

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 4

Дисциплина: «Технологии обработки информации»

Тема: «Разработка клиент-серверного приложения»

Выполнил: студент группы 211-723

Сергеев Станислав Олегович

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) **(Оценка)**

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва

2024

Тема:

Разработка клиент-серверного приложения

Цель работы:

Разработать консольное сетевое приложение, предусматривающее возможность установления соединения с сервером, передачу файла на сервер и чтение файла с сервера. В приложении должен быть полноценный консольный интерфейс и реализована защита от неправильного ввода данных.

Ход работы:

Client.cpp:

```
#define WIN32_LEAN_AND_MEAN

#pragma comment(lib, "ws2_32.lib")
#include <winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <iostream>
#pragma warning(disable: 4996)

using namespace std;

int main() {
    // Инициализация Winsock
    WSADATA ws;
    WSStartup(MAKEWORD(2, 2), &ws);

    // Создание сокета
    SOCKET clientSocket = socket(AF_INET, SOCK_STREAM, 0);
    if (clientSocket == INVALID_SOCKET) {
        perror("socket call failed");
        exit(1);
    }

    // Подключение к серверу
    sockaddr_in serverAddr;
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(1111);
    serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    int connectResult = connect(clientSocket, (sockaddr*)&serverAddr,
sizeof(serverAddr));
    if (connectResult == SOCKET_ERROR) {
        perror("connect call failed");
        exit(1);
    }

    // Выбор режима работы
```

```

char choice;
cout << "Read or write file (0/1): ";
cin >> choice;

// Отправка выбранного режима на сервер
send(clientSocket, &choice, sizeof(choice), 0);

// Режим чтения
if (choice == '0') {
    // Ввод имени файла
    string fileName;
    cout << "Enter file name: ";
    cin >> fileName;

    // Отправка имени файла на сервер
    send(clientSocket, fileName.c_str(), fileName.size() + 1, 0);

    // Получение содержимого файла
    char buffer[1024];
    int bytesReceived;
    while ((bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0)) > 0) {
        cout << buffer;
    }
    cout << endl;
    // Режим записи
}
else if (choice == '1') {
    // Ввод имени файла
    string fileName;
    cout << "Enter file name: ";
    cin >> fileName;

    // Отправка имени файла на сервер
    send(clientSocket, fileName.c_str(), fileName.size() + 1, 0);

    // Отправка содержимого файла
    ifstream file(fileName, ios::binary);
    if (!file) {
        cout << "Failed to open file." << endl;
        return -1;
    }

    char buffer[1024];
    while (!file.eof()) {
        file.read(buffer, sizeof(buffer));
        send(clientSocket, buffer, file.gcount(), 0);
    }

    file.close();
    cout << "File sent successfully." << endl;
}

// Закрытие сокета
closesocket(clientSocket);
WSACleanup();

return 0;
}

```

Server.cpp:

```
#define WIN32_LEAN_AND_MEAN

#pragma comment(lib, "ws2_32.lib")
#include <winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <iostream>
#include <string>
#pragma warning(disable: 4996)
using namespace std;

int main() {
    // Инициализация Winsock
    WSADATA ws;
    WSASStartup(MAKEWORD(2, 2), &ws);

    // Создание сокета
    SOCKET serverSocket = socket(AF_INET, SOCK_STREAM, 0);
    if (serverSocket == INVALID_SOCKET) {
        perror("socket call failed");
        exit(1);
    }

    // Привязка сокета к порту
    sockaddr_in serverAddr;
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(1111);
    serverAddr.sin_addr.s_addr = INADDR_ANY;
    int bindResult = bind(serverSocket, (sockaddr*)&serverAddr, sizeof(serverAddr));
    if (bindResult == SOCKET_ERROR) {
        perror("bind call failed");
        exit(1);
    }

    // Прослушивание порта
    listen(serverSocket, 5);

    // Прием подключения от клиента
    SOCKET clientSocket = accept(serverSocket, NULL, NULL);
    if (clientSocket == INVALID_SOCKET) {
        perror("accept call failed");
        exit(1);
    }

    // Получение выбранного режима от клиента
    char choice;
    recv(clientSocket, &choice, sizeof(choice), 0);

    // Режим чтения
    if (choice == '0') {
        // Получение имени файла от клиента
        char fileName[256];
        recv(clientSocket, fileName, sizeof(fileName), 0);

        // Открытие файла
        ifstream file(fileName, ios::binary);
        if (!file) {
            cout << "File not found." << endl;
            return -1;
        }

        // Отправка содержимого файла клиенту
    }
}
```

```

    string buffer, content;
    while (getline(file, buffer)) {
        content += buffer + "\n";
    }
    send(clientSocket, content.c_str(), content.size() + 1, 0);

    file.close();

    // Режим записи
}
else if (choice == '1') {
    // Получение имени файла от клиента
    char fileName[256];
    recv(clientSocket, fileName, sizeof(fileName), 0);

    // Создание файла
    ofstream file(fileName);

    if (!file) {
        cout << "Failed to create file." << endl;
        return -1;
    }

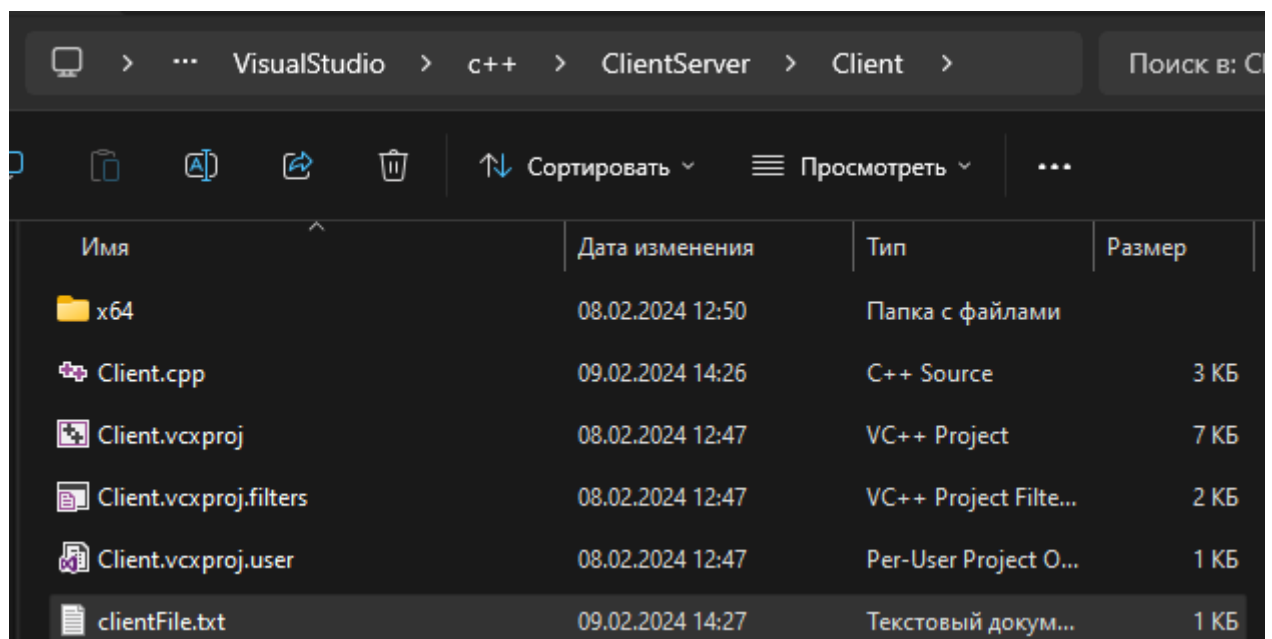
    // Получение содержимого файла от клиента
    char buffer[1024];
    int bytesReceived;
    while ((bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0)) > 0) {
        file.write(buffer, bytesReceived);
    }

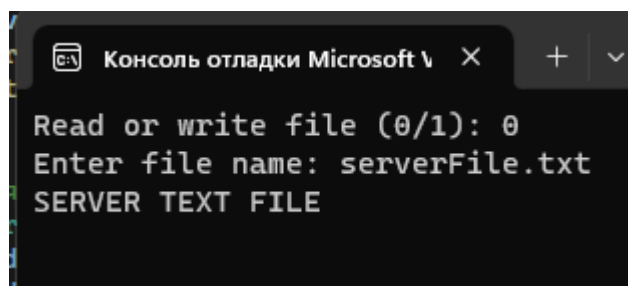
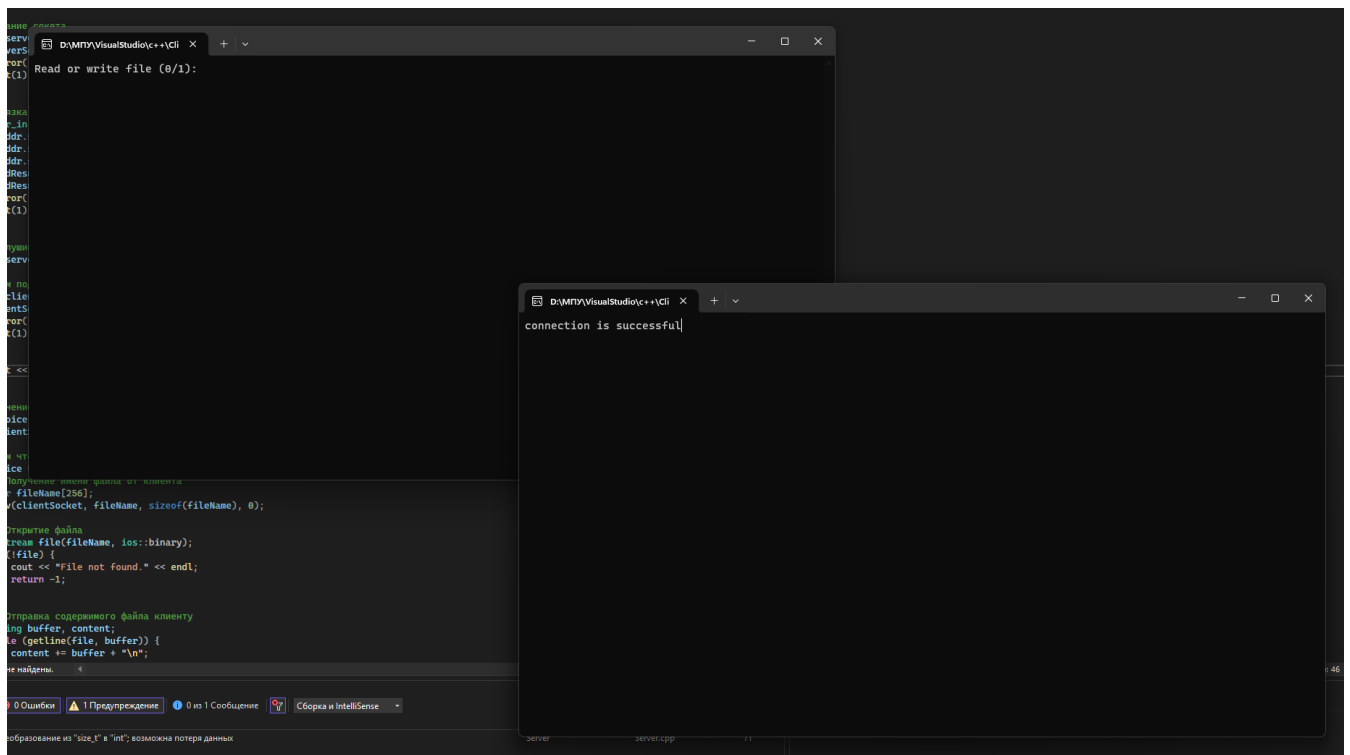
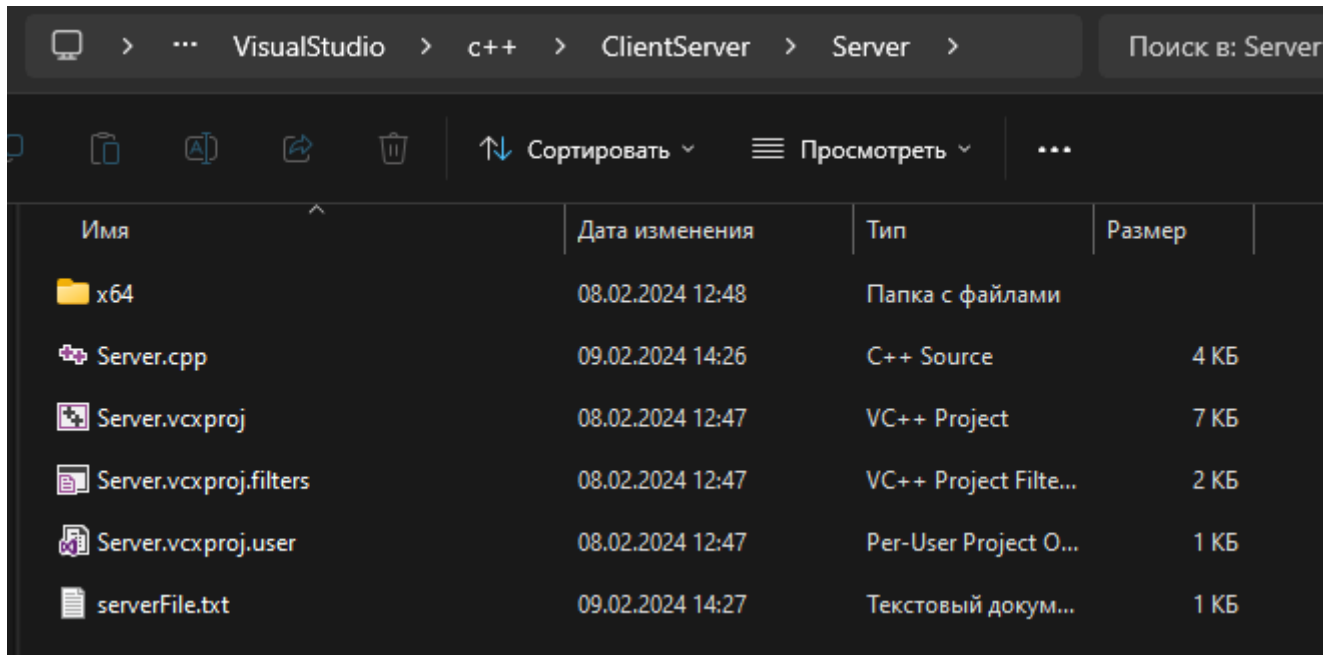
    file.close();
    cout << "File received successfully." << endl;
}

// Закрытие сокетов
closesocket(clientSocket);
closesocket(serverSocket);
WSACleanup();

return 0;
}

```





```
Консоль отладки Microsoft Visual Studio
Read or write file (0/1): 1
Enter file name: clientFile.txt
File sent successfully.

D:\МПУ\VisualStudio\c++\ClientServer
Чтобы автоматически закрывать консо
```

VisualStudio > c++ > ClientServer > Server					Поиск в: Ser
Сортировать					Просмотреть
Имя	Дата изменения	Тип	Размер		
x64	08.02.2024 12:48	Папка с файлами			
Server.cpp	09.02.2024 14:26	C++ Source	4 КБ		
Server.vcxproj	08.02.2024 12:47	VC++ Project	7 КБ		
Server.vcxproj.filters	08.02.2024 12:47	VC++ Project Filte...	2 КБ		
Server.vcxproj.user	08.02.2024 12:47	Per-User Project O...	1 КБ		
serverFile.txt	09.02.2024 14:27	Текстовый докум...	1 КБ		
clientFile.txt	09.02.2024 14:33	Текстовый докум...	1 КБ		

```
Консоль отладки Microsoft Visual Studio
Read or write file (0/1): 0
Enter file name: clientFile.txt
CLIENT TEXT FILE

D:\МПУ\VisualStudio\c++\ClientServer
Чтобы автоматически закрывать консо
томатически закрыть консоль при о
Нажмите любую клавишу, чтобы закр
```

Вывод:

В ходе выполнения лабораторной работы я научился разрабатывать консольное сетевое приложение, предусматривающее возможность установления соединения с сервером, передачу файла на сервер и чтение файла с сервера. Приложение имеет полноценный консольный интерфейс и реализована защита от неправильного ввода данных.