第 7 卷　第 4 期　　光学　精密工程　　Vol. 7, No. 4
1999 年 8 月　　OPTICS AND PRECISION ENGINEERING　　August, 1999

文章编号　1004-924X(1999)04-0119-05

虚拟设备驱动程序及其在视频采集卡中的应用

田志刚　　陈桂林
(中科院上海技术物理研究所　　上海 200083)

摘　要　Windows95 作为运行于 32 位保护模式下的操作系统, 为保证系统的安全, 通过屏蔽底层操作的方式将最终用户与硬件隔离开。本文结合视频采集卡的设计实例详细论述了基于 Windows95 平台的实时硬件控制的解决方案——虚拟设备驱动程序的基本原理和设计方法及其与 Win32 应用程序的接口。

关键词　VxD　特权级　Win32　Windows95　视频处理
中图分类号　TP317.4　　文献标识码　A

1　前　　言

Windows95 环境下的硬件设备被系统屏蔽, 在 DOS 环境下可以直接进行的操作, 在 Windows95 平台上可能导致异常错误; 其内存管理采用平面模式、分页式交换, 对内存的访问要经过一个对用户透明且不可控制的地址转换过程。在开发图像显微系统的过程中, 为了提高图像的精确度、降低系统成本, 采用了自行设计采集设备的方案, 因而在实现中不可避免地要访问硬件设备, 例如读/写卡上存储器映象、实时响应中断请求、访问端口等。这时, 普通应用程序无法达到系统性能要求。设备驱动程序是操作系统分隔用户与硬件的手段, 所以根本的解决办法就是开发设备驱动程序。

2　采集系统的组成

内存映象法——将外部扩充存储器当成系统主存储器的一部分, 利用系统主存储器物理地址进行访问, 由于它在操作的速度、灵活性以及实现上的简便性, 在系统设计中, 就选用了该种方法实现主机对视频数据采集卡的内置帧存储器进行访问和控制。

收稿日期: 1999-05-04
修稿日期: 1999-05-17

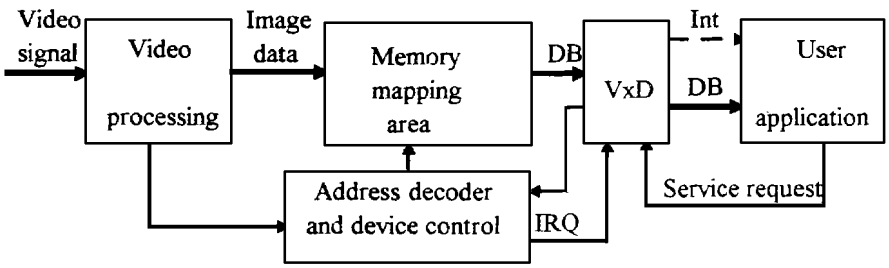

Fig. 1　Schematic of experimental video acquisition system

图1 为视频采集系统的原理框图, 视频信号经预处理和 A/D 转换后送入图象帧存储器。帧存储器分为两部分, 交错处于采集和读取状态。每当一帧图象采集完毕时, VxD 获得系统控制权, 根据用户的设置决定控制的流向。从图中可以看出虚拟设备驱动程序处于整个系统的核心地位, 是应用软件与硬件实时交互的唯一通道。它为软件处理部分封装了所有对硬件的操作, 并且提供了对设备事件的实时响应。显微图像处理系统的软件设计采用结构化程序设计方法来实现, 由不同层次的功能模块组成, 模块内实现代码封装, 向上一层提供规范的接口。图象采集卡硬件驱动层利用 Windows95 环境下的虚拟设备驱动程序完成 CCD 摄像头控制、映象存储器页面切换与访问、地址转换、中断控制和数据交换等直接操纵硬件和系统数据结构的任务, 该层构成了整个软件系统的基础。

3　虚拟设备驱动程序的编写

为了保障系统的安全性, Intel 处理器从硬件上采取层次式体系结构, 提供了四个特权级。在保护模式下, 内核(主要由虚拟机管理程序 VMM 和 VxD 组成) 运行在 Ring0 级, 而用户开发的应用程序大多只能运行于 Ring3 级(其它两个级别未用) 。Intel 的硬件结构提供了专门的保护机制使得 Ring3 级代码不能直接调用 Ring0 级代码, 某些指令的执行也要求有正确的特权级。正是由于受到系统特权级的制约, 使得特权级低的进程所能实现的功能有限。

所谓 VxD, 就是 Windows95 环境下的虚拟设备驱动程序, 它通过截获应用程序使用设备的请求来虚拟化物理设备。当应用程序使用特定的硬件设备时, 它可以透明地仲裁多个应用程序同时进行的访问, 甚至能够模拟硬件的行为来提供一个根本不存在的虚拟设备。VxD 作为操作系统的一部分运行于 Ring0 级, 可以看作是操作系统的一个动态链接库。由于具有最高特权级, VxD 能够完成直接访问物理设备、安装真正的中断处理器、直接修改页表等任务。

3.1　VxD 的控制流程

虚拟机管理程序以事件驱动方式协调多个 Windows 程序的运行。对可动态装载的 VxD 来说, Sys- Dynamic- Device- Init、Sys- Dynamic- Device- Exit 以及 W32- DeviceIoControl 三个消息会由虚拟机管理程序传送给该虚拟设备驱动程序。这三个消息由 VxD 中的控制分配表来决定处理过程。前两个消息分别由 CreateFile 和 CloseHandle 产生, 在 VxD 载入和卸载的时候使用。第三个消息通常由调用 Microsoft 提供的 DeviceIoControl 的应用程序来产生, 被称为

设备输入输出控制(DIOC)请求。根据系统控制消息决定控制流的工作要在虚拟机控制过程中完成,这项工作可以利用 DDK 提供的宏 Control_ Dispatch 来实现。例如:

Control_ Dispatch SYS_ DYNAMIC_ DEVICE_ INIT, VxD_ Init

对初始化动态装载 VxD 的消息进行响应,其中 VxD_ Init 为完成初始化工作的过程的入口地址。操作系统是根据进位标志的清除与设置来得知处理工作的成功或失败,所以虚拟设备驱动程序在返回之前要依据情况调用 clc 或 stc 指令。

Windows95 允许 VxD 提供能被 Ring3 级 Win32 代码使用的接口,将其称作 Win32 VxD 服务。硬件被驱动程序虚拟化之后,就可以作为一个文件而被系统引用。打开设备文件的方法如下所示:

CreateFile("\\\\.\\VxDName", 0, 0, 0, CREATE_ NEW, FILE_ FLAG_ DELETE_ ON_ CLOSE, 0);

接下来就能够从 Win32 应用程序中调用标准的 DeviceIoControl 接口与 VxD 通信。在 VC++ 开发环境中,也可以直接调用系统隐含提供的引出函数 VxDCall,DeviceIOControl 实际上也是间接调用该接口,它具有更快的响应速度,但可移植性较差,并且需要开发人员自行制作系统引出函数库,然后加入到 VC++ 链接库中去,以解决外部函数调用问题。

VxD 是 32 位程序,所以可以利用 32 位 C/C++ 编译器创建虚拟驱动程序。通过 C 例程实现对 Win32 服务请求的响应过程要在虚拟机控制过程中提供宏说明:

Control_ Dispatch W32_ DeviceIoControl, OnDeviceIoControl, cCall, <esi> 用来提供 Win32 服务的处理地址和调用参数的传递方式;而后,在用 C 编写的例程_ OnDeviceIoControl 中则根据输入参数 DIOCPARAMETERS 结构的成员变量 dwIoControlCode 的值判定用户申请的调用。

当采集卡发出中断请求后,VxD 响应并完成数据传输、设置标志及发送消息的工作。其它对时间延迟的要求不是很严格的工作由低特权级的模块实现。由于篇幅所限,下面的内容只说明 VxD 中涉及的系统服务和关键环节,具体参数省略。另外,对端口的存取只需在虚拟设备驱动程序中直接使用汇编指令"in"和"out"即可,所以在下面的内容中不再做详细介绍。

3.2 VxD 处理中断

处理硬件设备的中断请求要依靠虚拟可编程中断控制器驱动程序提供的服务。首先在 VxD 的固定数据段说明 VPICD_ IRQ_ Descriptor 类型的数据结构,并指定中断号以及中断服务例程等参数后,VxD 通过调用 VPIC_ Virtuslize_ IRQ 向系统注册中断服务例程,并保留返回的中断句柄作为以后判别中断类型的依据。

当中断请求信号出现时,控制权会在最小的延迟时间内被转交给中断服务例程(ISR),因此 ISR 必须位于固定代码段中以保证不会出现缺页中断。此时,EAX 装有唯一标识该中断的中断句柄,获得控制的中断服务例程据此判定预期的中断是否到来。否则,必须立即退出以免影响系统其它部分的处理,而且处理过程也应尽可能简洁。中断服务例程处理完毕之后,调用 Shell_ PostMessage 向应用程序窗口发送消息。

在中断服务例程的结尾处调用 VPIC_ Phys_ EOI,该服务仅仅清除中断屏蔽码。在中断服务例程的入口处,可编程中断控制器上特定类型中断的屏蔽码已被系统设置,并且该中断的

中断结束标志也已被送往可编程中断控制器。这时系统已能够响应其它类型的硬件中断,这也是操作系统为提高实时性和并行性采取的措施之一。从中断服务例程返回只需调用 RET 指令,而不是 IRET。如果成功地处理了中断,要在返回之前清除进位标志。如果该中断是被多个 ISR 共享的,通过设置进位标志可以使虚拟机管理程序能够将中断请求传送给中断服务链上的下一个处理例程。

3.3 VxD 访问映象内存

操作系统在将程序调入的过程中,负责将物理地址和线性地址通过映射机制对应起来。在 Windows95 环境下,VxD 位于 Win32 进程 4GB 地址空间的系统区,即在所有的 Win32 进程的地址空间中对应相同的地址。所以,在 VxD 中分配的内存,可以直接将其地址传送给 Win32 应用程序。如果 VxD 需要存取在 Win32 应用程序中分配的私有内存,VxD 必须运行在分配内存的线程文本中,Shedule_Thread_Event 服务可以通过切换线程文本来满足系统的要求。

对内存的访问包括两种情况:一是知道物理地址,想得到程序中可以访问的线性地址,例如即插即用设备中一般都有一段采用存储器映射的寄存器。在 VxD 中访问指定地址的物理内存,需要通过_MapPhysToLinear 服务将物理地址转换成为线性地址。只有在调用 Map_Lin_To_VM_Addr 服务将线性地址转化为"选择符:偏移量"形式的保护模式地址或"段:偏移量"格式的 V86(虚拟 8086)模式地址之后,才能在 Win32 应用程序中使用。服务 Map_Flat 可以将保护模式或者 V86 模式地址转换为线性地址,使得 VxD 能访问应用程序中的内存空间。同时必须保证页面已经锁定,而 Win32 服务没有实现该功能,只有通过调用虚拟机管理器提供的服务_LinPageLock 才能将内存页面锁定,并返回可被任何内存文本使用的页面的地址。所用地址只有在锁定之后,才能保证其长久有效性。解锁内存块则是_LinPageUnlock 的任务。

另一种情况是需要同时得到线性地址及其对应的物理地址,硬件接口需要内存的物理地址,而程序需要的是虚拟地址。向系统申请 PageAllocate 服务完成内存分配是实现这一目的的常用方法。由于应用程序不能保证分配连续的物理内存,为了提高对内存的利用效率,避免数据的两次传输,可以采用应用程序共享虚拟设备驱动程序分配的内存的工作模式。分配内存时要设置其申请标志为 PageContig、PageUseAlign 和 PageFixed 从而保障分配连续且固定的内存块。对任何与虚拟机相关的服务都要预先获得虚拟机的句柄,Get_Sys_VM_Handle 可以实现这一任务。当分配的内存块的使命结束后,必须由 PageFree 释放内存,避免造成内存碎片及泄漏。

由于采集系统所采用的工作方式涉及到固定存储器地址映象区和随机分配的图象处理缓冲区两类地址转换,所以上面所述的两种转换方式必须全部予以实现。一旦转换得到了所需的有效地址,就可以使用访问存储器的指令来访问采集到的数据了。

4 结　束　语

开发 VxD 需要 Windows 设备驱动程序包(DDK)、软件开发包(SDK)以及 VC++。为了加快虚拟设备驱动程序的开发进程,借助其他厂商提供的专用工具(如 VToolsD、VxDWriter)

也是一个较好的选择。这些开发工具向用户屏蔽了操作系统和 VxD 的底层细节,提供了高级语言接口,能够自动生成 VxD 必需的基本组件,配置 VC+ + 集成开发环境的工程文件以正确编译虚拟设备驱动程序。开发人员通常选择汇编语言和 C 语言结合使用。首先创建一个汇编语言程序,用来完成虚拟设备的声明、VxD 的固定代码段(其中包括控制分配过程和 VxD 消息处理器) 和数据段的定义; 其它的部分则可以利用 C 代码来实现,但必须确保对调用规范、需初始化的运行时间库及段的使用的正确性。WinDbg 和 Soft- ice for Windows95/98 是现阶段可用的功能强大的调试工具。

开发虚拟设备驱动程序是在 Windows95 平台上完成实时控制的最佳选择,它具有普通应用程序不可比拟的性能优势,而且不会引入其它方式常有的冗余开销及虚假的操作。VxD 还可以实现软件监视、调试器、改变系统中其它软件的行为表现等强大功能。但也正是由于 VxD 程序能够做任何事情,而不必受到 Ring3 级的操作限制,甚至使系统停止工作,所以开发 VxD 程序一定要仔细权衡,以免影响系统性能。

参 考 文 献

1　米东, 王森等译. Windows95 系统编程奥秘. 北京: 电子工业出版社, 1996
2　程荷, 武航等译. 32 位系统软件编程指南. 北京: 电子工业出版社, 1997
3　吕天宇等译. Windows95 开发指南. 北京: 电子工业出版社, 1995

Virtual Device Driver and Its Application in Video Acquisition Card

TIAN Zhi-Gang, CHEN Gui-Lin
(Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083)

Abstract

Windows95, as an operating system running under 32-bit protection mode, separates hardware from end users through the way of mask in order to keep the system safe. Giving an example of video acquisition card, this paper describes the basic principles and the scheme of virtual device driver which is the solution of real-time hardware control in detail. And its interfaces to Win32 applications are also introduced.

Key Words: VxD, Privilege level, Win32, Windows95, Video processing

田志刚　男, 1973 年生。毕业于黑龙江大学计算机软件专业。现为中国科学院上海技术物理研究所博士研究生,主要从事图象获取、信息处理以及光电检测等领域的研究。