

The Windows NT Device Driver Book: A Guide for Programmers

CMP

计算机网络基础
与应用系列丛书

由经验丰富的
专家撰写

内容丰富，通
俗易懂

Windows NT
4.0设备驱动程
序的专著

(美) Art Baker 著
科欣翻译组 译

Windows NT 设备驱动程序 设计指南

机械工业出版社

西蒙与舒斯特国际出版公司

计算机网络基础与应用系列丛书

Windows NT 设备驱动程序设计指南

(美) Art Baker 著

科欣翻译组 译

机械工业出版社
西蒙与舒斯特国际出版公司

本书是 Windows NT 设备驱动程序设计的专著。全书共有 18 章,第 1~5 章提供编写驱动程序所需的基本知识,包括 Windows NT 驱动程序体系结构的一般介绍,有关硬件的简要说明,以及 NT I/O 管理器和它的数据结构较详细的讨论。最后提出内核模式编程的一般指南和技术。第 6~13 章是本书的核心,介绍了编写 NT 内核模式设备驱动程序的所有详细信息,并讨论了全双工驱动程序的体系结构、超时的处理,以及设备错误的日志记录。第 14~15 章讨论 Windows NT 支持的替代驱动程序体系结构,包括在驱动程序中内核模式线程的使用和高层驱动程序。第 16 章讨论 BUILD 实用程序,第 17 章讨论驱动程序的测试和调试的各个方面,包括如何分析崩溃转储和如何使 WINDBG 真正地工作。第 18 章讨论驱动程序性能和如何把驱动程序连接到 NT 的性能监视机制中的关键问题。

本书内容全面实用,从理论到实际,由浅到深。书中提供了大量的设备驱动程序的范例代码,帮助读者理解和实践。本书可供广大计算机工作者和软件设计人员参考。

Art Baker: The Windows NT Device Driver Book: A Guide for Programmers
Authorized translation from the English language edition published by Prentice Hall, Inc.
Copyright 1997 by Prentice Hall, Inc.
All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签,无标签者不得销售。

版权所有,翻印必究。

本书版权登记号: 图字: 01-97-1254

图书在版编目 (CIP) 数据

Windows NT 设备驱动程序设计指南 / (美) 贝克尔 (Baker, A.) 著; 科欣翻译组译. - 北京: 机械工业出版社, 1997

(计算机网络基础与应用系列丛书)

书名原文: The Windows NT Device Driver Book: A Guide for Programmers

ISBN 7-111-06030-X

I . W… II . ①贝… ②科… III . 计算机网络-操作系统 (软件)-程序设计-指南
IV . TP393

中国版本图书馆 CIP 数据核字 (97) 第 23235 号

出 版 人: 马九荣 (北京市百万庄大街 22 号 邮政编码 100037)

北京昌平第二印刷厂 印刷·新华书店北京发行所发行

1997 年 12 月第 1 版·1998 年 4 月第二次印刷

787mm×1092mm 1/16·26 印张·622 千字

印数: 50 01—80 00 册

定价: 45.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

本书的目的是介绍如何编写、安装和调试 Windows NT 的内核模式设备驱动程序。如果读者正在进行一个 NT 驱动程序的设计或编程，或者正将一个已存在的驱动程序从另一个操作系统移植过来，本书是 Microsoft DDK 文档十分有用的伴侣。

如果读者只是需要了解 Windows NT 系统运行的一些内幕，特别是 I/O 子系统，本书也提供这方面的一些内容。读者或许正在决定 NT 是否是某个特定用途的理想平台，或者正在学习操作系统，想看到理论如何运用到实际工作中。

当然，我们不能忽视好奇心的力量。例如，我们开车经过一起交通事故现场时，我们会情不自禁地降低车速；同样，好奇心会驱使我们从书架上取下一本书来。

1. 读者预先具备的知识

在阅读本书之前，要求读者要具有以下几方面的知识：

(1) 需要具有所有基本的 Windows NT 用户技能，如登录和运行各种实用程序，因为驱动程序安装要求管理员级的特权，如果不知道如何使用系统，就会使事情变糟。

(2) 需要熟练的 C 语言编程技巧。作者努力避免在范例代码中使用各种“花招”，但是读者仍必须能够读懂这些范例程序。

(3) Win32 用户模式程序设计的经验是很有帮助的，但它不是必须的。如果读者没有使用过 Win32 API，可以浏览一下《Win32 程序员参考手册》第 2 卷，它介绍了各种系统服务。阅读有关 I/O 原语（CreateFile, ReadFile, WriteFile 和 DeviceIoControl）和线程模型的章节。关于 Win32 程序设计的其他参考书，参见本书后面的参考文献。

(4) 为了写驱动程序，要了解硬件的一些知识。如果已经有一些使用硬件的经验，那将是很帮助的，但是如果没有这样的经验，第 2 章将提供基本的介绍。本书后面的参考文献也将提供硬件信息的其他参考书。

2. 本书内容

任何作者不得不做出的一个最困难的选择是决定要写什么和不写什么。一般来说，本书集中于对内核模式驱动程序开发很关键的一些核心问题，并努力提供足够的背景信息，使得读者能够读懂 NT DDK 所带的范例代码，并为自己的驱动程序做出明智的设计选择。

本书的编排从理论到实际，由浅及深。全书的内容包括：

(1) 第 1~5 章。本书第 1 部分提供编写驱动程序所需的基本知识，包括 Windows NT 驱动程序体系结构的一般介绍，有关硬件的简要说明，以及 NT I/O 管理器和它的数据结构的较详细的讨论。最后提出内核模式编程的一些一般指南和技术。

(2) 第 6~13 章。这 8 章是本书的核心，介绍了编写 NT 内核模式设备驱动程序的所有详细信息，并讨论了全双工驱动程序的体系结构、超时的处理，以及设备错误的日志记录。如果读者对 NT 的驱动程序体系结构不甚熟悉，应该按顺序阅读这几章。

(3) 第 14~15 章。这两章讨论 Windows NT 支持的替代驱动程序体系结构, 包括在驱动程序中内核模式线程的使用和高层驱动程序。

(4) 第 16~18 章。本书的最后一部分涉及编写 NT 驱动程序的各种实际细节。第 16 章讨论 BUILD 实用程序, 第 17 章讨论驱动程序的测试和调试的各个方面, 包括如何分析崩溃转储和如何使 WINDBG 真正地工作。如果在阅读本书时正在实际写一个驱动程序, 可以不按顺序阅读这几章。第 18 章讨论驱动程序性能和如何把驱动程序连接到 NT 的性能监视机制中的关键问题。

(5) 附录

附录涉及作者在教学中学员经常问及的各种问题。附录 A 涉及建立驱动程序开发环境的要求。

附录 B 含有读者最可能遇到的错误检查代码的清单, 以及它们的各个参数的描述。与第 17 章中的材料一起使用, 这个附录中的信息有助于查找系统故障的原因。

3. 本书尚未包括的内容

由于某些原因, 本书尚未包括以下几方面的内容: 一些主题太庞大的内容, 不可能包括在本书中; 有些内容则是解决驱动程序设计者中极少一部分人的需要, 不具普遍性; 驱动程序开发的某些方面是 Microsoft 不支持的。具体来说, 本书中不包含下述内容。

(1) 文件系统驱动程序。在本书开印时, Microsoft 仍然没有公布 NT 文件系统驱动程序的任何开发人员工具包。事实上, 在 Microsoft 内部似乎有非常大的阻力。在这种状况改变之前, 对文件系统驱动程序的体系结构是不可能有多少东西可说的。

(2) 网卡和网络协议驱动程序。NDIS 和 TDI 驱动程序都是非常大的话题, 它们自己都可以用一本书的篇幅来讨论, 但是在本书中没有足够的空间来包括这方面的内容。值得欣慰的是, 本书中的材料能够提供了解 NDIS/TDI 框架内发生的事情所需要的许多背景知识。

(3) SCSI 小端口驱动程序和类驱动程序。尽管 SCSI HBA 小端口驱动程序是重要的系统组件, 但是实际写这些驱动程序的人还是极少的, 因此, SCSI 小端口的唯一参考是第 1 章中的综述材料。

作者本希望在这本书中包括 SCSI 类驱动程序的讨论, 然而遗憾的是, 没有更多的时间来写这方面的内容。第 15 章中开发中间驱动程序的材料提供了必要的大量背景知识, 在那里, 可以看到 NT DDK 所带的 CD-ROM 范例 SCSI 类驱动程序。

(4) 视频、显示和打印机驱动程序。这是作者不得不在写这类驱动程序的人数和可用于完成这本书的时间之间做出取舍的另一方面。不幸的是, 视频和硬拷贝设备的图形驱动程序这次被舍去了, 或许在以后某个时候本书的再版中包括它。

(5) 虚拟 DOS 设备驱动程序。从作者的观点来看, 在 Windows NT 下运行 16 位 MS-DOS 和 Windows 应用程序的最好方法是把源代码移植到 Win32。在任何情况下, Microsoft 的文献资料都提供了写 VDD 的详细说明, 所以在这里没有包括这方面的内容。

(6) 范例代码。在本书中, 包括了许多驱动程序范例代码。本书另附一套软盘, 含有这些驱动程序的完整程序, 在软盘上为每一章创建了独立的目录, 并为驱动程序中的各个组件创建了合适的子目录。

(7) 编码风格。因为本书的目的是指南, 所以采用了几个办法提高范例代码的清晰性。

首先,本书采用了一种避免各种技巧的编程风格,其中的一些例子可能只用几行代码。

同样为了清晰,在例子中只保留了基础的东西。例如,大多数驱动程序不含有任何错误日志记录和调试代码,尽管实际的驱动程序应该包含这些东西。这些话题有专门的章节进行讨论,把这些代码加入其他驱动程序范例中不会是很难的事情。

(8) 命名约定。本书中出现的几乎所有范例驱动程序都叫做“XXDRIVER”(唯有第 15 章中的高层驱动程序例外,它的名字是“YYDRIVER”)。这样做的好处是方便交换不同范例的各个部分,并且减少了在使用这些驱动程序时加到 Registry 中的数据量。

在任何特定的驱动程序中,采用了增加前缀的方法,把 Xx 加到任何驱动程序定义的函数的名字之前。类似地,设备寄存器、驱动程序结构和常量也有前缀 XX_。这样很清楚地看到哪些东西是自己必须写的,哪些来自 Microsoft。

(9) 平台依赖性。这里要指出的是,这些范例的目标是在 Intel 80x86 平台上运行。特别是,驱动程序都假定设备寄存器位于 I/O 空间中,而不是内存映射的。这对于纠正一些编码和修改每个驱动程序的硬件特定的头文件是比较容易的。

(10) 生成和运行范例程序。如果打算开展 Windows NT 的驱动程序开发,需要以下两个工具:①得到对 Microsoft Developer Network CD 的二级订阅。这是 NT DDK 和 Win32 SDK 的唯一来源;②另外,还需要 C 编译器。作者选择使用 Microsoft 编译器开发和调试本书中的所有代码。

关于设置驱动程序开发环境的更多信息,参见附录 A。

机械工业出版社

最新计算机图书书目

序号	书 名	定价	书号	序号	书 名	定价	书号
一、当代计算机职业培训系列教程				6	Java 快速查询手册	16.00	05578
1	个人计算机第一本入门书	26.00	05184	7	JavaScript 快速查询手册	12.00	05579
2	DOS6.XX 轻松入门	28.00	05185	8	HTML 3.2 快速查询手册	12.00	05718
3	Windows 95 轻松入门	25.00	05250	9	微机常用软件使用手册	11.00	05775
4	Excel for Windows 95 轻松入门	30.00	05251	三、最新多媒体电脑实用系列丛书			
5	Word for Windows 95 轻松入门	34.00	05248	1	多媒体电脑内存管理技巧	28.00	05181
6	Visual Basic 程序设计轻松入门	42.00	05187	2	多媒体电脑超级 CD-ROM 技术大全	28.00	05255
7	Visual Basic 4 程序设计轻松入门	27.00	05297	3	多媒体电脑使用大全	52.00	05253
8	C 语言程序设计轻松入门	39.00	05303	4	多媒体电脑虚拟现实技巧	64.00	05254
9	C++ 程序设计轻松入门	58.00	05298	5	多媒体电脑教育综艺大观	26.00	05257
10	计算机网络轻松入门	30.00	05186	6	多媒体电脑影视技术大全	49.00	05256
11	Internet for Windows 轻松入门	15.00	05397	7	光盘运作与应用技巧	15.00	05182
12	Netware 4.1 轻松入门	26.00	05467	8	中文 Windows 95 使用技巧	39.00	05183
13	Visual FoxPro 5 精简易懂	39.00	05658	9	看图例学多媒体制作	30.00	05662
14	Unix 轻松入门	25.00	05651	10	多媒体电脑选配与使用	24.00	05616
15	跟 CD-ROM 学 JAVA (附光盘)	45.00	05659	四、计算机图解大全系列			
16	按实例学 Delphi 2	38.00	05464	1	个人计算机图解大全	42.00	05299
17	中文版 Microsoft Office 97 轻松入门	29.00	05758	2	中文版 Excel for Windows 95 图解大全	42.00	05302
18	中文版 Microsoft Word 97 轻松入门	28.00	05755	3	中文版 Word for Windows 95 图解大全	42.00	05300
19	中文版 Microsoft Excel 97 轻松入门	29.00	05756	4	中文版 Microsoft Office for Windows 95 图解大全	42.00	05301
20	中文版 Microsoft Access 97 轻松入门	29.00	05752	5	Internet 图解大全	42.00	05305
21	中文版 Microsoft PowerPoint 97 轻松入门	32.00	05757	五、计算机软件开发与程序设计系列丛书			
22	Microsoft FrontPage 97 轻松入门	25.00	05754	1	The Microsoft Windows 95 开发人员指南	86.00	05459
23	HTML 3.2 轻松入门	25.00	05759	2	Visual Basic 4 开发人员指南	73.00	05461
二、快速学习手册				3	Visual FoxPro 3.0 开发指南	119.00	05465
1	Office 97 快速学习手册 (上— Word, PowerPoint)	24.00	05657	4	Windows 95 API 开发人员指南	74.00	05463
2	Office 97 快速学习手册 (中— Outlook)	9.00	05760	5	Visual Basic 4.0 API 程序设计	83.00	05460
3	Office 97 快速学习手册 (下— Excel, Access)	23.00	05663	6	用 Visual Basic 4 开发客户机/服务器应用	80.00	05655
4	Lotus Notes 4.5 快速学习手册	16.00	05661	7	Java 开发人员指南	64.00	05462
5	ActiveX 控件快速学习手册	16.00	05761	8	Visual C++ 4 开发人员指南	64.00	05656
				9	I/O 接口程序设计入门与应用	28.00	05453

序号	书 名	定价	书号	序号	书 名	定价	书号
10	精通 VISUAL BASIC 4.0 多媒体程序设计	15.00	05466	八、计算机网络实作与应用系列丛书			
11	Visual C++ 开发工具实用指南	68.00	05791	1	局域网解析	25.00	05590
12	Visual Basic 5 开发人员指南	66.00	05811	2	微软网实用指南	21.00	05589
13	ActiveX 开发人员指南	52.00	05580	3	用 Java 开发 Intranet 应用	42.00	05584
14	C/C++ 程序设计大全	89.00	05816	4	Windows NT Workstation 4.0 使用指南	36.00	05591
15	Borland C++ 环境下 Windows 3.1—95 编程技术及实例	96.00	05396	5	Novell 认证网络工程师 (CNE4) 基础学习指导	29.00	05585
六、Internet 实务系列丛书				6	Netware 命令大全	36.00	05454
1	快快乐乐畅游 Internet	34.00	05468	7	网络多媒体开发与应用	48.00	05810
2	WWW 文件设计-HTML 语言实务	29.00	05455	8	Windows NT 4.0 配置安装手册	53.00	05728
3	Java/VRML 设计大全	32.00	05457	9	看图例活学活用 Windows NT 网络	43.00	05828
4	Web 动态技术入门	23.00	05456	九、WEB 专家工作室系列			
5	Java 轻松上手	26.00	05473	1	商用 Web 页面创建指南 (之一) (附光盘)	61.00	05582
6	Internet 信息查询技巧	25.00	05592	2	Microsoft FrontPage 97 实战解析 (之二) (附光盘)	65.00	05753
7	WWW 上站与架站实务	20.00	05470	3	ActiveX & VBScript 实战解析 (之三)	43.00	05749
8	Mosaic 与 WWW	10.00	05469	4	JavaScript 实战解析 (之四)	33.00	05327
9	Archie-Internet 标题检索工具	10.00	05471	十、巧学活用—从新手到专家系列			
10	Web 站点创建与信息发布	16.00	05751	1	巧学活用 Microsoft Office 97 (中文版)	33.00	05829
11	Netscape Navigator Gold 3.0 与网络资源搜寻	20.00	05472	2	巧学活用 Microsoft Word 97 (中文版)	33.00	05804
12	CGI 编程指南	33.00	05717	3	巧学活用 Microsoft Excel 97 (中文版)	29.00	05838
13	ISDN 与 Internet	15.00	05707	十一、程序设计大全系列			
14	Internet 与 WWW 常见问题解答	23.00	05723	1	Java 语言程序设计大全	83.00	05583
15	组建与管理 Web 服务系统	28.00	05708	十二、其 他			
16	Internet 问答	29.00	05779	1	AutoCAD R13 技巧与捷径 1000	40.00	05509
七、最新 Internet 技术基础与应用系列丛书				2	AutoCAD R13 中文版使用指南	36.00	05660
1	Active X 使用指南	29.00	05581	3	微软客户机/服务器应用集锦 (二)	16.00	05605
2	Netscape 3 使用指南	25.00	05586	4	3D Studio MAX 设计指南	29.00	05714
3	Java Workshop 使用指南	32.00	05588	5	微型计算机常用软件使用手册 (续集) 中文之星篇	11.00	05775
4	Web 多媒体开发指南 (附光盘)	75.00	05698	特版精品系列丛书			
5	Web 数据库的建立与管理	73.00	05713	1	Visual Foxpro 5 开发使用手册	66.00	05811
6	Visual J++ 程序设计	32.00	05724				
7	WWW Plug-Ins 技术开发指南	35.00	05748				
8	Visual J++ 入门与应用	30.00	05719				
9	Windows NT Workstation 4.0 使用指南	39.00	05591				
10	Windows NT Workstation 4.0 Internet 和联网手册	56.00	05817				

目 录

前言

第 1 章 引言 1

1.1 系统总体结构 1

1.1.1 Windows NT 的设计目标 1

1.1.2 Windows NT 中的硬件特权级 1

1.1.3 操作系统的基础成分 2

1.1.4 Executive 的内容 3

1.1.5 基操作系统的扩展 5

1.1.6 Win32 子系统 6

1.2 内核模式 I/O 组件 7

1.2.1 I/O 子系统的设计目标 7

1.2.2 Windows NT 中的分层 驱动程序 7

1.2.3 SCSI 驱动程序 8

1.2.4 网络驱动程序 9

1.3 专用驱动程序体系结构 11

1.3.1 视频驱动程序 11

1.3.2 打印机驱动程序 12

1.3.3 多媒体驱动程序 14

1.3.4 遗留 16 位应用的驱动程序 15

1.4 小结 16

第 2 章 硬件环境 17

2.1 硬件基础知识 17

2.1.1 设备寄存器 17

2.1.2 访问设备寄存器 18

2.1.3 设备中断 19

2.1.4 数据传输机制 20

2.1.5 直接内存存取 (DMA) 机制 21

2.1.6 设备专用内存 21

2.1.7 自动配置的要求 22

2.2 总线与 Windows NT 23

2.2.1 ISA——工业标准体系结构 23

2.2.2 MCA——微通道体系结构 25

2.2.3 EISA——扩展的工业 标准总线 27

2.2.4 PCI——外围部件互连 30

2.3 处理硬件的提示 32

2.3.1 了解硬件 32

2.3.2 使用硬件智能 33

2.3.3 测试硬件 33

2.4 小结 33

第 3 章 内核模式 I/O 处理 34

3.1 内核模式代码的执行 34

3.1.1 异常 34

3.1.2 中断 34

3.1.3 内核模式线程 34

3.2 Windows NT 的中断使用 34

3.2.1 CPU 优先级 34

3.2.2 中断处理次序 35

3.2.3 软件产生的中断 35

3.3 推迟过程调用 (DPC) 36

3.3.1 DPC 的操作 36

3.3.2 DPC 的行为 36

3.4 访问用户缓冲区 37

3.5 内核模式驱动程序的结构 38

3.5.1 驱动程序初始化和清理例程 38

3.5.2 I/O 系统服务 Dispatch 例程 39

3.5.3 数据传输例程 39

3.5.4 资源同步回调 40

3.5.5 其他驱动程序例程 40

3.6 I/O 处理次序 41

3.6.1 Windows NT 的请求预处理 41

3.6.2 驱动程序的请求预处理 41

3.6.3 数据传输 41

3.6.4 驱动程序的后处理 42

3.6.5 I/O 管理器的后处理 42

3.7 小结 43

第 4 章 驱动程序和内核模式对象 44

4.1 数据对象与 Windows NT 44

4.1.1 Windows NT 和 OOP 44

4.1.2 NT 对象和 Win32 对象 44

4.2 I/O 请求包 (IRP) 44

4.2.1 IRP 结构 45

4.2.2 操纵 IRP 46

4.3 驱动程序对象	47	5.5.4 使用推迟过程调用的同步	66
4.4 Device 对象与 Device Extension	48	5.6 同步多个 CPU	66
4.4.1 Device 对象的结构	49	5.6.1 自旋锁的工作	66
4.4.2 操纵 Device 对象	49	5.6.2 使用自旋锁	66
4.4.3 Device Extension	50	5.6.3 使用自旋锁的规则	67
4.5 Controller 对象和 Controller Extension	50	5.7 链表	68
4.5.1 Controller 对象的结构	51	5.7.1 单向链表	68
4.5.2 操纵 Controller 对象	51	5.7.2 双向链表	69
4.5.3 Controller Extension	52	5.7.3 删除链表块	69
4.6 Adapter 对象	52	5.8 小结	70
4.6.1 Adapter 对象的结构	52	第 6 章 初始化和清理例程	71
4.6.2 操纵 Adapter 对象	52	6.1 写 DriverEntry 例程	71
4.7 Interrupt 对象	53	6.1.1 执行上下文环境	71
4.7.1 Interrupt 对象的结构	53	6.1.2 DriverEntry 例程的工作	71
4.7.2 操纵 Interrupt 对象	53	6.1.3 初始化 DriverEntry 点	72
4.8 小结	54	6.1.4 创建 Device 对象	72
第 5 章 一般的开发问题	55	6.1.5 选择缓冲策略	73
5.1 驱动程序设计策略	55	6.1.6 NT 和 Win32 设备名	73
5.1.1 使用正规设计方法	55	6.2 范例代码: 驱动程序初始化	74
5.1.2 使用渐进开发方法	55	6.3 写 Reinitialize 例程	81
5.1.3 使用样板驱动程序	56	6.3.1 执行上下文	81
5.2 编码约定和技术	56	6.3.2 Reinitialize 例程的作用	82
5.2.1 一般建议	56	6.4 写 Unload 例程	82
5.2.2 命名约定	56	6.4.1 执行上下文	82
5.2.3 头文件	57	6.4.2 Unload 例程的作用	82
5.2.4 状态返回值	58	6.5 范例代码: 驱动程序清理	83
5.2.5 Windows NT 驱动程序 支持例程	58	6.6 写 Shutdown 例程	86
5.2.6 丢弃初始化例程	59	6.6.1 执行环境	86
5.2.7 控制驱动程序分页	60	6.6.2 Shutdown 例程的作用	86
5.3 驱动程序内存分配	60	6.6.3 启用关闭通知	86
5.3.1 对驱动程序可用的内存	60	6.7 驱动程序的测试	87
5.3.2 使用内核堆栈	61	6.7.1 测试过程	87
5.3.3 使用池区域	61	6.7.2 WINOBJ 实用程序	87
5.3.4 内存再分配的系统支持	62	6.8 小结	88
5.4 Unicode 字符串	64	第 7 章 硬件初始化	89
5.4.1 Unicode 字符串数据类型	64	7.1 查找自动检测的硬件	89
5.4.2 处理 Unicode	64	7.1.1 自动检测如何工作	89
5.5 中断同步	65	7.1.2 自动检测硬件和 Registry	89
5.5.1 问题	65	7.1.3 查询硬件数据库	90
5.5.2 中断阻塞	65	7.1.4 ConfigCallback 例程的作用	92
5.5.3 阻塞中断的规则	66	7.1.5 使用配置数据	93
		7.1.6 转换配置数据	94
		7.2 范例代码: 查找自动检测硬件	94

7.2.1 XXDRIVER.H	95	9.2.3 连接到中断源	140
7.2.2 AUTOCON.C	96	9.2.4 从中断源断开	141
7.3 查找未识别的硬件	103	9.3 写 Start I/O 例程	142
7.3.1 把驱动程序参数加到 Registry	103	9.3.1 执行上下文	142
7.3.2 从 Registry 中检索参数	104	9.3.2 Start I/O 例程的工作	142
7.3.3 设备信息的其他来源	105	9.4 写中断服务例程 (ISR)	142
7.4 范例代码: 查询 Registry	106	9.4.1 执行上下文	142
7.5 分配和释放硬件	115	9.4.2 中断服务例程的工作	143
7.5.1 如何进行资源分配	115	9.5 写 DpcForIsr 例程	143
7.5.2 如何声明拥有硬件资源	116	9.5.1 执行上下文	143
7.5.3 如何释放硬件	118	9.5.2 DpcForIsr 例程的工作	144
7.5.4 映射设备内存	119	9.5.3 优先级增量	144
7.5.5 装入设备微代码	119	9.6 硬件: 并行端口	145
7.6 范例代码: 分配硬件	120	9.6.1 并行端口的工作	145
7.7 小结	124	9.6.2 设备寄存器	145
第 8 章 驱动程序 Dispatch 例程	125	9.6.3 中断行为	146
8.1 启用驱动程序 Dispatch 例程	125	9.6.4 并行端口的驱动程序	146
8.1.1 I/O 请求派发机制	125	9.7 范例代码: 并行端口驱动程序	147
8.1.2 启用特定的功能代码	126	9.7.1 XXDRIVER.H	147
8.1.3 决定支持哪些功能代码	126	9.7.2 INIT.C	147
8.2 扩展派发接口	126	9.7.3 TRANSFER.C	149
8.2.1 定义私有的 IOCTL 值	127	9.8 测试数据传输例程	155
8.2.2 IOCTL 参数传递方法	127	9.9 小结	155
8.2.3 编写 IOCTL 头文件	128	第 10 章 定时器	156
8.3 写驱动程序 Dispatch 例程	130	10.1 处理设备超时	156
8.3.1 执行环境	130	10.1.1 I/O 定时器例程的工作	156
8.3.2 Dispatch 例程的作用	130	10.1.2 如何截获设备超时条件	156
8.3.3 退出 Dispatch 例程	131	10.2 范例代码: 截获设备超时	157
8.4 处理特定种类的请求	132	10.2.1 XXDRIVER.H	157
8.4.1 处理读写请求	133	10.2.2 INIT.C	158
8.4.2 处理 IOCTL 请求	133	10.2.3 TRANSFER.C	158
8.4.3 管理 IOCTL 缓冲区	135	10.2.4 TIMER.C	161
8.5 测试驱动程序 Dispatch 例程	136	10.3 管理没有中断的设备	162
8.5.1 测试过程	136	10.3.1 处理无中断设备	163
8.5.2 测试程序范例	137	10.3.2 CustomTimerDpc 例程 的工作	163
8.6 小结	137	10.3.3 如何建立 CustomTimerDpc 例程	164
第 9 章 编程 I/O 数据传输	138	10.3.4 如何指定到期时间	164
9.1 编程 I/O 的工作	138	10.3.5 CustomTimerDpc 例程的其他 使用	165
9.1.1 编程 I/O 过程中发生什么	138	10.4 范例代码: 基于定时器的 驱动程序	165
9.1.2 同步各种驱动程序例程	139		
9.2 驱动程序初始化和清理	139		
9.2.1 初始化 Start I/O 入口点	139		
9.2.2 初始化 DpcForIsr 例程	140		

10.4.1	XXDRIVER.H	165	问题	203
10.4.2	INIT.C	166	12.1.3	用内存描述符表管理 I/O
10.4.3	TRANSFER.C	167	缓冲区	204
10.5	小结	171	12.1.4	维护 Cache 一致性
第 11 章	全双工驱动程序	172	12.1.5	DMA 驱动程序分类
11.1	同时做两件事情	172	12.1.6	NT DMA 体系结构的限制
11.1.1	处理并发的 IRP	172	12.2	使用 Adapter 对象
11.1.2	修改的驱动程序体系结构的 工作	173	12.2.1	找到正确的 Adapter 对象
11.1.3	全双工驱动程序的数据结构	173	12.2.2	获取和释放 Adapter 对象
11.1.4	实现辅路径	173	12.2.3	设置 DMA 硬件
11.2	使用 Device Queue 对象	174	12.2.4	清空 Adapter 对象 Cache
11.2.1	Device Queue 对象的工作	174	12.3	写基于包的从属 DMA 驱动程序
11.2.2	如何使用 Device Queue 对象	175	12.3.1	基于包的从属 DMA 的工作
11.3	写 CustomDpc 例程	176	12.3.2	分割 DMA 传输
11.3.1	如何使用 CustomDpc 例程	176	12.4	范例代码: 基于包的从属 DMA 驱 动程序
11.3.2	执行上下文	177	12.4.1	XXDRIVER.H
11.4	取消 I/O 请求	177	12.4.2	REGCON.C
11.4.1	取消 IRP 的过程	178	12.4.3	TRANSFER.C
11.4.2	同步问题	178	12.5	写基于包总线的主 DMA 驱动程序
11.4.3	Cancel 例程的工作	179	12.5.1	设置总线主硬件
11.4.4	Dispatch Cleanup 例程 的工作	180	12.5.2	带分散/集中支持的硬件
11.5	硬件: 16550 UART	182	12.5.3	用 IoMapTransfer 建立分散/集中 列表
11.5.1	16550 UART 的工作	182	12.6	写公共缓冲区从属 DMA 驱动程序
11.5.2	设备寄存器	182	12.6.1	分配公共缓冲区
11.5.3	中断行为	183	12.6.2	使用公共缓冲区从属 DMA 保持 吞吐率
11.6	范例代码: 全双工 UART 驱动程序	185	12.7	写公共缓冲区总线主设备 DMA 驱动 程序
11.6.1	目标	185	12.8	小结
11.6.2	XXDRIVER.H 中的 DEVICE _EXTENSION	185	第 13 章	设备错误的日志记录
11.6.3	DISPATCH.C	186	13.1	Windows NT 中的事件日志
11.6.4	DEVQUEUE.C	188	13.1.1	决定日志记录什么
11.6.5	INPUT.C	191	13.1.2	事件日志如何工作
11.6.6	ISR.C	193	13.2	处理消息
11.6.7	CANCEL.C	197	13.2.1	消息编码如何工作
11.7	小结	201	13.2.2	写消息定义文件
第 12 章	DMA 驱动程序	202	13.2.3	一个小例子: XXMSG.MC
12.1	Windows NT 下 DMA 如何工作	202	13.2.4	编译消息定义文件
12.1.1	使用 Adapter 对象隐藏 DMA 硬件的 变化	202	13.2.5	给驱动程序增加消息资源
12.1.2	解决映射寄存器的分散/集中			

13.2.6 登记驱动程序为事件源	242	15.2 编写分层驱动程序	278
13.3 生成日志项	242	15.2.1 分层驱动程序如何工作	278
13.3.1 准备驱动程序进行错误日志记录	242	15.2.2 分层驱动程序的初始化和清理	279
13.3.2 分配错误日志包	242	15.2.3 代码段: 连接到另一个驱动程序	280
13.3.3 日志记录错误	244	15.2.4 分层驱动程序的其他初始化考虑	281
13.4 范例: 一个错误日志记录例程	244	15.2.5 分层驱动程序中的 I/O 请求处理	282
13.5 小结	250	15.2.6 代码段: 调用低层驱动程序	283
第 14 章 系统线程	251	15.3 写 I/O Completion 例程	284
14.1 系统线程	251	15.3.1 请求 I/O Completion 回调	284
14.1.1 何时使用线程	251	15.3.2 执行上下文	285
14.1.2 创建和终止系统线程	251	15.3.3 I/O Completion 例程的作用	285
14.1.3 管理线程优先级	252	15.3.4 代码段: 一个 I/O Completion 例程	286
14.1.4 系统工作者线程	252	15.4 分配另外的 IRP	287
14.2 线程同步	253	15.4.1 IRP 的 I/O 堆栈再讨论	288
14.2.1 时间同步	253	15.4.2 控制 IRP 堆栈的大小	288
14.2.2 一般同步	253	15.4.3 用 IoBuildSynchronousFsdRequest 创建 IRP	289
14.3 使用派发器对象	254	15.4.4 用 IoBuildAsynchronousFsdRequest 创建 IRP	291
14.3.1 Event 对象	255	15.4.5 用 IoBuildDeviceIoControlRequest 创建 IRP	292
14.3.2 在驱动程序之间共享 Event 对象	256	15.4.6 从头建立 IRP	293
14.3.3 Mutex 对象	256	15.4.7 为低层驱动程序建立缓冲区	296
14.3.4 Semaphore 对象	257	15.4.8 记录驱动程序分配的 IRP	296
14.3.5 Timer 对象	258	15.5 写过滤器驱动程序	297
14.3.6 Thread 对象	259	15.5.1 过滤器驱动程序如何工作	298
14.3.7 Mutex 的变种	260	15.5.2 过滤器驱动程序中的初始化和清理	298
14.3.8 同步死锁	261	15.5.3 实际发生的操作	300
14.4 范例代码: 一个基于线程的驱动程序	262	15.5.4 使挂接透明	300
14.4.1 驱动程序的工作过程	262	15.6 范例代码: 过滤器驱动程序	301
14.4.2 XXDRIVER.H 中的 DEVICE_EXTENSION 结构	262	15.6.1 YYDRIVER.H- 驱动程序数据结构	301
14.4.3 INIT.C 中的 XxCreateDevice 函数	263	15.6.2 INIT.C- 初始化代码	301
14.4.4 DISPATCH.C 中的 XxDispatchRead-Write 函数	265	15.6.3 DISPATCH.C- 过滤器 Dispatch 例程	306
14.4.5 THREAD.C	266	15.6.4 COMPLETE.C- I/O Completion 例程	309
14.4.6 TRANSFER.C	268		
14.5 小结	276		
第 15 章 高层驱动程序	277		
15.1 中间驱动程序概述	277		
15.1.1 什么是中间驱动程序	277		
15.1.2 应使用分层结构吗	277		

15.7 写紧耦合驱动程序	312	17.2.3 减少调试的编码策略	335
15.7.1 紧耦合驱动程序如何工作	313	17.2.4 跟踪驱动程序错误	336
15.7.2 紧耦合驱动程序中的初始化和清理	313	17.3 阅读崩溃时显示的屏幕	336
15.7.3 紧耦合驱动程序中的 I/O 请求处理	314	17.3.1 系统崩溃时发生什么	336
15.8 小结	315	17.3.2 STOP 消息的结构	337
第 16 章 构造和安装驱动程序	316	17.3.3 STOP 消息解释	339
16.1 构造驱动程序	316	17.4 WINDBG 概述	339
16.1.1 BUILD 的用途	316	17.4.1 源代码调试的关键	340
16.1.2 如何构造驱动程序	317	17.4.2 几个 WINDBG 命令	340
16.1.3 写 SOURCES 文件	318	17.5 分析崩溃转储	342
16.1.4 BUILD 生成的日志文件	319	17.5.1 分析的目标	342
16.1.5 递归的 BUILD 操作	319	17.5.2 开始分析	342
16.2 其他 BUILD 时的活动	320	17.5.3 跟踪堆栈	342
16.2.1 使用预编译头文件	320	17.5.4 间接检查方法	345
16.2.2 在驱动程序中包含版本信息	320	17.5.5 使用 DUMPEXAM 发现崩溃	347
16.2.3 在 BUILD 中包含非标准组件	322	17.6 交互式调试	348
16.2.4 把驱动程序符号数据移入 .DBG 文件	323	17.6.1 开始和结束调试会话	348
16.3 安装驱动程序	324	17.6.2 设置断点	348
16.3.1 如何手工安装驱动程序	324	17.6.3 设置硬断点	349
16.3.2 驱动程序 Registry 项	324	17.6.4 使用打印语句	349
16.3.3 标准驱动程序的最后用户安装	326	17.7 写 WINDBG 扩展	349
16.3.4 非标准驱动程序的最后用户安装	327	17.7.1 WINDBG 扩展如何工作	349
16.4 控制驱动程序装入次序	327	17.7.2 初始化和版本检查函数	350
16.4.1 改变驱动程序的 Start 值	327	17.7.3 写扩展命令	351
16.4.2 建立驱动程序之间明确的依赖关系	328	17.7.4 WINDBG 帮助函数	351
16.4.3 建立全局组依赖关系	328	17.7.5 构造和使用扩展 DLL	352
16.4.4 控制组内的装入次序	330	17.8 范例代码：一个 WINDBG 扩展	352
16.5 小结	331	17.8.1 XXDBG.C	352
第 17 章 测试和调试驱动程序	332	17.8.2 XXDBG.DEF	357
17.1 驱动程序测试指南	332	17.8.3 SOURCES 文件	357
17.1.1 测试驱动程序的一般方法	332	17.8.4 样本输出	357
17.1.2 使用 Microsoft 硬件兼容性测试 (HCT)	333	17.9 其他调试技术	358
17.2 关于驱动程序错误的一些思考	334	17.9.1 在驱动程序中留下调试代码	358
17.2.1 驱动程序错误分类	334	17.9.2 截获不正确的假设	358
17.2.2 重复产生驱动程序错误	335	17.9.3 使用错误检查回调函数	359
		17.9.4 截获内存泄漏	359
		17.9.5 使用计数器，二进制位和缓冲区	360
		17.10 小结	362
		第 18 章 驱动程序性能	363
		18.1 一般指南	363
		18.1.1 知道要达到的目标	363
		18.1.2 了解硬件	363

18.1.3 探讨创造性的驱动程序设计	363	18.6 范例代码: 数据收集 DLL	377
18.1.4 创造性地优化代码	364	18.6.1 XXPERF.C	377
18.1.5 计量所做的每件事情	364	18.6.2 构造和安装这个例子	385
18.2 Windows NT 中的性能监视	364	18.7 小结	385
18.2.1 一些术语	364	附录 A 开发环境	386
18.2.2 性能监视系统如何工作	365	A.1 硬件和软件要求	386
18.2.3 驱动程序如何输出性能数据	366	A.2 调试符号文件	387
18.3 把计数器名字加到 Registry	366	A.3 在目标系统上启用崩溃转储	387
18.3.1 Registry 中的计数器定义	366	A.4 启用目标系统的调试客户程序	388
18.3.2 写 LODCTR 命令文件	367	附录 B 常见错误检查代码	390
18.3.3 使用 LODCTR 和 UNLODCTR	368	B.1 驱动程序的常见问题	390
18.4 性能数据的格式	369	B.2 同步问题	391
18.4.1 性能数据的总体结构	369	B.3 破坏的驱动程序数据结构	391
18.4.2 计数器的类型	371	B.4 内存问题	392
18.4.3 带多个实例的对象	373	B.5 硬件故障	393
18.5 写数据收集 DLL	374	B.6 配置管理器和 Registry 问题	394
18.5.1 数据收集 DLL 的内容	374	B.7 文件系统问题	395
18.5.2 数据收集 DLL 中的错误处理	375	B.8 系统初始化故障	395
18.5.3 安装 DLL	376	B.9 内部系统故障	397
		参考文献	398

第 1 章 引 言

一般来说,关于设备驱动程序的书籍一开始都要回答这个问题:“什么是驱动程序?”。然而,在 Windows NT 中问这个问题就有点像问“格子花呢是什么颜色?”,因为至少有十几个不同的软件组件都可以称为驱动程序。本章介绍 NT 支持的不同种类的驱动程序,同时,说明使这个操作系统如此令人感兴趣的一些设计策略。

1.1 系统总体结构

Windows NT 驱动程序当然不会独立存在,而是大型复杂操作系统的一个部分。本节简要介绍 Windows NT 的体系结构,并指出对编写设备驱动程序的人来说最感兴趣的特性。

1.1.1 Windows NT 的设计目标

与所有其他商业操作系统一样,Windows NT 是理想化的目标与推动市场运转的现实相互作用的结果。Windows NT 的设计目标是:

- 兼容性。该操作系统应支持大量已存在的软件和遗留的硬件。
- 坚固性和可靠性。该操作系统应能够经受用户的无意或恶意的攻击,而且,各个应用程序应尽可能地独立。
- 可移植性。该操作系统应能够在现在及将来的多种硬件平台上运行。
- 可扩展性。应能够增加新的功能和支持新的 I/O 设备,而不用涉及现有的代码基。
- 性能。该操作系统应能够在常用的硬件上提供合理的性能,还应利用像多处理硬件这样的功能部件。

平衡所有这些目标与投放市场的合理时间是一个复杂的过程。下面描述系统设计者提出的解决方案,首先讨论保持操作系统安全的保护机制。

1.1.2 Windows NT 中的硬件特权级

在多任务环境中,有许多的事情不允许应用程序去做。欺骗内存管理硬件或停止处理器就是引起严重问题的动作的两个例子。Windows NT 不依赖于应用程序自己的行为,而是利用硬件强制的特权检查机制来保证系统完整性。

为了避免硬件的依赖性,Windows NT 使用一个简化的模型描述硬件特权级,然后这个模型映射到指定 CPU 上可用的特权检查机制。如果 CPU 要支持 Windows NT 硬件特权级模型,它必须能够以两种模式运行,即内核模式和用户模式。

1. 内核模式

当 CPU 运行于内核模式时,一切都可运行。任务可以执行特权级指令,对任何 I/O 设备有全部的访问权,还能够访问任何虚地址和控制虚拟内存硬件。这种模式对应于 Intel 80x86 上 0 级环。

2. 用户模式

在这个模式中,硬件防止特权指令的执行,并进行内存和 I/O 空间的引用的检查。这

就允许操作系统限制任务对各种 I/O 操作的访问，并捕捉违反系统完整性的任何行为。在用户模式中，运行的代码如果不通过操作系统中的某种门机制，就不能进入内核模式。在 Intel 80x86 处理器上，这个模式对应于 3 级环。

1.1.3 操作系统的基础成分

Windows NT 的基础成分实现一个构造更复杂环境的通用操作系统平台。从图 1-1 可以看到，这些基础成分由 3 个主要的内核模式代码块组成。

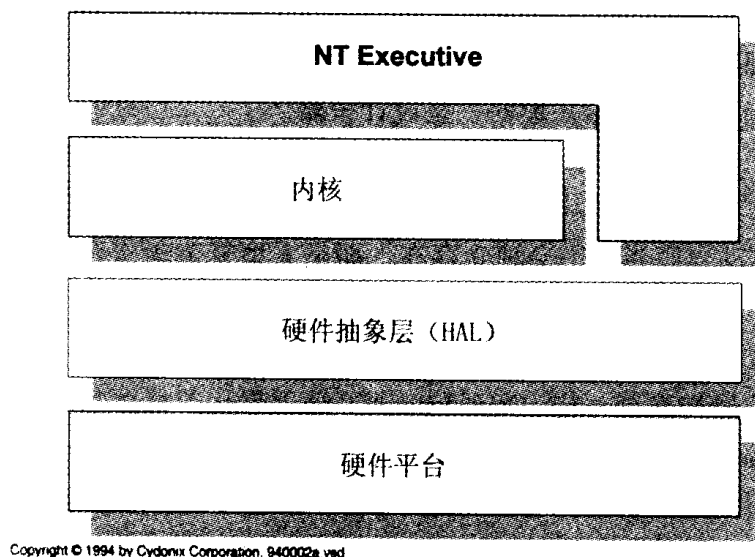


图 1-1 NT 内核模式组件的总体结构

1. 硬件抽象层 (HAL)

HAL 是一薄层软件，用不是 CPU 自身一部分的任何硬件的抽象模型代表系统的其余部分。HAL 显露一组很好定义的函数，其管理的项目有：

- 片外高速缓存。
- 定时器。
- I/O 总线。
- 设备寄存器。
- 中断控制器。
- DMA 控制器。

各种系统组件使用 HAL 函数与 CPU 外的硬件打交道，这样基本上把平台特定的细节对系统的其余部分隐藏起来，消除了对不同系统厂商有不同版本操作系统的要求。特别是使用 HAL 例程，使内核和设备驱动程序在有相同 CPU 体系结构的平台上二进制代码兼容。

2. 内核

HAL 是平台的抽象化，而内核代表 CPU 自身的理想化的视图。内核提供管理以下功能的机制：

- 中断和异常处理。
- 线程调度和同步。
- 多处理机同步。
- 定时控制。