

Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и Технологий

Кафедра Компьютерных Систем и Программных Технологий

## Отчет по лабораторной работе

"Организация сетевого взаимодействия. Протоколы TCP и UDP"

Дисциплина: Сети и телекоммуникации

Работу выполнил студент группы 43501/1

Никитин А.Е.

Работу принял преподаватель

Алексюк А.О.

Санкт-Петербург

2017

## 1. Цель работы

Изучение принципов программирования сокетов с использованием протоколов TCP и UDP.

## 2. Индивидуальное задание

Разработать приложение–клиент и приложение–сервер Сервиса коммунальных платежей. Управляющая компания регистрирует счетчики с уникальными номерами на Абонента. Абонент вносит показания Счетчиков.

Основные возможности

Серверное приложение должно реализовывать следующие функции:

1. Прослушивание определенного порта
2. Обработка запросов на подключение по этому порту от пользователей Сервиса (Управляющая компания, Абонент)
3. Поддержка одновременной работы нескольких пользователей Сервиса через механизмы мультитредов
4. Осуществление добавления счетчиков Управляющей компанией, назначение Счетчиков на Абонентов
5. Прием запросов на:
  - a. фиксацию показаний счетчиков Абонентом на заданное число;
  - b. отображение истории показаний счетчика для Абонента и Управляющей компании с отображением приращения;
  - c. для Управляющей компании: отображение счетчиков и Абонентов, которые не вносили показания на заданное число;
6. Обработка запроса на отключение клиента
7. Принудительное отключение клиента

Клиентское приложение должно реализовывать следующие функции:

- Установление соединения с сервером
- Передача запросов серверу
- Получение ответов на запросы от сервера
- Разрыв соединения
- Обработка ситуации отключения клиента сервером

## Настройки приложений

Разработанное клиентское приложение должно предоставлять пользователю настройку IP-адреса и номера порта, используемого сервером.

## Методика тестирования

Для тестирования приложений запускается сервер и несколько клиентов. В процессе тестирования проверяются основные возможности приложений по передаче и приему информации.

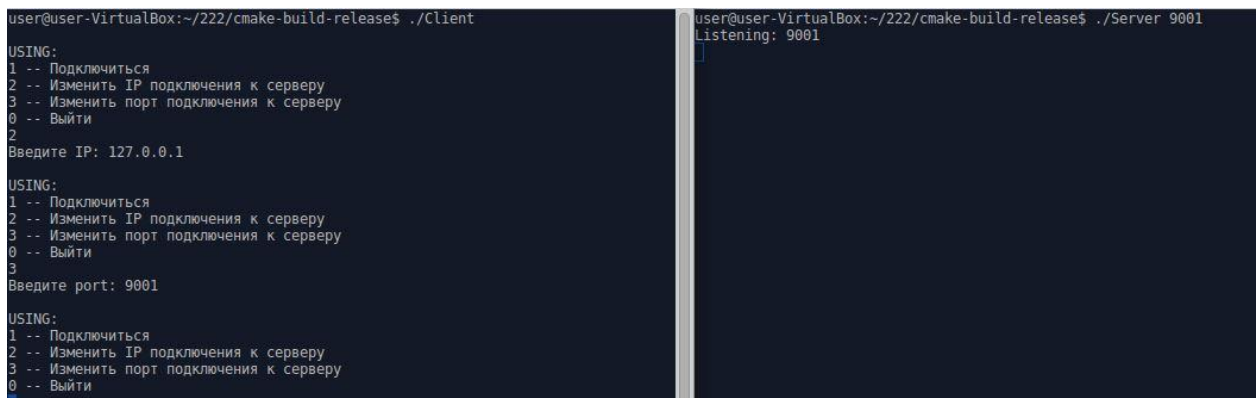
Вариант реализации: Сервер и Клиент - Linux

## 3. Принцип работы программы

Первоначально сервер прослушивает порт, введенный ему при создании как аргумент:

```
./Server 9001
```

После запуска сервера, он находится в режиме ожидания, пока к его порту не подключится клиент.



The screenshot shows two terminal windows side-by-side. The left window is the client terminal, and the right window is the server terminal. Both are running on a system with the path ~/222/cmake-build-release.

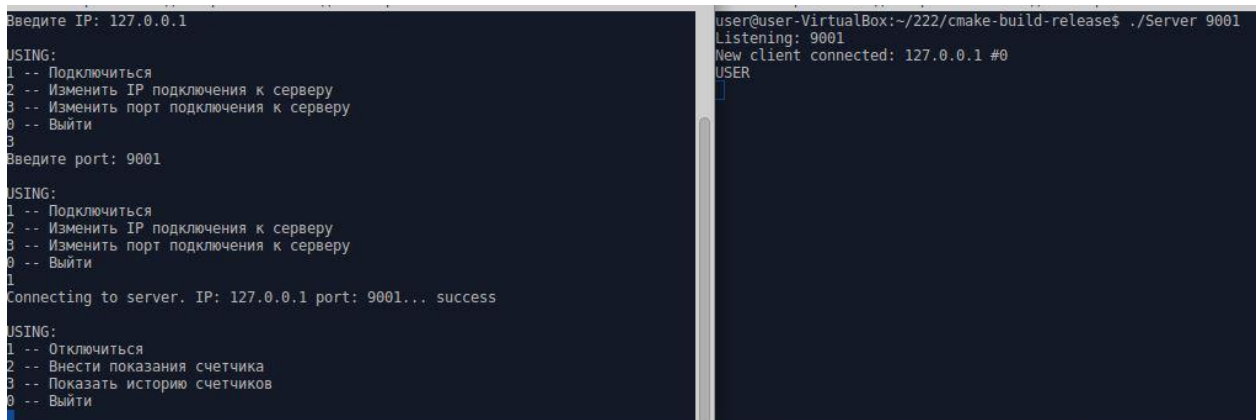
**Client Terminal (Left):**

```
user@user-VirtualBox:~/222/cmake-build-release$ ./Client
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
2
Введите IP: 127.0.0.1
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
3
Введите port: 9001
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
```

**Server Terminal (Right):**

```
user@user-VirtualBox:~/222/cmake-build-release$ ./Server 9001
Listening: 9001
```

Со стороны клиента выполняются настройки подключения: настройка IP адреса сервера и его порта. После того, как настройки были введены, клиент выбирает пункт "Подключиться".



The screenshot shows the same two terminal windows, but now the client is connected to the server. The server terminal shows a message indicating a new client connection.

**Client Terminal (Left):**

```
Введите IP: 127.0.0.1
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
3
Введите port: 9001
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
1
Connecting to server. IP: 127.0.0.1 port: 9001... success
USING:
1 -- Отключиться
2 -- Внести показания счетчика
3 -- Показать историю счетчиков
0 -- Выйти
```

**Server Terminal (Right):**

```
user@user-VirtualBox:~/222/cmake-build-release$ ./Server 9001
Listening: 9001
New client connected: 127.0.0.1 #0
USER
```

Со стороны сервера, Клиенту передается меню, в котором он может ввести показания своего счетчика, увидеть историю показаний, а также отключиться. В то же время Сервер, обработав, что к нему подключился клиент выделяет ему отдельный поток и ждет команды со стороны клиента.

```
Введите IP: 127.0.0.1
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
Введите port: 9001
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
1
Connecting to server. IP: 127.0.0.1 port: 9001... success
USING:
1 -- Отключиться
2 -- Внести показания счетчика
3 -- Показать историю счетчиков
0 -- Выйти

user@user-VirtualBox:~/222/cmake-build-release$ ./Server 9001
Listening: 9001
New client connected: 127.0.0.1 #0
USER
New client connected: 127.0.0.1 #1
ADMIN
```

Пока Клиент делал выбор в пунктах меню, к серверу подключился пользователь с правами админа, что видно в логе сервера.

```
0 -- Выйти
1
Connecting to server. IP: 127.0.0.1 port: 9001... success
USING:
1 -- Отключиться
2 -- Внести показания счетчика
3 -- Показать историю счетчиков
0 -- Выйти
2
Введите название счетчика:
nad_dverjy
Введите дату в формате дд.мм.гггг:
01.12.1935
Введите значение:
12345
Debugadd nad_dverjy 01.12.1935 12345
USING:
1 -- Отключиться
2 -- Внести показания счетчика
3 -- Показать историю счетчиков
0 -- Выйти

user@user-VirtualBox:~/222/cmake-build-release$ ./Server 9001
Listening: 9001
New client connected: 127.0.0.1 #0
USER
New client connected: 127.0.0.1 #1
ADMIN
add
nad dverjy
01.12.1935
12345
```

Пользователь не знает о том, что помимо него к серверу подключен еще 1 клиент(ADMIN), он вносит показания счетчика, которые видно в логах клиента(USER) и сервера.

После того, как пользователь закончил вносит данные, он выбирает пункт меню "Выйти". Сервер получает информацию о том, что 1 из клиентов отключены от сервера.

```
USING:
1 -- Отключиться
2 -- Внести показания счетчика
3 -- Показать историю счетчиков
0 -- Выйти
0

user@user-VirtualBox:~/222/cmake-build-release$ ./Server 9001
New client connected: 127.0.0.1 #1
ADMIN
add
nad dverjy
01.12.1935
12345
Client disconnected: 1 (error or hup)
```

Максимально возможное количество клиентов ограничено 500.

В случае, если клиент допустит ошибку в адресе сервера программа выдаст ошибку.

```
user@user-VirtualBox:~/222/cmake-build-debug$ ./Client
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
1
Connecting to server. IP: 192.168.31.106 port: 9000...
Socket exception: connect() No such process
Сервер не отвечает. Попробуйте изменить IP и/или порт.
USING:
1 -- Подключиться
2 -- Изменить IP подключения к серверу
3 -- Изменить порт подключения к серверу
0 -- Выйти
```

## 4. Выводы

В результате данной лабораторной работы был разработан и протестирован сервис коммунальных платежей. Для организации обмена между процессами используются сокеты. Для контроля доставки пакетов используется протокол TCP. Для передачи данных также используется TCP.

При использовании на транспортном уровне потокового протокола (TCP) для обработки нескольких клиентов удобно использовать на сервере механизм нитей. Каждая нить работает со своим сокетом: слушающая нить принимает соединения со слушающего сокета, при подключении клиента создаётся новая нить, которой передаётся порождённый сокет. На клиенте и на сервере возможна установка IP адреса и порта.