

Android

核心原理与系统级应用高效开发

■ 韩超 等著



本书为Android系统级别的开发者系统化而有条理性地展示了庞大的Android系统的软件结构，点明开发思路。



Android应用程序的开发者可以通过本书加深认识，拓展开发手段；从事Android硬件移植的开发者可以通过本书加深对系统全局的了解。



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Android

核心原理与系统级应用高效开发

■ 韩超 等著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本介绍 Android 核心开发的技术图书,主要目标是让 Android 的开发者更深入地理解 Android 系统,让开发的过程更正确、更有效率。本书主要由两大部分组成,第 1 篇自下而上介绍 Android 核心框架层的结构,并将开发的技巧和要点蕴含于其中;第 2 篇偏重系统级的应用层,关注 Android 开发过程中的重点方面。

本书为 Android 系统级别的开发者系统化而有条理性地展示了庞大的 Android 系统的软件结构,点明开发思路;Android 应用程序的开发者可以通过本书加深认识,拓展开发手段;从事 Android 硬件移植的开发者可以通过本书加深对系统全局的了解。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Android 核心原理与系统级应用高效开发/韩超等著.—北京:电子工业出版社,2012.7

ISBN 978-7-121-17361-5

I. ①A… II. ①韩… 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(2012)第 125170 号

策划编辑:李 冰

责任编辑:刘 舫

特约编辑:赵树刚

印 刷:北京东光印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:28 字数:717 千字

印 次:2012 年 7 月第 1 次印刷

印 数:4000 册 定价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

技术经验共享

经过数年的发展，Android 系统取得了巨大的进展。一方面，Android 设备层出不穷，迭代更新；另一方面，Android 第三方应用也成为不同领域开发者的开发重点。从软件系统的角度考虑，Android 系统已经逐渐趋于成熟和稳定。Android 2.3.x 版本的开源软件广泛应用于手机、电视等各种设备；Android 4.x 作为开源的新版本，整合了手机和平板电脑两方面的功能。

由于软件系统的架构已经成熟，Android 开源版本的升级已经不再对框架层做出较大的更改，而是横向增加外围的功能。从软件工程师的角度，深入 Android 系统开发已经到了比较适当的阶段。

但是，目前从事 Android 各个方面开发的工程师大都很难达到深入了解系统级别的目的，在开发乃至学习中存在着较大的盲区和误区。

Android 开发者主要的问题集中在以下几个方面。

(1) 领域的交叠造成混乱。由于 Android 的开发者来自不同领域，很可能不了解其他方面的技术背景。庞大的 Android 软件系统也涉及 Linux 开发、Java 开发，以及 Android 系统本身各个方面的技术，因此只熟悉一个方面的技术很难深入 Android 系统的开发。

(2) 开发者低估 Android 系统的复杂性。Android 系统的结构、功能已经和桌面电脑相当，在某些方面甚至需要考虑更多内容。在 Android 系统实现一个模块要涉及软件系统的各个层次，很多时候已经超出了开发者的认知范畴。

(3) Android 应用开发者的局限。很多 Android 应用程序的开发者仅仅使用 Windows 中的 SDK 开发环境。在实际操作中，此种环境并不足以开发 Android 应用程序，甚至只开发第三方应用程序也不能只使用此环境。这是由 Android 系统的内部结构决定的。

(4) 系统级的软件结构设计不成熟。Android 设备的开发者为了实现某种功能、某个模块，有时会使用权宜（Workaround）的方式。虽然暂时解决了问题，但是效率较低，事倍功半，还可能存在兼容性问题和安全漏洞。

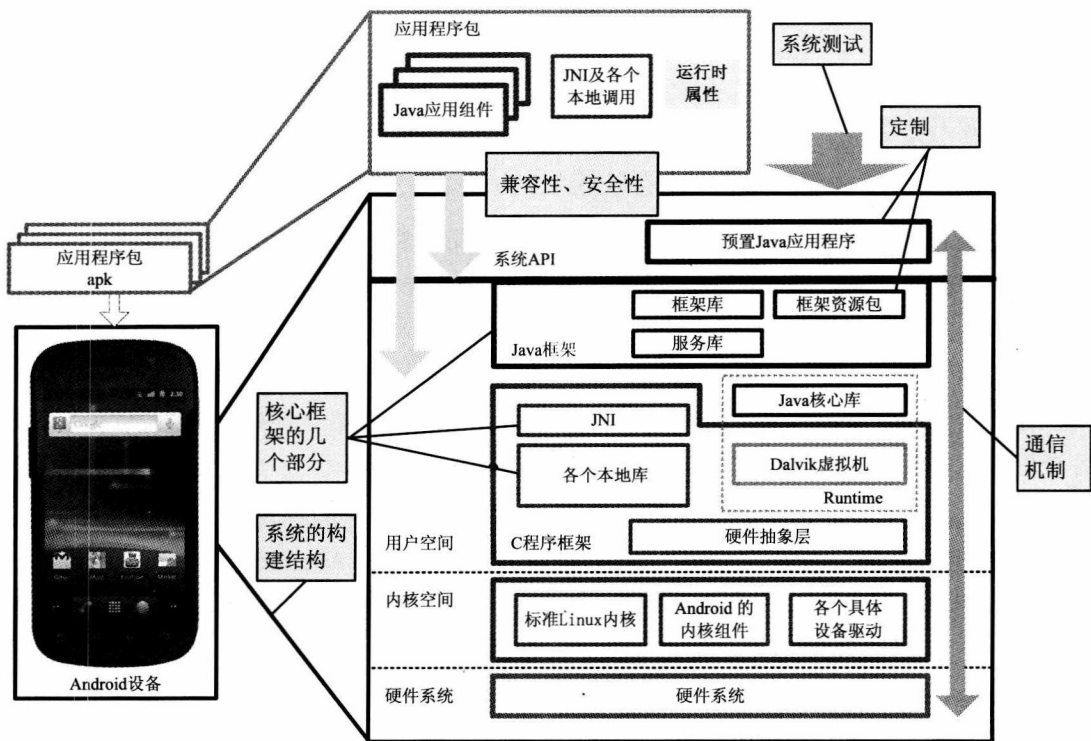
(5) 错误的理念，互相误导。Android 开发门槛较低，实际的难度却较深。由于目前网络交流比较方便，很多偏颇或者肤浅的开发经验得以传播。

在理想的状态下，Android 的高级开发者应该具有 Linux 开发经验、熟悉 C/C++ 和 Java 语言、具有 GUI 系统的经验，具有大型软件设计经验。这不意味着开发者必须达到这个程度才能开发 Android，但是开发者的确应当首先知道自己能做什么，不能做什么。不可一味拔高，做完全超出自己知识和技术能力外的开发工作。

关于本书的内容

本书的编写团队主要由各种岗位的开发人员组成，开发过程中的宝贵经验是本书主要的内容来源。本书力求为广大开发者提供一些宝贵的经验，让工程师的灵感得以传播，让各个不同领域的开发者都能够以正确、高效的方式进行学习和开发。Android 是一个实际的系统，能够作为传播的内容应该来自实际开发过程中归纳的经验，而不应来自对代码的阅读，更不应来自道听途说。这也是本书的核心理念。

本书主要由两篇组成，分别侧重于 Android 开发的不同方面。本书中 Android 系统的技术关系如下图所示。



第 1 篇称之为“Android 核心框架层开发”。第 1 章介绍框架的基础，第 2 章介绍编译系统，第 3 章至第 5 章介绍本地框架层，第 6 章是 Java 运行环境，第 7 章至第 9 章介绍 Java 框架层内容，第 10 章介绍 GUI 系统和应用程序管理。本篇自下而上介绍 Android 核心的框架层次结构，并将开发的技巧和要点蕴含于其中。

第 2 篇称之为“Android 系统级应用开发要点”，偏重系统级的应用层，关注 Android 开发过程中的重点方面。用 5 章篇幅关注了系统的兼容性、安全性、预置应用层、定制、调试和测试几个方面。

本书组织结构和特点

本书介绍 Android 系统中的通用内容，不依赖特定的硬件，使用仿真器的环境基本可以学习大部分的内容。

Android 中具有多个子系统和模块，其中某些部分的实现非常复杂。对于这些部件，本书主要给出它们的结构框图和代码路径。至于它们内部的实现，不作为本书的重点。为了体现实用性，本书的重点是对于开发有所帮助的部分。

第 1 篇内容，重点在于 Android 软件系统的中间两个层次。Android 是一个基于 Linux 的系统，其自下而上的软件结构是本书第 1 篇的主线。本书采用了自本地层到 Java 层，最后介绍 GUI 系统的方式，完整展现 Android 框架层的结构。

在主线之外，本书第 1 篇还有三条次线：第一条是 Android 系统较为特殊的通信机制，包括属性、套接字和 Binder；第二条是系统的运行流程；第三条是各个部分调试方法。除此之外，开发技巧和要点则位于每一章当中。本书第 1 篇尽量将分散的知识和经验总结成一个串行的结构，可以保证自前而后的顺序。当然也有部分前面的内容需要后面知识的情况，读者可以对照阅读。

第 2 篇内容，重点在于 Android 系统最上面的一个层次，本部分内容的复杂度不是很高。本篇在组织上的一个要点是应用程序包和框架层之间的关系。5 个章节相互独立，也有所联系。安全机制、NDK 特殊程序结构、可定制部分、调试和测试手段等开发的实用方面位于各章当中。

在 Android 系统的相关开发中，有两种不同方式：一种是快速的、表面化的开发，不追究系统原理，靠形式上的仿做来实现功能；另一种是在深厚的技术底蕴中，精研 Android 系统，以清晰思路进行最优化的开发。在本书作者看来，两种方式并无优劣之分，为了实现开发成果，各种开发方式都可使用。本书的内容，既有对 Android 系统的精研，也有对快速开发的支持。

对于两种方式的区别，本书关注几个要点：第一，开发者可以在不深入研究系统的情况下，进行快速的表面开发，但要注意不能陷入开发的误区，本书的编译板级支持、NDK 开发、预置应用、系统定制等几个部分是这种类型；第二，有些核心模块开发，必须在深入研究系统的情况下才能完成，不能以表层开发的方式开发核心模块，本书的本地框架、Java 框架、Java 本地调用几部分是这种类型；第三，精研系统的目的也是为了开发，应该避免“一叶障目，不见森林”，对片面和局部进行极端的深入研究往往效率低、难度大，因此本书核心部分的研究通常先给出整体框图和各个部分的代码路径。

由于 Android 系统开源已经升级了若干个版本，以后还将进行升级。在开发过程中的经验有些只适用某些版本，而不能适应所有版本。这也为 Android 系统的开发过程造成了一些困难。为了达到普遍的适应性，本书在编写上考虑了版本升级的问题。首先，本书对

照了从 Android 的历史版本和最新发布之前，大多数的内容在各个版本的 Android 系统中都具有，并且基本相同的，稳定的部分显然也是精华的内容；其次，对于 Android 2.3.x 之前版本不同的内容，本书会标注点明；最后，Android 4.x 对核心框架改动的内容并不多，本书单独构建一章予以介绍。

本书秉承实用理念，措辞精练，每一章、每一小节都为读者提供知识和技术的要点。目的是在有限的篇幅内，为读者展示更多的内容。本书在排版上吸收了各方面的经验，采取了相当紧凑的方式。书中列出的结构图以清晰的方式表示了各部分的结构，列出的代码选取了最精华的片断，并且增加了作者的注释。

本书读者

本书适合于一切在做或者准备做 Android 开发的技术人员。本书的内容涉及 Android 系统的内部核心实现。本书作者没有能力帮助读者快速获得十年的开发经验，但是可以为读者指明开发的方向，并加强读者对技术的认识，更上一层楼。

Linux 用户空间中的 C 语言开发和 Java 语言开发是本书的技术基础，也是 Android 系统技术最重要的两个基础。建议每个领域的读者都应增强这两个方面的知识。

Android 系统级的开发需要 Linux 主机环境，通过阅读本书学习 Android 系统的开发，最好具有此环境。在 Windows 中使用 Source Insight 等工具阅读 Android 源代码，并运行 SDK 中的仿真器是一种退而求其次的方式，这种方式也可以学习到很多知识。作者依然强烈建议读者阅读本书的时候可以对照 Android 源代码，并可以在 Linux 环境通过编译全部源代码生成仿真器。

不同的知识结构、从事不同领域的读者在阅读本书的过程中，重点可能不同。

如果读者希望全面了解 Android 系统，自前而后地阅读本书是非常适合的。通过对第 1 篇的系统框架结构和启动流程的学习，读者会更深入地理解 Android 系统是什么，由什么组成，涉及了哪些方面。第 2 篇作为第 1 篇的补充，也可以为读者提供大量较为实用的知识和技巧。

如果读者从事 Android 设备的全面开发，则通常需要具有 Linux 和 Java 较全面的知识，阅读要点如下所示：

- 第 1 篇的框架和系统结构部分是重点；
- 第 1 篇的编译系统部分帮助读者理解对系统的全局性配置；
- 第 1 篇通信部分和结构设计对于在系统中增加模块有所帮助；
- 第 1 篇的调试部分可以辅助读者的开发过程；
- 第 2 篇的系统定制有助于帮助读者理解系统的一些细节；

- 第2篇的兼容性和安全系统帮助读者了解和解决系统中深层次的问题。

如果读者从事 Android 设备硬件移植方面的开发，通常要对 Linux 语言的开发有较深的了解，阅读要点如下所示：

- 第1篇通信部分和具体硬件相关子系统结构相结合，了解一个硬件如何适配于 Android 系统；

- 第1篇的启动流程部分为读者提供了硬件优化方面的知识；
- 第1篇的 GUI 系统提供将 Android 系统适配于非手机设备的相关知识；
- 第2篇安全系统方面帮助读者了解新增硬件的安全管理。

如果读者从事 Android 应用程序的开发，则通常要对 Java 语言的开发有一定的理解，阅读要点如下所示：

- 第1篇 Java 框架部分帮助读者了解 Android 系统中的隐藏内容；
- 第2篇第10章帮助读者了解 SDK 开发的局限性，第三方应用和预置应用的区别；
- 第1篇 Java 层调试部分，第2篇测试和调试部分有助于读者深入调试应用；
- 第2篇预置应用部分帮助读者了解 Android 中预置应用程序层的结构；
- 第1篇的 JNI 部分和第2篇的 NDK 部分会对读者在应用程序中使用本地内容有所帮助。

本书作者

本书的编写工作由国内 Linux 和 Android 领域资深的技术人员韩超领衔。本书也是韩超付出心血最多，编写时间最长，亲自执笔比例最大的一本书。参与本书编写的还有来自 Android 开发领域的梁泉、崔海斌、马若劼、于仕林等，机锋网（www.gfan.com）的张宇、张超、赵家维、黄亮、沈桢、徐威特等，国软时代（www.gredu.org.cn）的孙丽娜，以及杨钰、曹道刚等。广大社区开发者也对本书的成稿做出了贡献。

本书的核心作者团队大都来自开发一线，在工作之余完成本书。为了使本书尽早和读者见面，编写过程略显仓促。虽然经过审校，可能依然存在一些错误，敬请读者谅解。

目 录

第 1 篇 Android 核心框架层开发

第 1 章 Android 系统框架	2
1.1 Android 系统基础结构	2
1.2 Android 的开发环境	3
1.2.1 开发环境	3
1.2.2 源代码获取	4
1.2.3 编译 Android 源代码	4
1.2.4 仿真器运行	5
1.2.5 使用工具	5
1.3 系统结构和代码结构	7
1.3.1 软件系统结构	7
1.3.2 源代码结构	9
第 2 章 编译结构和各种构建	13
2.1 Android 的系统构建	13
2.1.1 编译环境要求	13
2.1.2 构建流程	14
2.1.3 环境设置	16
2.1.4 系统构建结果	16
2.2 各种部件的构建	18
2.2.1 Android.mk 的语法	18
2.2.2 各种部件的构建方式	19
2.2.3 预编译内容的安装	23
2.3 Android 编译的板级支持	24
2.3.1 支持一个板的工作	24
2.3.2 buildspec.mk 文件的作用	25
2.3.3 vendersetup.sh 和 lunch 命令	26
2.3.4 BoardConfig.mk 文件的作用	27

2.3.5 AndroidProducts.mk 文件	28
2.3.6 AndroidBoard.mk 文件	29
2.3.7 编译中的层叠加	30

第 3 章 本地层基础部分	31
3.1 本地层基础部分结构	31
3.2 C 语言基础: Bionic	31
3.2.1 Bionic 各个部分的结构	32
3.2.2 Bionic 的头文件	32
3.2.3 内存泄露的检查库	33
3.2.4 属性系统的支持	34
3.3 C 语言底层库	35
3.3.1 Log 库	35
3.3.2 C 语言工具库 libcutils	36
3.3.3 属性和套接字支持	40
3.4 init 可执行程序和相关脚本	42
3.4.1 init 可执行程序的结构	42
3.4.2 启动脚本 init.rc	43
3.4.3 init 的内建动作和初始化	48
3.4.4 init 的循环	50
3.4.5 设备管理功能	55
3.5 shell 工具	57
3.5.1 shell 程序 sh	57
3.5.2 命令工具箱 toolbox	59
3.6 Android 属性和套接字机制	60
3.6.1 属性机制	60
3.6.2 套接字机制	63

第 4 章 本地层框架部分	67
4.1 本地层框架部分结构	67
4.2 框架层的工具库 libutls	67
4.2.1 libutls 的结构	68

4.2.2 基本的定义和类型.....	69	5.1.2 BootLoader 运行及加载 内核.....	124
4.2.3 数据结构和容器类.....	70	5.1.3 Linux 内核运行和加载根文件 系统.....	125
4.2.4 工具类.....	75	5.1.4 运行用户空间的第一个 进程.....	125
4.2.5 引用计数和强弱引用指针.....	77	5.1.5 挂接各个文件系统.....	126
4.2.6 线程类及相关辅助功能.....	83	5.1.6 系统各个进程的运行.....	126
4.2.7 文件操作的监听循环.....	87	5.1.7 启动图.....	128
4.3 作为全局守护进程的可执行程序.....	89	5.2 Android 本地部分的调试.....	130
4.3.1 app_process.....	89	5.2.1 标准 Linux 调试方法.....	130
4.3.2 keystore.....	90	5.2.2 使用 log 的方法.....	138
4.3.3 installld.....	91	5.2.3 Android 的特殊调试工具.....	139
4.3.4 vold.....	92	5.3 本地程序的结构设计.....	145
4.3.5 netd.....	94	5.3.1 目标和设计思路.....	145
4.4 Binder 的 IPC 机制结构.....	96	5.3.2 守护进程+保留 Socket 的 设计结构.....	146
4.4.1 Android Binder 机制的层次 结构.....	96	5.3.3 守护进程+本地层 Binder 的 设计结构.....	147
4.4.2 具体 Binder 的实现.....	97	第 6 章 Java 运行环境.....	149
4.5 Binder 库以及相关内容.....	98	6.1 Dalvik 虚拟机和相关工具.....	149
4.5.1 具体 Binder 的实现.....	98	6.1.1 Dalvik 的概述.....	149
4.5.2 Binder 库的核心功能.....	99	6.1.2 虚拟机的实现和运行.....	150
4.5.3 Binder 框架和具体 IPC 结构的实现.....	100	6.1.3 Dalvik 工具.....	154
4.5.4 IServiceManager 相关.....	110	6.1.4 nativehelper 库.....	156
4.5.5 Binder 库中的其他部分.....	113	6.2 Java 基础库 libcore.....	156
4.6 servicemanager 可执行程序 and 服务的实现.....	115	6.2.1 libcore 的源代码结构.....	156
4.6.1 服务管理器可执行程序 and 进程.....	115	6.2.2 luni 目录中的内容.....	158
4.6.2 在系统中实现“服务”.....	118	6.2.3 libcore 中的其他内容.....	159
4.6.3 servicemanager 对服务的 管理.....	121	第 7 章 Java 框架层.....	160
4.6.4 Android 中本地服务的层次 结构.....	121	7.1 Java 框架层结构.....	160
第 5 章 本地的启动流程、调试和 设计结构.....	123	7.2 Android 的 Java 层的 API.....	161
5.1 Android 启动流程的本地阶段.....	123	7.2.1 Java 框架层 API 的含义.....	161
5.1.1 启动流程概述.....	123	7.2.2 API 的描述文件.....	162
		7.2.3 被隐藏的内容.....	166
		7.2.4 不赞成使用的(过时的) API.....	166

7.3	Java 框架库 framework.jar.....	168
7.3.1	框架库的组成和作用.....	168
7.3.2	框架库的 API	168
7.3.3	框架库的编译结构.....	169
7.3.4	框架库的各个部分.....	170
7.3.5	属性机制.....	173
7.3.6	套接字机制.....	174
7.4	Java 服务库 services.jar	175
7.4.1	服务库的组成和作用.....	176
7.4.2	服务管理器.....	176
7.4.3	主要的服务功能.....	177
7.4.4	启动结束的处理.....	180
7.5	资源包 framework-res.apk	180
7.5.1	资源包的组成和作用.....	181
7.5.2	作为 API 的资源 id	182
7.5.3	资源包 AndroidManifest.Xml 文件.....	182
7.6	策略库.....	183
7.6.1	策略库的组成和作用.....	183
7.6.2	策略库的功能和使用.....	183
7.7	Java 层的 Binder 机制.....	184
7.7.1	Java Binder 的框架.....	185
7.7.2	Java 框架层的 Binder 使用... ..	189
7.7.3	Binder 中参数和返回值的 类型.....	194

第 8 章 Java 本地调用 198

8.1	Android 的 Java 本地调用	198
8.1.1	JNI 的基本概念.....	198
8.1.2	Android 的 JNI 实现基础.....	198
8.1.3	Android 中 JNI 的存在方式.....	201
8.2	在框架中使用 JNI.....	201
8.2.1	框架层使用 JNI 的结构.....	201
8.2.2	框架层 JNI 的实现.....	204
8.3	在应用程序包中使用 JNI.....	207
8.3.1	应用程序包的 JNI 部分	207

8.3.2	应用程序包中 JNI 的典型 实现方式.....	207
8.3.3	结构上的几种变化.....	210
8.4	JNI 的高级使用.....	211
8.4.1	JNI 的双向调用.....	211
8.4.2	线程的处理.....	215
8.4.3	JNIHelp 中的特殊处理	216

第 9 章 Java 启动流程、调试和程序 结构 218

9.1	Java 部分的启动流程	218
9.1.1	启动流程概述.....	218
9.1.2	Java 系统的守护进程.....	219
9.1.3	ZygoteInit 类的执行.....	220
9.1.4	系统服务器部分	222
9.1.5	Java 应用程序部分的启动.....	227
9.2	Java 部分的调试	229
9.2.1	在 Java 中使用 Log	229
9.2.2	Java 层的命令行程序.....	229
9.2.3	代码内容相关的调试.....	232
9.2.4	DDMS 工具	237
9.2.5	HierarchyViewer 工具	239
9.3	Java 层程序的结构设计	241
9.3.1	目标和设计思路.....	241
9.3.2	在系统服务器中服务的 设计结构.....	242
9.3.3	Service 组件的设计结构.....	243

第 10 章 GUI 系统和应用程序管理 245

10.1	GUI 系统的结构和模块关系	245
10.2	UI 框架库及相关	246
10.2.1	基本类型方面.....	246
10.2.2	显示相关.....	247
10.2.3	按键和事件的处理.....	247
10.2.4	其他内容.....	249
10.3	Surface 图层系统	249
10.3.1	Surface 系统的结构	250

10.3.2	Surface 本地的接口部分	251
10.3.3	Surface 本地实现部分	254
10.3.4	Surface 的 JNI 代码	255
10.3.5	Surface 的 Java 部分	256
10.4	2D 图形系统	258
10.4.1	2D 图形系统的结构	258
10.4.2	Skia 底层库	259
10.4.3	图形系统的 JNI 接口	263
10.4.4	2D 图形包	264
10.5	3D 图形系统	264
10.5.1	OpenGL 系统的结构	265
10.5.2	OpenGL 的本地代码	265
10.5.3	OpenGL 的 JNI 代码	267
10.5.4	OpenGL 的 Java 类	268
10.6	窗口和视图部分	270
10.6.1	窗口和视图部分的结构	270
10.6.2	窗口管理的底层实现	271
10.6.3	视图实现和上层的窗口 管理器的实现	275
10.7	活动和应用程序管理	277
10.7.1	活动和应用程序管理的 结构	277
10.7.2	活动界面的实现	278
10.7.3	应用程序管理的实现	282

第 2 篇 Android 系统级应用 开发要点

第 11 章 Android 应用层接口结构与 安全

11.1	Android 应用层接口结构	288
11.1.1	标准应用程序和源代码开发 的应用程序	288
11.1.2	第三方应用程序和预置应用 程序	290
11.2	Android 的安全系统	290

11.2.1	安全机制概述	290
11.2.2	Linux 的用户和组	291
11.2.3	Android 的许可	294
11.3	程序签名问题	296
11.3.1	Android 应用签名的概念	296
11.3.2	Android 应用签名的流程	296
11.3.3	不同的程序签名	298
11.3.4	证书生成工具	299
11.4	安全策略	300
11.4.1	安全策略的几个方面	300
11.4.2	安全策略的实现	306
11.4.3	安全策略的定制	310

第 12 章 应用层的本地内容和 NDK

	开发	312
12.1	本地应用的标准化	312
12.2	NDK 环境	313
12.2.1	NDK 的基本概念	313
12.2.2	从几个方面理解 NDK 的 不同含义	313
12.3	NDK 的组成结构	314
12.3.1	NDK 环境的结构	314
12.3.2	NDK 中包含的内容	315
12.4	NDK 开发基础	317
12.4.1	编译流程	317
12.4.2	NDK 编程要点	318
12.4.3	开发结构	319
12.5	使用 JNI 的方式	320
12.5.1	NDK 中使用 JNI 工程 结构	320
12.5.2	在本地操作 Bitmap 位图	320
12.5.3	在本地使用 OpenGL ES 绘制	321
12.5.4	程序实现	321
12.6	在本地构建应用	328
12.6.1	本地应用接口基础	328

12.6.2	实现本地活动的辅助库: android_native_app_glue.....	332	14.2.3	组件的 intent-filter	372
12.6.3	程序实现.....	333	14.3	桌面的定制	373
第 13 章	预置的应用程序层	339	14.3.1	桌面程序的入口和功能.....	374
13.1	预置应用层程序概述.....	339	14.3.2	桌面程序细节.....	375
13.2	预置综合应用程序.....	340	14.3.3	在桌面之前执行的程序.....	378
13.2.1	主屏幕.....	340	14.4	系统资源的定制	378
13.2.2	设置.....	342	14.4.1	定义资源的概念和原则.....	379
13.2.3	联系人和电话.....	345	14.4.2	一些配置的数值.....	379
13.2.4	短信息和彩信.....	347	14.4.3	可绘制内容的定制.....	379
13.2.5	浏览器和邮件.....	349	14.4.4	样式和主题的定制.....	380
13.2.6	音乐.....	352	14.4.5	内部布局的定制.....	382
13.2.7	照相机和录音机.....	354	14.5	框架内部组件和界面的定制	383
13.2.8	画廊.....	358	14.5.1	内部组件.....	383
13.2.9	搜索.....	359	14.5.2	内部界面.....	385
13.2.10	时钟、计算器和日历.....	360	14.6	状态栏界面的定制	385
13.3	预置内容提供者.....	361	14.6.1	状态栏界面的功能和 布局.....	385
13.3.1	联系人信息提供者.....	361	14.6.2	状态栏界面.....	387
13.3.2	媒体信息提供者.....	362	14.7	锁屏界面的定制	389
13.3.3	电话信息提供者.....	363	14.7.1	锁屏界面的功能.....	390
13.3.4	下载信息提供者.....	363	14.7.2	锁屏界面.....	391
13.4	预置输入法.....	364	14.8	开机动画的定制	393
13.5	活动墙纸系统.....	365	第 15 章	系统级的调试和测试	396
13.5.1	活动墙纸选择器.....	365	15.1	调试	396
13.5.2	几个具体的活动墙纸.....	366	15.1.1	命令行调试工具.....	396
13.6	实现框架功能的应用程序包.....	366	15.1.2	界面上的辅助工具.....	402
13.6.1	系统界面.....	367	15.2	测试系统和工具	403
13.6.2	设置信息提供者.....	368	15.2.1	测试系统的结构.....	404
13.6.3	几个主体为 Service 组件 的应用程序包.....	369	15.2.2	Monkey	404
			15.2.3	MonkeyRunner.....	406
第 14 章	Android 系统定制	371	15.3	代码级别的测试	409
14.1	Android 系统定制概述	371	15.3.1	代码测试框架.....	409
14.2	普通应用程序的定制.....	371	15.3.2	代码测试方式.....	411
14.2.1	普通应用定制的方面.....	371	15.4	兼容性及兼容性测试	414
14.2.2	预置的应用程序的组件 可见性.....	372	15.4.1	Android 兼容性	414
			15.4.2	CTS 兼容性验证	416
			15.4.3	CTS 的相关源代码和原理 ...	419

附录 A Android 版本演进和

Android 4.x	423
A.1 Android 版本的演进	423
A.2 编译系统的更新	423
A.3 本地框架层的更新	425
A.3.1 init.rc 中的关键字	425
A.3.2 surfaceflinger 作为独立的 守护进程	425
A.3.3 netd 守护进程	426
A.4 Java 框架的更新	426
A.4.1 Java 框架的 API 描述	426
A.4.2 Java 类库的升级	427
A.4.3 Java 服务包	428
A.4.4 工具	428
A.5 GUI 系统的更新	429
A.5.1 GUI 系统	429
A.5.2 Skia 引擎的改动	430
A.5.3 OpenGL 引擎的改动	430
A.5.4 窗口的变化	430
A.6 Java 应用程序层的更新	432
A.6.1 SystemUI	432
A.6.2 桌面	432
A.6.3 普通应用程序	433
A.7 本地标准应用和 NDK 环境	434

第1篇 Android 核心框架层开发

第1篇内容重点在于 Android 软件系统的中间两个层次。Android 是一个基于 Linux 的系统，其自下而上的软件结构是本书第1篇的主线。本书采用了自本地层到 Java 层，最后介绍 GUI 系统的方式，完整展现 Android 框架层的结构。

- 第1章 Android 系统框架
- 第2章 编译结构和各种构建
- 第3章 本地层基础部分
- 第4章 本地框架层部分
- 第5章 本地的启动流程、调试和设计结构
- 第6章 Java 运行环境
- 第7章 Java 框架层
- 第8章 Java 本地调用
- 第9章 Java 启动流程、调试和程序结构
- 第10章 GUI 系统和应用程序管理

第1章

Android 系统框架

1.1 Android 系统基础结构

Android 是一个开源的软件系统，框架层是 Android 系统的核心。框架层的开发也是 Android 开发的核心部分。

Android 系统的结构图和开发方面如图 1-1 所示。

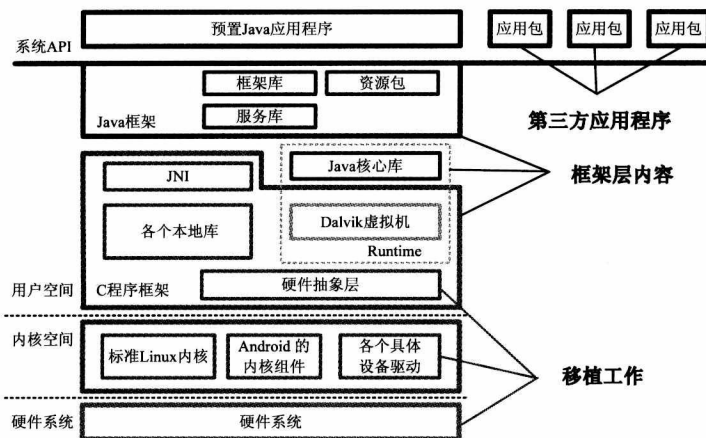


图 1-1 Android 系统的结构和开发方面

Android 的相关开发分为 Android 设备的开发和 Android 应用程序开发两个方面。Android 框架层对这两个方面的开发都具有支持作用。

研究和开发 Android 系统框架的意义主要在于下面几个方面。

- 深层次的跟踪和调试。作为开源系统，Android 的运行过程对开发者都是透明的。通过对框架层的深入研究，可以探寻到 Android 运行过程的全貌。
- 通用组件的实现。基于 Android 的框架，可以实现横向的一些模块，特定的 Android 设备，这些模块可以提供标准系统之外的额外功能。

- 辅助特定硬件平台的构建。在 Android 设备的特定硬件平台构建中，不仅需要在硬件移植层进行开发，还需要通过框架层以上的部分对其进行调试和测试。
- 应用程序的深入研究。Android 应用程序的开发依赖于框架层，虽然在经典模式中可以只关注 Java 框架层 API，但是 Java 框架层还有非 API 的隐藏接口和本地调用接口，基于它们可以实现额外的功能。
- 特殊系统级别应用程序。对于一个 Android 设备，有些预置的应用程序提供的是系统级别的功能，这种应用与框架层的耦合性较强，框架层研究对这部分内容的开发至关重要。

1.2 Android 的开发环境

1.2.1 开发环境

Android 的 Linux 开发环境一般使用 Ubuntu 主机。在基于 Ubuntu 的主机环境中，开发 Android 主机环境包括以下需求：git 工具、repo 工具、Java 的 JDK、主机编译工具等几个方面。编译过程中的主要目标机的编译工具则在 Android 的源代码包中自带。

Java 6 开发包（SUN 标准的 JDK 6）的安装方式如下所示：

```
$ sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"
$ sudo apt-get update
$ sudo apt-get install sun-java6-jdk
```

Java 5 开发包（SUN 标准的 JDK 5）的安装方式如下所示：

```
$ sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu hardy main multiverse"
$ sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu hardy-updates main multiverse"
$ sudo apt-get update
$ sudo apt-get install sun-java5-jdk
```

在 Android 1.6 至 Android 2.2 发布版本中，Android 系统要求使用 Java 5 来编译系统；Android 2.3.x 之后要求使用 Java 6 进行编译。

配置 Java 运行环境和 javac 编译器的方法如下所示：

```
$ update-alternatives --config java
$ update-alternatives --config javac
```



提示：不同版本的 JDK 可以共存，可以通过 update-alternatives 进行配置选择。

Android 系统在编译的过程中需要编译主机的工具，因此需要使用主机的 GCC 工具链。而对于编译目标机文件，Android 在 prebuilt 目录中集成了 GCC 交叉编译工具链。

新版本 Android 源代码编译应当使用 64 位的主机环境，配置方式如下所示：

```
$ sudo apt-get install git-core gnupg flex bison gperf build-essential \
zip curl zlib1g-dev libc6-dev lib32ncurses5-dev ia32-libs \
```