

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Сети и телекоммуникации

Отчет по лабораторной работе  
Программирование сокетов протоколов TCP и UDP

**Работу**  
**выполнил:**  
Ерниязов Т.Е.  
Группа: 43501/3  
**Преподаватель:**  
Алексюк А.О.

Санкт-Петербург  
2018

# 1. Цель работы

Изучение принципов программирования сокетов протоколов TCP и UDP.

# 2. Программа работы

- разработать простейший клиент и сервер на основе протоколов TCP и UDP
- разработать прикладной протокол в соответствии с индивидуальным заданием, реализовать протокол в виде клиент-серверного приложения на основе протоколов TCP и UDP
- выполнить дополнительное задание

# 3. Ход выполнения работы

## 3.1. Простейшие клиент и сервер

Простейшие клиент и сервер были выполнены на основе протоколов TCP и UDP, а также адаптированы под ОС Windows и Linux. Сервер выполняет функции эхо-сервера, т.е. принимает сообщения от клиентов и посылает копии обратно. Клиент посылает сообщение, после чего завершается.

## 3.2. Индивидуальное задание

В качестве индивидуального задания была выбрана система торгов. На торги выставляются лоты, имеющие начальную стоимость. Участники торгов могут повышать стоимость лота. Распорядитель может прекратить торги. При окончании торгов всем участникам рассылаются результаты торгов.

Серверное приложение реализует следующие функции:

- Прослушивание определенного порта
- Обработка запросов на подключение по этому порту от клиентов системы торгов (как распорядитель или участник торгов)
- Поддержка одновременной работы нескольких клиентов системы торгов через механизм нитей
- От участников торгов: прием запросов на передачу списка лотов
- От участников торгов: прием запросов на повышение стоимости лота
- От распорядителя: прием запроса на добавление нового лота с первоначальной стоимостью
- От распорядителя: прием запроса об окончании торгов
- Осуществление добавления лота, учет повышения стоимости лота участниками, завершение торгов и рассылка результатов торгов всем участникам
- Обработка запроса на отключение клиента

- Принудительное отключение клиента

Клиентское приложение реализует следующие функции:

- Установление соединения с сервером
- Регистрация в качестве распорядителя или участника
- Участник: передача запроса о выводе списка лотов
- Участник: передача запроса о повышении стоимости лота
- Распорядитель: передача запроса о добавлении лота
- Распорядитель: передача запроса о прекращении торгов
- Получение результатов торгов от сервера
- Разрыв соединения
- Обработка ситуации отключения клиента сервером

### 3.2.1. Реализация на ТСР

Для реализации данной системы был разработан текстовый асинхронный протокол. Максимальная длина сообщения 1000 символов.

**Описание протокола:** Сообщение клиента всегда содержит команду, определяющее тип сообщения. Некоторые типы сообщений содержат также поле опций. В конце каждого сообщения ставится символ \n.

Поле команды имеет размер до 5 байт. Поле опций может (в зависимости от команды) отсутствовать или содержать до 2 частей. Опции и команды разделяются пробелом.

- Команда для начала сеанса - new. Опций нет.
- Команда для авторизации - login. Опции - имя пользователя.
- Команда для получения списка лотов - 1. Опций нет.
- Команда для повышения ставки - bet. Опции - наименование лота, ставка.
- Команда для получения указаний по созданию нового лота - 2. Опций нет.
- Команда для создания нового лота - lot. Опции - наименование лота, начальная ставка.
- Команда для просмотра онлайн пользователей - 3. Опций нет.
- Команда на отключение от сервера - 4. Опций нет.
- Команда для окончания торгов - 5. Опций нет.

Сообщения сервера также всегда заканчиваются символом \n. Сервер может послать только один тип сообщения, который требует специальной обработки. Он начинается с символа # и означает отключение пользователя от сервера, далее следует текст с причиной отключения. Все остальные сообщения не требуют специальной обработки и просто выводятся пользователю.

**Описание программы:** Сервер имеет 1 слушающий порт, по которому принимает от клиентов запросы на соединение. При подсоединении очередного клиента, для связи с ним выделяется отдельный сокет, прием из которого осуществляется в отдельном потоке.

После подключения клиента сервер ожидает команды для авторизации нового пользователя. Остальные команды в этот момент для клиента недоступны. После авторизации сервер записывает его сокет, имя пользователя и информацию о типе клиента (распорядитель или участник).

Обработчик, в зависимости от соответствующего типа команды, формирует ответное сообщение пользователю.

Для управления сервером предусмотрен поток для опроса стандартного потока ввода. Его работа схожа с работой потока приема сообщений от пользователя. При вводе команды вызывается обработчик.

У клиента создаются два потока для отправки сообщений и принятия сообщений от сервера в асинхронном режиме.

### 3.2.2. Реализация на UDP

**Описание протокола:** Реализация прикладного протокола на UDP схожа с реализацией на TCP. Протокол отличается тем, что теперь после авторизации пользователя в начале сообщения передается id пользователя. Также теперь нельзя посмотреть онлайн-пользователей. Максимальное количество пользователей - 10, так как для id зарезервирован всего 1 байт.

После отправки команды на авторизацию клиент принимает сообщения от сервера с выделенным id для этого клиента. Это сообщение имеет длину 3 байта, последний байт и есть id клиента.

**Описание программы:** В отличие от варианта TCP, здесь не происходит установления соединения и все сообщения передаются через один сокет. Для каждого клиента больше не создается новый поток, поэтому при авторизации пользователя сервер отправляет клиенту его id, который клиент потом использует для его идентификации.

## 3.3. Дополнительное задание

В качестве дополнительного задания необходимо исследовать реальные прикладные протоколы. Необходимо "притвориться" клиентом и подключиться к одному из существующих общедоступных серверов.

В качестве утилиты для подключения к серверам была выбрана telnet.

### 3.3.1. Подключение к веб-серверу и запрос веб-страницы

Был произведен запрос веб-страницы с сервера tiger.ftk.spbstu.ru (рис. 3.2) Подключение производится по используемому протоколом http порту 80. Сервер вернул код 200 в заголовке ответа, что говорит об успешной обработке запроса.

```

lorismelik@lorismelik-Aspire-Z5700: ~
lorismelik@lorismelik-Aspire-Z5700:~$ telnet tiger.ftk.spbstu.ru 80
Trying 91.151.191.66...
Connected to tiger.ftk.spbstu.ru.
Escape character is '^J'.
GET / HTTP/1.1
Host: tiger.ftk.spbstu.ru

HTTP/1.1 200 OK
Cache-Control: must-revalidate
Content-Type: text/html; charset=utf-8
Content-Length: 8932
Date: Wed, 03 Jan 2018 17:51:34 GMT
Server: lighttpd/1.4.30

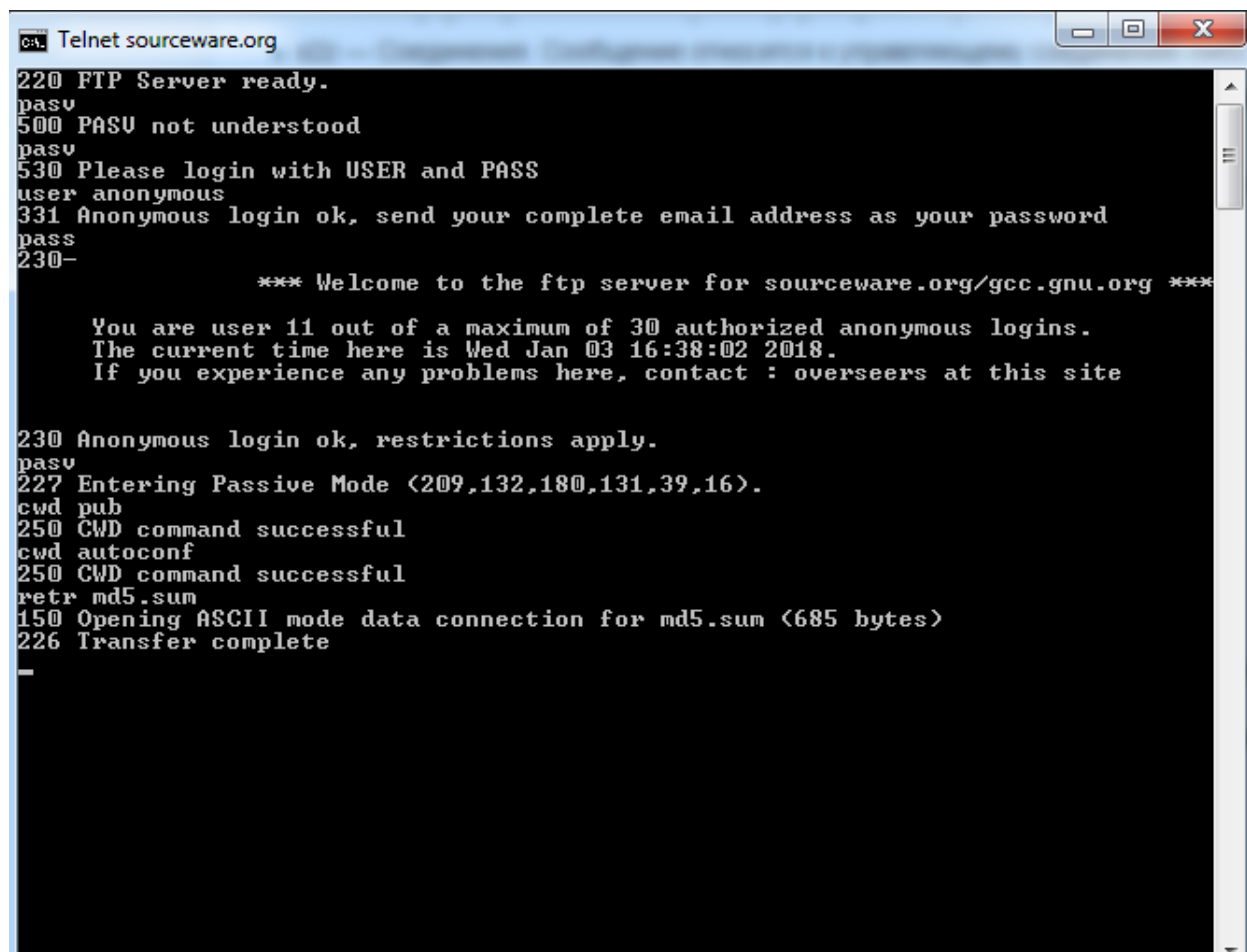
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Доступные проекты</title>
</head>
<body>
<h1>Доступные проекты</h1>
<ul>
<li>
<a href="/trac/aegisforsystemc" title="Static analysis methods for synchronization error detection in SystemC designs">AegisForSystemC</a>
</li><li>
<a href="/trac/diploma-2012" title="Дипломные проекты - 2012">Diploma 2012</a>
</li><li>
<a href="/trac/edu-se-2009-gps-navigator" title="GPS-навигатор">GPS-навигатор</a>
</li><li>
<a href="/trac/edu-se-2010-llvm-plugin" title="Haskell плагин к LLVM">Haskell плагин к LLVM</a>
</li><li>
<a href="/trac/jsa" title="JavaScript Dynamic and Static Analyzer">JavaScript Analyzer</a>
</li><li>
<a href="/trac/jdig" title="jDig: Java source code analysis & visualization tool">jDig</a>
</li><li>
<a href="/trac/llvm-analyzer" title="LLVM bitcode static analyzer in Haskell">LLVM Analyzer</a>
</li><li>
<a href="/trac/net-gui" title="Network Configuration GUI">Network Configuration GUI</a>
</li><li>
<a href="/trac/reliability-research" title="Разработка проекта &#34;Reliability Analyzer&#34;.>Reliability Research</a>
</li><li>
<a href="/trac/research" title="НИР аспирантов и магистрантов">Research</a>
</li><li>
<a href="/trac/sqleaf" title="SQLeaf">SQLeaf</a>
</li><li>
<a href="/trac/s2a" title="Static Software Analyzer">Static Software Analyzer</a>
</li><li>
<a href="/trac/vk-file-sharing" title="vk-file-sharing">vk-file-sharing</a>
</li><li>
<a href="/trac/practice-cfg-viewer" title="Визуализатор CFG">Визуализатор CFG</a>
</li>

```

Рисунок 3.1. Запрос веб-страницы

### 3.3.2. Загрузка файла с ftp-сервера

Протокол FTP использует 2 соединения - для передачи команд и для передачи данных. Поэтому подключение к нему производится в 2 этапа: сначала производится подключение к порту 21 (для передачи команд) и авторизация, затем переход в пассивный режим и подключение из другого терминала к порту, указанному сервером



```
CA: Telnet sourceware.org
220 FTP Server ready.
pasv
500 PASV not understood
pasv
530 Please login with USER and PASS
user anonymous
331 Anonymous login ok, send your complete email address as your password
pass
230-
      *** Welcome to the ftp server for sourceware.org/gcc.gnu.org ***

      You are user 11 out of a maximum of 30 authorized anonymous logins.
      The current time here is Wed Jan 03 16:38:02 2018.
      If you experience any problems here, contact : overseers at this site

230 Anonymous login ok, restrictions apply.
pasv
227 Entering Passive Mode (209,132,180,131,39,16).
cwd pub
250 CWD command successful
cwd autoconf
250 CWD command successful
retr md5.sum
150 Opening ASCII mode data connection for md5.sum (685 bytes)
226 Transfer complete
-
```

Рисунок 3.2. Загрузка файла с ftp-сервера

```
C:\Windows\system32\cmd.exe

c966dc72304c5e0fc0ad6694cd8685f7 autoconf-2.10-2.11.diff.gz
fe332d45a554c81bd5a1a758ea2c53be autoconf-2.10.tar.gz
8710adf0875a63acf831bc16ea17b9a4 autoconf-2.11-2.12.diff.gz
f0b5091d33a2d928b2e89b6d33db2efb autoconf-2.11.tar.gz
d96301bc0135b2d9f35026bb80d43528 autoconf-2.12-2.13.diff.gz
8d7a2b2eda07601308c3031197c78b8a autoconf-2.12.tar.gz
9de56d4a161a723228220b0f425dc711 autoconf-2.13.tar.gz
cab18748a71005c7df5591f8b125600d autoconf-2.7-2.8.diff.gz
3f7838eab23d34f58096c644628440f0 autoconf-2.7-2.9.diff.gz
ac1203d9708adb48416d598c9062f7fc autoconf-2.8-2.9.diff.gz
662cb6ece7a5809be5f4a86020516f15 autoconf-2.9-2.10.diff.gz
d9f2eccf891a9b4572a1a6e1dc2c46ea sha512.sum

Подключение к узлу утеряно.
C:\Users\Timur>
```

Рисунок 3.3. Загрузка файла с ftp-сервера

### 3.3.3. Отправка письма на SMTP-сервер

Попробуем авторизоваться на SMTP-сервере gmail





```
lorismelik@lorismelik-Aspire-Z5700: ~  
0010 - 13 35 1f 24 36 7b 7e bc-17 72 ea cd ac 04 d5 10 .5,$6{~..r.....  
0020 - b4 99 33 08 db bc 4d 41-ea ac 78 80 a2 1a b8 7f ..3...MA..X.....  
0030 - dd d7 f5 5b 74 e3 54 e0-8b 89 97 d1 fc dc d5 b0 ...[t.T.....  
0040 - 54 28 a9 00 3a 2c aa 5a-e5 9e 7a dd 34 65 81 c8 T(...,.Z..z.4e..  
0050 - 2b 60 7e f0 ab 7b 73 2b-41 90 06 f9 ac ed 2b e5 +~..{s+A.....+..  
0060 - cf 53 62 11 b8 82 02 ea-c8 2d ab 36 40 79 f6 0f .Sb.....60y..  
0070 - 5b 9e f3 63 3b ec d3 ac-e8 e7 15 56 e3 3a 28 ad [...c;.....V.:(..  
0080 - 32 d8 2a c5 dd f2 20 24-ca cd ec 91 58 02 85 95 2.*...$....X...  
0090 - 93 21 19 e9 fd c1 5b 08-d2 26 b7 c4 d1 ec 00 19 .!....[.&.....  
  
Start Time: 1515003843  
Timeout : 7200 (sec)  
Verify return code: 0 (ok)  
Extended master secret: no  
---  
250 ENHANCEDSTATUSCODES  
ehlo a  
250-smtp4j.mail.yandex.net  
250-8BITIME  
250-PIPELINING  
250-SIZE 42991616  
250-AUTH LOGIN PLAIN XOAUTH2  
250-DSN  
250 ENHANCEDSTATUSCODES  
auth login  
334 VXNlcm5hbWU6  
[REDACTED]  
334 UGFzc3dvcmQ6  
[REDACTED]  
235 2.7.0 Authentication successful.  
mail from:<timur.erniyazov@salesplatform.ru>  
250 2.1.0 <timur.erniyazov@salesplatform.ru> ok  
rcpt to:<lorismelik@yandex.ru>  
250 2.1.5 <lorismelik@yandex.ru> recipient ok  
subject: message for lab  
502 5.5.2 Syntax error, command unrecognized.  
data  
354 Enter mail, end with "." on a line by itself  
subject: message for lab  
  
Timur Erniyazov  
'  
250 2.0.0 Ok: queued on smtp4j.mail.yandex.net as 1515004022-z5HHLeKazR-PCief7WS  
quit  
221 2.0.0 Closing connection.  
read:errno=0  
lorismelik@lorismelik-Aspire-Z5700:~$
```

Рисунок 3.5. Отправка письма на SMTP-сервер через TLS-подключение

### 3.3.4. Получение письма с POP3-сервера

Проверим почту и получим письмо



```

lorismelik@lorismelik-Aspire-Z5700: ~
Issuer=/C=RU/O=Yandex LLC/OU=Yandex Certification Authority/CN=Yandex CA
---
No client certificate CA names sent
---
SSL handshake has read 4513 bytes and written 494 bytes
Verification: OK
---
New, TLSv1.2, Cipher is AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol : TLSv1.2
    Cipher   : AES256-GCM-SHA384
    Session-ID: 72E916D124AE41718958826299A42D52FBDB4F04394E983B00696E81EF90208B
    Session-ID-ctx:
    Master-Key: C2AFEFB8177B7F1785D3277AC11E670C66E7B4AC8A4A478CF2F8C6699AA6DB028AA90C4F81A86DAAAADB30E788358187
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 3f e7 a3 ad b1 81 40 2f ae 85 a2 bf 4a 3f b6 58 ?.....@/....J?.X
    0010 - 7f fa 6e 42 86 71 25 be c0 35 54 94 e2 70 3c 03 ..nB.q%..5T..p<.
    0020 - 55 22 d9 87 42 52 17 de 06 39 41 7d a7 83 1e 98 U"...BR...9A}....
    0030 - 87 ec de 78 af bb 5e 3e 1d 29 eb 58 78 15 0e 48 ...x...^>..).Xx..H
    0040 - a6 fd 94 e6 0f 22 5c 17 4f 9b e7 67 13 9c 9f 6e ....."\.0..g...n
    0050 - 52 7b ab 4c 56 39 8d c4 14 3e a5 3d 24 a3 bc 3f R{.LV9...>.=$.?
    0060 - f7 54 d5 bf d5 9c 38 2c e1 6e a8 3a 25 ba 8c 9b .T....8,.n.:%...
    0070 - 41 d4 33 03 c3 7b cd cb 35 de 65 f3 e0 3b 63 34 A.3..{(.S.e...;c4
    0080 - c9 02 e1 7c 53 76 ca 65 c1 a3 bd 62 52 8e 10 c3 ...|Sv.e...bR...
    0090 - 9c db ed bf ab 33 57 3d a8 e9 1b f5 3a 4d a0 8d .....3W=.....M..

    Start Time: 1515003487
    Timeout : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
---
+OK POP Ya! naq55p 7Ie7DUwdG8c1
user lorismelik@yandex.ru
+OK password, please.
pass 
-ERR [AUTH] login failure or POP3 disabled, try later. sc=7Ie7DUwdG8c1_031818_55p
read:errno=0
lorismelik@lorismelik-Aspire-Z5700:~$

```

Рисунок 3.7. POP3

Yandex отключил поддержку pop3 протокола, поэтому проверить почту не удалось.

## 4. Выводы

В ходе работы был разработан и реализован в виде приложения прикладной протокол. В результате этого были изучены принципы программирования сокетов TCP и UDP. Основной проблемой при реализации приложения на TCP была необходимость контроля длины послышки. Ее решением стало добавление символа окончания послышки. TCP требует установления соединения, поэтому на сервере выделяется поток, в котором происходит прием запросов на соединение от клиентов через выделенный для этого сокет. После подключения очередного клиента порождается отдельный поток, осуществляющий обмен пакетами с этим клиентом через отдельный сокет.

В реализации на UDP сервер обменивается пакетами со всеми клиентами в одном потоке и через один сокет, т.к. нет установления соединения. Для идентификации пользователей в сообщение добавилось поле id.

Также были исследованы прикладные протоколы. Как выяснилось, почтовые сервера могут требовать обязательного использования защищенного подключения.