

## Серверная часть.

### 1. Инициализация и создание сокетов

```
int listening = socket(AF_INET, SOCK_STREAM, 0);
```

AF\_INET – использование IPv4

SOCK\_STREAM и 0 – автоматический выбор TCP протокола

### 2. Настройка адреса сервера

```
hint.sin_port = htons(54000); – преобразование порта в сетевой порядок байт.
```

hint.sin\_addr.s\_addr = INADDR\_ANY; – слушать все сетевых интерфейс

### 3. Привязка и переход в режим ожидания

```
bind(listening, (sockaddr*)&hint, sizeof(hint)) – привязка сокета к указанному адресу и порту
```

listen(listening, SOMAXCONN) – перевести сокет в режим ожидания подключения

### 4. Принять подключение

```
int clientSocket = accept(listening, (sockaddr*)&client, &clientSize); – блокирует выполнение пока не поступило новое подключение, возвращает файловый дескриптор для общения с конкретным клиентом
```

### 5. Цикл обработки сообщений

```
while (true) {
```

```
    int bytesReceived = recv(clientSocket, buf, 4096, 0); – получение сообщения от клиента.
```

```
    send(clientSocket, buf, bytesReceived, 0); – отправка сообщения клиенту обратно.
```

```
}
```

## Клиентская часть

### 1. Подключение к серверу

```
int sock = socket(AF_INET, SOCK_STREAM, 0); – создание сокета
```

```
connect(sock, (sockaddr*)&hint, sizeof(hint)) – подключение к сокету
```

## 2. Общение с сервером

```
do
{
    cout << "> ";
    getline(cin, userInput);
    if (userInput.size() > 0)
    {
        int sendResult = send(sock, userInput.c_str(), userInput.size(), 0); –
        отправка сообщения на сервер
        int bytesReceived = recv(sock, buf, 4096, 0); – получение ответа от
        сервера
    }
} while(userInput.size() > 0);
```

При отправке пустого сообщения клиент разрывает соединение и закрывает сокет  
close(sock);

Вывод со стороны сервера:

```
λ build/server
localhost connected on port 50348
Hello
echo
Client disconnected
```

Вывод со стороны клиента:

```
λ build/client
24
Connected to server. Type messages (empty line to quit):
> Hello
SERVER> Hello
> echo
SERVER> echo
>
↙
```