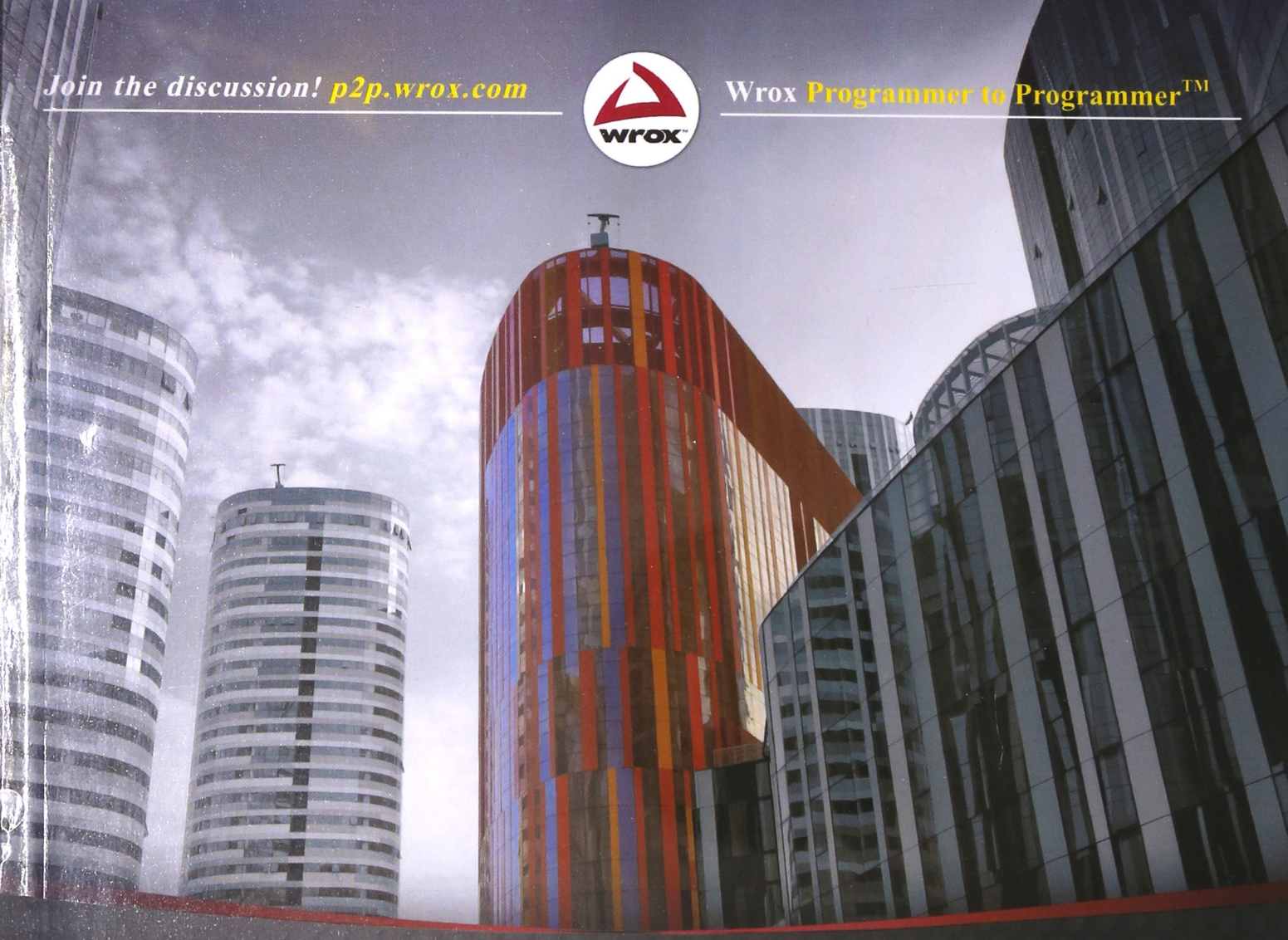

Mac OS X and iOS Internals: To the Apple's Core

深入解析Mac OS X & iOS操作系统

[美] Jonathan Levin 著
郑思遥 房佩慈 译

清华大学出版社

014031923

TP316.84
67

深入解析 Mac OS X & iOS 操作系统

[美] Jonathan Levin 著

郑思遥 房佩慈 译

北京航空航天大学
藏书
图书馆

清华大学出版社

北 京

北航 C1720011

TP316.84
67

Jonathan Levin
Mac OS X and iOS Internals: To the Apple's Core
EISBN：978-1-118-05765-0
Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc.授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字：01-2013-2565

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.
本书封面贴有 Wiley 公司防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

深入解析 Mac OS X & iOS 操作系统 /(美) 莱文(Levin, J.) 著；郑思遥，房佩慈 译.
—北京：清华大学出版社，2014

书名原文：Mac OS X and iOS Internals: To the Apple's Core

ISBN 978-7-302-34867-2

Ⅰ. ①深… Ⅱ. ①莱… ②郑… ③房… Ⅲ. ①操作系统 Ⅳ. ①TP316.84

中国版本图书馆 CIP 数据核字(2014)第 025702 号

责任编辑：王　军　刘伟琴
装帧设计：牛艳敏
责任校对：邱晓玉
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　印　张：46.25　　字　数：1241 千字
版　　次：2014 年 3 月第 1 版　　印　次：2014 年 3 月第1 次印刷
印　　数：1～4000
定　　价：108.00 元

产品编号：043731-01

译 者 序

根据网络分析公司 Net Applications 的数据，截至 2013 年 10 月，OS X 的市场占有率已经达到 7.73%，而且仍在稳步增长。OS X 不仅占据了各种高端专业领域，在服务器领域中也能见到其身影。在移动平台上，iOS(包括手机和平板电脑)的市场占有率具有绝对优势，达到了 55.39%。排名第二的是种类繁多的 Android 操作系统，占有率刚过 30%。各种数据表明，苹果系的操作系统——OS X 和 iOS——正变得越来越重要。虽然市场占有率主要得益于苹果公司对生态圈建设所做的努力，但是其操作系统的核心技术也是重要的幕后功臣。我作为操作系统的爱好者及研究者，一直遗憾图书市场缺乏深入介绍苹果操作系统内核的书籍，而介绍其他操作系统的书籍，特别是 Linux 的书籍，则汗牛充栋。甚至连闭源的 Windows 都有不少经典好书。与苹果操作系统有关的著作主要是介绍各种 BSD 的书籍和学术论文，以及早期的一些来自于卡内基梅隆大学的与 Mach 内核相关的学术著作。当然还有一本不得不提的书籍是 Amit Singh 的经典著作 *Mac OS X Internals: A Systems Approach*，该书同样也有些年头了，主要介绍的是 PowerPC 平台的 OS X 操作系统，当然也不会涉及 iOS。而现在这本《深入解析 Mac OS X & iOS 操作系统》则很好地填补了这个遗憾。

本书不是一开始就讲解内核，而是从现象出发，首先从“超级用户”的角度来讲解苹果的内核提供的各种功能，以及有自己特色的地方。然后再进入内核，从 Mach 和 BSD 的角度分别讲解内核中各个子系统的实现原理。讲解内核的时候，基本上以各个子系统提供的 API 和数据结构为脉络，全面而深入地涵盖内核实现的各种细节。

本书不仅涉及开源 XNU 核心的内容，还涉及不少关于 iOS 的闭源 XNU 核心的内容，这也是本书的一大特色。由于 iOS 的核心是闭源的，所以本书多采用逆向工程的方法，对汇编代码进行分析，顺便介绍了各种逆向工程方法在越狱中的应用，使读者可以了解神秘的越狱过程。此外，书中还有各种和苹果操作系统开发或越狱相关的八卦趣闻，因此本书也是一本有趣的书。

本书不是操作系统的教材，因此没有介绍操作系统的基本原理。阅读本书要求读者具备操作系统工作原理方面的相关知识，例如操作系统的基本架构、硬件的引导过程、进程及其调度的基本原理、虚拟内存和内存管理的基本原理、文件 I/O 以及网络 I/O 的基本原理。读者最好也熟悉苹果的操作系统，了解 Mac OS X 和 iOS 的基本操作和特性。此外，为了能读懂书中的示例代码，还要求读者熟悉 C 语言编程，最好也了解一些汇编语言的知识。

翻译本书旨在为传播知识贡献一点绵薄之力，译文中如有不当之处，敬请广大读者批评指正。

郑思遥

2013 年 10 月于北京

作 者 简 介

Jonathan Levin是一位经验丰富的技术培训师和咨询师，他的关注点是“三大系统”(Windows、Linux和Mac OS)以及它们的移动版本(Android和iOS)的内部工作原理。15 年来，Jonathan坚持传播内核工程和修改技术的真知灼见，在DefCON以及其他技术会议上发表了很多技术演讲。他是Technologeeks.com公司的创始人和首席技术官(CTO)，这是由一些志趣相投的专家合伙创办的公司，致力于通过技术培训传播知识，通过咨询解决棘手的技术难题。他们的专业领域覆盖软件架构中的实时及其他关键部分、系统/内核级编程、调试、逆向工程以及性能优化。

技术编辑简介

Arie Haenel 是 NDS Ltd.(现属于 Cisco)的一位安全和底层专家。Haenel 先生在整个数据安全和设备安全领域都有着丰富的经验。他拥有以色列耶路撒冷理工学院计算机科学系的科学工程学士学位，还拥有法国普瓦捷大学的 MBA 学位。他的兴趣包括学习犹太法典、柔道以及解谜语。他居住在以色列耶路撒冷。

Dwight Spivey 是好几本 Mac 相关著作的作者，包括 *OS X Mountain Lion Portable Genius* 和 *OS X Lion Portable Genius*。他还是 Konica Minolta 的产品经理，在那里他专门负责 Mac 操作系统、应用程序及硬件，还负责彩色和单色激光打印机。他在 Konica Minolta 教授 Mac 使用方面的课程，编写培训和技术支持材料，还是苹果开发者计划的成员。Dwight 和他漂亮的妻子 Cindy 及 4 个可爱的孩子 Victoria、Devyn、Emi 和 Reid 居住在阿拉巴马州的格尔夫海岸。他在越来越少的空闲时间会学习神学，画连环漫画，还会支持 Auburn Tigers 棒球队。

此为试读，需要完整PDF请访问：www.ertongbook.com

致　谢

“你知道吗，Johnny”，我的朋友 Yoav 在上海的一个温暖的夏夜边吐着烟边对我说，“写本书吧！”

于是我就开始写这本书了。最初是 Yoav(Yobo) Chernitz 点燃了我写书的热情，这是一个改变，多年来我只读别人写的书。从那时起，在远东、中东还有美国东部(以及之间无数的航班上)，想法开始生根发芽，本书开始成型。我还不知道这本书会变得如此庞大，不知何时这本书开始有了自己的生命，让我付出很大的努力才能完成这本书。经历了无数次预料之外的复杂和延迟，真难以相信这本书完成了。我尝试覆盖这庞杂知识结构中最晦涩难懂的部分，描述这些知识，不留任何死角。下面应该由读者来评判我是否做到了。不过要知道，没有下面这些人的帮助我是不可能完成这本书的：

我的挚友 Arie Haenel——天生的黑客，绝不要小聪明。总是给我最犀利的批判，绝对是技术评审的最佳人选。

Moshe Kravchik——作为本书的第一位读者提出了深刻的洞见和具有挑战性的问题，为后来的读者带来了一本可读性更好的书。

Yuval Navon——远在南半球的澳大利亚墨尔本，让我理解了好朋友是不会受到地域限制的。

最后，但绝对是同等重要的，我要感谢我亲爱的 Amy，她的耐心，她对我四处奔波的忍耐，以及无尽的理解支持我坚持到最后，她用无穷的智慧不断地提醒我，交稿的截止日期固然很重要，但对读者负责同样重要。

——Jonathan Levin

前　言

尽管 OS X 已经诞生了十几年，但讨论 OS X 架构的书籍却少之又少，讨论 iOS 架构的书更是几乎没有。尽管关于 Objective-C、框架和 OS X 的 Cocoa API 的文档非常多，但是这些文档的讨论往往不够深入，缺少系统调用层次和实现细节。尽管也有一些关于内核的文档(大部分都是 Apple 公司提供的)，但也同样只关注驱动程序的构建(利用 I/O Kit)，只展示了一些优美的部分，而对于 XNU 的基础 Mach 核心却几乎没有任何涉及。XNU 是开放源代码的，但是尽管如此，却有着一百多万行的源代码(和注释)，有一些源代码甚至可以追溯到 1987 年，读起来绝对不是一件轻松愉快的事情。

而对于其他操作系统却并非如此。Linux也完全是一个开源的操作系统，但是从来不缺乏相关的书籍，O'Reilly就有很多非常棒的系列。Windows尽管是闭源的，但是Microsoft却提供了非常好的文档(其源码也在一些场合开放了)。本书对于XNU的意义，就好像Bovet和Cesati的*Understanding the Linux Kernel*对于Linux的意义，以及 Russinovich的*Windows Internals*对于Windows的意义。这两本书都是很棒的书，非常清晰地阐述了这些异常复杂的操作系统的架构。幸运的是，您正在读的这本书会用同样的方式讲解Apple的操作系统的内部工作原理。

其实之前有过一本关于 Mac OS 的书，这就是 Amit Singh 的优秀著作 *MAC OS X Internals: A Systems Approach*，这是一本很棒的参考书，提供了大量有价值的信息。遗憾的是，该书针对的是 PowerPC 架构，而且在 Tiger 之后(2006 年左右)就没有任何更新了。从那时到现在，6 年过去了。在这漫长的 6 年里，OS X 抛弃了 PowerPC 架构，完全移植到了 Intel 平台，而且已经经过了 4 个大版本的迭代。经历了 Leopard(美洲豹)、Snow Leopard(雪豹)、Lion(狮子)和最新的 Mountain Lion(山狮)，野生猫科动物的家族正在扩大，越来越多的新特性被加入操作系统中。不仅如此，OS X 还经历了一次全新的移植。这一次移植的目标是 ARM 架构，改头换面成为了 iOS(根据某些统计资料，它是全世界领先的移动环境操作系统)。因此本书在前辈的基础上狗尾续貂，讨论 Apple 生态系统中新加入的猫科动物，还讨论了一些版本的 iOS。

Apple 的操作系统被认为是在不断地演进。本书最早是针对 iOS 5 和 Lion 编写的，但是这两个操作系统都在持续进化。在本书英文版即将付印的时候，iOS 的版本已经是 5.1.1 了，而且已经出现了 iOS 6 即将发布的迹象。而 OS X 版本依然在 Lion(10.7.4)，但是 Mountain Lion(10.8)已经推出开发者预览了，在本书英文版上架的时候应该已经发布正式版了。本书尽可能介绍最新的信息，覆盖所有的版本，并且持续地不断前进。

0.1　概述和阅读建议

这本书的规模很庞大。刚开始的时候，本来没有想把这本书设计得如此庞大细致，但是

随着我对 OS X 的深入了解，我发现了更多的深奥难解的地方，而对于这些难题我找不到详细的解释或文档。因此我发现自己在编写这本书的过程中开始涉及越来越多的方面。一个操作系统是一个完整的生态系统，有着自己的地理形态(硬件)、大气(虚拟内存)、植被和动物(进程)。这本书尝试着尽可能有条理地记录这些内容，同时不牺牲细节的清晰性(或反之亦然)。这不仅仅是一个壮举。

0.1.1 架构一瞥

OS X 和 iOS 有着复杂的架构，这个架构混杂了多项迥异的技术：NeXTSTEP 的 Cocoa 中遗留的 OS 9 对于 OS X 的 UI 和 API、BSD 的系统调用和内核层、源于 NeXTSTEP 的内核结构。尽管是一个融合体，但是各个组件之间的界线还是比较清晰的。图 0-1 给出了这个架构的鸟瞰图，并且标出了每一个组件对应本书的章节。

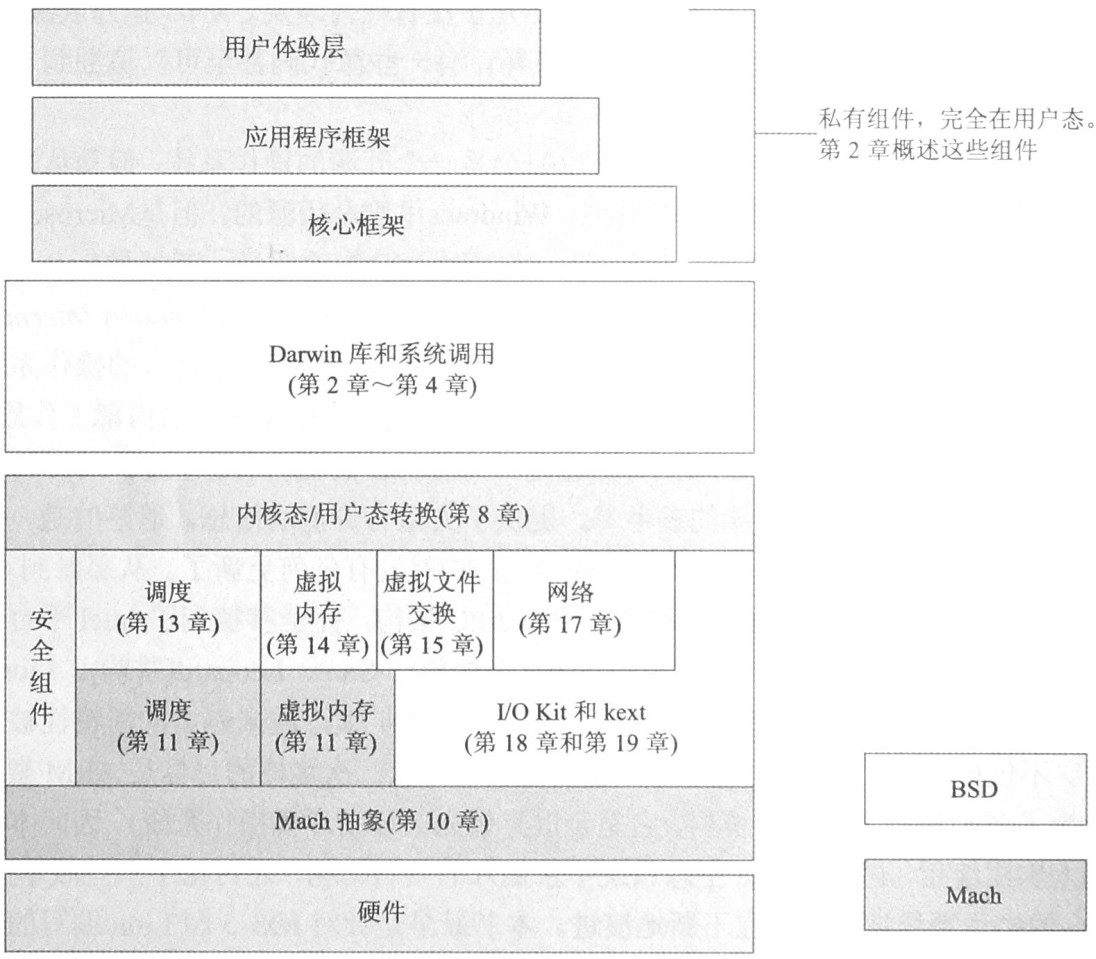

图 0-1 OS X 的架构以及每一个组件对应本书的章节

本书还另外包含了一些章节，介绍了和架构无关，但是非常重要的内容，例如调试(第 5 章)、固件(第 6 章)和用户态启动过程(第 7 章)、内核态启动过程(第 9 章)和内核模块(第 18 章)。最后还有两个附录，一个附录为 POSIX 系统调用和 Mach 陷阱提供了一个快速参考手册(这个附录在本书的支持网站上)，另一个附录对 Intel 架构和 ARM 架构的汇编语言做了一个简单的介绍。

0.1.2　目标读者

有 4 类读者可能会对这本书的全部或部分内容感兴趣：

- 想更深入了解 OS X 工作原理的高级用户和系统管理员。Mac OS 的占有率正在稳步增长，争夺了多年来被 PC 霸占的市场份额。Mac 在企业环境中越来越流行，在学术界已经盖过了 PC 的风头。
- 不满足于 Objective-C 层面的用户态开发人员，这些人想要了解他们编写的程序是如何真正在系统层次执行的。
- 想要探寻内核态底层驱动程序设计、内核增强或者文件系统和网络层的内核态开发人员。
- 不满足于使用现有工具或补丁进行越狱的黑客和越狱者，这些人想要理解补丁的工作机理和修补的内容，以及如何进一步对系统进行调整使其能满足自己的需求。注意，在这个上下文中，这些目标读者指的是为了兴趣、快乐和挑战而深入了解内部工作原理的人，而不是那些为了任何违法邪恶目的的人。

0.1.3　选择自己的学习路径

尽管这本书可以从头读到尾，不过不要忘了这毕竟是一本技术书籍。书中的章节设计为可以单独阅读，既可以作为详细的解释也可以作为快速参考。既可以按顺序阅读所有章节，也可以随机阅读感兴趣的章节，略读甚至跳过某些章节，以后可以跳回来进行更深入的阅读。如果一章内引用了之前章节讨论的概念或函数，那么这些内容会清晰地标注出来。

您还可以根据自己所属的读者类型选择自己的阅读策略。例如，本书第 I 部分中的几章可以分解为图 0-2 所示的流程：

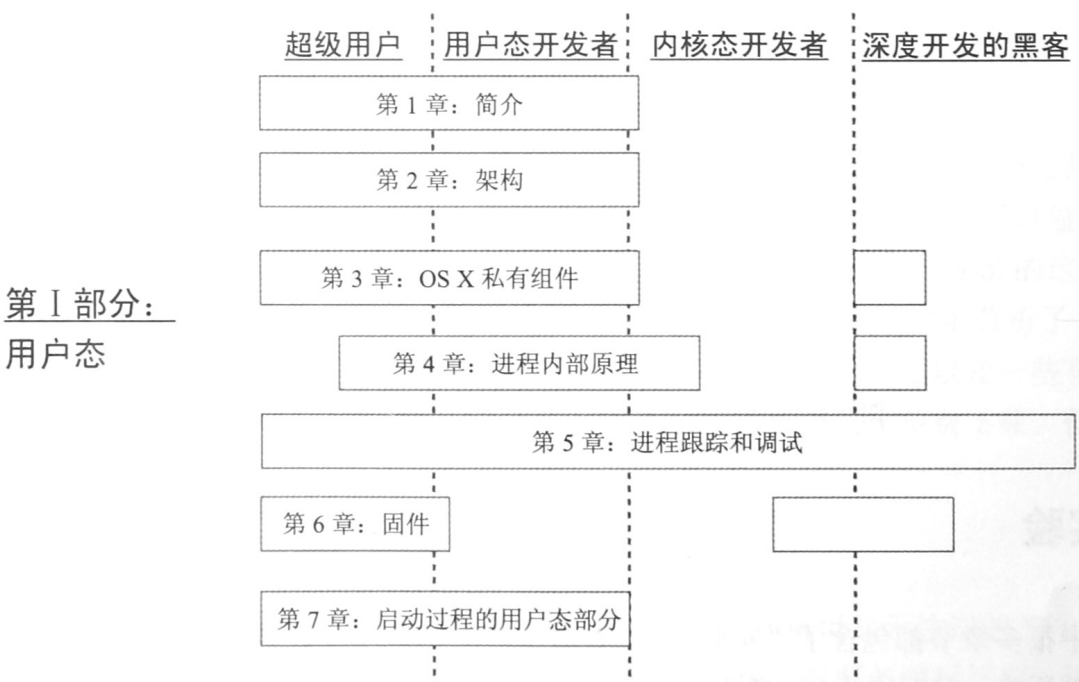

图 0-2　本书第 I 部分的阅读建议，这一部分关注于用户态的架构

在图 0-2 中，完整长度的条表示这一章的内容符合目标读者的兴趣，部分长度的条表示至少有部分符合目标读者的兴趣。当然，每一个读者的兴趣都不同。这也是为什么每一章开

头都要对本章讨论的内容做一个简单介绍的原因。同样地，只要看一下目录中每一章的小节标题就可以判断这一章是值得仔细阅读还是快速浏览即可。

本书的第II部分实际上可以自成一卷。这一部分关注于 XNU 内核的架构，因此比第 I 部分复杂得多。这是不可能避免的，内核本身就是一个更加复杂、实时且硬件受限的环境。这一部分包含更多的代码清单，甚至包含了少量用汇编实现的代码片段。本书第II部分的阅读建议如图 0-3 所示。

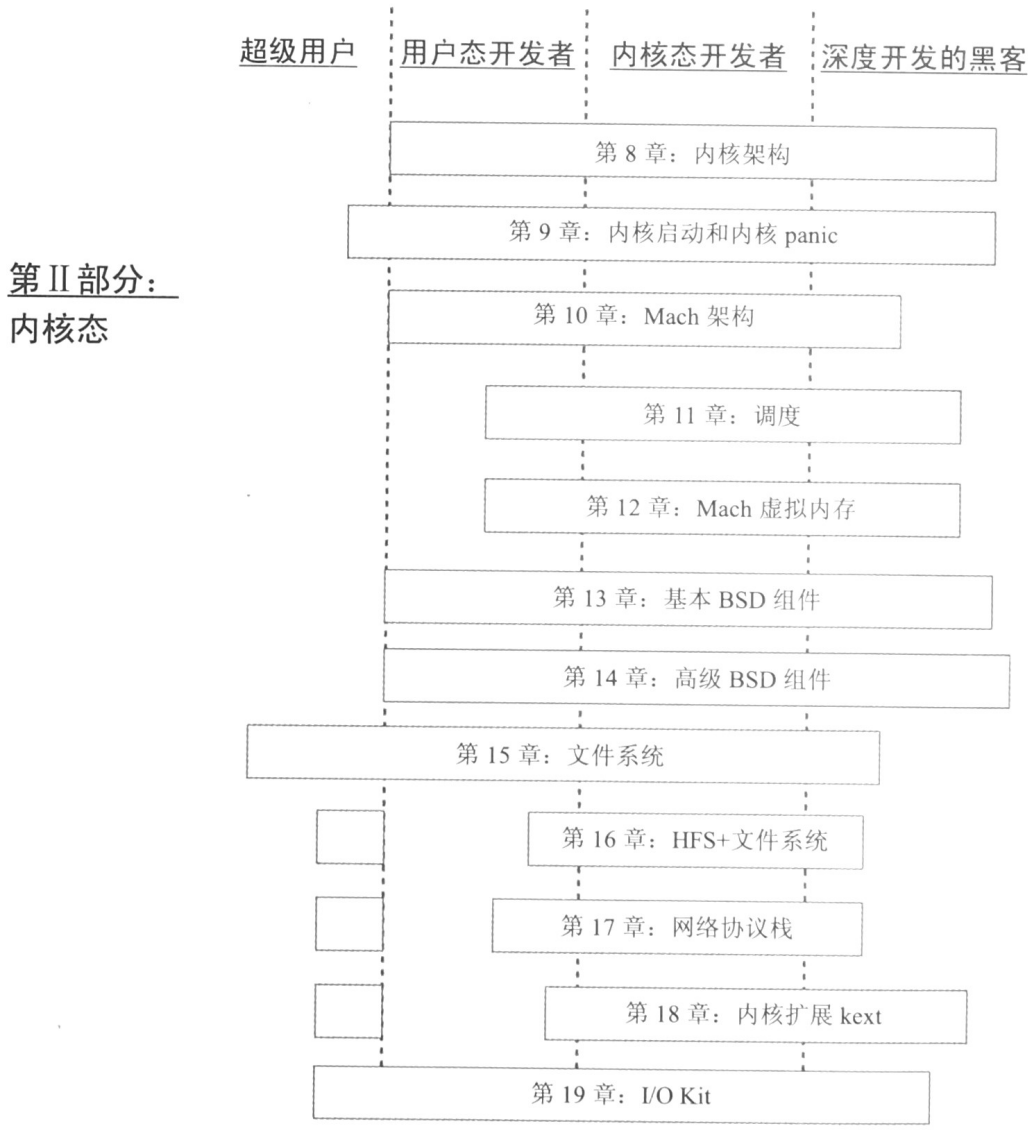

图 0-3　本书第II部分的阅读建议，这一部分关注于内核

0.2　实验

本书中很多章节都包含了“实验”，实验通常都需要运行一些 shell 命令，有时候需要运行一些示例程序。它们称为“实验”的原因是这些内容演示了操作系统中可能会变化的方面，这些方面可能会根据操作系统版本或配置发生变化。通常情况下，书中详细演示了这些实验的结果，但是鼓励读者在自己的系统上尝试这些实验，然后自己观察结果。和 UNIX 一样(Mac OS X 实现的就是一个 UNIX)，Mac OS X 也需要通过动手实践才能切身体会并且真正理解，

而通过道听途说是不够的。

在有些情况下，试验中有一些部分被留作练习让读者自己完成。尽管本书的支持网站上会有解答(练习的完整可用版本)，但是鼓励读者尽可能地自己独立完成这些部分。仔细地阅读本书，只要有少量的常识，您应该可以收获所有应该收获的内容。

0.3　工具

这本书还应用了一些工具，这些工具都是作者为了支撑这本书而开发的。这些工具都继承了 UNIX 的优良传统，都是命令行工具，而且输出结果很容易被 grep(1)处理，因此这些工具不仅适合手工使用，还适合在脚本中调用。

0.3.1　filemon

第 3 章介绍了一个名为“filemon”的工具，这个工具可以实时显示 OS X 和 iOS 上文件系统的活动。这个工具的命名是为了向 Russinovich 的同名工具致敬。这个简单的工具基于 OS X 和 iOS 5 上的 FSEvents 设备，能够跟踪文件系统相关的事件，例如文件的创建和删除。

0.3.2　psx

第 4 章介绍了一个名为 psx 的工具，这是一个类似 ps 的增强版工具，可以显示 OS X 中进程和线程的相关信息。这个工具在第 4 章中特别重要，该章讲解了进程的内部结构，并且演示了一个没有文档的系统调用 proc_info。如果查看的是自己创建的线程，这个工具不需要有特别的权限，但是查看其他进程则需要 root 权限。这个工具可以从本书的支持网站免费下载，带有完整的源代码。

0.3.3　jtool

对于大部分和二进制文件相关的功能，可以使用 OS X 自带的 otool(1)，这个工具能够很全面地分析数据区域(data section)，而且由于 ARM 架构具有两种模式的汇编，所以显示 ARM 二进制文件的时候会让人感到混淆。jtool 的目标是增强 otool，不仅解决了 otool 的这些缺陷，还提供了静态二进制分析的新特性。这个工具对第 4 章特别有用，该章详细分析了 Mach-O 文件格式，而且由于这个工具有很多有用的特性，例如寻找文件中的引用以及一些有限的反汇编能力，所以在后面几章也挺有用。这个工具可以从本书的支持网站免费下载，但是是闭源的。

0.3.4　dEFI

这个简单的程序可以导出 Intel Mac 上的固件(EFI)变量，还可以显示注册的 EFI 提供商。这个工具演示了基本的 EFI 编程技术——如何与引导服务和运行时服务进行接口。这个工具可以免费下载，带有完整的源代码。第 6 章介绍这个工具。

此为试读，需要完整PDF请访问：www.ertongbook.com

0.3.5 joker

第 8 章介绍的 joker 工具是一个简单的工具，这个工具可以用来和内核交互(特别是在 iOS 上)。这个工具可以查找并显示 iOS 和 OS X 内核的系统调用和 Mach 陷阱表，显示 sysctl 结构，并且在二进制文件中查找特定的模式。这个工具非常适合于逆向工程和黑客一类的活动，因为 Apple 已经不再导出陷阱和系统调用符号了。

0.3.6 corerupt

第 11 章讨论 Mach 虚拟内存管理器的底层 API。为了演示这些 API 是多么强大(和危险)，本书提供了 corerupt 工具。通过这个工具可以将任何进程的虚拟内存映射以核心转储兼容的格式导出到一个文件，类似于 Windows 的 Create Dump File 功能，也很像 *Mac OS X Internals: A Systems Approach* 一书中使用的 gcore 工具。corerupt 在其前身的基础上有了增强，提供了对 ARM 的支持，还支持对虚拟内存映射进行入侵式的操作，例如修改内存页面。

0.3.7 HFSleuth

HFSleuth 是本书中的一个重要工具，这是一个用于查看 OS X 原生的文件系统 HFS+支撑结构的多合一命令行工具。之所以开发这个工具，是因为确实没有什么其他方式能够展示这个非常复杂的文件系统的内部工作原理。Singh 的书 *Mac OS X Internals: A Systems Approach* 中也包含了一个类似的工具，称为 hfsdebug，这个工具的功能少一些，而且只支持 PowerPC，后来被一个商业工具 fileXRay 所替代。

为了在真实的文件系统上使用 HFSleuth，必须能够读这个文件系统。一种方法就是引导这个文件系统。HFSleuth 的功能几乎都是只读的，所以可以保证这个工具非常安全。但是访问文件系统所在的底层块设备(有时候是字符设备)的访问权限通常是 rw-r-----，意味着设备不能被其他用户读访问。如果您通常情况下不信任 root，并且坚持要使用较低的权限(这是一个明智的决定！)，那么一个同等有效的替代方案是通过 chmod(1)修改 HFS+分区所在设备的权限，使得自己的用户有权限读(通常使用 o+r 参数)。一些高级功能(例如修复和 HFS+/HFSX 转换)要求写访问权限。HFSleuth 可以从本书的支持网站免费下载，而且会一直免费。不过和其前任一样，这个工具也不是开源的。

0.3.8 lsock

netstat -o 是一个大家都非常需要的功能，能够显示进程对系统中 socket 的所有权，但是这个功能在 OS X 中就没有了。在 OS X 中，这个功能要通过 lsof(1)实现，但是后者需要剔除其他打开的文件将 socket 滤出来，所以比较麻烦。另一个缺失的功能是显示正在创建的 socket 连接，就像 Windows 的 TCPMon 提供的功能一样。第 17 章介绍的这个工具使用了一个没有正式文档的内核控制协议 com.apple.network.statistics，这个协议可以获得 socket 创建时候的实时通知。这个工具特别容易集成到脚本中，因此可以很方便地用作连接事件处理程序。

0.3.9 jkextstat

本书中使用的最后一个工具是 jkextstat，这是一个 kextstat(8)兼容的工具，能够列出内核

扩展。和最初的版本不一样，这个工具支持详细输出(verbose)模式，而且可以用于 iOS。因此，这个工具对于 iOS 内核的探讨来说价值重大，而这件事在这本书出现之前是很困难的，因为 iOS 的二进制 kextstat 使用了不再被支持的 API。这个工具在最初的版本上有了改进，允许更详细的输出信息，能关注于特定的内核扩展，而且还支持输出到 XML 格式。

这里提到的所有工具都可以免费获得，而且会一直免费下去，不论您有没有购买(或复制)这本书。这是因为这些工具都很有用，而且填补了很多高级功能的空白，这些高级功能原本要么没有，要么隐藏在 Apple 自己的工具中。

0.4　本书使用的约定

为了使本书更容易阅读，避免经常反复强调示例代码和程序的具体背景，本书使用了一些约定，通过这些约定可以巧妙地提醒您给定代码清单的背景。

0.4.1　出场演员表

本书中的演示和代码清单自然是在各种不同版本的 Apple 电脑和 i-设备上产生和测试的。根据系统管理员给设备命名的习惯，每一台主机都有自己的“个性”和名字。为了避免重复说明每一个演示是在哪一台设备和操作系统上运行的，书中保留了 shell 的命令行提示符，通过其中的主机名可以找到这个演示所在平台对应的 OS X 或 iOS 版本(详见表 0-1)。

表 0-1　本书演示程序所用的主机名和版本信息

主　机　名	设 备 类 型	操作系统版本	使 用 目 的
Ergo	MacBook Air, 2010	Snow Leopard, 10.6.8	OS X 的一般性特性演示。在 Snow Leopard 及更新的版本上测试
iPhonoclast	iPhone 4S	iOS 5.1.1	iOS 5 及更新的版本在 A5 处理器(ARM 的多核处理器)上的特性测试
Minion	Mac Mini, 2010	Lion, 10.7.4	Lion 相关的特性演示
Simulacrum	VMWare 镜像	Mountain Lion, 10.8.0 DP3	Mountain Lion(开发者预览版)相关的特性演示
Padishah	iPad 2	iOS 4.3.3	iOS 4 及更新的版本的特性
Podicum	iPod Touch, 4G	iOS 5.0.1	iOS 5 相关的特性，在 A4 或 A5 处理器上

此外，带有 root@的命令行提示符表示只能通过 root 用户运行的命令。通过这个标志很容易区分哪些示例能运行在哪些系统上，以及需要什么权限。

0.4.2 代码节选和示例

本书包含了大量代码示例，分为以下两类：

- **示例程序**：主要出现在第 I 部分。这些程序通常用于演示在用户态有效的简单概念和原理，或者演示一些特殊的 API 和库。这些示例程序都是由作者本人编写的，而且注释良好，您可以随意尝试这些程序，可以按照自己希望的方式修改，或者也可以不去碰这些程序。如果想要偷懒的话，还可以从本书的支持网站上下载这些程序的源代码和二进制程序。
- **Darwin 的代码节选**：主要出现在第 II 部分。这些代码几乎都是 XNU 代码的完整片段，截取自最新开源版本 1699.26.8(对应于 Lion 10.7.4)。所有代码都是开源的，但是要遵循 Apple 的 Public Source License。本书提供代码节选的目的是展示 XNU 架构中的相关部分。由于自然语言容易产生一些歧义，而代码既不用依赖上下文又准确(遗憾的是，有时候可读性不是那么好)，因此最准确的解释往往来自于对代码的阅读。当书中引用代码的时候，有可能指的是/usr/include 目录下的头文件(通过标准的 C 语言<>记号表示，例如<mach/mach-o.h>)。有时候，也有可能指的是来自于 XNU 或相关软件包的 Darwin 的源代码。在这种情况下会使用相对路径(例如 osfmk/kern/spl.c，指的是相对于 XNU 内核源代码解压的路径)。相关软件包总是会标注出来，不过书中第部 II 分引用的代码几乎都是 XNU 内核的代码。

XNU 和 Darwin 组件的代码都有很好的文档，不过本书尝试更进一步，在必要的时候在代码中以注释的形式添加额外的解释。为了区分，这种不属于原始代码的注释都是用 C++风格的注释清晰地标注出来，而不是 Darwin 中使用的 C 风格的注释，例如下面这个代码清单示例：

代码清单 0-1：示例代码清单

```
/* This is a Darwin comment, as it appears in the original source */

// This is an annotation provided by the author, elaborating or explaining
// something which the documentation may or may not leave wanting

// Where the source code is long and tedious, or just obvious, some parts may
// be omitted, and this is denoted by a comment marking ellipsis (...), i.e:

// ...

    important parts of a listing or output may be shown in bold
```

本书区分“输出清单”和“代码清单”。代码清单是直接从程序源代码文件或系统配置文件中摘抄出来的。而输出清单则是用户命令运行捕获的文本，用于演示在 OS X、iOS 或二者上运行的结果。本书旨在比较和对比两个系统，因此常常会见到同样的命令序列在两个系统上的执行结果。在输出清单中，可以看到用户命令用粗体表示，我们鼓励读者仿照书中在自己的系统上实验这些命令。

通常情况下，书中提供代码清单的目的是为了阐述清楚问题，而不是让人感到更迷惑。自然语言带有一定的二义性，但是代码则只有一种解释方式(即使有时候这样的方式并不完全清晰明了)。只要有可能，书中会用清晰的描述辅以详尽的图示，期望能够帮助您快速理解代码。对 C 语言(有时候要求一点汇编语言)当然能够帮助阅读书中的代码示例，但是并不是必要的。代码中的注释——特别是额外的注解——可以帮助您理解代码中的要点。书中使用更多的是框图和流程图，把函数当做黑盒子。读者可以自己选择是停留在大致了解的层次，还是深入研究实现中具体的变量和函数。不过值得注意的是，代码本身非常复杂，是很多人和很多编码风格的产物，在整个 XNU 中可以看到各种风格迥异的代码。

在 iOS 方面，XNU 仍然保持闭源。iOS 版本使用的 XNU 版本实际上比公开发布的版本领先很多版本。因而，书中也无法展示相应的示例源代码，但有时候会给出一些反汇编的代码(主要来自于 iOS 5.x)。这里使用的汇编是 ARM 汇编，代码中有一些帮助解释内部工作原理的注释(这些注释全部是由本书作者提供的)。对于和汇编相关的知识，可以参考本书的附录快速了解。

0.4.3　排版约定

本书约定命令后面括号中的数字表示这条命令所在 man 手册的节数(如果有的话)。例如：ls(1)表示一条用户命令，write(2)表示一个系统调用，printf(3)表示一个库函数调用，ipfw(8)表示一条系统管理命令。本书中描述的大部分命令和系统调用都在man手册中有很好的文档，本书也不打算替代精美手册的地位(因此，有问题请首先参阅手册)。然而，文档偶尔会遗漏一些内容——甚至根本没有听过文档记载——这时本书就会给出更多的信息。

0.5　支持网站

OS X和iOS都在高速地演进，而且会一直持续下去。我尽可能跟上演进的步伐，并且为本书更新维护了一个支持网站，地址是http://newosxbook.com。我的公司(http://technologeeks.com)也在LinkedIn上维护了OS X和iOS内核开发者小组(与那些Windows和Android的小组并列)，它包含一个问答论坛，这个论坛有希望成为一个热门的OS X和iOS相关问题的讨论场所。

本书支持网站的内容包括：

- 一个列出了各种 POSIX 和 Mach 系统调用的附录。
- 本书中所有实验中包含的示例程序——让有热情但是懒得敲代码的读者尝试。这些程序不仅提供了源代码的形式，还提供了二进制文件(给那些甚至懒得编译或不想使用 Xcode 的读者)。
- 本书中介绍的(也就是在这个前言中讨论的)工具，可以免费下载使用 OS X 平台和 iOS 平台的二进制文件，有些还提供了源代码。
- 其他网络资源的更新引用和链接。
- 随着时间的推移，会有一些关于新特性和改进的更新文章。
- 勘误表——人都会犯错误——尤其是 iOS 中的错误，因为大部分和 iOS 有关的细节都是通过反汇编挖掘出来的，这些内容中可能有一些不准确的地方或版本的差异需要修正。

本书是一段不可思议的旅程，您将带着(玩小猫的时候用的)护目镜，慢慢地揭开支持用户态应用程序的真实脉络。我真切地希望读者能像我一样得到启示，了解的不仅是关于 OS X 和 iOS 的思想，还有整个操作系统架构和软件设计的基本思想。

翻开这本书，开始这段奇妙的旅程吧！

目　录

第 I 部分　高级用户指南

第1章　达尔文主义：OS X的进化史 …… 3
1.1　前达尔文时代：Mac OS Classic …… 3
1.2　浪子回头：NeXTSTEP …… 4
1.3　走进新时代：OS X 操作系统 …… 4
1.4　迄今为止的所有 OS X 版本 …… 5
1.4.1　10.0——Cheetah，初出茅庐 …… 5
1.4.2　10.1——Puma，更强大 …… 5
1.4.3　10.2——Jaguar，渐入佳境 …… 6
1.4.4　10.3——Panther 和 Safari …… 6
1.4.5　10.4——Tiger，转投 Intel 的怀抱 …… 6
1.4.6　10.5——Leopard 和 UNIX …… 6
1.4.7　10.6——Snow Leopard …… 7
1.4.8　10.7——Lion …… 7
1.4.9　10.8——Mountain Lion …… 8
1.5　iOS——走向移动平台的 OS X …… 9
1.5.1　1.x——Heavenly，第一代 iPhone …… 9
1.5.2　2.x——App Store、3G 和企业级的特性 …… 10
1.5.3　3.x——告别第一代，迎来 iPad …… 10
1.5.4　4.x——iPhone 4、Apple TV 和 iPad 2 …… 10
1.5.5　5.x——iPhone 4S 和更新的硬件 …… 11
1.5.6　iOS 和 OS X 对比 …… 11
1.6　OS X 的未来 …… 13
1.7　本章小结 …… 14
参考文献 …… 15

第2章　合众为一：OS X和iOS的架构 …… 17
2.1　OS X 架构概述 …… 17
2.2　用户体验层 …… 19
2.2.1　Aqua …… 19
2.2.2　QuickLook …… 20
2.2.3　Spotlight …… 21
2.3　Darwin——UNIX 核心 …… 22
2.3.1　Shell …… 22
2.3.2　文件系统 …… 23
2.4　UNIX 的系统目录 …… 23
2.4.1　OS X 特有的目录 …… 24
2.4.2　iOS 文件系统的区别 …… 25
2.5　bundle …… 25
2.6　应用程序和 app …… 26
2.6.1　Info.plist …… 27
2.6.2　Resources 目录 …… 29
2.6.3　NIB 文件 …… 29
2.6.4　通过.lproj 文件实现国际化 …… 30
2.6.5　图标文件(.icns) …… 30
2.6.6　CodeResources …… 30
2.7　框架 …… 33
2.7.1　框架 bundle 格式 …… 33
2.7.2　OS X 和 iOS 公共框架列表 …… 35
2.8　库 …… 41
2.9　其他应用程序类型 …… 43
2.9.1　Java(仅限于 OS X) …… 43
2.9.2　Widget …… 43
2.9.3　BSD/Mach 原生程序 …… 44
2.10　系统调用 …… 44
2.10.1　POSIX …… 44
2.10.2　Mach 系统调用 …… 45

此为试读，需要完整PDF请访问：www.ertongbook.com