

了解了Android的发展过程，接下来进一步学习Android的体系结构。

1.2 Android体系结构

要深入学习Android，必须理解Android的体系结构，本节将分别从静态和动态两种视角对Android的体系结构进行介绍。

1.2.1 静态视角的体系结构

Android采用了分层的体系结构，各层的结构和功能非常清晰。从上往下看，Android分为四层，分别是：1）应用层，2）应用框架层，3）Android运行环境和系统运行库层，4）Linux内核层，如图1-1所示。

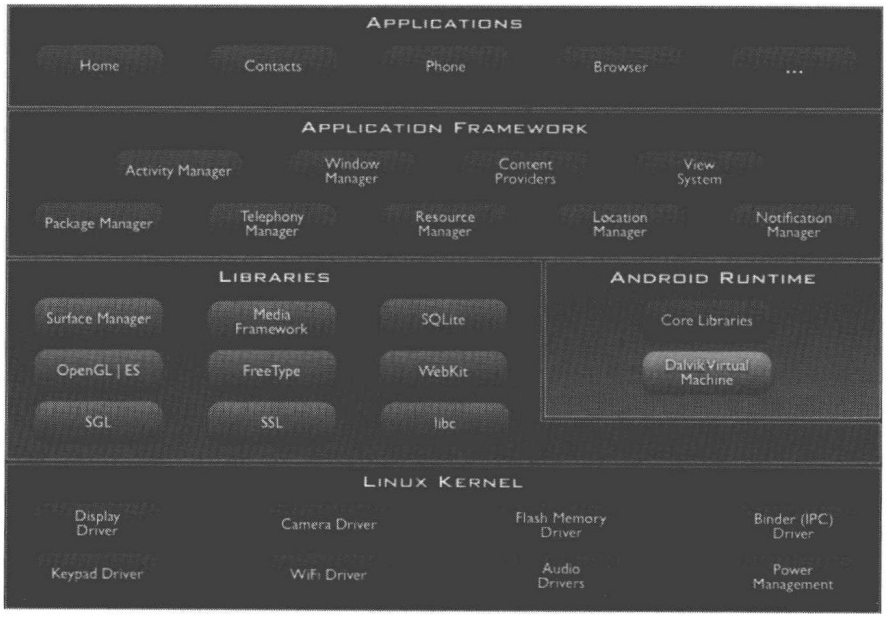


图1-1 静态视角的Android体系结构

1. 应用层

应用层位于 Android 体系结构的最上层。Google 在 Android 中内置了一些核心应用程序，如图 1-1 所示，有主屏幕（Home）、联系人（Contacts）、电话（Phone）、浏览器（Browser），另外还有日历（Calendar）、地图、SMS 短消息程序、图库（Gallery）、输入法、闹钟（Alarm）等。开发者还可以使用应用框架层提供的 API 编写自己的应用程序，这也是 Android 开源的巨大优势之一。

应用程序主要用 Java 语言编写，自 Android 1.5 开始，Google 提供了 NDK 开发工具，可以方便地开发基于 JNI 的应用程序。NDK 便于开发者开发需要基于 C/C++ 才能实现的功能，也可以提高程序执行效率。

2. 应用框架层

应用框架层是 Android 体系结构的第二层，它不仅为应用层提供 API，而且是一种重要的机制。这种机制为应用层提供了可以复用的组件，提供了应用层开发的规范，屏蔽了应用层与底层交互的复杂性。应用框架层提供的 API 并不完全对第三方应用程序开放，有一部分 API 是隐藏的。开发第三方应用程序需要依赖 Android SDK 提供的 API，它只是应用框架层 API 的一个子集。

本层主要是使用 Java 和 JNI 实现的，位于该层的主要组件如表 1-2 所示。

表 1-2 应用框架层主要组件

组件名称		功能描述
View System	视图系统	提供了可扩展的用于构建应用程序 UI 的控件，包括 List、Grid、TextBox、Button 以及可嵌入的 Web 浏览器等
Content Providers	内容提供器	提供应用程序之间数据共享的接口
Resource Manager	资源管理器	提供非代码资源的访问的接口，如字符串、图形资源和布局文件（Layout File）等
Notification Manager	通知管理器	在状态栏中显示自定义的提示信息
Activity Manager	活动管理器	用来管理应用程序生命周期并提供通用的导航回退功能
Window Manager	窗口管理器	管理所有的窗口程序
Package Manager	包管理器	提供 Android 程序安装包管理的接口

3. Android 运行环境和系统运行库层

Android 运行环境和系统运行库层位于 Android 体系结构的第三层。本层相当于中间件层，为应用框架层提供服务，它分成两个部分：一部分是系统运行库，包含各种系统库和第三方库；另一部分是 Android 运行环境，这里主要是使用 C++ 和 C 实现的。

应用框架层为应用层提供的功能，在底层大多是由系统运行库实现的。Android 应用层使用的多媒体、浏览器、数据库、图形引擎等，其功能实现均位于该层。

Android 应用层的 Java 程序运行在虚拟机中。Android 提供了 Dalvik 虚拟机以支持 Java 运行环境。

系统运行库的主要组件如表 1-3 所示。

表 1-3 系统运行库组件

组件名称	功能描述
libc（bionic）	此处的 libc 库并不是 GNU 的标准 libc 库，而是 Google 开发的基于 BSD 许可的 bionic 库，在体积、运行效率、线程实现等方面都做了优化
WebKit	开源浏览器引擎
Media Framework	多媒体框架，Android 2.3 之前采用 PacketVideo 的 OpenCore，自 2.3 版开始引入 Stagefright
SQLite	嵌入式数据库引擎
OpenGL ES	3D 图形渲染库，支持 OpengGL ES 1.x 和 2.0 规范
FreeType	位图和矢量字体渲染引擎
SurfaceManager	渲染管理器，显示系统的核心组成部分，负责绘图的合成管理
SGL	基于 Skia 的 2D 图形渲染引擎
SSL	网络通信安全协议

Android 运行环境的主要组件如表 1-4 所示。

表 1-4 Android 运行环境组件

组件名称	功能描述
Core Libraries	是对标准 Java 核心库的裁剪，保留了标准 Java API 的多数功能，并加入了 Android 特有的 API，比如 android.os 包下的 API
Dalvik Virtual Machine（DVM）	主要完成应用程序进程管理、堆栈管理、对象生命周期管理、安全和异常检查、垃圾回收等功能 Google 并没有采用标准 JVM 虚拟机，而是采用针对嵌入式设备内存、供电、处理速度有限的情况开发的 Dalvik 虚拟机。该虚拟机主要优点如下： • 基于寄存器。可以针对特定的硬件最大化优化代码 • 运行的是经过优化和压缩的 DEX 字节码。DEX 字节码可以比 JAR 字节码节省几乎一半的空间，提高 I/O 读取速度。而且可以进一步优化为 ODEX 字节码 • 进程安全和隔离。允许在有限的内存中同时运行多个虚拟机实例，每一个 Android 应用都运行在它自己的 DVM 实例中，都有自己独立的进程空间。DVM 依赖 Linux 的线程调度和管理机制，通过 fork 函数为每个应用程序创建一个 Linux 进程。所有的 Android 应用在底层都对应于自己的 Linux 进程。不同的应用在不同的进程空间里运行，独立的进程空间可以防止在 DVM 崩溃时所有程序都被“杀死”

4. Linux 内核层

Android 自 ICS 开始基于 Linux 3.0 内核，充分利用了 Linux 内核基于权限的安全模型、内存管理、进程管理、网络协议栈和驱动模型等优点，并在 Low Memory Killer、进程间通