МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по практической работе №2**

**«**Разработка сетевого интерфейса системы управления базой данных**»**

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ» *доцент кафедры ЗИ*

*Бурлаков И. Е. Архипова А. Б.*

«\_\_» \_\_\_\_\_\_ 2024 г.«\_\_\_» \_\_\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

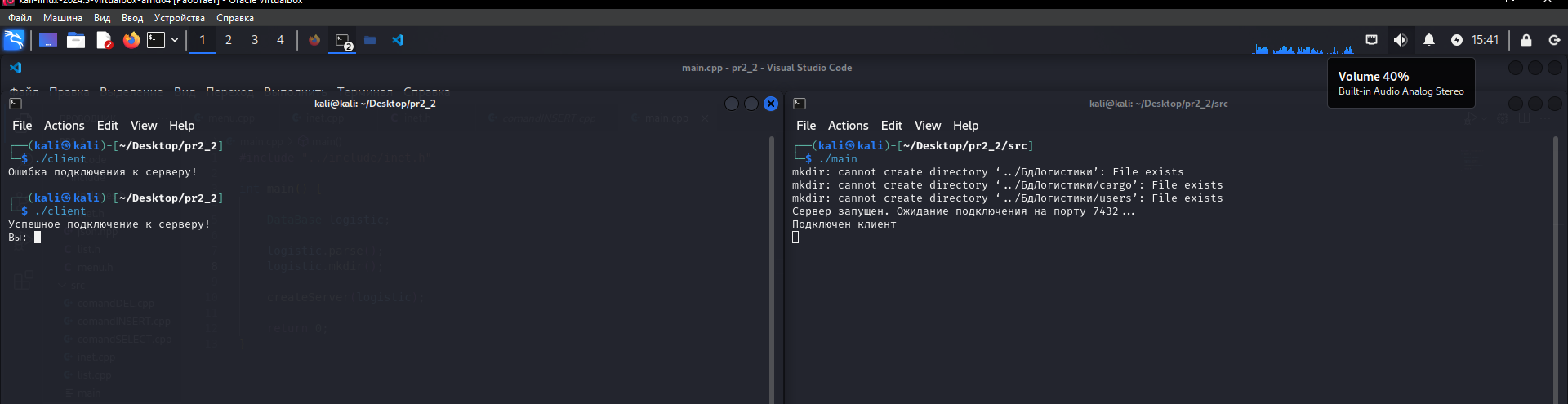
Новосибирск 2024

**Задание:** Требуется реализовать сетевой интерфейс для СУБД из практики 1.

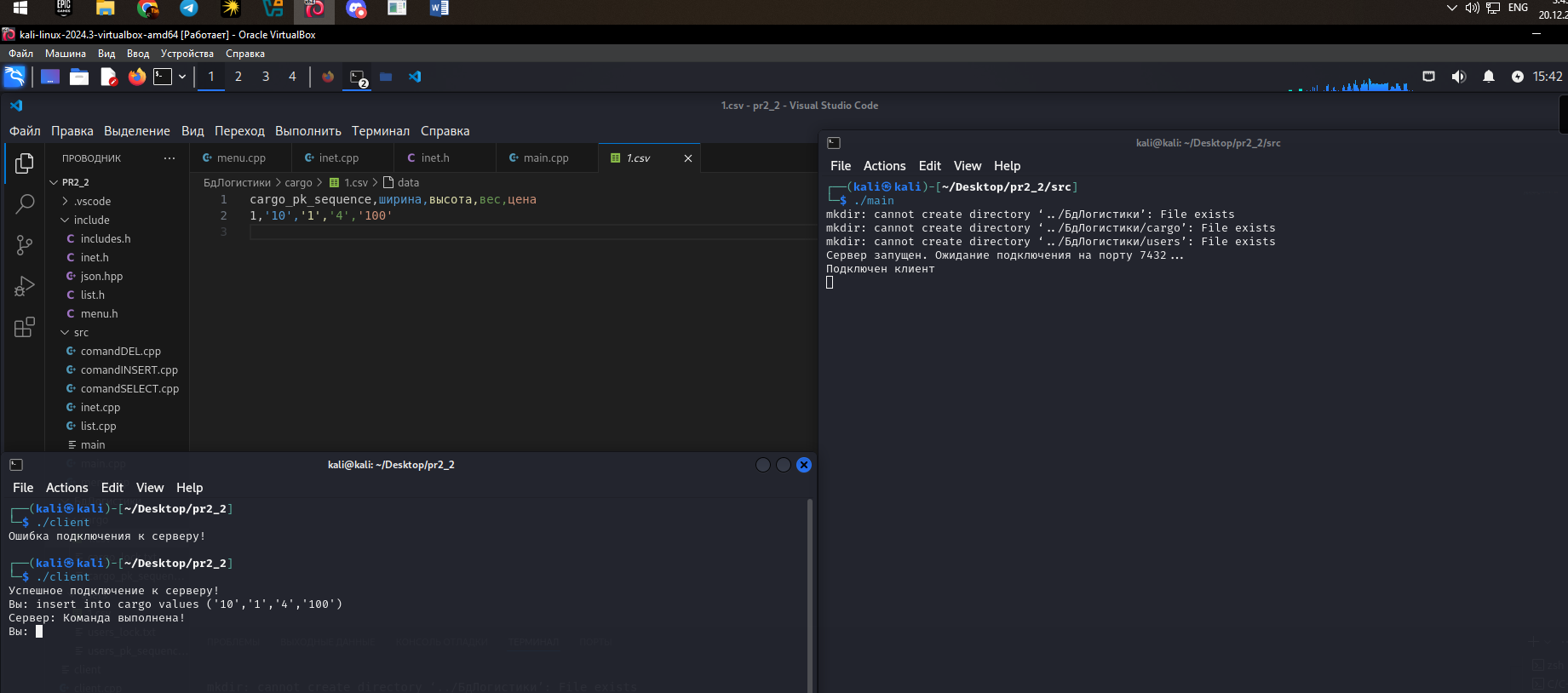
При запуске приложения из первой практики СУБД должна ожидать соединение по протоколу tcp на порту 7432. При подключении требуется обработать запрос либо в отдельном потоке, либо в отдельном процессе, либо асинхронной задачей и отдать результат в ответ на запрос.

Также необходимо позаботится о блокировках на структуре БД если используются потоки или процессы.

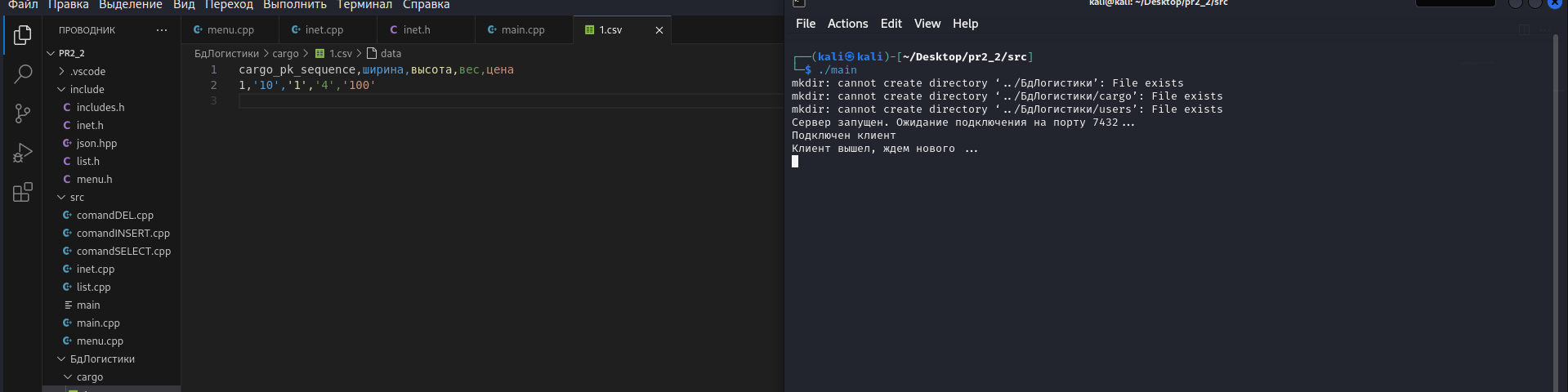
Подключение usera к серверу



Работа сервера на запросы usera



Отключение usera от сервера



Сделал файл inet.h

#include "../include/inet.h"

void createServer(DataBase& logistic) {

createSocket();

connectClient(logistic);

}

void createSocket() {

// создание TCP-сокета

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

if (server < 0) {

cout << "При создания сокета что-то пошло не так" << endl;

exit(-1);

}

// Настройка опции сокета

if (setsockopt(server, SOL\_SOCKET, SO\_REUSEADDR, &opt, sizeof(opt))) {

cout << "При установки опций сокета что-то пошло не так" << endl;

exit(-1);

}

// установка адреса и порта

server\_address.sin\_port = htons(PORT); // преобразование порта в сетевой порядок байт

server\_address.sin\_family = AF\_INET; // семейство адресов (IPv 4)

server\_address.sin\_addr.s\_addr = INADDR\_ANY; // привязка к любому адресу

// привязка сокета к адресу

if ((bind(server, (struct sockaddr\*)&server\_address, sizeof(server\_address))) < 0) {

cout << "при привязки сокета что-то пошло не так" << endl;

exit(-1);

}

}

void connectClient(DataBase& logistic) {

// прослушивание входящих соединений

if (listen(server, 2) < 0) {

cout << "При прослушивание что-то пошло не так!" << endl;

exit(-1);

}

cout << "Сервер запущен. Ожидание подключения на порту " << PORT << "..." << endl;

// принятие входящего соединения

while (true) {

new\_socket = accept(server, (struct sockaddr\*)&server\_address, (socklen\_t\*)&addrlen);

if (new\_socket < 0) {

cout << "При принятия соединения что-то пошло не так" << endl;

exit(-1);

}

cout << "Подключен клиент" << endl;

static mutex mx;

thread(procOfReq, new\_socket, ref(logistic), ref(mx)).detach(); // обработка запроса клиента в отдельном потоке

}

}

void procOfReq(int client\_socket, DataBase& logistic, mutex& mx) {

while (true) {

char buffer[1024] = {0};

int check = read(client\_socket, buffer, 1024);

if (check <= 0 ) { // клиент отключился

cout << "Клиент вышел, ждем нового ..." << endl;

break;

}

string conditions(buffer);

lock\_guard<mutex> lock(mx);

this\_thread::sleep\_for(chrono::milliseconds(5000));

string message = logistic.checkcommand(conditions);

// отправка ответа клиенту

send(client\_socket, message.c\_str(), message.size(), 0);

}

close(client\_socket);

}

Inet.cpp

#include "../include/inet.h"

void createServer(DataBase& logistic) {

createSocket();

connectClient(logistic);

}

void createSocket() {

// создание TCP-сокета

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

if (server < 0) {

cout << "При создания сокета что-то пошло не так" << endl;

exit(-1);

}

// Настройка опции сокета

if (setsockopt(server, SOL\_SOCKET, SO\_REUSEADDR, &opt, sizeof(opt))) {

cout << "При установки опций сокета что-то пошло не так" << endl;

exit(-1);

}

// установка адреса и порта

server\_address.sin\_port = htons(PORT); // преобразование порта в сетевой порядок байт

server\_address.sin\_family = AF\_INET; // семейство адресов (IPv 4)

server\_address.sin\_addr.s\_addr = INADDR\_ANY; // привязка к любому адресу

// привязка сокета к адресу

if ((bind(server, (struct sockaddr\*)&server\_address, sizeof(server\_address))) < 0) {

cout << "при привязки сокета что-то пошло не так" << endl;

exit(-1);

}

}

void connectClient(DataBase& logistic) {

// прослушивание входящих соединений

if (listen(server, 2) < 0) {

cout << "При прослушивание что-то пошло не так!" << endl;

exit(-1);

}

cout << "Сервер запущен. Ожидание подключения на порту " << PORT << "..." << endl;

// принятие входящего соединения

while (true) {

new\_socket = accept(server, (struct sockaddr\*)&server\_address, (socklen\_t\*)&addrlen);

if (new\_socket < 0) {

cout << "При принятия соединения что-то пошло не так" << endl;

exit(-1);

}

cout << "Подключен клиент" << endl;

static mutex mx;

thread(procOfReq, new\_socket, ref(logistic), ref(mx)).detach(); // обработка запроса клиента в отдельном потоке

}

}

void procOfReq(int client\_socket, DataBase& logistic, mutex& mx) {

while (true) {

char buffer[1024] = {0};

int check = read(client\_socket, buffer, 1024);

if (check <= 0 ) { // клиент отключился

cout << "Клиент вышел, ждем нового ..." << endl;

break;

}

string conditions(buffer);

lock\_guard<mutex> lock(mx);

this\_thread::sleep\_for(chrono::milliseconds(5000));

string message = logistic.checkcommand(conditions);

// отправка ответа клиенту

send(client\_socket, message.c\_str(), message.size(), 0);

}

close(client\_socket);

}

# Вывод

В процессе выполнения практической работы №2 я изучил концепцию сетевого подключения и интернет-протоколы. Я также научился создавать свой собственный сетевой интерфейс.

**Ссылка на репозиторий:** [**https://github.com/Evlusha/pr2\_3s**](https://github.com/Evlusha/pr2_3s)