

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

**Отчёт по лабораторной работе**

**Дисциплина:** Разработка сетевых приложений

**Тема:** Изучение прикладных протоколов в командной строке Linux

Выполнил студент гр. 43501/3  
Преподаватель

Мальцев М.С.  
Зозуля А.В.

Санкт-Петербург  
18 февраля 2019 г.

# 1 SMTP

## 1.1 Основные сведения о протоколе

Simple Mail Transfer Protocol (SMTP) - это интернет протокол для обмена почтой. Впервые был зафиксирован в RFC 821 в 1982 году. Был изменен в 2008 году RFC 5321. Используется по сегодняшний день.

## 1.2 Основные команды

Команды SMTP:

- EHLO - используется для приветствия в протоколе ESMTP. Рекомендуется к использованию по возможности
- HELO - команда приветствия. Используется для начала сессии
- MAIL FROM - задается адрес отправителя
- RCPT TO - указывается получатель
- DATA - текст сообщения
- QUIT - выход
- HELP - список доступных команд или описание запрашиваемой команды
- NOOP - пустая команда, предположительно используется для поддержания соединения
- RSET - сброс

## 1.3 Область применения и ограничения протокола

Простой протокол передачи почты (SMTP), используется для связи с удаленным сервером и последующей отправке сообщений с локального клиента на удаленный сервер, и в конечном итоге на сервер получателя сообщений. На вашем сервере электронной почты, этот процесс контролируется специальной службой (MTA). Стоит упомянуть, что SMTP используется исключительно для отправки сообщений.

Порты SMTP:

Порт 25 – порт без шифрования

Порт 465 – порт SSL/TLS, также известный как SMTPS

## 1.4 Пример использования

```
1 openssl s_client -connect smtp.yandex.ru:465
2      [...]
3 220 smtp4o.mail.yandex.net ESMTP (Want to use Yandex.Mail for your domain? Visit
    http://pdd.yandex.ru)
4 helo host
5 250 smtp4o.mail.yandex.net
6 auth login
7 334 VXNlcm5hbWU6
8 byolo1234Tk3
9 334 UGFzc3dvcmQ6
10 N3XoyolobHlyolouamyoloRjyoloQm123jM=
11 235 2.7.0 Authentication successful.
12 mail from: <mikle9997@yandex.ru>
13 250 2.1.0 <mikle9997@yandex.ru> ok
14 rcpt to: <mikle9997@yandex.ru>
15 250 2.1.5 <mikle9997@yandex.ru> recipient ok
16 data
17 354 Enter mail, end with "." on a line by itself
18 From: mikle9997@ya.ru
19 To: mikle9997@ya.ru
20 Subject: Hello world!
21
22 This is the test message...
23 new line
24 .
25 250 2.0.0 Ok: queued on smtp4o.mail.yandex.net as 1550417106-ffkNcyAJLD-NlO4FHGx
26 quit
27 221 2.0.0 Closing connection.
28 read:errno=0
```

Hello world!



**mikle9997@ya.ru** mikle9997@ya.ru

сегодня в 18:25



Вам: mikle9997@ya.ru ^

This is the test message...

new line

Рис. 1.1: Сообщение отправленное с использованием протокола smtp

## 2 POP3

### 2.1 Основные сведения о протоколе

POP3 (англ. Post Office Protocol Version 3 — протокол почтового отделения, версия 3) — стандартный интернет-протокол прикладного уровня, исполь-

зубый клиентами электронной почты для получения почты с удалённого сервера по TCP-соединению.

## 2.2 Основные команды

Команды POP3:

- USER - передаёт серверу имя пользователя
- PASS - передаёт серверу пароль почтового ящика.
- DELE - сервер помечает указанное сообщение для удаления. Сообщения, помеченные на удаление, реально удаляются только после закрытия транзакции (закрытие транзакций происходит обычно после посылки команды QUIT, кроме этого, например, на серверах закрытие транзакций может происходить по истечении определённого времени, установленного сервером).
- LIST - если был передан аргумент, то сервер выдаёт информацию об указанном сообщении. Если аргумент не был передан, то сервер выдаёт информацию обо всех сообщениях, находящихся в почтовом ящике. Сообщения, помеченные для удаления, не перечисляются
- NOOP - сервер ничего не делает, всегда отвечает положительно
- RETR сообщение - сервер передаёт сообщение с указанным номером
- RSET - этой командой производится откат транзакций внутри сессии. Например, если пользователь случайно пометил на удаление какие-либо сообщения, он может убрать эти пометки, отправив эту команду
- STAT - сервер возвращает количество сообщений в почтовом ящике и размер почтового ящика в октетах. Сообщения, помеченные как удалённые, при этом не учитываются.
- TOP - сервер возвращает заголовки указанного сообщения, пустую строку и указанное количество первых строк тела сообщения.

## 2.3 Область применения и ограничения протокола

POP3 (протокол почтового отделения версия 3) часто используется для связи с удалённым сервером электронной почты и загрузки сообщений на локальный почтовый клиент с последующим удалением его на сервере, к примеру Outlook, Thunderbird, Windows Mail, Mac Mail и т.д. Однако обычно почтовые клиенты предлагают выбор – оставлять или нет копии сообщений на

сервере. Если вы используете несколько устройств для отправки сообщений, то рекомендуется оставлять эту функцию включенной, в противном случае, на другом устройстве у вас не будет доступа к отправленным сообщениям, которые не были сохранены на удаленном сервере. Также стоит отметить, что POP3 – протокол работающий только в одном направлении, это означает, что данные берутся с удаленного сервера и отправляются на локальный клиент.

Порты POP3, по умолчанию являются такими:

Порт 110 – порт без шифрования

Порт 995 – порт SSL/TLS, также известный как POP3S

## 2.4 Пример использования

```
1 openssl s_client -connect pop.yandex.ru:995
2     [...]
3 +OK POP Ya! na@8o 4vVwfiwuP0U1
4 user mikle9997@yandex.ru
5 +OK password, please.
6 pass yoloyoloyoloyoloyolo
7 +OK 141 8600172
8 stat
9 +OK 141 8600172
10 list
11 +OK 141 8600172
12 1 1714
13 2 8768
14     [...]
15 140 7116
16 141 9267
17 .
18 noop
19 +OK noop
20 top 1 0
21 +OK 1714 octets.
22 X-Yandex-FolderName: Inbox
23 Received: from mxback13j.mail.yandex.net ([127.0.0.1])
24     by mxback13j.mail.yandex.net with LMTP id SsFBoMrt
25     for <mikle9997@yandex.ru>; Sun, 17 Feb 2019 18:25:06 +0300
26 Received: from mxback13j.mail.yandex.net (localhost.localdomain [127.0.0.1])
27     by mxback13j.mail.yandex.net (Yandex) with ESMTP id 2F97268C1381
28     for <mikle9997@yandex.ru>; Sun, 17 Feb 2019 18:25:06 +0300 (MSK)
29 X-Yandex-Internal: 1
30 Received: from smtp4o.mail.yandex.net (smtp4o.mail.yandex.net [2a02:6b8:0:1a2d
31     ::28])
32     by mxback13j.mail.yandex.net (nwsmtп/Yandex) with ESMTP id MbcW9nq0Ig-
33     P6TuLGrX;
34     Sun, 17 Feb 2019 18:25:06 +0300
35 X-Yandex-Front: mxback13j.mail.yandex.net
36 X-Yandex-TimeMark: 1550417106
37 DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=yandex.ru; s=mail; t
38     =1550417106;
39     bh=oMGQU4hVaHd+pvDa94cg5wNw8zPI8KCe9zxGqUAP3Wg=;
40     h=Message-Id:Date:From:To:Subject;
41     b=Se3jbrZfgY6Nz0YpKkEJS2uiPxYB/HhciwFWJUJtLP0mYlgcUYsy897kDLz5nKPY9
42     otEKZCclw7lVgda/KhUck60jFCeqcIchBtZzzzHQS3RJUCXbTjrcw3Cfw0NR5WsC1o
```

```
40      x6n7SMIZe4AyNKjusoPI/X3VT1xCo3ddYQ9VaaGE=
41 Authentication-Results: mxback13j.mail.yandex.net; dkim=pass header.i=@yandex.ru
42 Received: by smtp4o.mail.yandex.net (nsmtp/Yandex) with SMTP id ffkNcyAJLD-
      NlO4FHGx;
43      Sun, 17 Feb 2019 18:23:55 +0300
44      (using TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits))
45      (Client certificate not present)
46 X-Yandex-Front: smtp4o.mail.yandex.net
47 X-Yandex-TimeMark: 1550417035
48 Message-Id: <20190217182505.NlO4FHGx@smtp4o.mail.yandex.net>
49 Date: Sun, 17 Feb 2019 18:25:05 +0300
50 X-Yandex-Spam: 1
51 From: mikle9997@ya.ru
52 To: mikle9997@ya.ru
53 Subject: Hello world!
54 Return-Path: mikle9997@yandex.ru
55 .
56 quit
57 +OK shutting down.
58 read:errno=0
```

## 3 IMAP

### 3.1 Основные сведения о протоколе

IMAP (англ. Internet Message Access Protocol) — протокол прикладного уровня для доступа к электронной почте.

IMAP предоставляет пользователю обширные возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

### 3.2 Основные команды

Команды IMAP:

- LOGIN - позволяет клиенту при регистрации на сервере IMAP использовать идентификатор пользователя и пароль в обычном текстовом виде. Это не самый лучший метод, но иногда это единственная возможность подключиться к серверу
- CLOSE - закрывает почтовый ящик. Когда почтовый ящик закрыт с помощью этой команды, то сообщения, помеченные флагом DELETED, удаляются из него. Не имеет параметров

- LOGOUT - завершает сеанс для текущего идентификатора пользователя
- DELETE - применяется к почтовым ящикам. Сервер IMAP при получении этой команды попытается удалить почтовый ящик с именем, указанным в качестве аргумента команды. Сообщения удаляются вместе с ящиками и восстановлению не подлежат
- LIST - получить список всех почтовых ящиков клиента; имеет два параметра
- LSUB - в отличие от команды LIST используется для получения списка ящиков, активизированных командой SUBSCRIBE. Параметры — такие же, как у LIST
- EXPUNGE - удаляет из почтового ящика все сообщения, помеченные флагом DELETED, при этом почтовый ящик не закрывается. Ответ сервера на команду EXPUNGE представляет собой отчёт о новом состоянии почтового ящика
- FETCH - получить текст почтового сообщения. Команда применяется только для отображения сообщений. В отличие от POP3, клиент IMAP не сохраняет копию сообщения на клиентском ПК
- SEARCH - поиск сообщений по критериям в активном почтовом ящике с последующим отображением результатов в виде номера сообщения. Возможен поиск сообщений, в теле которых имеется определённая текстовая строка, или имеющих определённый флаг, или полученных до определённой даты и т.д.
- NOOP - команда ничего не делает

### 3.3 Область применения и ограничения протокола

IMAP, также как и POP3 используется для получения сообщений электронной почты на локальный клиент, однако, он имеет существенное отличие — загружаются только лишь заголовки электронных сообщений, сам текст письма остается на сервере. Данный протокол связи работает в две стороны, если происходят изменения на локальном клиенте, они передаются и на сервер. В последнее время IMAP стал более популярным, так как такие гиганты-провайдеры услуг электронной почты, как Gmail, стали рекомендовать использовать его вместо POP3.

Для отправки писем используется обычно протокол SMTP, так как собственная команда отправки протокола IMAP, называемая APPEND, считается «неудачной» и «небезопасной»

Порты IMAP, по умолчанию являются такими:

Порт 143 – порт без шифрования

Порт 993 – порт SSL/TLS, также известный как IMAPS

### 3.4 Пример использования

```
1 openssl s_client -crlf -connect imap.yandex.ru:993
2 [...]
3 * OK Yandex IMAP4rev1 at imap39p.mail.yandex.net:993 ready to talk with ::ffff
   :95.161.223.174:54762, 2019-Feb-18 14:02:24, O2QGNH1kRCg1
4 a1 login mikle9997@ya.ru yolooyooyooyooyooyo
5 * CAPABILITY IMAP4rev1 CHILDREN UNSELECT LITERAL+ NAMESPACE XLIST BINARY UIDPLUS
   ENABLE ID IDLE MOVE
6 a1 OK login Completed.
7 a2 list "" "*"
8 * list (\Unmarked \HasNoChildren) "|" "&BEAEMARBBEEESwQ7BDoeOA-"
9 * list (\Unmarked \HasNoChildren \Drafts) "|" Drafts
10 * list (\Unmarked \NoInferiors) "|" INBOX
11 * list (\Unmarked \HasNoChildren) "|" Outbox
12 * list (\Unmarked \HasNoChildren \Sent) "|" Sent
13 * list (\Unmarked \HasNoChildren \Junk) "|" Spam
14 * list (\Unmarked \HasNoChildren \Trash) "|" Trash
15 a2 OK list Completed.
16 a3 examine inbox
17 * FLAGS (\Answered \Seen \Draft \Deleted $Forwarded)
18 * 141 EXISTS
19 * 0 RECENT
20 * OK [PERMANENTFLAGS (\Answered \Seen \Draft \Flagged \Deleted $Forwarded \*)]
   Limited
21 * OK [UIDNEXT 251] Ok
22 * OK [UIDVALIDITY 1436432001] Ok
23 a3 OK [READ-ONLY] examine Completed.
24 a4 fetch 1 (body[])
25 a4 BAD Command syntax error. sc=O2QGNH1kRCg1_181102_39p
26 a4 fetch 1 body[]
27 * 1 FETCH (BODY[] {7088})
28 Received: from mxfront8j.mail.yandex.net ([127.0.0.1])
29   by mxfront8j.mail.yandex.net with LMTP id PtQm2XTJ
30   for <mikle9997@yandex.ru>; Sat, 29 Aug 2015 12:24:01 +0300
31 Received: from smtp01.steampowered.com (smtp01.steampowered.com [208.64.202.37])
32   by mxfront8j.mail.yandex.net (nwsmtpt/Yandex) with ESMTPS id sqAbBBv7gm-
   O1XKofxX;
33   Sat, 29 Aug 2015 12:24:01 +0300
34   (using TLSv1.2 with cipher DHE-RSA-AES256-SHA256 (256/256 bits))
35   (Client certificate not present)
36 X-Yandex-Front: mxfront8j.mail.yandex.net
37 X-Yandex-TimeMark: 1440840241
38 Authentication-Results: mxfront8j.mail.yandex.net; spf=pass (mxfront8j.mail.
   yandex.net: domain of steampowered.com designates 208.64.202.37 as permitted
   sender) smtp.mail=noreply@steampowered.com; dkim=pass header.i=
   @steampowered.com
39 X-Yandex-Spam: 1
40 DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed; d=steampowered.
   com; s=smtp;
41   h=Date:Message-Id:Content-Type:Subject:MIME-Version:Reply-To:From:To; bh=
   B3jH2v/RpG5AlyF1Wi68I3dCHOVg/oNdwBl07vUZDq0=;
42   b=MiqYzcJumq+B3xyjj9RX+4
   QVlrI5xHK5cMGyR1505p1udnwlJloHsRy7JWJMVgKtdatPdvLPuAGGGkhwFDxZM5a+
```



```

    rOkGNNiU3Sw4HhKJ/Sgm+3grcUwv/
    NFJxkE7cWrT6tojZlS7v7gm5rgXzxc4l9wzSnLbOqPILQ6g8/yGcUo=;
43 Received: from [10.3.3.29] (helo=valvesoftware.com)
44     by smtp01.steampowered.com with smtp (Exim 4.82)
45     (envelope-from <noreply@steampowered.com>)
46     id 1ZVcMh-0007zD-Lv
47     for mikle9997@yandex.ru; Sat, 29 Aug 2015 02:23:59 -0700
48 To: mikle9997@yandex.ru
49 From: Steam Support <noreply@steampowered.com>
50 Reply-To: <noreply@steampowered.com>
51 X-Steam-Message-Type: Account Information Confirmation
52 MIME-Version: 1.0
53 Subject: Your Steam account: Email address verification
54 Content-Type: multipart/alternative;
55     boundary="-----060908020109090601040503"
56 Message-Id: <E1ZVcMh-0007zD-Lv@smtp01.steampowered.com>
57 Date: Sat, 29 Aug 2015 02:23:59 -0700
58 Return-Path: noreply@steampowered.com
59 X-Yandex-Forward: 7b0d2314e1a45660ce6bff40e8f54e2c
60
61 This is a multi-part message in MIME format.
62 -----060908020109090601040503
63 Content-Type: text/plain; charset=UTF-8; format=flowed
64 Content-Transfer-Encoding: 7bit
65
66 In order to help maintain the security of your Steam account, please verify your
    email
67     [...]
68 )
69 a4 OK fetch Completed.
70 a5 logout
71 * BYE IMAP4rev1 Server logging out
72 a5 OK LOGOUT completed
73 read:errno=0

```

## 4 HTTP

### 4.1 Основные сведения о протоколе

HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных. Изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных.

### 4.2 Основные команды

Команды HTTP:

- OPTIONS - используется для определения возможностей веб-сервера или параметров соединения для конкретного ресурса
- GET - используется для запроса содержимого указанного ресурса

- HEAD - аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело
- POST - применяется для передачи пользовательских данных заданному ресурсу
- PUT - применяется для загрузки содержимого запроса на указанный в запросе URI
- DELETE - удаляет указанный ресурс

### 4.3 Область применения и ограничения протокола

HTTP — протокол прикладного уровня; аналогичными ему являются FTP и SMTP. Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URI. В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ». Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами (например, «куки» на стороне клиента, «сессии» на стороне сервера). Браузер, посылающий запросы, может отслеживать задержки ответов. Сервер может хранить IP-адреса и заголовки запросов последних клиентов. Однако сам протокол не осведомлён о предыдущих запросах и ответах, в нём не предусмотрена внутренняя поддержка состояния, к нему не предъявляются такие требования.

### 4.4 Пример использования

```

1 openssl s_client -connect yandex.ru:443
2     [...]
3 GET / HTTP/1.1
4 Host: yandex.ru
5
6 HTTP/1.1 200 Ok
7 Date: Mon, 18 Feb 2019 13:04:56 GMT
8 Content-Type: text/html; charset=UTF-8
9 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
10 Expires: Mon, 18 Feb 2019 13:04:56 GMT
11 Last-Modified: Mon, 18 Feb 2019 13:04:56 GMT
12     [...]
```

## 5 FTP

### 5.1 Основные сведения о протоколе

FTP (англ. File Transfer Protocol) — протокол передачи файлов по сети. Гарантирует передачу (либо выдачу ошибки) за счёт применения квитируемого протокола TCP. FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, и даже до TCP/IP, в 1971 году. В первое время он работал поверх протокола NCP. Он и сегодня широко используется для распространения ПО и доступа к удалённым хостам.

### 5.2 Основные команды

Команды FTP:

- ABOR — прервать передачу файла
- CDUP — сменить каталог на вышестоящий
- CWD — сменить каталог
- DELE — удалить файл (DELE filename)
- EPSV — войти в расширенный пассивный режим. Применяется вместо PASV
- HELP — выводит список команд, принимаемых сервером
- LIST — возвращает список файлов каталога. Список передаётся через соединение данных
- MDTM — возвращает время модификации файла
- MKD — создать каталог
- NLST — возвращает список файлов каталога в более кратком формате, чем LIST. Список передаётся через соединение данных
- NOOP — пустая операция
- PASS — пароль
- PASV — войти в пассивный режим. Сервер вернёт адрес и порт, к которому нужно подключиться, чтобы забрать данные. Передача начнётся при введении следующих команд: RETR, LIST и т.д.

- PORT — войти в активный режим. Например PORT 12,34,45,56,78,89. В отличие от пассивного режима для передачи данных сервер сам подключается к клиенту
- PWD — возвращает текущий каталог
- QUIT — отключиться
- REIN — реинициализировать подключение
- RETR — скачать файл. Перед RETR должна быть команда PASV или PORT
- RMD — удалить каталог
- RNFR и RNT0 — переименовать файл. RNFR — что переименовывать, RNT0 — во что
- SIZE — возвращает размер файла
- STOR — закачать файл. Перед STOR должна быть команда PASV или PORT
- SYST — возвращает тип системы (UNIX, WIN, ...)
- TYPE — установить тип передачи файла (бинарный, текстовый)
- USER — имя пользователя для входа на сервер

### 5.3 Область применения и ограничения протокола

Типичное применение FTP-протокола — загрузка сайтов и других документов с частного устройства разработки на общедоступные сервера хостинга. Протокол FTP (как и HTTP) имеет двоичный режим передачи, что сокращает накладные расходы трафика и уменьшает время обмена данными при передаче больших файлов.

Стандартный порт управления FTP-соединением — 21.

### 5.4 Пример использования

```

1 telnet ftp.sunet.se 21
2 Trying 194.71.11.173...
3 Connected to sunet.ftp.acc.umu.se.
4 Escape character is '^]'.
5 220 Please use http://ftp.acc.umu.se/ whenever possible.
6 user anonymous
7 331 Please specify the password.
8 pass mikle9997@yandex.ru

```

```

9 230 Login successful.
10 pwd
11 257 "/"
12 pasv
13 227 Entering Passive Mode (194,71,11,173,33,198).
14 list
15 150 Here comes the directory listing.
16 226 Directory send OK.
17 pasv
18 227 Entering Passive Mode (194,71,11,173,213,168).
19 retr robots.txt
20 150 Opening BINARY mode data connection for robots.txt (1859 bytes).
21 226 Transfer complete.
22 quit
23 221 Goodbye.
24 Connection closed by foreign host.

```

```

1 telnet 194.71.11.173 8646
2 Trying 194.71.11.173...
3 Connected to 194.71.11.173.
4 Escape character is '^]'.
5 -rw-r--r--      1 ftp      ftp      1216 Jun 02  2017 HEADER.html
6 lrwxrwxrwx      1 ftp      ftp           3 Mar 16  2010 Public -> pub
7 drwxr-xr-x      3 ftp      ftp          10 Jun 30  2017 about
8 drwxr-sr-x     30 ftp      ftp          33 Feb 18 10:25 cdimage
9 drwxr-xr-x      2 ftp      ftp           3 Jun 14  2006 conspiracy
10 lrwxrwxrwx      1 ftp      ftp          22 Mar 16  2010 debian -> cdimage/.
    debian-mirror
11 drwxr-xr-x      5 ftp      ftp           7 Mar 13  2016 debian-backports
12 lrwxrwxrwx      1 ftp      ftp          16 Mar 16  2010 debian-cd -> cdimage/
    release/
13 -rw-r--r--      1 ftp      ftp     15086 Apr 02  2018 favicon.ico
14 drwxr-xr-x     82 ftp      ftp           89 Feb 04 20:28 mirror
15 drwxr-xr-x      4 ftp      ftp           12 Feb 05  2016 pub
16 lrwxrwxrwx      1 ftp      ftp          23 Mar 16  2010 releases -> mirror/
    ubuntu-releases/
17 -rw-r--r--      1 ftp      ftp      1859 Mar 23  2017 robots.txt
18 lrwxrwxrwx      1 ftp      ftp           12 Aug 01  2016 tails -> mirror/tails
19 lrwxrwxrwx      1 ftp      ftp           13 Mar 16  2010 ubuntu -> mirror/ubuntu
20 Connection closed by foreign host.
21
22 telnet 194.71.11.173 54696
23 Trying 194.71.11.173...
24 Connected to 194.71.11.173.
25 Escape character is '^]'.
26 # robots.txt for http://ftp.acc.umu.se/
27 # This file specifies what harvesting robots are allowed to index and not.
28 #
29 # For information on the original format:
30 # http://www.robotstxt.org/
31 #
32 # For information on updated format that allows wildcards:
33 # https://developers.google.com/webmasters/control-crawl-index/docs/robots_txt
34
35 User-agent: *
36 # Rules without wildcards - original spec
37 Disallow: /cdimage/.debian-mirror
38         [...]
39 Disallow: /*.dmg$
40 Connection closed by foreign host.

```

## 6 TFTP

### 6.1 Основные сведения о протоколе

TFTP (англ. Trivial File Transfer Protocol — простой протокол передачи файлов) используется главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP, в отличие от FTP, не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP.

### 6.2 Основные команды

Основные команды протокола:

- GET — получить файл с сервера. Существует возможность указания имени файла на клиенте.
- PUT — загрузить один или несколько файлов в указанный файл или директорию.
- STATUS — показать текущее состояние.
- MODE — указать режим передачи: `ascii` или `binary`. По умолчанию — `ascii`.

### 6.3 Область применения и ограничения протокола

Основное назначение TFTP — обеспечение простоты реализации клиента. В связи с этим он используется для загрузки бездисковых рабочих станций, загрузки обновлений и конфигураций в «умные» сетевые устройства, записи статистики с мини-АТС (CDR) и аппаратных маршрутизаторов/файрволов.

### 6.4 Пример использования

```
1 nc -ul localhost 1917
2 #/home/miklenetascii#/home/miklenetascii#/home/miklenetascii#/home/miklenetascii
  #/home/miklenetascii
```

```
1 tftp localhost 1917
2 tftp> get /home/mikle 1917
3 Transfer timed out.
4
5 tftp> quit
```

## 7 WebDAV

### 7.1 Основные сведения о протоколе

WebDAV (Web Distributed Authoring and Versioning) или просто DAV — набор расширений и дополнений к протоколу HTTP, поддерживающих совместную работу пользователей над редактированием файлов и управление файлами на удаленных веб-серверах. В качестве миссии рабочей группы по созданию DAV было заявлено: "разработка дополнений к протоколу HTTP, обеспечивающих свободное взаимодействие инструментов распределенной разработки веб-страниц, в соответствии с потребностями работы пользователей". Однако в процессе эксплуатации DAV нашёл себе ряд других применений, выходящих за первоначально принятые рамки коллективной работы над веб-документами.

### 7.2 Основные команды

WebDAV расширяет HTTP следующими методами запроса:

- PROPFIND — Получение свойств объекта на сервере в формате XML. Также можно получать структуру репозитория (дерево каталогов);
- PROPPATCH — Изменение свойств за одну транзакцию;
- MKCOL — Создать коллекцию объектов (каталог в случае доступа к файлам);
- COPY — Копирование из одного URI в другой;
- MOVE — Перемещение из одного URI в другой;
- LOCK — Поставить блокировку на объекте. WebDAV поддерживает эксклюзивные и общие (shared) блокировки;
- UNLOCK — Снять блокировку с ресурса.

### 7.3 Область применения и ограничения протокола

Сегодня DAV применяется в качестве сетевой файловой системы, эффективной для работы в Интернете и способной обрабатывать файлы целиком, поддерживая хорошую производительность работы в условиях окружения с высокой временной задержкой передачи информации. Кроме того, DAV широко применяется в качестве протокола для доступа через Интернет и манипулирования содержимым систем документооборота (document management).

system). Ещё одной важной целью DAV является поддержка работы распределенных команд по разработке программного обеспечения. В качестве резюме задачу создания DAV можно указать так: на волне повсеместного использования HTTP в качестве стандартного уровня доступа к широкому кругу хранилищ информации расширить его возможности средствами записи информации (HTTP — доступ на чтение, DAV — доступ на запись).

## 7.4 Пример использования

```
1 mikle@mikle-Lenovo-V580c:~$ echo Hello world! > text
2 mikle@mikle-Lenovo-V580c:~$ ll text
3 -rw-r--r-- 1 mikle mikle 13 фев 18 17:26 text
4 mikle@mikle-Lenovo-V580c:~$ cat text
5 Hello world!
6 mikle@mikle-Lenovo-V580c:~$ curl -X MKCOL 'https://webdav.yandex.ru/test_folder '
   --user "mikle9997@ya.ru:yoloyoloyoloyolo"
7 mikle@mikle-Lenovo-V580c:~$ curl -T ~/text https://webdav.yandex.ru/test_folder /
   test_file --user "mikle9997@ya.ru:yoloyoloyoloyolo"
8 mikle@mikle-Lenovo-V580c:~$ curl https://webdav.yandex.ru/test_folder/text^C-
   user "mikle9997@ya.ru:yoloyoloyoloyolo"
9 mikle@mikle-Lenovo-V580c:~$ rm text
10 mikle@mikle-Lenovo-V580c:~$ cd Downloads/
11 mikle@mikle-Lenovo-V580c:~/Downloads$ curl https://webdav.yandex.ru/test_folder /
   test_file --user "mikle9997@ya.ru:yoloyoloyoloyolo"
12 Hello world!
```

## 8 Вывод

В ходе работы были использованы следующие протоколы: SMTP, POP3, IMAP, HTTP, FTP, TFTP, WebDAV.

Учитывая имеющиеся теоретические знания о работе рассматриваемых протоколов и небольшой опыт их использования, проделанная работа позволяет лучше проникнуться тонкостями работы с почтовыми серверами и веб-сервисами. В итоге работы были получены навыки и укреплены знания, благодаря которым разработка приложений с использованием рассматриваемых протоколов не вызовет больших сложностей.

Выполнение подобных заданий является полезным и необходимым действием перед написанием собственного приложения использующего указанные протоколы.