|  |  |
| --- | --- |
|  | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» |

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Лабораторная работа № 2  
по дисциплине «Web-технологии»  
на тему «Сокеты»

Выполнил  
студент группы М3О-107БВ-24  
Коркина К.В.

Приняла

Масленникова Т.Ю.

Москва  
2025

**Оглавление**

[Задание №1 3](#_Toc196259285)

[Код клиента 3](#_Toc196259286)

[Лог файл Клиента 4](#_Toc196259287)

[Задание №2 5](#_Toc196259288)

[Код синхронного сервера 5](#_Toc196259289)

[Задание №3 8](#_Toc196259290)

[Код многопоточного сервера 8](#_Toc196259291)

[Задание №4 12](#_Toc196259292)

[Параметры подключения: 12](#_Toc196259293)

[Лог файл синхронного сервера 13](#_Toc196259294)

[Лог файл многопоточного сервера 13](#_Toc196259295)

[Отличия 13](#_Toc196259296)

# Задание №1

*Разработать программу клиент, работающий на сокетах на выбранном Вами языке программирования. Данный клиент должен хранить лог файл соединений. В лог файле хранится время подключения и адрес сервера, время отправки сообщения и само сообщение, время получения сообщения и само сообщение. При подключении к серверу клиент через определенный промежуток времени (задается разработчиком клиента) отправляет сообщение серверу: ФИО и группа студента, а также ждет от сервера. Данные о сервере необходимо вводить через файл конфигурации. В отчете привести код клиента и один тестовый лог файл.*

## Код клиента

*import socket*

*import json*

*import datetime*

*import sys*

*student\_name = "Korkina Ksenia Vladimirovna"*

*student\_group = "M3O-107BV-24"*

*def write\_to\_log(action, message=None):*

*current\_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")*

*with open("client\_log.txt", "a", encoding='utf-8') as log\_file:*

*if action == "connect":*

*log\_file.write(f"[{current\_time}] Подключение к серверу {server\_address}:{server\_port}\n")*

*elif action == "send":*

*log\_file.write(f"[{current\_time}] Отправлено сообщение: {message}\n")*

*elif action == "receive":*

*log\_file.write(f"[{current\_time}] Получено сообщение: {message}\n")*

*elif action == "disconnect":*

*log\_file.write(f"[{current\_time}] Отключение от сервера\n")*

*sys.stdout.reconfigure(encoding='utf-8')*

*try:*

*with open("client\_config.json", "r", encoding='utf-8') as config\_file:*

*config = json.load(config\_file)*

*server\_address = config["server\_address"]*

*server\_port = config["server\_port"]*

*except Exception as e:*

*print(f"Ошибка чтения конфигурационного файла: {e}")*

*exit()*

*client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)*

*client\_socket.settimeout(5)*

*try:*

*client\_socket.connect((server\_address, server\_port))*

*write\_to\_log("connect")*

*message = f"{student\_name} {student\_group}"*

*client\_socket.send(message.encode())*

*write\_to\_log("send", message)*

*response = client\_socket.recv(1024).decode('utf-8')*

*if not response:*

*write\_to\_log("disconnect", "Сервер закрыл соединение")*

*else:*

*write\_to\_log("receive", response)*

*finally:*

*client\_socket.close()*

*write\_to\_log("disconnect")*

*print("Клиент завершил работу")*

## Лог файл Клиента

[2025-04-21 00:08:41] Подключение к серверу 127.0.0.1:12345

[2025-04-21 00:08:41] Отправлено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[2025-04-21 00:08:43] Получено сообщение: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[2025-04-21 00:08:43] Отключение от сервера

# Задание №2

*Разработать программу сервер, которая ожидает строку от клиента. Через промежуток времени (моделирование работы сервера) он отправляет в ответ эту же строку, только зеркально отраженную и добавляет к сообщению ФИО студента, написавшего сервер. (пример ответа: Чиволвап Йирю Вотит. Сервер написан Титовым Ю.П. М3О-1ХХБВ-ХХ).*

*Сервер должен эмулировать работу путем временной задержки. Отключение клиентов от сервера совершать через определенное время работы клиента. Сервер и клиент должны быть написаны на различных языках программирования.*

## Код синхронного сервера

*#include <iostream>*

*#include <fstream>*

*#include <string>*

*#include <chrono>*

*#include <ctime>*

*#include <winsock2.h>*

*#include <io.h>*

*#include <fcntl.h>*

*#pragma comment(lib, "ws2\_32.lib")*

*std::string reverseString(const std::string& str) {*

*return std::string(str.rbegin(), str.rend());*

*}*

*std::string getCurrentTime() {*

*auto now = std::chrono::system\_clock::now();*

*std::time\_t now\_time = std::chrono::system\_clock::to\_time\_t(now);*

*char buf[100];*

*ctime\_s(buf, sizeof(buf), &now\_time);*

*std::string timeStr(buf);*

*timeStr.erase(timeStr.find\_last\_not\_of("\n") + 1);*

*return timeStr;*

*}*

*void writeToLog(const std::string& message, std::ofstream& logFile) {*

*logFile << "[" << getCurrentTime() << "] " << message << std::endl;*

*}*

*int main() {*

*SetConsoleOutputCP(CP\_UTF8);*

*SetConsoleCP(CP\_UTF8);*

*WSADATA wsaData;*

*if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {*

*std::cerr << "Ошибка инициализации Winsock" << std::endl;*

*return 1;*

*}*

*SOCKET serverSocket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);*

*if (serverSocket == INVALID\_SOCKET) {*

*std::cerr << "Ошибка создания сокета" << std::endl;*

*WSACleanup();*

*return 1;*

*}*

*sockaddr\_in serverAddr;*

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

*serverAddr.sin\_addr.s\_addr = INADDR\_ANY;*

*serverAddr.sin\_port = htons(12345);*

*if (bind(serverSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) {*

*std::cerr << "Ошибка привязки сокета" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*if (listen(serverSocket, 1) == SOCKET\_ERROR) {*

*std::cerr << "Ошибка перехода в режим прослушивания" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*std::ofstream logFile("server\_log.txt", std::ios::app);*

*if (!logFile.is\_open()) {*

*std::cerr << "Ошибка открытия лог-файла" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*writeToLog("Сервер запущен", logFile);*

*std::cout << "Сервер ожидает подключения..." << std::endl;*

*while (true) {*

*sockaddr\_in clientAddr;*

*int clientAddrSize = sizeof(clientAddr);*

*SOCKET clientSocket = accept(serverSocket, (sockaddr\*)&clientAddr, &clientAddrSize);*

*if (clientSocket == INVALID\_SOCKET) {*

*writeToLog("Ошибка принятия подключения", logFile);*

*closesocket(serverSocket);*

*WSACleanup();*

*logFile.close();*

*return 1;*

*}*

*char clientIP[16];*

*const char\* ip = inet\_ntoa(clientAddr.sin\_addr);*

*if (ip != nullptr) {*

*strncpy(clientIP, ip, sizeof(clientIP));*

*writeToLog("Подключен клиент: " + std::string(clientIP), logFile);*

*} else {*

*writeToLog("Подключен клиент (не удалось получить IP)", logFile);*

*}*

*char buffer[1024];*

*int bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0);*

*if (bytesReceived == SOCKET\_ERROR) {*

*writeToLog("Ошибка получения данных", logFile);*

*}*

*buffer[bytesReceived] = '\0';*

*std::string clientMessage(buffer);*

*writeToLog("Получено сообщение: " + clientMessage, logFile);*

*Sleep(2000);*

*std::string reversedMessage = reverseString(clientMessage);*

*std::string response = reversedMessage + ". Server is written by Korkina K.V. M3O-107BV-24";*

*send(clientSocket, response.c\_str(), response.size(), 0);*

*writeToLog("Отправлен ответ: " + response, logFile);*

*Sleep(5000);*

*closesocket(clientSocket);*

*writeToLog("Соединение с клиентом закрыто", logFile);*

*}*

*closesocket(serverSocket);*

*WSACleanup();*

*logFile.close();*

*return 0; }*

# Задание №3

*Доработать программу сервер для выполнения асинхронных запросов от клиентов. Чётные варианты реализуют многопоточный асинхронный сервер, нечетные – многопроцессный. Номер варианта определяется номером по списку.*

## Код многопоточного сервера

*#include <iostream>*

*#include <fstream>*

*#include <string>*

*#include <chrono>*

*#include <ctime>*

*#include <winsock2.h>*

*#include <thread>*

*#include <vector>*

*#include <io.h>*

*#include <fcntl.h>*

*#pragma comment(lib, "ws2\_32.lib")*

*std::string reverseString(const std::string& str) {*

*return std::string(str.rbegin(), str.rend());*

*}*

*std::string getCurrentTime() {*

*auto now = std::chrono::system\_clock::now();*

*std::time\_t now\_time = std::chrono::system\_clock::to\_time\_t(now);*

*char buf[100];*

*ctime\_s(buf, sizeof(buf), &now\_time);*

*std::string timeStr(buf);*

*timeStr.erase(timeStr.find\_last\_not\_of("\n") + 1);*

*return timeStr;*

*}*

*void writeToLog(const std::string& message, std::ofstream& logFile) {*

*logFile << "[" << getCurrentTime() << "] " << message << std::endl;*

*std::cout << "[" << getCurrentTime() << "] " << message << std::endl;*

*}*

*void handleClient(SOCKET clientSocket, sockaddr\_in clientAddr, std::ofstream& logFile) {*

*char clientIP[16];*

*const char\* ip = inet\_ntoa(clientAddr.sin\_addr);*

*if (ip != nullptr) {*

*strncpy(clientIP, ip, sizeof(clientIP));*

*writeToLog("Подключен клиент: " + std::string(clientIP), logFile);*

*} else {*

*writeToLog("Подключен клиент (не удалось получить IP)", logFile);*

*}*

*char buffer[1024];*

*int bytesReceived = recv(clientSocket, buffer, sizeof(buffer), 0);*

*if (bytesReceived == SOCKET\_ERROR) {*

*writeToLog("Ошибка получения данных", logFile);*

*} else {*

*buffer[bytesReceived] = '\0';*

*std::string clientMessage(buffer);*

*writeToLog("Получено сообщение: " + clientMessage, logFile);*

*Sleep(2000); // Имитация обработки*

*std::string reversedMessage = reverseString(clientMessage);*

*std::string response = reversedMessage + ". Server is written by Korkina K.V. M3O-107BV-24";*

*send(clientSocket, response.c\_str(), response.size(), 0);*

*writeToLog("Отправлен ответ: " + response, logFile);*

*}*

*shutdown(clientSocket, SD\_SEND);*

*closesocket(clientSocket);*

*writeToLog("Соединение с клиентом закрыто", logFile);*

*}*

*int main() {*

*SetConsoleOutputCP(CP\_UTF8);*

*SetConsoleCP(CP\_UTF8);*

*WSADATA wsaData;*

*if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {*

*std::cerr << "Ошибка инициализации Winsock" << std::endl;*

*return 1;*

*}*

*SOCKET serverSocket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);*

*if (serverSocket == INVALID\_SOCKET) {*

*std::cerr << "Ошибка создания сокета" << std::endl;*

*WSACleanup();*

*return 1;*

*}*

*sockaddr\_in serverAddr;*

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

*serverAddr.sin\_addr.s\_addr = INADDR\_ANY;*

*serverAddr.sin\_port = htons(12345);*

*if (bind(serverSocket, (sockaddr\*)&serverAddr, sizeof(serverAddr)) == SOCKET\_ERROR) {*

*std::cerr << "Ошибка привязки сокета" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*if (listen(serverSocket, SOMAXCONN) == SOCKET\_ERROR) {*

*std::cerr << "Ошибка перехода в режим прослушивания" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*std::ofstream logFile("server\_log.txt", std::ios::app);*

*if (!logFile.is\_open()) {*

*std::cerr << "Ошибка открытия лог-файла" << std::endl;*

*closesocket(serverSocket);*

*WSACleanup();*

*return 1;*

*}*

*writeToLog("Сервер запущен", logFile);*

*std::cout << "Сервер ожидает подключения..." << std::endl;*

*std::vector<std::thread> clientThreads;*

*while (true) {*

*sockaddr\_in clientAddr;*

*int clientAddrSize = sizeof(clientAddr);*

*SOCKET clientSocket = accept(serverSocket, (sockaddr\*)&clientAddr, &clientAddrSize);*

*if (clientSocket == INVALID\_SOCKET) {*

*writeToLog("Ошибка принятия подключения", logFile);*

*continue;*

*}*

*// Создаем новый поток для обработки клиента*

*clientThreads.emplace\_back([clientSocket, clientAddr, &logFile]() {*

*handleClient(clientSocket, clientAddr, logFile);*

*});*

*}*

*// Ожидаем завершения всех потоков (на практике сюда никогда не попадем)*

*for (auto& thread : clientThreads) {*

*if (thread.joinable()) {*

*thread.join();*

*}*

*}*

*closesocket(serverSocket);*

*WSACleanup();*

*writeToLog("Сервер завершил работу", logFile);*

*logFile.close();*

*return 0;*

*}*

# Задание №4

*Запустить 5 клиентов в сети кафедры и 1 сервер (5 студентов). В отчете указать настройки для подключения и лог файлы клиентов и серверов. Проводить тестирование одновременного подключения, т.е. второй, третий и т.д. пользователь подключается к серверу пока первый пользователь еще не получил ответ. Желательно провести тестирование как на однопоточном, так и на асинхронном сервере, описать отличия.*

## Параметры подключения:

IPv4 сервера: **192.168.1.37**

PORT сервера: **12345**

## Лог файл синхронного сервера

[Mon Apr 21 19:36:39 2025] Сервер запущен

[Mon Apr 21 19:36:44 2025] Подключен клиент: 127.0.0.1

[Mon Apr 21 19:36:44 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Mon Apr 21 19:36:49 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Mon Apr 21 19:36:49 2025] Соединение с клиентом закрыто

## Лог файл многопоточного сервера

[Wed Apr 23 09:37:57 2025] Подключен клиент: 172.20.10.2

[Wed Apr 23 09:37:57 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Wed Apr 23 09:37:59 2025] Подключен клиент: 172.20.10.12

[Wed Apr 23 09:37:59 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Wed Apr 23 09:37:59 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:37:59 2025] Соединение с клиентом закрыто

[Wed Apr 23 09:37:59 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:37:59 2025] Соединение с клиентом закрыто

[Wed Apr 23 09:38:01 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:38:01 2025] Соединение с клиентом закрыто

[Wed Apr 23 09:40:53 2025] Подключен клиент: 172.20.10.11

[Wed Apr 23 09:40:53 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Wed Apr 23 09:40:55 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:40:55 2025] Соединение с клиентом закрыто

[Wed Apr 23 09:40:58 2025] Подключен клиент: 172.20.10.12

[Wed Apr 23 09:40:58 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Wed Apr 23 09:41:00 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:41:00 2025] Соединение с клиентом закрыто

[Wed Apr 23 09:41:03 2025] Подключен клиент: 172.20.10.2

[Wed Apr 23 09:41:03 2025] Получено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[Wed Apr 23 09:41:05 2025] Отправлен ответ: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[Wed Apr 23 09:41:05 2025] Соединение с клиентом закрыто

# Лог-файл клиента

[2025-04-23 09:38:00] Подключение к серверу 172.20.10.2:12345

[2025-04-23 09:38:00] Отправлено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[2025-04-23 09:38:02] Получено сообщение: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[2025-04-23 09:38:02] Отключение от сервера

[2025-04-23 09:40:59] Подключение к серверу 172.20.10.2:12345

[2025-04-23 09:40:59] Отправлено сообщение: Korkina Ksenia Vladimirovna M3O-107BV-24

[2025-04-23 09:41:02] Получено сообщение: 42-VB701-O3M anvorimidalV ainesK anikroK. Server is written by Korkina K.V. M3O-107BV-24

[2025-04-23 09:41:02] Отключение от сервера

# Отличия

При подключении нескольких клиентов к синхронному серверу все клиенты обрабатываются последовательно. Следовательно, пока сервер обрабатывает первого клиента пятый будет простаивать в ожидании ответа от сервера. В многопоточным сервере такой проблемы нет, все пять клиентов запускаются одновременно, что помогает правильно распределить время работы системы клиент-сервер.