Федеральное государственное бюджетное образовательное учреждение высшего образования «Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №8

Сетевое приложение

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверил: Путевская Ирина Валерьевна

Чебоксары, 2025

Серверное приложение, реализуюущее чат:  
const net = require('net');

const clients = []; // Массив подключённых клиентов

const server = net.createServer((socket) => {

console.log('Новый клиент подключился');

clients.push(socket);

// Отправляем приветствие

socket.write('Добро пожаловать в чат! Введите имя: ');

let username = '';

socket.on('data', (data) => {

const message = data.toString().trim();

// Если имя не задано, сохраняем его

if (!username) {

username = message;

socket.write(`Привет, ${username}! Теперь ты в чате.\n`);

broadcast(`${username} присоединился к чату.`, socket);

return;

}

// Рассылаем сообщение всем, кроме отправителя

broadcast(`${username}: ${message}`, socket);

});

// Обработка отключения клиента

socket.on('end', () => {

console.log(`${username} отключился`);

clients.splice(clients.indexOf(socket), 1);

broadcast(`${username} покинул чат.`, socket);

});

// Обработка ошибок

socket.on('error', (err) => {

console.log('Ошибка:', err.message);

});

});

// Функция рассылки сообщений всем клиентам, кроме отправителя

function broadcast(message, sender) {

clients.forEach(client => {

if (client !== sender) {

client.write(message + '\n');

}

});

console.log(message);

}

// Запуск сервера

const PORT = 3000;

const HOST = '127.0.0.1';

server.listen(PORT, HOST, () => {

console.log(`Сервер чата запущен на ${HOST}:${PORT}`);

});

net - встроенный модуль в Node.js для создания сетевых серверов и клиентов.

Создание сервера происходит через команду net.createServer(socket), для хранения клиентов используется массив clients.

Этапы работы:

1. Подключение клиента
2. Сервер запрашивает имя пользователя
3. Клиент вводит имя
4. Сервер закидывает имя в массив
5. Сервер начинает принимать сообщения

В коде обрабатываются события получения данных от клиента, отключение клиента и обработка ошибок.

Рассылка сообщений происходит через функцию broadcast.

Таким образом, при подключении клиента создается новый сокет, он представляет двусторонний канал связи.

Клиентское приложение:

const net = require('net');

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

const client = net.createConnection({ port: 3000, host: '127.0.0.1' }, () => {

console.log('Подключено к серверу чата!');

});

client.on('data', (data) => {

console.log(data.toString());

});

client.on('end', () => {

console.log('Отключено от сервера');

process.exit();

});

// Чтение ввода пользователя и отправка на сервер

rl.on('line', (input) => {

client.write(input);

});

Для создания клиента используется тот же net и readline для чтения коносли.

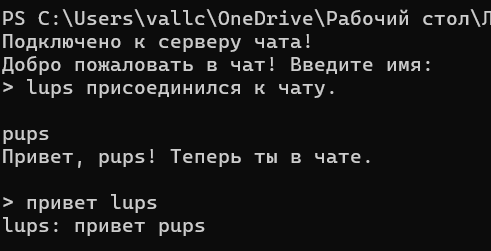
Подключение к серверу осуществляется через метод createConnection, в параметрах порт и ip адрес сервера.

Через метод client.on(‘data’) осуществляется получение данных от сервера

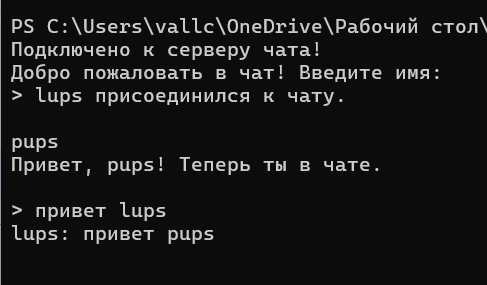
rl.on отвечает за обработку пользовательского ввода

Пример работы:

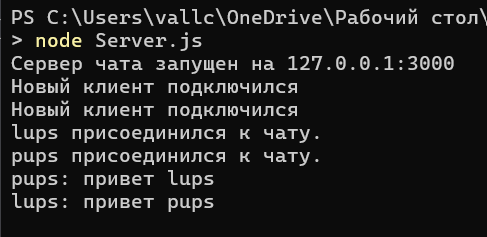
1 клиент:



2 клиент:



сервер:



Ответы на вопросы:

1. Какая структура кода сетевого приложения

Одна из распространённых структур кода сетевого приложения — клиент-серверная архитектура. В ней клиенты запрашивают ресурсы с сервера, а тот отправляет обратно ответы на запросы.

1. Как реализовать разделение сообщений по протокол обмена?

В коде добавляется спец-символ, обозначающий начало новой строки

1. Как создать клиент-серверное сообщение?

Для создания клиент-серверного сообщения используется сокет и его экземпляр.

1. Как создать распределенное сообщение?

Для создания распределённых сообщений используют брокеры сообщений — программные компоненты, которые служат посредниками между различными компонентами распределённой системы.

1. Как реализовать аутентификацию и авторизацию?

Можно сделать все по токенам. Например, при аутентификации выдается токен, и можно по опеределенному токену давать разрешение на переход в определенные вкладки.

1. Как защитить передачу данных по сети?

Шифование данных, хэширование.