

Windows 用户界面与 VSAT 网管软件之间的信息交换

张慧萍 吴波洋

Windows的SDK软件包提供了编程效率很高的软件开发工具,便于软件人员自行开发带菜单操作和窗口显示的应用程序。Windows环境为应用程序提供的只是用户界面,因此,用该环境开发的实时控制软件,应该确保在用户界面与控制程序之间能高效、可靠地交换信息。本文以VSAT网络管理软件为例,介绍了在Windows环境中开发实时控制软件的一些技术方法。

1 前言

Windows环境的软件开发工具(SDK)提供了丰富的Windows函数,供用户开发多种形式的操作菜单和显示窗口。菜单和窗口能以并列或重叠方式显示,并可用鼠标器或键盘操作和修改。C语言是Windows推荐的程序设计语言,用C语言编制应用程序,可以很方便地使用Windows函数,并调用汇编语言子程序。因此,Windows环境下的实时控制软件,最好使用C语言编制。这样,菜单操作和图表显示处理可以使用Windows函数功能,需要高效率处理以及直接与内存或外设打交道的部分又能用汇编语言编程。

以语音为主的VSAT试验系统是一种使用小口径天线、采用单路单载波方式、按需分配卫星信道传送话音和数据的卫星通信网。该系统由中央站主计算机的网管软件控制和管理网内所有的地面设备和卫星资源。网内各个小站通过主站的预处理器,向网管软件申请出、入网登录和通话接续处理。网管软件也通过预处理器,控制和监测各小站的工作状态、管理它们的通话和传送数据过程。主机通过RS-232C接口与预处理器交

换信息。网管软件提供下拉式菜单,用户可以通过菜单操作,对网络资源进行初始化配置、或者对网内各小站作控制、监测和接续处理。菜单被选中后将调出相应的对话框,供用户输入或者选择配置和处理信息。被用户的误操作触发的出错告警功能将向用户提示正确的操作对象或操作方式。网管软件在对收自人机接口的命令和收自预处理器的请求和应答信号作出相应的处理后,将改写多个数据表格。用户可以通过菜单操作选择某个表格,供显示或者打印输出。在一些特定的条件下,这些表格还将被存盘备用。

对于需要在Windows环境下开发菜单操作程序的软件设计人员,Microsoft公司在为他们提供的软件开发工具箱中,给出大量的程序样本供软件人员学习、分析和利用。本文将略过Windows用户界面的设计方法和网管软件的控制流程,着重介绍在Windows环境中开发VSAT网管软件时,为了高效可靠地在用户界面与控制程序之间交换信息所采用的一些技术方法。

Communications between Windows User Interface
and VSAT Network Management Software

by Zhang Huiping and Wu Boyang

2 模块间的信息传递

在用 C 语言编制的控制管理模块与用 C 语言和 Windows 函数编制的菜单操作和表格处理模块之间, 以及控制管理模块与用 C 语言和汇编语言编制的预处理器接口模块之间, 需要交换命令和数据信息。设计软件时, 应该为模块间的信息传递选择高效可靠的信息编码方式、被传送信息的数据缓冲方式及其读写方式。

2.1 信息编码方式

在控制管理模块与预处理器接口模块之间交换的信息, 采用 16 进制数据形式。每组信息都由 1 字节的命令码和 2~12 字节的后续数据所组成。不同的命令码各自对应固定长度的后续数据。由于后续数据都采用 BCD 码, 为此, 将命令码选在小于或等于 0FEH 的 16 进制数连续区间, 以便区分命令和后续数据。

由菜单操作模块送往控制管理模块的信息, 采用 1 字节命令码后续一个 ASCII 字符串的形式。命令码也选在小于或等于 0FEH 的连续区间, 可以与在菜单操作中输入字母、数字等常用字符相区别。两组信息之间用字符串结束符隔开。

控制管理模块送往菜单操作模块的数据, 写在多个 C 语言字符串结构数组中, 由表格处理或菜单操作模块转换成相应的数据表格, 根据需要显示输出, 或供操作者选择。

2.2 数据缓冲区

以命令码和后续数据形式构成的三种信息, 分别存放在相应的三个数据缓冲区中。每个数据缓冲区都设有头、尾、转折点、读、写等 5 个指针。表头、表尾和转折点固定不变。转折点设在表尾之前, 两者的间距(表尾与转折点之差)大于最长一组信息的长度。在程序初始化时, 读指针、写指针指向表头。每当发送数据的模块写完一组信息, 就将写指针指向该信息尾部的下一字节, 再判断写

指针是否大于转折点, 若大于, 则重新将写指针指向表头, 使缓冲区得以循环使用。接收数据的模块先比较读、写指针, 若不等, 则读取并处理一组信息, 将表头指向该信息的下一字节, 再判断读指针是否大于转折点, 若大于, 则重新将读指针指向表头。设转折点的目的在于避免一组命令被分段存储, 并减少判断读、写指针是否超过表尾, 以及修改读、写指针的次数。

2.3 数据表格与字符串结构数组

由控制管理模块送往菜单操作和表格处理模块的数据有两种形式。其中, 系统资源类和计费统计类表格的极限长度受通信网容量(地球站数、地面话路数和卫星信道数)的限制, 而事件记录类和接续记录类表格为长度不受限制的流水帐形式。但是, 每一种数据所对应的字符串结构数组都是定长的, 也就是说, 流水帐形式的数据表格在内存中的存储空间也是固定不变的。

系统资源类数据表格随时都可作增删修改。相应的字符串结构数组都设有表头、表尾和终点等三个指针。表头和表尾通常是固定不变的, 终点则随该表格已登录数据量的增删而上下浮动。处理该表格数据的模块, 从表头开始, 直到指针结构数组定义的元素数, 均由通信网容量决定。如果用户试图输入超过系统容量的数据, 程序将在显示屏幕上给出告警信号和出错信息, 并拒收输入的数据。删除某一行数据的方法为, 将表中末一行数据改写至将被删除处, 并将终点前移一行。

计费统计类数据只能增添或修改, 不能删除, 其处理方式与系统资源类大致相同。

流水帐形式的数据表格应能妥善处理无限制的表格长度与有限的存储空间这一对矛盾。为此, 除了表头、表尾和终点外, 还设有存盘指针和溢出标志。溢出标志有三种状态, 程序初始化时调为 0, 终点增至与表尾相同(即溢出)时改为 1, 溢出存盘后改为 2。在

程序运行后的最初一段时间内,表格长度未超过存储空间,溢出标志为 0,显示处理模块从表头所指之处开始,处理到终点的前一行。控制管理模块在填写数据后,将修改终点指针,如果它与表尾相等,则将它重新指向表头,并将溢出标志改为 1。溢出存盘和溢出标志为 1 作定时存盘后,溢出标志将被改为 2。首次溢出后,溢出标志不再等于 0。表格显示模块将从终点指针所指行开始,处理到表尾的前一行,再从表头开始,处理到终点指针的前一行。这样,不断增添的数据将被分批存在磁盘中,而在内存中则只存放最近输入的若干行数据,显示屏幕上的数据表格将随新数据行的加入而被滚动刷新。

3 菜单操作

VSAT 网管软件的菜单可分为图表处理、初始化配置和网络管理等三部分。图表处理菜单供用户选择显示或打印输出某个数据表格,选择显示某个系统图表,或显示系统帮助信息。初始化配置菜单供用户输入一个或一组系统资源数据,以及删除或修改已输入的某个系统资源数据。网络管理菜单供用户选择某个被管理对象,进行监测、控制和接续处理。本节将以两个菜单命令为例,介绍菜单操作方式及其处理内容。

3.1 语音转数据

通过鼠标器或键盘,先后选中“语音与数据”和“语音转数据”两级菜单之后,显示屏即出现一个对话框(Dialog Box)。对话框中有三个列表框(List Box)和两个控制按钮。其中两个列表框列出已入网的所有地面话路,供用户分别选作数据发送站和接收站,另一个列表框列出网内所有的卫星信道号,供用户选作数据信道。被选择的收、发数据话路和信道均应处于空闲状态,否则将调出出错框(Error Box)告警,并且指示正确的选择方式。两个控制按钮分别为“取消”和“确认”,

它们都可用鼠标器或键盘选择,取消后不作处理,直接退出本菜单命令处理过程。确认后,如果已选择的话路和信道中不处于空闲状态的,将再次告警,并拒绝作确认处理;如果均为空闲,则将语音转数据的命令码,发数据和收数据话路的站号与话路号,选用的卫星信道号均写入菜单操作→控制管理数据缓冲区,然后退出本菜单命令处理过程。

3.2 增加地球站

先后选中“配置”、“地球站”和“增加”这三级菜单之后,出现的对话框中有两行对话项(Dialog Item)和三个控制按钮供用户操作。上、下两行对话项供用户键入待增加的地球站站名及其站号。站名不能多于 14 个字符,站号应为 2 位 10 进制数字。三个控制按钮分别为“取消”、“确认”和“继续”,选中“继续”或“确认”将检查站名和站号的有效性。如果输入过长的站名或站号,或者输入站号有错时,网管软件将拒绝作进一步处理。选中“继续”后,有效的已输入站名、站号将随增加地球站的命令码之后,被写入菜单→网管数据缓冲区中,然后开始本菜单命令的新一轮处理过程;“确认”后,同样在缓冲区中写入相应的命令码和已输入的站名和站号,然后退出本菜单命令处理过程;“取消”后,直接退出本菜单命令的处理过程。

3.3 菜单命令处理

控制管理模块在处理菜单命令时,先比较相应数据缓冲区的读、写指针是否相等,不相等时,读入 1 字节数据,并判断它是否为命令码。若是,则转入相应的命令处理程序;否则丢弃该数据,并且在比较读指针与转折点,判断是否需要修改读指针后,退出菜单命令处理程序。

在命令处理程序中,首先要读取命令码后续的信息。读取方法多为按该命令所对应后续信息的类型,逐个根据其长度读出,个别的(例如读取长度不确定的站名)则一直读到字符串结束符。读取信息后应及时修改读指

针,并且判断是否已超过转折点,若是,则令其重置于表头处。

4 数据表格处理

由控制管理模块填写的数据表格,以字符串结构数组的形式存储在内存中。数据表格的每一行都对应一个结构,行内的每一项都对应一个字符串。需要改动表格中的某一项数据时,可以用指针直接找到数组中某个结构的相应字符串,从而能提高控制管理模块的处理速度。但是,字符串形式必须经过转换,才能作为数据表格显示或者输出。

4.1 将字符串转换成数据表格

网管软件需要处理许多数据表格。每个表格随时都可能被刷新,而且随时都可能被调用,供显示、存盘,或打印输出。为了提高软件效率,刷新后若未被调用,则不作转换。一旦被调用并作转换,则将以转换后的表格形式另行存放,以备再次调用。因此,每个表格都设有一个状态标志。该标志将在初始化和转换表格后清0。字符串结构数组被修改后,相应的标志加1,使标志不等于0,表示相应的数组已被修改。表格被调用时,先检查相应的标志。标志为0时,可以直接使用数据表格;标志非0时,必须先作转换处理。在网管软件初始化时,已为每个数据表格预留足够的存储空间,并且写入相应的表头。转换时,将每个结构所包含的各个字符串,按固定的间距拼成表格中的一行。转换后的数据表格,可以直接显示、存盘,或打印输出,行数则由2.3节所述的字符串结构数组中的有效区间而定。

除了上述供表格处理模块输出的数据表格外,另有一些如3.1节所述,列表供用户选择的数据表格。它们也都以字符串结构数组形式填写,由菜单操作模块直接转换,并且显示在列表框中。

4.2 数据存盘

网管软件中有三种存盘方式:(1)用于避免因意外停机后丢失数据的定时存盘;(2)当流水帐形式的数据表格写满缓冲区,使写指针重新指向表头后的溢出存盘;(3)用户要求退出网管软件时的退出存盘。每个系统资源类和计费统计类数据表格都对应有正本和副本文件各一个,每个流水帐形式的数据表格则只对应有一个正本文件。当正本文件退出存盘后,均写入当时的日期和时间。每一次调用网管软件时,均不抹除正本文件中原有的记录,新的工作数据将续写在原有记录之后。副本文件只在定时存盘时被刷新。

对于流水帐形式的数据表格,当溢出标志为1时,即作溢出存盘处理。一般从存盘指针处逐行处理到终点指针所指的前一行。溢出存盘后,应将溢出标志改置为2。接续记录类数据表格中,接续建立后未拆线的话路,只填到接续时刻这一项,其后的拆线时刻和接续时间两项则是空白。为此,在定时存盘和溢出存盘过程中,每行都得判断是否已填满,若未填满,则该行及其后的记录都留待下次存盘时再作处理。

5 网管软件处理流程

在DOS环境中运行的工作程序,可以独占计算机的I/O口、存储器、显示器等系统资源。Windows则是一种多用户工作环境,系统资源不由用户支配,运行中的用户窗口程序只有在将信息打入消息队列后才被调用。习惯于在DOS系统下编程的软件人员,在用Windows设计软件时,应该注意到两者之间的不同,确保用户程序在Windows的控制下,能与其它正在运行的应用程序共享系统资源。

Windows用户程序由主函数WinMain和一系列窗口函数构成。主函数在对用户窗口作初始化处理后,进入消息循环,根据鼠标器、键盘、定时器等发来的消息,作相应处理,

并在检索到退出消息时终止用户程序。窗口函数属于“回调函数”，它不能被用户程序直接调用，而只能等待消息队列中出现相关的信息后，被 Windows 系统调用。

VSAT 网管软件的处理流程如下：

```
Main:
  Init Win Proc
  Init Coml Proc
  while (GetMsg)
  {
    switch (Msg)
    {
      case TIMER:
        Coml I / O Proc
        if (Save Timer)
          Timer Save Proc
        break
      case PAINT:
        Paint Proc
        break
      case COMMAND:
        Menu Command Proc
        break
      case DESTROY:
        Destroy Proc
        break
      default:
        return Def Win Proc
    }
    TranslateMsg
    DispatchMsg
    if (Menu Msg)
      Menu Msg Proc
    if (Coml Msg)
      Coml Msg Proc
    Check Timer Proc
    if (OVER)
      Over Save Proc
    if (TableFlag)
      InvalidateRect
  }
  Exit Save Proc
  Exit Coml Proc
  End
```

上述流程由网管软件的主函数及其主窗口函数 MainWndProc 所组成。主窗口函数为 Switch (Msg) { ... } 所包含的程序段，它根据消息队列中的不同信息，控制应用程序转入相应的处理过程。其中，TIMER 作定时

处理，PAINT 根据要求显示数据表格、系统图表或帮助信息，COMMAND 伴随用户对菜单的操作而作出相应的处理，DESTROY 用于退出用户程序前删除用户窗口。如果所收到的信息与网管软件无关，则将它返回 Windows 系统，供其它应用程序处理。主函数除了初始化处理和退出处理之外，都工作于 While (GetMsg) 循环中。该循环是有条件的，仅仅在消息队列中出现新信息时，Windows 系统才使网管软件进入循环处理过程。实时控制程序必须引入定时处理，不断产生定时器消息，以使程序在用户不使用鼠标器或键盘操作时，也能正常运行。

流程中，菜单命令读处理(Menu Msg Proc)、串行口信息读处理(Coml Msg Proc)和监测定时处理(Over Save Proc)三部分属于网络控制管理模块，有关 Coml 的部分属于预处理接口模块，其余部分均属于 Windows 状态下的菜单操作和表格处理模块。从上述流程描述可见，模块间采用数据缓冲区方式交换数据，可以避免不同模块之间的互相调用，从而保证模块间的相对独立性。

6 结束语

网管软件成功地控制了 VSAT 试验系统的运行。它的开发过程表明，在确保 Windows 用户界面与控制软件之间能够可靠地交换信息的前提下，用 Windows 环境开发实时控制软件是可行的。尤其是利用 Windows 函数编制菜单操作、数据表格滚动显示等功能的程序，编程效率高、程序简明易读、运行效果好，很值得推广。

参考文献

- (1) G.E.Lewis, Windows 3.0 Programming Primer, Addison-Wesley Publishing Company, 1992.
- (2) A.Macro & J.Buxton, The Craft of Software Engineering, Addison-Wesley Publishing Company, 1987.