

## Протокол FTP

FTP (File Transfer Protocol) – протокол передачи файлов по TCP-сетям. Протокол построен по архитектуре клиент-сервер и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. FTP является старейшим протоколом, который появился ещё в 1971 году, задолго до появления протокола HTTP.

Протокол FTP разделяет поток данных на команды и, собственно, данные. Разделение происходит для того, чтобы не держать открытое TCP-соединение ради коротких команд. Команду выгодно быстро обработать и закрыть соединение. Передаваемые же данные (файлы) как правило, большие и задержка при их передаче не так существенна и не так заметна. Зато играет роль пропускная способность канала. Было бы удобно, если бы для передачи команд использовался быстрый, но тонкий канал (с малым временем отклика и небольшой пропускной способностью), а для передачи данных использовался бы медленный, но толстый канал (с большим временем отклика, но пропускной способностью).

Для того, чтобы каналы можно было различить, передача данных происходит по инициативе сервера. То есть сервер подключается к клиенту для передачи данных. Сначала клиент подключается к серверу по 20-му порту (управляющий порт) и передает ему команду на «взятие данных», сообщая при этом номер порта (порт с произвольным номером) на котором будет ждать эти данные (файл). Сервер подключается с 21-го порта (порт данных) к порту на клиенте (к порту, который до этого указал клиент) и пересылает данные (запрошенный файл). После передачи канал разрывается, и клиент перестает обрабатывать данные на этом порту.

Если клиент находится за сетевым экраном, то могут возникнуть трудности с передачей данных, так как не каждый системный администратор согласится открывать любой порт клиента при запросе с 21-го порта – это может повредить безопасности при работе в сети. Для преодоления этой ситуации придумали протокол Passive FTP. В этом протоколе оба соединения и для команд, и для данных устанавливает клиент. Клиент с помощью команды с 20-го порта «пассивно» запрашивает данные, а сервер выделяет произвольный порт, подключившись к которому клиент забирает данные. В этом случае, если со стороны сервера запрещено подключаться по любому порту, то передача данных будет не возможна. Это недостаток протокола FTP.

Ещё одним недостатком протокола является то, что передача данных об идентификации пользователя происходит открытым (не зашифрованным) образом.

Некоторые особенности FTP: основан на устанавливаемых сессиях работы, использует аутентификацию пользователей, предусмотрен в основном для передачи больших двоичных данных, использует двойное подключение, приспособлен и для приема и для передачи данных,

поддерживает и текстовый и двоичный режим передачи, поддерживает операции над файловой системой.

**Активный режим.** В активном режиме клиент создаёт управляющее TCP-соединение с сервером и отправляет серверу свой IP-адрес и произвольный номер клиентского порта, после чего ждёт, пока сервер не запустит TCP-соединение с этим адресом и номером порта.

**Пассивный режим.** В этом режиме клиент использует поток управления, чтобы послать серверу команду PASV, и затем получает от сервера его IP-адрес и номер порта, которые затем используются клиентом для открытия потока данных с произвольного клиентского порта к полученному адресу и порту.

**Аутентификация.** Для аутентификации пользователя используется обычная схема логин/пароль. Логин посылается серверу командой USER, а пароль командой PASS. Если предоставленная информация совпадает с настройками сервера, то для пользователя открывается сессия.

**Анонимный доступ.** Как правило, многие FTP-сервера предоставляют пользователям анонимный доступ, без использования логина и пароля. Тогда в качестве логина необходимо указать «anonymous». При анонимном режиме работы, как правило, существуют определенные ограничения.

Синтаксис запроса к FTP следующий:

`ftp://[<пользователь>[:<пароль>]@]<хост>[:<порт>]/<путь>`

При этом параметры, указанные в квадратных скобках, не обязательны.  
Примеры:

- `ftp://public.ftp-server.ru` – обращение к корневому каталогу FTP-сервера;
- `ftp://public.ftp-server.ru/doc/file.txt` – обращение к конкретному файлу на FTP-сервере;
- `ftp://public.ftp-server.ru:21` – обращение к корневому каталогу FTP-сервера с указанием порта;
- `ftp://login:password@private.ftp-server.ru` – обращение к корневому каталогу FTP-сервера с одновременной аутентификацией;
- `ftp://login:password@private.ftp-server.ru/doc/file.txt:21` – обращение к конкретному файлу на FTP-сервере с одновременной аутентификацией и указанием порта.

## **FTP-клиент**

FTP-клиент – это программа позволяющая получить доступ к своему (или общему) аккаунту на сервере при помощи протокола FTP. FTP-клиент позволяет: закачивать и скачивать папки и файлы с сервера, создавать, переименовывать, удалять папки и файлы, редактировать файлы, менять права доступа к ним.

Настройка FTP-клиента как правило проста и требует указания адреса сервера, имени и пароля пользователя.

Существует огромное количество как платных, так и бесплатных FTP-клиентов. Сравнить различные программы-клиенты FTP можно, например, здесь:

- [http://ru.wikipedia.org/wiki/Сравнение\\_FTP\\_клиентов](http://ru.wikipedia.org/wiki/Сравнение_FTP_клиентов)
- [http://www.3dnews.ru/software/ftp\\_client\\_for\\_home\\_use](http://www.3dnews.ru/software/ftp_client_for_home_use)

Остановимся на бесплатном кроссплатформенном FTP-клиенте FileZilla. Официальный сайт: <http://filezilla.ru/>. Для установки этого клиента можно скачать необходимые пакеты с официального сайта, однако пакеты с filezilla хранятся так же и в репозиториях многих дистрибутивов операционной системы Linux, поэтому, установить эту программу можно следующим образом:

```
aptitude install filezilla
```

Запустить установленную программу можно либо из консоли:

```
filezilla
```

либо, нажав Alt+F2, ввести название filezilla и нажать Enter.

Для доступа к серверу минимально необходимо указать адрес сервера, логин и пароль пользователя.

## FTP-сервер

FTP-сервер – это программа, запущенная на компьютере и работающая в режиме сервиса. Компьютер, на котором запущен подобный сервис становится FTP-сервером способным отдавать и принимать файлы.

Существует огромное количество как платных, так и бесплатных FTP-серверов. Сравнить различные программы-клиенты FTP можно, например, здесь:

- [http://en.wikipedia.org/wiki/List\\_of\\_FTP\\_server\\_software](http://en.wikipedia.org/wiki/List_of_FTP_server_software)
- <http://www.3dnews.ru/573906>

Остановимся на бесплатном FTP-сервере ProFTPD.

ProFTPD - это сервер FTP, первоначально разрабатываемый для различных систем семейства Unix, хотя теперь он будет собираться и под Win32. Он разрабатывался таким образом, чтобы быть похожим по своей концепции на Apache, переняв от последнего множество идей (формат конфигурационного файла, модульный дизайн и т.д.).

Официальный сайт: <http://ftp.proftpd.org/>. Для установки этого сервера можно скачать необходимые пакеты с официального сайта, однако пакеты с ProFTPD хранятся так же и в репозиториях многих дистрибутивов операционной системы Linux, поэтому, установить эту программу можно следующим образом:

```
aptitude install proftpd
```

Выбираем тип запуска «самостоятельно» (standalone) - proftpd будет работать как самостоятельная служба, а не через службу inetd.

Затем начинаем исправлять файл конфигурации proftpd.conf:

```
nano /etc/proftpd/proftpd.conf
```

Ниже приведены некоторые настройки параметров конфигурации proftpd:

```
# Включить или выключить поддержку IPv6.
  UseIPv6                off
# Имя сервера, которое будет отображаться при
соединении с proftpd.
  ServerName              my-FTP-server
# Сервер работает как самостоятельная служба.
  ServerType              standalone
# «DefaultRoot ~» - опция которая запирает
пользователя в его домашнем каталоге и не дает ему
подняться выше. Если мы храним данные пользователей на
ftp-сервере не в домашнем каталоге пользователя, а в
каком-то другом каталоге, то в опции DefaultRoot нужно
указать этот каталог и указать пользователя, который
привязан к этому каталогу. Такой каталог для
пользователя ftp-сервера становится корневым.
  DefaultRoot             /home/ftp/user1 user1
  DefaultRoot             /home/ftp/user2 user2
  DefaultRoot             ~
# Стандартный порт на который proftpd принимает
запросы.
  Port                    21
# Опция, позволяющая перезаписывать файл, если её
установить в off, то нельзя будет заменить одноименный
файл
  AllowOverwrite          on
# Пользователь и группа от которого запускается сервер
  User                    proftpd
  Group                   nogroup
```

После исправления файла конфигурации добавляем в систему пользователей. В нашем примере это два пользователя user1 и user2.

```
adduser user1
```

и

```
adduser user2
```

Если необходимо создать пользователя, который бы не мог обращаться

к серверу по ssh, то можно воспользоваться командой `adduser user1 --shell /bin/false`.

Созданных пользователей присоединяем к группе `nogroup`. Делается это следующим образом. Открываем файл `/etc/group` для редактирования, например, так:

```
nano /etc/group
```

находим в этом файле строку `nogroup:x:65534:` и добавляем к ней через запятую пользователей: `nogroup:x:65534:user1,user2` (здесь цифры 65534 могут быть другими).

Создаем каталоги ftp-сервера в которых пользователи будут хранить свои данные. Эти каталоги должны совпадать с теми, что указаны в файле `proftpd.conf` в параметре `DefaultRoot`.

```
mkdir -p /home/ftp/user1  
mkdir -p /home/ftp/user2
```

Изменяем пользователя и группу у созданных каталогов:

```
chown root:nogroup /home/ftp/user1  
chown root:nogroup /home/ftp/user2
```

Изменяем для этих каталогов права доступа:

```
chmod 775 /home/ftp/user1  
chmod 775 /home/ftp/user2
```

Перезапускаем сервер `proftpd` командой

```
service proftpd restart
```

Всё, теперь можно попробовать подключаться к серверу, например, при помощи клиента `FileZilla`, указав ip-адрес сервера, имя пользователя (например, `user1`) и его пароль.