Министерство Образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономической информатики

ОТЧЕТ

к лабораторной работе №3

на тему

**Создание параллельного многопоточного сервера с установлением логического соединения TCP**

по курсу

«Компьютерные сети»

Выполнил:                                   А. C. Летко

студент группы 377901

Проверил:                                     А. А. Бутов

Минск 2024

**Вариант 8**

1. На сервере хранится список разговоров на междугородней АТС. Каждая запись списка содержит следующую информацию о разговорах:

* дату разговора;
* код и название города;
* продолжительность разговора;
* тариф;
* номер телефона в этом городе;
* номер телефона абонента.

Таких записей должно быть не менее пяти.

Клиент посылает на сервер название города. Назад он получает суммарное время разговора с указанным городом.

**Пример кода для клиента:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#define WIN32\_LEAN\_AND\_MEAN

#include <iostream>

#include <winsock2.h>

#include <ws2tcpip.h>

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

using namespace std;

int main(void) {

WORD wVersionRequested = MAKEWORD(2, 2);

WSADATA wsaData;

if (WSAStartup(wVersionRequested, &wsaData) != 0) { return -1; }

while (true) {

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

sockaddr\_in dest\_addr;

dest\_addr.sin\_family = AF\_INET;

dest\_addr.sin\_port = htons(1280);

inet\_pton(AF\_INET, "127.0.0.1", &dest\_addr.sin\_addr.s\_addr);

connect(s, (sockaddr\*)&dest\_addr, sizeof(dest\_addr));

char buf[100];

cout << "Vvedite stroky(Pystaia stroke - vihod):";

cin.getline(buf, 100, '\n');

if (!strlen(buf)) break;

send(s, buf, strlen(buf) + 1, 0);

if (recv(s, buf, sizeof(buf), 0) != 0) {

cout << "Poluchenaya stroka:" << buf << endl;

}

closesocket(s);

}

WSACleanup();

return 0;

}

**Пример кода для сервира:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#define WIN32\_LEAN\_AND\_MEAN

#include <iostream>

#include <winsock2.h>

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

using namespace std;

struct LDTC {

char dateConversation[15], nameCity[15], tariff[15];

int code, callDuration, number, numberSub;

};

LDTC arr[5] = {

{"10.10.2024", "Minsk", "Bezlimitishe", 33, 678, 803566320, 33},

{"10.10.2024", "Brest", "VseVkl", 29, 123, 801536320, 29},

{"09.10.2024", "Minsk", "Interesno", 22, 430, 803234120, 22},

{"09.09.2024", "Brest", "Inet", 55, 1423, 804263221, 55},

{"08.09.2023", "Gantsevichi", "Velcom", 29, 1253, 802222222, 29}

};

DWORD WINAPI ThreadFunc(LPVOID client\_socket){

SOCKET s2 = ((SOCKET\*)client\_socket)[0];

char buf[100];

while (recv(s2, buf, sizeof(buf), 0)){

int counter = 0, length\_buf = strlen(buf);

for (int \_city = 0; \_city < 5; \_city++) {

int length\_city = strlen(arr[\_city].nameCity);

if (length\_buf != length\_city)

continue;

for (int \_letter = 0; \_letter < length\_city; \_letter++)

if (arr[\_city].nameCity[\_letter] != buf[\_letter])

break;

else if (\_letter + 1 == length\_city)

counter += arr[\_city].callDuration;

}

char line[11];

\_itoa(counter, line, 10);

int discharge = 0;

do {

counter /= 10; discharge++;

} while (counter);

line[discharge] = '\0';

send(s2, line, discharge + 1, 0);

}

closesocket(s2);

return 0;

}

int numcl = 0;

void print(){

if (numcl) printf("%d client connected\n", numcl);

else printf("No clients connected\n");

}

int main(void){

WORD wVersionRequested = MAKEWORD(2, 2);

WSADATA wsaData;

if (WSAStartup(wVersionRequested, &wsaData) != 0) { return -1; }

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

sockaddr\_in local\_addr;

local\_addr.sin\_family = AF\_INET;

local\_addr.sin\_port = htons(1280);

local\_addr.sin\_addr.s\_addr = 0;

bind(s, (sockaddr\*)&local\_addr, sizeof(local\_addr));

int c = listen(s, 5);

cout << "Server receive ready" << endl;

SOCKET client\_socket;

sockaddr\_in client\_addr;

int client\_addr\_size = sizeof(client\_addr);

while ((client\_socket = accept(s, (sockaddr\*)&client\_addr, &client\_addr\_size))){

numcl++;

print();

DWORD thID;

CreateThread(NULL, NULL, ThreadFunc, &client\_socket, NULL, &thID);

}

return 0;

}

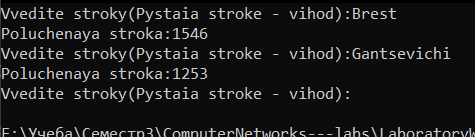


Рис.1 − Пример выполнения работы клиента №1

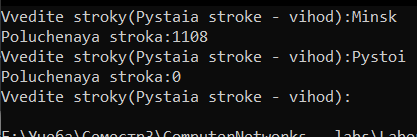


Рис.2 − Пример выполнения работы клиента №1

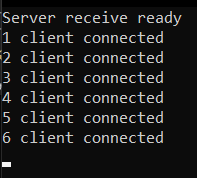


Рис. 3 − Пример выполнения работы сервера