

股票行情实时传输系统

戴志远 张雪松 (暨南大学)

摘要:通过计算机的串行通信口,使用 MODEM 和多用户卡在 Windows 3.1 环境下实现了对远方股票行情和交易数据的实时传输。

一、概述

现在已有的股票交易传输系统,多数是在 DOS 下工作的。系统设计是单任务工作型,一般是把传输和显示分开实现。这种设计的缺点为:浪费系统资源;界面不友好;操作不方便。所以我们在 Windows 环境下将原有的传输和显示系统进行改进,使它们在 Windows 环境下可以并发执行。而且可以和其他应用程序协调地共享资源。这不仅实现了多任务机制,而且也充分利用了 Windows 的图形界面优点。

行情显示部分用 Foxpro for Windows 2.5 实现。考虑和以前 DOS 版本的兼容以及开发的速度,行情显示和行情传输两模块之间采用文件接口方式。即传输模块

把接收到的数据按 dbf 文件格式存放。而显示部分则直接读取 dbf 文件。这样接口方式简单。便于分组开发设计。传输模块则是在 Windows 下通过串行口对 Modem 进行操作。Windows 3.1 提供了一组通讯函数(见附录 A),利用这些函数以及 Windows 的多任务机制,可以方便地设计一个具有实时响应和多任务能力的传输系统。

MODEM 有一个标准的 AT 命令集。在命令(COMMAND)状态时,给 MODEM 发一个 AT 命令,执行完毕后,MODEM 要返回一个状态码,表示该命令的执行情况。也就是说,在给 MODEM 发命令和 MODEM 返回状态码这两个事件之间是有时间延迟

的。为了正确地接收到 MODEM 的状态返回码,一般的 DOS 应用程序采用查询或中断的发式读 MODEM。而在 Windows 下,利用其多任务机制和 Windows 的通讯函数可以随时给 MODEM 发命令,而且只在 MODEM 的状态码返回之后,应用程序才去读返回码,不必循环等待,实现了系统的多任务机制。

当 MODEM 在数据(DATA)状态(即 MODEM 正通过电话线路传输数据)时,也可以按类似的原理来操作。MODEM 传输数据都有规定的协议。出于安全性考虑,一般要进行数据校验。采取的方法一般是发送方发送一帧数据时,在其中加入校验码,接收方收到数据后,按预先规定的算法对接收到的数据进行校验。若结果和发送过来的检验码相符,则数据正确;否则,数据在传输过程中出现错误。校验完毕之后,接收方给发送方返回一个状态码,表明接收数据的正确与否。而发送方则依据这个状态码来决定下一步的工作。

同命令状态一样,数据状态时的发送方在送完数据和接收到返回码两事件之间,也是有时间延迟的。(一般规定为 3 秒钟,超过 3 秒钟则认为线路出错)。所以发送方在发送完一帧数据后,就出让 Windows 的系统控制权,当有状态码返回时,应用程序再去读返回码并作进一步的处理。这样就可以对通讯事件实时响应(有事件时立即响应),并且具备多任务能力(事件处理完毕立即出让系统控制权)。这也是本股票行情实时交易传输系统的设计思想。

二、传输数据帧结构

由于本系统对数据安全性要求较高,而且数据量不是很大,所以选用高级数据链路控制规程(HDLC)的校验方法和帧结构。数据按帧发送,每帧数据长度为 1024 字节(帧结构见图 1);每一帧发出后,三秒内没有回应或响应为错误,认为出现线路故障或其他错误。发送方重发三次,以确定故障是否排除,若没有排除则系统自动重新启动。每一帧送出后,若响应正确则续发下一帧直到数据完毕。对这个标准的 HDLC 帧略作修改,主要在帧头和帧尾,见图 2。其中数据字段用来存放软件握手信号(数据传输前后),而在传输数据过程中则存放具体的传输数据,双方根据 HDLC 帧的 16 比特 FCS 校验方法计算该帧的 FCS,用作校验之用。本系统中帧的最大长度为

1024 字节。

标志	地址	控制字段	数据内容	FCS	标志
01111110	8 比特	8 比特	长度不定	16 比特	01111110

图 1 HDLC 标准帧结构

开始标志	帧序号	帧长度	数据字段	FCS	结束标志
4 字节	4 字节	2 字节	1000 字节	16 比特	4 字节

图 2 本文中的数据结构

三、传输过程的描述

传输过程首先起始于 MODEM 的拨号,当设好波特率等一系列参数后(作为外部参数设定),启动系统时,先引导初始化模块,该模块根据系统参数给 MODEM 一条条发命令。每一条命令执行成功后,就取下一条命令执行,直到执行完毕。这样就完成了初始化工作。

初始模块引导完毕,则本系统通过 MODEM 发一个申请帧,请求和远方交易中心的服务器建立联结。这部分称为联结模块。具体方法是本地 MODEM 自动拨打对方电话号码,等待服务器端的响应。双方 MODEM 响应后系统向服务器发送一个申请联结帧。服务器收到后若同意申请,则回送一个同意申请帧。这样双方就完成了联结工作。

本系统的传输共有三种数据类型:1.股票行情的文件传送(服务器发送)。2.委托报盘的记录传送(本地买卖股票后由本地系统发送)。3.成交回报的记录传送(股票买卖情况的回送,由本地申请,服务器发送)。

每种数据传输都是由本地机发起的,它向服务器发送一个申请帧(不同数据类型,代码不同)服务器根据情况做出回应。若收到同意传送的帧,则本地机回送一个帧已收到的信息,然后由发送方开始发送数据。接收方接收数据并检验正确,则回送一个接收正确帧。发送方则继续发送,错误则回送一个接收错误帧,由发送方重发刚才的数据帧(若一帧数据三次发送后仍然不正确,则认为出现致命错误。系统中止当前工作,重新启动),这样直到数据发送完毕。此时发送方发送一个数据结束帧。在收到回应后,结束双方的收发工作。

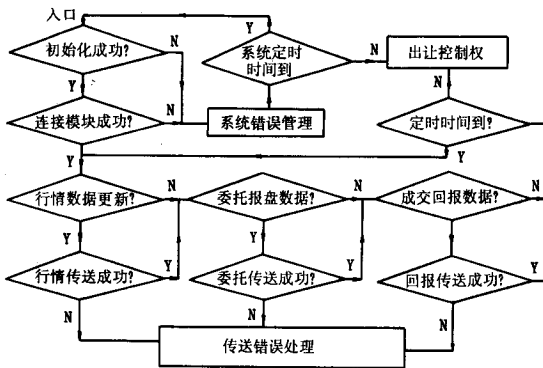


图 3 系统工作流程简图

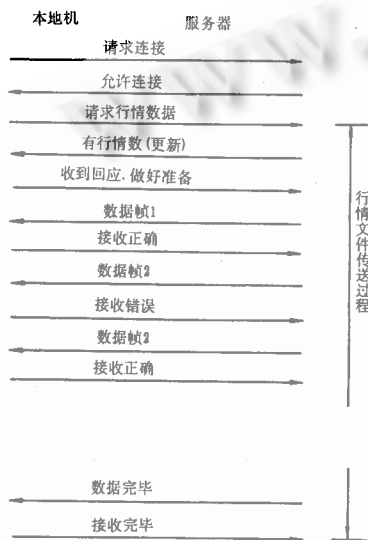


图 4 行情文件数据传输过程

要说明的是,每次本地机向服务器申请时,我们设计一个循环的方式:(1)先申请行情文件(看股票行情是否更新;更新则请求行情数据)(2)传送委托报盘记录,(若有股票的买卖,则本地机请求向服务器发送数据)(3)成交回报记录的传送(若有成交回报,则由服务器将成交回报发送过来)。这三个过程完成之后,启动定时器,在设定时间(如 3 分钟)之后再启动这样一个循环。因而可以随时了解远方行情,并可以随时发送本地数据。定时时间由用户在外调整。

四、定时器

定时功能对于一个通讯系统来说是极为重要的。为

了确保系统安全有效地运行,在线路故障,通讯错误以及出现致命错误时,都要使用定时功能来进行错误处理。

定时功能是在系统中安装一个定时器,并对定时器消息和通讯口消息进行综合判断处理。

每当成功地向通讯口写入数据之后,就打开定时器。若超时(一般为 3 秒)没有响应,则重发数据。三次重发之后没有响应,就认为系统出了致命错误,由第三次定时器消息重新启动系统。每当从通讯口读取数据之前,就关闭定时器,以免定时消息误导系统的运行。所以通讯口消息的优先级高于定时消息的优先级,因而确保前者不会被后者屏蔽而误导系统。

出现致命错误导致系统关闭后,打开定时器,在设定的时间(如 30 秒)以后再启动系统,这就是系统的自启动功能。同样的道理可以进行延时发送。

五、结论

股票行情传输系统具备实时响应和多任务能力。在 Windows 3.1 下可以和其他应用程序协调工作。通信时采用软件握手传送协议,安全性高,稳定可靠,整个系统具有错误时的自动保护,自动纠错和自动启动功能。

附录 A Windows 3.1 的通讯函数

BuildCommDCB()	根据控制码加载设备控制块(DCB)
ClearCommBreak()	清除通讯设备的通讯中断状态
CloseComm()	关闭通讯设备
EscapeCommFunction()	表明此设备完成一扩展函数
FlushComm()	刷新通讯设备中的字符
GetCommError()	获得通讯设备状态
GetCommEventMask()	获得并清除事件屏蔽
GetCommState()	把设备控制块装入缓冲区
OpenComm()	打开通讯设备
ReadComm()	从通讯设备读
SetCommBreak()	设置通讯设备的中断状态
SetCommEventMask()	获得并设置事件屏蔽码
SetCommState()	将通讯设备的状态设置成设备控制块的状态
TransmitCommChar()	将一字符放在发送队列的顶部
UngetCommChar()	说明下一个要读的字符
WriteComm()	从缓冲区写到通讯设备