

探索和品味Android大师们的内核设计艺术



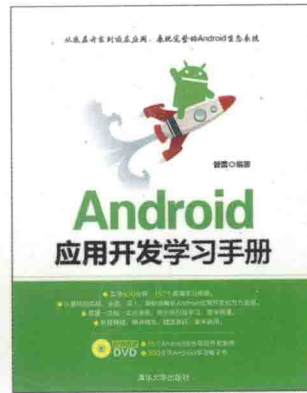
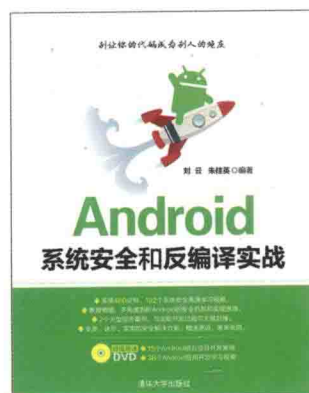
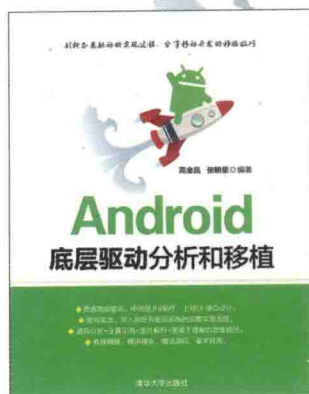
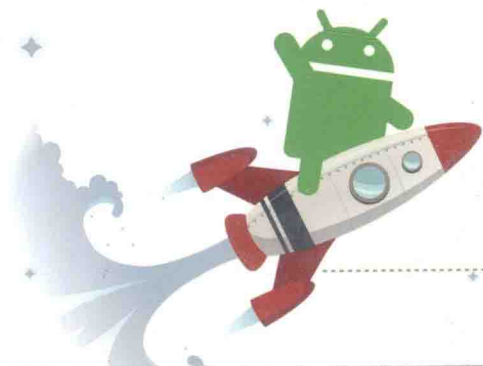
张元亮◎编著

深入理解 Android系统

- 全面剖析进程/线程、内存管理、Binder机制、显示系统、多媒体管理、输入系统等核心知识在Android中的实现原理。
- 源码分析+全真示例+图片解析=更易于理解的思维路径。
- 由浅入深，由总体框架到细节实现，快速获取对Android系统的二次开发能力。
- 教授精髓，精讲精练。赠送源码，拿来就用。

清华大学出版社





深入理解 Android 系统

张元亮 编著

清华大学出版社

北 京

内 容 简 介

本书内容共 18 章,循序渐进地分析了整个 Android 系统的基本架构知识,从获取源码开始讲起,依次讲解了 Android 系统介绍,包括获取并编译 Android 源码,分析 JNI,内存系统架构详解,硬件抽象层架构详解,Binder 通信机制详解,init 启动进程详解,Zygote 进程详解,System 进程详解,应用程序进程详解,ART 机制架构详解,Sensor 传感器系统架构详解,蓝牙系统架构详解,Android 多媒体框架架构详解,音频系统框架架构详解,视频系统架构详解,WebKit 系统架构详解,Android 5.0 中的 WebView, Wi-Fi 系统架构详解等内容。本书几乎涵盖了所有 Android 系统架构的主要核心内容,讲解方法通俗易懂并且详细,不但适合应用高手们学习,也特别便于初学者学习和理解。

本书适合 Android 源码分析人员、Android 系统架构师、Linux 开发人员、Android 物联网开发人员、Android 爱好者、Android 底层开发人员、Android 驱动开发人员、Android 应用开发人员、Android 传感器开发人员、Android 智能家居开发人员、Android 可穿戴设备开发人员学习,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。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

深入理解 Android 系统/张元亮编著. —北京:清华大学出版社,2015
ISBN 978-7-302-40439-2

I. ①深… II. ①张… 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 122705 号

责任编辑:朱英彪
封面设计:刘超
版式设计:魏远
责任校对:王云
责任印制:宋林

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 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:203mm×260mm 印 张:44.5 字 数:1248 千字

版 次:2015 年 7 月第 1 版 印 次:2015 年 7 月第 1 次印刷

印 数:1~3000

定 价:88.00 元

前 言

2007年11月5日，谷歌公司宣布基于Linux平台的开源手机操作系统Android诞生，该平台号称是首个为移动终端打造的真正开放和完整的移动软件平台。本书将和广大读者一起深入理解Android系统的架构知识，共同领略这款神奇系统的奥妙之处。

市场占有率高居第一

截至2014年9月，Android在智能手机市场上的占有率从2013年第一季度的68.8%上升到85%。而iOS则从2013年的19.4%下降到15.5%，WP系统从原来的2.7%小幅上升到3.6%。从数据上看，Android平台占据了市场的主导地位。

由数据可知，iOS有所下降，WP市场小幅增长，但Android市场的占有率增加幅度较大。就目前来看，智能手机的市场已经饱和，大多数用户都在各个平台间转换。而就在这样一个市场上，Android还增长了10%左右的占有率确实不易。

为开发人员提供了发展的平台

(1) 保证开发人员可以迅速向Android应用开发转型

Android应用程序是使用Java语言开发的，只要具备Java开发基础，就能很快入门并掌握。作为单独的Android应用开发，对Java编程门槛的要求并不高，即使没有编程经验，也可以在突击学习Java之后学习Android。另外，Android完全支持2D、3D和数据库，并且和浏览器实现了集成。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、管理软件、互联网应用和游戏等。

(2) 定期举办奖金丰厚的Android开发大赛

为了吸引更多的用户使用Android开发，已经成功举办了奖金为数千万美元的开发竞赛。鼓励开发人员创建出创意十足且实用的软件。对于开发人员来说，这种大赛不但能提升自己的开发水平，并且高额的奖金也是学员们学习的动力。

(3) 开发人员可以利用自己的作品赚钱

为了能让Android平台引起更多的关注，谷歌提供了一个专门下载Android应用的门店Android Market，地址是<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。只要所开发的软件程序足够吸引人，就可以获得很可观的金钱回报。这样实现了程序员学习和赚钱的两不误，所以吸引了更多开发人员加入到Android开发大军中来。

本书的内容

本书内容共 18 章，循序渐进地分析了整个 Android 系统的基本架构知识。本书从获取源码开始讲起，依次讲解了 Android 系统介绍，包括获取并编译 Android 源码，分析 JNI，内存系统架构详解，硬件抽象层架构详解，Binder 通信机制详解，init 启动进程详解，Zygote 进程详解，System 进程详解，应用程序进程详解，Sensor 传感器系统架构详解，蓝牙系统架构详解，Android 多媒体框架架构详解，音频系统框架架构详解，视频系统架构详解，WebKit 系统架构详解，Android 5.0 中的 WebView，Wi-Fi 系统架构详解，ART 机制架构详解等内容。本书几乎涵盖了所有 Android 系统架构的主要核心内容，讲解方法通俗易懂并且详细，不但适合应用高手们学习，也特别便于初学者学习和理解。

本书的版本

Android 系统自 2008 年 9 月发布第一个版本 1.1 以来，截至 2014 年 10 月发布最新版本 5.0，一共存在十多个版本。由此可见，Android 系统升级频率较快，一年之中最少有两个新版本诞生。如果过于追求新版本，会造成力不从心的结果。所以在此建议广大读者不必追求最新的版本，只需关注最流行的版本即可。据官方统计，截至 2014 年 11 月 10 日，占据前 3 位的版本分别是 Android 4.4、Android 4.3 和 Android 5.0，其实这 3 个版本的差别并不是很大，只是在某些领域的细节上进行了更新。为了在市场普及率和新版本之间做好兼顾，本书将以最新的 Android 5.0 作为讲解主线，并且结合了 Android 4.4 的架构知识。

本书特色

本书内容十分丰富，讲解细致。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）内容全面，讲解细致

本书几乎涵盖了 Android 系统架构所需要的所有主要知识点，详细讲解了每一个 Android 系统的具体实现过程。每一个知识点都力求用详实易懂的语言展现在读者面前。

（2）遵循合理的主线进行讲解

为了使广大读者彻底弄清楚 Android 系统架构的各个知识点，在讲解每一个知识点时，从 Linux 内核开始讲起，依次剖析了底层架构、API 硬件抽象层和顶层应用的具体知识。遵循了从底层到顶层的顺序，实现了 Android 系统架构大揭秘的目标。

（3）章节独立，自由阅读

本书中的每一章内容都可以独自成书，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针对性的学习。并且和传统古板的计算机书籍相比，阅读本书会带来很大的快乐。

（4）版本新颖，代表性强

本书以最新的 Android 5.0 作为讲解主线，结合 Android 4.4 的架构知识进行讲解，这样可以涵盖大多数读者群体，代表性更强。

本书的读者对象

- ☑ Android 源码分析人员。
- ☑ Android 系统架构师。
- ☑ Linux 开发人员。
- ☑ Android 物联网开发人员。
- ☑ Android 爱好者。
- ☑ Android 底层开发人员。
- ☑ Android 驱动开发人员。
- ☑ Android 应用开发人员。
- ☑ Android 传感器开发人员。
- ☑ Android 智能家居开发人员。
- ☑ Android 可穿戴设备开发人员。
- ☑ 相关培训学校的学生。
- ☑ 相关大专院校的学生。

售后服务

参与本书编写的人员还有周秀、付松柏、邓才兵、钟世礼、谭贞军、张加春、王教明、万春潮、郭慧玲、侯恩静、程娟、王文忠、陈强、何子夜、李天祥、周锐、朱桂英、张元亮、张韶青、秦丹枫。本书在编写过程中，得到了清华大学出版社工作人员的大力支持，正是各位编辑的求实、耐心和效率，才使得本书在这么短的时间内出版。另外也十分感谢我的家人在我写作时给予的巨大支持。

由于编者水平有限，如有纰漏和不尽如人意之处，诚请读者提出意见或建议，以便修订并使之更臻完善。另外我们提供了售后支持网站（<http://www.chubanbook.com/>）和 QQ 群（192153124），读者朋友如有疑问可以在此提出，一定会得到满意的答复。

编 者

目 录

第 1 章 获取并编译 Android 源码.....	1	2.5 JNIEnv 接口	61
1.1 获取 Android 源码	1	2.6 开发 JNI 程序	62
1.1.1 在 Linux 系统获取 Android 源码	1	2.6.1 开发 JNI 程序的步骤	62
1.1.2 在 Windows 平台获取 Android 源码	3	2.6.2 开发一个自己的 JNI 程序.....	63
1.2 分析 Android 源码结构	6	第 3 章 内存系统架构详解.....	66
1.2.1 总体结构.....	6	3.1 分析 Android 的进程通信机制	66
1.2.2 应用程序部分	7	3.1.1 IPC 机制介绍.....	66
1.2.3 应用程序框架部分	9	3.1.2 Service Manager 是 Binder 机制的上下文管 理者	67
1.2.4 系统服务部分	10	3.1.3 Service Manager 服务	84
1.2.5 系统程序库部分	12	3.2 分析匿名共享内存子系统	87
1.2.6 系统运行库部分	15	3.2.1 Ashmem 系统基础.....	87
1.2.7 硬件抽象层部分	16	3.2.2 基础数据结构	88
1.3 分析源码中提供的接口	17	3.2.3 初始化处理	89
1.3.1 暴露接口和隐藏接口	17	3.2.4 打开匿名共享内存设备文件	90
1.3.2 调用隐藏接口	23	3.2.5 实现内存映射	93
1.4 编译源码	26	3.2.6 实现读/写操作	94
1.4.1 搭建编译环境.....	26	3.2.7 实现锁定和解锁	96
1.4.2 在模拟器中运行	29	3.2.8 回收内存块	102
1.5 编译源码生成 SDK	30	3.3 分析 C++ 访问接口层	103
第 2 章 分析 JNI	35	3.3.1 接口 MemoryHeapBase	103
2.1 JNI 基础	35	3.3.2 接口 MemoryBase.....	112
2.1.1 JNI 的功能结构	35	3.4 分析 Java 访问接口层	115
2.1.2 JNI 的调用层次	36	第 4 章 硬件抽象层架构详解	120
2.1.3 分析 JNI 的本质	36	4.1 HAL 基础	120
2.2 分析 MediaScanner	38	4.1.1 推出 HAL 的背景	120
2.2.1 分析 Java 层.....	38	4.1.2 HAL 的基本结构	121
2.2.2 分析 JNI 层	45	4.2 分析 HAL module 架构	123
2.2.3 分析 Native (本地) 层	46	4.2.1 hw_module_t	124
2.3 分析 Camera 系统的 JNI	54	4.2.2 hw_module_methods_t	124
2.3.1 Java 层预览接口	54	4.2.3 hw_device_t	125
2.3.2 注册预览的 JNI 函数	56	4.3 分析文件 hardware.c	126
2.3.3 C/C++ 层的预览函数	59		
2.4 Java 与 JNI 基本数据类型转换	60		

4.3.1	寻找动态链接库的地址.....	126	5.1.17	释放物理页面.....	173
4.3.2	数组 <code>variant_keys</code>	126	5.1.18	分配内核缓冲区.....	174
4.3.3	载入相应的库.....	127	5.1.19	释放内核缓冲区.....	176
4.3.4	获得 <code>hw_module_t</code> 结构体.....	127	5.1.20	查询内核缓冲区.....	179
4.4	分析硬件抽象层的加载过程.....	128	5.2	Binder 封装库.....	179
4.5	分析硬件访问服务.....	132	5.2.1	Binder 的 3 层结构.....	180
4.5.1	定义硬件访问服务接口.....	132	5.2.2	类 <code>BBinder</code>	181
4.5.2	具体实现.....	133	5.2.3	类 <code>BpRefBase</code>	183
4.6	分析官方实例.....	134	5.2.4	类 <code>IPCThreadState</code>	185
4.6.1	获取实例工程源码.....	135	5.3	初始化 Java 层 Binder 框架.....	188
4.6.2	直接调用 <code>service()</code> 方法的实现代码.....	136	5.3.1	搭建交互关系.....	188
4.6.3	通过 <code>Manager</code> 调用 <code>service</code> 的实现代码.....	141	5.3.2	实现 <code>Binder</code> 类的初始化.....	188
4.7	HAL 和系统移植.....	144	5.3.3	实现 <code>BinderProxy</code> 类的初始化.....	190
4.7.1	移植各个 Android 部件的方式.....	144	5.4	实体对象 <code>binder_node</code>	190
4.7.2	设置设备权限.....	144	5.4.1	定义实体对象.....	191
4.7.3	<code>init.rc</code> 初始化.....	148	5.4.2	增加引用计数.....	192
4.7.4	文件系统的属性.....	148	5.4.3	减少引用计数.....	193
4.8	开发自己的 HAL.....	150	5.5	本地对象 <code>BBinder</code>	194
4.8.1	封装 HAL 接口.....	150	5.5.1	引用了运行的本地对象.....	195
4.8.2	开始编译.....	153	5.5.2	处理接口协议.....	201
第 5 章	Binder 通信机制详解.....	155	5.6	引用对象 <code>binder_ref</code>	205
5.1	分析 Binder 驱动程序.....	155	5.7	代理对象 <code>BpBinder</code>	208
5.1.1	数据结构 <code>binder_work</code>	155	5.7.1	创建 Binder 代理对象.....	208
5.1.2	结构体 <code>binder_node</code>	156	5.7.2	销毁 Binder 代理对象.....	209
5.1.3	结构体 <code>binder_ref</code>	157	第 6 章	init 启动进程详解.....	213
5.1.4	通知结构体 <code>binder_ref_death</code>	158	6.1	什么是 <code>init</code> 进程.....	213
5.1.5	结构体 <code>binder_buffer</code>	158	6.2	入口函数.....	214
5.1.6	结构体 <code>binder_proc</code>	159	6.3	<code>init</code> 配置文件.....	217
5.1.7	结构体 <code>binder_thread</code>	160	6.3.1	<code>init.rc</code> 基础.....	217
5.1.8	结构体 <code>binder_transaction</code>	161	6.3.2	<code>init.rc</code> 解析.....	219
5.1.9	结构体 <code>binder_write_read</code>	162	6.4	解析 <code>Service</code>	223
5.1.10	<code>BinderDriverCommandProtocol</code>	162	6.4.1	<code>Zygote</code> 对应的 <code>service action</code>	224
5.1.11	枚举 <code>BinderDriverReturnProtocol</code>	163	6.4.2	<code>init</code> 组织 <code>Service</code>	224
5.1.12	结构体 <code>binder_ptr_cookie</code> 和 <code>binder_transaction_data</code>	164	6.4.3	解析 <code>Service</code> 用到的函数.....	226
5.1.13	结构体 <code>flat_binder_object</code>	164	6.5	解析 <code>on</code>	230
5.1.14	设备初始化.....	165	6.5.1	<code>Zygote</code> 对应的 <code>on action</code>	230
5.1.15	打开 Binder 设备文件.....	167	6.5.2	结构体 <code>action</code>	232
5.1.16	实现内存映射.....	168	6.5.3	解析 <code>on</code> 字段所在的 <code>option</code>	232
			6.6	<code>init</code> 控制 <code>Service</code>	233

6.6.1 启动 Zygote	233	8.7 分析实现性能统计	292
6.6.2 启动 Service	234	8.7.1 构造函数	292
6.6.3 总结 4 种启动 Service 的方式	238	8.7.2 进行性能统计	293
6.7 启动属性服务	243	8.7.3 输出统计文件	295
6.7.1 引入属性	243	8.8 剪贴板服务	302
6.7.2 设置内核变量	245	8.8.1 复制数据到剪贴板	302
6.7.3 初始化属性服务	246	8.8.2 从剪贴板粘贴数据	304
6.7.4 实现具体启动工作	247	8.8.3 管理 CBS 中的权限	306
6.7.5 获取属性值	249	第 9 章 应用程序进程详解	309
6.7.6 处理请求	251	9.1 创建应用程序	309
第 7 章 Zygote 进程详解	253	9.1.1 发送创建请求	309
7.1 Zygote 基础	253	9.1.2 保存启动参数	312
7.2 启动 Zygote	254	9.1.3 创建指定的应用程序	314
7.2.1 init.c 启动脚本	254	9.1.4 创建本地对象 LocalSocket	315
7.2.2 创建一个 Socket	258	9.1.5 接收创建新应用程序的请求	316
7.2.3 入口函数 main()	260	9.2 启动线程池	320
7.2.4 启动函数创建一个虚拟机实例	262	9.3 创建信息循环	322
7.2.5 和 Zygote 进程中的 Socket 实现连接	264	第 10 章 ART 机制架构详解	324
第 8 章 System 进程详解	271	10.1 分析 ART 的启动过程	324
8.1 启动前的准备	271	10.1.1 运行 app_process 进程	325
8.1.1 获取创建的 Socket	271	10.1.2 准备启动	329
8.1.2 启动 System 进程	272	10.1.3 创建运行实例	336
8.2 分析 SystemServer	272	10.1.4 注册本地 JNI 函数	338
8.2.1 分析主函数 main()	272	10.1.5 启动守护进程	339
8.2.2 分析函数 init2()	275	10.1.6 解析参数	340
8.3 第一个启动的 ServiceEntropyService	275	10.1.7 初始化类、方法和域	350
8.3.1 将内容写到 urandom 设备	276	10.2 进入 main() 主函数	357
8.3.2 将和设备相关的信息写到 urandom 设备	277	10.3 查找目标类	358
8.3.3 读取 urandom 设备的内容	277	10.3.1 函数 LookupClass()	359
8.3.4 发送 ENTROPY_WHAT	278	10.3.2 函数 DefineClass()	361
8.4 生成并管理日志文件	278	10.3.3 函数 InsertClass()	365
8.4.1 分析 DBMS 构造函数	278	10.3.4 函数 LinkClass()	366
8.4.2 添加 dropbox 日志文件	280	10.4 类操作	368
8.4.3 DBMS 和 settings 数据库	284	10.5 实现托管操作	370
8.5 分析 DiskStatsService	285	第 11 章 Sensor 传感器系统架构详解	376
8.6 监测系统内部存储空间的状态	289	11.1 Android 传感器系统概述	376
8.6.1 构造函数	289	11.2 Java 层详解	377
8.6.2 内存检查	290		

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《深入理解ANDROID系统》张元亮 著.pdf

请登录 <https://shgis.com/post/3093.html> 下载完整文档。

手机端请扫码查看：

