

虚拟设备驱动程序 VxD 的开发

何海波 成建波 张浩 (西安交通大学电气学院工企教研室 710049)

摘要:本文介绍了虚拟设备驱动程序的基本概念和几种常用的开发工具,并对它们的使用进行了简略评述。

关键词: Windows 95/NT 虚拟设备驱动程序(VxD) DDK VToolsD WinDriver

在实际应用中,为了使应用软件实现某些特殊功能,我们经常需要深入到 Windows 95 内核去编写其系统程序—虚拟设备驱动程序。本文将就当前流行的几种设备驱动程序开发工具,结合作者的使用体会做大致介绍。

一、什么是虚拟设备驱动程序

设备驱动程序是管理系统软硬件资源的二进制可执行代码。它是一个 32 位保护模式下的可执行 DLL,在 16 位的 Windows 3.x 中, VxDs 一般具有后缀名 .386,并且在 SYSTEM.INI 文件的 [386Enh] 节中静态地加载。在 Windows 95 中,由于即插即用的引入, VxD 的功能也被增强了。现在 VxD 可以动态地加载或卸载,以节省系统的资源。从很大程度上 VxD 起到了原来 BIOS 的作用。

VxD 中的“x”代表“某些”, VxD 是虚拟设备驱动程序如虚拟的显示设备驱动程序(VDD)、虚拟的鼠标驱动程序(VMD)等的统称, VxD 对硬件设备进行虚拟化,软件可通过 VxD 的服务来使用相应的设备。

所有的 VxD 的运行都处于 Windows 95 虚拟机管理器(Virtual Machine Management, 即 VMM)的监控下。VMM 包含了所有基本的系统功能,如任务调度、虚拟内存操作、程序装入及终止,任务间通信。此外,还负责处理主要的中断处理及例外情况。正是 VMM 和 VxD 构成了 Windows 95 的系统核心。

二、开发工具的介绍

1. DDK (Device Driver Kit)

DDK 是 Microsoft 公司出品的设备驱动程序开发包,有 Windows 95/NT 两个版本。用它开发 VxD,需要对 Windows 95/NT 的系统结构、VxD 的构成、VMM 的功能有较深入的认识,并且还要有一定的汇编语言的编程经验。它较早出现,使用的范围也较广,故大多数读者对它都有所了解,本文将对它不做过多的介绍,而将主要的篇幅留在后面两种开发工具上。

2. VToolsD 工具箱

(1) 组成。VToolsD 是美国 Vireo Software 公司开发的用于编写设备驱动程序的工具包。它包括一个可视化

VxD 代码生成器 QuickVxD、可加载和卸载 VxD 的工具 VxDLoad,可给出系统以加载 VxD 系统信息的 VxD-View,以及 ANSI C 运行库、VMM/VxD 服务库、VxD 的 C++ 类库。此外, VToolsD 还为刚接触它的新手提供了大量的 C 或 C++ 例程。

在编写过程中,绝大多数的 VMM 和 VxD 的服务都可以通过类库中的成员函数来实现。由于 VToolsD 在这些类中可靠封装了一些任务,使我们在编程中节省了大量的时间,同时也减少了错误的发生。即使是运用它的 C 运行库,也可以使我们的编程效率大大提高。

(2) 应用。首先我们从运行 QuickVxD 程序开始。其主界面如图 1 所示。

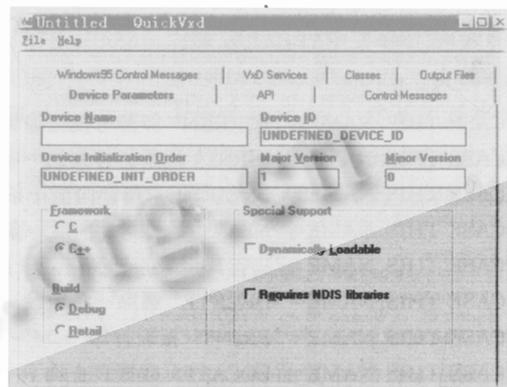


图 1 QuickVxD 主界面

对于开发人员来说, QuickVxD 是一种可以自动生成 VxD 基本框架的表格式工具。它的目的就是让你能够快速开始编写一个新的 VxD, 而无需了解太多 VxD 的低层细节。只要正确定义好设备名、控制消息以及其他一些文件名和类名等后, QuickVxD 便可以生成一个 MAKE 文件、一个头文件和一个包含 VxD 框架的 C 或 C++ 的程序文件。一般设备 ID 和设备初始序号分别使用缺省值。

完成了 VxD 框架构成后,我们需要在其中再加入自

己的代码,以最终实现我们要求的功能。在这里我们要特别注意以下几条 Windows 95 控制消息:SYS-DYNAMIC-DEVICE-INIT 和 SYS-DYNAMIC-DEVICE-EXIT,它们分别是 VxD 程序动态加载和卸载时的控制消息,W32-DEVICEIOCONTROL 则是当应用程序在调用 CreateFile 或 DeviceIoControl 函数时,系统对特定的 VxD 发出的消息,它为 32 位应用程序提供了一个利用 VxD 服务的途径。

此外, VToolD 类库中有三个基本的类 VDevice、VVirtualMachine 和 VThread 非常重要。其中 VDevice 的成员函数用于响应 VMM 或由其他 VxD 传来的消息,所有使用 VToolD 类库的 VxD 程序都必须从 Vdevice 类中派生。VVirtualMachine 和 Vthread 则提供基于虚拟机和线程的控制消息函数。如果我们要想利用其他一些 VMM/VxD 的服务,例如中断、DMA、热键、I/O 端口、内存分配、定时、事件处理等等, VToolD 类库中还实现了五十多个类以实现这些服务。

值得特别一提的是, VToolsD 还为我们提供了一个 .inf 文件编写工具 INF Editor,使用它可以轻松地编写我们的 .inf 文件。

(3) 编译与调试。使用 Visual C++ 的 NMAKE 或 Borlandc 的 MAKE 都可以用来编译。在调试过程中, Numega Technologies 公司的 Soft/ICE for Windows 95 以其方便的中断功能正被广泛地使用。当然, VToolsD 工具箱中的 Microsoft WDEB 也是个不错的调试工具。

3. WinDriver 工具包

(1) 组成。WinDriver 是美国 KRFTech 公司出品的用于编写驱动程序的另一种工具包。它包括一个类似与 QuickVxD 的代码生成器 WinDriver Wizard, 一个 WinDriver 发行包,两个公用程序 pci-scan.exe、pci-dump.exe 和一些例程,其中 pci-scan.exe 可以给出以安装的 PCI 卡以及系统为它们分配资源的列表,pci-dump.exe 则负责得到已安装 PCI 卡的系统配置信息。

与 VToolsD 一样, WinDriver 工具包的优点在于可以使编程人员用他们熟悉的 C 或 C++ 语言来编写设备驱动程序,而不是将大量精力放在编写那些复杂的、难于调试的内核模式代码。这不但会节省不少时间,而且会避免犯许多错误。

(2) 应用。首先我们从运行 WinDriver Wizard 开始。其主界面如图 2 所示。

WinDriver Wizard 可以让你立即接触到硬件而不用做任何事情。这种便利来自与它的自动检测功能,它会自动地读你卡上的内存、IO 地址乃至为你侦听中断。在此之后,通过选择产生代码选项, WinDriver Wizard 会

为你的卡产生基本的程序代码。而我们到现在做的实际上仅仅是选择了一个选项而已,其余的都是 WinDriver Wizard 为我们自动产生的。

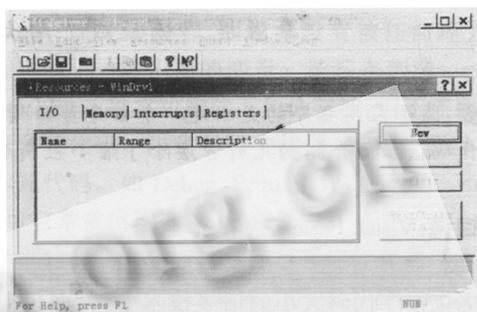


图 2 WinDriver Wizard 主界面

接下来该是我们工作的时候了, WinDriver 提供了许多例程,使用者可以利用它们来产生自己驱动程序的基本框架。与此同时,在 WinDriver 为我们提供的在线帮助里,可以查到许多 WinDriver 提供的功能函数,这些函数使我们能方便地实现中断处理、DMA 传输、I/O 操作、内存映射以及即插即用等功能。

这里要特别注意以下几个函数:WD-Open(), WD-Close, WD-CardRegister() 和 WD-CardUnRegister()。其中前两者是你的程序在开始和结束时所必须调用的两个函数,而后两者则负责对卡登记的建立和删除以及资源分配和释放。这里的资源主要指的是中断处理、I/O 操作和内存分配。

(3) 编译与调试与 VToolsD 基本相同。

三、三种工具比较

以上介绍的三种工具各有优点。DDK 功能强大,编程灵活,适用范围大,可应用于各类硬件驱动程序的编写。但编程难度较大,对编程人员的要求也较高。VToolsD 主要工作环境是在 Windows 95 下,它既具有较强的开发能力,又具有较高的开发效率,是个非常优秀的工具。WinDriver 的适用面要较前面两种工具窄,它主要针对 ISA/PCI 插卡,而对其他类硬件的技术支持较少,但它编写的程序可同时工作在 Windows95/NT 两种操作系统中。读者在选用工具时,应根据自己的需要正确进行选择。

(来稿时间:1998年8月)