Windows 下设计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 环境下采用中断方式和查询方式为硬件设备编写驱动程序的原理和方法。

关键词：Windows　驱动程序　中断　查询

1. 引言

在电力系统中，广泛使用 SCADA（Supervisory Control And Data Acquisition，监视控制和数据采集）系统对电网运行进行监视控制，以保证电力生产的安全性、可靠性和实时性。SCADA 系统由主站端和厂站端二部分组成。主站端以 PC 机为核心，实时接收厂站端 RTU（Remote Terminal Unit，一种以单片机为核心的远方监控装置）传送来的远动信息（反映发电厂或变电所的电压、电流、功率、开关状态等信息），然后进行数据处理、显示和实时控制。

主站端采用串行通信方式接收厂站端发送数据。因主站端需同时接收 n 个厂站端的远动信息，所以需要设计 1:n 串行通信扩展板，插入 PC 机扩展槽中，并编写相应硬件设备驱动程序，这样才能正常工作。由于 Windows 以其强大的功能和优秀的界面在工业控制中得到广泛应用，因此主站端 PC 机采用了基于 Windows 下的程序设计。下面就 Windows 下用中断方式和查询方式设计硬件驱动程序的方法展开讨论。

2. 设计方案

1:n 串行通信扩展板使用了 n 片可编程串行通信芯片 Intel 8251 及一些辅助电路构成 n 路 RS－232－C 串行口。Intel 8251 有二种工作方式，即异步串行通信和同步串行通信方式。当工作于异步时，通信速率较低，对硬件设备要求不高，因此 PC 机可采用查询方式读取数据；当工作于同步时，通信速率较高，发、收双方硬件严格要求保持同步，实时性较强。这时若仍用查询方式读取数据，则会造成通信中丢失数据，严重时甚至只能收到同步字，收不到有用数据。因此必须使用中断方式读取数据。由以上分析可知，主站端 PC 机需具备中断和查询二种方式接收数据的能力。

3. Windows 下查询方式程序设计

查询方式的设备管理，即按用户的要求，从外设读入/写出数据和状态，根据状态标志位来决定是否读/写数据。查询方式的特点是对硬件设备要求较低，程序设计简单。缺点是实时性较差，当数据传送速率较高时，会丢失数据。

下面以主站端 PC 机接收一路厂站端数据为例，说明查询方式接收程序设计方法。由于篇幅有限，其他程序省略。程序用 BC＋＋编写。

```
# include<windows.h>
# include<conio.h>
# include<stdlib.h>
# include<dos.h>
# define DATA－8251－1 0x100 // 1 号 Intel 8251 数据口地址
# define STATE－8251－1 0x101 // 1 号 Intel 8251 状态口地址
extern int *buf1; // 接收 1 号厂站数据缓冲区
int count; // 计数器
void receive1();
void receive1()
{
int zt;
for (count=0; count<=300; count＋＋)
{ lp:
zt=(inp(STATE－8251－1))&0x02; // 判状态口 D1 位 RxRDY 是否为 1
if (zt==0) goto lp;
buf1[count]=inp(DATA－8251－1); // 读取数据存入缓冲区
}
}
```

4. Windows 下中断方式程序设计

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。这就使得一些外部实时事件可能得不到及时处理。为避免此现象发生，这里采用了 Windows 下的 INT 31H 设置中断向量，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。

下面仍以主站端 PC 机接收一路厂站端数据为例，说明中断方式接收程序设计方法。硬件上，1 号 Intel 8251 芯片的 RxRDY 端接至 PC 机内中断控制器主 8259A 的 IR3 端，作为外部硬件中断申请信号，IR3 的中断号为 0BH。

```
# include<windows.h>
# include<conio.h>
# include<stdlib.h>
# include<dos.h>
# define DATA-8251-1 0x100 // 1 号 Intel 8251 数据口地址
# define S-8259 0x21 // 主 8259A 中断屏蔽寄存器
# define R-8259 0x20 // 主 8259A 中断请求寄存器
# define IR3 0x0b // 主 8259A IR3 中断号
extern int * buf1; // 接收 1 号厂站数据缓冲区
int count; // 计数器
int ddz; // 存原中断向量段地址
int pyl; // 存原中断向量偏移量
void clxs(); // 数据处理显示子程序
void initial(); // 初始化子程序
void interrupt far Interface(); // 中断程序
void initial()
{
asm{
    cli;
    mov ax,0x204;
    mov bl,IR3
    int 0x31;
    mov ddz,cx;
    mov pyl,dx; // 存原中断向量入口地址
    mov ax,0x205;
    mov bl,IR3;
    mov cx,seg Interface;
    mov dx,offset Interface;
    int 0x31; // 指向新的中断向量入口地址
    sti;
    }
outp(S-8259,(inp(S-8259)&0xf7)); // 允许 IR3 中断
}

void clxs()
{
for (count=0; count<=300; )
{
.....
.....
..... // 数据处理显示
}
asm{
    cli;
    mov ax,0x205;
    mov bl,IR3;
    mov cx,ddz;
    mov dx,pyl;
    int 0x31; // 恢复原中断向量入口地址
    sti;
    }
outp(S-8259,(inp(S-8259)|0x08)); // 禁止 IR3 中断
}

void interrupt far Interface()
{
buf1[count]=inp(DATA-8251-1);
count++;
outp(R-8259,0x20); // 发 EOI 信号，清主 8259A 中断请求寄存器内容
}
```

5. 结束语

由于主站端 PC 机采用基于 Windows 下的 BC++ 进行程序设计，并且使用了中断和查询二种方式接收现场实时数据，因此整个系统具有较好的易用性、较快的响应速度和友好的用户界面，很好地满足了电力系统对实时性的要求。

主要参考文献

[1] 程铁皋，Windows 动态数据交换程序设计，北京航空航天大学出版社，1995

[2] 魏彬，Windows 3.0 软件开发指南(三)，清华大学出版社，1992

(来稿时间：1998 年 1 月)