Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра Математичних проблем управління і кібернетики

Лабораторна робота № 1.1

“Тема: СТВОРЕННЯ КАНАЛУ ЗВ'ЯЗКУ. TCP”

з дисципліни

“Програмування комп’ютерних мереж”

Виконав:

ст. гр. 541 Гришко Владислав Валерійович

Прийняв:

Доцент Лазорик В. В.

Чернівці – 2024

**Хід виконання роботи**

Код Серверу:

#include <iostream>  
#include <winsock2.h>  
#include <Ws2tcpip.h>  
#include <fstream>  
  
#pragma comment(lib, "ws2\_32.lib")  
  
#define PORT 27015  
#define BUFFER\_SIZE 1024  
  
void startServer() {  
 WSADATA wsaData;  
 SOCKET serverSocket, clientSocket;  
 sockaddr\_in serverAddr, clientAddr;  
 int clientAddrSize = sizeof(clientAddr);  
 SetConsoleOutputCP(CP\_UTF8);  
  
 if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {  
 std::cerr << "Не вдалося ініціалізувати Winsock. Код помилки: " << WSAGetLastError() << std::endl;  
 return;  
 }  
  
 serverSocket = socket(AF\_INET, SOCK\_STREAM, 0);  
 if (serverSocket == INVALID\_SOCKET) {  
 std::cerr << "Не вдалося створити сокет. Код помилки: " << WSAGetLastError() << std::endl;  
 WSACleanup();  
 return;  
 }  
  
 serverAddr.sin\_family = AF\_INET;  
 serverAddr.sin\_addr.s\_addr = INADDR\_ANY;  
 serverAddr.sin\_port = htons(PORT);  
  
 if (bind(serverSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) {  
 std::cerr << "Не вдалося виконати прив'язку. Код помилки: " << WSAGetLastError() << std::endl;  
 closesocket(serverSocket);  
 WSACleanup();  
 return;  
 }  
  
 if (listen(serverSocket, SOMAXCONN) == SOCKET\_ERROR) {  
 std::cerr << "Не вдалося почати прослуховування. Код помилки: " << WSAGetLastError() << std::endl;  
 closesocket(serverSocket);  
 WSACleanup();  
 return;  
 }  
  
 std::cout << "Сервер запущений на порту " << PORT << std::endl;  
  
 clientSocket = accept(serverSocket, (sockaddr\*)&clientAddr, &clientAddrSize);  
 if (clientSocket == INVALID\_SOCKET) {  
 std::cerr << "Не вдалося прийняти підключення. Код помилки: " << WSAGetLastError() << std::endl;  
 closesocket(serverSocket);  
 WSACleanup();  
 return;  
 }  
  
 std::cout << "Клієнт підключено." << std::endl;  
  
 std::string fileName = "Server.exe";  
  
 send(clientSocket, fileName.c\_str(), fileName.size(), 0);  
  
 std::ifstream file(fileName, std::ios::*binary*);  
 if (!file.is\_open()) {  
 std::cerr << "Не вдалося відкрити файл." << std::endl;  
 closesocket(clientSocket);  
 closesocket(serverSocket);  
 WSACleanup();  
 return;  
 }  
  
 char buffer[BUFFER\_SIZE];  
 while (file.read(buffer, sizeof(buffer)) || file.gcount() > 0) {  
 int bytesSent = send(clientSocket, buffer, file.gcount(), 0);  
 if (bytesSent == SOCKET\_ERROR) {  
 std::cerr << "Помилка під час відправки даних. Код помилки: " << WSAGetLastError() << std::endl;  
 break;  
 }  
 }  
  
 std::cout << "Файл відправлено." << std::endl;  
  
 closesocket(clientSocket);  
 closesocket(serverSocket);  
 WSACleanup();  
}  
  
int main() {  
 startServer();  
 return 0;  
}

Код клієнта:   
#include <iostream>  
#include <Ws2tcpip.h>  
#include <winsock2.h>  
#include <fstream>  
  
#pragma comment(lib, "ws2\_32.lib")  
  
#define SERVER\_IP "127.0.0.1"  
#define PORT 27015  
#define BUFFER\_SIZE 1024  
  
void startClient() {  
 WSADATA wsaData;  
 SOCKET clientSocket;  
 sockaddr\_in serverAddr;  
 SetConsoleOutputCP(CP\_UTF8);  
  
 if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {  
 std::cerr << "Не вдалося ініціалізувати Winsock. Код помилки: " << WSAGetLastError() << std::endl;  
 return;  
 }  
  
 clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);  
 if (clientSocket == INVALID\_SOCKET) {  
 std::cerr << "Не вдалося створити сокет. Код помилки: " << WSAGetLastError() << std::endl;  
 WSACleanup();  
 return;  
 }  
  
 serverAddr.sin\_family = AF\_INET;  
 InetPton(AF\_INET, SERVER\_IP, &serverAddr.sin\_addr.s\_addr);  
 serverAddr.sin\_port = htons(PORT);  
  
 if (connect(clientSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) {  
 std::cerr << "Не вдалося підключитися до сервера. Код помилки: " << WSAGetLastError() << std::endl;  
 closesocket(clientSocket);  
 WSACleanup();  
 return;  
 }  
  
 std::cout << "Підключено до сервера." << std::endl;  
  
 char fileName[BUFFER\_SIZE];  
 int fileNameSize = recv(clientSocket, fileName, BUFFER\_SIZE, 0);  
 if (fileNameSize <= 0) {  
 std::cerr << "Не вдалося отримати назву файлу від сервера." << std::endl;  
 closesocket(clientSocket);  
 WSACleanup();  
 return;  
 }  
 fileName[fileNameSize] = '\0';  
  
 std::cout << "Отримано назву файлу: " << fileName << std::endl;  
  
 std::ofstream outputFile(fileName, std::ios::*binary*);  
 if (!outputFile.is\_open()) {  
 std::cerr << "Не вдалося відкрити файл для запису." << std::endl;  
 closesocket(clientSocket);  
 WSACleanup();  
 return;  
 }  
  
 char buffer[BUFFER\_SIZE];  
 int bytesReceived;  
 while ((bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0)) > 0) {  
 outputFile.write(buffer, bytesReceived);  
 }  
  
 std::cout << "Файл отримано і збережено як " << fileName << std::endl;  
  
 closesocket(clientSocket);  
 WSACleanup();  
}  
  
int main() {  
 startClient();  
 return 0;  
}

**Результат**:

Передача файлу Report.docx

1. Запускаємо сервер на порту 27015:

Зображення, що містить знімок екрана, програмне забезпечення, текст, Мультимедійне програмне забезпечення

Автоматично згенерований опис

1. Запускаємо клієнта

Зображення, що містить текст, Мультимедійне програмне забезпечення, програмне забезпечення, знімок екрана

Автоматично згенерований опис

1. Клієнт підключається до серверу та сервер надсилає файл

Зображення, що містить текст, програмне забезпечення, Мультимедійне програмне забезпечення, знімок екрана

Автоматично згенерований опис

1. Клієнт отримує назву файлу та приймає файл

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис