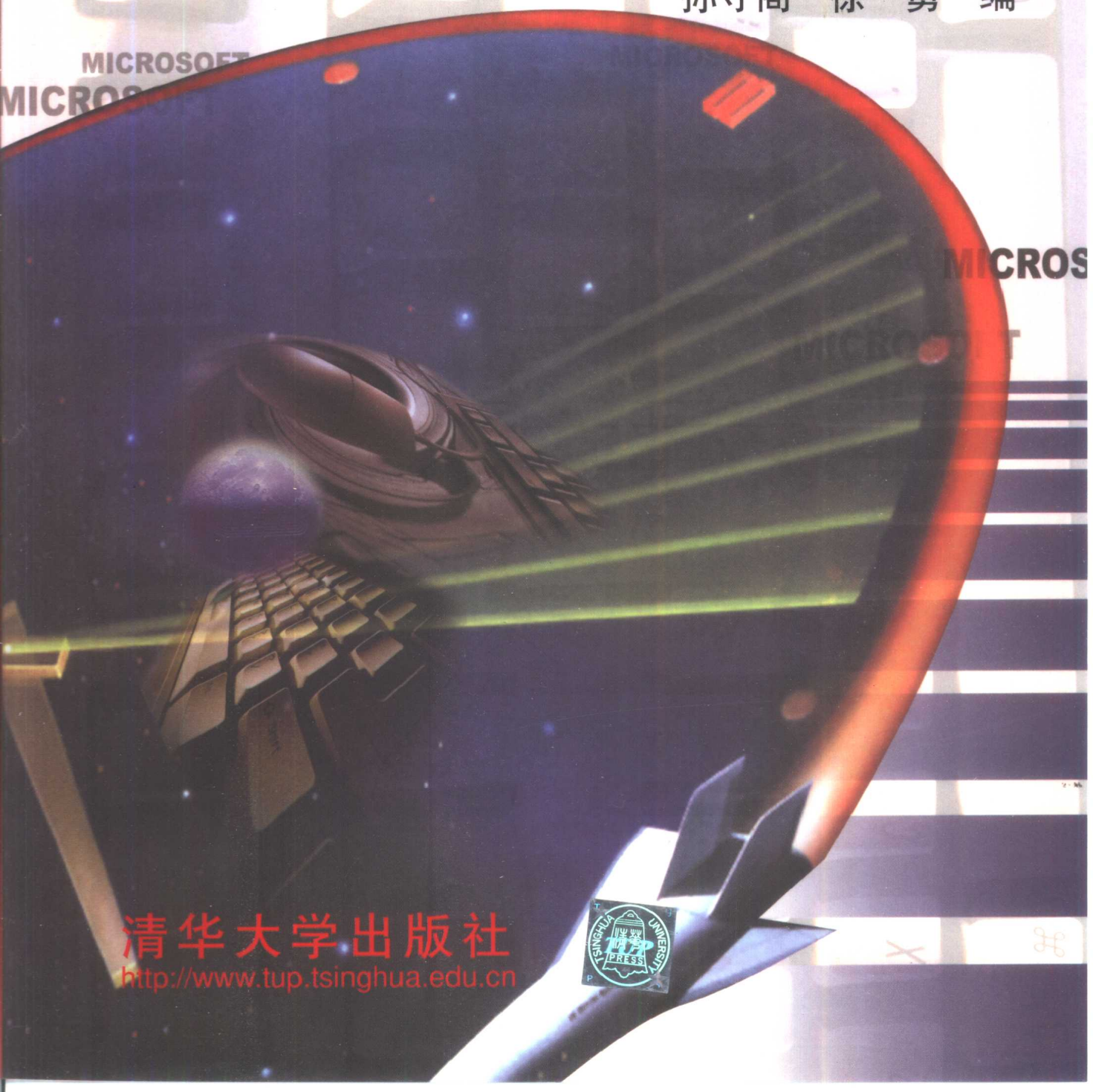
Windows
设备驱动程序技术内幕

孙守阁 徐 勇 编



清华大学出版社
http://www.tup.tsinghua.edu.cn

Windows

设备驱动程序技术内幕

Windows 设备驱动程序技术内幕

孙守阁　徐　勇　编

清 华 大 学 出 版 社

（京）新登字 158 号

内 容 摘 要

Windows 设备驱动程序技术内幕一书，详细讲解了设备驱动程序的原理及实现方法，并例举了大量的程序实例，便于读者学习和掌握。

本书分为三个部分。第一部分主要讲怎样用 DOS 的方法写设备驱动程序，该部分的驱动程序被封装为 DLL 形式。第二部分讲怎样写 Windows 的虚拟设备驱动程序（VxD）。第三部分介绍 Windows 系统下标准模式设备驱动程序的编写方法。

本书是一本技术性较强的工具书，它主要针对有一定计算机基础的程序开发者和硬件设计者，同时也适应需要进一步了解计算机应用的读者。

书　　名：Windows 设备驱动程序技术内幕
作　　者：孙守阁　徐　勇
出 版 者：清华大学出版社(北京清华大学学研楼，邮编 100084)
http:// www.tup.tsinghua.edu.cn
责任编辑：唐　玲
印 刷 者：北京市通州区大中印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印 张**：14　**字 数**：331 千字
版　　次：2000 年 5 月第 1 版　2000 年 6 月第 2 次印刷
书　　号：ISBN 7-302-00900-7/TP・329
印　　数：5001～10000
定　　价：22.00 元

前　言

现在市面上关于 Windows 编程的书很多，但是关于 Windows 设备驱动程序编程的书很少，而关于虚拟设备驱动程序的编程就更少了。在工作中往往需要对标准的硬件设备进行操作，有时还需要使用设备驱动程序对物理设备进行操作，这就需要编程人员掌握这些方面的知识，能够设计出自己需要的设备驱动程序。

本书主要介绍了 3 种设备驱动程序的编写方法。全书分为 3 部分，每一部分先介绍一些基础知识，再进行详细内容的讲述。各部分的内容都是循序渐进的，便于学习和掌握。

第一部分主要讲述怎样用 DOS 的方法编写设备驱动程序。该部分的驱动程序被封装为 DLL 形式，可以被 Windows 应用程序访问，但不能被 DOS 应用程序访问。

第二部分讲述怎样编写 Windows 的虚拟设备驱动程序（VxD）。这些虚拟设备驱动程序作为被托管的 Windows 核心成员运行，可以“到任何地方做任何事”，最终可以知道 Windows 是怎样实现虚拟机的。它们的结构十分直观，便于了解和掌握。当运行环境变得复杂时，VxD 本身和基于 DLL 的驱动程序同样简单。

第三部分介绍 Windows 系统下标准模式设备驱动程序的编写。这一部分讲述了怎样编写一个高质量的驱动程序，Windows 程序怎样与 DOS 的 TSR 进行通信，在 Windows 中怎样用 DMA 进行工作，以及其他一些高级应用。

本书是一本技术性较强的工具书，它主要适用于有一定计算机基础的程序开发者和硬件设计者，同时也适应于需要进一步了解计算机应用的读者。

编　者

1999 年 12 月

目　录

第三部分　特殊技术

第1章 概 述

现在有许多关于 Windows 编程的书，但很少有关于 Windows 设备驱动程序编程的书，而关于虚拟设备驱动程序的编程就更少了。如果在工作中需要对标准的硬件设备进行操作，而且 Windows 使用设备驱动程序来对物理设备进行操作，就需要编程人员掌握这方面的知识，并对这方面的具体应用有所了解。以前的解决办法是设备生产商将他们的驱动程序开发人员送到 Microsoft 学校，进行专门地培训，从而完成所需驱动程序的开发工作。而一般的用户需要使用相应的 Windows 应用程序接口（API）来与驱动程序通信。

但在实际应用中，并非所有的设备都是标准设备。特别是当 Windows 系统的使用越来越广泛的时候，许多领域的专业编程人员发现需要自己来开发驱动程序，以满足对 Windows 应用程序开发的需要。

此外，有些不与非标准设备直接通信的开发者，也需要知道怎样编写一种特殊的设备驱动程序，如利用虚拟设备驱动程序（VxD）来实现自己设计的要求。现在 VxD 已被广泛应用，在 Windows 环境中，VxD 是开发者扩展操作系统的工具——“到任何地方做任何事”。如果你想编写功能强大的调试代码，想改变 Windows 初始化的方法，在 DOS 应用程序和 Windows 应用程序之间通信，编写高性能的驱动程序，你就需要知道怎样创建 VxD，这也是编写本书的目的。

1.1 本书的读者对象

如果你要创建支持非标准设备的驱动程序，你就需要读这本书。标准设备是指计算机的标准配置，如显式器、键盘、鼠标、串口或打印机等的设备；非标准设备是指除标准设备之外的一切设备。这些设备是各个用户根据自己的需要设计的，它与计算机之间没有固定的连接方式，在 Windows 系统中也没有指定这类设备的软件接口，所以它与计算机的连接可以有几种方式选择。本书将逐一介绍这些可供选择的方法，从最简单的动态连接库（DLL）到复杂的双模态中断处理器。本书还将介绍怎样使 Windows 应用程序、DOS 的内存驻留程序（TSR）与 DOS 设备驱动程序进行通信等内容。

本书的读者对象是设备接口的程序开发者。如果你不明白其他程序开发者怎样使他们的程序在 Windows 系统工作起来，你就需要读这本书。如果你想明白一个调试器怎样捕获指针错误；还不明白怎样访问硬件，Windows 是怎样将 DOS 应用程序的结果输出到一个屏幕窗口，怎样编写 Screen Grabbers；想了解有的设备驱动程序在 Windows 启动时检测遗漏的硬件，而有的驱动程序只能在应用程序开始的时候检测遗漏的硬件的原因等，并想解决它，你就需要读这本书。

所有这些问题的答案是虚拟设备驱动程序 VxD（以下简称 VxD），它允许你将新的

功能直接加入到操作系统中。本书将告诉你怎样写 VxD，同时还告诉你如何用 VxD，而不被 Windows 的琐事和 API 访问所干扰，还会告诉你什么对于 VxD 是重要的，什么是你不需要知道的。

1.1.1 哪些人应该读这本书

如果你曾经在 DOS 或其他操作系统下编写过设备驱动程序或者设备接口程序，你适合读这本书。为了从本书中得到更多的东西，你需要对 C 语言和汇编语言非常熟悉，而且还要知道怎样使用 INT 21H 调用 DOS 的应用，熟悉掌握“段”怎样被 DOS 编译器和汇编器使用，你才能很好地读完这本书。

读这本书，不必是一个 Windows 应用程序编程人员。实际上，本书中的程序类似于传统的 DOS 程序。由此可见，预先具备 DOS 编程的经验是有用的，但不是必须的。

1.1.2 哪些人不能读这本书

本书对初学者不适用。要学习这本书，必须熟悉汇编语言和 C 语言，而且对操作系统和计算机结构有一定的了解和经验。例如本书解释了怎样在 Windows 应用程序中正确分配和寻址直接内存访问（DMA），但没有解释什么是 DMA；介绍了怎样在 Windows 下实现一个临界区域，但没有解释什么是临界区域，以及为什么需要它。

本书不是在讲述怎样编写标准的 Windows 设备驱动程序，而是讲述怎样编写非标准的设备驱动程序或者自己所需的特殊设备驱动程序。如果你只想编写一个打印机驱动程序或者通信接口驱动程序，你就应该读其他的书。虽然本书的内容可以直接用于标准的驱动程序，但是专门关于标准设备驱动程序的书会更有用。当然，你也可以先读本书，再读关于标准驱动程序的书。从这个意义上讲，本书是关于所有 Windows 驱动程序的，因为标准的驱动程序只是非标准驱动程序中的一种特殊例子。本书的内容不包括写磁盘或 CD－ROM 这样的数据块驱动程序。

1.2 应 用 平 台

本书的应用在 Windows 平台上。如果没有另外的说明，它适用于 Windows 3X 和 Windows 9X 平台。本书的前两部分通常忽略 Windows 的标准模式应用。由于 PC 286/386 的机器在市场上变得越来越少，开发者主要选择的是提供增强模式的支持。由于硬件趋势以及非标准设备趋于高级化，使得标准模式对于设备驱动程序的编写者变得越来越不重要，而对于非标准模式的设备驱动程序的编写却显得越来越重要。

如果读者学会了用增强模式（非标准模式）来对硬件进行操作，就会更容易理解标准模式的问题。因为标准模式是非标准模式的一种特殊情况。正因为这个原因，本书直到第三部分才讨论标准模式的问题。

虽然 Windows 9X 在结构上与 Windows 3X 有所不同，但是本书中的大部分内容适用于 Windows 9X。虽然许多标准驱动程序的 API 已经改变，而且数据块设备也适用 VxD 设备驱动程序，但是许多非标准的 Windows 3X 驱动程序可以不需要任何改变，就能在 Windows9X 下运行。

1.3 本书的结构

本书分为三个部分，每一部分都介绍一些后面所需的基础知识，再进行详细地讲述。而且各部分的内容都是循序渐进的，便于学习和掌握。

第一部分主要讲的是怎样用 DOS 的方法编写设备驱动程序。该部分的驱动程序被封装为 DLL 形式，它可以被 Windows 应用程序访问，但不能被 DOS 应用程序访问。

第二部分讲的是怎样编写 Windows 的虚拟设备驱动程序。这些虚拟设备驱动程序作为被托管的 Windows 核心成员运行，可以“到任何地方做任何事”。当然，这些驱动程序需要学习者详细了解和掌握，最终可以知道 Windows 是怎样实现虚拟机，而它们的结构又十分直观，便于了解。当运行环境变得复杂时，VxD 本身和基于 DLL 的驱动程序一样很简单。

第三部分介绍了 Windows 系统下标准模式设备驱动程序的编写。这一部分讲述了怎样编写一个高质量的驱动程序，Windows 程序怎样与 DOS 的 TSR 进行通信；在 Windows 中怎样用 DMA 进行工作，以及其他一些高级的应用。

1.4 学习的方法

Windows 可能是一个非常复杂的环境。这本书的主要目的是帮助你理解在这个环境中，对于不同类型的驱动程序，它的关键是什么。每一章介绍一个新的驱动程序或者与之相关的应用程序，而且每一章只介绍能够帮助、理解新例子的有关资料。尽量使每个例子的驱动程序简单，以便于关键的特性一看就明白。每一章都按照循序渐进排列，每个例子都在前一章的基础上建立。所以读者在学习这本书时需要从头到尾进行，只有这样才能很好地理解和掌握本书的内容。有条件的读者在看书的同时，可以结合书上的例子，进行一些实际的操作，这样收获会更大。

本书中的大部分例子程序，都是用 Microsoft C v8.0 编写的，不过在有一些地方使用了汇编语言 Microsoft Assembly（MASM）v6.0 编写。

1.5 需要的工具

对于编写基于 DLL 的驱动程序，需要准备一个可以生成 Windows 应用程序的编译器，

如果编译器可以嵌入汇编语言则更好。对于编写 VxD 程序，则需要一个专门的汇编器和连接器。它是 Windows Device Developer Kit（DDK）的一部分，只能通过定购 Microsoft Developer Network 得到。本书提供的测试驱动程序还使用了一个工具（DBWIN.EXE），它也是 Windows Device Developer Kit（DDK）的一部分，能够对所编写的程序进行方便地测试和运行。

第一部分

驱动程序基础

第2章 基础知识

本章的主要内容是通过建立一个标准的开发环境，为本书中的其他部分作准备。在本章中解释什么是驱动程序以及怎样去建立它，同时介绍一个在本书中使用的测试应用程序，并且介绍其他将要用到的调试程序。本章中介绍的框架驱动程序是第一部分中所有驱动程序的基础。

2.1 驱动程序的定义

从广义上讲，驱动程序是指一系列控制硬件设备的函数。一个驱动程序分类的方法是看这些函数是怎样去封装的。在 DOS 系统中，一个驱动程序可能是一个连接到应用程序.EXE 中的一个模块，或者是软件中与应用程序完全分开的另一部分（一个 TSR）。封装 Windows 驱动程序的最好方法是制作一个 DLL。DLL 是上述两种封装方式的交叉，它虽然连接到应用程序上，但它实际上又不是应用程序的某一部分。

2.2 特权封装

有一些操作系统如 OS/2 和 UNIX，在应用中禁止应用程序直接支持硬件。在这种环境下，只有被称为设备驱动程序的代码才能被允许与硬件相连。控制硬件的应用程序必须使用这些驱动程序所提供的服务程序。严格地讲，应当尽量地为这些构造的、特权的设备控制模块保留设备驱动程序的内容。Windows 系统下的 VxD 就属于这种情况。

2.3 非特权封装

虽然可安装的设备驱动程序已被 DOS 2.0 及其以上的版本所支持，但真正用于 DOS 环境下的驱动程序几乎只用于大容量存储器（如硬盘、软盘和 CD-ROM）以及网络适配器。对于那些非标准接口的硬件，DOS 应用程序一般使用 TSR。这些 TSR 有时被归类为驱动程序，其中的一些惯例也用于 Windows 中。

在 Windows 环境中，驱动程序通常是指与一个 Windows 系统认为是标准设备相连的 DLL。Windows 为这些标准设备类型定义了不同的驱动程序接口。关于这些设备驱动程序更多的信息，可参见 Daniel Norton 的书《编写 Windows 设备驱动程序》（见参考目录）。

Windows 不支持其他类型设备的接口，禁止将支持硬件的 TSR 以及 DOS 设备驱动程序直接放入 Windows 应用程序中。