

技术新颖，基于最新iOS 6技术撰写，系统讲解开发iPhone和iPad应用所需掌握的理论知识和高级技巧

从认知学角度进行内容规划和撰写，一个综合案例贯穿全书，能大幅降低学习的时间成本，并提高学习效率



刘铭 朱舸 著

iOS 6 in Action

iOS 6

应用开发实战



机械工业出版社
China Machine Press



iOS 6 in Action

iOS 6 应用开发实战

刘铭 朱舸 著



机械工业出版社
China Machine Press

图书在版编目(CIP)数据

iOS 6应用开发实战 / 刘铭, 朱舸著. —北京: 机械工业出版社, 2013.4
(实战系列)

ISBN 978-7-111-41914-3

I. i… II. ①刘… ②朱… III. 移动电话机—应用程序—程序设计 IV.TN929.53

中国版本图书馆CIP数据核字(2013)第057846号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书是目前iOS 6领域最全面系统和易于阅读的著作之一, 有两大特点: 第一, 技术新颖, 基于最新iOS 6技术撰写, 系统讲解开发iPhone和iPad应用所需掌握的基础技术和高级技巧, 以及其流程和方法; 第二, 易于阅读, 从认知学角度进行内容规划, 一个案例贯穿全书, 不仅能从很大程度上降低学习的时间成本, 降低阅读门槛, 而且能至始至终让读者在动手实践中保持学习的热情, 坚持把这本书读完。

全书共22章, 可分为两个部分: 基础部分(1~13章)分别介绍了开发iOS应用前应该做的准备工作、Xcode 4的基本使用、Objective-C的基本语法、Interface Builder的基本操作、视图控制器、通过设备获取用户位置、列表、视图及视图控制器、各种控件、多媒体、偏好设置等知识, 这些知识通过一个名称为MyDiary的应用完美地串联在一起, 读者从一开始就能动手实践并从实践中掌握这些理论知识; 高级部分(14~22章)则非常详细地讲解了iOS应用的架构、iOS应用的测试与调试、可滚动视图的创建、自动宣传和自动调整大小、表格视图的编辑、手势识别、警告、应用程序本地化、日历和事件等高级话题, 是iOS开发工程师进阶修炼必须掌握的核心内容。

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 姜 影

北京京师印务有限公司印刷

2013年5月第1版第1次印刷

186mm×240mm·25.5印张

标准书号: ISBN 978-7-111-41914-3

定 价: 69.00元

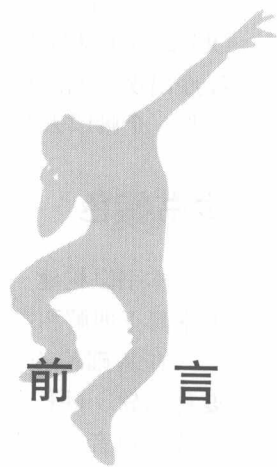
凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsjj@hzbook.com



前言

为什么要写这本书

我在开始学习 iOS 开发之前一直从事着 PHP 和 Flex 的应用程序开发，因为四年前为苹果移动设备开发应用程序至少在国内毕竟还不是很热门。当时的互联网开发大多还是围绕着 Web 2.0 理念的开发以及各种社交网站上的 Flash 交互游戏开发。记得一次在回家的路上，我静静地思考这样一个问题：从现在开始，我是否要放弃多年驾轻就熟的语言，转而去一个全新的开发领域呢？挑战肯定是有的，但是挑战与机遇共存。就目前情况而言，iOS 程序员的数量远远低于传统 C、C++ 的人数。不仅如此，在笔者参加过的多次 Adobe 开发者大会上，与会人员的数量和那些人的热情是大家完全可以想象的（尽管笔者还没有真正看到过一款使用 Flash 或 Flex 开发的商业应用）。众所周知，虽然目前 iOS 设备在国内大量普及（不管是个人，还是机关、单位），但是真正符合国人需求的应用并不是很多，尤其是符合某一领域的商用、垂直化应用还远远不够，这就为 iOS 开发团队带来了巨大的机遇。因此在经过一段时间的学习和开发实践以后，便有了写这本书的想法，主要想通过本书让更多希望从事 iOS 开发的程序员尽快入门，同时为将来的创业打下良好的基础。

2012 年 9 月 19 日，苹果第六代手机操作系统 iOS 6.0 正式发布，iOS 6 SDK (Software Development Kit) 也随之放出，该 SDK 为我们进行应用程序开发提供了更多的功能和特性。其中包括全新的苹果地图，Siri 发送短信、预约会议，与朋友分享照片流，Passbook 优惠券处理和 iPhone 全新呼叫等功能。如果说从 iOS 1.0 到 iOS 4.0 苹果更多的是做系统深度开发，那么从 iOS 5.0 到 iOS 6.0 的时代就是广度的开发，与此同时操作系统的容量也在不断的变大。

截至目前，苹果对于 iOS 操作系统的更新和升级还在继续，这也就代表着我们对 Objective-C 和 iOS 开发的学习始终不会停止。新的 iOS 硬件设备可能会带动更多更好的应用程序出现，而这些应用程序的背后将是一个巨大的财富。希望大家一切从用户的角度出发，一步一个脚印去做，最终实现自己的梦想。

本书特色

本书以构建一个 MyDiary 项目的实践案例贯穿全书，将所有知识点融入到实践当中，使大家真正理解和掌握如何通过 Xcode SDK 和 Objective-C 语言来开发 iOS 应用程序。

在基础篇的学习中，可以了解到什么是 iOS，什么是 Xcode。作为一名 iOS 开发者需要什么软、硬件条件。然后就是如何通过 Xcode 和 Objective-C 来搭建一个能够记录文本、图片、声音，查看设备当前位置的 MyDiary 项目。还结合该项目讲授了如何使用表格来组织信息。

在高级篇中，大部分的内容都是针对之前所学内容的深入和扩展。包括如何组织表格，如何处理设备的旋转，多语言和地区的处理以及如何操作日历等。

读者对象

本书适合具备以下几方面知识和硬件条件的群体阅读。

- ☐ 面向对象的开发经验，熟悉类、实例、方法、封装、继承、重写等概念。
- ☐ 有 Objective-C 或 C、C++ 的开发经验。
- ☐ 有 MVC 设计模式开发经验。
- ☐ 有简单的图像处理的经验。
- ☐ 有一台 Inter 架构的 Mac 电脑（MacBook Pro、MacBook Air、Mac Pro 或 Mac Mini）。
- ☐ 如果加入了 iOS 开发者计划，还可以准备一台 iOS 移动设备。

如何阅读本书

本书逻辑上分为两个部分：

第 1 章到第 13 章为基础部分，通过构建一个 MyDiary iPhone 应用程序项目，向大家介绍了如何使用 Xcode SDK 开发工具及通过 Objective-C 语言编写一个可以记录文本、保存图片 and 声音的 App。

第 14 章到第 22 章为高级部分，着重讲解 iOS 开发中可能会用到的高级功能，包括视图的旋转和滚动处理、表格的高级操作、多语言环境等。

本书自始至终都通过 MyDiary iPhone 应用程序这个实际项目来展开的，所以建议初学者从第 1 章开始学习，以达到良好的学习效果。

勘误和支持

由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或不准确的地方，恳请读者批评指正。书中的全部源文件可以从华章网站^①下载。如果你有更多的宝贵意见，也欢迎发送邮件至邮箱 liuming_cn@qq.com，期待能够得到你们真挚反馈。

致谢

首先要感谢伟大的可以改变这个世界的 Steven Jobs，他的精神对我产生了非常大的影响。

感谢机械工业出版社华章公司的编辑杨福川老师，在这一年多的时间中始终支持我的写作，他的鼓励和帮助引导我顺利完成全部书稿。

最后感谢我的爸爸、妈妈、老婆、乐乐、张燕、赵霞、秦琼、王艳标、杨晓龙、刘天翔、梁涛，感谢他们对我的支持与帮助，并时时刻刻为我灌输着信心和力量！

谨以此书献给我最亲爱的家人，以及众多热爱 iOS 的朋友们！

刘 铭

① 参见华章网站 www.hzbook.com。——编辑注



目 录

前 言

第 1 章 开发前的准备 / 1

- 1.1 iOS 的历史 / 2
- 1.2 iOS 5 和 iOS 6 的新特性 / 3
 - 1.2.1 iCloud 云服务 / 3
 - 1.2.2 LLVM 3.0 编译器 / 3
 - 1.2.3 自动引用计数器 / 3
 - 1.2.4 故事板 / 4
 - 1.2.5 整合 Twitter 框架 / 4
 - 1.2.6 iOS 5 的其他特性 / 4
 - 1.2.7 iOS 6 的新特性 / 6
- 1.3 iOS 设备的相关介绍 / 7
 - 1.3.1 各种 iOS 设备 / 7
 - 1.3.2 iOS 设备的显示分辨率 / 8
 - 1.3.3 iOS 应用程序与硬件的关系 / 9
 - 1.3.4 iPhone 的网络连接 / 9
 - 1.3.5 iPhone 的输入与反馈机制 / 10
- 1.4 成为一名 iOS 开发者 / 10
 - 1.4.1 注册成为 Apple Developer / 11
 - 1.4.2 加入 iOS 开发者计划 / 13

1.5 下载并安装 iOS 开发工具 / 14

1.6 本书实战项目简介 / 16

第 2 章 认识 Xcode 4 / 17

2.1 Xcode 的欢迎界面 / 18

2.2 使用模板创建 MyDiary 项目 / 19

2.3 Xcode 的工作界面 / 21

2.3.1 Xcode 工具栏 / 21

2.3.2 导航区域 / 22

2.3.3 编辑区域 / 25

2.3.4 通用区域 / 26

2.4 使用代码编辑器 / 28

2.5 iOS 模拟器 / 30

2.5.1 iOS 模拟器的特性 / 31

2.5.2 模拟器中 iOS 系统的基本设置 / 32

2.5.3 在模拟器中安装和卸载应用程序 / 33

2.5.4 iOS 模拟器的限制 / 34

2.6 管理 MyDiary 项目的配置选项 / 34

2.6.1 设置设备的支持方向 / 35

2.6.2 设置应用程序的图标 / 36

2.6.3 设置显示状态栏 / 37

2.6.4 设置应用程序的启动画面 / 38

第 3 章 有关 Objective-C 的知识储备 / 40

3.1 内存管理基础 / 41

3.1.1 C 语言中的内存管理 / 42

3.1.2 面向对象的内存管理 / 42

3.2 引用计数器 / 43

3.2.1 使用引用计数器 / 44

3.2.2 使用 autorelease / 48

3.3 类的 setter 与 getter 方法 / 51

3.4 dealloc 方法 / 54

3.5 使用 properties 简化访问器 / 54

3.6 手动内存管理的规则 / 56

3.7 自动引用计数器 / 57

3.7.1 强引用 / 59

3.7.2 弱引用 / 60

第4章 Xcode 中的 Interface Builder / 61

4.1 了解 Interface Builder / 62

4.1.1 Interface Builder 中的关联 / 62

4.1.2 故事板 / 63

4.2 创建用户界面 / 66

4.2.1 对象库 / 66

4.2.2 向视图添加界面元素 / 67

4.2.3 Interface Builder 的布局工具 / 68

4.3 自定义界面的外观 / 71

4.3.1 属性检查窗口 / 71

4.3.2 设置辅助功能 / 72

4.3.3 在模拟器中使用辅助功能 / 73

4.4 与代码进行关联 / 74

4.4.1 要完成的效果 / 74

4.4.2 outlet 变量和 action 方法简介 / 75

4.4.3 为 ViewController 类添加成员变量 / 75

4.4.4 创建 outlet 关联 / 76

4.4.5 创建 action 关联 / 77

4.4.6 使用快速检查器查看关联 / 78

第5章 视图控制器 / 80

5.1 MVC 设计模式简介 / 81

5.2 MyDiary 项目中的“关于作者”控制器 / 82

5.2.1 为“关于作者”控制器准备照片素材 / 83

5.2.2 设置 Segmented Control / 84

5.2.3 为控制器准备相关数据 / 85

5.2.4 UIImage 的类方法介绍 / 87

5.3 视图控制器的重构 / 88

5.4 标签栏控制器 / 90

5.4.1 标签栏和其中的标签 / 90

5.4.2 在故事板中添加标签栏控制器 / 91

5.4.3 设置标签栏配置条目 / 94

5.5 与视图控制器相关的方法介绍 / 96

第 6 章 通过设备获取用户位置 / 99

- 6.1 项目、目标和框架 / 100
- 6.2 Core Location 简介 / 101
- 6.3 创建 LocationViewController 控制器 / 103
- 6.4 Core Location 框架 / 105
 - 6.4.1 从 CLLocationManager 获取信息 / 107
 - 6.4.2 在程序中确认定位服务是否可用 / 109
 - 6.4.3 CLLocation 类 / 111
- 6.5 委托 / 112
 - 6.5.1 协议 / 112
 - 6.5.2 协议方法 / 114
 - 6.5.3 委托、控制器和内存管理 / 115
- 6.6 使用 MapKit / 116
 - 6.6.1 高德地图 iOS API / 116
 - 6.6.2 使用高德 iOS API 显示地图 / 116
 - 6.6.3 在真机上测试高德地图 / 121

第 7 章 创建日记列表 / 123

- 7.1 表格视图的组成部分 / 124
- 7.2 准备要显示的数据 / 124
 - 7.2.1 向 Diary 类的成员变量赋值 / 125
 - 7.2.2 对象的分配和初始化 / 126
 - 7.2.3 在故事板中添加表格视图 / 128
 - 7.2.4 Arrays 类 / 131
- 7.3 UITableViewController 类 / 133
 - 7.3.1 剖析 DiaryListViewController / 134
 - 7.3.2 UITableView 的数据源 / 134
- 7.4 UITableViewCell 类 / 135
 - 7.4.1 创建与检索单元格 / 137
 - 7.4.2 UITableViewCell 的复用 / 140
- 7.5 增加点击交互功能 / 141

第 8 章 通过导航控制器显示多个视图 / 144

- 8.1 使用导航控制器进行视图间的导航 / 145

- 8.1.1 导航控制器介绍 / 146
- 8.1.2 创建导航控制器 / 147
- 8.1.3 导航栏 / 151
- 8.2 增加日记的详细页面 / 155
- 8.3 使用 UINavigationController 进行导航 / 159
 - 8.3.1 推出视图控制器 / 159
 - 8.3.2 在视图控制器间传递数据 / 160

第 9 章 为日记添加文本记录功能 / 162

- 9.1 创建新的视图控制器 / 163
 - 9.1.1 创建 CreateDiaryViewController 类 / 163
 - 9.1.2 创建 Create Diary View Controller 场景 / 164
 - 9.1.3 建立 IBOutlet 和 IBAction 关联 / 168
 - 9.1.4 在应用程序中呈现和销毁 CreateDiaryViewController / 170
- 9.2 几种常用的文本控件 / 175
 - 9.2.1 UILabel 控件 / 175
 - 9.2.2 UITextField 控件 / 176
 - 9.2.3 UITextView 对象 / 177
- 9.3 虚拟键盘的使用 / 178
- 9.4 NSDate 及其相关类 / 180
- 9.5 日记记录传回 DiaryListViewController / 181

第 10 章 为日记本添加照相功能 / 184

- 10.1 为项目添加新的视图控制器 / 185
 - 10.1.1 创建 CameraViewController 类 / 185
 - 10.1.2 CameraViewController 的呈现与销毁 / 187
- 10.2 使用 UIImagePickerControllerController 进行拍照 / 188
- 10.3 在应用程序中存储图片 / 192
 - 10.3.1 NSDictionary 类 / 193
 - 10.3.2 创建和使用键 / 195
 - 10.3.3 在 Detail Diary View Controller 场景中显示日记照片 / 198

第 11 章 保存与载入日记 / 200

- 11.1 应用程序沙箱 / 201
- 11.2 创建单例模式 DiaryStore 类 / 202

- 11.3 获取指定目录的路径 / 204
- 11.4 归档 / 205
 - 11.4.1 对象的归档 / 205
 - 11.4.2 对象的解档 / 206
- 11.5 应用程序的状态与过渡 / 209
- 11.6 使用 NSData 将数据写入文件系统 / 212
- 11.7 在 Mac 系统中查看应用程序的资源 / 214

第 12 章 在日记中录制和播放声音 / 216

- 12.1 创建录音机的界面 / 218
- 12.2 声音的录制 / 220
 - 12.2.1 保存录制的音频文件 / 220
 - 12.2.2 完善 RecordViewController 控制器 / 224
 - 12.2.3 按照用户的要求进行录音 / 226
- 12.3 声音的播放 / 227
- 12.4 中断的处理 / 229
 - 12.4.1 在播放声音时处理中断 / 229
 - 12.4.2 在录制声音时处理中断 / 230
- 12.5 在 Detail Diary View Controller 场景中播放声音 / 231

第 13 章 应用程序的偏好设置 / 234

- 13.1 了解设置绑定资源包 / 235
- 13.2 为 MyDiary 添加设置绑定资源包 / 235
 - 13.2.1 向项目中添加设置绑定资源包 / 236
 - 13.2.2 设置属性列表 / 236
 - 13.2.3 添加文本框设置 / 239
 - 13.2.4 添加密码文本框设置 / 241
 - 13.2.5 添加多值字段 / 241
 - 13.2.6 添加开关设置 / 243
 - 13.2.7 添加滑块设置 / 243
 - 13.2.8 设置滑块图标 / 243
 - 13.2.9 增加设置子视图 / 245
- 13.3 在应用程序中读取偏好设置内容 / 247

第 14 章 iOS 应用程序架构介绍 / 250

- 14.1 Objective-C 和 Cocoa Touch / 251

- 14.1.1 Cocoa Touch 层 / 251
- 14.1.2 Foundation 框架 / 252
- 14.1.3 iOS 中的各种框架 / 252
- 14.2 iPhone 应用程序的架构 / 253
 - 14.2.1 main.m 文件 / 254
 - 14.2.2 自动释放池 / 254
 - 14.2.3 UIApplicationMain() 函数 / 255
 - 14.2.4 应用程序委托 / 256
 - 14.2.5 视图控制器 / 257
- 14.3 iOS 应用程序的组成 / 258
 - 14.3.1 应用程序文件夹的结构 / 258
 - 14.3.2 可执行文件 / 258
 - 14.3.3 Info.plist 文件 / 258
 - 14.3.4 程序图标和启动画面 / 259
 - 14.3.5 XIB 文件 / 259
 - 14.3.6 IPA 文档 / 259
- 14.4 平台的限制 / 260

第 15 章 应用程序的调试 / 262

- 15.1 iOS 帮助文档 / 263
 - 15.1.1 快速安装帮助文档 / 264
 - 15.1.2 帮助文档的窗口 / 265
 - 15.1.3 类的文档页面介绍 / 268
- 15.2 代码调试器 / 269
 - 15.2.1 断点的设置 / 270
 - 15.2.2 调试代码 / 272
 - 15.2.3 调试控制台 / 274
 - 15.2.4 检视对象 / 276

第 16 章 创建可滚动的视图 / 277

- 16.1 UIScrollView 类 / 278
- 16.2 Scroll View 和 Text Field / 283
 - 16.2.1 完善 Scroll View Controller 中的界面元素 / 283
 - 16.2.2 编写与滚动相关的代码 / 284
- 16.3 消息通知中心 / 288

16.4 完善 CreateDiaryViewController 控制器 / 290

16.4.1 使用 Scroll View / 290

16.4.2 在 Text View 中让键盘消失 / 293

第 17 章 自动旋转和自动调整大小 / 296

17.1 自动旋转的机制 / 297

17.1.1 点、像素和视网膜显示 / 298

17.1.2 旋转的方式 / 298

17.2 通过 autosize 属性处理旋转 / 298

17.2.1 创建一个新的视图控制器 / 299

17.2.2 配置所支持的方向 / 299

17.2.3 在控制器中设定支持方向 / 300

17.2.4 使用 autosize 属性设计界面 / 303

17.2.5 设置按钮的 autosize 属性 / 305

17.3 旋转时重构视图 / 306

17.3.1 创建和关联 Outlet / 307

17.3.2 旋转时移动按钮 / 307

17.4 旋转时切换视图 / 309

17.4.1 设计两个视图 / 309

17.4.2 执行旋转时的切换 / 312

17.4.3 改变 Outlet Collections / 314

第 18 章 编辑表格视图 / 315

18.1 表格视图的编辑模式 / 316

18.2 使用不同类型的附件指示器 / 317

18.3 创建自定义的单元格附件指示器 / 319

18.4 显示具有层级的表格视图 / 321

18.5 为表格创建 Header 和 Footer / 322

18.5.1 创建头视图 / 323

18.5.2 载入 XIB 文件中的视图 / 324

18.5.3 设置头视图 / 325

18.5.4 设置脚视图 / 326

18.6 删除行 / 328

18.7 移动表格视图中的行 / 329

18.8 在表格视图中使用刷新控件 / 331

第 19 章 手势识别 / 333

- 19.1 多点触摸概述 / 334
 - 19.1.1 基本的操作手势 / 335
 - 19.1.2 手势识别的分类和状态 / 335
 - 19.1.3 触摸事件所响应的方法 / 336
- 19.2 拖曳手势的检测 / 336
 - 19.2.1 简单的手势识别 / 337
 - 19.2.2 UIPanGestureRecognizer 类 / 339
- 19.3 轻划手势的检测 / 340
- 19.4 旋转手势的检测 / 342
- 19.5 长按手势的检测 / 344
- 19.6 掐捏手势的检测 / 345

第 20 章 警告用户 / 348

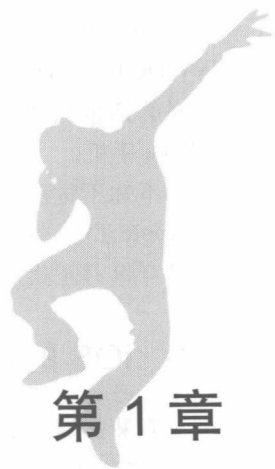
- 20.1 通过警告对话框与用户进行交互 / 349
 - 20.1.1 构建简单的 Alert 警告对话框 / 349
 - 20.1.2 设置 Alert 的 delegate / 353
 - 20.1.3 显示警告对话框 / 354
 - 20.1.4 警告的类型 / 354
- 20.2 在警告对话框中呈现菜单 / 356

第 21 章 应用程序的本地化 / 359

- 21.1 使用 NSLocale 将项目国际化 / 360
- 21.2 本地化资源 / 366
 - 21.2.1 本地化资源文件 / 366
 - 21.2.2 使用 NSLocalizedString 和 String Tables / 369

第 22 章 日历和事件 / 373

- 22.1 Event Kit 框架 / 374
 - 22.1.1 通过 CalDAV 同步 Google 日历 / 374
 - 22.1.2 获取日历列表 / 375
 - 22.1.3 向日历中添加事件 / 380
 - 22.1.4 访问日历的事件列表 / 382
 - 22.1.5 从日历中移除事件 / 386
 - 22.1.6 添加重复发生的事件 / 389
- 22.2 Event Kit UI 框架 / 392



第 1 章

开发前的准备

本章内容

- 1.1 iOS 的历史
- 1.2 iOS 5 和 iOS 6 的新特性
- 1.3 iOS 设备的相关介绍
- 1.4 成为一名 iOS 开发者
- 1.5 下载并安装 iOS 开发工具
- 1.6 本书实战项目简介

2007 年，乔布斯站在苹果电脑全球研发者大会（Apple Worldwide Developers Conference, WWDC）上向世界宣布：运行在 iPhone 手机上面的操作系统至少领先业内同行五年。我想，这样的话也只能出自乔布斯这样的疯狂人之口。要知道那一年苹果才刚刚发布其第一代的 iPhone 手机。

不知不觉已经过去了六个春秋，基于 iOS 操作系统的设备除了 iPhone 以外，还有 iPad 和 Apple TV。这期间，苹果的操作系统也经历了 5 次重大的升级，最新操作系统 iOS 6 也于 2012 年 9 月正式发行。

1.1 iOS 的历史

苹果公司在 2007 年发布了第一代 iPhone 后，宣布 iPhone 将运行精简版 OS X。那时 iOS 的名称为 iPhone OS，因为当时还没有 iPad。iPhone OS 1.0 最终在同年 6 月 29 日正式发布，预装了邮件、iPod、日历、照片、短信、Safari、备忘录、YouTube、计算器、地图、设置、照相机、股票和电话等应用。iPhone OS 1.0 并不包含 App Store 或 iTunes 商店。

那时，乔布斯鼓励开发者为 iPhone 开发网页应用而不是原生应用。几个月之后，苹果改变了主意，并在 2008 年 3 月发布了供第三方开发人员使用的公共 SDK（Software Development Kit，软件开发工具）。从这以后，每当苹果发布一个更新版本的操作系统，都会包含很多新的特性并带来 API 的改变。现在 App Store 已经拥有超过 50 万个的应用。

iPhone OS 2.0 最大的功能就是加入了 App Store。这样，用户就可以使用第三方应用了。App Store 的出现帮助苹果轻松战胜了竞争对手，使 iPhone OS 变成了一款万能的系统。随后苹果开始使用“*There's App for that*”广告语，标榜任何事情都可以通过应用实现。

iPhone OS 2.0 还加入了邮件推送功能。为了让屏幕容下更多的应用，苹果加入了主屏翻页功能。其他主要功能包括：可以打开微软 Office 文档，可以截屏，可以将 Safari 中的图片储存至照片，等等。

从 iPhone OS 3.0 开始，Mac 中的 Core Data 功能被带到了 iPhone 之中，除此以外，还包括消息推送服务、外部设备工具包、通过 StoreKit 框架实现的应用程序内部购买功能、程序内电子邮件、允许开发者在程序中内置 Google 地图的 MapKit 框架、读取 iPod 媒体库等。iPhone OS 3.1 则增加了视频编辑支持，这只是一个小幅度的升级。iPhone OS 3.2 增加了 Core Text 和手势识别、文件共享和 PDF 生成支持，这些也属于次要更新。另外，iPhone OS 3.2 增加了对新产品 iPad 的支持。要开发基于 iPad 或通用（Universal）的应用程序，就需要使用 iPhone OS 3.2。开发者要清楚的是，iPhone OS 3.2 只适用于 iPad，不能安装在 iPhone 或 iPod Touch 上。

从 iPhone OS 4.0 开始，苹果将操作系统的名称改为 iOS 4，包括了众望所归的多任务支持特性、本地通知、通过 EventKit 框架只读访问用户日程、应用程序内短信息支持和视网膜