## Команда wget

Работая в консоли, нам достаточно часто приходится взаимодействовать с сетью, например скачивать файлы или целые интернет страницы. Если интернет страницы мы скачиваем довольно редко, то с файлами дело обстоит совсем по-другому. Это могут быть различные скрипты, установочные пакеты, программы, ключи, списки пакетов и многое другое. Скачать файл в консоли Linux можно с помощью утилиты wget.

Это очень мощная утилита, способная работать по протоколам HTTP, HTTPS и FTP. Кроме того, поддерживается работа через прокси. Команда wget, может выполнять загрузку файлов даже в фоновом режиме - без участия пользователя, в отличии от большинства веб браузеров.

Кроме скачивания файлов, есть возможность сохранять веб страницы или даже целые веб-сайты, благодаря функции открытия ссылок на страницах. Такую возможность еще называют рекурсивной загрузкой.

## Команда curl

На самом деле, curl — это больше, чем просто утилита командной строки для Linux или Windows. Это набор библиотек, в которых реализуются базовые возможности работы с URL страницами и передачи файлов. Библиотека поддерживает работу с протоколами: FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, LDAP, а также POP3, IMAP и SMTP. Она отлично подходит для имитации действий пользователя на страницах и других операций с URL адресами.

Поддержка библиотеки curl была добавлена в множество различных языков программирования и платформ. Утилита curl - это одна из утилит, использующих эту библиотеку.

## cUrl vs wget

## Что делают обе команды

* Обе команды – это утилиты командной строки, которые могут загружать данные по протоколам FTP, HTTP(S)
* Обе команды могут посылать запросы HTTP POST
* Обе поддерживают HTTP cookies
* Обе спроектированы для работы без непосредственного взаимодействия с пользователем, например, вызовом из скриптов
* Обе – полностью свободны и с открытым исходным кодом (хотя и под разными лицензиями)
* Оба проекта имеют длительную историю и зарождались еще в 90-е.

## Чем они отличаются:

### Curl

* Библиотека. Утилита curl работает на базе библиотеки libcurl – кроссплатформенной библиотеки со стабильным API, которым могут пользоваться кто угодно, в том числе и другие утилиты.
* Конвейеры. Curl работает больше как команда cat, посылая данные в стандартный вывод и принимая из стандартного ввода в манере «всё как конвейер» - имеется в виду, конвейер Unix/Linux. Утилита wget похожа больше на команду cp.
* Команда однократного действия. Curl заточена под выполнение однократной передачи данных – передает данные лишь по тому URL, который указывает пользователь, и не содержит ни логики рекурсивной выгрузки, ни возможностей парсинга HTML.
* Больше поддерживаемых протоколов. Curl поддерживает FTP, FTPS, Gopher, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS, FILE, POP3, IMAP, SMB/CIFS, SMTP, RTMP и RTSP. А Wget – только HTTP, HTTPS и FTP.
* Лучше переносимость. Curl работает на большем количестве платформ, чем wget. Например, OS/400 и другие экзотические платформы, не являющиеся клонами или форками UNIX.
* Более обширная поддержка SSL и поддержка большего числа SSL-библиотек – до 11 штук.
* Большее количество поддерживаемых методов аутентификации HTTP, особенно через HTTP-прокси: Basic, Digest, NTLM и Negotiate
* Поддержка SOCKS-протоколов для доступа через прокси.
* Полностью двунаправленный обмен данными. В отличие от curl, wget поддерживает только метод HTTP POST. С другой стороны, есть проект wput, но это совершенно отдельная утилита и проект, никак не связанные с wget, поэтом сравнивать wget+wput с curl не совсем правильно.
* Поддержка отправки данных multipart/form-data, что дает пользователям возможность загружать на веб-ресурсы бинарные данные и в целом лучше эмулировать поведение браузера и выполнять некоторую автоматизацию для протокола HTTP
* Curl поддерживает gzip и может выполнять автоматическую декомпрессию загруженных файлов.
* Curl умеет распаковывать сжатые Tranfer-Encoded HTTP-файлы на лету.
* Curl поддерживает HTTP/2 и умеет работать с алгоритмом Happy Eyeballs, что дает возможность работать в режиме dual stack – с доступом по протоколам IPv4 и IPv6 одновременно.
* Активность разработчиков выше – активность в списках рассылки, частота коммитов и релизов.

### Wget

* Это просто консольная утилита. Без библиотек и API.
* Поддержка рекурсивной выгрузки! Это основная сильная сторона Wget – возможность выгрузить всё, что обнаружится в ссылках на удаленном ресурсе, будь то HTML-страница или директория на FTP-сервере.
* Старше, хотя и всего на год (1995 против начала разработки в 1996 у curl)
* GPL – wget на 100% GP v3, в то время как wget лицензирован MIT.
* GNU – wget является частью проекта GNU и права принадлежат Free Software Foundation, в то время как curl разрабатывался независимо.
* C wget чуть проще скачать файл по ссылке в текущую директорию – для этого не нужно никаких дополнительных аргументов, в то время как curl требует аргументов -o или -O
* Wget поддерживает только библиотеки GnuTLS и OpenSSL для поддержки SSL/TLS
* Через HTTP прокси поддерживается только базовая аутентификация HTML
* Не поддерживаются SOCKS-прокси
* Возможность возобновить прерванную загрузку, в отличие от curl
* Wget по умолчанию поддерживает многое из того, что в curl нужно в явном виде включать ключами в команде либо другими способами: cookies, переход по редирект-ссылкам, сохранение временных меток (timestamp) с удаленного ресурса и т.д.

## Примеры wget

### Синтаксис

Команда wget linux имеет очень простой синтаксис:

$ wget опции адрес\_ссылки

Можно указать не один URL для загрузки, а сразу несколько. Опции указывать не обязательно, но в большинстве случаев они используются для настройки параметров загрузки.

### Опции

Синтаксис опций очень свободный. У каждой опции, как правило есть как длинное, так и короткое имя. Их можно записывать как до URL, так и после. Между опцией и ее значением не обязательно ставить пробел, например вы можете написать -o log или -olog. Эти значения эквивалентны. Также если у опций нет параметров, не обязательно начинать каждую с дефиса, можно записать их все вместе: -drc и -d -r -c. Эти параметры wget тоже эквивалентны.

А теперь давайте перейдем к списку опций. У wget слишком много опций, мы разберем только основные.

* -V (--version) - вывести версию программы
* -h (--help) - вывести справку
* -b (--background) - работать в фоновом режиме
* -o файл (--out-file) - указать лог файл
* -d (--debug) - включить режим отладки
* -v (--verbose) - выводить максимум информации о работе утилиты
* -q (--quiet) - выводить минимум информации о работе
* -i файл (--input-file) - прочитать URL из файла
* --force-html - читать файл указанный в предыдущем параметре как html
* -t (--tries) - количество попыток подключения к серверу
* -O файл (--output-document) - файл в который будут сохранены полученные данные
* -с (--continue) - продолжить ранее прерванную загрузку
* -S (--server-response) - вывести ответ сервера
* --spider - проверить работоспособность URL
* -T время (--timeout) - таймаут подключения к серверу
* --limit-rate - ограничить скорость загрузки
* -w (--wait) - интервал между запросами
* -Q (--quota) - максимальный размер загрузки
* -4 (--inet4only) - использовать протокол ipv4
* -6 (--inet6only) - использовать протокол ipv6
* -U (--user-agent)- строка USER AGENT, отправляемая серверу
* -r (--recursive)- рекурсивная работа утилиты
* -l (--level) - глубина при рекурсивном сканировании
* -k (--convert-links) - конвертировать ссылки в локальные при загрузке страниц
* -P (--directory-prefix) - каталог, в который будут загружаться файлы
* -m (--mirror) - скачать сайт на локальную машину
* -p (--page-requisites) - во время загрузки сайта скачивать все необходимые ресурсы

### Установка

Команда wget linux, обычно поставляется по умолчанию в большинстве дистрибутивов, но если нет, ее можно очень просто установить. Например установка с помощью yum будет выглядеть следующим образом:

yum -y install wget

А в дистрибутивах основанных на Debian:

sudo apt install wget

Теперь перейдем непосредственно к примерам:

### Загрузка файла

Команда wget linux скачает один файл и сохранит его в текущей директории. Во время загрузки мы увидим прогресс, размер файла, дату его последнего изменения, а также скорость загрузки:

wget http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz

[](https://losst.ru/wp-content/uploads/2016/05/wget1.png)

### Сохранить файл с другим именем

Опция -О позволяет задать имя сохраняемому файлу, например, скачать файл wget с именем wget.zip:

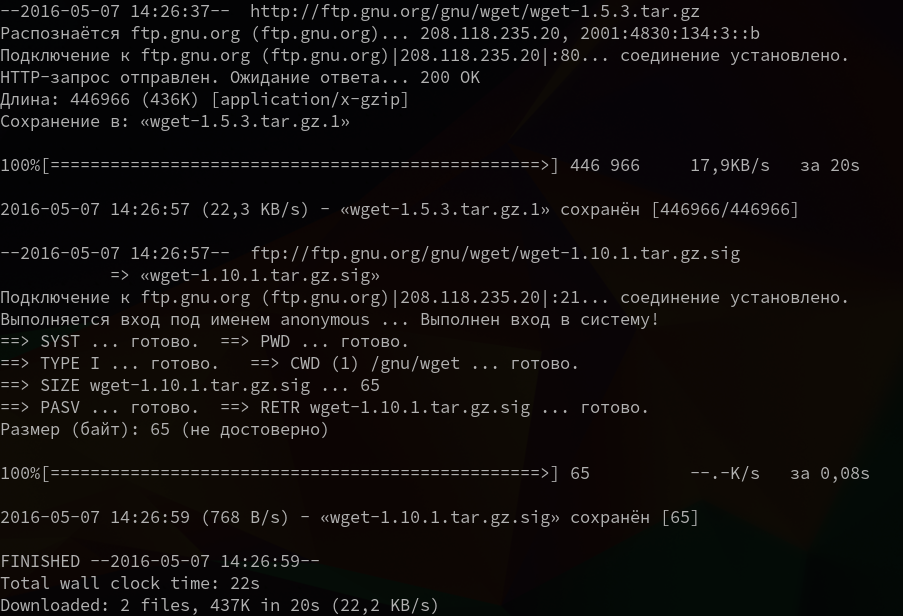
wget -O wget.zip http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz

[](https://losst.ru/wp-content/uploads/2016/05/wget2.png)

### Скачать несколько файлов

Вы можете скачать несколько файлов одной командой даже по разным протоколам, просто указав их URL:

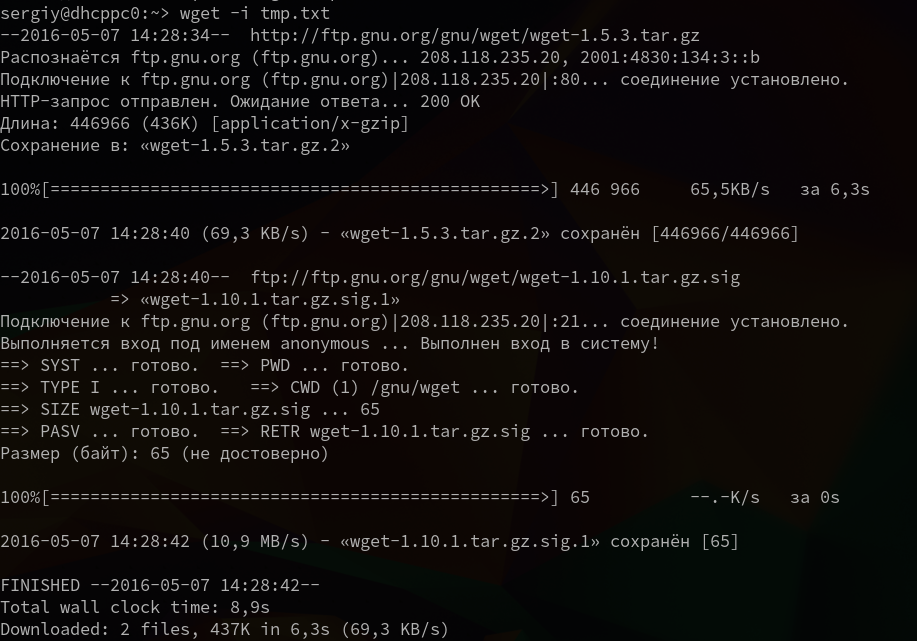
wget http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz ftp://ftp.gnu.org/gnu/wget/wget-1.10.1.tar.gz.sig

[](https://losst.ru/wp-content/uploads/2016/05/wget3.png)

### Взять URL из файла

Вы можете сохранить несколько URL в файл, а затем загрузить их все, передав файл опции -i. Например создадим файл tmp.txt, со ссылками для загрузки wget, а затем скачаем его:

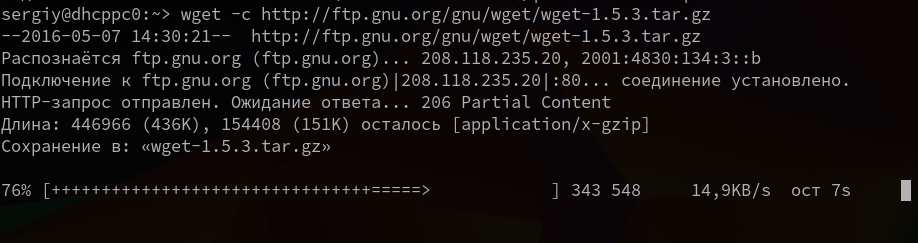
wget -i /wget/tmp.txt

[](https://losst.ru/wp-content/uploads/2016/05/wget4.png)

### Продолжить загрузку

Утилита wget linux рассчитана на работу в медленных и нестабильных сетях. Поэтому если вы загружали большой файл, и во время загрузки было потеряно соединение, то вы можете скачать файл wget с помощью опции -c.

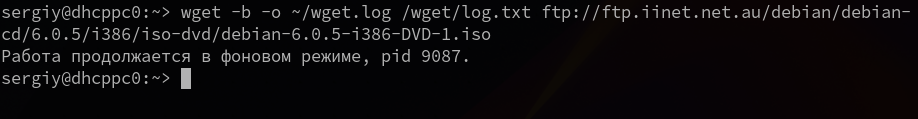
wget -c http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz

[](https://losst.ru/wp-content/uploads/2016/05/wget5-e1462621578180.png)

### Загрузка файлов в фоне

Опция -b заставляет программу работать в фоновом режиме, весь вывод будет записан в лог файл, для настройки лог файла используются специальные ключи wget:

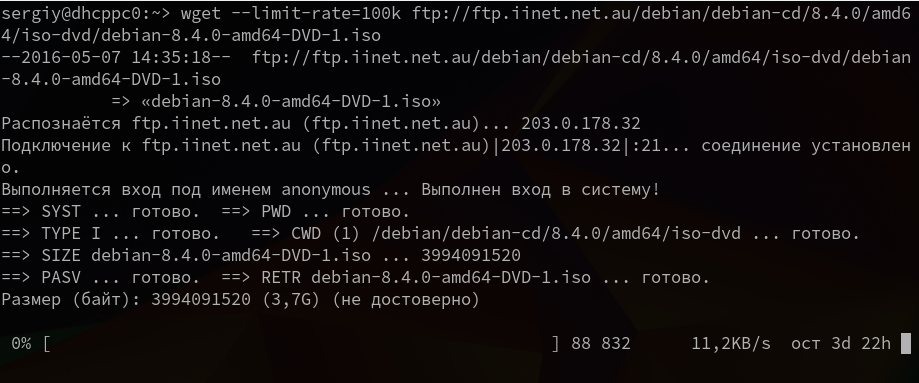
wget -b -o ~/wget.log http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz

[](https://losst.ru/wp-content/uploads/2016/05/wget6.png)

### Ограничение скорости загрузки

Команда wget позволяет не только продолжать загрузку файлов, но и ограничивать скорость загрузки. Для этого есть опция --limit-rate. Например, ограничим скорость до 100 килобит:

wget --limit-rate=100k ftp://ftp.iinet.net.au/debian/debian-cd/8.4.0/amd64/iso-dvd/debian-8.4.0-amd64-DVD-1.iso

[](https://losst.ru/wp-content/uploads/2016/05/wget7.png)

Здесь доступны, как и в других подобных командах индексы для указания скорости - k - килобит, m - мегабит, g - гигабит, и так далее.

### Подключение по логину и паролю

Некоторые ресурсы требуют аутентификации, для загрузки их файлов. С помощью опций --http-user=username, –http-password=password и --ftp-user=username, --ftp-password=password вы можете задать имя пользователя и пароль для HTTP или FTP ресурсов.

wget --http-user=narad --http-password=password http://mirrors.hns.net.in/centos/6.3/isos/x86\_64/CentOS-6.3-x86\_64-LiveDVD.iso

Или:

wget --ftp-user=narad --ftp-password=password ftp://ftp.iinet.net.au/debian/debian-cd/6.0.5/i386/iso-dvd/debian-6.0.5-i386-DVD-1.iso

### Загрузить и выполнить

Вы, наверное, уже видели такие команды. Wget позволяет сразу же выполнять скачанные скрипты:

wget -O - http://сайт/скрипт.sh | bash

Если опции -O не передать аргументов, то скачанный файл будет выведен в стандартный вывод, затем мы его можем перенаправить с помощью интерпретатора bash, как показано выше.

### Сохранить файл в папке

По умолчанию wget сохраняет файл в текущую папку, но это поведение очень легко изменить с помощью опции -P:

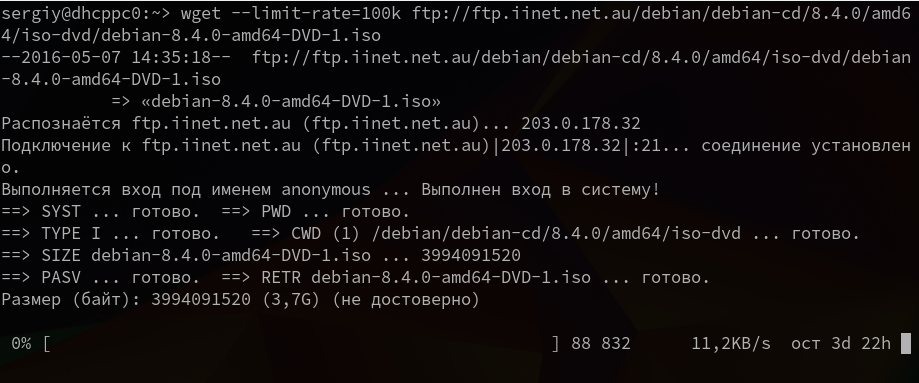
wget -P ~/Downloads/ http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz

[](https://losst.ru/wp-content/uploads/2016/05/wget8.png)

### Передать информацию о браузере

Некоторые сайты фильтруют ботов, но мы можем передать фальшивую информацию о нашем браузере (user-agent) и страницу с которой мы пришли (http-referer).

wget ‐‐refer=http://google.com ‐‐user-agent=”Mozilla/5.0 Firefox/4.0.1″ //losst.ru

[](https://losst.ru/wp-content/uploads/2016/05/wget7-1.png)

### Количество попыток загрузки

По умолчанию wget пытается повторить загрузку 20 раз, перед тем как завершить работу с ошибкой. Количество раз можно изменить с помощью опции --tries:

wget --tries=75 http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86\_64/CentOS-7.0-1406-x86\_64-DVD.iso

### Квота загрузки

Если вам доступно только ограниченное количество трафика, вы можете указать утилите, какое количество информации можно скачивать, например разрешим скачать файлов из списка только на десять мегабайт:

wget -Q10m -i download-list.txt

Здесь работают те же индексы для указания размера - k, m, g, и т д.

### Скачать сайт

Wget позволяет не только скачивать одиночные файлы, но и целые сайты, чтобы вы могли их потом просматривать в офлайн. Использование wget, чтобы скачать сайт, выглядит вот так:

wget --mirror -p --convert-links -P ./<Local-Folder> адрес\_сайта

### Еще примеры

|  |  |
| --- | --- |
| Команда | Описание |
| wget http://example.com/file.zip | скачивание файла file.zip в текущую директорию |
| wget -P /path/to/save http://example.com/file.zip | скачивание файла file.zip в директорию /path/to/save |
| wget -c http://example.com/file.zip | докачивание файла file.zip в случаи обрыва |
| wget -O arch.zip http://example.com/file.zip | скачивание файла file.zip и сохранение под именем arch.zip |
| wget -i files.txt | скачивание файлов из списка в files.txt |
| wget --tries=10 http://example.com/file.zip | количество попыток на скачивание |
| wget -Q5m -i http://example.com/ | квота на максимальный размер скачанных файлов, квота действует только при рекурсивном скачивании (-r) |
| wget --save-cookies cookies.txt --post-data 'username=proft&password=1' http://example.com/auth.php | идентификация на сервере с сохранением кук для последующего доступа |
| wget --user-agent="Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.9 Safari/536.5" http://example.com/ | указание User Agent |
| echo 'wget http://example.com/file.zip' | at 12:00 | скачать http://example.com/file.zip в 12:00 в текущую директорию |
| wget ftp://example.com/dir/\*.zip | скачивание всех файлов по шаблону |
| wget http://example.com/dir/file{1..10}.zip | скачивание всех файлов по шаблону |
| wget -S http://example.com/ | вывод заголовков HTTP серверов и ответов FTP серверов |
| wget --spider -i urls.txt | проверка ссылок в файле на доступность |
| wget -b http://example.com/file.zip | скачивание файла в фоне, лог пишется в wget.log, wget.log.1 и т.д. |
| export http\_proxy=http://proxy.com:3128/;wget http://example.com/file.zip | скачивание файла \*file.zip\* через прокси |
| wget -m -w 2 http://example.com/ | зеркалирование сайта с сохранением абсолютных ссылок и ожиданием 2-х секунд между запросами |
| wget --limit-rate=200k http://example.com/file.zip | ограничение скорости скачивания |
| wget -R bmp http://example.com/ | не скачивать bmp файлы |
| wget -A png,jpg http://example.com/ | скачивать только файлы png и jpg |

Пример использования для скачивания документации Django:

wget -r -k -l 5 -p -E -nc -np https://docs.djangoproject.com/en/1.5/

* -r - ходим по ссылкам (рекурсивное скачивание)
* -k - преобразовываем ссылки к локальному виду
* -p - скачивание ресурсов необходимых для отображения html-страницы (стили, картинки и т.д.)
* -l - глубина скачивания, *0* - бесконечная вложенность ссылок
* -nc - не перезаписывать существующие файлы
* -np - не подниматься выше начального адреса при рекурсивной загрузке

Часто используемые **wget** параметры можно вынести в [~/.wgetrc](http://www.gnu.org/software/wget/manual/html_node/Wgetrc-Commands.html).

## Примеры curl

### Синтаксис

Перед тем как перейти к описанию того как может использоваться команда curl linux, давайте разберем саму утилиту и ее основные опции, которые нам понадобятся. Синтаксис утилиты очень прост:

$ curl опции ссылка

### Опции

Теперь рассмотрим основные опции:

* -# - отображать простой прогресс-бар во время загрузки;
* -0 - использовать протокол HTTP 1.0;
* -1 - использовать протокол шифрования TLSv1;
* -2 - использовать SSLv2;
* -3 - использовать SSLv3;
* -4 - использовать IPv4;
* -6 - использовать IPv6;
* -A - указать свой USER\_AGENT;
* -b - сохранить Cookie в файл;
* -c - отправить Cookie на сервер из файла;
* -C - продолжить загрузку файла с места разрыва или указанного смещения;
* -m - максимальное время ожидания ответа от сервера;
* -d - отправить данные методом POST;
* -D - сохранить заголовки, возвращенные сервером в файл;
* -e - задать поле Referer-uri, указывает с какого сайта пришел пользователь;
* -E - использовать внешний сертификат SSL;
* -f - не выводить сообщения об ошибках;
* -F - отправить данные в виде формы;
* -G - если эта опция включена, то все данные, указанные в опции -d будут передаваться методом GET;
* -H - передать заголовки на сервер;
* -I - получать только HTTP заголовок, а все содержимое страницы игнорировать;
* -j - прочитать и отправить cookie из файла;
* -J - удалить заголовок из запроса;
* -L - принимать и обрабатывать перенаправления;
* -s - максимальное количество перенаправлений с помощью Location;
* -o - выводить контент страницы в файл;
* -O - сохранять контент в файл с именем страницы или файла на сервере;
* -p - использовать прокси;
* --proto - указать протокол, который нужно использовать;
* -R -  сохранять время последнего изменения удаленного файла;
* -s - выводить минимум информации об ошибках;
* -S - выводить сообщения об ошибках;
* -T - загрузить файл на сервер;
* -v - максимально подробный вывод;
* -y - минимальная скорость загрузки;
* -Y - максимальная скорость загрузки;
* -z - скачать файл, только если он был модифицирован позже указанного времени;
* -V - вывести версию.

### Загрузка файлов

Самая частая задача — это загрузка файлов. Скачать файл очень просто. Для этого достаточно передать утилите в параметрах имя файла или HTML страницы:

curl https://raw.githubusercontent.com/curl/curl/master/README.md

Но тут вас ждет одна неожиданность, все содержимое файла будет отправлено на стандартный вывод. Чтобы записать его в какой-либо файл используйте:

curl -o readme.txt https://raw.githubusercontent.com/curl/curl/master/README.md

А если вы хотите, чтобы полученный файл назывался так же, как и файл на сервере, используйте опцию -O:

curl -O https://raw.githubusercontent.com/curl/curl/master/README.md

Если загрузка была неожиданно прервана, вы можете ее возобновить:

curl -# -C - -O https://cdn.kernel.org/pub/linux/kernel/v4.x/testing/linux-4.11-rc7.tar.xz

Если нужно, одной командой можно скачать несколько файлов:

curl -O https://raw.githubusercontent.com/curl/curl/master/README.md -O https://raw.githubusercontent.com/curl/curl/master/README

Еще одна вещь, которая может быть полезной администратору - это загрузка файла, только если он был изменен:

curl -z 21-Dec-17 https://raw.githubusercontent.com/curl/curl/master/README.md -O https://raw.githubusercontent.com/curl/curl/master/README

Данная команда скачает файл, только если он был изменен после 21 декабря 2017.

### Ограничение скорости

Вы можете ограничить скорость загрузки до необходимого предела, чтобы не перегружать сеть с помощью опции -Y:

curl --limit-rate 50K -O https://cdn.kernel.org/pub/linux/kernel/v4.x/testing/linux-4.11-rc7.tar.xz

Здесь нужно указать количество килобайт в секунду, которые можно загружать. Также вы можете разорвать соединение если скорости недостаточно, для этого используйте опцию -Y:

curl -Y 100 -O https://raw.githubusercontent.com/curl/curl/master/README.md

### Передача файлов

Загрузка файлов, это достаточно просто, но утилита позволяет выполнять и другие действия, например, отправку файлов на ftp сервер. Для этого существует опция -T:

curl -T login.txt ftp://speedtest.tele2.net/upload/

Или проверим отправку файла по HTTP, для этого существует специальный сервис:

curl -T ~/login.txt http://posttestserver.com/post.php

В ответе утилита сообщит где вы можете найти загруженный файл.

### Отправка данных POST

Вы можете отправлять не только файлы, но и любые данные методом POST. Напомню, что этот метод используется для отправки данных различных форм. Для отправки такого запроса используйте опцию -d. Для тестирования будем пользоваться тем же сервисом:

curl -d "field1=val&fileld2=val1"http://posttestserver.com/post.php

Если вас не устраивает такой вариант отправки, вы можете сделать вид, что отправили форму. Для этого есть опция -F:

curl -F "password=@pass;type=text/plain" http://posttestserver.com/post.php

Здесь мы передаем формой поле password, с типом обычный текст, точно так же вы можете передать несколько параметров.

### Передача и прием куки

Куки или Cookie используются сайтами для хранения некой информации на стороне пользователя. Это может быть необходимо, например, для аутентификации. Вы можете принимать и передавать Cookie с помощью curl. Чтобы сохранить полученные Cookie в файл используйте опцию -c:

curl -c cookie.txt http://posttestserver.com/post.php

Затем можно отправить cookie curl обратно:

curl -b cookie.txt http://posttestserver.com/post.php

### Передача и анализ заголовков

Не всегда нам обязательно нужно содержимое страницы. Иногда могут быть интересны только заголовки. Чтобы вывести только их есть опция -I:

curl -I https://losst.ru

А опция -H позволяет [отправить нужный заголовок](https://losst.ru/proverka-koda-otveta-servera-i-http-zagolovkov) или несколько на сервер, например, можно передать заголовок If-Modified-Since чтобы страница возвращалась только если она была изменена:

curl -I --header 'If-Modified-Since: Mon, 26 Dec 2016 18:13:12 GMT' https://losst.ru

### Аутентификация curl

Если на сервере требуется аутентификация одного из распространенных типов, например, HTTP Basic или FTP, то curl очень просто может справиться с такой задачей. Для указания данных аутентификации просто укажите их через двоеточие в опции -u:

curl -u ftpuser:ftppass -T - ftp://ftp.testserver.com/myfile\_1.txt

Точно так же будет выполняться аутентификация на серверах HTTP.

### Использование прокси

Если вам нужно использовать прокси сервер для загрузки файлов, то это тоже очень просто. Достаточно задать адрес прокси сервера в опции -x:

curl -x proxysever.test.com:3128 <http://google.co.in>

### Еще примеры

|  |  |
| --- | --- |
| Команда | Описание |
| curl http://proft.me | получаем содержания главной страницы |
| curl -o index.html http://proft.me | получаем содержания главной страницы в файл index.html |
| curl -L http://example.com | при получении содержимого страницы следовать по редиректам (если такие есть) |
| curl -u username:password http://example.com/login/ | получение страницы скрытой за Basic HTTP Authentication |
| curl -x proxy.com:3128 http://proft.me | получение страницы используя прокси |
| curl -I proft.me | получаем http-заголовки с сайта |
| curl -H 'Host: google.ru' http://proft.me | подменить домен при обращении к серверу (передача своего заголовка) |
| curl --request POST "http://example.com/form/" --data "field1=value1&field2=value2" | передача данных POST-запросом |
| curl -X POST "http://example.com/form/" --data "field1=value1&field2=value2" | передача данных POST-запросом |
| curl -X POST -H "Content-Type: application/json" -d '"title":"Commando","year":"1985"' http://example.com/api/movies/ | передача данных POST-запросом, данные в виде JSON |
| curl --request PUT "http://example.com/api/movie/1/" --data "title=DjangoUnchained" | передача данных PUT-запросом |
| curl -F uploadfiled=@file.zip -F submit=OK http://example.com/upload/ | загрузка файла file.zip в форму (multipart/form-data) |
| curl -u username:password -O ftp://example.com/file.zip | скачать файл с FTP |
| curl -u username:password -T file.zip ftp://example.com/ | закачать файл по FTP |
| curl --cookie "login=proft" http://example.com/login/ | установить кукис |
| curl --cookie-jar cookies.txt http://example.com | сохранение кукисов в файл |
| curl --cookie cookies.txt http://example.com/login/ | использование сохраненных кукисов |

## Дополнительно

Команда curl предназначена больше для анализа и имитации различных действий на сервере, тогда как wget больше подходит для загрузки файлов и краулинга сайтов.

Таблица сравнения возможностей wget, curl и других утилит со схожим набором функций и целевым назначением: <https://curl.haxx.se/docs/comparison-table.html>