Broadview
www.broadview.com.cn

Windows内核设计思想

陈树宝 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Windows内核设计思想

陈树宝 著

電子工業出版社
Publishing House of Electronics Industry

内容简介

本书主要讲述 Windows 内核的设计过程，从最底层的细节使用源码一步一步分析，结合 Bochs 和 WinDbg 调试器进行验证。本书提供全部源代码和能直接编译的项目工程，集理论、架构、编码、运行和调试于一体进行讲述，从多种角度呈现内核构架的基本流程。本书主要包括了 Windows 内核加载器（ntldr）的分析，Windows 内核调试组件的设计，实现结构化异常处理的支持，并对内存管理和对象管理进行了精心讲解，同时对基于 IRP 请求包的 I/O 系统进行了论述，并且介绍了如何设计文件系统，最后简单讲解了进程和线程的一些基本知识。

本书适合希望深入了解 Windows 内核框架的程序员及对此感兴趣的读者阅读。

图书在版编目（CIP）数据

Windows 内核设计思想 / 陈树宝著. — 北京：电子工业出版社，2015.3
ISBN 978-7-121-25314-0
I. ① W…II. ① 陈…III. ① Windows 操作系统－程序设计 IV. ① TP316.7

中国版本图书馆 CIP 数据核字（2014）第 307151 号

策划编辑：李　冰
责任编辑：白　涛
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
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：39.75　字数：992 千字
版　　次：2015 年 3 月第 1 版
印　　次：2015 年 3 月第 1 次印刷
印　　数：2500 册　　定价：108.00 元

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

服务热线：（010）88258888。

推荐序1

关于操作系统内核设计的书一般内容都庞大复杂，细致精微，读起来相当费劲。比起一般的技术书籍来说，阅读的过程痛苦程度高了很多。其实最要命的是将这些章节一一读完弄懂之后，我们稍有的成就感转瞬即逝，另一个问题接踵而来：我搞明白了这些东西，究竟能做什么用？要知道满大街的程序员，会被拉去开发操作系统的，一百个里边也许都没有一个。

操作系统是一种基础设施，在信息化的世界里，它就像电力和自来水一样。现在有不少呼声，我们要开发自己的操作系统。然而每次真正有适合PC端的操作系统开发出来，立刻会被网上一大批人一边倒地鄙视："抄袭"、"拿Linux改个名就说是自己的"、"骗科研经费"等等说法不绝于耳。其实并非这些人真的体验过新的操作系统，也不是他们真的了解这些开发过程的容易或艰辛。而是每个人心里都很清楚，无论新的操作系统做得有多好，它都不会像Windows这样在千万个用户的电脑上流行起来。所以他们的鄙视之言永远都不会得到事实的反驳，他们可以放心大胆地在网络言论上通过"喷"来轻松地显示自己的睿智和存在感。

这是因为操作系统并非一个孤立的东西。就像自来水厂绝非一处水源加一个处理工厂。自来水厂的核心竞争力反而不是水源，而是那千万根连通到客户家中的水管。我很容易就能新找一处水源，建设一个自来水厂。但是谁家愿意为我把自己家的水管敲掉换成我的？

在漫长的时间里，我们已经习惯了使用Windows上的软件。就算现在有一款更好的操作系统，但它无法运行我需要的软件，我也会毫不犹豫地拒绝它。假设这款操作系统能完美地兼容Windows的软件，又会面临另一个问题：既然开发者都知道只需要给Windows开发软件，就能兼容另一个操作系统，那么开发者就永远只会给Windows开发软件了。本来因为软件数量不足而兼容Windows软件，然后又因此而导致软件数量更少了。这成为一个恶性循环！Windows只要略微升级，改变特性，这款新的操作系统也得随之升级，否则新版的软件又无法支持。命运完全悬于人手。

所以操作系统能形成垄断，并非因为技术有多高明。说什么Windows有几千万行代码所以永远也不能超过那完全是扯淡。难道说越臃肿的东西就越是强大？Windows的几千万行代码的规模反而有可能是历史的不断积累，垃圾代码无法清除（为了兼容旧版软件和扯不清的部门利益）而导致的。从头开发一个新的操作系统反而轻松很多。问题是操作系统天然属于占坑型的软件。一旦占住了这个坑，除非占坑者主动放弃，否则垄断很难被攻破。

这不是一个技术问题。PC 上的 Windows 如此，移动平台上的 Android 与 iOS 也如此。

那为什么我们还要学习它？

因为我们可以使用它，但不能任它忽悠我们，即便它像自来水一样便宜（目前 Windows 依然贵得离谱，这是当代奇葩之一），或者干脆就是免费的（比如 Android）。笔者倾向于操作系统会越来越便宜，最终变成免费或者接近于免费的，经济上应该不会成为一个问题。但我们必须了解它，熟知它的问题。就像我们得了解我们每天喝的自来水，里边有漂白粉，甚至抗生素！

只有真正明白学了做什么用，才有可能有学习的动力，才有可能真正学好。如今要开发自主版权的操作系统的呼声，很大一部分来自安全的需要。没有人愿意让自己安全的命脉攥在别人的手里。其实我反而觉得，想要真正把握自己安全的命脉，与其等待自己开发的操作系统去夺了已经被占的坑，不如充分地了解现在占坑的操作系统是怎么运作的。一个操作系统不管多么封闭多么神秘，它总归是一堆二进制代码。它的执行过程是可以被分析被监控的。至于有人说缺少源代码，事实上国家是可以要求微软提供源代码的。问题倒在于分析源代码的人：如果你不懂操作系统，那么纵然三千万行源代码摆在你面前，里边堂而皇之地写上一百个漏洞，你又能找出来一个吗？

学习这些前辈们的作品，读者不但能了解前辈们的英明睿智，还能发现他们的局限。希望读者不要做一个单纯的膜拜者和盲从者。

如果你觉得 Windows 的 64 位版驱动要验签名了，内核有 PatchGuard 了，所以更安全了；如果你觉得你用的是“水果”，不越狱所以固若金汤；如果你觉得 Android 不 root 就是安全的，我只能说你还处于蒙昧无知的状态。如果你感觉到了哪里可能会有漏洞，而你可以一一填补，说明你混沌初开了。如果你像某些大公司的系统设计员一样，忽然冒出一个前无古人的想法，认为这样可以一劳永逸地解决安全问题，说明你开始了独立思考，可以着手安全问题了！如果你终于回到现实，甘愿像被缚的普罗米修斯一样，今夜痊愈，明日又开始新一轮的开膛破肚，永无休止，那你终于走上了正途！

想要开始这个奇妙的旅程，就打开本书，认真了解 Windows 内核设计的每个细节吧。我谨以此文，祝贺陈老师，祝贺《Windows 内核设计思想》一书出版。

畅销书《天书夜读——从汇编语言到 Windows 内核编程》
《寒江独钓——Windows 内核安全编程》作者 谭文
2015 年 1 月 13 日

推荐序2

认识陈树宝先生到现在大概也有5年时间了，他是个技术深厚、勤奋踏实的开发牛人，常常承担项目主要的开发职责。刚接触他时就知道他在研究Windows内核方面下了很大的工夫，从系统启动阶段到系统各种管理机制都有研究，例如内存管理、对象管理、进程管理、I/O系统、文件系统等，我甚是佩服他在这方面的功底造诣。由于自己对Windows内核了解不多，遇到这方面的疑问，偶尔也会请教于他，总能得到一些收获。

相比于历来都是扎堆热门的Linux内核研究，以及雨后春　般蓬勃发展起来的Andriod系统类研究书籍，国内对Windows内核进行系统地全面深入研究的书籍实在太少了。据我浅薄的见闻所知，大概也只有毛德操先生和潘爱民先生的著作属于这类书籍，当然还有翻译自国外经典的《Windows Internals》（中文版为《深入解析Windows操作系统》）一书。实际上，国内研究Windows内核的人非常多，但常常限于坊间研究，能愿意公布出来分享的不多。或者是限于本职工作范围内所使用到的，或者是一些零零散散的资料。

当我知道陈树宝先生打算写《Windows内核设计思想》一书，我感到非常高兴。一来能将他这几年积累的知识呈现出来，我觉得这对国内想进入或者想了解这个领域的读者来说是幸运的，至少可以少走一些他以前走过的弯路；二来通过书籍能够实现自己一定的价值，并且为这个领域贡献出一份力量。我们都知道，沉醉于研究对于那些务实的技术人来说是孤独的，需要有毅力恒心。而花费时间将自己的研究心血分享出来更是难得的。在起笔此书时，我也曾经问过陈树宝先生打算写哪方面的内容，也有一些疑虑，毕竟要写好这类“大气”的书是不容易的，不仅涉及知识面广，而且也需要这方面能力的自信。所幸的是，由于知识的积累，陈树宝先生轻车熟路，手到擒来，竟似不费功夫，让我消除了这个疑虑。

我很高兴为陈树宝先生的《Windows内核设计思想》一书作序推荐。我了解到的信息，也正如陈树宝先生在《Windows内核设计思想》一书前言所说的，这本书是以介绍源码（这个源码有不少是自己整理和修改过并经过验证的），结合相应的调试来展现出Windows内核的设计思路，通过调试代码能让读者更好地理解整个内核框架。我细看了此书的目录，第1章介绍了搭建开发和调试环境，第2章主要介绍OS的引导阶段，而第3章更是直捣黄龙，进一步深入地介绍Windows的调试机制，可见调试在这本书里是非常重要的。作者非常注重实践，毫无保留地呈现核心的内容。在第6章开始介绍了热门的，也是必不可少的内存管理机制。我们知道，内存管理对一个OS来说太重要了，Windows设计了一套特别的内存管理手段，这部分必然是非常值得一看的。第7章的对象管理更是Windows独特的地

方，对象管理充斥于 Windows 的每个地方，这个机制非常重要。I/O 系统、文件系统、进程管理在本书中都能看到。总而言之，这是一本非常棒的好书。

作者在根据自己心得总结整理出来的代码基础上，结合系统机理，通过调试验证。我认为这是本书的特别之处，它有别于其他过于理论描述的书籍，也有别于其他直接介绍源代码的书籍，是一本值得期待的书。

畅销书《x86/64 体系探索及编程》《处理器虚拟化技术》作者 邓志

2015 年 1 月 11 日

序言

与 Linux 作为一套自由开发的类 UNIX 操作系统不同，Windows 内核代码长期以来为微软所控制，始终保持云遮雾罩的神秘感。尽管在 2006 年微软针对教学和科研的目的，开放了 Windows 内核的部分源码 WRK（Windows Research Kernel），但是由于知识产权的问题，并不能摆在台面上来讲，有关这方面的原因，在毛德操老师所著的《Windows 内核情景分析——采用开源代码 ReactOS》中讲过。为此，笔者在本书中为求系统架构、程序流程设计的完整性，同时保证代码的高质量，经历了数次重构，其中第一次设计使用的就是 WRK 代码，因为这是最正统的 Windows 代码，只是代码不能大幅度引用，造成了完整性的缺失，而本书要求能完整编译的开源代码，如果不这样，会影响读者的理解，于是放弃。笔者随后想到了 ReactOS 开源项目，但是 ReactOS 源码偏多，于是在第二次设计中使用的是 WRK 和 ReactOS 代码的结合，即使用 ReactOS 代码去填充 WRK 的不完整部分，这种方法虽然保证了代码的完整性，但是由于代码的风格和设计不同，使得代码整体变得杂乱，因此这种方案也不可取。最终笔者抛弃了 WRK，采用完整的 ReactOS 代码（经过部分修改），虽然与正统的 WRK 代码不同，但是原理是一样的，而且表达很简洁，完全保留了 Windows 内核设计的中心思想，更有利于读者的学习。为了照顾部分读者学习 WRK，笔者在 MBR 处提供了切换回 WRK 的系统编译选项，有关 WRK 的更多知识，请参考潘爱民老师所著《Windows 内核原理与实现》。

无论使用的是什么样的代码方式，本书都保证了理论上的正确性，大多数的理论都经过了笔者的调试验证，而笔者也将在本书中向读者一步步展示自己对这些理论的理解和验证过程，集理论、架构、编码、运行和调试于一体进行讲解，多角度全方位体现一个操作系统内核的完整设计过程。

Windows 内核的源码是相当多的，单就一本书是不可能完全讲解完的，笔者也自认为还没有足够的能力去全部讲解。但重要的不是量多，而是质量上乘，精炼，笔者本着确保读者能在 Windows 内核设计上形成一条清晰主线的目的，精选了内核设计上最重要的代码片段供读者学习。笔者认为，只要读者认真理解了这些代码片段，想要再深入学习别的部分就基本不成问题了，无论你是做 Win32 应用程序编程的，还是做内核驱动编程的，都已经有足够的思路去设计你的代码，写自己的东西，而且一旦出现问题，也有足够的能力去找到解决问题的方法。

在本书代码的设计中，笔者本人也按照 Windows 内核原理写过不少代码，但最终还是决定不收纳在本书中。原因是本书的用意是讲解 Windows 设计的核心原理，应该力求让读者掌握 Windows 设计最本质的东西，而不被其他繁杂的细节所影响。当读者认真读完这本书之后，相信也会有像笔者一样的冲动，想要根据自己的想法在功能不变的基础上去设计自己的代码片段，自由构架操作系统。

如果读者在学习过程中遇到问题，可以登录网站 book.yetingyu.com 获得更多的知识。

特别致谢

郑少华，华南师范大学计算机学院研究生，参与了本书部分章节的编写，处理了全部书稿的文字错误，并进行了代码的整理、编译、链接和调试验证。

谭添升，微软公司程序员，在本书第 6 章的设计和编写过程中，给予了不少建议并修改了不少问题。

邓志，《x86/64 体系探索及编程》和《处理器虚拟化技术》的作者，修正了本书第 2 章中的不少问题并为本书作序推荐。

谭文，《天书夜读——从汇编语言到 Windows 内核编程》和《寒江独钓——Windows 内核安全编程》的作者，为本书作序推荐。

李冰，本书的策划编辑，对于笔者的写作给予了充分认可和支持，她富有主见，决策果断，参与了本书的命名。

白涛，本书的技术编辑，处理问题非常细致，认真修改了书中的多处错误，并且把本书流程整理得有条不紊。

黄爱萍，电子工业出版社博文视点编辑，在本书写作过程中非常耐心地解答笔者提出的问题，并给予鼓励。

陈树宝

2015 年 1 月 14 日

前言

本书最大的特色在于不从纯理论角度进行介绍，而是以源码与调试相结合来阐释。在ReactOS源码的基础上，剔除掉一些不那么重要，且影响大家整体把握Windows内核设计的部分，修改和整理出一个新的源码版本，通过对这个源码版本用调试工具Bochs和WinDbg进行调试，再结合重点模块的论述，以及关键函数完整详细的分析，使读者能对整个Windows内核框架透彻理解，并能根据自己的想法自由设计操作系统。

以下是书中介绍的读者可能比较感兴趣的知识点：

1. 物理内存如何收集？
2. 实模式和保护模式如何互相切换？
3. 系统中断如何安装？
4. 调试系统如何设计？
5. 分页机制如何开启？
6. 内核模块和启动型驱动如何加载？
7. 结构化异常如何为编程语言提供支持？
8. 如何为数据结构与算法学习提供支持？
9. 物理内存和虚拟内存如何进行管理？
10. 内核对象如何设计？
11. I/O管理器如何为驱动程序提供支持？
12. 磁盘类驱动和微端口驱动如何读/写数据？
13. 文件系统如何设计？缓存怎样提供支持？
14. 原型PTE是什么？（section object）如何实现进程间共享？
15. NTDLL.DLL如何映射到进程地址空间？
16. 第一个用户进程和线程怎样创建？线程如何切换？

为了使读者能从整体上把握本书讲述的内容，我们先做总体概述，提供一些视图供大家理解，这些视图跟部分章节相关。

内核启动基本流程

本书主要探索的是 Windows 内核的设计部分，分析内核构架和代码细节。本书编写的理论基础源于 ReactOS 和 WRK，为了保证本书编写的理论正确性，每一部分内容都经过了调试验证。学习流程从 MBR（master boot record，即主引导记录）到内核创建的第一个用户进程（smss.exe），图 1大体展示了内核初始化的各个阶段。

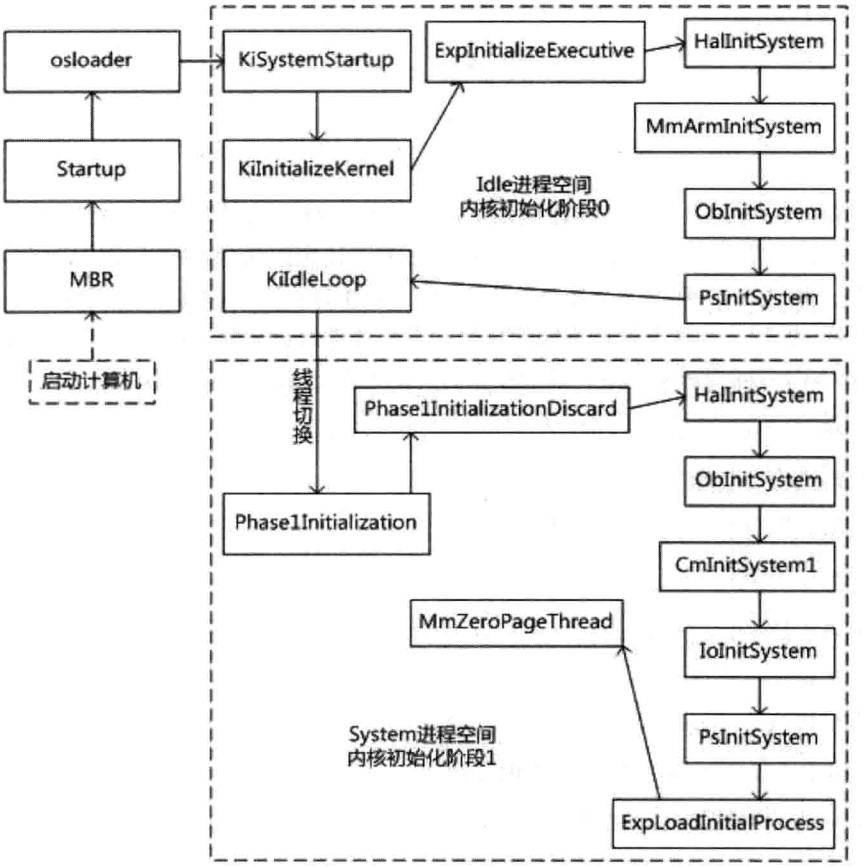


图 1

1. 计算机启动时控制权交给 MBR，MBR 负责把 NTLDR 加载到物理内存 0x20000。控制权交给 NTLDR 前半部分 SU 模块（startup.com）。

2. SU 收集物理内存、开启 A20、开启保护模式、解析 NTLDR 文件的后半部分 Loader 模块（osloader.exe）。解析 PE 文件格式重定位到物理内存 0x00400000。控制权交给 Loader 模块。

3. Loader 模块初始化调试系统，初始化内存管理器，开启分页机制，加载内核模块和启动型驱动程序到内核地址空间。控制权交给 ntoskrnl。

4. KiSystemStartup 进行内核初始化阶段 0：初始化调试系统，初始化 Idle 进程和线程，初始化内核执行体，初始化 HAL 系统，初始化内存系统，初始化对象系统，初始化进程系统（创建 System 进程和线程）。最后这条主线程转到 KiIdleLoop 成为真正的空闲线程。

5. KiIdleLoop 切换 Idle 线程到 System 线程执行，进行内核初始化阶段 1：初始化 HAL 系统，初始化对象系统，初始化注册表系统，初始化 I/O 系统，初始化进程系统，创建第一个用户进程（smss.exe）。最后这条主线程成为零页面线程。

由于内核涉及的内容非常多，因此并不是每一部分都详细讲述，但是经过本书的学习，再去理解那些未讲述的部分，会相对容易得多。

调试运行 WRK

图 2是使用 ntldr 加载运行的 WRK v1.2 系统界面，物理内存为 256MB，硬盘为 16MB，调试器为 WinDbg，通信端口为 COM 1。我们使用的虚拟机是 VMware Workstation。

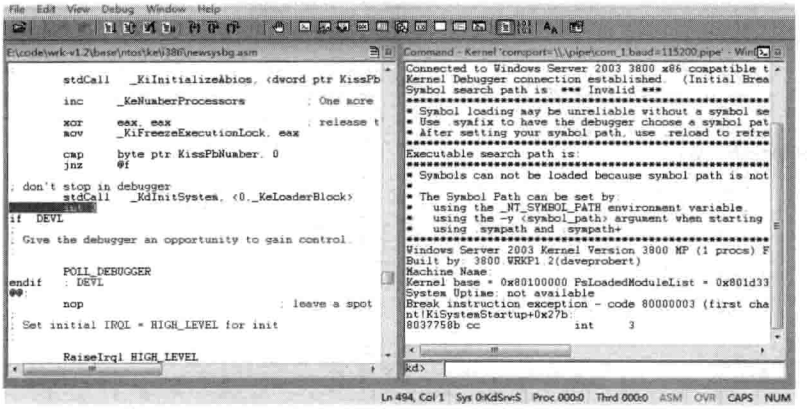


图 2

图 3所示是在内核初始化阶段 1，函数 IoInitSystem 初始化启动型驱动程序后的界面。

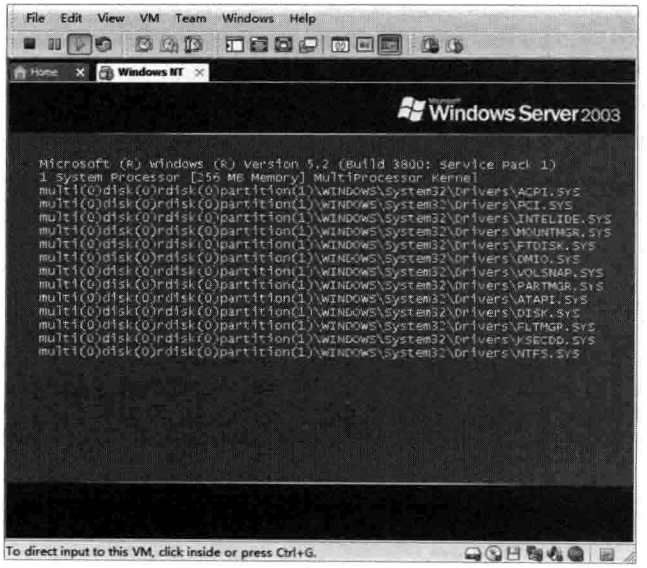


图 3

系统加载的文件是从“Windows Server 2003 Enterprise Edition Server Pack 1”操作系统里复制过来的。可使用本书编写的简单文件系统 ntfs.sys 替换原版 ntfs.sys。

调试运行 ReactOS

图 4是笔者自己编译的 ReactOS 的调试运行系统界面，物理内存为 256MB，硬盘为 16MB，调试器为 WinDbg，通信端口为 COM 1。通过修改文件 boot.asm 里面的代码，ntldr 加载的就是 ReactOS 而不是 WRK 了。我们使用的虚拟机是 Bochs。

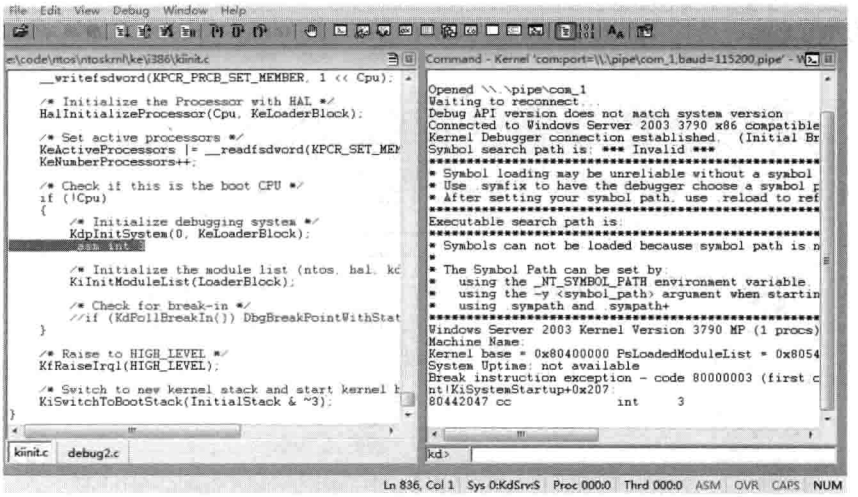


图 4

图 5所示是在内核初始化阶段 1，函数 IoInitSystem 初始化启动型驱动后显示的界面。

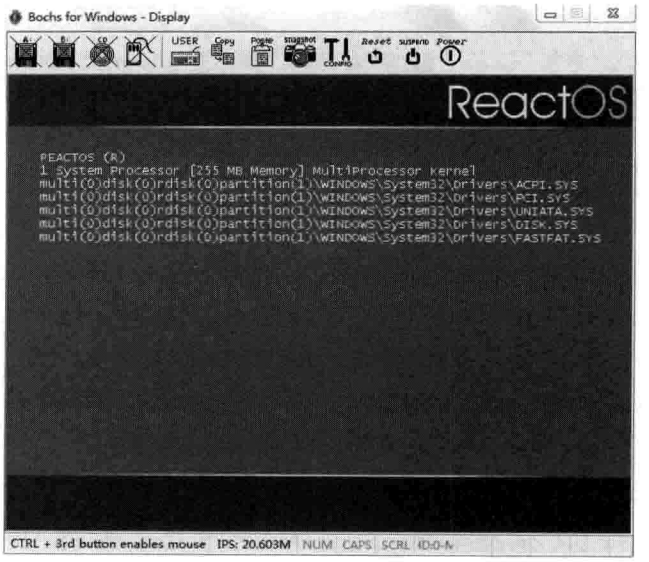


图 5

这些驱动程序全部以源码方式编译，可跟踪调试。FASTFAT 是我们自己实现的简单文件系统驱动（重新构架）。磁盘类驱动 DISK 和微端口驱动 UNIATA 实现了对磁盘数据的读/写支持。通过 I/O 管理器，弄清楚一个文件如何解析，如何从分区读到内存。

使用原版 ReactOS 系统环境调试

一直跟随 ReactOS 成长的读者，可以按照下面的操作替换内核模块。从http://sourceforge.net/projects/reactos/files/ReactOS/0.3.15/ 下载 ReactOS-0.3.15-REL-iso.zip，解压安装 ReactOS 0.3.15 系统，注意虚拟磁盘类型选择 IDE（默认是 SCSI）。系统安装完成后，虽然 ReactOS 默认提供了调试器 RosDBG.exe 可供调试，但是它并不友好，也无法进行源码调试，非常不利于学习。现在我们寻求一种替代方案——WinDbg。我们已经编译好了内核模块 ntoskrnl.exe，这个模块自带了 kd 调试组件，现在我们把 ntoskrnl.exe 替换原版 ReactOS 中的 ntoskrnl.exe。编辑虚拟机参数：在主界面中单击“Edit virtual machine settings”，弹出如图 6所示窗口。

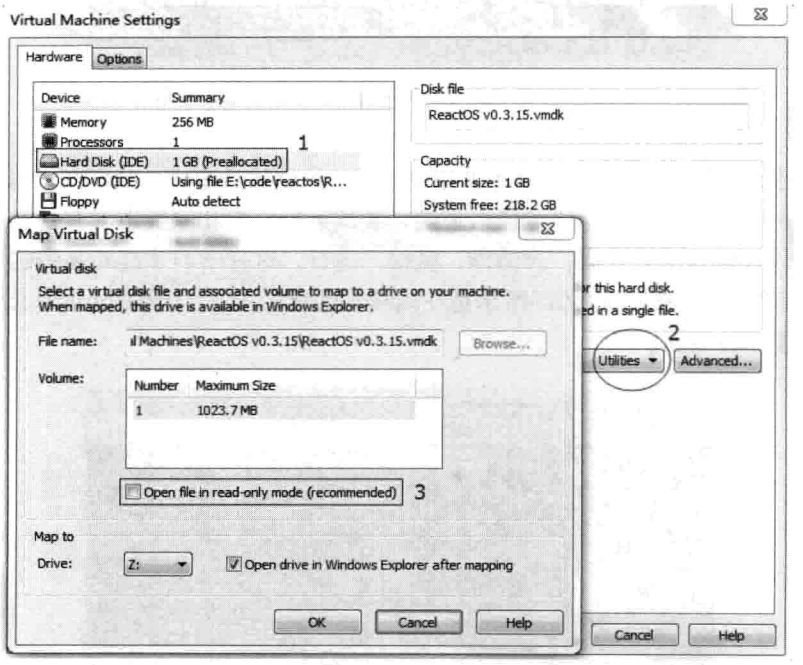


图 6

单击设备“Hard Disk (IDE)”，在右边的“Utilities”下拉菜单中选择“Map”，在弹出的窗口中取消勾选“Open file in read-only mode (recommended)”，单击“OK”按钮确认。这样我们就使用映射的方式打开了 ReactOS 系统盘。找到目录Z:\ReactOS\system32，把原文件 ntoskrnl.exe 改名为 ntoskrnl2.exe，方便以后我们需要时再改回去。复制我们的 ntoskrnl 项目生成的文件 ntoskrnl.exe（在ntos\Debug 目录下）到Z:\ReactOS\system32。复制完成后关闭窗口，再在“Utilities”下拉菜单中选择“Disconnect”，如果弹出提示框，单击“Force Disconnect”。这样就替换了 ntoskrnl.exe。如果需要可以通过同样的方式替换生成的启动型驱动程序 acpi.sys、pci.sys 和 disk.sys。说到这里，我们简单提一点，在第 2 章中，我们讲述到 LDFS 文件格式，我们也可以使用 NTFS 文件格式，映射 ntos.vmdk 到虚拟盘符后，格式

成 NTFS 分区就行。但是这样一来，我们要修改不少代码，MBR 代码需要重写，因为需要解析 NTFS 格式来找到 NTLDR。

替换完内核模块后，还需要开启 com_1 端口用于 WinDbg 连接通信。启动系统后，在屏幕提示选项中按上下键选择“ReactOS (Debug)”回车，如图 7所示。

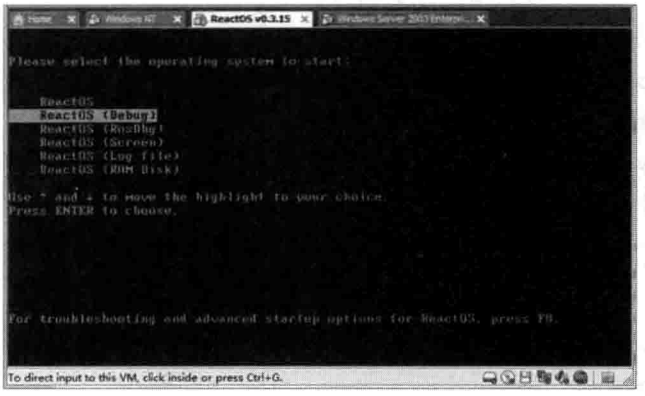


图 7

等待一小会，WinDbg 和 ReactOS 就连接上了。通过这种方式，我们在学习时可以进行扩展，对不同的模块，可以分块编译后替换，尝试运行，看和原版中的有什么差别，也可以按自己的设计修改部分代码，再编译、链接、运行。这样学习才能对系统的整体构架比较熟悉，理解也会比较深刻。图 8所示是在内核初始化阶段 1，函数 IoInitSystem 初始化启动型驱动后显示的界面。

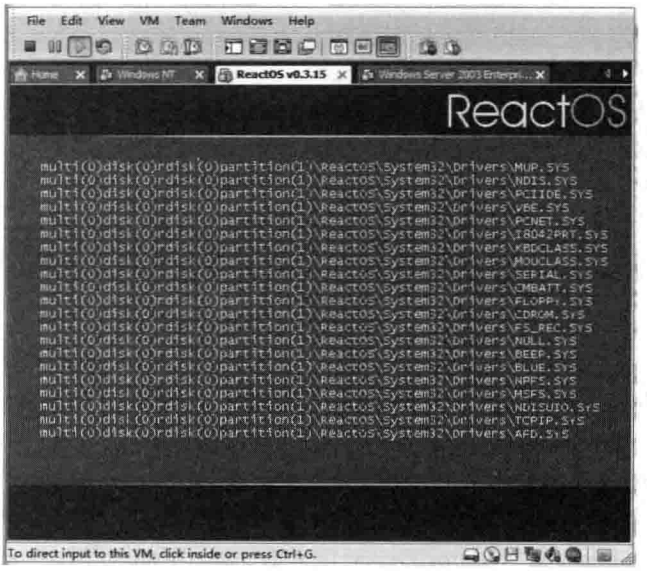


图 8

这里我们能观察到启动型（SERVICE_BOOT_START）驱动的加载和系统初始化阶段型（SERVICE_SYSTEM_START）驱动的加载。

阅读须知

在本书的某些章节中，有些函数相对较长，比如第 7 章讲述的查找对象名函数（ObpLookupObjectName）就占用了十几页，书中我们保留了全部的细节，是为了保持函数的完整性，而且一旦读者熟悉了这个函数，整个对象管理器的核心就一目了然了。

在将章节内容整理成书时，难免出现一些错误，有些章节虽然经过了多次修改，也仍然有不如意的地方，读者在阅读时要结合源码调试去理解，以免出现理解偏差。

本书配套资源下载地址：http://www.broadview.com.cn/25314。

参考资料

本书成书过程中主要参考了 *Windows Internals, 6th Edition*（Mark Russinovich、David A. Solomon 和 Alex Ionescu 合著，中文版为《深入解析 Windows 操作系统（第 6 版）》）和 *Intel 64 and IA-32 Architectures Software Developer's Manual*。此外，还参考了 ReactOS 官方网站（http://www.reactos.org）、微软网站（http://www.microsoft.com）、维基百科（http://en.wikipedia.org）及http://www.debuginfo.com/articles/debuginfomatch.html上的一些资料。

新书力荐

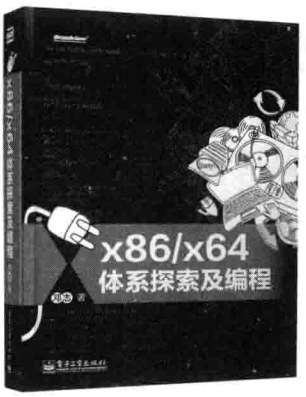
《Windows内核安全与驱动开发》
谭文 陈铭霖 等编著
2015年5月出版 敬请关注！

本书系《天书夜读》与《寒江独钓》的升级版本。

本书通篇介绍了在安全行业有着不同应用的各种驱动程序：键盘过滤、虚拟磁盘、文件系统过滤和多个层次的网络过滤。读者可以用它们开发保护密码、保护机密文件、监控网络、主动防御等各方面的信息安全软件。本书由多年从事安全行业的资深工程师亲自执笔，读者将体会到将理论和技术实际应用的乐趣。

试读地址：http://bbs.fishc.com/thread-56187-1-1.html。

邓志著作



《x86/x64体系探索及编程》
ISBN 978-7-121-18176-4
定价：119.00元

本书是对 Intel 手册所述处理器架构的探索和论证。全书分五大部分，对多个方面对处理器架构相关的知识进行了梳理介绍。书中每个章节都有相应的测试实验，所运行的实验例子都可以在真实的机器上执行。

通过阅读本书，读者应能培养自己动手实验的能力。如果再有一些 OS 方面的相关知识，基本上就可以写出自己简易的 OS 核心。

《处理器虚拟化技术》
ISBN 978-7-121-23019-6
定价：109.00元

本书针对在 Intel 处理器端的虚拟化技术（Intel Virtualization Technology for x86，即 Intel VT-x）进行全面讲解。在 Intel VT-x 技术下实现了 VMX（Virtual-Machine Extensions，虚拟机扩展）架构平台来支持对处理器的虚拟化管理。因此，VMX 架构是 Intel VT-x 技术的核心。本书内容围绕 VMX 架构实现细节展开全面讲解。但 Intel VT-d（Virtualization Technology for Directed I/O）和 Intel VT-c（Virtualization Technology for Connectivity）技术并不在本书的描述范围。同时，也不针对 AMD-v 技术进行讨论。

全书共分为 7 章，书的整体结构也较为规整，可读性比较强。本书共提供 14 个例子，对 VMX 架构的一些特色功能进行辅助讲解。