

## Задание 1

### Необходимые знания

1. Компиляция программ на языке C.
2. Аргументы командной строки.
3. Протокол TCP.
4. Протокол UDP.

В этой лабораторной работе вам предстоит потрогать два клиент-серверных приложения. Первое использует проткол TCP, второй UDP. Вам необходимо:

- Скомпилировать все четыре программы.
- Вынести все константы (объявленные через `#define`) в аргументы командной строки.
- Скомпилировать оба приложения через `makefile`.

## Протокол TCP (Transmission Control Protocol)

### Что это?

TCP — это **протокол транспортного уровня**, обеспечивающий **надежную передачу данных** между двумя компьютерами.

### Основные свойства:

- **Установление соединения** (т.н. "рукопожатие" — three-way handshake).
- **Гарантия доставки** данных.
- **Контроль целостности** — проверка ошибок.
- **Сохранение порядка** пакетов.
- **Повторная передача** утерянных данных.

## Протокол UDP (User Datagram Protocol)

### Что это?

UDP — это **простой протокол транспортного уровня**, который **не гарантирует доставку** данных.

### Основные свойства:

- **Без установления соединения** (нет "рукопожатия").
- **Нет гарантии доставки** или порядка.
- **Быстрая передача** данных (меньше задержек).
- **Нет контроля ошибок** на уровне транспорта.

## Задание 2

Ответить на следующие вопросы:

1. Что делают оба приложения?
2. Что произойдет, если tcpclient отправит сообщение незапущенному серверу?
3. Что произойдет, если udpclient отправит сообщение незапущенному серверу?
4. Что произойдет, если tcpclient отвалится во время работы с сервером?
5. Что произойдет, если udpclient отвалится во время работы с сервером?
6. Что произойдет, если udpclient отправит сообщение на несуществующий / выключенный сервер?
7. Что произойдет, если tcpclient отправит сообщение на несуществующий / выключенный сервер?
8. В чем отличия UDP и TCP протоколов?

tcp

```
@ccurecc →/workspaces/os_lab_2019/lab7/src (master) $ ./tcpserver 12345
Connection established
Привет!
Как дела?
□

@ccurecc →/workspaces/os_lab_2019/lab7/src (master) $ ./tcpclient 127.0.0.1 12345
Input message to send
Привет!
Как дела?
```

Udp

```
@ccurecc →/workspaces/os_lab_2019/lab7/src (master) $ ./udpserver 12346
SERVER starts...
REQUEST Привет!
FROM 127.0.0.1 : 59126
REQUEST Как дела?
FROM 127.0.0.1 : 59126
□

@ccurecc →/workspaces/os_lab_2019 (master) $ cd lab7/src
@ccurecc →/workspaces/os_lab_2019/lab7/src (master) $ ./udpclient 127.0.0.1 12346
Enter string
Привет!
REPLY FROM SERVER= Привет!

Как дела?
REPLY FROM SERVER= Как дела?
```

### 1. Что делают оба приложения?

**TCP клиент и сервер:**

- **TCP-сервер** слушает на заданном порту, ожидая подключения клиентов. Когда клиент подключается, сервер принимает соединение и может получать данные от клиента, которые затем выводятся на экран.

- **TCP-клиент** устанавливает соединение с сервером по указанному IP-адресу и порту, отправляет сообщения, введенные пользователем, и может отправлять их серверу.

#### **UDP клиент и сервер:**

- **UDP-сервер** слушает на заданном порту и ждет получения сообщений от клиентов. Когда сервер получает сообщение, он выводит его на экран и отправляет обратно клиенту (эхо-сервер).

- **UDP-клиент** отправляет сообщения на сервер по указанному IP-адресу и порту, а затем ожидает ответа от сервера.

#### **2. Что произойдет, если tcpclient отправит сообщение незапущенному серверу?**

- **TCP-клиент** не сможет установить соединение и получит ошибку при вызове функции `connect()`. Сообщение об ошибке будет выведено на экран, и программа завершит выполнение.

#### **3. Что произойдет, если udpclient отправит сообщение незапущенному серверу?**

- **UDP-клиент** успешно отправит сообщение, но не получит ответа от сервера, так как сервер не запущен. В этом случае программа продолжит работать, и пользователь увидит сообщение о том, что отправка прошла успешно, но не получит ответа.

#### **4. Что произойдет, если tcpclient отвалится во время работы с сервером?**

- Если **TCP-клиент** отключится (например, из-за закрытия окна или вызова `exit()`), сервер получит ошибку на чтении данных от клиента, и соединение будет закрыто. Сервер продолжит работать, ожидая следующих подключений.

#### **5. Что произойдет, если udpclient отвалится во время работы с сервером?**

- Если **UDP-клиент** отключится, то сервер не узнает об этом, так как UDP не поддерживает установление соединения. Сервер продолжит работать и будет ждать новых сообщений.

#### **6. Что произойдет, если udpclient отправит сообщение на несуществующий / выключенный сервер?**

- **UDP-клиент** отправит сообщение, но не получит ответ, поскольку сервер не активен. Программа продолжит работу, и пользователь не получит никаких уведомлений о проблемах с отправкой.

## **7. Что произойдет, если tcpclient отправит сообщение на несуществующий / выключенный сервер?**

- **TCP-клиент** не сможет установить соединение с сервером, поскольку сервер не слушает на указанном порту. Программа выдаст ошибку при попытке подключиться, и выполнение завершится с сообщением об ошибке.

## **8. В чем отличия UDP и TCP протоколов?**

### **Установление соединения:**

- **TCP**: Протокол ориентирован на соединение. Перед передачей данных устанавливается соединение между клиентом и сервером.

- **UDP**: Протокол без соединения. Данные могут быть отправлены без предварительного установления соединения.

### **Надежность:**

- **TCP**: Гарантирует доставку данных и порядок их получения. В случае потери пакетов происходит автоматическая повторная передача.

- **UDP**: Не гарантирует доставку и порядок. Пакеты могут быть потеряны, дублированы или прийти в неправильном порядке.

### **Скорость:**

- **TCP**: Медленнее из-за необходимости установления соединения и управления потоком.

- **UDP**: Быстрее, так как нет дополнительных накладных расходов на установление соединения и управление.

### **Использование:**

- **TCP**: Используется в приложениях, где важна надежность передачи данных (например, веб-браузеры, FTP).

- **UDP**: Используется в приложениях, где важна скорость (например, потоковое видео, онлайн-игры).