

对等通信资源互享系统的设计与实现

屈景辉 许卫中 唐靖飏 (第四军医大学 710032)

摘要:本文简述在 NOVELL 环境下,借助现有网络硬件连接,利用 IPX/SPX 通信协议,实现对等通信中资源互享系统的设计和具体实现方法。

关键词:对等通信 资源 互享 IPX/SPX 设计 工作站

计算机局部网络的主要用途是使网络上的用户共享高质量的打印机、大容量硬盘;允许网络上的用户进行信息的快速交换;网络上的用户通过工作站使用网络系统的全部软件、硬件资源,这就是局域网络系统中的资源共享。网络上的每个工作站都是智能工作站,用户程序的执行,数据的处理都是在本地工作站上进行的。也就是说网络系统的构成为网上用户提供了高质量打印机、大容量硬盘供打印和存放公用软件、资料以及 CD-ROM 的访问,即仅能共享网上资源,而工作站资源只能独占不能互享。

然而,值得注意的是网络上各个工作站的性能往往是不一样的,如 CD-ROM 安装与否,内存配置大小, CPU 档次及速度快慢等。那么,低档工作站能否享用高档工作站的硬件环境,未安装 CD-ROM 的能否享用安装 CD-ROM 的,本地工作站无某种软件能否不通过文件拷贝或传输而以远程方式使用这种软件呢?回答是肯定的。

为使工作站的软件、硬件资源做到物尽其用,最大限度发挥计算机系统的软、硬件资源利用率。我们在现有网络连接情况下,利用 IPX/SPX 通信协议和 NetWare C 接口函数库,设计开发了对等通信资源互享系统。

1. 系统开发环境

· 硬件环境: NOVELL 网络硬件系统, 386/486/586 工作站

· 软件环境: Microsoft C++ 6.0/7.0 for DOS, NetWare C 接口库函数, DOS ODI 工作站软件

2. 系统结构

本系统由一个主控模块和三个子功能模块组成,生成可执行文件 PPTALK.EXE,其结构如图 1 所示:

图 1 中“系统菜单”部分主要完成系统逻辑连接的建立、菜单显示以及击键方选择功能号的发送,使通信双方同时进入相应功能模块,该模块是系统主控模块,其结构

如图 2 所示。从图 2 看出:

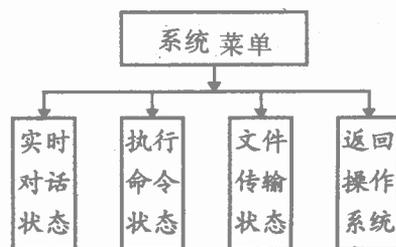


图 1 系统结构图

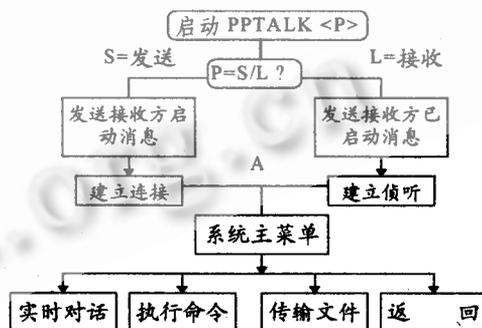


图 2 主控模块结构图

(1)发送与接收两端运行同一程序,命令参数 P 确定主从关系。当 S 方接收到 L 方已经启动消息后建立逻辑连接。

(2)L 方若未侦听到连接的数据报时一直在循环等待,S 方在连接失败时自动返回到 DOS 状态。

(3)当连接成功程序运行到 A 点时,双方的屏幕显示是一致的,此时二者已失去主从关系,具体执行哪项功能由先击键一方确定。

(4)当击键一方选择后并发送功能号数据报通知对方,双方程序自动进入相应功能模块,避免事先约定不便或选择不一程序各行其道的弊病。

(5)进入某功能模块后,双方具有同等权利,互不影响,任一方都可以执行本功能指定的操作,仅当一方选择退出功能后另一方也只有顺其而行,达到双方同步运行的目的。

3. 系统功能

本系统具有实时对话、执行命令、对方屏幕访问和文件传输四大功能:

(1)通信双方实时对话。所谓实时对话就是在网络硬件连接一定的情况下,运用 IPX/SPX 协议建立两工作站间的逻辑连接,提供二者之间的实时对话功能,即甲乙两工作站任一方任一时刻的输入请求立即反映在另一方的显示表面,就象在本地工作站操作一样,达到实时对话的目的。

(2)执行操作系统命令或运行应用程序。这是对等通信中资源互享的关键,是系统设计的主要目标,也是系统设计中的难点。具体方法是通过 C 语言的函数调用实现。要达到工作站甲独占工作站乙的软硬件资源的首要问题是能够在工作站甲下达命令后由工作站乙去执行,如驱动器切换(A:/B:/C:),DOS 命令(Dir/CD)、可执行文件(.bat/.com/.exe)等,如果这个问题能够解决,就为工作站甲独占工作站乙的软硬件资源,执行各种操作打开了方便之门,反之亦然。这样对等通信中的资源互享就成为现实。

(3)访问对方屏幕。虽然乙能够执行甲指定的某种操作,但实际中通信双方往往相隔一定距离,操作结果(屏幕输出)甲是不能直接看到的,如果不设法将执行结果告诉甲,那么即使具备了执行命令的功能也毫无意义。因此,就要提供传送屏幕功能,实现本地工作站对远程工作站屏幕的访问。当甲需要察看乙的屏幕时即下达传送屏幕命令,在乙接收到该命令后立即传送指定显示页内存中的内容,由甲负责接收并在本地工作站指定显示页重显。这样甲就象在使用本地工作站一样。

(4)传输文件。在通信中的某一方若需要将对方的某文件(可执行程序、各种数据文件)复制到本地工作站,可利用文件传输功能将文件传送到本地工作站享用。

4. 功能实现

(1)功能选择同步。当通信双方逻辑连接建立后,屏幕显示系统功能菜单,二者显示内容一致。为使二者能

够同时执行某一功能,不致于因选择不同程序而各行其道,采取“先入为主”的方法使双方同步运行,当一方按键之后马上将键值发往对方,双方程序判断是否为指定功能号,若是则同时进入相应状态,否则一直在循环等待。程序结构简述如下:

```
void mainmenu(char * menu[], WORD spxConnectionID)
{…… //变量初始化
do {…… //屏幕窗口设定,显示菜单
str[0] = 0;
do { if(str[0]) { //根据键入字符选定执行功能
switch(str[0]) {
case 0x31: RealTimeTalk(spxConnectionID);
done1 = 1; break;
case 0x32: ExecCommand(spxConnectionID);
done1 = 1; break;
……
case ESCAPE-KEY: ……
}
str[0] = 0; }
if(kbhit()) { //接收击键字符
while(sendECB -> inUseFlag);
str[0] = (char) getch();
if(str[0] == ESCAPE-KEY) {
…… // 循环变量设置
} // 发送选择功能号数据报
SPXSendSequencedPacket(spxConnectionID, sendECB);
} } while(! done1);
while (sendECB -> inUseFlag);
} while(! done2);
}
```

(2)实时对话。在实时对话处理中,一方面要设置接收、发送、状态三个窗口,使接收和发送的字符显示在相应窗口位置。另一方面要提供全屏编辑功能,对约定功能字符和控制字符进行必要的处理,使发送方的操作完全反映在接收方,使对话的实时性得到具体体现。

当然,在对话模块设计时也可以字符串形式进行发送(接收),这样能给对方一个完整的概念,但失去的却是对话的实时性,因此以传送字符形式更有意义。

```
void RealTimeTalk(WORD spxConnectionID)
{…… // 变量初始化,设置接收、发送与状态窗口
done = 0;
while (! done) {
if(str[0]) { ……
```

```

switch(str[0]) { // 接收字符处理、显示
    case RETURN-KEY:
        -outtext(" \n"); break;
    case ESCAPE-KEY:
        done = 1; break;
    case BACKSPACE:
        ..... // 坐标定位等 break;
    default:
        ..... // 有关处理
}
..... ;
str[0]=0; }
if(kbhit()) { .....//键入字符处理、显示和发送字符过程
while(sendECB ->inUseFlag) ;
str[0] = (char) in-key();
switch(str[0]) {
    case ESCAPE-KEY: .....
    case RETURN-KEY: .....
    case BACKSPACE: .....
    default: .....// case 语句处理同上
}
SPXSendSequencedPacket(spxConnectionID, sendECB);
while(sendECB ->inUseFlag) IPXRelinquishControl();
.....
str[0]=0;
} }
while(sendECB ->inUseFlag);
return ;
}

```

程序运行后双方可同时击键,相应字符一方面显示在本地工作站的发送窗口,另一方面经传送显示在远程工作站的接收窗口,象面对面以文本方式对话一样。

(3)执行命令。在实时对话功能实现以后,要实现执行DOS命令功能就很容易。具体处理过程是:

①在执行命令状态仍以实时形式传送和接收命令字符,包括对输入错误及特殊字符的处理。

②将接收到的有效字符放入临时字符串变量(CommandStr[n])中,当接收到对方发来的回车符时,由程序给字符串变量加入串结束符('\0')。

③用C语言函数执行字符串指定操作,即用system(CommandStr)语句实现。

程序结构简述如下:

```

..... //变量初始化及屏幕窗口设置
while (! done) {
if(str[0]) { //执行命令方
switch(str[0]) {
    case RETURN-KEY: // 接收到回车符执行命令
        -outtext(" \n");
        CommandStr[ii] = 0;
        if(strlen(CommandStr) == 0) break;
        SetPage(2); // 程序执行结果显示在第3页
        system(CommandStr);
        CommandStr[0] = 0; ii = 0;
        SetPage(0);
        break;
        ..... // 处理特殊字符和功能字符
    default:
        CommandStr[ii] = str[0]; ii = ii + 1; break;
} str[0] = 0; }
if(kbhit()) { //发送命令方
.....
while(sendECB ->inUseFlag) ;
str[0] = (char) in-key();
switch(str[0]) {
    case ESCAPE-KEY:
        done = 1; break;
    case RETURN-KEY:
        -outtext(" \n"); break;
        .....
    default:
        CommandStr[0] = str[0]; break;
}
SPXSendSequencedPacket(spxConnectionID, sendECB);
while(sendECB ->inUseFlag) IPXRelinquishControl();
str[0]=0; } // if( kbhit() )
} // while(! done)
while(sendECB ->inUseFlag); return ; } }

```

(4)远程屏幕访问与文件传输。屏幕与文件传送的过程基本相同。为防止双方处理过程中的时间差,在数据报(地址str)的前5个字节设置了传输控制标志:

str[0]	0/0xfe	发送开始与结束标志
str[1],str[2]		传送块顺序号
str[3],str[4]		传送块长度

这几个标志的作用是:

·发送开始与结束标志是双方通信联络信号,由接收方设置。

·块顺序号由接收方发出,发送方根据顺序号确定发送数据的起始地址。

·块长度由发送方设置,接收方根据此值确定处理数据报字符的个数。

程序流程与实时对话部分类似,不再赘述。

5. 特点与说明

上述资源互享方法不仅实时性强且具有较高的实用性。通信双方除实时对话过程必须参与外,当一方需要使用另一方的软硬件资源时,另一方不必介入。

文中有关通信函数由 NetWare C 接口函数库提供,

程序在 Microsoft C++ 6.0/7.0 下编译连接,工作站为 386 以上机型。限于篇幅,文中所附程序片段在原程序基础上作了压缩,请谅解。

参考文献

- [1] 庄德秀等著. NOVELL Netware 386 网络操作系统,吉林科学技术出版社,1991
- [2] 羊羽编写. 用 C 语言开发局域网络程序的技术、技巧和实例,学苑出版社,1993
- [3] 李继敏等编. NOVELL NetWare 386 深开发环境技术,学苑出版社,1993

(来稿时间:1996年10月)