|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Лабораторная работа № 2 | | |
| по дисциплине «Операционные системы и компьютерные сети» | | |
| **Технология клиент-сервер: эхо-повтор** | | |
|  | | |
|  | Бригада 11 | уваров артём |
| Группа ПМ-23 | румянцев артём |
| Место для ввода текста. |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Кобылянский Валерий Григорьевич |
|  | сивак мария алексеевна |
| Новосибирск, 2024 | | |

**Цель**

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

**Задание**

Написать простейшее приложение с одним сервером и одним клиентом, используя API-интерфейс низкого уровня. Пример реализации интерфейса приведен в приложении 1.

Сервер начинает свою работу с ожидания запроса от клиента на соединение. Клиент устанавливает связь с сервером и посылает набор данных, введенный пользователем, на сервер. Сервер получает от клиента набор данных, выполняет указанные в варианте действия и возвращает клиенту результат. После этого сервер снова переходит в состояние ожидания запроса на соединение. Клиент, получив ответ с сервера, распечатывает его на экране и прекращает свою работу.

**Вариант 3**

Клиент пересылает серверу данные (строки текста). Сервер в полученном тексте в конец каждого предложения вставляет свой IP-адрес и номер порта и возвращает в таком виде данные клиенту.

**Текст программ:**

**server.cpp**

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <winsock2.h>

#include <ws2tcpip.h>

#include <stdlib.h>

#include <stdio.h>

#include <iostream>

#include <sstream>

#include <string>

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

**using** **namespace** std**;**

string process**(**string input**,** string ip**,** unsigned short port**)** **{**

string result **=** input**;**

string toReplace **=** "."**;**

ostringstream oss**;**

oss **<<** ". (" **<<** ip **<<** " " **<<** port **<<** ")"**;**

std**::**string replacement **=** oss**.**str**();**

size\_t pos **=** 0**;**

**while** **((**pos **=** result**.**find**(**toReplace**,** pos**))** **!=** std**::**string**::**npos**)** **{**

result**.**replace**(**pos**,** toReplace**.**length**(),** replacement**);**

pos **+=** replacement**.**length**();**

**}**

**return** result**;**

**}**

int main**(**void**)**

**{**

SetConsoleCP**(**1251**);**

SetConsoleOutputCP**(**1251**);**

int retVal**;**

WSADATA wsaData**;**

WSAStartup**(**MAKEWORD**(**2**,** 2**),** **&**wsaData**);**

SOCKET SendRecvSocket **=** socket**(**AF\_INET**,** SOCK\_DGRAM**,** IPPROTO\_UDP**);**

**if** **(**SendRecvSocket **==** INVALID\_SOCKET**)**

cout **<<** "Создание сокета не удалось. Код ошибки: " **<<** WSAGetLastError**()** **<<** endl**;**

sockaddr\_in ServerAddr**;**

ServerAddr**.**sin\_family **=** AF\_INET**;**

ServerAddr**.**sin\_addr**.**s\_addr **=** inet\_addr**(**"25.14.120.71"**);**

ServerAddr**.**sin\_port **=** htons**(**1234**);**

retVal **=** bind**(**SendRecvSocket**,** **(**sockaddr**\*)&**ServerAddr**,** **sizeof(**ServerAddr**));**

**if** **(**retVal **==** SOCKET\_ERROR**)**

**{**

cout **<<** "Привязка не удалась. Код ошибки: " **<<** WSAGetLastError**()** **<<** endl**;**

closesocket**(**SendRecvSocket**);**

WSACleanup**();**

**return** 1**;**

**}**

cout **<<** "Сервер запущен по адресу " **<<** inet\_ntoa**(**ServerAddr**.**sin\_addr**)** **<<** ", port " **<<** htons**(**ServerAddr**.**sin\_port**)** **<<** endl**;**

**while** **(true)**

**{**

sockaddr\_in ClientAddr**;**

int ClientAddrSize **=** **sizeof(**ClientAddr**);**

char szReq**[**1024**];**

retVal **=** recvfrom**(**SendRecvSocket**,** szReq**,** 1024**,** 0**,** **(**sockaddr**\*)&**ClientAddr**,** **&**ClientAddrSize**);**//Пытаемся получить данные от клиента

**if** **(**retVal **==** SOCKET\_ERROR**)**

cout **<<** "Получить данные не удалось. Код ошибки: " **<<** WSAGetLastError**()** **<<** endl**;**

cout **<<** "Данные получены от " **<<** inet\_ntoa**(**ClientAddr**.**sin\_addr**)** **<<** ", port " **<<** htons**(**ClientAddr**.**sin\_port**)** **<<** endl**;**

szReq**[**retVal**]** **=** 0**;**

string s **=** **(**const char**\*)**szReq**;**

**if** **(**s**[**0**]** **==** 's'**)** // Команда на выключение сервера

**{**

retVal **=** sendto**(**SendRecvSocket**,** "Выключение сервера"**,** 20**,** 0**,** **(**sockaddr**\*)&**ClientAddr**,** **sizeof(**ClientAddr**));**

closesocket**(**SendRecvSocket**);**

**break;**

**}**

**else** //Обработка полученных предложений

**{**

char szResp**[**1024**]** **=** **{** '\0' **};**

string result **=** process**(**s**,** inet\_ntoa**(**ServerAddr**.**sin\_addr**),** htons**(**ServerAddr**.**sin\_port**));**

**if** **(**result**.**length**()** **>** 1023**)**

result **=** result**.**substr**(**0**,** 1023**);**

**for** **(**int i **=** 0**;** i **<** result**.**length**();** i**++)**

szResp**[**i**]** **=** result**[**i**];**

szResp**[**result**.**length**()]** **=** 0**;**

cout **<<** "Отправляем ответ из сервера" **<<** endl**;** //Пытаемся отослать данные клиенту

retVal **=** sendto**(**SendRecvSocket**,** szResp**,** result**.**length**(),** 0**,** **(**sockaddr**\*)&**ClientAddr**,** **sizeof(**ClientAddr**));**

**if** **(**retVal **==** SOCKET\_ERROR**)**

cout **<<** "Невозможно отправить данные. Код ошибки: " **<<** WSAGetLastError**()** **<<** endl**;**

**}**

**}**

closesocket**(**SendRecvSocket**);** //Завершение работы

WSACleanup**();**

**return** 0**;**

**}**

**client.cpp**

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <winsock2.h>

#include <ws2tcpip.h>

#include <stdlib.h>

#include <stdio.h>

#include <iostream>

#include <string>

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

using namespace std;

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int retVal = 0;

WSADATA wsaData;

WSAStartup(MAKEWORD(2, 2), &wsaData);

SOCKET SendRecvSocket = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP);

if (SendRecvSocket == INVALID\_SOCKET)

cout << "Создание сокета не удалось. Код ошибки: " << WSAGetLastError() << endl;

string ip;

unsigned short port;

cout << "ip> ";

cin >> ip;

cout << "port>";

cin >> port;

cin.ignore();

sockaddr\_in ServerAddr;

ServerAddr.sin\_family = AF\_INET;

ServerAddr.sin\_addr.s\_addr = inet\_addr(ip.c\_str());

ServerAddr.sin\_port = htons(port);

cout << "Соединение установлено. Введите данные:" << endl;

char pBuf[1024] = { '\0' };

gets\_s(pBuf);

// отправляем запрос на сервер

cout<< "Отправляем запрос" << endl;

retVal = sendto(SendRecvSocket, pBuf, strlen(pBuf), 0, (sockaddr\*)&ServerAddr, sizeof(ServerAddr));

if (retVal == SOCKET\_ERROR)

{

cout << "Невозможно отправить" << endl;

WSACleanup();

return 1;

}

// получаем результат

char szResponse[2048];

retVal = recvfrom(SendRecvSocket, szResponse, 1024, 0, 0, 0);;

if (retVal == SOCKET\_ERROR) {

cout << "Невозможно получить данные" << endl;

WSACleanup();

return 1;

}

char\* Resp = szResponse;

Resp[retVal] = 0;

cout << Resp;

//Завершение работы

closesocket(SendRecvSocket);

WSACleanup();

return 0;

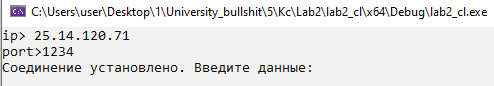
}

**Тестирование:**

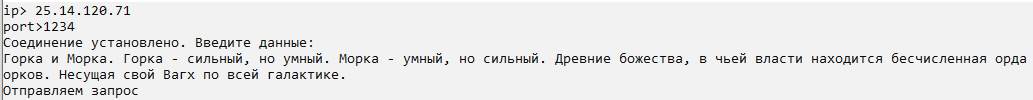
**Запуск сервера:**



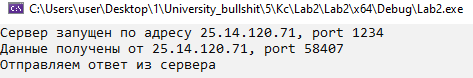
**Запуск клиента:**

****

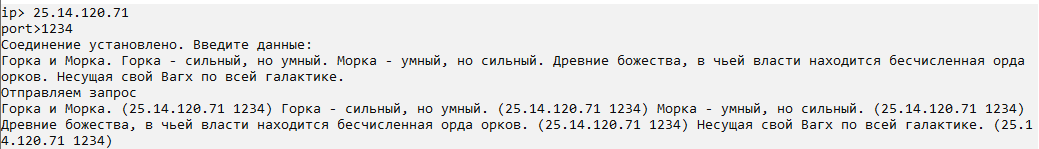
**Ввод данных:**

****

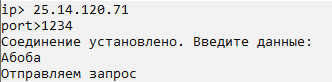
**Реакция сервера:**

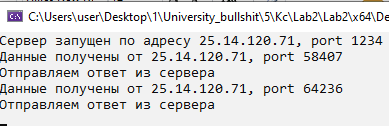
****

**Получение данных клиентом:**

****

**Новое соединение:**

****

****