

让成千上万的人
享受我们开发出来的游戏!



Android

开发关键技术与精彩案例

这不是一本只讲Android开发技术的图书

本书让开发者站在移动互联产业链条上思考自己该怎么做
详解37个常见经典案例, 领悟关键技术、提升实战水平
《最后的防线》作者薛永挖掘人生第一桶金的经验分享



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

詹建飞 等编著

想到做到： Android

开发关键技术与精彩案例

詹建飞 等编著

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书按照入门篇、进阶篇和游戏篇三大部分,结合37个案例系统全面地介绍Android应用和游戏开发的知识。结合实践和设计模式是本书的一大特色,书中深入浅出地讲解了Android平台的运行环境、应用程序模型、用户界面与图形引擎、数据持久化存储方案、移动多媒体框架、Service、联网接口、高级通信技术、访问硬件层能力、游戏引擎和射击类游戏开发等内容。值得一提的是,本书并未局限于技术的讲解,而是结合国内个人开发者基于Android平台创业的成功案例,与读者分享了移动互联网时代如何做好产品和推广取得成功的经验。

本书适合有一定Java编程基础,希望从Symbian、Java ME或者Windows Mobile等平台过渡到Android平台的软件开发人员阅读,也可以作为高校师生的参考教材。

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

想到做到:Android开发关键技术与精彩案例/詹建飞等编著. —北京:电子工业出版社,2012.6
ISBN 978-7-121-17032-4

I. ①想… II. ①詹… 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096491号

策划编辑:胡辛征

责任编辑:李云静

印 刷:

装 订: 中国电影出版社印刷厂

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:25 字数:640千字

印 次:2012年6月第1次印刷

印 数:4000册 定价:79.80元(含光盘1张)

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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前言

移动开发者的机会

据 Canalys 的统计, 2011 年售出的智能手机中, 超过 50% 的智能手机搭载的是 Android 操作系统。苹果公司的 iOS 占的比例达到 20%, 而 Symbian 则大幅度下滑。从智能手机操作系统份额的比例, 我们可以看出, Android 是开发者最值得关注的平台, 前途不可限量。而 Symbian 和 RIM 平台则风光不再。如果您是一名 Symbian 或者 RIM 开发者, 则应该考虑重新选择未来工作的重点。

另一方面, Android Market 的增长速度超乎人们的想象, 截至 2011 年底, 在 Android Market 上架的游戏和应用已经超过了 40 万款, 其中的免费应用占 2/3。免费下载的应用并非赚不到钱, 开发者可以在应用中嵌入广告, 通过广告赚取利润。另外一种付费方式是 IAP (应用内付费)。作为一种新型的商业模式, IAP 的交易量在 2011 年有了显著的增长, 正在成为未来应用程序付费方式的主流。从下载量和营收规模看, 苹果公司的 App Store 是目前最成功的移动应用商店, 但是 Android Market 的快速增长, 也同样预示着广阔的商机, 让我们拭目以待。

智能终端的增长、移动应用商店的成熟、用户对付费应用的逐渐认可等因素为从事移动开发领域工作的开发者提供了前所未有的机会。这个机会不单单指你可以凭借技术在这个领域找到一份待遇不菲的工作, 而且还提供了难得的创业机会。本书正是想帮助读者掌握 Android 应用与游戏开发的知识, 向读者展示成功创业案例, 与读者一起分享成功创业的经验教训。

本书主要内容

结合丰富的案例, 系统全面地介绍 Android 应用和游戏开发的知识是本书的重要目标。为此, 笔者将全书分为以下三大部分。

入门篇 (第 1~8 章)

介绍 Android 平台系统架构和 Android Market 的商务模式; 帮助读者搭建 Android 开发环境, 掌握 Android 应用程序模型, 以及 Activity、BroadcastReceiver、Content Provider 和 Service 主要组件; 从图形用户界面、数据持久化存储、Service, 以及多媒体、电话和短信等方面介绍 Android 应用开发的知识。

进阶篇 (第 9~12 章)

结合俄罗斯方块和手机铃声 DIY 工具两个案例复习前面介绍的 Android 入门知识, 从

联网应用程序开发、Android 3D 应用程序开发、访问摄像头、LBS 和传感器等硬件层等方面进一步讲解 Android 开发的难点和重点。

游戏篇（第 13~15 章）

本篇从游戏引擎入手，结合经典的雷电游戏案例向开发者介绍了 Android 游戏开发的内容，由于篇幅有限，未能涉及游戏开发的方方面面。值得一提的是，这里深入介绍了薛永以游戏为切入点的个人创业案例，分析了游戏产品要取得成功的几个重要因素，相信会对读者有很大的帮助。

关于作者

詹建飞，国内移动开发领域的一名老兵，在 Symbian、Java ME、Android 领域具有多年开发经验。曾先后出版过《Symbian OS J2ME 编程指南》、《Java ME 核心技术与最佳实践》等书籍。目前，重点研究智能终端的发展、移动互联网产品设计和推广等方面的内容。

其他参与本书编写工作的还有田淼、吴博、姚岚、曹斌、薛頔、金镒、李丹、徐志德、嘉园、胡晓琳、程子川、王佳、聂嘉兴、王曦。

由于作者水平有限，书中疏漏和不足之处在所难免，恳请广大读者及专家不吝赐教。

如何使用光盘中的代码

本书的案例代码全部基于 Eclipse 和 Android SDK 2.3 开发完成，读者可以参考附录 A “如何导入源代码”将案例导入到 Eclipse 中。读者在阅读本书的过程中，如果有任何疑问，都可以登录移动开发网（<http://www.doodev.com>）寻求帮助。

詹建飞

2012 年 3 月 5 日

目 录

入 门 篇

第 1 章 开放的 Android	12	2.2.2 运行 Android 项目	27
1.1 Android 的历史	12	2.2.3 修改 Android 项目	29
1.1.1 Google 收购 Android	12	2.3 调试 Android 应用程序	32
1.1.2 Open Handset Alliance	13	2.3.1 设置断点	33
1.1.3 Android 版本演进	13	2.3.2 启动调试	33
1.2 Android 的架构	14	2.3.3 单步跟踪	34
1.2.1 Linux 内核	15	2.3.4 真机调试	34
1.2.2 本地库	15	2.4 常用 Android SDK 工具	36
1.2.3 Android 运行环境	15	2.4.1 Android 模拟器	36
1.2.4 应用程序框架	16	2.4.2 模拟 SD 卡	37
1.2.5 应用程序	16	2.4.3 ADB	37
1.3 开发 Android 应用程序	17	2.4.4 DDMS	38
1.3.1 开发语言	17	2.4.5 Android	39
1.3.2 Android SDK	17	2.5 Android NDK 开发介绍	40
1.4 Android Market 的致富之路	18	2.5.1 什么是 Android NDK	40
1.4.1 Android Market 的商业 模式	18	2.5.2 使用 NDK 的优缺点	40
1.4.2 Android Market 发布流程	18	2.5.3 使用 NDK 的两种方式	40
第 2 章 Android 开发环境和流程	21	第 3 章 Android 应用程序模型	41
2.1 安装 Android SDK 和 Eclipse	21	3.1 Android 应用程序基础	41
2.1.1 安装 Java SDK	21	3.1.1 Android 应用程序运行 环境	41
2.1.2 安装 Eclipse	22	3.1.2 Android 应用程序的组成	42
2.1.3 安装 Android SDK	22	3.2 Activity	47
2.1.4 添加必要的 Android SDK 组件	23	3.2.1 Activity 创建与声明	48
2.1.5 安装 ADT 插件	24	3.2.2 Activity 的生命周期	51
2.2 第一个 Android 应用程序	25	3.2.3 Activity 和 Intent	56
2.2.1 新建 Android 项目	26	3.2.4 使用 Intent 调用系统服务	58
		3.2.5 Activity 和 Task	60
		3.3 Content Provider	64

3.4	BroadcastReceiver	66	4.4.3	绘制图形	103
3.5	Service	67	4.4.4	构建自己的组件	111
3.6	安全与许可	68	4.5	图形动画	116
3.6.1	安全架构	68	4.5.1	Animation	116
3.6.2	许可	68	4.5.2	Interpolator	121
3.7	数字签名	70	4.5.3	AnimationSet	122
3.7.1	签名策略	70	4.5.4	自定义动画	122
3.7.2	签名步骤	70	4.6	Resource 介绍	126
第 4 章	图形用户界面	72	4.6.1	资源类型	126
4.1	用户界面概述	72	4.6.2	使用资源	127
4.1.1	手机软硬件特性的发展	72	4.6.3	资源适配	128
4.1.2	如何影响应用程序开发	73	4.7	App Widgets	130
4.1.3	Android 图形引擎	73	4.7.1	AppWidgetProvider	130
4.2	用户界面设计	74	4.7.2	AppWidgetProviderInfo	131
4.2.1	声明布局文件	74	4.7.3	App Widget 的布局文件	132
4.2.2	编写 XML 文件	74	4.7.4	RemoteViews	132
4.2.3	加载 XML 文件	75	4.7.5	配置 App Widget 的 Activity	133
4.2.4	将数据绑定到 AdapterView	76	第 5 章	数据持久化存储	135
4.2.5	创建菜单	76	5.1	文件存储	135
4.2.6	创建 Dialog	78	5.1.1	存储至默认文件夹	136
4.2.7	通知用户	79	5.1.2	存储至指定文件夹	137
4.2.8	处理用户输入	80	5.1.3	存储至 SD 卡	138
4.2.9	样式与主题	81	5.1.4	读取资源文件	138
4.3	常用 Widget	82	5.2	SharePreference	139
4.3.1	TextView	82	5.3	SQLite	142
4.3.2	CheckedTextView	84	5.3.1	创建数据库	142
4.3.3	Button	85	5.3.2	SQLiteOpenHelper	144
4.3.4	ImageView	86	5.3.3	创建触发器	147
4.3.5	ProgressBar	87	5.3.4	创建索引	147
4.3.6	DatePicker/TimePicker	89	5.3.5	创建视图	148
4.3.7	GridView	90	5.3.6	操作数据	149
4.3.8	Spinner	92	5.3.7	将 Cursor 绑定到 AdapterView	154
4.3.9	Gallery	93	5.4	Content Provider	157
4.3.10	TabHost	94	5.4.1	概述	157
4.4	高级图形用户界面技术	96			
4.4.1	图形系统类结构	96			
4.4.2	常用布局类	98			

5.4.2 创建自定义的 Content Provider	160	7.2 Service 编程实践	196
5.4.3 Content Provider 更新的 通知机制	167	7.2.1 创建 Service	196
5.4.4 DAO 接口及实现	169	7.2.2 启动和停止 Service	197
第 6 章 移动多媒体编程	171	7.2.3 通知用户	199
6.1 多媒体文件格式与编码	171	7.2.4 不阻塞用户操作	200
6.1.1 多媒体文件格式	171	7.2.5 Service 与 Activity 通信	203
6.1.2 编码	172	7.3 后台播放音乐	205
6.1.3 Android 支持的媒体格式	173	7.3.1 跨进程调用	205
6.1.4 选择合适的媒体文件	173	7.3.2 创建 AIDL 接口文件	205
6.2 音频和视频播放	174	7.3.3 实现 AIDL 接口	206
6.2.1 三种不同的数据源	174	7.3.4 绑定 Service	208
6.2.2 MediaPlayer 的状态	177	7.3.5 调用 IPC 方法	209
6.2.3 音乐播放器实例	180	第 8 章 电话和短信	211
6.2.4 播放视频	188	8.1 电话	211
6.3 录制音频	190	8.1.1 电话呼叫	211
6.3.1 MediaRecorder 的状态图	190	8.1.2 监听电话状态	213
6.3.2 录音器实例	191	8.1.3 查询电话属性	215
第 7 章 让程序在后台运行	195	8.1.4 读取通话记录	215
7.1 Service 概述	195	8.2 短消息	217
		8.2.1 发送短信	218
		8.2.2 接收短消息	219

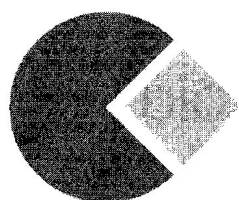
进 阶 篇

第 9 章 Android 进阶案例	224	9.2.5 分割 MP3 文件	246
9.1 俄罗斯方块实例	224	9.2.6 铃声 DIY	248
9.1.1 方块的数据结构	224	9.2.7 设置铃声	249
9.1.2 方块渲染	227	第 10 章 访问网络数据和服务	251
9.1.3 游戏区域设计	228	10.1 HTTP 协议简介	251
9.1.4 碰撞检测	231	10.1.1 HTTP 协议的主要 特点	251
9.1.5 输入处理	232	10.1.2 HTTP 连接过程	252
9.1.6 积分排行榜	234	10.1.3 HTTP 消息格式	252
9.2 手机铃声 DIY 工具	235	10.2 线程管理	254
9.2.1 MP3 文件介绍	235	10.2.1 匿名 Thread	254
9.2.2 ID3V1 标签	236	10.2.2 java.util.concurrent 框架	255
9.2.3 ID3V2 标签	237		
9.2.4 MP3 帧结构	242		

10.2.3	AsyncTask	260	12.1.1	OpenGL ES 设计准则	301
10.3	网络编程接口	263	12.1.2	OpenGL ES 与 OpenGL 的不同	302
10.3.1	HttpClient API 介绍	263	12.1.3	一个简单的 OpenGL ES 程序	305
10.3.2	GET 方法的使用和 限制	266	12.2	3D 观察与变换	307
10.3.3	使用 POST 方法上传 附件	268	12.2.1	一个简单的例子: 绘制 立方体	307
10.3.4	从服务器端下载图片	274	12.2.2	理解变换	311
10.4	设计 C/S 通信数据格式	275	12.2.3	投影变换	313
10.4.1	对象序列化	277	12.2.4	视口变换	315
10.4.2	使用 XML 传输对象	280	12.2.5	矩阵堆栈	316
10.5	Android 数据连接管理	285	12.3	绘制图元	316
10.5.1	查询 APN	286	12.3.1	图元种类	316
10.5.2	查询网络连接状态	286	12.3.2	指定顶点数据	317
10.5.3	监听网络状态变化	288	12.3.3	绘制图元	320
10.5.4	开关 WIFI 和无线连接	288	12.4	颜色和光照	321
第 11 章	访问硬件层	290	12.4.1	设置颜色和材质	321
11.1	访问相机	290	12.4.2	光照	322
11.1.1	创建 Camera 对象	290	12.5	光栅化和片元处理	326
11.1.2	设置 Camera 参数	291	12.5.1	背面剪裁	326
11.1.3	预览	292	12.5.2	纹理映射	326
11.1.4	拍摄照片	292	12.5.3	雾	335
11.2	访问位置服务	294	12.5.4	反走样	336
11.2.1	创建 LocationManager	294	12.5.5	像素测试	338
11.2.2	LocationProvider 类	294	12.5.6	将片元合并到颜色 缓存	341
11.2.3	监听位置更新	295	12.6	帧缓存操作	344
11.2.4	接近警报	296	12.6.1	清空缓存	344
11.3	访问传感器	296	12.6.2	读取颜色缓存	344
11.3.1	SensorManager 类	297	12.6.3	强制完成绘图指令	345
11.3.2	监听传感器事件	297	12.7	其他	345
11.3.3	方向传感器应用 ——指南针	298	12.7.1	行为控制函数	345
第 12 章	使用 OpenGL ES 开发 3D 程序	301	12.7.2	状态查询	345
12.1	OpenGL ES 概述	301	12.8	EGL 简介	346

游 戏 篇

第 13 章 游戏引擎介绍.....	350	14.6 碰撞检测.....	384
13.1 什么是游戏引擎.....	350	14.7 游戏逻辑.....	386
13.2 几种游戏引擎框架.....	350	第 15 章 100 万游戏收入是 怎么炼成的.....	388
13.3 结构化游戏引擎模块.....	351	15.1 创业前的思考.....	388
13.3.1 游戏循环.....	351	15.1.1 创业的成本几乎为零.....	388
13.3.2 图像渲染.....	352	15.1.2 为了实现个人梦想.....	388
13.3.3 声音系统.....	354	15.1.3 市场趋于成熟.....	389
13.3.4 用户交互系统.....	355	15.2 艰辛、刺激的创业.....	389
13.3.5 数据存储.....	357	15.3 产品研发与推广经验.....	391
13.3.6 游戏时间.....	359	15.3.1 游戏策划最重要.....	391
第 14 章 雷电——飞行射击游戏 开发.....	361	15.3.2 首选跨平台技术方案.....	392
14.1 卷屏.....	361	15.3.3 不懈追求性能优化.....	392
14.1.1 什么是卷屏.....	361	15.3.4 灵活的商务模式.....	393
14.1.2 卷屏的原理.....	361	15.3.5 专业推广手段.....	394
14.1.3 卷屏的实现.....	362	附录 A 如何导入源代码.....	395
14.2 设计战机.....	365	附录 B Resin 安装与 Servlet 部署.....	396
14.3 设计子弹.....	376	参考文献.....	399
14.4 设计 BOSS.....	381		
14.5 布局设计.....	383		



入门篇

- 第 1 章 开放的 Android
- 第 2 章 Android 开发环境和流程
- 第 3 章 Android 应用程序模型
- 第 4 章 图形用户界面
- 第 5 章 数据持久化存储
- 第 6 章 移动多媒体编程
- 第 7 章 让程序在后台运行
- 第 8 章 电话和短信

第1章

开放的 Android

1.1 Android 的历史

目前，如果说有一个操作系统能够与苹果公司的 iOS 平分秋色，引领智能终端发展的话，那么，这个操作系统一定是 Google 的 Android。自 2007 年 11 月到现在，Android 平台快速发展，从 Android 1.0 版本演进到最新的 Android 4.0，系统的稳定性和用户体验都有了很大的提升。与此同时，市场上 Android 设备的出货量增长速度更是惊人，每天激活的 Android 设备甚至超过 50 万台，这太不可思议了，但是 Android 确实做到了。现在很多开发商都在追捧 Android，他们希望自己开发的游戏或者应用可以被众多 Android 用户使用，同时为公司创造财富。

1.1.1 Google 收购 Android

事实上，Android 是由 Andy Rubin、Rich Miner 等人在 2003 年 11 月创建的，当时公司设在加利福尼亚州帕洛阿尔托。尽管创始人 Andy Rubin 等人之前已经在业界小有名气，但是 Android 公司的行事却非常低调，他们传递给外界的信息是，Android 是一家从事手机软件开发的，仅此而已。

2005 年 8 月，Google 收购了 Android 公司，Android 成为 Google 的全资子公司，公司的创始人 Andy Rubin、Rich Miner，以及核心员工都留在了 Google，继续从事 Android 的开发工作。收购完成之后，业界已经在猜测，Google 想通过此次收购进入移动电话市场。历史证明，收购 Android 是 Google 最成功的一次交易。

在 Google，Andy Rubin 带领他的团队继续开发基于 Linux 内核的智能终端操作系统。同时，Google 与业界的硬件厂商、软件开发商等展开一系列的合作，目的是将 Android 作为一个稳定的、用户体验出色的操作系统提供给手机制造商和运营商。2006 年，BBC 就曾报道说，Google 希望借助 Android 将其搜索和应用程序延伸到手机上，揭示了这个搜索巨头进军移动互联网的野心。之后，谣言四起，有人说 Google 正在秘密研制自己的手机，

有人说 Google 正在制定技术规范。

📌 1.1.2 Open Handset Alliance

2007 年 11 月 5 日, Android 终于从幕后走向前台, Google 联合高通、HTC、Intel、Motorola、T-Mobile 等发起了 Open Handset Alliance (OHA) 联盟, OHA 的使命是制定移动设备的开放标准。同一天, OHA 发布了第一个项目, 就是基于 Linux 内核的智能终端软件平台 Android。后来, ARM、Sony Ericsson 等公司陆续加入 OHA, 参与到 Android 的研发工作中, 加速了 Android 的发展和市场化进程。

Android 是开源的, 由 Google 领导开发和维护。Google 会定期向开源社区发布 Android 的源代码, 以便任何公司可以使用 Android 制造移动终端设备。Android 的目标就是提供一个稳定的智能终端平台, 提升用户体验。为了避免设备分裂, Google 同时负责 Android Compatibility Program (ACP), ACP 确保使用 Android SDK 和 NDK 开发的应用程序可以在所有的 Android 设备运行, 减小软件开发商移植的成本。Android 兼容性测试可以通过 Google 提供的 Compatibility Test Suite 完成, Google 不强制手机厂商做 CTS 测试, 即便做 CTS 测试也是免费的。

📌 1.1.3 Android 版本演进

截至 2011 年 11 月, Android 在全球智能手机市场的份额达到了 52%, 超过了 Nokia 的 Symbian 操作系统, 让我们一起回顾一下 Android 版本演进的过程。

- 2007 年 11 月, Google 发布了 Android 1.0 beta 版和 Android SDK。但是, 第一个商业 Android 版本是在 2008 年 9 月份发布的, HTC 发布了 G1——第一款搭载 Android 系统的手机, 如图 1-1 所示。
- 2009 年 4 月, Google 发布了 Android 1.5, 基于 Linux 内核 2.6.27。在这个版本中, Google 引入了 Widgets 的概念, 并支持 MPEG-4 和 3GP 多媒体格式。
- 2009 年 9 月, Google 发布了 Android 1.6, 基于 Linux 内核 2.6.29。相比 Android 之前的版本, Google 主要优化了搜索方面的体验, 支持了 WVGA 分辨率。
- 2009 年 11 月, Google 发布了 Android 2.0, 同样基于 Linux 内核 2.6.29, 距离发布 Android 1.6 版本只有两个月的时间。可以看出 Google 一直以一种快速迭代的方式不断修正 Android 的 bug, 提升稳定性, 增加系统功能。Android 2.0 支持了动态壁纸、蓝牙 2.1 等新功能。
- 2010 年 1 月, Google 发布了 Android 2.1, 这个版本只是对上一版本的优化和升级修复了一些 bug。
- 2010 年 5 月, Google 发布了 Android 2.2, 基于 Linux 内核 2.6.32。Google 在 Android 2.2 中引入了很多新的技术, 大幅提升了系统的性能, 其中包括集成了 Chrome 的 V8 JavaScript 引擎, 使用 JIT 技术提升应用程序的执行速度, 以及支持 Adobe Flash 等。
- 2010 年 12 月, Google 发布了 Android 2.3, 基于 Linux 内核 2.6.35。Google 在 Android 2.3 中更新了用户界面接口设计, 用于减少操作步骤, 提升用户体验。值得一提的是, SIP VOIP 和 NFC 在这个版本得到了支持。

- 2011 年 2 月，Google 发布了 Android 3.0，基于 Linux 内核 2.6.36。Android 3.0 是为平板电脑设计的，Motorola 在同年 2 月份，发布了基于 Android 3.0 的平板电脑 Xoom。
- 2011 年 5 月，Google 在 I/O 大会上展示了 Android 4.0 版本，并在 11 月份正式对外发布了 Android 4.0。Android 4.0 是手机和平板电脑融合的展品，拥有 10 大重点新特性，包括更新 Linux 内核到 3.0 版本，支持 Google TV，以及全新的 UI 人性化体验界面等。

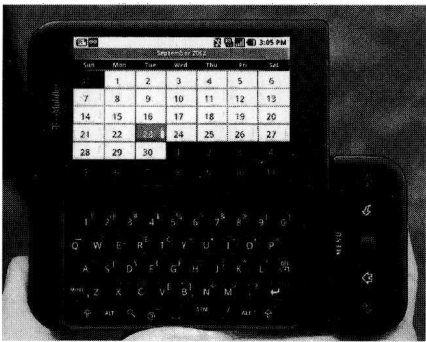


图 1-1 HTC G1

1.2 Android 的架构

Android 由底层操作系统、本地系统库、Android 运行环境、应用程序框架和应用程序等部分组成。Android 平台结构图如图 1-2 所示，本节介绍 Android 架构的主要部分，帮助开发者了解 Android 的体系结构。

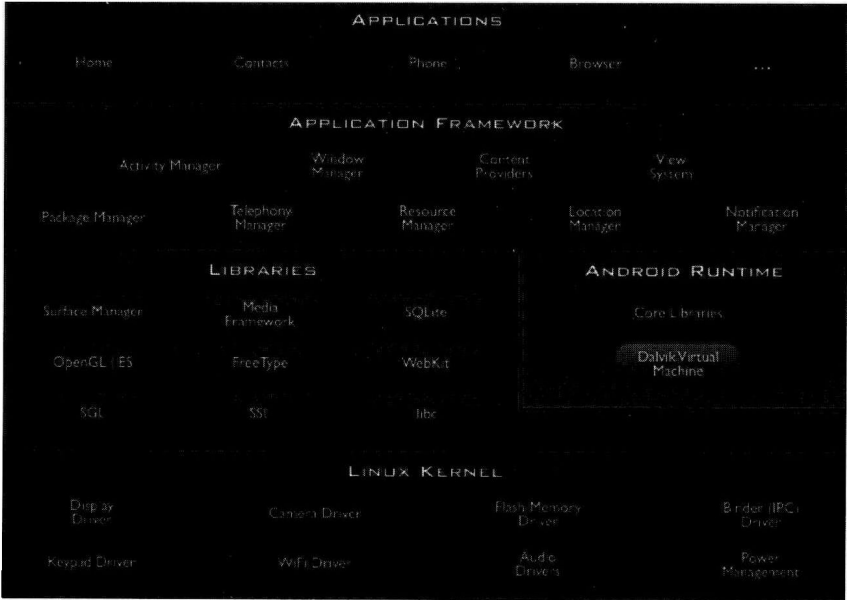


图 1-2 Android 结构图

1.2.1 Linux 内核

Android 最早的版本基于 Linux 2.6 版内核，最新的 Android 4.0 基于 Linux 3.0 版内核。内核为上层系统提供了安全、内存管理、线程管理、网络协议栈和驱动模型等系统服务。同时，内核还提供了一套抽象层接口，在向下的硬件层和向上的软件层之间架起桥梁。

1.2.2 本地库

Android 强大的功能来源于底层的本地库，这些本地库通过上层的应用程序框架将编程接口提供给开发者调用，本地库和应用程序框架通过 JNI（Java Native Interface）连接。以多媒体编程接口为例，应用程序框架层的 MediaPlayer 对象通过 JNI 调用使用 C/C++ 实现的 MediaPlayer 类，MediaPlayer 使用系统提供的接口向上提供服务。调用关系如图 1-3 所示。

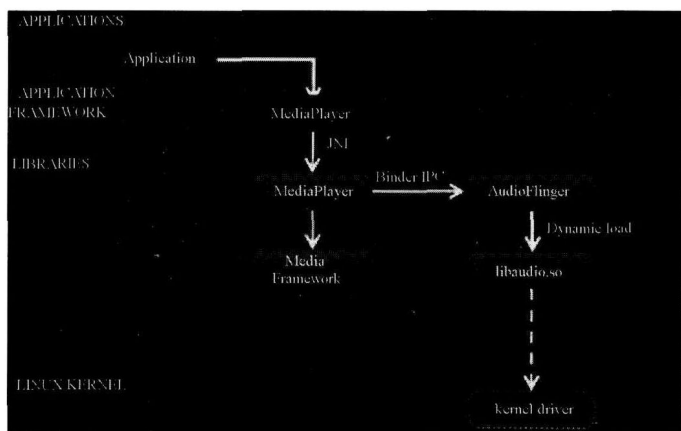


图 1-3 多媒体编程调用关系

下面列出了 Android 包含的核心本地库。

- 标准 C 系统库——针对嵌入式 Linux 设备优化的标准 C 系统库。
- 多媒体库——基于 PacketVideo 的 OpenCORE 引擎，支持多种音频/视频格式的播放和录制，以及静态图片文件的显示。目前支持 MPEG4、H.264、MP3、AAC、AMR、JPEG、PNG 和 GIF 等格式。
- SGL——面向嵌入式终端设备的 2D 图形引擎，在较低的硬件要求下即可提供高质量的图形效果。SGL 引擎基于 C 设计，代码共计 8 万多行，目前已经开源。
- 3D 库——实现了 OpenGL ES 1.0 API，使用 3D 库可以开发出丰富的界面效果。
- SQLite——轻量级的关系型数据库引擎，增强了 Android 的数据存储能力，尤其是存储的数据结构较复杂时。
- Webkit——提供一个浏览器引擎，服务于 Android 的浏览器应用程序和 WebView 组件。
- FreeType——服务于位图和矢量字体的绘制。

1.2.3 Android 运行环境

Android 应用程序以 apk 文件形式发布，apk 文件运行在 Android 运行环境之中，准确

地说是在 Dalvik 虚拟机内运行。Dalvik 虚拟机与普通的 Java 虚拟机不同，它针对嵌入式设备进行了优化，class 文件经过“dx”工具转换成.dex 文件格式，.dex 文件格式更为紧凑，执行效率更高。在 Android 的运行环境中还包含了 Java 核心类库，包括 java.lang、java.io 等包。

每个 Android 应用程序都运行在单独的虚拟机实例之上，这样可以保证应用程序之间不相互影响，即使一个程序崩溃，也不会影响其他程序，因此也不会导致整个系统不能正常运行。由于 Dalvik 虚拟机的设计可以高效地生成多个虚拟机实例，所以不用担心性能问题。关于 Android 的应用程序模型和运行环境将在第 3 章进行详细介绍。

1.2.4 应用程序框架

Android 为应用程序提供了一个开放的运行环境，无论是内置应用程序，还是后续安装的应用程序，所有应用程序访问底层框架的能力是一致的，开发者可以使用应用程序框架提供的 API 开发自己的应用程序。下面简要介绍应用程序框架中包含的常用模块，更详细的介绍将在后续的章节中一一呈现：

- 丰富的图形用户界面组件，包括 ListView、WebView、Button 等。
- 用于在各个应用程序之间共享数据的 Content Provider 机制。
- SQLite 提供的相关数据库操作 API。
- 资源管理器，管理应用程序的文本、图片、XML 等资源。
- 管理应用程序生命周期的 ActivityManager。
- 移动多媒体框架提供了音频/视频播放、音频录制、在线媒体播放等功能。
- 允许应用程序在后台运行的 Service 组件。

1.2.5 应用程序

在 Android 中内置了常用的核心应用程序，包括邮件客户端、电话、短消息、日历、浏览器和联系人等。用户也可以从 Android Market 等下载第三方的应用程序安装到手机上。图 1-4 是 Android 2.3 SDK 模拟器的 Home 和电话应用程序。

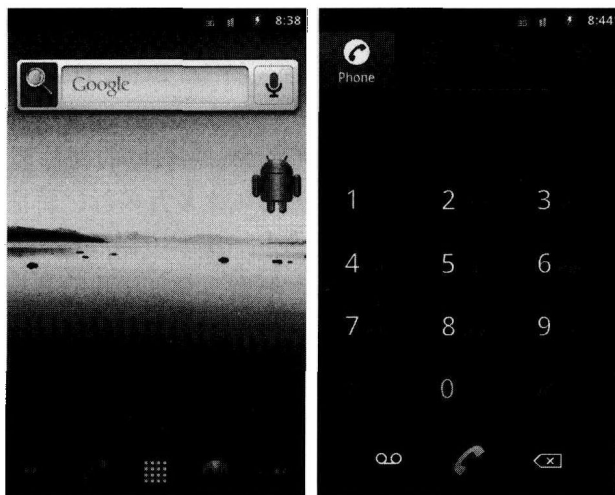


图 1-4 Android 2.3 SDK 模拟器的 Home 和电话应用程序