**---------Server------------**

# Function.Cpp

#include"function.h"

#include "string.h"

#include "datagram.h"

Status AddInformation(SOCKET \*server,Status SEND(SOCKET \*server,Contacts cont))//添加联系人

{

int tmp;

char name[20];

int age;

bool sex;

char unit[30];

char tel[15];

char ip[20];

printf("请依次输入姓名,年龄,性别(男:1;女:0),单位,电话(以回车隔开)\n");

scanf("%s", name);

scanf("%d", &age);

scanf("%d", &tmp);

if (tmp == 1)

sex = 1;

else

sex = 0;

scanf("%s", unit);

scanf("%s", tel);

strcpy(ip,"localhost");

Contacts cont;

memcpy(cont.name, &name, sizeof(name));

cont.age = age;

cont.sex = sex;

memcpy(cont.unit, &unit, sizeof(unit));

memcpy(cont.tel, &tel, sizeof(tel));

memcpy(cont.ip, &ip, sizeof(ip));

if (SEND == 0)

{

//#########调试用########

printf("%s\t%d\t%s\t%s\t%s\t%s\n", cont.name, cont.age, cont.sex ? "男" : "女", cont.unit, cont.tel, cont.ip);

//#######################

}

else

SEND(server,cont);

Insert(cont);

printf("#########信息录入完毕########\n");

return OK;

}

Status AllAddress(SOCKET \*client,Status SEND(SOCKET \*client,Contacts))//遍历联系人

{

FILE \*filein = fopen("contacts.dat", "r");

Contacts tmp1;

if (!filein)

{

return NO\_ADDRESSEXIST;

}

while (!feof(filein))

{

int ret = fread(&tmp1, sizeof(Contacts), 1, filein);

if (ret>0)

{

if (SEND == 0)

{

//printf("%d\n", ret);

printf("%s\t%d\t%d\t%s\t%s\t%s\n", tmp1.name, tmp1.age, tmp1.sex, tmp1.unit, tmp1.tel, tmp1.ip);

}

else

SEND(client,tmp1);

}

}

fclose(filein);

return OK;

}

Status Del(char \*name, SOCKET \*client, Status SEND(SOCKET \*client, Contacts))//删除联系人

{

Contacts tmp,tmp1;

int ret;

//备份通讯录

FILE \*filein = fopen("contacts.dat", "rb");

if (!filein)

return NO\_ADDRESSEXIST;

FILE \*fileout = fopen("contacts.dat.bak", "wb");

while (!feof(filein))

{

if (fread(&tmp, sizeof(Contacts), 1, filein) > 0)

fwrite(&tmp, sizeof(Contacts), 1, fileout);

}

fclose(filein);

fclose(fileout);

//######end######

filein = fopen("contacts.dat.bak", "rb");

if (!filein)

return NO\_ADDRESSEXIST;

fileout = fopen("contacts.dat", "wb");

while (!feof(filein))

{

ret = fread(&tmp1, sizeof(Contacts), 1, filein);

if (ret>0)

{

if (SEND == 0 && !strcmp(tmp1.name, name))

{

printf("删除的信息为:%s\t%d\t%d\t%s\t%s\t%s\n", tmp1.name, tmp1.age, tmp1.sex, tmp1.unit, tmp1.tel, tmp1.ip);

continue;

}

else if((SEND != 0 && !strcmp(tmp1.name, name)))

{

Send(client, tmp1);

continue;

}

fwrite(&tmp1, sizeof(Contacts), 1, fileout);

}

}

fclose(filein);

fclose(fileout);

return OK;

}

Status Search(Contacts &cont, SOCKET \*client, Status SEND(SOCKET \*client, Contacts))//后台数据的查找

{

Contacts tmp1;

FILE \*filein = fopen("contacts.dat","rb");

if(!filein)

return NO\_ADDRESSEXIST;

while(!feof(filein))

{

int ret = fread(&tmp1,sizeof(Contacts),1,filein);

if (ret>0)

{

cont = tmp1;

if (SEND == 0 && !strcmp(tmp1.name, cont.name))

{

printf("%s\t%d\t%d\t%s\t%s\t%s\n", tmp1.name, tmp1.age, tmp1.sex, tmp1.unit, tmp1.tel, tmp1.ip);

}

else if ((SEND != 0 && !strcmp(tmp1.name, cont.name)))

Send(client, tmp1);

}

}

struct Contacts tmp = {"0",0,true,"","",0};

cont = tmp;

fclose(filein);

return NO\_SUCHNAME;

}

Status Insert(Contacts cont)//后台数据的插入

{

FILE \*fileout = fopen("contacts.dat", "ab");

if (!fileout)

return NO\_ADDRESSEXIST;

fwrite(&cont,sizeof(Contacts),1,fileout);

fclose(fileout);

return OK;

}

Status Send(SOCKET \*client,Contacts cont)//发送联系人至client

{

send(\*client, (char \*)&cont, sizeof(cont), 0);

return OK;

}

# Local.cpp

#include "function.h"

int Local()

{

while(1)

{

char no[2];

printf("请选择选项:\n");

scanf("%s", no);

if (atoi(no) == 1)

AllAddress();

if (atoi(no) == 2)

{

AddInformation();

}

if (atoi(no) == 3)

{

char \*name;

name = (char \*)malloc(20);

printf("请输入你要删除联系人的名字:");

scanf("%s", name);

Del(name);

}

if (atoi(no) == 4)

break;

}

return 0;

}

# Net.cpp

#include "function.h"

int Net()

{

WORD sockVersion = MAKEWORD(2, 2);//初始化WSA

WSADATA wsaData;

if (WSAStartup(sockVersion, &wsaData) != 0)

{

return 0;

}

SOCKET server = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);//创建套接字

if (server == INVALID\_SOCKET)

{

printf("socket error!");

return 0;

}

//绑定通信端口

sockaddr\_in sin;

sin.sin\_family = AF\_INET;

sin.sin\_port = htons(8088);

sin.sin\_addr.S\_un.S\_addr = INADDR\_ANY;

if (bind(server, (LPSOCKADDR)&sin, sizeof(sin)) == SOCKET\_ERROR)

{

printf("bind error!");

}

//开始监听

if (listen(server, 5) == SOCKET\_ERROR)

{

printf("listen error!");

return 0;

}

printf("等待消息接收!\n");

//等待接收数据

SOCKET s\_client;

sockaddr\_in destaddr;

int nAddrlen = sizeof(destaddr);

char \*sendData;

char \*revData;

while (1)

{

s\_client = accept(server, (SOCKADDR \*)&destaddr, &nAddrlen);

if (s\_client == INVALID\_SOCKET)

{

printf("accpet error!\n");

continue;

}

sendData = (char \*)malloc(MAX\_DATA\_LEN);

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "0(来自服务器的欢迎信息)欢迎使用基于Socket的通讯录！\n请输入你要执行的功能序号：\n");

send(s\_client, sendData, strlen(sendData), 0);

printf("接收到一个连接:%s\n", inet\_ntoa(destaddr.sin\_addr));

revData = (char \*)malloc(MAX\_DATA\_LEN);

while (1)

{

memset(revData, '0', MAX\_DATA\_LEN);

int ret = recv(s\_client, revData, MAX\_DATA\_LEN, 0);

revData[ret] = 0x00;

if (ret > 0 && atoi(revData) > 0 && isdigit(revData[0]))

{

revData[ret] = 0x00;

printf("客户端请求的指令为%s\n", revData);

if (atoi(&revData[0]) == 1)

{

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "1\n");

send(s\_client, sendData, strlen(sendData), 0);

int retn = AllAddress(&s\_client, &Send);

if (retn == NO\_ADDRESSEXIST)

{

memset(sendData, '0', MAX\_DATA\_LEN);

Contacts tmp1 = { "",0,0,"","",0 };

send(s\_client, (char \*)&tmp1, sizeof(tmp1), 0);

strcpy(sendData, "0服务端没有通讯录!\n");

send(s\_client, sendData, strlen(sendData), 0);

}

if (retn == OK)

{

memset(sendData, '0', MAX\_DATA\_LEN);

Contacts tmp1 = { "",0,0,"","",0 };

send(s\_client, (char \*)&tmp1, sizeof(tmp1), 0);

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "0通讯录发送完毕!\n");

send(s\_client, sendData, strlen(sendData), 0);

}

}

if (atoi(&revData[0]) == 2)

{

Contacts cont;

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "2请录入信息:\n");

send(s\_client, sendData, strlen(sendData), 0);

memset(revData, '0', MAX\_DATA\_LEN);

while ((ret = recv(s\_client, (char \*)&cont, MAX\_DATA\_LEN, 0))<0);

if (ret > 0)

{

strcpy(cont.ip, inet\_ntoa(destaddr.sin\_addr));

Insert(cont);

}

}

if (atoi(&revData[0]) == 3)

{

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "0(服务端确认)客户端主动关闭通信!\n");

printf("%s", &sendData[1]);

send(s\_client, sendData, strlen(sendData), 0);

break;

}

//信息确认接收

char \*tmp;

tmp = (char\*)malloc(MAX\_DATA\_LEN);

recv(s\_client, tmp, MAX\_DATA\_LEN, 0);

free(tmp);

//###end###

continue;

}

else if (ret>0)

{

revData[ret] = 0x00;

printf("%s\n", revData);

memset(sendData, '0', MAX\_DATA\_LEN);

strcpy(sendData, "0你好!\n");

send(s\_client, sendData, strlen(sendData), 0);

}

}

closesocket(s\_client);

}

closesocket(server);

if (!revData)

free(revData);

if (sendData)

free(sendData);

WSACleanup();

return 0;

}

# Server.cpp

#include "function.h" //文件读取的函数包含

#define MAX\_DATA\_LEN 1024 //最大接收字长

//外部函数声明

extern int Local();

extern int Net();

//#####end#####

int main()

{

char no[2];

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*本服务默认端口:8088\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*通讯录服务端\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("##################基于tcp的简易通讯录##################\n");

printf("# 1.显示所有联系人信息 #\n");

printf("# 2.添加联系人信息 #\n");

printf("# 3.删除联系人 #\n");

printf("# 4.退出 #\n");

printf("#######################################################\n");

printf("请选择模式:1.本地管理模式 2.网路服务器模式\n");

while (true)

{

scanf("%s", &no);

if (atoi(no) !=1&&atoi(no) !=2&&atoi(no) != 4)

{

printf("非法的输入!\n");

printf("退出系统\n");

Sleep(1000);

return -1;

}

if (atoi(no) == 2)

{

printf("服务器模式\n");

Net();

}

else if (atoi(no) == 1)

{

printf("本地模式\n");

Local();

}

else if (atoi(no) == 4)

{

printf("正常退出系统\n");

Sleep(1000);

return 0;

}

printf("退回主菜单!\n");

}

}

# Datagram.h

#pragma once

typedef struct Contacts

{

char name[20];

int age;

bool sex;

char unit[30];

char tel[15];

char ip[20];

}Contacts;

# Function.h

#pragma once

#include <stdio.h>

#include <winsock2.h>

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

#include"datagram.h"//用于通讯录读取的结构体

//文件读取状态的宏定义

#define NO\_ADDRESSEXIST -1

#define NO\_SUCHNAME -2

#define OK 0

#pragma warning(disable:4996) //屏蔽scanf等不安全警告的宏声明

#define MAX\_DATA\_LEN 1024

typedef int Status;

Status AddInformation(SOCKET \*server = NULL, Status SEND(SOCKET \*server, Contacts cont) = 0);//添加联系人

Status Send(SOCKET \*client,Contacts cont);//后台数据发送

Status AllAddress(SOCKET \*client = NULL ,Status SEND(SOCKET \*client,Contacts) = 0);//后台数据遍历

Status Del(char \*name, SOCKET \*client = NULL, Status SEND(SOCKET \*client, Contacts) = 0);

Status Search(Contacts &cont, SOCKET \*client, Status SEND(SOCKET \*client, Contacts) = 0);//后台数据的查找

Status Insert(Contacts cont);//后台数据的插入

**---------Client------------**

# Client.cpp

#include <winsock2.h>

#include <stdio.h>

#include "../server/datagram.h"

#define MAX\_DATA\_LEN 1024

char destip[20];

struct hostent \*hp = NULL;

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

int main()

{

system("color 0a");

printf("\t\tC/S通讯录客户端\n");

printf("\t\t本服务默认端口:8088\n");

printf("\t##################################\n");

printf("\t请输入服务器ip地址或域名:");

scanf("%s",destip);

WORD sockVersion = MAKEWORD(2, 2);

WSADATA data;

if (WSAStartup(sockVersion, &data) != 0)

{

return 0;

}

SOCKET c\_server = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

if (c\_server == INVALID\_SOCKET)

{

printf("invalid socket !");

return 0;

}

sockaddr\_in serAddr;

if ((hp = gethostbyname(destip)) != NULL)

{

memcpy(&(serAddr.sin\_addr),hp->h\_addr,hp->h\_length);

serAddr.sin\_family = hp->h\_addrtype;

}

else

{

printf("请输入正确的地址!");

Sleep(1000);

return 0;

}

//serAddr.sin\_family = AF\_INET;

serAddr.sin\_port = htons(8088);

//serAddr.sin\_addr.S\_un.S\_addr = inet\_addr(destip);

if (connect(c\_server, (sockaddr \*)&serAddr, sizeof(serAddr)) == SOCKET\_ERROR)

{

printf("connect error !");

closesocket(c\_server);

return 0;

}

printf("\t\t1.显示所有联系人信息\n");

printf("\t\t2.添加联系人信息\n");

printf("\t\t3.退出\n");

printf("\t##################################\n");

char \*sendData;

char \*recData;

Sleep(1000);

recData = (char \*)malloc(MAX\_DATA\_LEN);

sendData = (char \*)malloc(MAX\_DATA\_LEN);

memset(recData, 0, sizeof(recData));

int ret = recv(c\_server, recData, MAX\_DATA\_LEN, 0);

if (ret > 0)

{

recData[ret] = 0x00;

printf(&recData[1]);

}

while (1)

{

memset(sendData, 0, sizeof(sendData));

scanf("%s", sendData);

send(c\_server, sendData, strlen(sendData), 0);

memset(recData, 0, sizeof(sendData));

int ret = 0;

ret = recv(c\_server, recData, MAX\_DATA\_LEN, 0);

if (ret > 0&&atoi(&recData[0]) == 0)

{

recData[ret] = 0x00;

printf(&recData[1]);

}

if (ret > 0 && atoi(&recData[0]) == 1)

{

Contacts tmp;

ret = 0;

while (ret = recv(c\_server, (char \*)&tmp, sizeof(Contacts), 0) > 0)

{

if (strcmp(tmp.name, ""))

{

printf("%s\t%d\t%s\t%s\t%s\t%s\n", tmp.name, tmp.age, tmp.sex?"男":"女", tmp.unit, tmp.tel, tmp.ip);

}

else

break;

}

memset(recData, 0, MAX\_DATA\_LEN);

ret = recv(c\_server,recData,MAX\_DATA\_LEN,0);

if (ret > 0)

{

recData[ret] = 0x00;

printf(&recData[1]);

}

}

if (ret > 0 && atoi(&recData[0]) == 2)

{

int tmp;

char name[20];

int age;

bool sex;

char unit[30];

char tel[15];

int ip;

printf("请依次输入姓名,年龄,性别(男:1;女:0),单位,电话(以回车隔开)\n");

scanf("%s", name);

scanf("%d", &age);

scanf("%d", &tmp);

if (tmp == 1)

sex = 1;

else

sex = 0;

scanf("%s", unit);

scanf("%s", tel);

ip = 0;

Contacts cont;

memcpy(cont.name, &name, sizeof(name));

cont.age = age;

cont.sex = sex;

memcpy(cont.unit, &unit, sizeof(unit));

memcpy(cont.tel, &tel, sizeof(tel));

//#########调试用########

printf("%s\t%d\t%s\t%s\t%s\n", cont.name, cont.age, cont.sex ? "男" : "女", cont.unit, cont.tel);

//#######################

send(c\_server, (char \*)&cont, sizeof(Contacts), 0);

printf("#########信息录入完毕########\n");

}

if (ret > 0 && atoi(&recData[0]) == 3)

{

memset(sendData,0,MAX\_DATA\_LEN);

strcpy(sendData,"3\n");

send(c\_server,sendData,sizeof(sendData),0);

memset(recData, 0, MAX\_DATA\_LEN);

recData[recv(c\_server, recData, MAX\_DATA\_LEN, 0)] = 0xff;

printf("%s",recData);

break;

}

//###发送确认接收消息###

memset(sendData, 0, MAX\_DATA\_LEN);

strcpy(sendData, "OK\n");

send(c\_server, sendData, strlen(sendData), 0);

//######################

}

closesocket(c\_server);

WSACleanup();

if (!sendData)

free(sendData);

if (!recData)

free(recData);

return 0;

}