Белорусский Национальный Технический Университет

Кафедра «Инновационные процессы»

Лабораторная работа № 5

 «Клиент-серверное сокетное соединение в сети»

|  |
| --- |
| Выполнил: студент группы 42221  Мельник К.А.  Проверил:  Станкевич С.Н. |

МИНСК 2023

Цель работы:

Изучить механизм сокетного сетевого взаимодействия в Linux.

Задания:

**Задание 1:**

Напишите программы с клиент-серверной архитектурой по следующим требованиям:

1. Обменяться числами. Сервер возводит получаемое от клиента число в квадрат и возвращает ответ клиенту.

Листинг кода:

Client.c:

#include <stdio.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <unistd.h>

#define PORT 12345

void client(int number) {

int client\_socket;

struct sockaddr\_in server\_addr;

// Создание сокета

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

// Настройка адреса сервера

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

server\_addr.sin\_addr.s\_addr = INADDR\_ANY;

server\_addr.sin\_port = htons(PORT);

// Подключение к серверу connect(client\_socket, (struct sockaddr\*)&server\_addr, sizeof(server\_addr));

// Отправка данных серверу

char buffer[1024];

sprintf(buffer, "%d", number);

send(client\_socket, buffer, strlen(buffer), 0);

// Получение результата от сервера

recv(client\_socket, buffer, sizeof(buffer), 0);

printf("Результат для числа %d: %s\n", number, buffer);

// Закрытие сокета

close(client\_socket);

}

int main() {

client(160);

return 0;

}

Server.c:

#include <stdio.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <unistd.h>

#define PORT 12345

int main() {

int server\_socket, client\_socket;

struct sockaddr\_in server\_addr, client\_addr;

socklen\_t client\_addr\_len = sizeof(client\_addr);

// Создание сокета

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

// Настройка адреса сервера

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

server\_addr.sin\_addr.s\_addr = INADDR\_ANY;

server\_addr.sin\_port = htons(PORT);

// Привязка сокета к адресу

bind(server\_socket, (struct sockaddr\*)&server\_addr, sizeof(server\_addr));

// Ожидание подключения клиента

listen(server\_socket, 1);

printf("Сервер запущен. Ожидание подключения клиента...\n");

// Принятие подключения

client\_socket = accept(server\_socket, (struct sockaddr\*)&client\_addr, &client\_addr\_len);

printf("Подключен клиент\n");

while (1) {

char buffer[2048];

int num, result;

// Получение данных от клиента

recv(client\_socket, buffer, sizeof(buffer), 0);

// Преобразование строки в число

num = atoi(buffer);

// Вычисление квадрата числа

result = num\*num;

// Отправка результата клиенту

sprintf(buffer, "%d", result);

send(client\_socket, buffer, strlen(buffer), 0);

}

// Закрытие сокетов

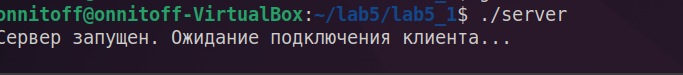
close(client\_socket);

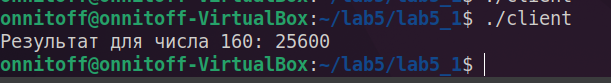
close(server\_socket);

return 0;

}

Работоспособность:





2. Создать два клиента на один и тот же сокет. Используют один и тот же сокет. Первый клиент получает результат счета от 1 до 10, второй – результат счета от 100 до 110 (подумайте, как распознать, какой клиент обратился к серверу).

3. Создать два клиента на один и тот же сервер. Сокеты должны быть разными (именованное соединение в каждом случае должно быть различным). Первый клиент получает результат счета от 1 до 10, второй – от 100 до 110.

**Задание 2:**

Интернет сокеты Напишите программу, которая читает главную страницу из какого-нибудь Web-сервера. Запуск программы должен производиться с консоли.

**Задание 3:**

Дейтаграммные сокеты Напишите программу с клиент-серверной архитектурой для передачи данных через дейтаграммные сокеты. Количество клиентов должно соответствовать количеству членов вашей команды, но не менее двух. Семейства протоколов и способы взаимодействия сокетов-клиентов должны отличаться. Запустите программы в разных терминалах и определите какие программыклиенты получили данные. Объясните полученный результат.