

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Сети и телекоммуникации

Отчет по лабораторной работе  
по сетевым технологиям

**Работу**  
**выполнила:**  
Фильчакова М.В.  
Группа: 43501/1  
**Преподаватель:**  
Алексюк А.О.

Санкт-Петербург  
2018

# Содержание

<b>1. Цель работы</b>	<b>3</b>
<b>2. Описание выполненных работ по TCP и UDP</b>	<b>3</b>
<b>3. Разница TCP и UDP</b>	<b>3</b>
3.1. TCP . . . . .	4
3.2. UDP . . . . .	4
<b>4. Индивидуальное задание</b>	<b>5</b>
<b>5. Разработанный прикладной протокол</b>	<b>6</b>
<b>6. Передача данных через TCP</b>	<b>7</b>
<b>7. Передача и прием дейтаграмм UDP</b>	<b>7</b>
<b>8. Тестирование приложения на основе TCP</b>	<b>8</b>
<b>9. Тестирование приложения на основе UDP</b>	<b>9</b>
<b>10. Листинги программ</b>	<b>10</b>
<b>11. Дополнительное задание</b>	<b>10</b>
11.1. Подключение к HTTP-серверу и запрос веб-страницы . . . . .	10
11.2. Подключение к FTP-серверу и запрос веб-страницы . . . . .	12
11.3. Подключение к SMTP-серверу и отправка письма . . . . .	16
11.4. Подключение к POP-серверу и отправка письма . . . . .	18
<b>12. Выводы</b>	<b>19</b>

# 1. Цель работы

Ознакомиться с принципами программирования собственных протоколов, созданных на основе TCP и UDP.

# 2. Описание выполненных работ по TCP и UDP

В ходе выполнения лабораторных работ были написаны простейшие клиент-серверные приложения на базе протоколов TCP и UDP. В приложениях TCP создается сокет, ставится на прослушивание и при подключении клиента создается отдельный сокет, по которому клиент общается с сервером. С помощью протокола UDP передаются сегменты, состоящие из 8-байтного заголовка, за которым следует поле полезной нагрузки.

# 3. Разница TCP и UDP

Разница между протоколами TCP и UDP – в так называемой “гарантии доставки”. TCP требует отклика от клиента, которому доставлен пакет данных, подтверждения доставки, и для этого ему необходимо установленное заранее соединение. Также протокол TCP считается надежным, тогда как UDP получил даже именование “протокол ненадежных датаграмм. TCP исключает потери данных, дублирование и перемешивание пакетов, задержки. UDP все это допускает, и соединение для работы ему не требуется. Процессы, которым данные передаются по UDP, должны обходиться полученным, даже и с потерями. TCP контролирует загруженность соединения, UDP не контролирует ничего, кроме целостности полученных датаграмм.

С другой стороны, благодаря такой не избирательности и бесконтрольности, UDP доставляет пакеты данных (датаграммы) гораздо быстрее, потому для приложений, которые рассчитаны на широкую пропускную способность и быстрый обмен, UDP можно считать оптимальным протоколом. К таковым относятся сетевые и браузерные игры, а также программы просмотра потокового видео и приложения для видеосвязи (или голосовой): от потери пакета, полной или частичной, ничего не меняется, повторять запрос не обязательно, зато загрузка происходит намного быстрее. Протокол TCP, как более надежный, с успехом применяется даже в почтовых программах, позволяя контролировать не только трафик, но и длину сообщения и скорость обмена трафиком. Рассмотрим основные отличия tcp от udp.

1 TCP гарантирует доставку пакетов данных в неизменном виде, последовательности и без потерь, UDP ничего не гарантирует. 2 TCP нумерует пакеты при передаче, а UDP нет 3 TCP работает в дуплексном режиме, в одном пакете можно отправлять информацию и подтверждать получение предыдущего пакета. 4 TCP требует заранее установленного соединения, UDP соединения не требует, у него это просто поток данных. 5 UDP обеспечивает более высокую скорость передачи данных. 6 TCP надежнее и осуществляет контроль над процессом обмена данными. 7 UDP предпочтительнее для программ, воспроизводящих потоковое видео, видеоконференции и телефонии, сетевых игр. 8 UDP не содержит функций восстановления данных

### 3.1. TCP

Для инициализации, запуска и завершения TCP-сервера необходимо выполнить следующие системные вызовы:

1. `socket()` - создание сокета
2. `bind()` - привязка созданного сокета к заданным IP-адресам и портам
3. `listen()` – перевод сокета в состояние прослушивания
4. `accept()` - прием поступающих запросов на подключение и возврат сокета для нового соединения
5. `recv()` - чтение данных от клиента из сокета, полученного на предыдущем шаге
6. `send()` - отправка данных клиенту с помощью того же сокета
7. `shutdown()` - разрыв соединения с клиентом
8. `close()` - закрытие клиентского и слушающего сокетов

TCP-клиенты выполняют следующую последовательность действий для открытия соединения, отправки и получения данных, и завершения:

1. `socket()` - создание сокета
2. `connect()` - установка соединения для сокета, который будет связан с серверным сокетом, порожденным вызовом `accept()`
3. `send()` - отправка данных серверу
4. `recv()` - прием данных от сервера
5. `shutdown()` - разрыв соединения с сервером
6. `close()` - закрытие сокета

Так же был реализован сервер, поддерживающий работу с несколькими клиентами. Для этого при подключении клиента создается поток, который в котором создается сокет для общения с клиентом.

### 3.2. UDP

В приложениях UDP сервер принимает сообщение от клиента и отправляет сообщение об успешной доставке. UDP протокол не подразумевает логических соединений, поэтому не создается слушающего сокета.

Реализация UDP-сервера имеет следующий вид:

1. `socket()` - создание сокета
2. `bind()` - привязка созданного сокета к заданным IP-адресам и портам
3. `recvfrom()` - получение данных от клиента, параметры которого заполняются функцией

4. `sendto()` - отправка данных с указанием параметров клиента, полученных на предыдущем шаге
5. `close()` - закрытие сокета

UDP-клиент для обмена данными с UDP-сервером использует следующие функции:

1. `socket()` - создание сокета
2. `recvfrom()` - получение данных от сервера, параметры которого заполняются функцией
3. `sendto()` - отправка данных с указанием параметров сервера, полученных на предыдущем шаге
4. `close()` - закрывает сокет

Проверено 2 способа написания клиента:

- С использованием функции `connect`
- Без использования функции `connect`

В первом случае устанавливается соединение и клиент, и сервер работают аналогично TCP. Во втором случае при отсутствии доступа к серверу сообщение об ошибке не возникает, и клиент считает, что данные отправлены корректно.

## 4. Индивидуальное задание

Информационная система

Задание: разработать распределенную информационную систему, состоящую из приложения-сервера и приложения-клиента. Информационная система является иерархическим хранилищем статей, каждая из которых состоит из названия, автора и текста статьи. Информационная система должна обеспечивать параллельный доступ к информации нескольким клиентам.

Основные возможности.

Серверное приложение должно реализовывать следующие функции:

- 1) Прослушивание определенного порта
- 2) Обработка запросов на подключение по этому порту от клиентов
- 3) Поддержка одновременной работы нескольких клиентов через механизм нитей
- 4) Хранение иерархической структуры статей информационной системы
- 5) Передача пользователю списка текущих разделов системы, списка статей
- 6) Переход в конкретный раздел системы по запросу клиента
- 7) Возврат на предыдущий уровень по запросу клиента
- 8) Передача пользователю конкретной статьи по названию
- 9) Передача пользователю всех статей текущего раздела, принадлежащих определенному автору
- 10) Приём от клиента новой статьи и сохранение в информационной системе
- 11) Обработка запроса на отключение клиента
- 12) Принудительное отключение клиента

Клиентское приложение должно реализовывать следующие функции:

- 1) Установление соединения с сервером

- 2) Получение и печать списка подразделов и статей раздела
- 3) Передача команды на переход в конкретный раздел
- 4) Передача команды на переход в раздел на уровень выше
- 5) Получение конкретной статьи из информационной системы
- 6) Получение статей конкретного автора
- 7) Посылка новой статьи в систему
- 8) Разрыв соединения
- 9) Обработка ситуации отключения клиента сервером

Настройки приложений.

Разработанное клиентское приложение должно предоставлять пользователю возможность введения идентификационной информации, настройки IP-адреса или доменного имени, а также номера порта сервера информационной системы.

Разработанное серверное приложение должно предоставлять пользователю возможность настройки начальной точки входа в информационную систему каждого пользователя.

Методика тестирования.

Для тестирования приложений запускается сервер информационной системы и несколько клиентов. В процессе тестирования проверяются основные возможности приложений по передаче статей и навигации по разделам информационной системы.

Источники: конспект лекций [1], [2]

## 5. Разработанный прикладной протокол

После подключения клиента к серверу, сервер ждет команд от клиента. Создано 5 возможных команд от клиента

1. ls - вывести список файлов. (article - статья, section - папка)
2. cd <sec> - перейти в папку
3. get <art> - вывести статью
4. getByAuthor <author> - вывести статьи заданного мастера
5. put - добавить статью
6. quit - выход

Если клиент ввел несуществующую команду, то клиенту будет предложено заново ввести существующие команды и их список.

Реакция на команды клиента:

1. Unknown command - не существующая команда
2. Failed to change current section - при попытке перейти в несуществующую папку
3. Sorry, the article is corrupted - при попытке вывода несуществующей статьи
4. При вызове cd . выводится директория, в которой находится клиент
5. Current section is changed - при попытке перейти на высшие ветки
6. Failed to save the article - при некорректном добавлении статьи
7. Article successfully saved - при удачном создании статьи
8. Bye - при выходе из клиента

Поддерживаемые команды сервера:

1. ./server /home/masha/.. - по умолчанию сервер подключается к папке data, но, при желании, мы можем указать любую другую папку
2. disconnect <name> - отключение клиента
3. ls - список клиентов, подключенных к серверу

4. `command` - ожидание ввода команды

Отклик-сообщения на сервере:

1. `Endpoint created`

`Address and port assigned to socket` - подключение сервера

2. `get started` - начало работы сервера

## 6. Передача данных через TCP

Для обмена данными при помощи протокола TCP используются функции `send` и `recv`. Функция `send` предназначена для отправки данных:

```
1 ssize_t send(int socket, const void *buffer,  
2 size_t length, int flags);
```

Функция выполняет передачу данных через указанный сокет партнеру. Аргумент `socket` определяет дескриптор файла сокета, через который отправляются данные. `buffer` указывает на буфер, содержащий данные для передачи. Длина передаваемых данных определяется аргументом `length`. Аргумент `flags` определяет тип передачи данных. Значение `flags` является результатом логического ИЛИ нуля или большего числа следующих констант:

`MSG_OOB` - передать срочные данные.

`MSG_DONTROUTE` - игнорировать параметры маршрутизации.

В случае успешного завершения `send` возвращает число переданных байт. В противном случае возвращаемое значение равно -1.

Для приема данных используется функция `recv`.

```
1 ssize_t recv(int socket, void *buffer, size_t length, int flags);
```

Функция `recv` принимает данные из сокета, заданного первым аргументом. Аргумент `buffer` указывает на буфер в который будут помещены принятые данные. `length` определяет длину буфера. Аргумент `flags` определяет параметры получения данных. Значение `flags` является результатом логического ИЛИ нуля или большего числа следующих констант:

`MSG_PEEK` - данные не удаляются из буфера приема. Следующий вызов функции `recv` прочитает те же данные.

`MSG_OOB` - принять срочные данные.

`MSG_WAITALL` - блокировать функцию, пока не будет принят полный объем данных, определенный аргументом `length`. Функция может вернуть меньший объем данных в случае обрыва соединения, ошибки, связанной с сокетом, использования флага `MSG_PEEK`. В случае успешного завершения функция возвращает число принятых байт. В противном случае возвращается -1.

## 7. Передача и прием дейтаграмм UDP

UDP реализуется на хостах. Протокол не обеспечивает целостности доставки данных, поскольку эта функция возлагается на обменивающиеся данными приложения. Именно

они проверяют целостность доставляемых данных.

Приложение, которое хочет переслать данные с помощью UDP, передает блок данных в UDP, а протокол UDP просто добавляет к ним заголовок и производит их пересылку по сети.

Участвующие во взаимодействии по UDP приложения могут посылать сообщения с пользовательскими датаграммами в любое время. Клиент и сервер, которые настроены над UDP, несут ответственность за все взаимоотношения при обмене пользовательскими датаграммами.

Класс `QUdpSocket` может использоваться для отправки и приема дейтаграмм UDP. UDP — это ненадежный, ориентированный на дейтаграммы протокол. Некоторые приложения применяют протокол UDP, поскольку с ним легче работать, чем с протоколом TCP. По протоколу UDP данные передаются пакетами (дейтаграммами) от одного хоста к другому. Для него не существует понятия соединения, и если доставка пакета UDP в пункт назначения завершается неудачей, никакого сообщения об ошибке не передается отправителю.

## 8. Тестирование приложения на основе TCP

Для тестирования приложения запускается сервер и несколько клиентов. Проверялись все команды на работоспособность.

Работа клиента

Листинг 1:

```
1 masha@masha-PC:~/data/client$ ./client me
2 Connected successfully client:
3 Message to send: ls
4 article: gth
5 section: a
6 article: 2
7 article: lala
8 section: b
9 article: 56
10 article: e
11 Message to send: put
12 Title: 678
13 Author: rt
14 Article content: rdytfuy
15
16 Article successfully added to the section
17 Message to send: ls
18 article: gth
19 article: 678
20 section: a
21 article: 2
22 article: lala
23 section: b
24 article: 56
25 article: e
26 Message to send: cd a
27 /a/
28 Message to send: cd .
29 /a/
30 Message to send: get 2
31 Article does not exist
32 Message to send: get 56
33 Article does not exist
```



```

34 Message to send: cd ..
35 /
36 Message to send: get 2
37 Title: 2
38
39 Author: Somebody
40 Article content:
41 hello
42 some
43 Message to send: getByAuthor Somebody
44 2
45 Message to send: quit
46 Bye!
47 masha@masha-PC:~/data/client$

```

Работа сервера

Листинг 2:

```

1 masha@masha-PC:~/data/server$ ./server
2 Connection accepted
3 Handler assigned
4 Enter command: ls
5 me
6 Enter command: disconnect me
7
8 _____
9 Enter command
10 Client Disconnected

```

## 9. Тестирование приложения на основе UDP

Для тестирования приложения запускается сервер и несколько клиентов. Проверялись все команды на работоспособность.

Работа клиента

Листинг 3:

```

1 masha@masha-PC:~/data/u/udp$ ./client
2 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): ls
3 section: a
4 article: 2
5 article: 1
6 section: b
7 article: 3
8 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): ls
9 section: a
10 article: 2
11 article: 1
12 section: b
13 article: 3
14 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): put 3
15 Unknown command
16 Please, use one of the following: ls, cd <sec>, get <art>, put, quit
17 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): put
18 Title: 123456789
19 Author: 123
20 Article content (use empty line for the end of content): trtyfugtgh
21 uhjk

```

```

22 ghjbk
23 ghjkl
24
25 Article successfully saved
26 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): ls
27 section: a
28 article: 2
29 article: 1
30 section: b
31 article: 3
32 article: 123456789
33 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>):
    ↪ getByAuthor 123
34 Unknown command
35 Please, use one of the following: ls, cd <sec>, get <art>, put, quit
36 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>):
    ↪ getByAuthor 123
37 123456789
38 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): get 3
39 Title: 3
40 Author: Robin Good
41 Article content:
42 meow
43 meow
44 meow
45 meow
46 Input command (ls, cd <sec>, get <art>, put, quit, getByAuthor <author>): quit
47 Bye
48 masha@masha-PC:~/data/u/udp$

```

Работа сервера

Листинг 4:

```

1 masha@masha-PC:~/data/u/udp$ ./server
2 Endpoint created
3 Address and port assigned to socket
4 42489
5 /
6 42489
7 /
8 42489
9 42489
10 /
11 42489
12 42489
13 get started

```

## 10. Листинги программ

Листинг на гитхабе

## 11. Дополнительное задание

### 11.1. Подключение к HTTP-серверу и запрос веб-страницы

С помощью команды telnet подключаемся к HTTP-серверу:

```
1 masha@masha-PC:~$ telnet avalon.ru 80
```

avalon.ru - подключаемый сервер, 80 номер порта.

```
1 GET / HTTP/1.0
2 host: avalon.ru
```

Веб-страница успешно выведена:

```
1 masha@masha-PC:~$ telnet avalon.ru 80
2 Trying 195.209.230.130...
3 Connected to avalon.ru.
4 Escape character is '^]'.
5 GET / HTTP/1.0
6 host: avalon.ru
7
8
9 HTTP/1.1 200 OK
10 Connection: close
11 Content-Length: 31080
12 Date: Thu, 20 Dec 2018 20:09:32 GMT
13 Content-Type: text/html; charset=utf-8
14 Server: Microsoft-IIS/7.5
15 Cache-Control: private
16 X-Powered-By: NULL
17 X-Xss-Protection: 1; mode=block
18 X-Content-Type-Options: nosniff
19 X-Frame-Options: SAMEORIGIN
20 X-AspNet-Version: 0
21
22 <!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01_Transitional//EN" "http://www.w3.
    ↳ org/TR/html4/loose.dtd" >
23
24
25
26
27 <html xmlns:fb="http://ogp.me/ns/fb#">
28 <head>
29 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
30 <meta name="verify-v1" content="wjJ9jC5DMXALKf6wTbC4bscIjkRfvyIi2h/9altvKYY=" >
31 <meta name="verify-v1" content="WQ858c6UOwwCvaGFyLB3BF2J45D6EFdIXludlNf/lM8=" >
32 <meta name="verify-v1" content="F8rIPD9QlvxqwxuzyLaXUnNHAPCjggT8qF3xeJUgIE8=" >
33 <meta name="yandex-verification" content="7fa66c2f3ad59d38" />
34 <title Компьютерные курсы, второе высшее образование, курсы для IT специалистов,
    ↳ авторизованные курсы – обучение в СанктПетербурге – Высшая инженерная школа
    ↳ СПбПУ (</title>
35
36 <link rel="shortcut_icon" href="/favicon.ico" type="image/x-icon">
37 <link rel="apple-touch-icon-precomposed" href="/_images/appleicon.png" />
38
39 <meta http-equiv="Content-Language" content="ru">
40
41 <link rel="image_src" href="http://img.avalon.ru/Logos/fst4fb.png" />
42 .....
43 (function() {
44 var ga = document.createElement('script');
45 ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www')
    ↳ + '.google-analytics.com/ga.js';
46 ga.setAttribute('async', 'true');
47 document.documentElement.firstChild.appendChild(ga);
48 })();
```

```

49|</script>
50|<!-- /Google -->
51|</div>
52|</body>
53|</html>Connection closed by foreign host.

```

Как видно, все успешно работает.

## 11.2. Подключение к FTP-серверу и запрос веб-страницы

Подключаемся к серверу с помощью команды:

```
1 masha@masha-PC:~$ telnet mirror.yandex.ru 21
```

Подключение происходит через 21 порт.

Нужно ввести имя пользователя и пароль. Затем переходим в пассивный режим. Кодировем IP-адрес и порт для соединения.

Содаем соединение к указанному IP-адресу через порт, который вычисляется по формуле  $p1*256+p2$ , в которой  $p1$  и  $p2$  -два последних числа в присланом сообщении сервера.

```

1
2 Trying 213.180.204.183...
3 Connected to mirror.yandex.ru.
4 Escape character is '^]'.
5 220-Welcome to Yandex Mirror FTP service. Your served by: node01e.mirror.yandex.
   ↪ net
6 220
7 user anonymous
8 331 Please specify the password.
9 pass mikhsbstu@mail.ru
10 230 Login successful.
11 pasv
12 227 Entering Passive Mode (213,180,204,183,224,84)
13 list
14 150 Here comes the directory listing.
15 226 Directory send OK.
16
17 _____
18 maria_rheon@Rheon:~$ telnet 213.180.204.183 57428
19 Trying 213.180.204.183...
20 Connected to 213.180.204.183.
21 Escape character is '^]'.
22 drwxr-xr-x    21 ftp      ftp      4096 Dec 20 04:46 altlinux
23 drwxr-sr-x    21 ftp      ftp      4096 Dec 06 04:15 altlinux-beta
24 drwxr-xr-x     9 ftp      ftp      4096 Dec 20 15:14 altlinux-nightly
25 drwxr-xr-x     5 ftp      ftp      4096 Dec 12 19:46 altlinux-starterkits
26 drwxr-xr-x    17 ftp      ftp      4096 Dec 20 18:34 archlinux
27 drwxr-xr-x     7 ftp      ftp      4096 Oct 12 2015 archlinux-arm
28 drwxr-xr-x     8 ftp      ftp      4096 Dec 20 19:32 archlinux32
29 dr-xr-xr-x     7 ftp      ftp      4096 Dec 06 03:12 archserver
30 drwxr-xr-x     4 ftp      ftp      4096 Dec 12 15:46 astra
31 drwxr-xr-x    10 ftp      ftp      4096 Dec 20 17:56 calculate
32 drwxrwxr-x    48 ftp      ftp      4096 Dec 20 18:09 centos
33 drwxr-xr-x     9 ftp      ftp      4096 Dec 20 18:52 debian
34 drwxrwsr-x     5 ftp      ftp      4096 Nov 20 15:22 debian-backports
35 drwxr-xr-x     5 ftp      ftp      4096 Dec 20 15:36 debian-cd
36 drwxr-sr-x     8 ftp      ftp      4096 Dec 20 02:20 debian-multimedia
37 drwxr-xr-x    19 ftp      ftp      4096 Dec 20 17:43 debian-ports
38 drwxr-xr-x     7 ftp      ftp      4096 Dec 20 18:05 debian-security

```

```

39 drwxrwsr-x      7 ftp      ftp      4096 Dec 19 05:32 epel
40 drwxr-xr-x      9 ftp      ftp      4096 Dec 20 17:54 fedora
41 drwxrwxr-x      5 ftp      ftp      4096 Dec 20 17:50 fedora-secondary
42 drwxrwxr-x      7 ftp      ftp      4096 Dec 20 17:16 freebsd
43 drwxr-xr-x      6 ftp      ftp      4096 Dec 20 19:32 gentoo-distfiles
44 drwxr-xr-x     171 ftp      ftp      4096 Dec 20 20:02 gentoo-portage
45 drwxr-xr-x     10 ftp      ftp     12288 Dec 05 21:02 knoppix
46 drwxr-xr-x      3 ftp      ftp      4096 Dec 20 15:06 libreoffice
47 drwxr-xr-x      5 ftp      ftp      4096 Dec 18 13:34 linuxmint
48 drwxr-xr-x      5 ftp      ftp      4096 Dec 20 11:16 linuxmint-packages
49 drwxr-xr-x      6 ftp      ftp      4096 Dec 20 17:26 macports
50 drwxr-xr-x      6 ftp      ftp      4096 Dec 20 20:10 mageia
51 drwxr-xr-x     53 ftp      ftp      4096 Sep 21 15:11 mirrors
52 drwxr-xr-x      6 ftp      ftp      4096 Mar 11 2014 mopslinux
53 drwxr-xr-x     58 ftp      ftp      4096 Dec 20 15:23 openbsd
54 drwxr-xr-x      5 ftp      ftp      4096 Dec 20 19:06 openmandriva
55 drwxr-xr-x      8 ftp      ftp      4096 Dec 20 20:22 opensuse
56 drwxr-xr-x      9 ftp      ftp      4096 Dec 20 19:05 pub
57 drwxr-xr-x     36 ftp      ftp      4096 Dec 18 15:33 puia
58 drwxr-xr-x     15 ftp      ftp      4096 Dec 10 21:00 puppyrus
59 drwxrwxrwx     14 ftp      ftp      4096 Dec 20 18:42 rosa
60 drwxr-xr-x      6 ftp      ftp      4096 Dec 11 21:00 runtu
61 drwxrwxr-x      9 ftp      ftp      4096 Dec 20 17:35 sabayon
62 drwxr-xr-x     27 ftp      ftp      4096 Dec 20 17:34 scientificlinux
63 drwxr-xr-x      6 ftp      ftp      4096 Apr 02 2014 simplelinux
64 drwxr-xr-x     49 ftp      ftp      4096 Dec 20 17:35 slackware
65 drwxr-xr-x     15 ftp      ftp      4096 Dec 15 21:39 slackwarearm
66 drwxr-xr-x      7 ftp      ftp      4096 Dec 20 18:54 ubuntu
67 drwxr-xr-x     28 ftp      ftp      4096 Dec 20 20:16 ubuntu-cdimage
68 drwxr-xr-x      6 ftp      ftp      4096 Dec 20 15:50 ubuntu-ports
69 drwxr-xr-x     13 ftp      ftp      4096 Dec 20 19:16 ubuntu-releases
70 Connection closed by foreign host.
71 -----FILE DOWNLOADING
72 .....
73
74 | |-- a/           The A (base) package series.
75 | |-- ap/         The AP (applications) package series.
76 | |-- d/          The D (development) package series.
77 | |-- e/          The E (GNU Emacs) package series.
78 | |-- f/          The F (FAQ/Documentation) package series.
79 | |-- k/          The K (kernel source) package series.
80 | |-- kde/        The KDE package series.
81 | |-- kdei/       The KDE internationalization package series.
82 | |-- l/          The L (libraries) package series.
83 | |-- n/          The N (networking) package series.
84 | |-- t/          The T (TeX) package series.
85 | |-- tcl/        The TCL (Tcl/Tk and related) package series.
86 | |-- x/          The X (X Window System) package series.
87 | |-- xap/        The XAP (X applications) package series.
88 | |-- xfce/       The XFCE (Xfce desktop) package series.
89 | |-- y/          The Y (BSD games) package series.
90 |
91 | |-- source/     This directory contains source code for the core
92 | |              software packages in Slackware.
93 |
94 | |-- a/          Source for the A (base) series.
95 | |-- ap/         Source for the AP (applications) series.
96 | |-- d/          Source for the D (development) series.
97 | |-- e/          Source for the E (GNU Emacs) series.
98 | |-- f/          slack-desc files for the F (FAQ) series.

```

```

99 | |-- installer/      Source for utilities used in the installer.
100 | |-- k/             Source for the K (kernel source) series.
101 | |-- kde/          Source for the KDE series.
102 | |-- kdei/         Source for the KDEI series.
103 | |-- l/            Source for the L (libraries) series.
104 | |-- n/            Source for the N (networking) series.
105 | |-- t/            Source for the T (TeX) series.
106 | |-- tcl/          Source for the TCL (Tcl/Tk and related) series.
107 | |-- x/            Source for the X (X Window System) series.
108 | |-- xap/          Source for the XAP (X applications) series.
109 | |-- xfce/         Source for the XFCE (Xfce desktop) series.
110 | |-- y/            Source for the Y (BSD games) series.
111 |
112 | |-- testing/       This directory contains packages being tested
113 | |                 for possible use in future versions of Slackware.
114 | |                 For this release, it contains efibootmgr-0.6.0,
115 | |                 tmux-2.2, and sample kernel configuration files
116 | |                 for Linux 4.6.
117 | |
118 | |-- source/       This directory contains source code for the
119 | |                 packages in testing.
120 |
121 | |-- usb-and-pxe-installers These are alternate installers useful
122 | |                 for large-scale Linux deployment or if
123 | |                 your machine can boot PXE and/or USB but
124 | |                 not an optical disc. For PXE, the initrd.img
125 | |                 file in isolinux/ is also required.
126 |
127 | |-- README_PXE.TXT
128 | |-- README_USB.TXT
129 | |-- pxelinux.cfg_default
130 | |-- usbboot.img    The Slackware installer for bootable USB flash.
131 | |-- usbimg2disk.sh Script to make a partitioned USB flash installer
132 | from usbboot.img.
133 |
134 |
135 | If you like Slackware, please consider supporting the project by becoming
136 | a Slackware subscriber. The announcement (ANNOUNCE_14.2) in this directory
137 | has information about ordering or subscribing to the Slackware CD-ROM or
138 | DVD releases. Note that the x86_64 release is only available as part of the
139 | DVD. You can find out about ordering Slackware on media, or read more about
140 | it (and check out other Slackware-related items for sale) by visiting the
141 | Slackware store. We also gratefully accept financial contributions under
142 | the "All" section of the store site menu.
143 |
144 | Thanks for your consideration and help!
145 |
146 | http://store.slackware.com
147 |
148 |
149 | Enjoy!
150 |
151 | Patrick Volkerding
152 | volkerdi@slackware.com
153 |
154 | Connection closed by foreign host.

```

### 11.3. Подключение к SMTP-серверу и отправка письма

Подключаемся к серверу с помощью команды:

```
1 masha@masha-PC:~$ gnutls-cli smtp.mail.ru -p 465
```

Команда telnet не подходит для подключения, тк она работает только с нешифрованными соединениями.

Соединение, которое защищено протоколом TLS, обладает безопасностью, аутентификацией и целостностью.

Так же необходимо было пройти аутентификацию. Для нее использовали кодировку BASE64.

```
1 Processed 148 CA certificate(s).
2 Resolving 'smtp.mail.ru'...
3 Connecting to '217.69.139.160:465'...
4 - Certificate type: X.509
5 - Got a certificate list of 2 certificates.
6 - Certificate[0] info:
7 - subject 'C=RU,L=Moscow,O=LLC Mail.Ru,OU=IT,CN=*.mail.ru', issuer 'C=US,O=
  ↳ DigiCert_Inc,OU=www.digicert.com,CN=GeoTrust_RSA_CA_2018', RSA key 2048
  ↳ bits, signed using RSA-SHA256, activated '2017-12-15 00:00:00 UTC',
  ↳ expires '2020-12-14 12:00:00 UTC', SHA-1 fingerprint '
  ↳ fbb4bf9db6aaac58ce533b41e483ac2ac3f0c5c5'
8 Public_Key_ID:
9 0ca28f9da4ad384258ba65ef9db6b629cb99dd1a
10 Public_key's random art:
11 +--[ RSA 2048 ]-----+
12 |
13 |
14 |      . .
15 |    . . . o
16 | . o . .   S
17 | o.oB .
18 | .+o.= E
19 |+. oo=o=
20 |o...*=O+.
21 +-----+
22
23 - Certificate[1] info:
24 - subject 'C=US,O=DigiCert Inc,OU=www.digicert.com,CN=GeoTrust RSA CA 2018',
  ↳ issuer 'C=US,O=DigiCert_Inc,OU=www.digicert.com,CN=DigiCert_Global_Root_CA
  ↳ ', RSA key 2048 bits, signed using RSA-SHA256, activated '2017-11-06
  ↳ 12:23:45 UTC', expires '2027-11-06 12:23:45 UTC', SHA-1 fingerprint '7
  ↳ ccc2a87e3949f20572b18482980505fa90cac3b'
25 -_Status:_The_certificate_is_trusted.
26 -_Description:_(TLS1.2)-(ECDHE-RSA-SECP256R1)-(AES-128-GCM)
27 -_Session_ID:_77:65:3B:DB:03:31:74:42:FE:8B:CC:AF:A2:08:53:87:BB:8D:00:60:B9:C7
  ↳ :09:24:1D:0B:03:A4:29:9C:52:CE
28 -_Ephemeral_EC_Diffie-Hellman_parameters
29 -_Using_curve:_SECP256R1
30 -_Curve_size:_256_bits
31 -_Version:_TLS1.2
32 -_Key_Exchange:_ECDHE-RSA
33 -_Server_Signature:_RSA-SHA256
34 -_Cipher:_AES-128-GCM
35 -_MAC:_AEAD
36 -_Compression:_NULL
37 -_Options:_safe_renegotiation,
38 -_Handshake_was_completed
39
40 -_Simple_Client_Mode:
```

```

41 |
42 | 220 smtp16.mail.ru_ESMTP_ready_(Looking_for_Mail_for_your_domain?_Visit_https://
    | ↪ biz.mail.ru)
43 | EHLO_host-163-154-66-217.spbmts.ru.
44 | 250-smtp16.mail.ru
45 | 250-SIZE_73400320
46 | 250-8BITMIME
47 | 250-PIPELINING
48 | 250_AUTH_PLAIN_LOGIN_XOAUTH2
49 | AUTH_LOGIN
50 | 334_VXNlcm5hbWU6
51 | dGVzdF9zbXRwOTlAbWFpbC5ydQ==
52 | 334_UGFzc3dvcmQ6
53 | MTIzNDVxMTIzNDU=
54 | 235_Authentication_succeeded

```

Далее, используя ниже приведенные команды, отправляем любое письмо кому хотим.

```

1 | MAIL FROM: test_smtp99@mail.ru
2 | 250 OK
3 | RCPT TO: test_smtp99@mail.ru
4 | 250 Accepted
5 | DATA
6 | 354 Enter message, ending with "." on a line by itself
7 | subject: my name is masha
8 | hello from masha
9 | .
10 | 250 OK id=1ga5Mf-0000vi-K5

```

После проведения процедуры, на почте появилось новое сообщение, которое полностью соответствует отправленному.

## 11.4. Подключение к POP-серверу и отправка письма

Чтобы прочитать посланное письмо, нужно подключиться к POP-3-серверу:

```

1 | masha@masha-PC:~$ gnutls-cli pop.mail.ru -p 995

```

Подключаемся

Затем вводим логин и пароль, чтобы прочитать письмо

```

1 | - Simple Client Mode:
2 |
3 | +OK
4 | USER test_smtp99@mail.ru
5 | +OK
6 | PASS 12345q12345
7 | +OK Welcome!
8 | stat
9 | +OK 9 350466
10 | list
11 | +OK 9 messages (350466 octets)
12 | 1 80547
13 | 2 105711
14 | 3 152991
15 | 4 1646
16 | 5 1625
17 | 6 2460

```



```

18 7 2038
19 8 1642
20 9 1806
21 .
22 retr 8
23 +OK 1642 octets
24 Delivered-To: test_smtp99@mail.ru
25 Return-path: <mikhspbstu@mail.ru>
26 Received: by smtp59.i.mail.ru with esmtpa (envelope-from <mikhspbstu@mail.ru>)
27 id 1ga5LR-0002oC-Ry
28 for test_smtp99@mail.ru; Thu, 20 Dec 2018 23:55:39 +0300
29 subject: mashaaaaaaaaaaaaaa
30 Message-Id: <Elga5LR-0002oC-Ry.mikhspbstu-mail-ru@smtp59.i.mail.ru>
31 From: mikhspbstu@mail.ru
32 Date: Thu, 20 Dec 2018 23:55:39 +0300
33 Authentication-Results: smtp59.i.mail.ru; auth=pass smtp.auth=mikhspbstu@mail.ru
    ↪ smtp.mailfrom=mikhspbstu@mail.ru
34 X-77F55803:
    ↪ BBE463BEF7A60BD05A78504BD2AC29419868B496DAF47A9F35BF3FB0B27CB4996055AC856B477515F090
    ↪
35 X-7FA49CB5: 0
    ↪ D63561A33F958A56F16C0898A6A22D305A0B2A982EC6C01878C2A97FB14A3038941B15DA834481FA18204F
    ↪
36 X-Mailru-Sender:
    ↪ C8A6F306A889CEAC8CED9B47B8D6BDB2736727E9DDF2C863B5C0DC700C4D5C8B490970484213A17C3480
    ↪
37 X-Mras: OK
38 X-Mailru-Intl-Transport: d,b26656f
39
40 masha bistree delay!

```

В итоге, письмо было прочитано, это видно по самой нижней строчке.

## 12. Выводы

В данной лабораторной работе были реализованы клиент-серверные программы выставления курса валют, с написанием собственного протокола на основе TCP и UDP.

В ходе выполнения работы были изучены принципы организации многопоточного сервера, изучены принципы синхронизации доступа к глобальным переменным. В разработанном в ходе работы сервере для каждого клиента создается отдельный поток. Такой подход оправдан, т.к. клиенты могут исполнять долгие операции и операции различной трудоемкости. В этом случае использование отдельного потока для каждого клиента обеспечивает минимизацию взаимного влияния клиентов друг на друга. Создание потока – достаточно ресурсоемкая операция как по времени выполнения, так и по расходу оперативной памяти, поэтому при использовании данного подхода нужно следить за расходом памяти.

Однако TCP может не подходить в некоторых ситуациях обмена по сети вследствие медленной (по сравнению с UDP) работы. Например, передавая по сети данные требующие быстрого отклика в реальном времени, необходимо соблюдать жесткие временные рамки, которые не может выдержать протокол TCP

В ходе данной работы я так же ознакомилась с протоколом UDP и реализовала клиент-серверное приложение. По сравнению с TCP, UDP – более простой, основанный на сообщениях протокол без установления соединения, однако требует дополнительного контроля доставки сообщений ввиду следующих особенностей:

- Ненадежность – когда сообщение посылается, неизвестно, достигнет ли оно своего назначения – оно может потеряться по пути.

- Неупорядоченность – если два сообщения отправлены последовательно, порядок их получения не может быть предугадан.