**Простейший эхо-сервер (на базе потоковых сокетов) и простейший клиент**

* Создать 2 (два) консольных проекта и поместить туда код сервера и клиента
* Запустить сначала сервер
* Потом запускать клиентов
* Убедиться, что все работает
* Потом в режиме отладки (по шагам ) смотреть как работает сервер и как работает клиент. Понять, в какие моменты происходит зависание программы.

Код сервера

using namespace std;

#include <stdio.h>

#include <winsock2.h>

#include <conio.h>

#include <iostream>

#pragma comment(lib, "Ws2\_32.lib")// подключили библиотеку

#define SERVER\_PORT 3820

int main()

{setlocale(LC\_CTYPE, "Russian"); // включаем русскую кодировку

struct sockaddr\_in SrvAddr; // Адресная структура сервера

struct sockaddr\_in ConnAddr; // Адресная структура клиента

SOCKET SrvSock, Conn;

WSADATA wsaData; // для работы Windows с сокетами

int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData); // запустили библиотеку

if(iResult<0)

{cout<<"ошибка"<<endl;

getchar();

exit(0);

}

SrvSock = socket(AF\_INET, SOCK\_STREAM, 0);//Создаем сокет сервера

//Задаем адрес сервера

SrvAddr.sin\_family = AF\_INET;

SrvAddr.sin\_addr.s\_addr = htonl(INADDR\_LOOPBACK);// 127.0.0.1

SrvAddr.sin\_port = htons(SERVER\_PORT);

//Настраиваем сокет

bind(SrvSock, (sockaddr \*)&SrvAddr, sizeof SrvAddr);

//Слушаем

listen(SrvSock, 5);

printf("...Сервер работает...\n");

int AddrLen = sizeof(ConnAddr);

char buf\_in[20];

char exit[20]="exit";

char buf\_out[34]="Привет от сервера!";

int nsize;

int bytes=0;

bool stop=false;

//Ждем очередного клиента

while(1)

{

Conn = accept(SrvSock, (struct sockaddr \*) &ConnAddr, &AddrLen); // сервер завис в режиме ожидания

//HOSTENT\* hst ;

//hst = gethostbyaddr((char \*)&ConnAddr. sin\_addr.s\_addr, 4, AF\_INET);

//cout<<"Подключился " << inet\_ntoa(ConnAddr.sin\_addr)<<endl;

bytes = recv(Conn, (char \*)buf\_in, sizeof (buf\_in), 0);//принял информацию в буфер

cout<<buf\_in<<endl;

send(Conn, (char \*) buf\_out,sizeof (buf\_out) , 0);// отправил "привет"

}

shutdown(Conn,2); // запретили передачу и прием сообщений

closesocket(Conn); //закрыли=разрушили сокет

cout<<endl<<"Выход из серверной части программы" << endl;

getch();

return 0;

}

Код клиента

using namespace std;

#include <stdio.h>

#include <winsock2.h>

#include <conio.h>

#include <iostream>

#pragma comment(lib, "Ws2\_32.lib")

#define SERVER\_PORT 3820

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian"); // включаем русскую кодировку

SOCKET ClientSock = INVALID\_SOCKET;

WSADATA WSStartData; /\* Структура данных библиотеки сокета. \*/

CHAR DefaultIPAddr[] = "127.0.0.1"; /\* Локальная система. \*/

struct sockaddr\_in Addr; /\* Адрес сокета \*/

WSADATA wsaData;

int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData);

if (iResult < 0)

{

cout << endl << "ошибка на клиенте" << endl;

getchar();

exit(0);

}

// Создали сокет

ClientSock = socket(AF\_INET, SOCK\_STREAM, 0);

// Задаем адрес сервера

memset(&Addr, 0, sizeof(Addr));

Addr.sin\_family = AF\_INET;

Addr.sin\_addr.s\_addr = htonl(INADDR\_LOOPBACK);// адрес для отладки

Addr.sin\_port = htons(SERVER\_PORT);

// HOSTENT\* hst ;

char hostname[] = " ";

DWORD size = 20;

char\* localIP;

//Соединяемся с сервером

int u;

u = connect(ClientSock, (struct sockaddr\*)&Addr, sizeof(Addr));

if (u == INVALID\_SOCKET)

{

cout << "Не соединились "; getchar(); return 0;

}

else cout << "Cоединились!\n ";

int bytes = 0;

char rez[34] = "";

char buf[20] = "начало";

int len;

cout << "input: ";

cin >> buf;

len = sizeof(buf);

//Послали сообщение

send(ClientSock, (char\*)buf, len, 0);

//Получили сообщение

bytes = recv(ClientSock, (char\*)rez, sizeof(rez), 0);

if (bytes == -1) cout << "результат с ошбкой";

else cout << rez;

shutdown(ClientSock, 2); /\* Запретить посылку и прием сообщений. \*/

closesocket(ClientSock);//закрыть сокет

WSACleanup();//закрыть библиотеку

cout << endl << "Выход из клиентской части программы" << endl;

getchar(); getchar();

return 0;

}