计算机网络

**课程设计报告**

题目： 基于IP多播的网络会议程序

**姓名： 唐韬**

**学号： 1305010206**

**班级： 13计算机二班**

**指导老师： 姜磊**

**湖南科技大学计算机科学与工程学院**

**2016年 1 月**

1. **实验题目**

基于IP多播的网络会议程序

1. **实验内容**

1、了解IP多播的基本原理

2、了解IP多播的Winsock实现步骤

3、参考附录3的局域网多播程序，设计一个图形界面的网络会议程序（实现文本多播方式即可）。

1. **实验步骤**

1、通过网络或者书籍，了解IP多播的基本原理

2、通过网络或者书籍，了解IP多播的Winsock实现步骤

3、通过课程设计指导书，参照附录3，实现一个图形界面的网络会议程序，实现文本多播即可

1. **代码调试**

// Ip3Dlg.cpp : 实现文件

//

#include "stdafx.h"

#include "Ip3.h"

#include "Ip3Dlg.h"

#include <winsock2.h>

#include<ws2tcpip.h>

#include<stdio.h>

#include<stdlib.h>

#pragma comment(lib, "ws2\_32.lib ")

#define MCASTADDR "233.0.0.101"//本例使用的多播组地址

#define MCASTPORT 8000 //本地端口号

#define BUFSIZE 1024//发送数据缓冲大小

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

CString Str;

SOCKET sock,sockM;

struct sockaddr\_in local,remote,from;

// 用于应用程序“关于”菜单项的CAboutDlg 对话框

UINT ThreadFun(LPVOID pParam){ //线程要调用的函数

WSADATA wsd;

int len=sizeof(struct sockaddr\_in);

int ret;

TCHAR recvbuf[BUFSIZE];

if(WSAStartup(MAKEWORD(2,2),&wsd)!=0)

{

printf("WSAStartup()failed\n");

return -1;

}

if((sock=WSASocket(AF\_INET,SOCK\_DGRAM,0,NULL,0,

WSA\_FLAG\_MULTIPOINT\_C\_LEAF|WSA\_FLAG\_MULTIPOINT\_D\_LEAF|

WSA\_FLAG\_OVERLAPPED))==INVALID\_SOCKET)

{

printf("socket failed with :%d\n",WSAGetLastError());

WSACleanup();

return -1;

}

//将sock绑定到本机端口上

local.sin\_family=AF\_INET;

local.sin\_port=htons(MCASTPORT);

local.sin\_addr.s\_addr=INADDR\_ANY;

if(bind(sock,(struct sockaddr\*)&local,sizeof(local))==SOCKET\_ERROR)

{

printf("bind failed with %d\n",WSAGetLastError());

closesocket(sock);

WSACleanup();

return -1;

}

//加入多播组

remote.sin\_family=AF\_INET;

remote.sin\_port=htons(MCASTPORT);

remote.sin\_addr.s\_addr=inet\_addr(MCASTADDR);

if((sockM=WSAJoinLeaf(sock,(SOCKADDR\*)&remote,sizeof(remote),

NULL,NULL,NULL,NULL,

JL\_BOTH))==INVALID\_SOCKET)

{

printf("WSAJoinLeaf()failed:%d\n",WSAGetLastError());

closesocket(sock);

WSACleanup();

return -1;

}

//接收多播数据，当收到的数据为QUIT时退出

while(1)

{

if((ret=recvfrom(sock,recvbuf,BUFSIZE,0,

(struct sockaddr\*)&from,&len))==SOCKET\_ERROR)

{

//printf("recvfrom failed with :%d\n",WSAGetLastError());

closesocket(sockM);

closesocket(sock);

WSACleanup();

return -1;

}

if(strcmp(recvbuf,"QUIT")==0) break;

else{

recvbuf[ret]='\0';

Str+=inet\_ntoa(from.sin\_addr);

Str+=":";

Str+=recvbuf;

Str+="\r\n";

HWND hWnd = GetDlgItem(AfxGetMainWnd()->m\_hWnd,IDC\_EDIT1);

CWnd \*pWnd = CWnd::FromHandle(hWnd);

pWnd->SetWindowText(\_T(Str));

}

}

closesocket(sockM);

closesocket(sock);

WSACleanup();

return 0;

}

class CAboutDlg : public CDialog

{

public:

CAboutDlg();

// 对话框数据

enum { IDD = IDD\_ABOUTBOX };

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持

// 实现

protected:

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)

{

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialog)

END\_MESSAGE\_MAP()

// CIp3Dlg 对话框

CIp3Dlg::CIp3Dlg(CWnd\* pParent /\*=NULL\*/)

: CDialog(CIp3Dlg::IDD, pParent)

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CIp3Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

}

BEGIN\_MESSAGE\_MAP(CIp3Dlg, CDialog)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

//}}AFX\_MSG\_MAP

ON\_EN\_CHANGE(IDC\_EDIT1, &CIp3Dlg::OnEnChangeEdit1)

END\_MESSAGE\_MAP()

// CIp3Dlg 消息处理程序

BOOL CIp3Dlg::OnInitDialog()

{

CDialog::OnInitDialog();

// 将“关于...”菜单项添加到系统菜单中。

// IDM\_ABOUTBOX 必须在系统命令范围内。

ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);

ASSERT(IDM\_ABOUTBOX < 0xF000);

CMenu\* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != NULL)

{

CString strAboutMenu;

strAboutMenu.LoadString(IDS\_ABOUTBOX);

if (!strAboutMenu.IsEmpty())

{

pSysMenu->AppendMenu(MF\_SEPARATOR);

pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);

}

}

// 设置此对话框的图标。当应用程序主窗口不是对话框时，框架将自动

// 执行此操作

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

// TODO: 在此添加额外的初始化代码

::AfxBeginThread(ThreadFun, this);

return TRUE; // 除非将焦点设置到控件，否则返回TRUE

}

void CIp3Dlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == IDM\_ABOUTBOX)

{

CAboutDlg dlgAbout;

dlgAbout.DoModal();

}

else

{

CDialog::OnSysCommand(nID, lParam);

}

}

// 如果向对话框添加最小化按钮，则需要下面的代码

// 来绘制该图标。对于使用文档/视图模型的MFC 应用程序，

// 这将由框架自动完成。

void CIp3Dlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作矩形中居中

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialog::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标显示。

//

HCURSOR CIp3Dlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CIp3Dlg::OnEnChangeEdit1()

{

// TODO: 如果该控件是RICHEDIT 控件，则它将不会

// 发送该通知，除非重写CDialog::OnInitDialog()

// 函数并调用CRichEditCtrl().SetEventMask()，

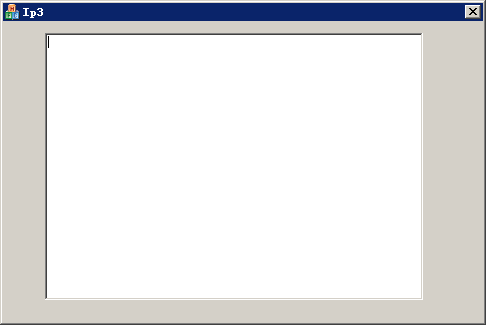
// 同时将ENM\_CHANGE 标志“或”运算到掩码中。

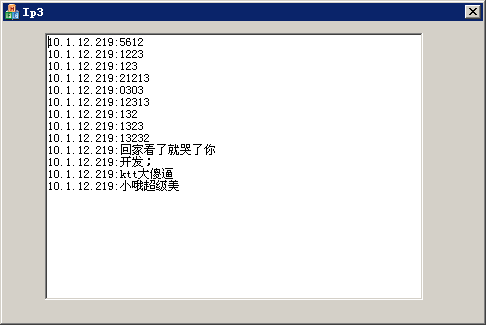
// TODO: 在此添加控件通知处理程序代码

}

1. **实验结果及结果分析**

多播接收方：





多播发送方：



1. **心得体会**

真是受益匪浅啊！万万没想到，MFC编程中的多线程编程竟然和普通的C++编程不一样，我在那里弄了半天，最后通过百度，结合谷歌，用以搜狗的辅助，终于弄明白了这些问题，这一点让我意识到，很多东西在课堂上都是学不来的，正所谓是“纸上得来终觉浅”，实践出真知，要真正的把所学的运用到学习工作中，只能通过多写代码来实现，我认为，我学到的，应该是这一点，非常好玩。