|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 2 | | |
| по дисциплине «Операционные системы и компьютерный сети» | | |
| Технология клиент-сервер: эхо-повтор | | |
|  | | |
|  | Бригада 4 | Лойченко данила |
| Группа ПМ-12 | ушатов сегргей |
|  |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | кобылянский валерий григорьевич |
|  | Сивак Мария Алекссевна |
| Новосибирск, 2023 | | |

1. Цель работы

Изучить основные принципы разработки клиент-серверных приложений на примере простейшей однопользовательской программы.

1. Выполнение работы

Клиент пересылает серверу данные (строки текста). Сервер создает файл с уникальным именем, записывает в него полученные от клиента данные и в качестве результата обработки данных отправляет клиенту имя созданного файла. После получения ответа с сервера клиент выводит на экран содержимое указанного сервером файла.

Файл сервера

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <winsock2.h>

#include <fstream>

#include <string>

#include <cstring>

#pragma comment(lib, "WS2\_32.lib")

using namespace std;

string FileName() { //Функция создания уникального имени файла

time\_t t = time(nullptr);

tm\* now = localtime(&t);

strftime(curTime, sizeof(curTime), "%d\_%m\_%Y\_%X", now);

int i = 0;

while (curTime[i]) {

if (curTime[i] == ':') curTime[i] = '.';

i++;

}

return curTime;

}

int main() {

WSADATA WSAData; //Создание структуры, в которую загружаются данные о версии сокетов, используемых ОС.

WSAStartup(MAKEWORD(2, 2), &WSAData); //Запуск сокетов с указанием версии реализации

SOCKET client, server = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP); //Создание сокетов

if (server == INVALID\_SOCKET) {

cout << "Socket creation failed. Error number:" << WSAGetLastError() << endl;

return 1;

}

//Адресная информация

SOCKADDR\_IN serverAddr;

serverAddr.sin\_addr.s\_addr = INADDR\_ANY; //Регистрирование сервера на всех адресах машины, на которой он запущен

serverAddr.sin\_family = AF\_INET; //Семейство

serverAddr.sin\_port = htons(2004); //Порт

if (bind(server, (SOCKADDR\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) //Связывание сокета с семейством адресов

{

cout << "Bind function failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

cout << "Server started" << endl;

if (listen(server, 0) == SOCKET\_ERROR) { //Формирование очереди для входящего подключения и перевод сокета в режим прослушки

cout << "Listen function failed. Error number:" << WSAGetLastError() << endl;

return 1;

}

while (true) {

if ((client = accept(server, NULL, NULL)) != INVALID\_SOCKET) { //Создание нового сокета, для обмена данными

cout << "Connection with client is set" << endl;

char buffer[1024];

ofstream outFile;

string fileName = FileName() + ".txt";

outFile.open(fileName);

while (true) {

memset(buffer, 0, sizeof(buffer)); //Очищение буфера

if (recv(client, buffer, sizeof(buffer), 0) == SOCKET\_ERROR) { //Проверка на удачное получение данных из буфера

cout << "recv function failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

if (strcmp(buffer, "/exit\n") == 0) { //Выход из диалогового окна

memset(buffer, 0, sizeof(buffer)); //Очищение буфера

outFile.close();

break;

}

outFile << buffer;

cout << buffer;

}

strcpy\_s(buffer, fileName.c\_str());

if (send(client, buffer, sizeof(buffer), 0) == SOCKET\_ERROR) { //Отправка клиенту имени файла

cout << "Send failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

closesocket(client);

cout << "Close socket client" << endl;

}

}

if (closesocket(server) == SOCKET\_ERROR) {

cout << "Close socket failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

WSACleanup();

return 0;

}

Файл клиента

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <iostream>

#include <cstdio>

#include <cstring>

#include <fstream>

#include <winsock2.h>

#include <string>

#pragma comment(lib, "WS2\_32.lib")

using namespace std;

int main()

{

WSADATA WSAData; //Создание структуры, в которую загружаются данные о версии сокетов, используемых ОС.

SOCKET server;

SOCKADDR\_IN addr;

WSAStartup(MAKEWORD(2, 2), &WSAData); //Запуск сокетов с указанием версии реализации

if ((server = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP)) == INVALID\_SOCKET) {

cout << "Socket creation failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

//Адресная информация

SOCKADDR\_IN addr;

string ipServ;

int portServ;

cout << "Input IP server: ";

cin >> ipServ;

cout << "Input port server: ";

cin >> portServ;

addr.sin\_addr.s\_addr = inet\_addr(ipServ.c\_str()); //IP

addr.sin\_family = AF\_INET; //Семейство

addr.sin\_port = htons(portServ); //Порт

if (connect(server, (SOCKADDR\*)&addr, sizeof(addr)) == SOCKET\_ERROR) { //Присоединение к серверу

cout << "Server connection failed. Error number: " << WSAGetLastError() << endl;

return 1;

}

char buffer[1024] = { 0 };

cout << "Write what you want to write to the file. To stop recording, write \"/exit\"" << endl;

while (true) {

fgets(buffer, 1024, stdin); //Отправление серверу текста

if (send(server, buffer, sizeof(buffer), 0) == SOCKET\_ERROR) {

cout << "send failed with error: " << WSAGetLastError() << endl;

return 1;

}

if (strcmp(buffer, "/exit\n") == 0) {

memset(buffer, 0, sizeof(buffer));

break;

}

memset(buffer, 0, sizeof(buffer));

}

if (recv(server, buffer, sizeof(buffer), 0) == SOCKET\_ERROR) { //Получение от сервера данных

cout << "recv function failed with error: " << WSAGetLastError() << endl;

return 1;

}

cout << "Output file name: " << buffer << endl;

closesocket(server);

WSACleanup();

}

Проверка работоспособности программы





