

基于 IPX / SPX NOVELL 网点通信技术

高鹏翔 (山东纺织工学院)

摘要: 为了在使用 NOVELL 网不局限于共享打印机和文件共享,必需用到新的系统设计技术和新的程序设计技巧,提高计算机的应用水平,本文结合实际讨论了 NOVELL 网点对点通信技术。

一、引言

NOVELL 网是美国 NOVELL 公司开发一种高性能局域网系统, NETWARE 是局域网操作系统,也是 NOVELL 网的核心技术。目前, NETWARE 已成为世界上流行的局域网操作系统中的佼佼者,并逐渐成为局域网操作系统的标准。NOVELL 网自推出之日起,就以其独特的设计思想、优越的性能与灵活的应用吸引了众多的用户,成为 PC 机连网的主导型 LAN,尽管 NOVELL 网提供了各种服务,在一定程度上,满足了用户连网的目的。然而,为了在使用 PC 网络不局限于共享打印机和文件共享,必须要用到新的系统设计技术和新的程序设计技巧,实现连网工作站之间直接交换数据,而不过文件服务器,这就要求用户自己进行开发设计,同时不影响网络的正常工作。这些技术和方法的应用将提高网络的应用范围和应用水平。为此,必须进一步了解网络内部的工作原理,以便在开发应用程序时调用网络功能、灵活的应用网络通信。在网络通信最低一级,连网的 PC 机和其它 PC 机用信息报文(即帧)进行通信,所有网的最基本的活动就是这些帧被网络服务器和它的支撑软件发送和接收,实现数据共享、设备共享。

二、IPX / SPX 协议

IPX / SPX 是 NOVELL 网的基本协议,要二次开发 NOVELL 应用软件必须了解 IPX / SPX 协议原理。网间信息交换(IPX)与 OSI 模型的网络比较,实现了编址路由选择,交换发送信息到目标址。顺序报文交换(SPX)为一会话级的,基于连接的协议,在 SPX 报文发送和接收时,必须在发送信息的双方建立一个连接,连接建立以后,信息可以沿每个方向传送并保证一定收到,

SPX 同时也保证报文将以正确的次序到达。

1. IPX / SPX 安装测试功能

IPX / SPX 功能在调用之前必须先测试,存在才能使用。其 C 语言程序如下:

```
void * ipx_spx(void);
int IPX__SPXInstalled(void)
{
    union REGS regs;
    struct SREGS sregs;
    regs.x.ax = 0x7a00;
    int86x(0x2f,&regs,&regs,&sregs);
    if(regs.h.a1 != 0xff) return -1;
    ipx_spx = MK_FP(sregs.es,regs.x.di);
    __BX = 0x0010;
    __AL = 0x00;
    ipx_spx();
    if(__AL == 0x00) return 0;
    return 1;
}
```

说明:

(1)当返回(-1)时,IPX 和 SPX 都不存在,不能执行 IPX / SPX 的功能调用。

(2)当返回(0)时,IPX 存在,SPX 不存在,只能执行 IPX 的功能调用。

(3)当返回(1)时,IPX / SPX 都存在,执行 IPX / SPX 的全部功能调用。

2. IPX / SPX 部分功能介绍

通过设置 CPU 寄存器,以确定 IPX / SPX 的功能,然后使用在安装测试中获得的远指针 ipx_spx()函数执行其功能。

int IPXGetLocalTarget(void) 返回目标节点的路径信息

int IPX__SPXInstalled(void) 测试 IPX / SPX

int IPXOpenSocket(WORD socketnumber) 打开一个套接口
 int IPXCloseSocket(WORD socketnumber) 关闭一个套接口
 void IPXRelinquishControl(void) 给 IPX 处理事件的 CPU 时间
 int GetInternetAddress(BYTE connectionnumber) 返回站点网络
 号和节点地址
 int GetConnectionInfo(CHAR * who) 返回给定用户入网连接
 号码
 void IPXSendPacket(ECB * eventCB) 执行发送信息包功能
 void IPXListenForPacket(ECB * eventCB) 执行接收信息包功能

三、点对点通信技术

点对点通信程序实现入网的工作站之间直接进行数据交换,提高了数据通信的实时性。

在发送工作站打开了一个套接口,获得了网络号码、节点地址和接收工作站的套接口号码后,并放入报文头和事件控制块 ECB 中,这时就可以执行 IPXSendPacket()函数发送信息记录了,要发送的数据存放在数组 databuff[80];接收工作站打开一个套接口并初始化事件控制块 ECB,这时就可以执行 IPXListenForPacket()函数接收信息记录,接收的数据存放在数组 databuff[80]中;IPX 在返回到应用程序之前,不需等待报文被发送或接收完,协议仅仅作初始化操作。可以有两种方式处理这些操作:应用程序进入一个循环以检查接收与发送操作是否已完成,即判 ECB 中 inuseflag 和 completcode 值;或者应用程序提供给 IPX 这些操作完成后 IPX 要进行的例程的地址,即 ECB 中 ESR 的地址。

```

/* 点对点通信程序清单 */
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <dos.h>
#include <bios.h>
#include <stdlib.h>
#include "ipx_spx.c"
#define OK 0
typedef unsigned char BYTE;
typedef unsigned int WORD;
typedef unsigned long LONG;
BYTE databuffer[80];
BYTE netaddress[4] = {0,0,0,0};
BYTE nodeaddress[6] = {0xff,0xff,0xff,0xff,0xff,0xff};
typedef struct IPXaddress
    
```

```

{
    BYTE network[4];
    BYTE node[6];
    BYTE socket[2];
}IPXaddress;
typedef struct IPXheader
{
    WORD checksum;
    WORD length;
    BYTE transportControl;
    BYTE packetType;
    BYTE destnet[4];
    BYTE destnode[6];
    WORD destsocket;
    BYTE sournet[4];
    BYTE sournode[6];
    WORD soursocket;
}IPXheader;
typedef struct ECB
{
    void far * link;
    void far * ESRaddress;
    BYTE inUseFlag;
    BYTE completionCode;
    WORD socketNumber;
    BYTE IPXworkspace[4];
    BYTE driverWorkspace[12];
    BYTE immediateaddress[6];
    WORD packetCount;
    struct {
        void far * address;
        WORD length;
    }packet[2];
}ECB;
ECB initialECB;
IPXheader initialIPXheader;
void far (* ipx_spx) (void);
main(int argc, char * argv[ ])
{
    if((argc > 3) (argc < 2)){
        printf("IF RECEIVE DATA PRESS IPXCOMM / r\n");
        printf("IF SEND DATA PRESS IPXCOMM / s <lanname> \n");
        exit(1);
    }
    if (!strcmp(argv[1], "r")){
        MainListen( );
        exit(1);
    }
}
    
```

```

    }
    if(!strcmp(argv[1]," / s")){
MainSend(argv[2]);
exit(1);
    }
}
MainSend(char * targetname)
{
    int status;
    WORD targetSocket = 0x5555;
    BYTE targetConn;
    status = IPX__SPXInstalled( );
    if(status!= 1) {
printf("IPX not installed");
return ;}
    targetConn = GetConnectionInfo(targetname);
    if(! targetConn)
    {
printf("\tTarget User %s do not Logged in. \n",targetname);
return ;
    }
    GetInternetAddress(targetConn);
    printf("\tSend data to %s !\n",targetname);
    databuffer[0]= 0;
    status = IPXOpenSocket(targetSocket);
    SetupSendPacket( );
    IPXSendPacket(&initialECB);
    while(initialECB.inUseFlag!= OK)
IPXRelinquishControl( );
    if(initialECB.completionCode!= OK)
printf("Send error, status = %x", initialECB.completionCode);
return;
    }
}
MainListen( )
{
    WORD targetSocket = 0x5555;
    int status;
    char prob[80];
    printf("\t Receive Data !\n");
    status = IPX__SPXInstalled( );
    if(status!= 1) {
printf("IPX not installed!");
return ;}
    SetupRcvPacket( );
    status = IPXOpenSocket(targetSocket);

```

```

IPXListenForPacket(&initialECB);
while(1)
{
    if(!initialECB.inUseFlag)
if(!initialECB.completionCode)
printf("%s¥ n",databuffer);
    }
}
SetupRcvPacket(void)
{
    memset(&initialECB,0,sizeof(struct ECB));
    memset(&initialIPXheader,0,sizeof(struct IPXheader));
    initialECB.socketNumber = 0x5555;
    initialECB.packetCount = 2;
    initialECB.packet[0].address = &initialIPXheader;
    initialECB.packet[0].length = sizeof(struct IPXheader);
    initialECB.packet[1].address = databuffer;
    initialECB.packet[1].length = sizeof(databuffer);
}
SetupSendPacket(void)
{
    int i;
    memset(&initialECB,0,sizeof(struct ECB));
    initialECB.socketNumber = 0x5555;
    i = IPXGetLocalTarget( );
    if(i!= 0)return;
    initialECB.packetCount = 2;
    initialECB.packet[0].address = &initialIPXheader;
    initialECB.packet[0].length = sizeof(struct IPXheader);
    initialECB.packet[1].address = databuffer;
    initialECB.packet[1].length = sizeof(databuffer);
    initialIPXheader.packetType = 4;
    memcpy(initialIPXheader.destnet,netaddress,4);
    memcpy(initialIPXheader.destnode,nodeaddress,6);
    initialIPXheader.destsocket = 0x5555;
}

```

参考文献:

- [1]羊羽编译 用 C 语言开发局域网程序的技术、技巧和实例 北京电脑公司
- [2]张公忠 王钰编著 NOVELL 网组网原理与实用技术 清华大学出版社
- [3]袁荣编译 NOVELL netware 程序员指南 北京希望电脑公司
- [4]张锋 曾炼 冯大勤译 SFT/ NOVELL netware 286,386 最新使用大全 e 北京希望电脑公司