



2011年度
韩国文化体育观光部
优秀学术图书

Android 框架揭秘

Inside the Android Framework

[韩] 金泰延 宋亨周 朴知勋 李白 林起永 著 武传海 译

人民邮电出版社
POSTS & TELECOM PRESS



Android 框架揭秘

Inside the Android Framework

[韩] 金泰延 宋亨周 朴知勋 李白 林起永 著 武传海 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Android框架揭秘 / (韩) 金泰延等著 ; 武传海译
— 北京 : 人民邮电出版社, 2012. 4
ISBN 978-7-115-27476-2

I. ①A… II. ①金… ②武… 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014937号

版 权 声 明

“**Inside Android**” by Tae Yeon Kim, Hyung Joo Song, Ji Hoon Park, Bak Lee and Ki Young Lim
Copyright © 2010 Tae Yeon Kim, Hyung Joo Song, Ji Hoon Park, Bak Lee and Ki Young Lim
All rights reserved.

Originally Korean edition published by WIKIBOOKS

The Simplified Chinese Language edition © 2012 Posts & Telecom Press

The Simplified Chinese translation rights arranged with Tae Yeon Kim, Hyung Joo Song, Ji Hoon Park,
Bak Lee and Ki Young Lim through EntersKorea Co., Ltd., Seoul, Korea.

本书中文简体字版由 **WIKIBOOKS** 授权人民邮电出版社独家出版。未经出版者书面许可, 不得
以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

Android 框架揭秘

◆ 著 [韩] 金泰延 宋亨周 朴知勋 李 白 林起永
译 武传海
责任编辑 汪振

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷

◆ 开本: 800×1000 1/16
印张: 27.5
字数: 573 千字 2012 年 4 月第 1 版
印数: 1—4 000 册 2012 年 4 月河北第 1 次印刷

著作权合同登记号 图字: 01-2011-4239 号

ISBN 978-7-115-27476-2

定价: 69.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223
反盗版热线: (010)67171154

· 内容提要 ·

本书通过对 Android 系统源代码的分析，主要介绍 Android 框架的初始化过程及主要组件的工作原理。作者直接分析和整理了 Android 框架的主要源代码，并详细讲解了理解框架工作原理所需的各种基础知识和构成实际 Android 平台骨干的服务框架。其中的主要内容包括：通过启动程序了解 Android 框架的概要，移植 Android 以及如何开发适合各种机器的应用程序，分析 Android 框架所需的基础知识，JNI（Java Native Interface）与 Binder 基础知识，Zygote、Service Manager、Service Server 等 Android 核心组件，Android 服务框架的结构与理解，通过 Camera Service、Activity Manager Service 等服务分析实际的 Android 服务等。

本书不仅可以供具备一定开发经验的 Android 开发人员参考阅读，也可作为 Android 开发初学者的 Android 框架入门教材使用。

· 作者序 ·

宋亨周

笔者第一次接触 Android 是在 2009 年 5 月，当时苹果的 iPhone 获得了巨大成功，作为唯一能与之抗衡的操作系统，Android 引起了众多研发者的关注。一些出版社也相继推出了基于 Android SDK 应用程序的图书。跟随这股潮流笔者开始学习编写 Android 应用程序。

也许此前一直从事系统编程工作的缘故，笔者在学习编写 Android 应用程序的过程中，慢慢对 Android 系统的内部运行机制产生了兴趣。Android 系统是开放源码的，笔者想深入研究 Android 系统的内部运行机理，但无论在网络上还是书店中相关资料都很少，笔者在研究过程中处处碰壁，逐渐认识到单枪匹马、孤军作战不是良策，很难取得实质性进展。

于是，笔者成立了“水原 Android 平台研究小组”，和一群志同道合的朋友开始了对 Android 平台的探索。尽管大家的工作领域不尽相同，但是在共同兴趣爱好的驱使下，每周六我们都会聚集在一起，分析研究 Android 系统，分享各自的研究心得并整理出相关的研究资料。

本书正是在这些研究资料的基础上编写而成的，它凝聚了小组所有人的心血，是集体智慧的结晶。虽然本书并未涵盖 Android 的所有内容，但相信对初学 Android 系统的朋友会有相当大的帮助。记得有个诗句：“不要在雪地中盲目前行，请循着我的脚印，它将为你引路。”希望本书能成为读者的引路人。

本书是 Android 研究小组所有成员共同努力的结果，借此机会向各位表示深深的谢意。最后，还要感谢我的妻子贤静、儿子承民，一直以来默默地关心、理解、支持我，在此将本书一同献给你们，感谢你们无私的付出。

金泰延

从诞生到现在，Android 系统不断升级，每次升级都会增添许多新的功能与特性，开发人员很难适应这种变化，并且 Android 系统糅合了各种先进技术，要完全理解它很难。所以，学习 Android 系统编程时，不仅要熟悉 Linux、C++、Java 等各种技术，还要理解 Android 系统平台，研究它的组成结构及内部运行机制。

学习 Android 平台有什么好方法吗？我认为，可以借鉴我高中时学习数学的方法，首先将相关的公式和基本原理弄清楚，然后再去做应用题目，所有问题将迎刃而解。同理，学习 Android 应用程序开发，首先要理解其系统内部结构及核心原理。本书对 Android 平台中的 init 进程、Binder、JNI、Zygote、服务框架、核心系统服务等运行原理与机制进行了剖析与说明。

在开始编写本书时，Android 的版本为 1.6，到本书完稿时，已更新到 2.2。但是一般而言，核心原理不会轻易改变，而且本书侧重基础原理的讲解，受版本更新影响不大，请放心阅读。对于编程人员而言，本书是学习 Android 平台最好的起点。

在 Android 平台相关资料匮乏的前提下，本书内容完全基于对源代码的分析编写而成。源码分析过程耗时长达 6 个月，本书是研究小组所有成员智慧的结晶，正是他们无私的奉献与付出，本书才最终得以出版。在此，对研究小组的所有成员表示最诚挚的谢意。

在此特别要感谢宋亨周，Android 平台研究小组的组织者，亦是本书的执笔者，他付出了大量劳动，并给我们提供了参与的相会。同时，还要感谢参与本书编写的李白、林起永、朴知勋，感谢他们长期以来的努力。本书凝聚了全体成员的心血，大家辛苦了！

最后，还要感谢我的朋友与家人。首先，感谢蔡兴锡教授 4 年来对我孜孜不倦的教诲与爱护；其次是燕赞，感谢他把我介绍到 Android 研究小组，还有 Optimus Q 开发小组的曹尚贤、薛美英，李东勋，以及 301 同仁们，感谢他们的鼓励与支持。此外，还要感谢研究室的朋友如善烈、郑浩、显宰、燕智等，感谢他们试读本书并提出了中肯的意见。最后，感谢我的家人，感谢父母的关心与爱护，感谢妻子珍淑的理解与支持，使我能够在蜜月旅行中编写稿子。

朴知勋

加入水原 Android 平台研究小组，研究 Android 系统对我来说是一次全新的挑战。之前我从事 SoC (System On Chip) 工作，主要跟硬件开发打交道，对我而言，Android 融合的各种技术让我着迷，大大激发了我的好奇心与求知欲望。在分析 Android 系统内核的过程中，我学到了许多有关 Linux 深层次的知识，认识到了遍布整个框架的 Java 富于弹性的特征，还有了解了连接各种核心库的 JNI，以及进程孵化器 Zygote 等各种组成要素，这是一个充满乐趣的过程。而且，我们可以任意查看源码，这是 Android 的开源性带来的好处，也正是其魅力所在。

本书在 Android 源码分析的基础上编写而成。如果将分析源码比作一次 Android 框架探险，那么本书就是一本探险日志。那些在大学讲堂里开着投影仪，一点点分析源码的日子，至今仍历历在目，苦思冥想后豁然开朗的喜悦仍然记忆犹新。我们将这些过

程整理成书，希望借助本书跟大家分享知识、共同进步。

尽管没能做到面面俱到，我们还是尽可能地把各种重要的内容、细节纳入书中。从最终的成稿来看，我们做得相当不错。这得益于大家的大力支持，在此特别感谢林明柏，站在读者的立场上审校原稿，提出了很多宝贵意见；还要感谢高炫哲先生，在百忙之中抽出时间对 Binder IPC 原理部分进行审查，他的专业性意见是本书质量的重要保障。在此，感谢你们付出的每份努力，谢谢！

李白

从加入 Android 平台研究小组到现在，不知不觉已有一年多了。期间，很多其他兴趣小组解散了，但我们一直坚持了下来，并且将研究成果整理成书，这是小组所有成员共同努力的结果。回首过去，小组成员在一起共同学习的情景仍然历历在目，那种攻克难题后的喜悦仍然在脑海中萦绕，这些都将成为我永不磨灭的美好回忆。借此机会，向小组的各位成员表示感谢，谢谢你们每个人的付出。

最后，感谢我的家人。由于参加研究小组与编写本书而没有更多的时间陪伴你们，但你们毫无怨言，一直默默地关心我、支持我，谢谢你们！妻子朴美拉、儿子俊民，我爱你们！

林起永

当谷歌发布 Android OS 时，我非常兴奋。在 20 年前的个人电脑市场上，Apple、Amiga、IBM、MSX 等公司竞争非常激烈，后来 IBM 发布了开放式结构电脑，随之兼容 IBM 的个人电脑迅速占领了市场。

嵌入式设备的发展也经历了类似的时期与阶段。各大手机厂商开发生产的操作系统与手机硬件都是封闭的，并且所生产的手机通常只能用于通话，功能比较单一。就这样，这些厂商在各自相对封闭的领域内独占着市场并慢慢发展着。但是随着时代的发展和技术的进步，人们提出的要求越来越高，手机从简单的通话工具逐步发展成为掌上多媒体设备，于是智能手机出现了。苹果公司推出的智能手机备受消费者喜爱与推崇，市场点有率越来越大。

但是，包括苹果在内的许多智能手机厂商所开发的操作系统，相对而言仍然处于封闭状态，直到谷歌发布开源的 Android 系统，这种相对封闭的状态才被打破。大批手机厂商开始关注 Android，并使用这种系统开发手机。渐渐地，搭载 Android 系统的手机受到越来越多消费者的青睐，市场份额不断攀升。相信在不远的将来，谷歌 Android 会像 IBM 的兼容 PC 一样颠覆整个行业，究竟对我们的生活会产生怎样的影响，让我们翘首以待。

然而，在谷歌 Android 兴起并不断发展之时，有关 Android 平台研究的资料并不多见。市面上所见到的大部分图书，多是关于如何开发 Android 应用程序的，而对于其

系统框架内部的运行机制，平台核心元素分析的资料、图书仍然非常稀少。

Android 是个非常庞大、复杂的系统，仅凭个人的力量，想完全了解掌握它几乎是不可能的。于是在 2008 年，一个特别炎热的夏天，我们几个对 Android 系统感兴趣的人聚集在一起，结成了 Android 平台研究小组。我们利用业余时间聚在一起，一同分析研究 Android 系统，相互交流心得与体会，慢慢地，我们对 Android 系统了解得越来越多。为了将研究成果分享给大家，我们编写了本书，这是我们小组所有成员勤奋努力、集体智慧的结晶。Android 犹如一座巨型山脉，任何一本书想要将其完全包含进去都是不可能的，本书也不例外，但书中包含了 Android 系统核心部分的重要内容，相信对初学 Android 的朋友会有一定帮助，这也是编写此书的初衷。

本书由多名作者集体写作而成，他们分别是宋亨周、朴知勋、金泰延、李白，他们的热心、真诚令人印象深刻，我非常喜欢跟他们在一起学习、分享知识。在编写本书的过程中，我们相互鼓励，相互配合，最终写成此书，在此对他们表示最诚挚的谢意。

最后，感谢我的妻子惠珠，谢谢她一直以来对我的关心、照顾与支持，在此将本书献给我亲爱的惠珠。惠珠，感谢有你，我永远爱你！

· 前言 ·

在移动开发中，对开发者而言，哪些能力是必须具备的呢？

不同的人可能有不同的答案，但不管怎样，理解掌握目标设备中的系统框架是不可少的。如果你是一名程序开发人员，那么必须要学习 Windows 或 Linux 等操作系统的运行原理；如果你是一名网站开发人员，那么你必须要学习浏览器的运行机制。同样，如果你想成为移动开发领域中专家级的开发人员，那么你必须要学习相关系统框架的工作原理。

换言之，在移动开发领域中，研究学习相关系统框架的内部运行机制原理是成为高级开发人员的必由之路。这条规则也适用于 Android 开发人员。但是，与 Android 框架运行原理相关的资料，无论在网络上，还是书店中都很难找到。事实上，除了谷歌或谷歌 IO（谷歌开发者大会）提供的有限的资料之外，研究 Android 框架的资料和图书非常少。

目前，研究学习 Android 框架运行原理的最基本、最原始的做法就是直接分析 Android 系统源代码。由于有关 Android 框架分析的资料非常少，笔者们也是通过分析系统源码来研究 Android 框架的。但是面对庞大的 Android 系统源码，仍然不知从何处入手分析。

幸运的是，在谷歌 IO 2008 Android 开发者大会上发布了一份名为“Anatomy & Physiology of an Android”(<http://sites.google.com/site/io/anatomy--physiology-of-an-android>)的资料，资料中包含了 Android 框架初始化过程等内容，为我们分析 Android 框架指明了方向，提供了便利。

开始写作本书时，Android 最新版本为 1.5（代号：Cupcake），至本书完稿时，Android 更新到 2.2 版本（代号：Froyo）。所以我们根据 Android 2.2 对全书内容重新做了调整，以使本书内容与 Android 版本保持一致。事实上，从 Android 1.5 版本到 Android 2.2 版本，有关系统框架的源码并未发生重大变化，因此本书的内容受 Android 版本升级的影响不大。也许你阅读本书时，Android 又升级到新的版本，请不要担心，本书讲解的都是最基本、最核心的内容，对 Android 版本更新不敏感，请放心阅读。

本书以讲解分析 Android 框架内部运行原理为主，涵盖了 Android 框架的核心和

· 目录 ·

第 1 章	Android Framework 概要	1
1.1	Android 源代码组成	2
1.2	通过启动过程分析 Android Framework	3
第 2 章	搭建 Android 开发环境	7
2.1	主机环境构成	7
2.1.1	安装 VirtualBox	7
2.1.2	安装 Ubuntu	8
2.2	搭建 Android 平台编译环境	9
2.2.1	编译工具	10
2.2.2	安装 Repo	11
2.2.3	下载 Android 源代码	11
2.2.4	编译 Android 源代码	12
2.3	搭建 Android SDK 开发环境	13
2.3.1	下载、安装 Eclipse	13
2.3.2	下载 Android SDK starter	13
2.3.3	安装 ADT 插件	14
2.3.4	设置 Android SDK 路径	16
2.3.5	安装 Android SDK	16
2.4	开发 Android 应用程序	18
2.5	应用程序 Framework 源码级别调试	21
2.5.1	加载应用程序 Framework 源	21
2.5.2	调试 HelloWorld Framework (源码级)	24
2.6	小结	27

第 3 章	init 进程	29
3.1	init 进程运行过程	29
3.2	init 进程源码分析	31
3.3	init.rc 脚本文件分析与执行	40
3.3.1	动作列表 (Action List)	41
3.3.2	服务列表 (Service List)	43
3.3.3	init.rc 文件分析函数	44
3.3.4	动作列表与服务列表的运行	48
3.4	创建设备节点文件	52
3.4.1	创建静态设备节点	52
3.4.2	动态设备感知	57
3.5	进程的终止与再启动	58
3.6	属性服务	62
3.6.1	属性初始化	63
3.6.2	属性变更请求处理	65
3.7	小结	67
第 4 章	JNI 与 NDK	69
4.1	Android 与 JNI	69
4.2	JNI 的基本原理	72
4.2.1	在 Java 中调用 C 库函数	72
4.2.2	小结	83
4.3	调用 JNI 函数	84
4.3.1	调用 JNI 函数的示例程序结构	84
4.3.2	Java 层代码 (JniFuncMain.java)	85
4.3.3	分析 JNI 本地函数代码	87
4.3.4	编译及运行结果	101
4.3.5	在 Android 中的应用举例	102
4.4	在 C 程序中运行 Java 类	102
4.4.1	Invocation API 应用示例	103
4.4.2	编译及运行	108
4.4.3	Invocation API 在 Android 中的应用举例: Zygote 进程	110
4.5	直接注册 JNI 本地函数	110
4.5.1	加载本地库时, 注册 JNI 本地函数	111

4.5.2	Android 中的应用举例	115
4.6	使用 Android NDK 开发	122
4.6.1	安装 Android NDK	123
4.6.2	使用 Android NDK 开发步骤	127
4.6.3	小结	136
第 5 章 Zygote		137
5.1	Zygote 是什么	137
5.2	由 app_process 运行 ZygoteInit class	142
5.2.1	生成 AppRuntime 对象	143
5.2.2	调用 AppRuntime 对象	144
5.2.3	创建 Dalvik 虚拟机	145
5.2.4	运行 ZygoteInit 类	146
5.3	ZygoteInit 类的功能	147
5.3.1	绑定/dev/socket/zygote 套接字	149
5.3.2	加载应用程序 Framework 中的类与平台资源	150
5.3.3	运行 SystemServer	155
5.3.4	运行新 Android 应用程序	158
第 6 章 Android 服务概要		163
6.1	示例程序：理解 Android 服务的运行	163
6.2	Android 服务的种类	166
6.3	Android 应用程序服务	168
6.4	Android 系统服务	182
6.5	运行系统服务	185
6.5.1	分析媒体服务器(Media Server)的运行代码	186
6.5.2	分析系统服务器(System Server)的运行代码	188
6.6	Android Service Framework、Binder Driver 概要及相关术语	192
第 7 章 Android Binder IPC		197
7.1	Linux 内存空间与 Binder Driver	197
7.2	Android Binder Model	199

7.2.1	Binder IPC 数据传递	201
7.2.2	Binder IPC 数据流	202
7.2.3	Binder 协议 (Binder Protocol)	204
7.2.4	RPC 代码与 RPC 数据	206
7.2.5	Binder 寻址 (Binder Addressing)	206
7.3	Android Binder Driver 分析	209
7.3.1	从进程的角度看服务的使用	210
7.3.2	从 Binder Driver 角度看服务的使用	214
7.3.3	Binder Driver 函数分析	219
7.4	Context Manager	251
7.5	小结	256

第 8 章 Android Service Framework 257

8.1	服务框架 (Service Framework)	257
8.2	服务框架 (Service Framework) 的构成	259
8.2.1	各层构成元素的配置	260
8.2.2	各层构成元素间的相互作用	261
8.2.3	类的结构	264
8.3	运行机制	266
8.3.1	服务接口	267
8.3.2	服务	273
8.3.3	服务代理 (Service Proxy)	276
8.3.4	Binder IPC 处理	280
8.4	本地服务管理器 (Native Service Manager)	282
8.4.1	Service Manager 概要	282
8.4.2	Service Manager 类	284
8.4.3	Service Manager 的运行	286
8.5	编写本地服务	314
8.5.1	设计 HelloWorld 系统服务	314
8.5.2	HelloWorld 服务接口	315
8.5.3	HelloWorld 服务	316
8.5.4	HelloWorld 服务代理	319
8.5.5	运行 HelloWorld 服务	320
8.6	小结	325

第 9 章 本地系统服务（Native System Service）分析327

9.1 相机服务（Camera Service）	327
9.2 相机应用程序	328
9.3 相机服务框架（Camera Service Framework）	331
9.3.1 相机服务框架层次结构	331
9.3.2 相机服务框架类	333
9.4 相机服务框架的运行	334
9.4.1 初始化相机服务	334
9.4.2 连接相机服务	335
9.4.3 相机服务连接过程分析	337
9.4.4 相机设置与控制	340
9.4.5 相机设置与控制分析	341
9.4.6 相机事件处理	342
9.4.7 相机事件处理分析	343
9.5 小结	345

第 10 章 Java 服务框架（Java Service Framework）347

10.1 Java 服务框架（Java Service Framework）	347
10.1.1 Java 服务框架的层次结构	348
10.1.2 Java 服务框架中各个类间的相互作用	351
10.2 运行机制	354
10.2.1 Java 服务框架初始化	355
10.2.2 Binder	355
10.2.3 BinderProxy	361
10.2.4 Parcel	364
10.3 Java 系统服务的实现	367
10.3.1 闹钟服务（Alarm Manager Service）分析	368
10.3.2 编写 HelloWorldService 系统服务	372
10.3.3 使用 HelloWorldService 系统服务	375
10.3.4 编译 HelloWorldService 系统服务	378
10.4 Java Service Manager	380
10.4.1 Java Service Manager 简介	380
10.4.2 BinderInternal	381
10.4.3 Java Service Manager 的运行实例	383

10.5	使用 AIDL 生成服务代理与服务 Stub	389
10.5.1	在 AIDL 文件中定义服务接口	390
10.5.2	使用 AIDL 编译器, 生成服务接口、服务 Stub 以及服务代理	391
10.5.3	继承 Stub 类创建服务	392
10.5.4	服务接口的调用	393
10.6	小结	394
第 11 章	Java 系统服务运行分析	395
11.1	Activity Manager Service	395
11.2	Activity Manager Service 创建服务分析	397
11.2.1	Controller Activity-调用 startService()方法	398
11.2.2	Activity Manager Service 的 startService()方法的调用过程 (使用 Binder RPC)	399
11.2.3	Activity Manager Service——运行 startService()Stub 方法	405
11.2.4	运行 ActivityThread 类的 main()方法	409
11.2.5	Activity Manager Service——attachApplication() Stub 方法	414
11.3	小结	421
附录	AIDL 语法	423

第 1 章

Android Framework 概要

安卓（Android）是一个移动终端操作系统平台，由谷歌在 2007 年 11 月 5 日发布，它采用软件堆层（software stack，又名软件叠层）的架构，主要由操作系统、中间件、核心应用程序组成。安卓提供了一整套的软件框架，方便开发者开发基于移动终端的各种应用程序。关于安卓更详细的介绍，在安卓开发者网站（<http://developer.android.com>）中，可以查找到相关资料。

如果你有过开发 Android 应用程序的经验，即使你对 Android Framework 掌握得不深，通过谷歌提供的 Android SDK，也能非常容易地开发出基于 Android 的交互应用程序。这得益于 Android 为开发者提供了一套定义良好的软件框架，开发者即使不具备特别高深的专业知识，在短时间内同样能开发出强大的 Android 应用程序。

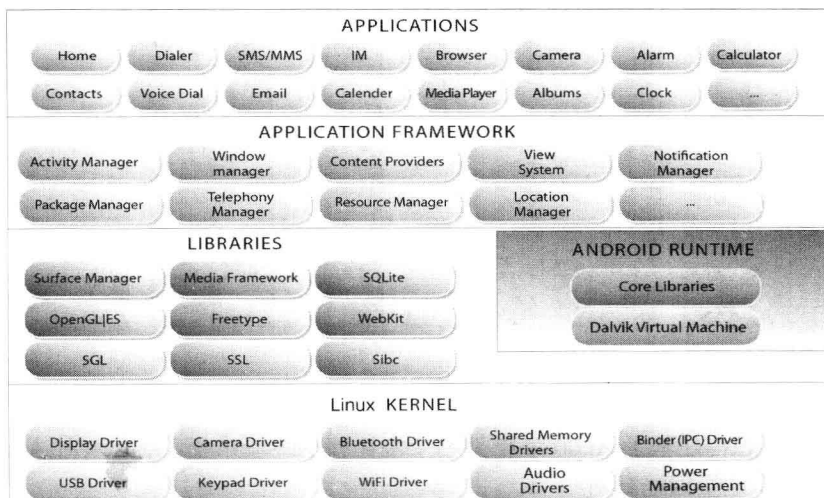


图 1-1 | Android 系统架构

图 1-1 是 Android 系统架构图，如图所示，Android 采用了分层架构，分为 4 个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 核心层。

观察 Android 系统架构图可以发现，其提供的基本应用程序（Home, Camera, Dialer, Browser 等）运行在应用程序框架层之上。同样，开发者借助 Android SDK，调用应用程序框架 API 开发出的应用也运行在应用程序框架层之上。

既然 Android 框架如此重要，那是否意味着开发者要全面、系统地掌握它呢？

关于这点，前面已略有提及。对于一个开发者而言，即使没有完全掌握 Android 框架，同样能够编写出一些不错的应用程序。当然，这并不是鼓励大家不去学习它。事实上，如果你对该框架了解得越多、越熟悉，掌握了相关原理，你会开发出更适合于 Android 框架，更优化的应用程序。在本书中，不会讨论 Android 自带的 AlarmClock、Contacts 等程序，但这些程序是经过严格测试的、高质量的程序，研读这些程序的源代码，对于开发者编写优秀的应用程序非常有益。

当然，如果你想成为一名优秀的 Android 平台应用程序开发者，那么你有必要深入了解、学习 Android 框架。Android 是一个真正开放的移动开发平台，访问其网站即可轻松地获取其源代码。每个硬件厂商可以根据自身需要定制基本的 Android 框架，开发出与竞争对手不同的产品。图 1-2 中显示的是各硬件厂商分别定制不同的 Android UI（安卓用户界面），从左往右依次为 Android 模拟器基本 UI、HTC UI、三星 UI、索尼爱立信 UI。



Android 模拟器基本 UI

HTC Sense UI

Samsung Touchwiz UI

Sony Ericsson Rachel UI

图 1-2 | Android 模拟器基本 UI 及各硬件厂商定制的 Android UI

像这样，若想构建基于 Android 框架的个性化移动终端系统，则必须对 Android 框架进行系统、全面地学习与研究。

1.1 Android 源代码组成

那么，我们该如何学习 Android 框架呢？其实，学习 Android 框架时，最准确、最