

仿真终端方式下数据的传输技术

姚家亮 陈致明 陈立云 王红霞 (军械工程学院)

摘要:在一些分布处理环境下,微机常作为小型机的一个仿真终端来使用,此时常常需要在微机与小型机(多任务主机)之间传输数据。本文给出了在 VAX/ DECnet 上解决此问题的一种有效方法。

一、问题的提出

在基于多用户计算机系统(如 VAX/VMS 系统, UNIX 等)开发应用系统时,通常会遇到要在微机与主机之间相互交换数据的问题。一般,系统中均提供了若干独立的实用程序以解决此问题,如文件传输,终端仿真等典型的应用软件,但如果用户希望在一个应用系统中将这些实用程序完美地结合起来却十分困难。

通过微机使用多用户系统,利用计算机网络与终端仿真软件是一个最为简单与直接的途径。它能够十分方

便地在多任务系统上开发应用软件。但这种方式在如何利用微机资源,在微机与主机之间交换信息却十分不便,因为随系统提供的软件一般不提供这个功能。因此,有必要研究仿真终端方式下数据的传输技术。

二、仿真终端与数据传输技术

1. 终端仿真原理

终端是指通过通讯线路与计算机主机相连的一种能与主机进行交互通信的设备。其工作过程是:

(1)主机发送到终端的数据流由控制序列组成,对于

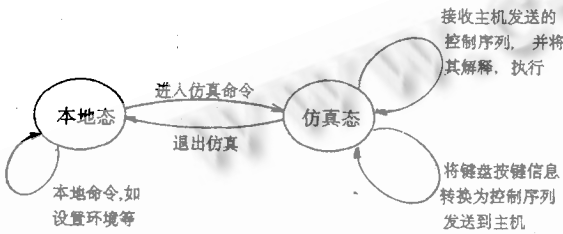
每个控制序列,终端将其解释为一组动作,如将字符显示在屏幕上和控制光标位置等。

(2)将终端键盘上的按键信息按一特定的数据序列发送给主机。

这些控制序列可用一有限状态自动机来描述。

终端仿真是指用一台计算机模仿计算机终端的功能,使之能够用来与主机进行交互式的通讯。使用的环境一般有两类:一类是直接使用异步通信接口和终端通信协议;另一类是利用计算机网络(也可能是通过异步接口)和在网络上运行的远程终端协议,如 DECnet 的 CTERM 协议和 LAT 协议等。

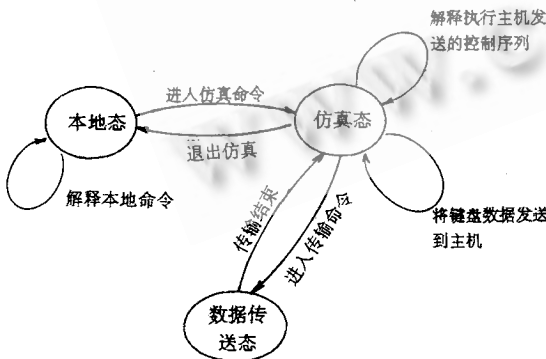
终端仿真软件的工作原理如下:



事实上,上面的模型是无法实现额外的数据通信功能的。要实现数据传输,必须对上面的模型进行适当的扩充。

2.功能扩充方案

为使仿真终端软件能够实现数据的传输,可在模型中增加一个状态,即数据传输状态。



为使软件能够从仿真态进入数据传输态,可使用原终端控制协议中不存在的一控制序列来实现(对于一标

准终端,通常忽略非法的控制序列)。

3.数据传输通道的选择

有两种传输通道可供选择:

(1)终端通信通道。这条通道总是一条可供选择的通道。使用此通道的方法是,从主机发出进入传输状态的命令开始,终端线路上传输的数据不再使用终端控制协议,而使用自定义的数据传输协议,直至数据传输结束。

这种方法在终端通信线路上和在建立新的逻辑链路不方便的环境下是唯一可供选择的方案。但在网络及能够建立新链路的环境下,常常显得效率低,数据传输速度慢且通讯量大等(如在 Ethernet 下,每个数据包很短,使得额外开销增大)。

(2)使用新的数据通道。这种方法只有在计算机网络环境下才是可行的。在数据传输前,首先建立一条专用逻辑链路,数据经过此链路进行传输,传输结束后关闭此链路。

使用此方法后可有效地提高数据传输效率,特别是在高速局域网环境下,例如使用 DECnet / Ethernet,用第一种方法,数据传输率在 1KB-2KB / S 左右,而用第二种方法,可得到 40KB-80KB / S 的传输率。

三、在 DECnet 网上数据传输过程的实现

DECnet 为网络上各节点提供了对等的功能,因此,可通过多种途径实现数据传输过程。经过对 DECnet 的认真分析,发现利用 DECnet 的任务——任务通讯手段来实现,其效果最为理想。

为了在仿真终端的基础上实现对用户透明的数据传输,必须解决如下几个问题:任务——任务通讯;在建立逻辑链路过程中的访问权限的处理;真实终端与仿真终端的识别。

DECnet 任务——任务通讯分为透明与非透明通讯。在 VAX 机上使用透明任务——任务通讯即可完成此任务,其过程与对文件的操作类似,如下所示:

- 使用 OPEN 命令请求建立一逻辑链路。
- 使用 READ / WRITE 进行 I / O 操作
- 使用 CLOSE 命令关闭此链路

在微机上,为了方便编程,可使用 DECnet-DOS 提供的过程库,按照约定的次序来使用,即可实现与其它节

点的任务——任务通讯(非透明过程)。

在实现一应用系统时,权限问题是一个必须十分注意的问题。在 VAX 机上,对于一个人境连接请求,必须给出用户名与口令说明,否则系统就按一省缺规则进行处理。而让仿真终端软件可靠地掌握用户名和口令不易实现。

另外,在 VAX 机上允许入境连接请求时,也就意味着允许远程节点可以使用任何其它软件(实用程序)访问本地的数据,这会给系统权限的管理带来一些麻烦,降低数据的安全性。

由于以上原因,较为可靠与方便的手段是在 VAX 主机上使用出境连接,而作为仿真终端的微机上使用入境连接,由于微机上运行的 DOS 系统为单用户系统,一般可以不必设置权限,而不会降低安全性,这样访问权限的问题便不存在了。

为了便于应用软件的设计,使应用软件既可以运行于真实终端上,又可运行于仿真终端上,必须能够识别这两类终端,使得在进行数据传输时,不对真实终端进行操作。对终端的识别通常是使用一询问一回答序列,以确定终端的类别。对于各种类型终端,通常有自己的标识码。操作系统中通常设置一终端驱动层,用于完成对各种终端的驱动,从而可使应用程序与设备无关。

在写仿真软件时,针对所仿终端的类型,对于终端查询序列,仿真软件也必须回答所仿终端的特征码,以保证应用软件的兼容。那么如何区别真实终端与仿真终端呢?通常,真实终端在接收到一非预定义过的转义序列时,只是将其“吃掉”,而无任务动作,利用这一点便可以将它们区分开来。

在前面提到对仿真终端软件的功能进行扩充时,使用了一转义序列,真实终端对此序列不会有任何响应,而仿真软件则将状态转到数据传输状态。设此序列为 A,终端查询序列为 Q。

我们定义仿真软件的行为是:

对于询问序列 Q,在终端状态下,回答所仿终端的特征序列 T1,而在数据传输状态下回答新的特征序列 T2。

由于真实终端无数据传输状态,因此对于询问序列 Q,它总是回答 T1。

这样便可以通过在发送了状态转换序列 A 后,发送

终端类型询问序列 Q,来区分真实终端与仿真终端。

整个数据传输过程可描述如下:

主要部分:

发送 AQ 询问序列:

读响应序列 ANS;

S = 0;

if(在规定时间内无回答序列)

非正常终端设备;

else if (ANS == T2)

是仿真终端软件,S = 1;

else

不是仿真终端软件,不能进行数据传输;

if(S == 1){

发送数据传输控制信息(如数据传输方向及文件名等);

发连接请求,建立逻辑链路;

将数据写到逻辑链路上、从逻辑路上读数据;

关闭逻辑链路;

}

至此数据传输过程结束。

仿真软件部分:

接到 A 序列,转到数据传输状态;

接到 Q 序列,发特征序列 T2;

(在终端仿真态时,接到 Q 序列,发送特征序列 T1)

接收数据传输控制信息;

接收建立逻辑链路请求,建立逻辑链路;

从逻辑链路上读数据、写数据到逻辑链路上;

关闭逻辑链路;

返回终端仿真状态;

至此数据传输过程结束。

四、结束语

在终端仿真软件中引入此技术后可大大增强仿真终端的功能,便于应用系统的开发与集成。此外,在此基础上可方便地扩充其它功能,如图形处理功能等。因此,该技术具有一定的应用价值。