


移动开发技术丛书

# iOS

# 开发实战体验

© 编著/DevDiv移动开发社区



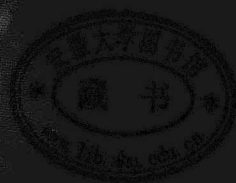
 海军出版社

移动开发技术丛书

# iOS

# 开发实战体验

◎ 编著 / DevDiv移动开发社区



海洋出版社

2012年·北京

## 内 容 简 介

iOS 是移动开发三大平台之一。本书是 DevDiv 移动开发社区版主、资深会员在该平台实际开发经验的总结，通过大量原创示例全面介绍了 iOS 应用开发的方法和技巧。

**主要内容：**全书共 15 章，分别为 iOS 开发框架、Object-C 高级知识、MVC 设计和 Push 消息、视图高级使用技巧、数据持久化、TableView 使用、文件 I/O、硬件和通信、iOS 多媒体、定位和地图、网络编程、连接到互联网、多线程编程、2D 和 3D 绘图编程、调试和优化。

**适用范围：**

- iOS 初中级开发者参考用书
- 高等院校及社会培训机构教材
- 自学人员学习用书

### 图书在版编目(CIP)数据

iOS 开发实战体验/DevDiv 移动开发社区编著. —北京：海洋出版社，2012.8

ISBN 978-7-5027-8311-2

I. ①i… II. ①D… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 146691 号

**总 策 划：**吕允英

**责任编辑：**吕允英

**责任校对：**肖新民

**责任印制：**赵麟苏

**排 版：**海洋计算机图书输出中心 晓阳

**出版发行：**海洋出版社

**地 址：**北京市海淀区大慧寺路 8 号 (716 房间)  
100081

**经 销：**新华书店

**技术支持：**(010) 62100059 hyjccb@sina.com

**发 行 部：**(010) 62174379 (传真) (010) 62132549

(010) 68038093 (邮购) (010) 62100077

**网 址：**www.oceanpress.com.cn

**承 印：**北京画中画印刷有限公司

**版 次：**2012 年 8 月第 1 版

2012 年 8 月第 1 次印刷

**开 本：**787mm×1092mm 1/16

**印 张：**20.25

**字 数：**490 千字

**印 数：**1~3000 册

**定 价：**58.00 元

本书如有印、装质量问题可与发行部调换

## 序 言

苹果 iOS 系统已经历 5 年多 17 个版本的更新，其在多代 iPhone、iPad 和 iPod 设备上取得巨大成功，目前已经成长为市场上影响力最大、功能最丰富、生态最完整的移动操作系统。

iOS 的成功得益于苹果公司对移动互联网的深刻理解，苹果公司始终专注于用户体验与服务，并通过平台整合构建了完整的应用生态链。iOS 从诞生之初即专注用户体验与服务的提升，通过不断的技术革新，一次次引领着 UI 交互方式的变革。比如，Siri 技术就是 iOS5 最大的亮点，实现了语音控制输入功能，可以通过 Siri 使用语音提问和评论，并且可以与包括日历等在内的所有应用通信。同时，新“retina”显示屏像素要比历史版本的显示屏像素高出 3 倍左右，将用户体验提升到前所未有的高度，同时，也拉大了与竞争对手之间的距离！

APP Store 是苹果公司构建应用生态链的关键形式，它让众多的开发者找到了自己的商业模式和商业机会，而苹果公司因此积累了庞大数量的应用。云服务 iCloud 让所有的 iOS 设备实现了互联互通，通过 iCloud，使用同一账号的用户可以在不同 iOS 设备上同步信息和 APP，实现文件备份、存储等功能。可以看出，以 iCloud 为“媒”，统一不同设备系统平台是苹果公司的既定战略，苹果公司在平台整合上已经迈出了坚实的一步。

在 iOS 过去 5 年多时间里，业界对苹果公司的疑问从“这是智能手机吗”发展到了“它可以代替 PC”吗，可见其对智能手机发展的贡献，现在已经没有人怀疑以 iOS 引领的新一代智能手机将成为最重要的个人计算终端、个人娱乐终端和个人通信终端，其地位将与传统 PC 分庭抗礼。苹果公司依靠其强大的垂直一体化战略，不仅通过 iPhone、iPad 等产品赚取了高额利润，同时，通过吸引开发者不断地提供创新的应用，为苹果公司源源不断地注入活力。

此外，值得关注的是，苹果公司并不单纯是“以质取胜”，在数量上，苹果公司也有望在未来几年实现对 Windows 设备（安装微软 Windows 操作系统的设备）的反超。相关统计数据显示：过去，Windows 设备在同苹果设备的销量对比上，一直占据绝对优势，这一优势在 2000 年左右曾经达到过一个峰值，每销售 50 台 Windows 设备才售出一款苹果设备。但是，随着 iPad 和 iPhone 等一系列基于移动互联产品的问世，这一数字比例在急剧减小，截止目前，Windows 设备与苹果设备的销量比例约为 2:1，分析师称，苹果设备数量有望在未来两年内超过微软。

当前，基于 iOS 的应用具有广阔的前景和良好的发展趋势。希望广大开发者借 iOS 之势，利用《iOS 开发实战体验》及书中提供的相关代码，快速进入 iOS 开发领域，实现开发者梦想！

中国科学院博士、博士生导师，软件研究所研究员



2012 年夏

# 前 言

## iOS 平台

iOS 是由苹果公司开发的手持设备操作系统，发布于 2007 年，最初是为 iPhone 设计的，后来陆续套用到 iPod touch、iPad 以及 Apple TV 等苹果产品上。iOS 与苹果的 Mac OS X 操作系统一样，也是以 Darwin 为基础的。原本这个系统名为 iPhone OS，在 2010 年 6 月 WWDC 大会上宣布更名为 iOS。

苹果的 iPhone 发布至今已经接近 5 年，而 iPad 也已发布两年。虽然在智能手机市场的绝对占有率上，iPhone 已经不及谷歌的 Android 系列产品，但若看单一智能手机厂商的出货量，苹果的 iPhone 仍处在第一的位置。如果进一步考虑营业收入和销售利润，苹果公司在智能手机领域的领先优势相当大。若仅以此作为衡量标准的话，所谓的挑战者几乎是不存在的。扩展到平板电脑领域，与智能手机相比，领先优势更是有过之而无不及，50% 以上的市场份额，70%~80% 的营业收入都为苹果公司的 iPad 所占据。

当人们将苹果公司的巨大成功归结于一款款惊艳的产品时，作为行业中人，我们更应该看到其背后的完整生态系统：首先制造出完美的硬件，然后很好地将软硬件进行整合，并在提供网络服务（iTunes Store、Newsstand 和 iCloud 等）的同时，通过自己的 Apple Store 来控制销售。正是如此完善的生态系统，使 iOS 获得移动互联网开发者的广泛认可，成为目前最受欢迎的平台。

DevDiv.com 移动开发社区作为国内最具人气的综合性移动开发社区，继成功推出《移动开发全平台解决方案——Android、iOS、Windows Phone》、《Windows Phone 开发实战体验（应用+游戏）》后，特召集经验丰富的版主和资深会员总结多年 iOS 的工作经验，奉上《iOS 开发实战体验》，希望此书能帮助广大初学者快速进入 iOS 开发领域，以实现大家自己的移动创业梦想。

## 作者

参与本书写作的版主和资深会员具体如下：

作者	社区 ID	简介
陈东严	RealTool	<p>DevDiv 版主</p> <p>毕业于河海大学计算机与信息学院，具有多年的 iOS 应用软件设计与开发经验，同时具有丰富的项目研发和管理经验。曾带领团队完成几十款 iPhone/iPad 的应用开发，包括视频监控、移动 OA 办公、手机订票、工具、社交、团购、房地产行业信息展示、休闲游戏、旅游、电子杂志等应用类型</p> <p>2011 年 4 月发起成立睿拓工作室，专注于 iOS 和 Android 移动平台应用开发与设计，本着“不断追求完美”的做事风格，为移动互联网企业提供客户端开发、资讯、企业内训等业务。工作室官方网站：<a href="http://www.realtool168.com">http://www.realtool168.com</a></p>





作者	社区 ID	简介
张大伟	David_Zhang	<p>DevDiv 版主</p> <p>毕业于安徽理工大学计算机科学与技术系，5 年以上的软件开发经验。擅长 Java 编程语言，对 Java 虚拟机和软件逆向工程有比较深入的研究；熟悉 Python、Scala、JavaScript 等多种脚本语言，曾参与多个大型 J2EE 项目的设计与开发，具有多年的项目研发经验和项目管理经验。现主要从事 Android 和 iOS 两个平台移动互联网应用软件的设计与开发</p>
赵坤	keviniszk	<p>DevDiv 核心会员</p> <p>主要从事终端软件开发，早期开发了基于 Java 的物流、安全等方面的项目，现专注于移动互联网领域，熟悉 iOS、Windows Phone 平台，两年多来，涉及项目涵盖了移动中间件、ERP、商城、订票、监控等多个方面，在项目分析、设计架构和开发方面有非常丰富的经验</p>

此外，本书写作过程中，王兴隆 (Max) 承担了策划及部分沟通、协调工作，周智勋 (BeyondVincent) 协助完成沟通、协调工作并承担了审稿任务。在编辑和审稿阶段，还得到了张金明的帮助，在此表示感谢。

## 学习指南

本书共为 15 章，各章的基本情况如下：

**第 1 章 iOS 开发框架** 主要为基础知识的介绍，重点讲解应用程序生命周期和设计规范，其中参考、翻译了 iOS 框架支持，作者在翻译的基础上做了部分修改并加入了自身的经验总结。

**第 2 章 Object-C 高级知识** 重点是 Object-C 和类别以及静态库的使用方法。静态库使用的比较多，对保护代码版权有很大帮助。本章参考了部分互联网上关于 Object-C 语法介绍的资料，同时加入了作者的经验总结。除了类别部分，其他所有示例均为原创，其中包括静态库的使用等。

**第 3 章 MVC 设计和 Push 消息** 参考了部分官方提供的 MVC 设计和 Push 消息资料，所有示例均为原创。其中 iPhone 手机发送 Push 消息为本章精华，汇集了作者多年实际工作经验，均为原创内容，本部分重点为 Push 消息的实现流程和嵌入式设备发送 Push 消息的实现方法。

**第 4 章 视图高级使用技巧** 为重要的基础知识章节，也是本书的重点章节之一。重点是基础控件定制、动画特效以及页面布局的实现方法。其中，基础控件定制部分为作者多年工作经验总结，示例均为原创；动画特效部分的示例参考了网络相关内容，作者根据本书需要重写了代码。

**第 5 章 数据持久化** 为本书重点章节之一，是实战性的知识，汇聚了作者多年编程经验，所有示例均为原创。重点介绍了 Plist 文件、Sqlite 和 CoreData 的使用。

**第 6 章 TableView 使用** 为本书的重中之重章节，它是界面知识中的重点，内容涵盖 TableView 的组成、样式、定义、数据源、委托、编辑、气泡效果的实现、拖动以显示更多数据等，基本包括了表格视图 90% 的知识，从基础到提高都有涉及，原创性 80% 以上，例子也很经典，非常值得一读。

**第 7 章 文件 I/O** 以一个文件浏览器为例系统介绍了如何对文件系统进行操作，重点给出了文件及文件夹的读取、创建、删除等操作的相关编程方法，并说明了苹果应用文件的存储规则。其中部分为官方文档翻译，其余为作者工作经验总结。



第8章硬件和通信 首先依次介绍了摄像头、加速度计、陀螺仪的功能原理和使用方法,其中重点是陀螺仪的使用方法和具体实现,该部分除了陀螺仪和加速计的原理介绍参考了互联网上的资料之外,其他均为原创内容,有非常详细的示例说明。本章还介绍了通讯录调用、打电话、发短信、发邮件等相关内容,其中通讯录中加入了代码编辑的功能和例子,绝大多数内容为原创。

第9章iOS多媒体 分别介绍了图像处理、声音处理及视频播放的相关内容。其中,图像处理部分重点介绍如何进行图像绘制及GIF效果实现,声音处理部分属于基础性内容,视频处理部分给出了视频的播放方法。本章着重通过两个示例帮大家熟悉上层的API,未对底层复杂API进行介绍,全章60%左右为原创内容。

第10章定位和地图 为iOS开发重点章节,涵盖了定位的实现、地图的基础知识和实现、MKReverseGeocoder地理位置反向编码、地图标注方法、在地图上画线、在地图上长按加地图标注等内容,知识讲解以及代码均为原创内容,尤其是将地图上的各个点连线,网上资料比较少,为作者长期经验总结。

第11章网络编程 介绍了iOS网络编程、NSURLConnection、ASIHTTPRequest、检测网络连接状态等内容,参考了官方文档,原创内容主要是例子,占50%左右。

第12章连接到互联网 从iPhone的内嵌浏览器入手,首先介绍了如何使用SDK提供的UIWebView构建简单浏览器,然后使用官方API进行XML深入解析,并使用第三方解析库编写简单的天气应用,最后介绍了iPhone的JSON及JSON解析库。

第13章多线程编程 为本书重点章节,从各个角度对多线程进行了剖析,知识点非常丰富。多线程知识介绍和例子实现均为原创,其中介绍了3种多线程的实现方法,即传统UNIX下实现多线程、NSThread实现多线程、线程池NSOperationQueue实现多线程,并给出了生产者—消费者模型的多线程示例。

第14章2D和3D绘图编程 内容较为复杂,为开发中的提高篇。本章基础知识部分参考官方文档的同时给出了原创示例。2D部分介绍比较全面,从基础知识到具体操作都有涉及,很实用,而且加入了图片的倒影实现,该部分为原创,也是本章一大亮点。3D部分介绍相对比较少,但也给出了理论知识的大体框架,且实现了一个简单的OpenGL ES例子,相对来说比较实用,而且示例为个人原创。

第15章调试和优化 首先分析了几种常见的错误,然后介绍iOS下的几种调试和优化方法,最后详细介绍了如何使用Instruments中的Leaks工具来查找和定位程序中的内存泄漏。

## 例程代码

本书中的代码均可在<http://www.devdiv.com/forum-218-1.html>免费下载,DevDiv移动开发社区(<http://www.devdiv.com>)负责代码的维护、更新工作。

由于时间仓促以及作者水平有限,书中难免有不足、缺点,甚至错误,如果您在阅读本书中有任何疑问、建议和意见,欢迎大家发信([webmaster@devdiv.com](mailto:webmaster@devdiv.com))或者发帖(<http://www.devdiv.com/rum-218-1.html>)反馈给我们,我们将第一时间为您解答。

DevDiv.com 创始人 吴学友



# 目 录

<b>第 1 章 iOS 开发框架</b> ..... 1	<b>3.1 MVC 框架设计</b> ..... 35
1.1 苹果产品和重要的事件 ..... 1	3.1.1 MVC 设计思想..... 35
1.2 应用商店——App Store ..... 2	3.1.2 iPhone 开发中的 MVC..... 37
1.3 iOS 软件的体系结构 ..... 2	3.1.3 iPhone 中 MVC 的实现..... 37
1.3.1 核心操作系统层 (Core OS) ..... 3	3.2 通知中心 ..... 41
1.3.2 核心服务层 (Core Service) ..... 4	3.2.1 NSNotification 类 ..... 41
1.3.3 媒体层 (Media) ..... 6	3.2.2 Notifications 的常见误解 ..... 42
1.3.4 可轻触层 (Cocoa Touch) ..... 7	3.3 Push 机制 ..... 44
1.4 应用程序运行周期 ..... 8	3.3.1 Push 消息需要的条件 ..... 47
1.4.1 应用程序的生命周期..... 8	3.3.2 在代码中使用 Push 消息 ..... 48
1.4.2 应用程序的入口..... 9	3.3.3 通过 Mac 发送 Push 消息 ..... 48
1.4.3 应用程序的委托..... 9	3.3.4 通过 iPhone 发送 Push 消息 ..... 53
1.4.4 加载主 Nib 文件..... 10	<b>第 4 章 视图高级使用技巧</b> ..... 62
1.4.5 事件处理周期..... 10	4.1 界面工具 Interface Builder ..... 62
1.5 应用程序运行环境 ..... 11	4.2 定制基础控件 ..... 63
1.5.1 应用程序沙箱..... 12	4.2.1 定制 UIButton..... 63
1.5.2 自动休眠定时器..... 13	4.2.2 定制 UIPickerView 以实现隐 藏功能 ..... 67
1.6 iOS 软件设计规范 ..... 13	4.3 动画特效 ..... 72
1.6.1 平台间的差异..... 13	4.3.1 UIViewAnimation 动画 ..... 72
1.6.2 3 种应用程序样式..... 14	4.3.2 使用公有 CATransition 实现 动画效果 ..... 75
1.7 iOS 开发工具——Xcode..... 15	4.3.3 使用私有 CATransition 实现 动画效果 ..... 77
<b>第 2 章 Object-C 高级知识</b> ..... 18	4.4 页面布局——横竖屏处理..... 78
2.1 Object-C 语言介绍..... 18	<b>第 5 章 数据持久化</b> ..... 81
2.1.1 数据类型与表达式..... 18	5.1 Plist 文件操作 ..... 81
2.1.2 流程控制 ..... 19	5.2 UserDefaults 操作 ..... 87
2.1.3 类与结构 ..... 19	5.3 SQLite 数据库操作 ..... 88
2.2 类别 (Category) 介绍..... 21	5.4 Core Data 文件操作 ..... 93
2.2.1 认识类别 (Category) ..... 21	5.4.1 CoreData 特性 ..... 94
2.2.2 扩展 NSString..... 23	5.4.2 为何要使用 Core Data ..... 94
2.2.3 扩展 NSDictionary ..... 25	5.4.3 关于 Core Data 的常见误解..... 94
2.2.4 扩展 NSArray ..... 27	5.4.4 建立数据库模型..... 95
2.2.5 Object-C 与 C++混合编程 ..... 28	
2.2.6 静态库 ..... 30	
<b>第 3 章 MVC 设计和 Push 消息</b> ..... 35	



5.4.5 创建实体类.....	96	8.6 发短信.....	166
5.4.6 数据库操作.....	96	8.7 发邮件.....	168
<b>第 6 章 UITableView 使用.....</b>	<b>102</b>	<b>第 9 章 iOS 多媒体.....</b>	<b>170</b>
6.1 UITableView 的组成及样式.....	102	9.1 图像.....	170
6.2 UITableView 的定义.....	104	9.1.1 加载 UIImage.....	170
6.3 UITableView 的数据源.....	106	9.1.2 UIImageView.....	172
6.3.1 UITableViewDataSource 协		9.1.3 访问照片.....	174
议.....	106	9.2 声音.....	176
6.3.2 表格视图的实现.....	107	9.2.1 System Sound Services.....	176
6.3.3 表格单元.....	110	9.2.2 音频.....	179
6.3.4 创建表格单元的数据源.....	111	9.3 视频.....	183
6.4 UITableView 的委托.....	115	<b>第 10 章 定位和地图.....</b>	<b>188</b>
6.5 UITableView 的编辑.....	116	10.1 基础知识.....	188
6.6 UITableView 实现气泡效果的表格.....	120	10.2 iPhone 定位方法.....	189
6.7 UITableView 拖动以显示更多数据.....	128	10.3 MKReverseGeocoder 地理位置反	
<b>第 7 章 文件 I/O.....</b>	<b>133</b>	向编码.....	194
7.1 文件系统.....	133	10.4 LBS 应用的类型.....	195
7.2 文件管理.....	134	10.5 谷歌地图.....	196
7.2.1 读取并显示对应目录下的文		10.5.1 在地图上增加大头针标注	
件.....	135	的方法.....	198
7.2.2 获取文件属性信息.....	138	10.5.2 在地图上画线.....	202
7.2.3 创建文件夹.....	141	<b>第 11 章 网络编程.....</b>	<b>208</b>
7.2.4 创建文件.....	146	11.1 iOS 网络编程.....	208
7.2.5 删除文件.....	149	11.1.1 NSURLConnection.....	208
7.3 本地数据存储规则.....	150	11.1.2 网络编程示例.....	210
<b>第 8 章 硬件和通信.....</b>	<b>151</b>	11.2 ASIHTTPRequest.....	214
8.1 摄像头.....	151	11.2.1 使用 ASIHTTPRequest.....	215
8.1.1 拍照.....	151	11.2.2 ASIHTTPRequest 使用示	
8.1.2 摄像.....	155	例.....	218
8.1.3 定制拍照界面.....	157	11.3 检查网络状态.....	224
8.2 加速度计.....	158	11.3.1 SCNetworkReachability.....	224
8.2.1 加速度计原理.....	158	11.3.2 Reachability.....	225
8.2.2 加速度计使用.....	158	<b>第 12 章 连接到互联网.....</b>	<b>229</b>
8.3 陀螺仪.....	159	12.1 使用 UIWebView.....	229
8.3.1 陀螺仪原理.....	160	12.2 解析 XML.....	233
8.3.2 陀螺仪使用.....	161	12.2.1 iOS 下的 XML 解析库.....	234
8.4 调用通讯录.....	162	12.2.2 NSXMLParser.....	234
8.4.1 读取通讯录.....	162	12.2.3 第三方解析器.....	237
8.4.2 编辑通讯录.....	164	12.2.4 编写简单天气解析应用.....	238
8.5 打电话.....	166	12.3 解析 JSON.....	245





12.3.1	iPhone 的 JSON .....	245	14.1.4	图形状态 .....	269
12.3.2	JSON 解析库 .....	247	14.1.5	Quartz 2D 坐标系统 .....	270
<b>第 13 章</b>	<b>多线程编程 .....</b>	<b>248</b>	14.1.6	内存管理 .....	271
13.1	UNIX 多线程机制的使用 .....	248	14.1.7	绘制图形图像 .....	271
13.2	NSThread 创建多线程的方法 .....	251	14.1.8	绘制 OpenGL 效果的倒影 .....	287
13.2.1	线程的创建与启动 .....	251	14.2	3D 绘图 OpenGL ES .....	289
13.2.2	线程的同步与锁 .....	252	14.2.1	OpenGL 与 OpenGL ES 简介 .....	290
13.2.3	线程的交互和其他控制方法 .....	255	14.2.2	OpenGL ES 在 iPhone 绘图中的应用 .....	290
13.2.4	线程的睡眠 .....	257	<b>第 15 章</b>	<b>调试和优化 .....</b>	<b>302</b>
13.3	线程池 NSOperationQueue .....	258	15.1	常见错误 .....	302
13.3.1	创建线程操作 NSOperation .....	258	15.1.1	版本错误 .....	302
13.3.2	任务控制 .....	259	15.1.2	证书错误 .....	303
13.4	生产者—消费者模型 .....	262	15.1.3	编写错误 .....	304
13.4.1	使用@synchronized .....	262	15.1.4	导入错误 .....	305
13.4.2	使用 NSLocking 协议 .....	264	15.2	调试跟踪 .....	308
<b>第 14 章</b>	<b>2D 和 3D 绘图编程 .....</b>	<b>267</b>	15.2.1	使用调试器 .....	308
14.1	Quartz 2D .....	267	15.2.2	使用日志 .....	310
14.1.1	画布 (Canvas) .....	267	15.3	使用 Instruments .....	311
14.1.2	绘图上下文 (Graphics Context) .....	268			
14.1.3	Quartz 2D 数据类型 .....	268			

# 第 1 章 iOS 开发框架

自从 2007 年 iPhone 第一代手机发布以来，苹果公司改变了世界。当年乔布斯在产品发布会上宣讲时曾说：苹果重新发明了手机！现在看来，确实如此。苹果为科技世界重新树立了标杆，它将技术与艺术完美结合，到目前为止，没有哪一款电子产品能像 iPhone 那样让消费者疯狂。你可以想象，那些顶着寒风在苹果专卖店门口排队七八个小时，甚至通宵的人们，仅仅是为了购买一款手机！

随着 iPhone 在国内的热销，国内的 iOS 开发也掀起了一轮高潮。如果此时的你正打算从事 iOS 开发，那么恭喜你，因为你选择了在很长一段时间内可能都是最吸引人的平台。苹果公司用一种极端追求完美的精神给世界带来了神奇的产品，作为开发者，我们也应当秉承这种精神，自己的努力和热情，开发出完美的应用软件。下面，就让我们一起开始 iOS 编程之旅吧。

## 1.1 苹果产品和重要的事件

在学习 iOS 开发之前，我们有必要先来了解这家世界级的科技公司。苹果公司自创建以来，为世界带来了许多划时代的产品，也发生了许多影响深远的事件，按时间顺序列举如下：

- ① 1977 年，发售最早的个人计算机 Apple II。
- ② 1982 年，推出 Apple Lisa 计算机。
- ③ 1984 年，推出革命性的 Macintosh 计算机。
- ④ 1997 年，推出彩色的 iMac 计算机。
- ⑤ 2001 年，推出 iPod 数码音乐随身听。
- ⑥ 2005 年，史蒂夫·乔布斯宣布下一年度将采用英特尔处理器。
- ⑦ 2007 年，史蒂夫·乔布斯在 Mac World 上发布了 iPhone 与 iPod touch。
- ⑧ 2008 年，史蒂夫·乔布斯在 Mac World 上发布了 MacBook Air。
- ⑨ 2008 年，史蒂夫·乔布斯在 Mac World 上发布了新设计的 MacBook 和 MacBook Pro，以及全新的 24 英寸 Apple LED Cinema Display。
- ⑩ 2008 年 7 月 11 日，苹果公司推出 3G iPhone，iOS 2x 版正式提供全球语言。
- ⑪ 2010 年 1 月 27 日，史蒂夫·乔布斯在苹果公司发布会上发布 iPad 平板电脑。
- ⑫ 2010 年 6 月 8 日凌晨 1 点，在苹果全球开发者大会（WWDC 10）上，史蒂夫·乔布斯发布了全新的 iPhone 第四代手机，型号为 iPhone 4。
- ⑬ 2011 年 3 月 2 日推出 iPad2 系列产品。
- ⑭ 2011 年 6 月 7 日介绍 iOS5、Mac OS X 10.7 Lion、iCloud。
- ⑮ 2011 年 10 月 5 日推出 iPhone4S、iOS5、iCloud。
- ⑯ 2012 年 3 月 7 日，推出 iPad 第三代产品，命名为 The New iPad。

目前，这些产品有些已经淡出了人们的视线，有些还在热销中，而且创造了一个又一个销售奇迹。苹果现在的产品线如图 1-1 所示。

从图 1-1 中我们可以看到，苹果的产品线非常简洁，这是苹果一贯的作风，把精力集中在最少





的事情上面，做到专注。



图 1-1 苹果现有的产品线

## 1.2 应用商店——App Store

App Store 即 Application Store，通常理解为应用商店。App Store 是一个由苹果公司为 iPhone 和 iPod Touch、iPad 以及 Mac 创建的服务，允许用户从 iTunes Store 或 Mac App Store 浏览和下载一些用 iOS SDK 或 Mac 开发的应用程序。用户可以购买或免费试用，使该应用程序可直接下载到 iPhone 或 iPod touch、iPad、Mac 计算机上。这些软件应用包含游戏、工具、图书、图库，以及许多实用的软件。App Store 从 iPhone、iPod touch、iPad 以及 Mac 的应用程序商店都是相同的名称。

App Store 模式的意义在于为第三方软件提供者提供了方便而又高效的软件销售平台，使得第三方软件提供者参与其中的积极性空前高涨，同时适应了手机用户们对个性化软件的需求，手机软件业由此进入高速、良性发展的轨道。可以说，是苹果公司把 App Store 这样的一个商业行为升华到了能够令人效仿的经营模式。App Store 无疑会成为手机软件业发展史上的一个重要的里程碑，其意义已远远超越了“iPhone 的软件应用商店”本身。

读者掌握了本书后面讲解的 iPhone 开发技能后，可以将自己的应用程序上传到 App Store 平台上，供全世界的用户下载使用，这也是大部分开发者开发软件的最终目的。

## 1.3 iOS 软件的体系结构

对于 iPhone 开发的基本常识，可以很方便地从互联网获得，在此不进行详细说明。这里重点要介绍的是 iOS 的体系结构，如图 1-2 所示。

可以看出，按从低到高的顺序，大致可分为以下 4 层：

- Core OS——核心操作系统层。

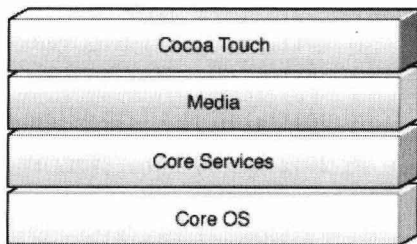


图 1-2 iOS 的体系结构



- Core Services——核心服务层。
- Media——媒体层。
- Cocoa Touch——可轻触层，也叫应用层。

### 1.3.1 核心操作系统层 (Core OS)

Core OS 位于最底层，iOS 的许多技术都是基于 Core OS 的。在应用程序中，开发者虽然不会直接用到 Core OS 提供的功能，但是所引用的其他库很有可能会用到或者依赖于该层。开发者在处理安全问题或者与硬件进行沟通的时候，会直接用到 Core OS 提供的功能。Core OS 层模块如图 1-3 所示。

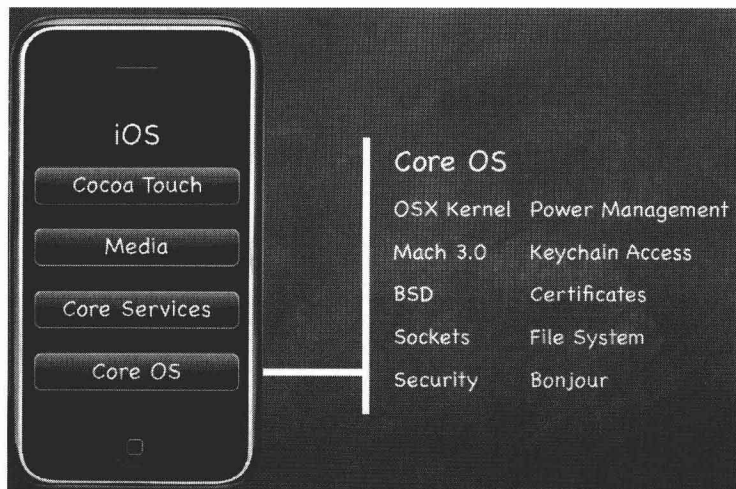


图 1-3 Core OS 层模块

Core OS 层所包含的库如下：

#### (1) Accelerate Framework

包含数学计算、大号码以及数字信号处理等一系列接口。使用这个库的好处在于开发者可以对其进行重写，用以优化基于 iOS 的不同设备的硬件特征，并且只需要写一次就可以确保它在全部设备上有效运行。

#### (2) External Accessory Framework

负责 iOS 设备与各种附属设备的沟通。iOS 设备可以通过自带的数据线、Wi-Fi 和蓝牙与附属设备进行沟通。External Accessory Framework 提供接口让开发者可以获取各种附属设备的信息并且进行初始化，从而可以放心地发送指令以控制设备。

#### (3) Security Framework

用来保证应用程序数据管理的安全性。这个库会提供一些接口让开发者管理证书、公共密钥、私有密钥和信任策略。它支持安全加密随机数生成，同时也支持密钥的证书存储。

#### (4) System 层

包括内核、驱动和 OS 的各种 UNIX 底层接口。其中内核是基于 Mach（用于 Mac OS X 的微内核）的，它主要负责处理虚拟内存管理、线程、文件系统、网络和进程间通信。驱动主要用来为各种硬件和系统库之间提供接口。iOS 提供接口让应用程序可以访问 OS 的各种功能，开发者可以通过 LibSystem 库来调用它们，这些接口都是基于 C 语言的，它们提供的功能如下：



- ① 线程。
- ② 网络。
- ③ 文件系统。
- ④ 标准输入输出。
- ⑤ Bonjour 和 DNS 服务。
- ⑥ 本地信息。
- ⑦ 内存管理。
- ⑧ 数学计算。

★提示 Bonjour 也称为零配置联网，能自动发现 IP 网络上的计算机、设备和服务。Bonjour 使用工业标准的 IP 协议来允许设备自动发现彼此，而不需输入 IP 地址或配置 DNS 服务器。

### 1.3.2 核心服务层 (Core Service)

Core Service 由两个部分组成：核心服务库和基于核心服务的高级功能。

#### 1) 核心服务层模块

核心服务层模块如图 1-4 所示。

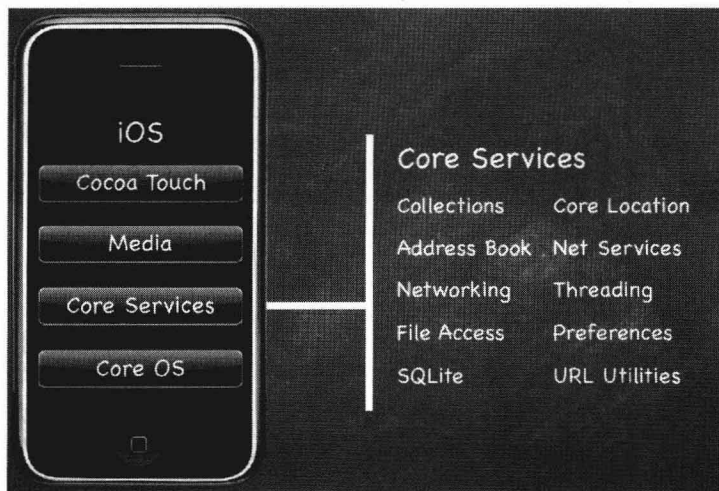


图 1-4 核心服务层模块

核心服务层所包含的库如下：

#### (1) Address Book Framework

用于地址簿管理，开发者可以通过这个库管理电话联系人列表，访问联系人列表的数据库，然后进行添加、删除和编辑等其他操作。

#### (2) CFNetwork Framework

提供了一系列的接口，可以让开发者更方便、快捷地进行网络沟通。开发者无需关注过多的细节，可以花更多的精力在应用程序上面。

#### (3) Core Data Framework

用于管理基于 MVC 模式 (Model 模型、View 视图、Controller 控制器) 应用程序的数据模型，它是一个关系数据管理系统。Core Data 提供了在存储器中保存、管理、更改以及获取数据等基础功能。



#### (4) Core Foundation Framework

它是一个 C 语言库，为应用程序提供了各种基础设施。

#### (5) Core Location Framework

提供定位的功能，可以运用 GPS、3G 和 Wi-Fi 来侦测用户的位置，地图应用程序可以使用这个功能来显示用户在地图上的位置。开发者可以将其加入到应用程序中实现相关的功能，如根据位置提供附近餐馆、商店以及银行等的搜索服务。

#### (6) Core Media Framework

提供比较底层的媒体处理，通常很少用到这个库。

#### (7) Core Telephony Framework

提供接口帮助用户收集电话商的服务信息，例如，用户可以知道自己用的是哪个电话商的服务，知道自己的设备现在是不是在打电话。

#### (8) Event Kit Framework

可以让用户在自己的设备上访问日历事件，用它来获取现有的日历事件或者添加一个新的事件，例如进行闹钟控制。

#### (9) Foundation Framework

提供的功能和 Core Foundation Framework 提供的功能差不多，区别在于它是 Objective-C 库。

#### (10) Mobile Core Services Framework

为 UTI (Uniform Type Identifiers, 统一类型的标识符) 定义了比较底层的数据类型。

#### (11) Quick Look Framework

可以让用户对文件的内容进行预览。

#### (12) Store Kit Framework

为应用程序与 App Store (应用程序商店) 之间的通信提供服务，应用程序可以通过该库从 App Store 接收那些用户需要的产品信息，并显示出来供用户购买。当用户需要购买某件产品时，程序调用 Store Kit 来收集购买信息。

#### (13) System Configuration Framework

通过该库可以让用户决定设备的网络配置，例如，是否使用 Wi-Fi 连接或者是否连接某个网络服务。

### 2) 基于核心服务的高级功能

#### (1) Block Objects

C 语言构造体，开发者可以将它插入到 C 代码或者 Objective-C 代码中。从本质上来说，一个 Block Objects 就是一个封闭函数，或者说是伴随这个函数的数据。一般来说，Block Objects 可以运用到下面几种情形：

- ① 代替代理和代理方法。
- ② 代替回调函数。
- ③ 与分发堆栈一起实现异步工作。

#### (2) Grand Central Dispatch

简称 GCD，它可以根据处理器的数量调整应用程序的工作负荷，而且只会使用任务所需数量的线程，从而提高应用程序的效率。例如，在不使用 GCD 时，如果一个应用程序在最大负载时需要 20 条线程，那么即使在空载时，它也会建立 20 条线程，并占用相关资源。而使用 GCD 时则不





然，GCD 会释放闲置资源，以加快整个系统的响应速度。

### (3) In App Purchase

基于 Store Kit Framework 的高级功能，通过这个功能用户可以让自己的应用程序很好地处理账号、App Store 与应用程序之间的关系。

### (4) Location Services

基于 Core Location Framework 的服务功能，可以让应用程序给用户定位，查找用户当前位置。

### (5) SQLite (嵌入式数据库)

可以让开发者在应用程序里面嵌入一个轻量级的 SQL 数据库，无需建立一个分开的数据库服务器，就可以在应用程序里面创建一个 Database 文件，然后进行列表和记录的管理。

### (6) XML Support

可以让开发者对 XML 文件进行解析。

## 1.3.3 媒体层 (Media)

媒体层主要提供图像渲染、音频播放和视频播放的功能，具体的模块如图 1-5 所示。

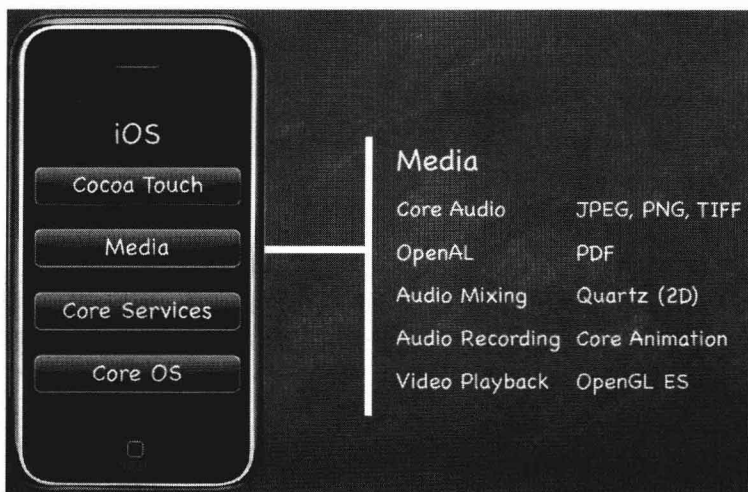


图 1-5 媒体层模块

### 1) 图像渲染功能

实现图像渲染功能的方法如下：

- ① 用 Core Graphics 进行图像的 2D 渲染。
- ② 用 Core Animation 提供动画效果。
- ③ 用 OpenGL ES 提供对 2D 和 3D 渲染的支持，并支持硬件加速。
- ④ Core Text 提供了流畅的字体渲染引擎。
- ⑤ Image I/O 提供接口用以各种格式图像的读写。
- ⑥ Assets Library Framework 用以对用户照片库里的照片和视频进行访问。

### 2) 音频播放功能

媒体层的 Audio 模块提供了以下功能：

- ① Media Player Framework 可以让用户方便访问 iTunes 的最新版本库并且支持列表播放。





- ② AV Foundation 提供了简单易用的接口用以管理音频回放和记录。
- ③ OpenAL 提供跨平台的接口支持音频播放。
- ④ Core Audio Frameworks 提供了一系列简单的接口，开发者可以通过这些接口进行音频播放和记录，也可以用它们来播放系统声音，同时管理本地音频文件或者音频流的多通道缓冲和回放。

### 3) 视频播放功能

媒体层的 Video 模块提供了以下功能：

- ① Media Player Framework 为开发者提供了一系列简单易用的接口，开发者可以在应用程序里面调用这些接口来进行视频播放。
- ② AV Foundation 用以管理视频的抓频和回放。
- ③ Core Media 提供底层的的服务，供上面的功能调用。

## 1.3.4 可轻触层 (Cocoa Touch)

作为整个 iOS 的最顶层，这一层是最为核心的部分，负责屏幕上的多点触摸事件处理、文字输出、图片网页显示、相机或文件的存取以及加速感应等，如图 1-6 所示。



图 1-6 可轻触层模块

可轻触层所包含的库如下：

#### (1) Address Book UI Framework

用来显示 Address Book 数据库中联系人的数据。这个基于 C 的框架可以提供一个图形界面来访问用户的联系人信息，同时用户也可以创建新的联系人，删除或者编辑已存在的联系人。

#### (2) Event Kit UI Framework

基于 Event Kit Framework，它主要为查看和编辑事件提供视图控制器。

#### (3) Game Kit Framework

为游戏应用程序提供点对点的网络连接和语音通信来支持对战游戏。这个框架支持所有的应用程序，并且无需配对。

#### (4) iAd Framework

用于播放广告，这样可以为应用程序带来额外的收入。

