文章编号:1008-0570(2007)09-1-0279-03

用 DriverStudio 开发 PCI 总线设备驱动程序

Development of PCI bus device driver using DriverStudi

(国防科技大学)杨 波 柳 征 姜文利

YANG BO LIU ZHENG JIANG WENLI

摘要:以 PCI9656 为例,介绍了利用 DriverStudio 开发 PCI 设备驱动程序的方法步骤,并实现了具有 I/O 和内存读写、中断处理和 DMA 传输功能的驱动程序。

关键词:PCI 局部总线;驱动程序;DriverStudio

中图分类号:TP311 **文献标识码:A**

Abstract: Through an example of PCI9656,the method and step that develop the driver for PCI device is introduced. The driver for PCI device that can read and write memory,I/O and handle interrupt,and DMA transmission is realized in this paper.

Key words:PCI local bus,device driver,DriverStudio

PCI(Peripheral Component Interconnect,即外围部件互连)总线是一种高性能的且独立于处理器的 32/64 位地址数据复用总线,它与 CPU 无关,与时钟频率亦无关,可适用于各种平台。它支持无限读写猝发操作,支持并发工作方式。PCI 总线以其优良性能和可适应性成为微型计算机系统的主流总线。开发微机接口设备,通常采用 PCI 总线。

Windows 操作系统为了保证系统的安全性、稳定性和可移植性,不支持应用程序对硬件资源的直接操作。因而在设计开发 PCI 设备时,往往需要硬件开发人员自行开发设备驱动程序来实现对 PCI 设备的操作,应用程序通过驱动程序来访问 PCI 设备。

开发 PCI 驱动程序,笔者使用 DriverStudio。DriverStudio 是一套 NuMega 公司为简化 Windows 设备驱动程序和应用程序而开发的编写、编译和调试的软件工具包。它以面向对象(OOP)的方式,将驱动程序编写所需的与内核访问及对硬件的访问封装成类,加上设计的驱动程序代码生成向导,大大简化了驱动程序开发的难度,减少了工作量。同时,DriverStudio 被嵌入到 VC 中,方便了开发。本文根据笔者编写 PCI9656 驱动的实际经验,介绍使用 DriverStudio 开发 PCI 设备驱动程序的流程和方法。

1 利用 Wizard 生成 PCI 设备驱动程序框架

DriverWizard 是 DriverStudio 创建框架程序的工具,它能够生成驱动程序的基本框架和用户自定义信息,虽然没有实现 PCI 设备的具体功能,但对于功能的实现设置了框架,功能代码都在该框架下完成。创建程序框架,共有 11 个步骤,大部分步骤采用默认设置即可,下面介绍其中的几个关键步骤:

(1) 第 4 步,选择设备的类型并填写相应信息。我们选择 PCI,并填写 PCI 设备的设备标识符。PCI Vendor ID 是厂商标识符,此例填 10B5h;PCI Device ID 为设备标识符,填 9656h;PCI Subsystem ID 为子系统 ID 号,填默认设置 00000000 即可;PCI Revision ID 为修订号,填 BAh。这些信息由 PCI 设备在 PCI 总线配置时提供给系统,系统根据这些信息来匹配已注册的 PCI 设备的驱动程序。

(2) 第 7 步,选择 IRP 串行处理的类型和串行处理的函数。分发例程可以将 IRP 进行排队,交给统一处理 IRP 的例程 StartIo 串行处理。串行的处理例程有两种类型:驱动程序自行处理和系统处理。驱动程序可以设置多个 StartIo 处理函数,对于 PCI 设备来说,一般只对一个设备操作,只需要一个 StartIo 处理。一般处理 ReadFile 和 WriteFile 函数的分发例程需要进行 IRP 排队,在此全部选中。

(3)第 9 步,选择设备文件中的类名和接口类型,类名通常取默认值。在 Resources 项中,需要声明所需的硬件资源如存储空间和 I/O 空间,中断和 DMA 等。如图 1 所示。

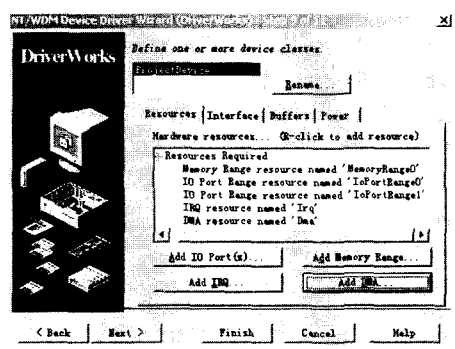

图 1 DriverStudio 驱动程序向导第 9 步

最后一个步骤是选择生成一个 Win32 Console 应用程序。单击 Finish 按钮,驱动程序框架就生成好了。

2 添加功能代码

在开始驱动程序代码编写前,我们先了解一下操作系统与驱动程序之间的通信方式。当用户模式程序需要读取设备数据时,它就调用 Win32 API 函数,如 Readfile。在 ReadFile 调用中,调用首先到达系统 DLL(NTDLL.DLL)中的一个入口点:NtReadFile 函数。NtReadFile 函数接着调用系统服务接口,最后由系统服务接口调用内核模式中的服务例程,该例程同样名为

杨 波:研究生

NtReadFile。

驱动程序的编写主要有三个方面的问题:硬件访问、中断处理和 DMA 传输。以 PCI9656 驱动代码为例,下面就这三个方面的编程展开讨论。

2.1 硬件访问

2.1.1 I/O 端口的访问

类 KIoRange 封装了对 I/O 端口的操作,它实现对 I/O 映射芯片的访问。对于所设计的 PCI 设备,其配置空间的基地址寄存器(0~5)值决定请求的地址空间类型和大小,驱动程序根据以上情况声明 KIoRange 类实例。例如:

```
KIoRange    m_IoPortRange0;
KIoRange    m_IoPortRange1;
```

对 PCI 设备首先需要在 OnStarDevice 函数中取得 I/O 资源,然后初始化。如下:

```
status=m_IoPortRange1.Initialize(
pResListTranslated,    //转换后的资源列表指针
pResListRaw,           //原始的资源列表指针
PciConfig.BaseAddressIndexToOrdinal(1) //'1' 代表分配的 I/O 资源在 PCI
//基地址的序数,PCI 设备最多可以有 6 个 I/O 资源空间
);
```

参数 pResListTranslated 和 pResListRaw 在 DDK 中从 IRP_MN_START_DEVICE的 IRP 中取出。DriverStudio 封装了 IRP 为 KIrp。在 OnStartDevice 函数中由函数 AllocatedResources 和 TranslatedResources 得到上述参数。

初始化完之后,就可以调用访问函数了,如下所示:

```
status=m_IoPortRange0.ind(INTCSR);
m_IoPortRange0.outd(DMAMODE0,0x20800);
```

在驱动程序停止运行之前,一定要卸载资源,对于 I/O 端口调用成员函数 Invalidate,如下所示:

```
m_IoPortRange0.Invalidate( );
m_IoPortRange1.Invalidate( );
```

2.1.2 内存的访问

类 KMemoryRange 封装了对 PCI 设备的映射内存的操作,它实现对映射内存的访问。在 WDM 中,驱动程序得到的只是 PCI 总线配置机构分配的物理内存,如果在驱动程序中是无法使用的,必须将其转换成系统可以访问的非分页内存,在 DDK 中,调用函数 MmMapIoSpace。在 KMemoryRange 的成员函数 Initialize 中实现这个转换。在 OnStartDevice 函数中,如下:

```
status=m_MemoryRange0.Initialize(
pResListTranslated,
pResListRaw,
PciConfig.BaseAddressIndexToOrdinal(0)
);
```

函数的参量同 I/O 端口的操作。KMemoryRange 的其他成员函数操作同 I/O 端口的操作。

2.2 中断的处理

KInterrupt 类实现硬件中断的处理。中断服务例程不是 KInterrupt 类的成员函数,这样做的目的是减少中断延迟时间。如果中断延迟时间不是问题,可以用 MEMBER_ISR 宏声明中断服务例程为 KDevcie 或其它类的一个成员函数。

中断处理需要中断服务例程和延迟过程调用例程,需要声明:

```
class PCI9656Device:public KPnpDevice
{…
puclic:
MEMBER_ISR(PCI9656Device,Isr_Irq); //声明连接的中断服务例程
BOOLEAN Isr_Irq(void);          //声明中断服务例程函数
VOID DpcFor_Irq(PVOID Arg1,PVOID Arg2);
protected:
KInterrupt    m_Irq;            //中断变量
KDeferredCall  m_DpcFor_Irq;    //延迟调用
}
```

在 OnStartDevice 函数中,调用 KInterruptr 的成员函数 InitializeAndConnect 初始化中断变量和调用宏 LinkTo 连接中断例程。如下:

```
status=m_Irq.InitializeAndConnect(
pResListTranslated,          //转换后的资源列表指针
LinkTo(Isr_Irq),             //连接中断服务例程
this
);
```

在中断服务例程中,相应的处理时间应当尽可能地短,因为中断例程运行的级别很高,当有中断请求时,不但会打断应用程序的执行,而且会打断在硬件中断级以下的所有运行程序。所以,在 WDM 中提供了 DPC(Deferred Procedure Call)例程,将在中断例程中耗时的但不需要立即处理的任务延时处理。DPC 的操作封装成类 KDeferredCall,其用法如下:

```
m_DpcFor_Irq.Setup(LinkTo(DpcFor_Irq),this);
```

在中断服务例程中,首先必须根据硬件信息来判断中断是否是自己的设备产生的,若不是,则返回 FALSE;若是,进行必要的处理,中断服务例程完成后,则请求一个 DPC,一旦处理器获得 DISPATCH_LEVEL 级别的运行权,内核就运行它的延迟过程调用,可以在延迟过程调用例程中做大部分的中断处理工作。中断服务例程的流程框图如图 2 所示。

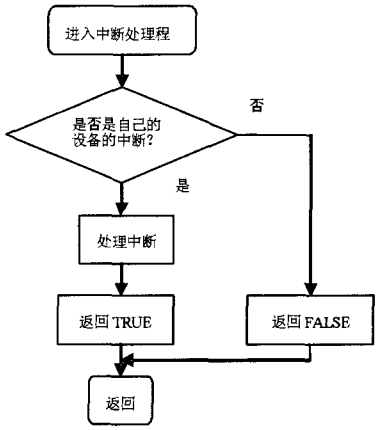

图 2 中断服务例程的流程框图

在驱动程序停止运行之前,一定要调用类 KInterrupt 的成员函数 Invalidate 断开中断的连接,释放资源。如下:

```
m_Irq.Disconnect();
Invalidate();
```

2.3 DMA 传输

PCI 设备根据总线访问能力分为主模式和从模式两种,主模式设备可以发起总线操作,而从模式只能接受总线操作。主模式设备可以进行 DMA 操作,从模式没有该功能。

DriverStudio 提供了三个类:KDmaAdapter,KDmaTransfer 和 KCommonDmaBuffer 类,用于实现 DMA 操作。KDmaAdapter 类

用于建立一个 DMA 适配器,它说明 DMA 通道的特性。DMA 数据传输的具体操作则由类 KDmaTransfer 来实现的。在 DMA 的传输中,PCI 设备将数据直接传输到系统物理内存中,管理这些内存的方式有两种:Common Buffer 和 Packet。第一种方式是在系统的物理内存中预先开辟一段地址连续的内存空间,使 CPU 和 PCI 设备都能对其访问,管理这种功能的是类 KCommonDmaBuffer。另一种方式是使用 MDL(Memory Descriptor List)描述的内存空间来作为 DMA 传输数据的源或目标。

首先要声明 DMA 编程的三个类实例和 DMA 准备就绪回调例程,如下:

```
public:
DEVMEMBER_DMAREADY(PCI9656Device,OnDmaReady)
protected:
KDmaTransfer m_DmaTransfer;
KDmaAdapter  m_Dma;
KCommonDmaBuffer m_CommonDmaBuffer;
```

在函数 OnStartDevice 中,初始化 DMA 变量:

```
m_Dma.Initialize(&dd,m_Lower.TopOfStack());
m_CommonDmaBuffer.Initialize(m_Dma,0x100);
```

在 SeriaWrite 函数中,启动 DMA 如下:

```
void PCI9656Device::SerialWrite(KIrp I)
{...
FromMemoryToDevice, //传输方向,或 FromDeviceToMemory
LinkTo(OnDmaReady), //连接的回调函数
}
```

DMA 传输结束后,应当使实例 m_Buffer 无效:

```
m_Buffer.Invalidate();
```

3 驱动程序的安装与调试

Windows 使用一个扩展名为 INF 的文本文件来控制与安装 PCI 设备驱动程序相关的信息。DriverWizard 已经给我们在所建工程的 SYS 目录下生成了该 INF 文件。在安装之前,我们首先将 INF 文件中[Strings]项的提示改为相应内容,然后将该文件拷贝到"..\sys\objchk\i386"目录下。驱动安装可运行"控制面板"里的"添加硬件向导",根据提示指定我们修改过的 INF 文件来完成安装;也可使用 DriverStudio 中的 EZDriverInstaller 工具,可以快速地安装所测试的 PCI 驱动程序。

调试 PCI 驱动程序,我们使用 DriverStudio 中的 SoftICE。SoftICE 是一个功能非常强大的调试软件,它结合了硬件调试器的强大功能和符号调试程序的易用性,能够显示程序的源代码,允许通过符号名访问局部和全局的数据。基本调试过程如下:(1)安装驱动程序,然后使用 SoftICE 跟踪调试,确认驱动程序正常加载;(2) 对核心的中断响应程序代码,用 SoftICE 中的 Genint 命令产生虚拟中断,单步跟踪中断;(3)向 PCI 发送大量的数据,通过查看内存的数据,确认数据传输是否正确。

由于 PCI 设备的硬件资源是动态配置的,只有当 PCI 设备上电后,驱动程序才能获得资源信息,而且驱动程序具有与操作系统相同的特权,能够直接操作硬件,所以驱动程序的调试要非常小心,最好采用循序渐进的方法,一步一步实现功能,否则,系统很容易出现"死机"、"蓝屏"等现象。

4 结论

较为流行的驱动开发工具还有微软公司提供的软件包 DDK(Device Driver Kits)和 Jungo 公司的 WinDriver。采用 DDK 开发驱动程序需要深入了解操作系统的内核工作方式和驱动程序的工作细节,虽然程序代码效率高,但开发难度较大;采用 WinDriver 很容易进行驱动的开发,花费的时间很少,事实上开发者只需定制和调用它提供的通用驱动而已,因此程序的执行效率不高。DriverStudio 中的函数库封装了针对驱动程序的各种通用操作,大大减少了驱动程序的代码长度,开发者可以使用向导直接进入到驱动程序的开发,而且 DriverStudio 还提供了一套完整的驱动程序安装测试工具 EZDriverInstaller、SoftICE,使用 DriverStudio 可以在缩短开发周期的同时开发出高质量、结构化的驱动程序。在目前,从速率和效率的综合考虑,用 DriverStudio 来开发驱动程序是最好的选择。笔者通过使用 DriverStudio 开发的 PCI9656 驱动程序应用在基于 PCI 总线的数据采集卡上,在实际工作中稳定可靠,达到了预期的效果,具有较高的应用价值。

本文作者创新点:本文开发 PCI 驱动程序,硬件采用的是 PCI9656.PCI9656 是 PLX 公司为扩展适配板卡推出的能提供混合高性能 PCI 总线目标模式的接口电路,跟以往的接口电路相比,其主要优点有:支持 64 位、66MHz 时钟 PCI 总线;配有 DMA 引擎、可编程直接主控;具有 PCI 优先判决器;局部时钟与 PCI 时钟异步工作;支持多路复用和非多路复用;最大数据传输速率达 528Mb/s,远远大于 ISA 总线 5Mbyte/s 的速度。软件采用 DriverStudio。DriverStudio 克服了以往工具 DDK 开发周期长和 WinDriver 代码效率低的缺点,能最大限度地缩短开发周期、提高程序效率,编制出结构化的驱动程序。PCI9656+DriverStudio,实现了软硬件的完美结合,是目前开发 PCI 驱动的最好选择,具有较高的实用价值。

技术创新

项目经济效益:50 万元

参考文献:

[1]Walter Oney (美).Programming the Microsoft Windows Driver Model[S],1999.

[2]Microsoft.Windows 2000 DDK Documents[M],2000.

[3]武安河.Windows 2000/XP WDM 设备驱动程序开发[M].北京:电子工业出版社,2005.

[4]尹勇,李宇.PCI 总线设备开发宝典[M].北京:北京航空航天大学出版社,2005.

[5]郜铭,武安河.Windows 下 PCI 接口卡 WDM 驱动程序的 DMA 编程技术[J]微计算机信息,2003,(10):79-81.

[6]周云,黄巍,汪学刚.PCI 接口卡硬件与驱动程序设计[J].电讯技术,2004,(4):143-146.

作者简介:杨波,男,1976 年 7 月出生,广西玉林人,国防科技大学电子科学与工程学院研究生,主要研究方向:通信与信息系统。

Biography: Yang Bo(1976-), man, Guangxi Province, National University of Defence Technology, Master, Research area:communication and information system.

(410073 湖南长沙 国防科技大学 电子科学与工程学院) 杨波 柳征 姜文利

(School of Electronic Science and Engineering,National University of Defence Technology,Changsha 410073,China) Yang Bo Liu Zheng Jiang Wen-li

通讯地址:(410073 湖南 湖南长沙市丽臣路国防科技大学继续教育学院学员二队三中队)杨波

(收稿日期:2007.7.23)(修稿日期:2007.8.15)