

· 移动平台开发书库 ·

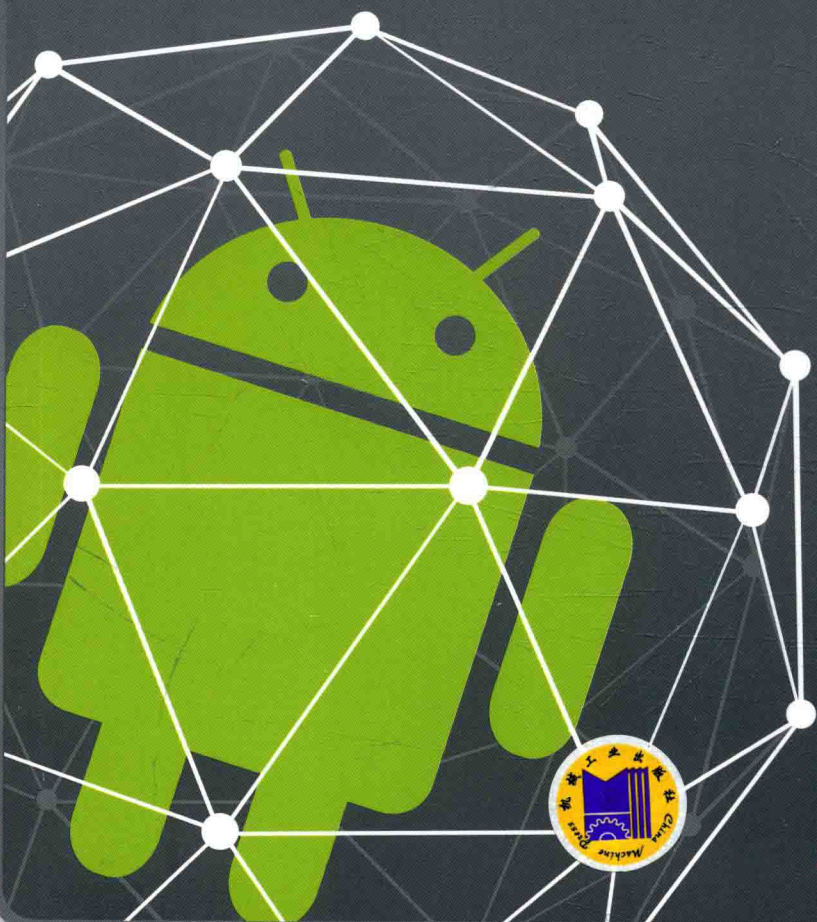
本书配套电子课件和源代码
可从 www.cmpbook.com 下载

Android

系统优化

胡郁 等编著

从入门到精通



 机械工业出版社
CHINA MACHINE PRESS

移动平台开发书库

Android 系统优化从入门到精通

胡 郁 等编著



机械工业出版社

Android 是当前最热门的智能手机操作系统之一，已经稳居智能手机操作系统第一名。本书采用理论结合实际的方法，循序渐进地讲解了优化 Android 系统的基本知识。全书分为 3 篇 15 章，内容包括 Android 系统介绍，获取并编译 Android 源码，分析内存系统，Android 内存优化，UI 布局优化，优化代码性能，Dalvik 虚拟机垃圾收集机制，Dalvik 虚拟机内存优化机制，Dalvik 虚拟机异常处理，JIT 编译，ART 优化之启动过程，ART 优化之执行主程序，ART 优化之安装 APK 准备，ART 优化之安装 APK 应用程序，系统优化。

本书不但适用于 Android 开发的初学者，也适用于准备向 Android 开发转型的编程人员，也可以作为有一定 Android 开发经验的程序员的参考书。

图书在版编目 (CIP) 数据

Android 系统优化从入门到精通 / 胡郁等编著. —北京: 机械工业出版社, 2015.10

(移动平台开发书库)

ISBN 978-7-111-51616-3

I. ①A… II. ①胡… 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 225126 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 丁 诚 责任编辑: 丁 诚 张 恒

责任校对: 张艳霞 责任印制: 乔 宇

保定市中国画美凯印刷有限公司印刷

2015 年 11 月第 1 版 · 第 1 次印刷

184mm×260mm · 30 印张 · 744 千字

0001—3000 册

标准书号: ISBN 978-7-111-51616-3

定价: 75.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88361066

机工官网: www.cmpbook.com

读者购书热线: (010) 68326294

机工官博: weibo.com/cmp1952

(010) 88379203

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金书网: www.golden-book.com

前 言

Android 是一款基于 Linux 平台的开源手机操作系统，该平台由操作系统、中间件、用户界面和应用软件组成，是为移动终端打造的真正开放和完整的移动软件。根据国际数据公司 (IDC) 公布的统计数据，在 2014 年度，Android 和 iOS 系统的装机量占到所有智能手机出货量的 96.3%，其中安装 Android 系统的新智能手机数量跃升至 10.59 亿部，同比增长 32%；市场份额为 81.5%；iOS 系统智能手机出货量 1.927 亿部，同比增长 25.6%；市场份额为 14.8%。有理由相信，在未来一段时间内，Android 将依旧牢牢地占据智能手机操作系统第一的位置。

强大的市场占有率使更多开发人员关注这款系统，当然也不乏初学者涌入学习大军中来，因此帮助初学者学习、解惑的相关图书非常畅销。但是在众多相关书籍中，大多数是入门级的，而 Android 系统优化领域的书籍屈指可数，Android 系统优化领域的专业级书籍更是寥寥无几。

只有更加专业才是通往 Android 殿堂级高手的必经之路。为了让广大初学者可以对 Android 开发有一个更加深入的认识，而不是停留在所谓的入门阶段。本书对 Android 优化方面的知识进行了细致的分析，“提炼”出了埋藏在 Android 系统深处的本质。并以此为基础，学以致用地讲解了实现项目优化的流程。

本书的内容

全书分为 3 篇 15 章，内容包括 Android 系统介绍，获取并编译 Android 源码，分析内存系统，Android 内存优化，UI 布局优化，优化代码性能，Dalvik 虚拟机垃圾收集机制，Dalvik 虚拟机内存优化机制，Dalvik 虚拟机异常处理，JIT 编译，ART 优化之启动过程，ART 优化之执行主程序，ART 优化之安装 APK 准备，ART 优化之安装 APK 应用程序，系统优化。本书几乎涵盖了 Android 系统优化中的所有主要内容，并且内容言简意赅，讲解方法通俗易懂，不但适合应用高手们学习，而且特别有利于初学者入门及提高。

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 2013 年 11 月发布版本 4.4，一共存在十多个版本。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。如果读者过于追求新版本，一定会力不从心。所以在此建议读者：“不必追逐最新的版本，我们只需关注最通用的版本即可”。据官方统计，截至 2013 年 11 月 25 日，占据应用前三位的版本分别是 Android 4.3，Android 4.2 和 Android 4.1，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。

2014 年 6 月 26 日，谷歌 I/O 大会在旧金山开幕。在会上谷歌发布了 Android L 系统，其

正式版于 2014 年 10 月 16 日推出。本书的内容以编者撰稿时的最新版本 Android L 为基础，并且兼容了 Android 4.4 及其以前的版本，详细讲解了 Android 系统优化的基本知识。

广大读者在调试使用本书中的源码时，可以使用任意版本的 Android SDK 进行调试，前提是在文件 AndroidManifest.xml 中指明版本号。例如，如果想使用 Android L 进行调试，需要在文件 AndroidManifest.xml 中加入如下指定 Android SDK 的代码。

```
<uses-sdk
    android:minSdkVersion="L"
    android:targetSdkVersion="L"
/>
```

本书特色

本书内容丰富，分析细致、全面。本书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。本书内容的编写具有以下特色。

(1) 结构合理

从用户的实际需要出发，科学安排知识结构，内容由浅入深，叙述清晰。全书详细地讲解了和 Android 优化开发有关的源码，内容循序渐进，由浅入深。

(2) 遵循“理论介绍—演示实例—综合演练”这一主线

为了使广大读者彻底弄清楚 Android 优化的每一个知识点，在讲解时依次剖析了基本理论、演示实例分析、综合实战演练等内容。遵循了从理论到实践，实现了实践教学这一目标。

(3) 易学易懂

本书内容条理清晰、语言简洁，可以帮助读者快速掌握各个知识点。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针对性的学习。

(4) 实用性强

本书彻底摒弃枯燥的理论和简单的操作，注重实用性和可操作性，通过通俗的语言和细腻的描述，详细讲解了各个知识点的基本知识。

(5) 内容全面

本书可称为“内容最全面的一本 Android 系统优化开发图书”之一，无论是开发环境搭建，还是各个常用、常见的网络系统问题，在本书中都能找到解决问题的方法。

读者对象

- Android 编程的自学者。
- 优化开发人员。
- 大中专院校的教师和学生。
- 正在完成毕业设计的学生。

- Android 编程爱好者。
- 相关培训机构的教师和学员。
- 从事 Android 开发的程序员。

本书主要由胡郁编写，参与编写的还有管西京、周秀、张余、李佐彬、王梦、王书鹏、唐凯、关立勋、张建敏、杨靖宇、谭贞军、杨絮、刘英田、高秀云、任杰、张子帝、黄河、孟娜、杨国华、王南荻、翟明、焦甜甜、张储、刘继虎。由于时间仓促，书中难免有疏漏和不妥之处，敬请广大读者与同行专家批评指正。

编者

目 录

前言

第一篇 基础知识篇

第 1 章 Android 系统介绍	1
1.1 智能手机系统介绍	1
1.1.1 什么是智能手机	1
1.1.2 主流智能系统的发展现状	2
1.2 Android 系统的发展现状	3
1.2.1 Android 系统的诞生和发展现状	3
1.2.2 常见的 Android 设备	4
1.2.3 Android 系统的巨大优势	6
1.3 搭建 Android 应用开发环境	7
1.3.1 安装 Android SDK 的系统要求	7
1.3.2 安装 JDK	8
1.3.3 获取并安装 Eclipse 和 Android SDK	11
1.3.4 安装 ADT	14
1.3.5 设定 Android SDK Home	16
1.3.6 验证开发环境	17
1.3.7 创建 Android 虚拟设备 (AVD)	18
1.3.8 启动 AVD 模拟器	21
1.4 创建第一个 Android 程序	23
第 2 章 获取并编译 Android 源码	28
2.1 获取 Android 源码	28
2.1.1 在 Linux 系统获取 Android 源码	28
2.1.2 在 Windows 平台获取 Android 源码	29
2.2 分析 Android 源码结构	32
2.3 编译 Android 源码	33
2.3.1 搭建编译环境	34
2.3.2 开始编译	35
2.3.3 在模拟器中运行	36
2.3.4 常见的错误分析	37
2.3.5 实践演练——两种编译 Android 程序的方法演示	38

第二篇 核心技术篇

第3章 分析内存系统	43
3.1 分析 Android 的进程通信机制.....	43
3.1.1 Android 的进程间通信 (IPC) 机制 Binder.....	43
3.1.2 Binder 机制的上下文管理者——Service Manager.....	44
3.1.3 Service Manager 服务.....	64
3.2 匿名共享内存子系统详解.....	67
3.2.1 基础数据结构.....	67
3.2.2 初始化处理.....	68
3.2.3 打开匿名共享内存设备文件.....	70
3.2.4 内存映射.....	72
3.2.5 读写操作.....	74
3.2.6 锁定和解锁.....	76
3.2.7 回收内存块.....	82
3.3 C++访问接口层详解.....	83
3.3.1 接口 MemoryHeapBase.....	83
3.3.2 接口 MemoryBase.....	92
3.4 Java 访问接口层详解.....	96
第4章 Android 内存优化	100
4.1 Android 内存优化的作用.....	100
4.2 查看 Android 内存和 CPU 使用情况.....	101
4.2.1 利用 Android API 函数查看内存.....	101
4.2.2 直接对 Android 文件进行解析查询.....	101
4.2.3 通过 Runtime 类实现.....	102
4.2.4 使用 DDMS 工具获取.....	102
4.2.5 其他方法.....	107
4.3 Android 系统的内存泄露.....	110
4.3.1 什么是内存泄露.....	110
4.3.2 为什么会发生内存泄露.....	111
4.3.3 shallow size 和 retained size.....	112
4.3.4 查看 Android 内存泄露的工具.....	113
4.3.5 查看 Android 内存泄露的方法.....	116
4.3.6 Android (Java) 编码时的注意事项.....	118
4.4 常见的引起内存泄露的陋习.....	119

4.4.1	查询数据库时忘记关闭游标	119
4.4.2	构造 Adapter 时不习惯使用缓存的 convertView	120
4.4.3	没有及时释放对象的引用	121
4.4.4	不在使用 Bitmap 对象时调用 recycle()释放内存	122
4.5	演练解决内存泄露	122
4.5.1	使用 MAT 根据 heap dump 分析 Java 代码内存泄漏的根源	123
4.5.2	演练 Android 中内存泄露代码优化及检测	130
4.6	Android 图片的内存优化	132
第 5 章	UI 布局优化	134
5.1	和布局相关的组件	134
5.1.1	View 视图组件	134
5.1.2	ViewGroup 容器	134
5.2	Android 中的五种布局方式	135
5.2.1	线性布局 LinearLayout	135
5.2.2	框架布局 FrameLayout	136
5.2.3	绝对布局 AbsoluteLayout	137
5.2.4	相对布局 RelativeLayout	137
5.2.5	表格布局 TableLayout	139
5.3	使用<merge />标签优化 UI 界面	140
5.3.1	注意事项	140
5.3.2	具体实现	140
5.4	优化 Bitmap 图片	143
5.4.1	显示一副图片	143
5.4.2	获取图片的宽度和高度	143
5.5	FrameLayout 布局优化	145
5.5.1	使用<merge />减少视图层级结构	146
5.5.2	使用<include />重用 layout 代码	147
5.5.3	延迟加载	149
5.6	使用 Android 提供的优化工具	150
5.6.1	Layout Optimization 工具	150
5.6.2	Hierarchy Viewer 工具	153
5.6.3	联合使用<merge />和<include />标签实现互补	155
第 6 章	优化代码性能	160
6.1	编写更高效的 Android 代码	160
6.1.1	避免建立对象	160

6.1.2	优化方法调用代码	162
6.1.3	优化代码变量	163
6.1.4	优化代码过程	166
6.1.5	提高 Cursor 查询数据的性能	169
6.1.6	编码中尽量使用 ContentProvider 共享数据	169
6.2	Android 控件的性能优化	174
6.2.1	ListView 控件的代码优化	174
6.2.2	Adapter (适配器) 优化	178
6.2.3	ListView 异步加载图片优化	181
6.3	优化 Android 图形	186
6.3.1	2D 绘图的基本优化	186
6.3.2	触发屏幕图形触摸器的优化	186
6.3.3	SurfaceView 绘图覆盖刷新及脏矩形刷新方法	187
6.4	资源存储优化	193
6.4.1	Android 文件存储	193
6.4.2	Android 中的资源存储	195
6.4.3	Android 资源的类型和命名	198
6.4.4	Android 文件资源 (raw/data/asset) 的存取	198
6.4.5	Android 对 Drawable 对象的优化	200
6.4.6	建议使用 Drawable, 而不是 Bitmap	202
6.5	加载 APK 文件和 DEX 文件	205
6.5.1	APK 文件介绍	207
6.5.2	DEX 文件介绍和优化	208
6.5.3	Android 类动态加载技术实现加密优化	208

第三篇 Dalvik 虚拟机优化篇

第 7 章	Dalvik 虚拟机垃圾收集机制	212
7.1	引用计数算法	212
7.2	Mark Sweep 算法	212
7.3	和垃圾收集算法有关的函数	214
7.4	垃圾回收的时机	234
7.5	调试信息	236
7.6	Dalvik 虚拟机和 JVM 垃圾收集机制的区别	236
第 8 章	Dalvik 虚拟机内存优化机制	239
8.1	sp 和 wp 简介	239

8.1.1	sp 基础	239
8.1.2	wp 基础	240
8.2	智能指针详解	241
8.2.1	智能指针基础	242
8.2.2	轻量级指针	243
8.2.3	强指针	245
8.2.4	弱指针	257
第 9 章	Dalvik 虚拟机异常处理	261
9.1	Java 异常处理机制	261
9.1.1	方法调用栈	261
9.1.2	Java 提供的异常处理类	263
9.2	Java 虚拟机异常处理机制详解	264
9.2.1	Java 语言及虚拟机的异常处理机制	265
9.2.2	COSIX 虚拟机异常处理的设计与实现	265
9.3	分析 Dalvik 虚拟机异常处理的源码	269
9.3.1	初始化虚拟机使用的异常 Java 类库	269
9.3.2	抛出一个线程异常	270
9.3.3	持续抛出进程	271
9.3.4	找出异常原因	272
9.3.5	清除挂起的异常和等待初始化的异常	276
9.3.6	解决“现在等待”异常	276
9.3.7	输出跟踪当前异常的错误信息	277
9.3.8	搜索和当前异常相匹配的方法	278
9.3.9	获取匹配的捕获块	279
9.3.10	进行堆栈跟踪	280
9.3.11	生成堆栈跟踪元素	282
9.3.12	将内容添加到堆栈跟踪日志中	283
9.3.13	将异常日志信息输出为堆栈跟踪信息	284
9.4	常见异常的类型与原因	284
9.4.1	SQLException: 操作数据库异常类	285
9.4.2	ClassCastException: 数据类型转换异常	285
9.4.3	NumberFormatException: 字符串转换为数字类型时抛出的异常	285
9.5	调用堆栈跟踪分析异常	286
9.5.1	解决段错误	286
9.5.2	跟踪 Android Callback 调用堆栈	289

第 10 章 JIT 编译	294
10.1 JIT 简介	294
10.1.1 JIT 概述	294
10.1.2 Java 虚拟机主要的优化技术	296
10.1.3 Dalvik 中 JIT 的实现	296
10.2 Dalvik 虚拟机对 JIT 的支持	296
10.3 汇编代码和改动	298
10.3.1 汇编部分代码	298
10.3.2 对 C 文件的改动	298
10.4 Dalvik 虚拟机中的 JIT 源码	299
10.4.1 入口文件	299
10.4.2 核心函数	310
10.4.3 编译文件	313
10.4.4 BasicBlock 处理	322
10.4.5 内存初始化	323
10.4.6 对 JIT 源码的总结	326
第 11 章 ART 优化之启动过程	328
11.1 运行环境的转换	328
11.2 运行 app_process 进程	329
11.3 准备启动	333
11.4 创建运行实例	340
11.5 注册本地 JNI 函数	342
11.6 启动守护进程	343
11.7 解析参数	344
11.8 初始化类、方法和域	352
第 12 章 ART 优化之执行主程序	359
12.1 进入 main 主函数	359
12.2 查找目标类	361
12.2.1 函数 LookupClass()	361
12.2.2 函数 DefineClass()	364
12.2.3 函数 InsertClass()	368
12.2.4 函数 LinkClass()	369
12.3 类操作	371
12.4 实现托管操作	373
第 13 章 ART 优化之安装 APK 准备	380
13.1 PackageManagerService 概述	380

13.2	主函数 main	380
13.3	调用初始化函数	381
13.4	创建 PackageManagerService 服务	384
13.5	扫描并解析	386
13.6	保存解析信息	405
第 14 章	ART 优化之安装 APK 应用程序	408
14.1	Android 安装 APK 概述	408
14.2	启动时安装	408
14.3	ART 安装	420
14.4	实现 dex2oat 转换	426
14.4.1	参数解析	427
14.4.2	创建 oat 文件指针	429
14.4.3	dex2oat 准备工作	430
14.4.4	提取 classes.dex 文件	431
14.4.5	创建 oat 文件	438
14.5	APK 文件的转换	439
第 15 章	系统优化	441
15.1	基本系统优化	441
15.1.1	刷机重启	441
15.1.2	刷内核	442
15.1.3	精简内置应用	442
15.1.4	基本系统优化总结	444
15.2	进程管理	444
15.2.1	Android 进程跟 Windows 进程是两回事	445
15.2.2	查看当前系统中正在运行的程序	445
15.2.3	枚举 Android 系统的进程、任务和服务的信息	449
15.2.4	研究 Android 进程管理器的实现	455
15.3	将 Android 软件从手机内存转移到存储卡	460
15.3.1	第一步: 准备工作	460
15.3.2	第二步: 存储卡分区	463
15.3.3	第三步: 将软件移动到 SD 卡	463
15.4	常用的系统优化工具	465
15.4.1	优化大师	465
15.4.2	360 优化大师	466

第一篇 基础知识篇

第1章 Android 系统介绍

Android 是一款移动智能设备（手机、平板电脑等）的操作系统，它以 Linux 开源操作系统作为内核基础，能够为企业和开发人员迅速建立移动智能设备软件解决方案。从 2011 年开始，Android 系统一直占据全球智能手机市场占有率第一的位置。本章将简要介绍 Android 系统的发展历程和背景，并介绍搭建 Android 应用开发环境的基本知识，为学习本书后面知识打下基础。

1.1 智能手机系统介绍

近年来，智能手机大大丰富了人们的生活，得到了广大手机用户的青睐。各大手机厂商纷纷建立了自己的智能手机操作系统，并且大肆招兵买马来抢夺市场份额。在本节的内容中，将简要介绍智能手机系统的基本知识。

1.1.1 什么是智能手机

智能手机是指具有像个人电脑那样强大的功能，拥有独立的操作系统，用户可以自行安装应用软件、游戏等第三方提供的程序，并且可以通过移动通信网络接入到无线网络中。在 Android 系统诞生之前，已经有很多优秀的智能手机操作系统存在，例如，家喻户晓的 Symbian 系列和微软的 Windows Mobile 系列等。

一般来说，智能手机必须具备如下几个功能。

- (1) 操作系统必须支持新应用的安装。
- (2) 处理器拥有高速度处理的能力。
- (3) 可以播放各种音频和视频文件。
- (4) 大容量存储芯片和存储扩展能力。
- (5) 支持 GPS 导航。

根据上述标准，手机联盟公布了智能手机的如下几个主要特点。

- (1) 具备普通手机的所有功能，例如，拨打、收听电话和收发短信等。
- (2) 是一个开放性的操作系统，在系统上可以安装第三方应用程序，从而实现功能的无限扩充。
- (3) 具备上网功能，例如，可以浏览网页。
- (4) 具备 PDA 的功能，例如，能够实现个人信息管理、日程记事、任务安排、多媒体应用、浏览网页等功能。



(5) 扩展性能强，可以根据个人需要扩展机器的功能。

1.1.2 主流智能系统的发展现状

在当今市面中或曾经一段时期，最主流的智能手机系统当属微软 Windows Mobile、塞班 Symbian、Palm、黑莓 BlackBerry、苹果 iOS 和本书的主角 Android。

1. 微软的 Windows Mobile

Windows Mobile 是微软公司的一款杰出的产品，Windows Mobile 将熟悉的 Windows 桌面扩展到了个人设备中。使用 Windows Mobile 操作系统的设备主要有 PPC 手机、PDA、随身音乐播放器等。Windows Mobile 操作系统有三种，分别是 Windows Mobile Standard、Windows Mobile Professional, Windows Mobile Classic。目前较新的版本是 Windows Phone 7 和 Windows Phone 8。

2. Symbian (塞班)

塞班系统出自诺基亚、索尼爱立信、摩托罗拉、西门子等几家大型移动通讯设备商共同出资组建的一个合资公司，专门研发手机操作系统，后来被诺基亚全额收购。Symbian 有着良好的界面，采用内核与界面分离技术，对硬件的要求比较低，支持 C++、Visual Basic 和 J2ME。目前根据人机界面的不同，Symbian 体系的用户界面 (User Interface, UI) 平台分为 Series60、Series80、Series90、UIQ 等。其中 Series60 主要是给数字键盘手机用，Series80 是为完整键盘所设计，Series90 则是为触控笔方式而设计。

2013 年 9 月 3 日，微软公司宣布将以 37.9 亿欧元的价格收购诺基亚的设备和服 务部门，同时还将以 16.5 亿欧元的价格收购诺基亚的相关技术专利，本次交易总额达到 54.4 亿欧元，其中有 3.2 万名员工将从诺基亚转入微软，整笔交易预计将于 2014 年第一季度完成。微软收购诺基亚公司后，将完全抛弃塞班系统，而主推 Windows Phone 操作系统的移动设备产品。

3. Palm

Palm 是流行的个人数字助理 (PDA, 又称掌上电脑) 的传统名称。从广义上讲，Palm 是 PDA 的一种，是 Palm 公司发明的。而从狭义上讲，Palm 是 Palm 公司生产的 PDA 产品，区别于 SONY 公司的 Clie 和 Handspring 公司的 Visor/Treo 等其他运行 Palm 操作系统的 PDA 产品。其显著特点之一是写入装置输入数据的方法，能够通过单击显示器上的图标选择输入的项目。2009 年 2 月 11 日，Palm 公司 CEO Ed Colligan 宣布以后将专注于 WebOS 和 Windows Mobile 的智能设备，将不会再有基于“Palm OS”的智能设备推出，除了 Palm Centro 会在以后和其他运营商合作时继续推出。

4. 黑莓 BlackBerry

BlackBerry 是加拿大 RIM 公司推出的一种移动电子邮件系统终端，其特色是支持推动式电子邮件、手提电话、文字短信、互联网传真、网页浏览及其他无线资讯服务，其最大优势在于收发邮件。在苹果和 Android 的强大市场攻势下，BlackBerry 已经在考虑整体出售。

5. iOS

iOS 作为苹果移动设备 iPhone 和 iPad 的操作系统，在 App Store 的推动之下，成为了世界上引领潮流的操作系统之一。原本这个系统名为“iPhone OS”，直到 2010 年 6 月 7 日 WWDC 大会上宣布改名为“iOS”。iOS 的用户界面的概念基础是能够使用多点触控直接操作。控制



方法包括滑动、轻触开关及按键。与系统交互包括滑动 (Swiping)、轻按 (Tapping)、挤压 (Pinching, 通常用于缩小) 及反向挤压 (Reverse Pinching or Unpinching, 通常用于放大)。此外通过其自带的加速器, 可以令其旋转设备改变其 y 轴以令屏幕改变方向, 这样的设计令 iPhone 更便于使用。

6. Android

Android 是本书的主角, 最早于 2007 年 11 月 5 日发布, 该平台由操作系统、中间件、用户界面和应用软件组成, 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。

根据国际数据公司 (IDC) 公布的数据, 在 2014 年第一季度, Android 和 iOS 系统所占装机量占有所有智能手机出货量的 92.3%, 安装 Android 系统的新智能手机数量跃升至 1.821 亿部, 大大超过去年同期的 9030 万部。在运往世界各地所有的新智能手机中, 谷歌的移动操作系统的市场占有率已经达到 82%, 比 2013 年第一季度的 75% 有显著提高。

截止到本书截稿之时, Android 系统的最新版本是 Android 5.0。

1.2 Android 系统的发展现状

Android 一词最早出现于法国作家利尔亚当 (Auguste Villiers de l'Isle-Adam) 在 1886 年发表的科幻小说《未来夏娃》中, 他将外表像人的机器起名为 Android。本书的主角就是 Android 系统, 在本节将简要介绍 Android 系统的诞生和发展历程。

1.2.1 Android 系统的诞生和发展现状

从 2008 年 HTC 和 Google 联手推出第一台 Android 手机 G1 开始, 在 2011 年第一季度, Android 在全球的市场份额首次超过塞班系统, 跃居全球第一。下面的几条数据能够充分说明 Android 系统的霸主地位。

(1) 2011 年 11 月数据, Android 占据全球智能手机操作系统市场 52.5% 的份额, 中国市场占有率为 58%。2014 年 8 月 15 日消息, 根据 IDC 发布的 2014 年第二季度智能手机市场的最新数据显示, 苹果 iOS 和谷歌 Android 两大系统平台继续领跑。Android 阵营增长则更惊人, 达到了 33.3%, 出货量达到了 2.553 亿台。Android 系统的市场份额得到了提高, 从 2013 年第二季度的 79.6% 增长到了 2014 年第二季度的 84.7%。具体信息如图 1-1 所示。

(2) 如果从某一个时间段进行统计, Android 系统也是雄踞市场占有率第一的位置。据著名互联网流量监测机构 Net Applications 发布的最新数据显示, 从 2013 年 9 月到 2014 年 7 月, 在将近一年的时间里, Android 市场占有率一直处于稳步攀升状态, 从最初的 29.42% 狂飙至 44.62%。

(3) 如果从市场硬件产品出货量方面进行比较, Android 系统则具有压倒性的优势, 其市场份额高达 85%。

由上述统计数据可见, Android 系统的市场占有率位居第一。Android 机型种类数量庞大, 简单易用, 相当自由的系统能让厂商和客户轻松地定制各样的 ROM, 定制各种桌面部件和主

Operating System	Q2 2014 Shipment Volume	Q2 2014 Market Share	Q2 2013 Shipment Volume	Q2 2013 Market Share	Year- Over-Year Growth
Android	255.3	84.7%	191.5	79.6%	33.3%
iOS	35.2	11.7%	31.2	13.0%	12.7%
Windows Phone	7.4	2.5%	8.2	3.4%	-9.4%
BlackBerry	1.5	0.5%	6.7	2.8%	-78.0%
Others	1.9	0.6%	2.9	1.2%	-32.2%
Total	301.3	100.0%	240.5	100.0%	25.3%

图 1-1 2014 年 8 月智能手机平台调查表



题风格。简单而华丽的界面得到了广大用户的认可，对手机进行刷机也是不少 Android 用户所津津乐道的事情。

可惜 Android 版本数量较多，市面上同时存在着 1.6 到 4.4.2 等各种版本的 Android 系统手机，应用软件对各版本系统的兼容性对程序开发人员是一个不小的挑战。同时由于开发门槛低，导致应用数量虽然很多，但是质量参差不齐，甚至出现不少恶意软件，导致一些用户受到损失。同时 Android 没有对各厂商在硬件上进行限制，导致一些用户在低端机型上体验不佳。另一方面，因为 Android 的应用主要使用 Java 语言开发，其运行效率和硬件消耗一直是其他手机用户所诟病的地方。

1.2.2 常见的 Android 设备

因为 Android 系统的免费和开源，也因为系统本身强大的功能，使得 Android 系统不仅被用于手机设备上，而且也被广泛用于其他智能设备中。在接下来的内容中，将简要介绍除了手机产品之外，常见的搭载 Android 系统的智能设备。

(1) Android 智能电视

Android 智能电视，顾名思义是搭载了安卓操作系统的电视。它使得电视智能化，能让电视实现网页浏览、视频电影观看、聊天办公游戏等，实现与平板电脑和智能手机一样的功能。其凭借安卓系统让电视实现智能化的提升，数十万款安卓市场的应用、游戏等内容随意安装。例如，海尔的 MOOKA 模卡 U42H7030 便是一款搭载 Android 4.2 系统的智能电视，如图 1-2 所示。



图 1-2 搭载 Android 4.2 系统的智能电视

(2) Android 机顶盒

Android 机顶盒是指像智能手机一样，具有全开放式平台，搭载了安卓操作系统，可以由用户自行安装和卸载软件、游戏等第三方服务商提供的程序，通过此类程序来不断对电视的功能进行扩充，并可以通过网线、无线网络来实现上网冲浪的的新一代机顶盒总称。

通过使用 Android 机顶盒，可以让电视具有上网、看网络视频、玩游戏、看电子书、听音乐等功能，使电视成为一个低成本的平板电脑。Android 机顶盒不仅仅是一个高清播放器，更具有一种全新的人机交互模式，既区别于计算机、又有别于触摸屏，Android 机顶盒配备红