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*

*Клаус В. А. 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.

# Модули программы

## network.h:

#pragma once

#include "includes.h"

#include "menu.h"

#define PORT 7432

mutex mx;

int server, new\_socket; // идентификатор сокетов сервера и нового(для взаимодействия)

struct sockaddr\_in server\_address; // информация о адресе сервера

int addrlen = sizeof(server\_address);

int opt = 1; // переменная для настройки сокета

void createServer(DataBase& carshop);

void createSocket(); // ф-ия создания сокета сервера

void connectClient(DataBase& carshop); // ф-ия прослушивания и принятия входящих соединений

void procOfReq(int client\_socket, DataBase& carshop); // ф-ия обработки запроса от клиента

#include "../src/network.cpp"

## network.cpp:

#include "../include/network.h"

void createServer(DataBase& carshop) {

createSocket();

connectClient(carshop);

}

void createSocket() {

// создание TCP-сокета

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

if (server < 0) {

cout << "Ошибка создания сокета!" << endl;

exit(-1);

}

// Настройка опции сокета

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

cout << "Ошибка установки опции сокета" << 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 << "Ошибка привязки сокета!" << endl;

exit(-1);

}

}

void connectClient(DataBase& carshop) {

// прослушивание входящих соединений

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

cout << "Ошибка прослушивания!" << 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 << "Ошибка принятия соединения!" << endl;

exit(-1);

}

cout << "Клиент подключен" << endl;

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

}

}

void procOfReq(int client\_socket, DataBase& carshop) {

while (true) {

char buffer[1024] = {0};

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

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

cout << "Клиент отключился, ожидание нового подключения..." << endl;

break;

}

string conditions(buffer);

lock\_guard<mutex> lock(mx);

string message = carshop.checkcommand(conditions);

// отправка ответа клиенту

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

}

close(client\_socket);

}

## 

## main cpp:

#include "../include/network.h"

int main() {

DataBase carshop;

carshop.parse();

carshop.mkdir();

createServer(carshop);

return 0;

}

# Вывод

В ходе выполнения практической работы №2 я познакомился с понятием сетевого подключения, интернет протоколы. Научился реализовывать свой собственный сетевой интерфейс.

**Ссылка на репозиторий:** [нажмите](https://github.com/JamesSevil/2kurs-pract2)