用 Delphi 开发 Windows 下硬件设备驱动程序的方法

鞠　阳　(南京电力高等专科学校电信系　210013)

摘要:本文介绍了在 Windows 环境下采用 Delphi 语言设计硬件设备中断程序的原理和方法。

关键词:Windows　Delphi　中断　驱动程序

1.引言

Delphi 是一个面向对象、优秀的可视化软件开发语言。它可以高效地开发出美观、精巧的 Windows/Windows95/WindowsNT 应用程序。它利用所谓视觉化的环境来支持所有的应用软件的界面设计。其整体结构十分完整,从抽象层次极高的程序产生器到 Windows 程序设计最基本元素之一的 Windows API 都有完整的支持。在这一方面,可以说是目前市面上最杰出的 Windows 软件开发工具之一。Delphi 的精华还在于强大易用、有效率的视觉化组件库(VCL)。VCL 不仅仅是体系完整的类库,它更提供视觉化设计环境的支持。对于程序设计者而言,设计程序时看到的界面不再只是对象类的属性、方法等代码,对每个对象类可在设计阶段就能直接看到它执行时的轮廓。

Delphi 提供了功能强大的快速程序开发工具,利用其真编译系统编译出的可执行文件的效率相当高,其程序运行速度与 C/C++ 相当。另外,它支持对低层输入输出端口的直接访问和对外部实时时件中断申请的支持。这在对于实时性要求很高的工业控制系统中,具有非常重要的现实意义。

2.中断方式原理和实现方法

硬件设备的控制实质就是从外设上读取设备状态,把控制信号传送到外部设备,在微机与扩展卡之间实现数据传输和交换的过程。微机与外设进行数据交换的方法常用的有查询方式、中断方式和 DMA 方式三种。由于中断方式具有 CPU 利用率高、不需增加额外硬件、实时性好等特点,因而得到广泛应用。下面主要就中断方式展开讨论。

在 DOS 下,硬件中断程序的设计较为简单。因 DOS 是单任务操作系统,当一有中断请求发生,立即可得到响应。但 Windows 是多任务操作系统,是一个消息驱动式系统。Windows 对每一个输入事件都产生一个消息,并把这些消息一起放入一个应用程序队列中。应用程序队列是属于各个应用程序所有的先进先出队列。这就有可能使得一些外部实时事件得不到及时响应。为避免此现象发生,这里采用了 Delphi 提供的二个重要函数 GetIntVec 和 SetIntVec 来截获外部中断申请,在中断服务程序中直接发送消息给应用程序的消息队列。由于它绕过了系统的消息队列,因此具有较好的实时性。

下面介绍一下中断函数的使用方法。

(1)取中断向量函数

GetIntVec(num:Integer; save－p:Pointer);

其中:

num －－－ 中断向量类型码。

save－p －－－ 保存 num 号中断的中断向量,供以后恢复原中断向量用。

(2)置中断向量函数

SetIntVec(num:Integer; int－p:Pointer);

其中:

num －－－ 中断向量类型码。

int－p －－－ 中断服务程序入口地址可用 Addr(函数名)取得。

3.Windows 下中断程序设计

下面以笔者开发的 1:8 串行通信扩展卡为例说明程序的设计方法。扩展卡上有 8 片可编程串行通信芯片 8251、1 片中断控制器芯片 8259A 及一些辅助电路构成 8 路 RS－232－C 串行通信接口电路。微机工作在中断方式时,可同时与 8 路外设进行串行数据通信。

为简化程序,就以微机接收其中 1 路外设数据为例说明中断程序的设计方法。在硬件上,扩展卡上 1 号 8251 芯片的 RxRDY 端接至扩展卡上中断控制器芯片 8259A 的 IR0,8259A 的 INT 端作为外部硬件中断申请信号接到微机内中断控制器芯片主 8259A 的 IR5 端,IR5 的中断号为 0DH。

下面程序采用基于 Windows 下的 Delphi 2.0 编写。

由于篇幅有限，只给出与中断程序有关的几个子程序，其他程序省略。

```
VAR                                    (*定义全局变量*)
  Save 0d Pointer:Pointer;             (*保存原 0d 号中断向量*)
  Recrive Buffer:Buffer;               (*接收数据缓冲区*)
  Receive Word Count:Integer;          (*接收数据计数器*)
Implementation
PROCEDURE MySetVect;(*设置中断向量子程序*)
BEGIN
  GetIntVec($0d,Save 0d Pointer);      (*保存原 0d 号中断向量地址*)
  SetIntVec($0d,Addr(MyIntISR));       (*指向新的中断向量地址*)
END;
PROCEDURE RestoreVect;(*恢复原 0d 号中断向量地址*)
BEGIN
  SetIntVec($0d,Save 0d Pointer);
END;
POOCEDURE Init8259A;(*初始化扩展卡上中断控制器芯片 8259A*)
BEGIN
  Port[$110]:=$13;      (*输出 ICW1,$110 为偶地址*)
  Port[$111]:=$40;      (*输出 ICW2,$111 为奇地址*)
  Port[$111]:=$01;      (*输出 ICW4*)
  Port[$111]:=$0fe;     (*输出 OCW1,允许扩展卡上 8259A 的 IR0 中断*)
  Port[$110]:=$0c2;     (*输出 OCW2*)
  Port[$20]:=$0c4;(*初始化微机内中断控制器芯片主 8259A,$20 为偶地址*)
  Port[$21]:=Port[$21] AND $0df;(*输出 OCW1,允许微机内主 8259A 的 IR5 中断,$21 为奇地址*)
END;
PROCEDURE MyIntISR;   (*中断服务程序*)
BEGIN
Receive Buffer[Receive Word Count]:=Port[$100];(*$100 为扩展卡上 1 号 8251 芯片数据口地址。即把接收到的数据放入缓冲区,以便进行数据处理、显示等工作*)
Receive Word Count:= Receive Word Count+1;(*接收数据计数器加 1*)
Port[$110]:=$20;(*发 EOI 信号,清扩展卡上 8259A 中断请求寄存器内容*)
Port[$20]:=$20;(*发 EOI 信号,清微机内主 8259A 中断请求寄存器内容*)
END;
```

4. 结束语

由于微机采用基于 Windows 下的 Delphi 2.0 进行程序设计，并且使用中断方式进行数据通信，因此整个装置具有较好的易用性、较快的响应速度和友好的用户界面，很好地满足了工业控制中对实时性的要求。

参考文献

[1] 汪亚文. Delphi 程序设计指南. 学苑出版社，1995

[2] 张育荣. Delphi－－从入门到精通. 清华大学出版社，1996

[3] 毕维生. Delphi 入门及实例详解. 电子工业出版社，1996

（来稿时间：1998 年 7 月）