HZ BOOKS

软件开发技术丛书

Windows WDM 设备驱动程序开发指南

Writing Windows WDM Device Drivers

（美）Chris Cant 著　　孙义 马莉波 国雪飞 等译

机械工业出版社
China Machine Press

R&D BOOKS

软件开发技术丛书

Windows WDM设备驱动程序开发指南

(美) Chris Cant 著

孙 义 马莉波 国雪飞 等译

机械工业出版社
China Machine Press

本书主要介绍Windows 98和Windows 2000新的驱动程序模型(WDM)。描述了WDM设备驱动程序的结构、功能和开发方法；通过实际的设备驱动程序例子，说明WDM设备驱动程序的实现技术、测试和调试方法；介绍了本书提供的一个调试软件DebugPrint；说明如何在客户驱动程序中使用USB驱动程序接口(USBDI)访问USB设备，以及如何利用HID类驱动程序以标准方法访问多种输入设备。

本书版权登记号：图字：01-1999-2865

图书在版编目(CIP)数据

Windows WDM设备驱动程序开发指南 / (美) 坎特(Cant, C.)著；孙义等译. －北京：机械工业出版社，2000. 1

(软件开发技术丛书)

书名原文：Writing Windows WDM Device Drivers

ISBN 7-111-07709-1

Ⅰ. W... Ⅱ. ①坎... ②孙... Ⅲ. 窗口软件，Windows-程序设计 Ⅳ. TP316.7

中国版本图书馆CIP数据核字(1999)第55433号

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：陈剑匡

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

2000 年 1 月第 1 版 · 2000 年 3 月第 2 次印刷

787mm × 1092mm 1/16 · 26印张

印数：6 001- 10 000 册

定价：56.00元(附光盘)

前　言

首先，建议读者查看本书的网站www.phdcc.com/wdmbook/，这些网页列出勘误表等信息，并包括一个意见反馈页。本书主要针对Windows 98和Windows 2000 Beta 2版，并根据Beta 3版进行了一些更新。我将设法保证在本书的网站上发布所有必要的更新。

开发设备驱动程序时，尽管本书和Microsoft文档能够提供很好的起点，但只有在你真正写一个自己的驱动程序时，才会真正学会驱动程序的开发。往往在你出错时，才会真正发现如何正确地写程序。在本书中，我试图给出自己曾经出错的情况和发现并解决这些问题的技术。我发现对例子代码进行详细说明有助于我尽快掌握驱动程序的开发技术，所以建议读者向其他人一行一行地讲解自己的驱动程序的代码。

本书的读者

如果想学习如何写Windows 98和Windows 2000的设备驱动程序，本书将提供最好的参考，它全面介绍了核心的Windows驱动程序模型(WDM)，并详细讨论了两种类型的系统类驱动程序——通用串行总线(USB)和人工输入设备(HID)。

本书还讨论了Windows NT 4和NT 3.51式的驱动程序，这些NT式驱动程序也可在Windows 2000中运行，有些还可在Windows 98中运行。

建议读者运行所有的例子驱动程序，并研究它们的源代码，从这些例子中得到最大的收获。本书配套软件包括一个叫做DebugPrint的软件，它使我们可以查看例子驱动程序产生的跟踪调试语句。

如果使用专有的设备驱动程序工具包，本书仍然是很有用的。首先，关于核心WDM的几章介绍整个主题的背景知识，并帮助解释你的工具包如何工作。一个工具包可能没有涉及WDM的一些功能，所以需要“手工”实现。更重要的是，关于类驱动程序的那几章提供为各种设备开发驱动程序所需要的信息。

固件工程师和硬件工程师也会发现，阅读本书其中的一些章节是有用的。设计外部设备时，软件工程师需要开发一个设备驱动程序来访问这个硬件设备。阅读本书的第1章可以了解基本的术语。向软件工程师提供合适层次的硬件说明，可以帮助他们进行设备驱动程序的开发。

如果读者目前是一个VxD驱动程序的开发者，那么对本书中的许多概念应该是熟悉的，但会发现实现技术是非常不同的。

如果读者写过NT 4和NT 3.51的驱动程序，那么对一些核心WDM的章节应该是熟悉的，但即插即用(PnP)和电源管理是非常重要的新功能。支持Windows Management Instrumentation (WMI)也是必要的。系统类驱动程序对于这一类读者也是新概念。

NT驱动程序在Windows 2000下不经修改就可运行，且有时候可以在Windows 98中运行。但是，如果合适的话，应该严肃考虑把这些驱动程序转换成带即插即用功能的驱动程序。支持电源管理将帮助减少开机和关闭系统的延时，使PC更容易使用。直接存储器访问(DMA)系

统由WDM设备驱动程序以稍微不同的方法使用。

约定

全局唯一描述符(GUID)是一个很长的字符串，用于标识组件对象模型(COM)对象和设备接口。完全的GUID在书中可能采用缩写形式，如{C0CF0640-5F6E-11d2-B677-00C0DFE4C1F3}常常缩写为{C0CF0640...}。

在说到“Windows”的地方，是指Windows 98和Windows 2000。如果没有特别的说明，“Windows”不代表NT 3.51和NT 4。有时用W98指代Windows 98，W2000指代Windows 2000。

编码风格

读者应适应本书和例子驱动程序源代码所使用的编码风格，光盘上的实际驱动程序源代码常常含有比书中的程序清单更多的注解。

大多数的例子代码是使用C++写的，但没有怎么使用C++的特性。例如，没有使用任何C++类。

布尔真和假值使用两个不同的类型表示。内核调用必须使用BOOLEAN类型，它实际上是一个unsigned char，0表示FALSE，非0表示TRUE。对于任何其他的布尔值，使用C++固有的bool类型，有false和true二种值。

文中常常提及Win32应用程序。我现在还没有完全掌握Win64(即Windows 2000的64位版本)的含义。但是，可以假定Win32应用程序可以进行的任何调用，Win64应用程序也可以调用。尤其重要的是，我还没有发现在Win64系统中设备驱动程序会发生什么情况。

参加本书翻译工作的有孙义、马莉波、国雪飞、张亮、张永君、扬永亭、赵强、赵彬等。
原书书号：ISBN 0-87930-565-7

目　录

第1章 引 言

本书介绍如何编写某些类型的Windows设备驱动程序，主要描述Windows 98和Windows 2000的Windows驱动程序模型(WDM)。另外，还将介绍也可以在Windows NT 3.51和NT 4中运行的设备驱动程序，称为“NT式”驱动程序。

设备驱动程序提供连接到计算机的硬件的软件接口，它是操作系统的一个信任部分。用户应用程序以一种规范的方式访问硬件，而不必考虑必须如何控制硬件。例如，磁盘驱动程序可以隐藏数据就必须以512字节的块写这样的事实。

驱动程序是一个软件，在装入后成为操作系统内核的一部分。它使一个或多个设备可用于用户态程序员，每个设备代表一个物理的或逻辑的硬件。例如，一个物理硬盘可以看成两个逻辑盘C:和D:。

在Windows中，驱动程序总是使设备看起来像是一个文件，可以打开设备的一个句柄，然后应用程序可以在设备句柄最后关闭之前向驱动程序发出读写请求。

显然，有许多基本上是一样的硬件，因为它们共享一个总线或完成类似的任务。Microsoft提供了几个通用的驱动程序，执行这些常见任务，设备驱动程序可以使用这些标准驱动程序的功能。这个方法使公共总线的共享更容易，且更容易写出新的驱动程序。

所以，写新驱动程序的任务常常从标识可以使用哪些通用驱动程序开始。一个驱动程序栈分阶段处理用户的请求，这些驱动程序一个个相互叠加在一起。低层的总线驱动程序可以用于处理与硬件的所有基本通信，中间的类驱动程序对整个一类驱动程序提供共同的设施。

在Windows 98和Windows 2000中，设备驱动程序必须根据Windows驱动程序模型(WDM)设计，这个模型将在下面介绍。WDM基于在Windows NT 4和NT 3.51中使用的设备驱动程序模型。

1.1 Windows驱动程序模型

Windows驱动程序模型有两个不同的但同样重要的方面。首先，核心模型描述设备驱动程序的标准结构；其次，Microsoft为常见类型的设备提供一系列的总线驱动程序和类驱动程序。

核心WDM模型描述设备驱动程序如何安装和启动，以及如何为用户请求服务和与硬件打交道。WDM设备驱动程序必须适应于即插即用(PnP)系统，允许用户插入可以在软件中配置的设备。

Microsoft提供了一系列的系统驱动程序，它们具有为许多标准类型设备服务所需的所有基本功能。第一种系统驱动程序支持不同类型的总线，如通用串行总线(USB)、IEEE 1394(FireWire)和音频端口设备。其他的类驱动程序实现标准的Windows功能，如人工输入设备(HID)和内核流功能。最后，静态图像体系结构(STI)提供一个处理静态图像、扫描仪等的框架。

这些系统类驱动程序使一些类型的设备驱动程序的设计变得容易得多了。例如，USB系

统驱动程序处理USB总线上的所有低层通信，这样其他驱动程序有了一个定义好的接口可以使用，这使得向USB总线发出请求是相当直接的。

源代码兼容与二进制兼容

一开始，Microsoft宣称WDM驱动程序会是Windows 98和Windows 2000 x86之间二进制兼容的，且与Windows 2000 Alpha平台源代码兼容。但是，现在看来并不能保证二进制兼容，尽管DDK在这方面并不清楚。

笔者为了安全起见，仅安装为正确的操作系统构造的驱动程序，也就是说在为Windows 98构造驱动程序时，使用Windows 98驱动程序开发工具包(DDK)，而对Windows 2000使用Windows 2000 DDK。

如果使用仅在Windows 2000中出现的一些WDM功能，则就不能达到源代码兼容。例如，Windows 2000 USB系统驱动程序支持一些Windows 98驱动程序不可用的功能。

下面通过开发一个简单的设备驱动程序，首先讨论WDM的核心功能。然后，讨论必须使用访问内存和处理中断这样的硬件资源的驱动程序。最后，介绍USB和HID系统驱动程序。使用内核例程IoIsWdmVersionAvailable确定要求的WDM版本是否可用。DDK头文件定义两个常量WDM_MAJORVERSION和WDM_MINORVERSION，对于Windows 98，这些常量是1和0；而对于Windows 2000，它们是1和0x10。

1.1.1 WDM与NT式驱动程序

图1-1粗略表示了WDM和NT式驱动程序之间的差别，本书后面的内容将说明这个图中提到的所有功能。

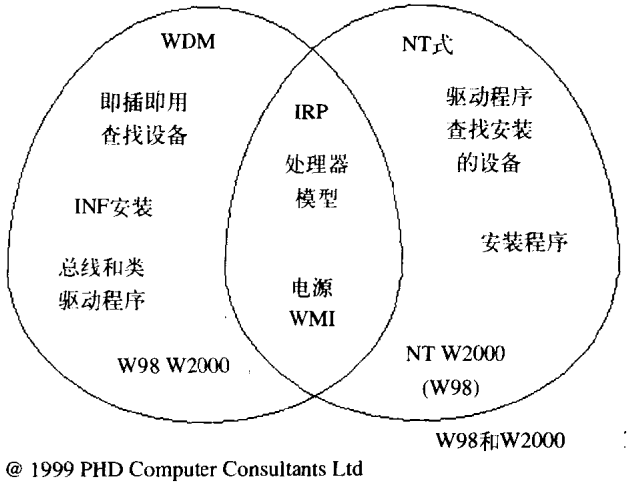

图1-1 WDM和NT式设备驱动程序

这两种驱动程序之间的重叠是相当多的。实际上，写WDM和NT式驱动程序基本上是相同的工作，驱动程序代码中的主要不同是如何创建设备。

在WDM驱动程序中，即插即用(PnP)管理器告知何时向系统添加一个设备，或者从系统删除设备。PnP管理器使用安装INF文件查找新设备的正确驱动程序；相反，NT式驱动程序必须发现它自己的设备，通常在它的初始化例程中。NT式驱动程序通常使用专门的安装程序

安装。

新的总线驱动程序和类驱动程序仅可用于WDM设备驱动程序，新的WDM和NT式驱动程序应支持电源管理和WMI特性。

1.1.2 可用的驱动程序

如果从头开始写一个驱动程序，似乎大多数的代码都与访问设备无关，在执行一些实际的输入输出(I/O)之前，有许多“基础构架”必须设置。本书使用一些实际有用的驱动程序作为例子，会帮助读者尽快入门。一些驱动程序是可以直接使用的，而其他驱动程序形成读者自己的驱动程序的基础。

一个虚拟设备驱动程序用于解释核心的WDM功能。虚拟设备不使用任何实际的硬件。3个驱动程序Wdm1、Wdm2和Wdm3逐渐实现更多的功能。一开始，它们提供一个共享内存缓冲区，使得它们形成其他有用驱动程序的基础。实际上，Wdm2驱动程序就被用作本书其他几个驱动程序的基础，包括使用系统驱动程序的驱动程序。

在本书中，DebugPrint软件用于提供驱动程序的跟踪调试输出。DebugPrint驱动程序将在第4章中介绍。读者可以在自己的设备驱动程序中使用DebugPrint。

WdmIo和PHDIo是通用驱动程序，可以马上用于提供对简单硬件设备的访问，一个控制Win32程序可以使用一组简单但功能很强的命令与硬件对话。这些驱动程序支持中断驱动的I/O。应用程序例子说明这些通用驱动程序如何用于访问并行端口。

UsbKbd和HidKbd驱动程序都访问连接到USB总线的键盘，这些驱动程序说明使用USB和HID类驱动程序要求的技术。最后，Win32应用程序HidKbdUser说明用户态应用程序如何发现和访问HID设备。

1.1.3 本书配套光盘

本书配套的光盘含有前面提到的所有驱动程序，包括这些驱动程序的全部源代码和编译好的可执行文件。每个驱动程序至少有一个测试程序，用于测试驱动程序。另外，本书还有一些有用的Win32实用程序，用于辅助驱动程序的开发。第4章有安装本书配套光盘上软件的完整指令。大多数的测试Win32程序是控制台应用程序，使它们很容易写和理解；没有必要让这些代码成为成熟的窗口式或MFC应用程序。

为了充分利用本书，应该安装全部驱动程序例子并运行Win32测试程序，所有的驱动程序都包括跟踪调试语句，如果安装了DebugPrint软件，则可以在DebugPrint Monitor应用程序中看到跟踪输出。

仔细检查每个例子的源代码是非常有用的，在实际工作中，一个人只有通过写实际的代码才能学会编程。可以使用这些驱动程序例子作为起点并增加一些功能来进一步掌握设备驱动程序的编写。

1.1.4 设备驱动程序软件工具

有两类软件可用于帮助设备驱动程序的开发，如表1-1所示。驱动程序类是源代码C++类包装程序，提供驱动程序需要的大多数缺省功能；通用驱动程序可以从用户态访问许多类型的设备。OSR公司有一个叫做OSRDDK的调试辅助工具。

第4章讨论对开发设备驱动程序有用的其他工具。

表1-1 设备驱动程序软件工具

公 司	网 站	驱动程序类	通用驱动程序
Blue Water Systems	www.bluewatersystems.com	WinDK	WinRT
KRF Tech	www.krftech.com		WinDriver
Compuware Numega	www.vireo.com	DriverWorks	DriverAgent
Open Systems Research	www.osr.com		

在一些情况下，对新的硬件可能不需要新的设备驱动程序。有现成的通用产品处理简单的I/O，不管它是内存映射的空间还是在I/O空间中。实际上，这可能是更好的选择，因为NT和Windows 95使用相同的接口支持，使用户的代码是可移植的。即使编写简单的NT设备驱动程序或Windows 95 VxD也不是容易的任务。这些通用的驱动程序可以是脚本驱动的。如果在用户态线程被调度时必须运行脚本，则中断延时将是一个问题。本书的WdmIo和PHDIo例子是简单的通用设备驱动程序。

读者可能想到了需要写一个新的设备驱动程序处理连接到并行端口的特殊设备。应使用完全的Win32 API接口检查是否能够得到需要的功能，或许要发出异步读或写调用并检查超时。

1.1.5 未讨论的驱动程序类型

本书不讨论如何编写Windows 95或Windows 98的VxD，本书中的驱动程序都不能在Windows 95中运行；本书不描述Windows CE软件，不讨论打印机驱动程序或虚拟DOS驱动程序。

本书描述的设备驱动程序模型是许多类型的驱动程序的基础。本书没有讨论以下驱动程序类型的详细信息：文件系统驱动程序、视频驱动程序、网络驱动程序和SCSI驱动程序。

前面已经提到，本书仅详细描述USB和HID系统驱动程序。

本书中的驱动程序应可以在非x86系统中运行，但是没有测试过。

1.2 新的思维方式

如果读者只写过Win32应用程序，会发现写设备驱动程序需要新的思维方式，没有窗口和消息需要处理，没有Windows的保护手段来防止破坏其他进程和操作系统。源代码级的调试更难设置，大多数的支持库都不可用，甚至C标准库和C++的new运算符也不可用。我们甚至必须使用makefile构造驱动程序，但从Visual Basic这样的开发工具很容易控制构造进程。

因为设备驱动程序是操作系统的信任部分，所以很容易破坏系统。所以，开发者必须写安全可靠的代码，并仔细注释和测试驱动程序的代码。检查从每个内核调用返回的错误值。

设备驱动程序问题一直是用户和支持人员的困难。在发布驱动程序之前。一定要彻底测试它。在各种机器上测试驱动程序。

1.2.1 设备驱动程序环境

设备驱动程序在一个需求的环境中工作，可能有多个用户应用程序向驱动程序发出请求。

一个发出打开I/O请求的用户程序可能突然终止。驱动程序可能在一个多处理器PC中运行，在不同的处理器上运行驱动程序的不同部分。事实上，两个读请求可能在两个不同的处理器上由驱动程序的同一部分同时处理。

低层的设备驱动程序必须处理可能在任何时候到达的硬件中断。在某个时刻，只有驱动程序的一个部分可以访问硬件电路。必须使用直接存储器访问(DMA)把数据从设备传输到内存，或者从内存传输到设备。

支持可配置的和可热拔插的总线也增加了设备驱动程序开发者的负担。例如，用户可以在任何时候删除一个即插即用设备，而且内核可以在任何时候决定它需要停止设备，使得它可以重新分配所有的硬件资源。

但是，正如前面提到的，使用标准的WDM总线和类驱动程序可帮助减少编写某些类型驱动程序的工作量。

1.2.2 术语与资源

设备驱动程序面临大量的术语，好的规范文档可以帮助硬件和软件工程师一起工作来实现一个共同的目标。

设备驱动程序的开发者需要了解设备是如何工作的。尽管不需要了解设备电路实现的细节，但如果有该设备技术的工作知识，肯定是有帮助的。例如，通用串行总线对包的大小有一定的限制，所以需要划分数据传输来满足这些要求。

WDM自身使用许多技术术语来描述它的操作，后面将逐渐介绍编写设备驱动程序所需的所有结构和概念。

每项技术有它自己的专门术语。例如，USB总线有中断传输(interrupt transfer)的概念，它们与硬件中断没有任何关系。但是，USB规范就是这样描述它的操作的，所以在这里还是要坚持使用正确的术语。

可能产生混淆的一个方面是“类”这个词。Windows类驱动程序是用于访问特定类型的驱动程序的标准驱动程序，如USB、HID或IEEE 1394。相反，USB设备被分类成各种设备类，对打印机有一个USB设备类，对音频设备有另一个设备类(感谢上帝，我在源代码中没有使用C++类，否则就更混乱了)。

为了帮助读者开始设备驱动程序的开发，第24章列出了许多信息资源、图书、新闻组和邮件列表。第24章概述了PC 99规范，它是在新的Windows计算机上应该提供的硬件和软件的规范。最后，附录D“词汇表”解释了在开发设备驱动程序时遇到的许多缩写词。

1.3 Win32程序接口

在进一步介绍设备驱动程序的开发之前，值得讨论Win32程序如何调用设备驱动程序。Win32规范为驱动程序开发者提供了各种指示。

1.3.1 基本I/O

对于Win32程序员来说，可以使用表1-2中列出的函数像访问一个文件那样访问设备。除了打开、读、写和关闭函数外，DeviceIoControl为驱动程序提供任何特殊功能的选择。关于这些函数的详细信息，参见Win32文档。