

原创经典， 威盛一线工程师倾力打造

深入驱动核心， 剖析操作系统底层运行机制

通过实例引导， 快速学习编译、安装、调试的方法

Broadview
www.broadview.com.cn

Windows

驱动开发技术详解

Windows Driver Development Internals

张帆 史彩成 等编著

珍藏版

- ◎ 从Windows最基本的两类驱动程序的编译、安装、调试入手讲解，非常容易上手
- ◎ 用实例详细讲解PCI、USB、虚拟串口、虚拟摄像头、SDIO等驱动程序的开发
- ◎ 归纳了多种调试驱动程序的高级技巧，如用WinDbg和VMWare软件对驱动进行源码级调试
- ◎ 介绍了多种实用的工具软件，如BusHound、IRPTrace、DebugView等
- ◎ 深入Windows操作系统的底层和内核，透析Windows驱动开发的本质



CD-ROM



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

Windows

驱动开发技术详解

Windows Driver Development Internals

张帆 史彩成 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由浅入深、循序渐进地介绍了 Windows 驱动程序的开发方法与调试技巧。本书共分 23 章，内容涵盖了 Windows 操作系统的基本原理、NT 驱动程序与 WDM 驱动程序的构造、驱动程序中的同步异步处理方法、驱动程序中即插即用功能、驱动程序的各种调试技巧等。同时，还针对流行的 PCI 驱动程序、USB 驱动程序、虚拟串口驱动程序、摄像头驱动程序、SDIO 驱动程序进行了详细的介绍，本书最大的特色在于每一节的例子都是经过精挑细选的，具有很强的针对性。力求让读者通过亲自动手实验，掌握各类 Windows 驱动程序的开发技巧，学习尽可能多的 Windows 底层知识。

本书适用于中、高级系统程序员，同时也可用做高校计算机专业操作系统实验课的补充教材。

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

Windows 驱动开发技术详解 / 张帆等编著. —北京: 电子工业出版社, 2008.7

ISBN 978-7-121-06846-1

I. W… II. 张… III. 窗口软件, Windows—驱动程序—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字 (2008) 第 080993 号

责任编辑: 高洪霞

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

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: 34.75 字数: 851 千字

印 次: 2008 年 7 月第 1 次印刷

印 数: 5000 册 定价: 65.00 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 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。

院士推荐



目前,电子系统设计广泛采用通用操作系统,达到降低系统的设计难度和缩短研发周期。实现操作系统与硬件快速信息交换是电子系统设计的关键。

通用操作系统硬件驱动程序的开发,编写者不仅需要精通硬件设备、计算机总线,而且需要 Windows 操作系统知识以及调试技巧。学习和掌握 Windows 硬件驱动程序的开发是电子系统设计人员必备的能力。

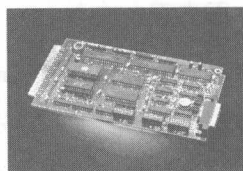
本书是作者结合教学和科研实践经验编写而成的,不仅详细介绍了 Windows 内核原理,并且介绍了编程技巧和应用实例,兼顾了在校研究生和工程技术人员的实际需求,对教学、生产和科研有现实的指导意义,是一本值得推荐的专著。

中国工程院院士

A handwritten signature in black ink, appearing to be '王珂' (Wang Ke).

2008年5月

自序



写这本书，是为了圆自己一个梦！

1. 你有这样的困惑吗？

你在学习 Windows 驱动程序开发的时候，有没有这样的感觉：觉得入门太难了；总有一大堆莫名其妙的术语，如“中断请求级别”、“派遣函数”、“线程上下文”、“完成例程”等；总能碰到很多诸如 PCI 总线、USB 总线等内容；还有那些无休止的死机、蓝屏等错误……

这可能让你感到很困惑。但这很正常，因为 Windows 驱动开发涉及 Windows 操作系统底层的很多知识，而且很多概念非常抽象，不容易理解。这对于入门人员，甚至有一定基础的开发者都有一定的困难。我也曾经有过和你们一样的经历，当然除了困惑之外，还有解决困惑之后的喜悦。

2. 我的经历

当我第一次接触 Windows 驱动开发时，就觉得非常吃力。那是在研究生一年级的時候，实验室在开发一个 PCI 总线视频采集卡，需要编写驱动程序来读取 PCI 卡上的数据。由于不熟悉 Windows 驱动程序，编译、安装等很简单的问题都困扰我很长时间。通过长时间的摸索，当我第一次用自己编写的驱动程序采集到 PCI 卡上的数据时，我感到非常兴奋。虽然几秒钟后，Windows 无情地蓝屏重启了，但我依然感觉很有成就感。那时候最喜欢做的事情，就是叫周围的同学“欣赏”设备管理器中我编写的设备。又经过很长时间，我才最终将蓝屏死机的原因找出，让驱动程序稳定地运行。

后来，我又开发了很多驱动程序，有 PCI 驱动、USB 驱动、摄像头驱动、SDIO 驱动。渐渐地，我发现驱动程序开发并没有想象中的那么困难。只要对驱动程序开发“入门”后，它就不再是一个神秘的事情了。

我还编写过一些 Linux 驱动程序，结果发现 Linux 设备驱动程序和 Windows 设备驱动程序有很多类似的地方。相比而言，Linux 驱动模型比较简单，加之 Linux 内核的源码是开放的，所以很多地方可以对照 Linux 内核源码进行学习。而 Windows 驱动程序模型比较复杂，其内核也没有提供源码，所以 Windows 驱动程序的编写相对困难一些。

3. 圆自己一个梦

回想当初自己学习 Windows 驱动开发的情景，感想颇多。各种各样的困难，完成一个驱动开发后的喜悦，为了找一本好的学习资料几乎翻遍了图书馆……这些至今都还深深地印在我的脑海里。

随着开发经验的积累和技术水平的提升，越来越想写本 Windows 驱动开发的书，以便向更多的人介绍 Windows 驱动程序的开发经验，使那些初学者快速入门，少走弯路，也能让已经有一定基础的人有所借鉴。这也算是圆我自己的一个梦吧。

当我把这个计划告诉我的老师史彩成时，他非常支持我，而且也愿意和我一起来完成这本书的写作。经过一年多的努力，我们终于完成了这个“大工程”，心中也自然非常喜悦。这无论是对我们，还是对渴望学习 Windows 驱动开发的人，都算是有了一个“交代”。

但愿这本书能够成为想致力于 Windows 驱动开发人员的良师益友，让你有所获益。我也相信，当你读完本书后应该已经能够编写大部分的 Windows 设备驱动程序了。

张 帆

前言

PREFACE

你是否想知道 USB 移动硬盘插入 PC 后，Windows 是如何识别的？
你是否想知道 Windows 是如何得到显卡中的数据的数据的？
你是否想知道什么导致了系统蓝屏死机？
你是否被老板或者导师逼着写一个 PCI、USB 等驱动程序，正感到无从下手？
你是否对 Windows 内核怀着强烈的好奇心？
如果你的回答为“是”，那么阅读本书将是最佳的选择！



上图是 Windows 操作系统的示意图。一般的 Windows 程序员都是编写应用程序或者用户 DLL，而不会对 Windows 底层有更深入的了解。而驱动程序位于操作系统的底层，它和内核紧密联系。另外，驱动程序直接操作硬件设备，但究竟如何操作，大部分程序员都不能清楚地讲出来。这些都使得驱动程序开发变得很神秘，仿佛都应该是编程高手的事情。

对于驱动程序开发，书店里很少能见到这方面的书籍。笔者在学习的时候尝到了各种苦头。为了帮助大家快速掌握驱动程序开发，笔者萌生了写一本书的想法。

本书的特点

1. 快速上手：为了让读者快速上手，笔者先给出两个驱动程序的例子。这两个例子分别代表 Windows 两类最基本的驱动程序，NT 式驱动程序和 WDM 式驱动程序。笔者非常详细地介绍了驱动程序编译、安装、调试的方法。编译驱动程序一般使用 build 工具，但是考虑到很多读者都是 VC 程序员，笔者特意介绍了如何用 VC 编译器编译驱动程序。

2. 内容翔实，实例丰富：本书详细地介绍了 PCI 驱动程序、USB 驱动程序、虚拟串口程序、虚拟摄像头程序、SDIO 驱动程序的开发，并辅以大量实例，使读者可以边学技术，边进行实践。

3. 介绍多种调试技巧：驱动程序由于运行在内核模式下，很难像普通应用程序那样可以方便地调试。尤其对于 VC 程序员来说，以前的那些调试技巧，很多都不能用了。另外，莫名其妙的“蓝屏死机”也会成为驱动程序开发人员的梦魇。笔者结合自己开发驱动

程序多年的经验，归纳了多种调试驱动程序的高级技巧。这些包括用 WinDbg 和 VMWare 软件对驱动进行源码级调试、用 WinDbg 调试蓝屏后的 Dump 文件等。

4. 灵活地使用一些工具：工欲善其事，必先利其器。很多工具软件会帮助我们更好地了解驱动程序内部的运行情况。本书将介绍很多实用的工具软件，如调试 USB 驱动程序的 BusHound 软件、查看 IRP 的 IRPTrace 软件、查看调试信息的 DebugView 软件、加载 NT 式驱动的 DriverMonitor 软件、加载 WDM 式驱动的 EzDriverInstaller 及查看设备对象的 DeviceTree 工具等。

5. 分析本质：本书对驱动程序的讨论不是仅停留在“表面”，更多地方是带领读者深入到操作系统的底层。本书对驱动程序涉及的操作系统中各个组件都有深入的介绍。另外，本书详细地介绍了驱动程序中的同步处理和异步处理。正确处理同步与异步，会使驱动程序更稳定，运行效率更高。

6. 探讨 Windows 内核：驱动程序和 Windows 的内核紧密相连。本书讲述了很多 Windows 内核的原理。由于 Windows 不是开源的操作系统，所以很少有书籍涉及 Windows 内核的原理。深入理解 Windows 内核的构造与原理，将更好地帮助程序员写出稳定的驱动程序。

本书的内容

本书由 23 章组成，内容分布如下：

入门篇	编译、安装方法 (1)	介绍 NT 式、WDM 式驱动程序的编译、安装方法
	驱动程序开发的基本方法 (2~7)	介绍驱动程序的基本概念、基本数据结构。介绍驱动程序中经常用到的内核函数。介绍驱动程序的入口函数、卸载函数、IRP 处理函数等
进阶篇	同步和异步处理 (8~9)	介绍驱动程序内部对同步操作请求和异步操作请求的处理。介绍如何编写同步和异步的 IRP 处理函数
	定时器 (10)	介绍两种内核模式下的定时器使用方法，另外还介绍了 4 种在内核模式下等待的方法
	驱动程序之间的调用 (11~12)	介绍驱动程序之间的调用方法
	即插即用和电源管理 (13~14)	介绍驱动程序中即插即用和电源管理功能。这些都是 WDM 驱动程序的重点内容
实用篇	各类硬件设备或者模拟设备的驱动程序 (15~20)	介绍几类硬件设备或者模拟设备的驱动程序。包括 USB 设备驱动、PCI 设备驱动、虚拟串口驱动、虚拟摄像头驱动、SDIO 设备驱动等。
提高篇	再论 IRP (21)	讨论一些高级的 IRP 处理方法
	过滤驱动程序 (22)	介绍如何编写过滤驱动程序
	高级调试技巧 (23)	介绍一些高级的驱动程序调试技巧

第 1 篇 入门篇

第 1 章 从两个最简单的驱动谈起 2

本章向读者呈现两个最简单的 Windows 驱动程序,一个是 NT 式的驱动程序,另一个是 WDM 式的驱动程序。这两个驱动程序没有操作具体的硬件设备,只是在系统里创建了虚拟设备。在随后的章节中,它们会作为基本驱动程序框架,被本书其他章节的驱动程序开发所复用。笔者将带领读者编写代码、编译、安装和调试程序。

1.1 DDK 的安装	2
1.2 第一个驱动程序 HelloDDK 的代码分析	3
1.2.1 HelloDDK 的头文件	4
1.2.2 HelloDDK 的入口函数	5
1.2.3 创建设备例程	6
1.2.4 卸载驱动例程	8
1.2.5 默认派遣例程	9
1.3 HelloDDK 的编译和安装	9
1.3.1 用 DDK 环境编译 HelloDDK	9
1.3.2 用 VC 集成开发环境编译 HelloDDK	11
1.3.3 HelloDDK 的安装	14
1.4 第二个驱动程序 HelloWDM 的代码分析	16
1.4.1 HelloWDM 的头文件	16
1.4.2 HelloWDM 的入口函数	17
1.4.3 HelloWDM 的 AddDevice 例程	18
1.4.4 HelloWDM 处理 PNP 的回调函数	20
1.4.5 HelloWDM 对 PNP 的默认处理	22
1.4.6 HelloWDM 对 IRP_MN_REMOVE_DEVICE 的处理	23
1.4.7 HelloWDM 对其他 IRP 的回调函数	23
1.4.8 HelloWDM 的卸载例程	24
1.5 HelloWDM 的编译和安装	24
1.5.1 用 DDK 编译环境编译 HelloWDM	24
1.5.2 HelloWDM 的编译过程	25
1.5.3 安装 HelloWDM	25
1.6 小结	29

第2章 Windows 操作驱动的基本概念

31

驱动程序被操作系统加载在内核模式下,它与 Windows 操作系统内核的其他组件进行密切交互。本章主要介绍 Windows 操作系统内核的基本概念,同时还介绍应用程序和驱动程序之间的通信方法。

2.1	Windows 操作系统概述	31
2.1.1	Windows 家族	31
2.1.2	Windows 特性	32
2.1.3	用户模式和内核模式	34
2.1.4	操作系统与应用程序	36
2.2	操作系统分层	37
2.2.1	Windows 操作系统总体架构	37
2.2.2	应用程序与 Win32 子系统	38
2.2.3	其他环境子系统	40
2.2.4	Native API	41
2.2.5	系统服务	41
2.2.6	执行程序组件	42
2.2.7	驱动程序	44
2.2.8	内核	44
2.2.9	硬件抽象层	45
2.2.10	Windows 与微内核	45
2.3	从应用程序到驱动程序	46
2.4	小结	48

第3章 Windows 驱动编译环境配置、安装及调试

49

本章将带领读者一步步对驱动程序进行编译、安装和简单的调试工作。这些步骤虽然简单,但往往困惑着初次接触驱动程序的开发者。

3.1	用 C 语言还是用 C++ 语言	49
3.1.1	调用约定	50
3.1.2	函数的导出名	52
3.1.3	运行时函数的调用	53
3.2	用 DDK 编译环境编译驱动程序	54
3.2.1	编译版本	55
3.2.2	nmake 工具	55
3.2.3	build 工具	56
3.2.4	makefile 文件	57
3.2.5	dirs 文件	58
3.2.6	sources 文件	58
3.2.7	makefile.inc 文件	59
3.2.8	build 工具的环境变量	60

3.2.9	build 工具的命令行参数	61
3.3	用 VC 编译驱动程序	62
3.3.1	建立驱动程序工程	62
3.3.2	修改编译选项	62
3.3.3	修改链接选项	63
3.3.4	其他修改	64
3.3.5	VC 编译小结	65
3.4	查看调试信息	66
3.4.1	打印调试语句	66
3.4.2	查看调试语句	67
3.5	手动加载 NT 式驱动	68
3.6	编写程序加载 NT 式驱动	68
3.6.1	SCM 组件和 Windows 服务	69
3.6.2	加载 NT 驱动的代码	71
3.6.3	卸载 NT 驱动的代码	74
3.6.4	实验	76
3.7	WDM 式驱动的加载	78
3.7.1	WDM 的手动安装	78
3.7.2	简单的 INF 文件剖析	79
3.8	WDM 设备安装在注册表中的变化	81
3.8.1	硬件子键	81
3.8.2	类子键	83
3.8.3	服务子键	85
3.9	小结	86

第 4 章 驱动程序的基本结构

87

本章首先对 Windows 驱动程序的两个重要数据结构进行介绍,分别是驱动对象和设备对象数据结构。另外还要介绍 NT 驱动程序和 WDM 驱动程序的入口函数、卸载例程、各种 IRP 派遣上函数等。

4.1	Windows 驱动程序中重要的数据结构	87
4.1.1	驱动对象 (DRIVER_OBJECT)	87
4.1.2	设备对象 (DEVICE_OBJECT)	89
4.1.3	设备扩展	91
4.2	NT 式驱动的基本结构	92
4.2.1	驱动加载过程与驱动入口函数 (DriverEntry)	92
4.2.2	创建设备对象	95
4.2.3	DriverUnload 例程	97
4.2.4	用 WinObj 观察驱动对象和设备对象	98
4.2.5	用 DeviceTree 观察驱动对象和设备对象	101
4.3	WDM 式驱动的基本结构	102

4.3.1	物理设备对象与功能设备对象	102
4.3.2	WDM 驱动的入口程序	104
4.3.3	WDM 驱动的 AddDevice 例程	105
4.3.4	DriverUnload 例程	107
4.3.5	对 IRP_MN_REMOVE_DEVICE IRP 的处理	108
4.3.6	用 Device Tree 查看 WDM 设备对象栈	109
4.4	设备的层次结构	110
4.4.1	驱动程序的垂直层次结构	111
4.4.2	驱动程序的水平层次结构	112
4.4.3	驱动程序的复杂层次结构	112
4.5	实验	114
4.5.1	改写 HelloDDK 查看驱动结构	114
4.5.2	改写 HelloWDM 查看驱动结构	116
4.6	小结	117

第 5 章 Windows 内存管理 118

本章围绕着驱动程序中的内存操作进行了介绍。在驱动程序开发中，首先要注意分页内存和非分页内存的使用。同时，还需要区分物理内存地址和虚拟内存地址这两个概念。

5.1	内存管理概念	118
5.1.1	物理内存概念 (Physical Memory Address)	118
5.1.2	虚拟内存地址概念 (Virtual Memory Address)	119
5.1.3	用户模式地址和内核模式地址	120
5.1.4	Windows 驱动程序和进程的关系	121
5.1.5	分页与非分页内存	122
5.1.6	分配内核内存	123
5.2	在驱动中使用链表	124
5.2.1	链表结构	124
5.2.2	链表初始化	125
5.2.3	从首部插入链表	126
5.2.4	从尾部插入链表	126
5.2.5	从链表删除	127
5.2.6	实验	129
5.3	Lookaside 结构	130
5.3.1	频繁申请内存的弊端	130
5.3.2	使用 Lookaside	130
5.3.3	实验	132
5.4	运行时函数	133
5.4.1	内存间复制 (非重叠)	133
5.4.2	内存间复制 (可重叠)	134
5.4.3	填充内存	134

5.4.4	内存比较	135
5.4.5	关于运行时函数使用的注意事项	135
5.4.6	实验	137
5.5	使用 C++ 特性分配内存	137
5.6	其他	139
5.6.1	数据类型	139
5.6.2	返回状态值	140
5.6.3	检查内存可用性	142
5.6.4	结构化异常处理 (try-except 块)	142
5.6.5	结构化异常处理 (try-finally 块)	144
5.6.6	使用宏需要注意的地方	146
5.6.7	断言	147
5.7	小结	147

第 6 章 Windows 内核函数

148

本章介绍了 Windows 内核模式下的一些常用内核函数, 这些函数在驱动程序的开发中将会经常用到。

6.1	内核模式下的字符串操作	148
6.1.1	ASCII 字符串和宽字符串	148
6.1.2	ANSI_STRING 字符串与 UNICODE_STRING 字符串	149
6.1.3	字符初始化与销毁	151
6.1.4	字符串复制	152
6.1.5	字符串比较	153
6.1.6	字符串转化成大写	154
6.1.7	字符串与整型数字相互转换	155
6.1.8	ANSI_STRING 字符串与 UNICODE_STRING 字符串相互转换	157
6.2	内核模式下的文件操作	158
6.2.1	文件的创建	158
6.2.2	文件的打开	161
6.2.3	获取或修改文件属性	163
6.2.4	文件的写操作	166
6.2.5	文件的读操作	167
6.3	内核模式下的注册表操作	169
6.3.1	创建关闭注册表	170
6.3.2	打开注册表	172
6.3.3	添加、修改注册表键值	173
6.3.4	查询注册表	175
6.3.5	枚举子项	178
6.3.6	枚举子键	180
6.3.7	删除子项	182
6.3.8	其他	183

6.4 小结	185
--------	-----

第7章 派遣函数 186

本章重点介绍了驱动程序中的处理 IRP 请求的派遣函数。所有对设备的操作最终将转化为 IRP 请求, 这些 IRP 请求会被传送到派遣函数处理。

7.1 IRP 与派遣函数	186
7.1.1 IRP	186
7.1.2 IRP 类型	188
7.1.3 对派遣函数的简单处理	188
7.1.4 通过设备链接打开设备	190
7.1.5 编写一个更通用的派遣函数	191
7.1.6 跟踪 IRP 的利器 IRPTrace	193
7.2 缓冲区方式读写操作	196
7.2.1 缓冲区设备	196
7.2.2 缓冲区设备读写	197
7.2.3 缓冲区设备模拟文件读写	200
7.3 直接方式读写操作	203
7.3.1 直接读取设备	204
7.3.2 直接读取设备的读写	205
7.4 其他方式读写操作	207
7.4.1 其他方式设备	207
7.4.2 其他方式读写	208
7.5 IO 设备控制操作	209
7.5.1 DeviceIoControl 与驱动交互	209
7.5.2 缓冲内存模式 IOCTL	210
7.5.3 直接内存模式 IOCTL	212
7.5.4 其他内存模式 IOCTL	214
7.6 小结	216

第2篇 进阶篇

第8章 驱动程序的同步处理 218

本章介绍了驱动程序中常用的同步处理办法, 并且将内核模式下的同步处理方法和用户模式下的同步处理方法做了比较。另外, 本章还介绍了中断请求级、自旋锁等同步处理机制。

8.1 基本概念	218
8.1.1 问题的引出	218
8.1.2 同步与异步	219
8.2 中断请求级	219
8.2.1 中断请求 (IRQ) 与可编程中断控制器 (PIC)	220
8.2.2 高级可编程控制器 (APIC)	221

8.2.3	中断请求级 (IRQL)	221
8.2.4	线程调度与线程优先级	222
8.2.5	IRQL 的变化	223
8.2.6	IRQL 与内存分页	223
8.2.7	控制 IRQL 提升与降低	224
8.3	自旋锁	224
8.3.1	原理	224
8.3.2	使用方法	225
8.4	用户模式下的同步对象	225
8.4.1	用户模式的等待	226
8.4.2	用户模式开启多线程	226
8.4.3	用户模式的事件	227
8.4.4	用户模式的信号灯	229
8.4.5	用户模式的互斥体	230
8.4.6	等待线程完成	232
8.5	内核模式下的同步对象	232
8.5.1	内核模式下的等待	232
8.5.2	内核模式下开启多线程	234
8.5.3	内核模式下的事件对象	236
8.5.4	驱动程序与应用程序交互事件对象	237
8.5.5	驱动程序与驱动程序交互事件对象	239
8.5.6	内核模式下的信号灯	240
8.5.7	内核模式下的互斥体	241
8.5.8	快速互斥体	243
8.6	其他同步方法	244
8.6.1	使用自旋锁进行同步	245
8.6.2	使用互锁操作进行同步	247
8.7	小结	249

第9章 IRP 的同步

250

本章详细地介绍了 IRP 的同步处理方法和异步处理方法。另外,本章还介绍了 StartIO 例程、中断服务例程、DPC 服务例程。

9.1	应用程序对设备的同步异步操作	250
9.1.1	同步操作与异步操作原理	250
9.1.2	同步操作设备	252
9.1.3	异步操作设备(方式一)	253
9.1.4	异步操作设备(方式二)	254
9.2	IRP 的同步完成与异步完成	256
9.2.1	IRP 的同步完成	256
9.2.2	IRP 的异步完成	257

9.2.3 取消 IRP	262
9.3 StartIO 例程	264
9.3.1 并行执行与串行执行	264
9.3.2 StartIO 例程	265
9.3.3 示例	267
9.4 自定义的 StartIO	270
9.4.1 多个串行化队列	270
9.4.2 示例	271
9.5 中断服务例程	273
9.5.1 中断操作的必要性	273
9.5.2 中断优先级	274
9.5.3 中断服务例程 (ISR)	274
9.6 DPC 例程	275
9.6.1 延迟过程调用例程 (DPC)	275
9.6.2 DpcForISR	275
9.7 小结	276

第 10 章 定时器

277

本章总结了在内核模式下的四种等待方法，读者可以利用这些方法灵活地用在自己的驱动程序中。最后本章还介绍了如何对 IRP 的超时情况进行处理。

10.1 定时器实现方式一	277
10.1.1 I/O 定时器	277
10.1.2 示例代码	278
10.2 定时器实现方式二	280
10.2.1 DPC 定时器	280
10.2.2 示例代码	282
10.3 等待	284
10.3.1 第一种方法：使用 KeWaitForSingleObject	284
10.3.2 第二种方法：使用 KeDelayExecutionThread	285
10.3.3 第三种方法：使用 KeStallExecutionProcessor	285
10.3.4 第四种方法：使用定时器	286
10.4 时间相关的其他内核函数	286
10.4.1 时间相关函数	286
10.4.2 示例代码	288
10.5 IRP 的超时处理	289
10.5.1 原理	289
10.5.2 示例代码	289
10.6 小结	291

第 11 章 驱动程序调用驱动程序

292

本章主要介绍了如何在驱动程序中调用其他驱动程序。比较简单的方法是将被调用的驱动程序以文件的方式操作。比较高级的方法是构造各种 IRP，并将这些 IRP 传送到被调用的驱动程序中。

11.1 以文件句柄形式调用其他驱动程序	292
11.1.1 准备一个标准驱动	292
11.1.2 获得设备句柄	294
11.1.3 同步调用	295
11.1.4 异步调用方法一	297
11.1.5 异步调用方法二	299
11.1.6 通过符号链接打开设备	301
11.2 通过设备指针调用其他驱动程序	303
11.2.1 用 IoGetDeviceObjectPointer 获得设备指针	304
11.2.2 创建 IRP 传递给驱动的派遣函数	305
11.2.3 用 IoBuildSynchronousFsdRequest 创建 IRP	306
11.2.4 用 IoBuildAsynchronousFsdRequest 创建 IRP	308
11.2.5 用 IoAllocateIrp 创建 IRP	311
11.3 其他方法获得设备指针	314
11.3.1 用 ObReferenceObjectByName 获得设备指针	314
11.3.2 剖析 IoGetDeviceObjectPointer	317
11.4 小结	318

第 12 章 分层驱动程序

319

本章主要介绍了分层驱动的概念。分层驱动可以将功能复杂的驱动程序分解为多个功能简单的驱动程序。多个分层的驱动程序形成一个设备堆栈，IRP 请求首先发送到设备堆栈的顶层，然后依次穿越每层的设备堆栈，最终完成 IRP 请求。

12.1 分层驱动程序概念	319
12.1.1 分层驱动程序的概念	319
12.1.2 设备堆栈与挂载	321
12.1.3 I/O 堆栈	322
12.1.4 向下转发 IRP	323
12.1.5 挂载设备对象示例	324
12.1.6 转发 IRP 示例	325
12.1.7 分析	326
12.1.8 遍历设备栈	327
12.2 完成例程	330
12.2.1 完成例程概念	330
12.2.2 传播 Pending 位	332
12.2.3 完成例程返回 STATUS_SUCCESS	333