

Утилита загрузки web-ресурсов – wget. Как скачать сайт целиком?

Category: GNU/Linux, wget, утилиты
2025-06-01

Утилита wget.

Wget – это утилита командной строки для загрузки файлов из интернета в операционных системах GNU/Linux и других Unix-подобных системах. Она поддерживает загрузку по протоколам HTTP, HTTPS и FTP, а также может работать через прокси-серверы.

Официальная документация: [кликни сюда](#).

Основные возможности wget:

- **Рекурсивная загрузка:** wget может рекурсивно загружать целые сайты, включая все связанные файлы (например, изображения, CSS, JavaScript).
- **Продолжение загрузки:** Если загрузка прерывается, wget может продолжить её с того места, где она остановилась.
- **Автоматизация:** wget может быть использован в скриптах для автоматизации загрузки файлов.
- **Ограничение скорости:** Можно установить ограничение на скорость загрузки, чтобы не перегружать сеть.
- **Загрузка в фоновом режиме:** wget может работать в фоновом режиме, не занимая терминал.
- **Поддержка зеркалирования:** wget может создавать зеркала сайтов, сохраняя структуру каталогов и файлов.
- **Поддержка аутентификации:** wget поддерживает базовую аутентификацию и работу с куками.

Установка утилиты.

Установка для:

- Fedora, CentOS, RHEL или другом дистрибутиве на основе RPM.

```
$ sudo dnf -y install wget
```

- Ubuntu, Debian, Linux Mint или другом дистрибутиве на основе Debian.

```
$ sudo apt -y install wget
```

Файл конфигурации.

Глобальный (системный) файл `wgetrc` расположен по адресу: `/etc/wgetrc`

```
$ sudo mc /etc/wgetrc
```

Это системный файл настроек, который применяется ко всем пользователям на хосте.

Обычно требует прав администратора для редактирования.

Приоритет загрузки настроек.

Wget загружает настройки в следующем порядке:

1. Глобальный файл (`/etc/wgetrc`);
2. Пользовательский файл (`~/.wgetrc`);
3. Файл в текущем каталоге (`./wgetrc`);
4. Команды, переданные через `-e` (имеют наивысший приоритет).

Пользовательский файл конфигурации.

Если файл `.wgetrc` отсутствует, его можно создать вручную, заполнить и использовать, где требуется:

```
$ touch ~/.wgetrc
```

Если проверить версию `wget`, то он покажет версию `wget` и список поддерживаемых функций, но не сами настройки.

```
$ wget --version
```

Ответ (фрагмент):

```
Wgetrc:
  /home/user/.wgetrc (user)
  /etc/wgetrc (system)
Locale:
  /usr/share/locale
```

Запустите `wget` с ключом `-d` (`--debug`), чтобы увидеть подробный вывод, включая загруженные настройки:

```
$ wget -d https://example.com
```

В выводе будут указаны настройки, загруженные из файла конфигурации `.wgetrc`.

Если вы хотите проверить конкретную настройку, используйте ключ `-e` для её временного изменения и сравните поведение `wget`.

Например:

```
$ wget -e "limit_rate=100k" https://example.com
```

Если вы хотите использовать ключ `--config`, укажите путь к файлу конфигурации.

Этот файл будет использоваться вместо стандартных `.wgetrc`.

Например:

```
$ wget --config=/path/to/custom.wgetrc https://example.com
```

Синтаксис.

Команда `wget` 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.

Кончено же это не все ключи `wget`, которые я представлю ниже.

Официальная документация: [кликни сюда](#).

Базовые ключи запуска.

-V, --version – Отображает текущую версию `wget`. Полезно для проверки установленной версии программы.

-h, --help – Выводит справочную информацию с описанием всех доступных ключей командной строки `wget`. Используйте этот ключ, чтобы быстро ознакомиться с возможностями программы.

-d, --debug – Включает вывод отладочной информации, которая может быть полезна разработчикам при диагностике проблем. Обратите

внимание, что сборка wget может быть выполнена без поддержки отладки, и в этом случае ключ `-d` не сработает. Сборка с поддержкой отладки безопасна и не выводит отладочную информацию без явного запроса.

-b, --background – Переводит wget в фоновый режим сразу после запуска. Если выходной файл не указан с помощью ключа `-o`, все сообщения будут перенаправлены в файл `wget-log`.

-e command, --execute command – Позволяет выполнить команду, как если бы она была частью файла `.wgetrc`. Команда, заданная через этот ключ, выполняется после команд в `.wgetrc` и имеет приоритет над ними. Для выполнения нескольких команд используйте несколько ключей `-e`.

Например:

```
$ wget -e "cookies=off" -e "limit_rate=50k"  
https://example.com/largefile.zip
```

-4, --inet4only – Использует только протокол IPv4 для всех соединений. Полезно, если требуется ограничить wget работой только с IPv4-адресами.

-6, --inet6only – Использует только протокол IPv6 для всех соединений. Применяется, если необходимо ограничить wget работой только с IPv6-адресами.

-r, --recursive – Включает рекурсивное скачивание, позволяя wget просматривать каталоги и подкаталоги на удалённом сервере. Полезно для загрузки всего содержимого сайта или определённого каталога. Wget будет скачивать не только указанную страницу, но и все связанные с ней ресурсы (например, страницы, изображения, CSS, JavaScript).

-l <depth>, --level=<depth> – Устанавливает максимальную глубину рекурсии при скачивании. Например, `-l 2` ограничивает загрузку двумя уровнями вложенности. По умолчанию глубина равна 5.

-k, --convert-links – Преобразует ссылки в загруженных HTML-файлах для локального использования. Это позволяет корректно отображать страницы на локальной машине. Преобразуются только ссылки на реально загруженные ресурсы, то есть только лишь в конце работы wget будет выполняться окончательное преобразование.

-m, --mirror – Создаёт зеркало сайта на локальной машине. Этот ключ включает рекурсивное скачивание, сохранение временных меток и преобразование ссылок. Полезен для полного копирования сайта.

-p, --page-requisites – Скачивает все ресурсы, необходимые для

отображения страницы, включая изображения, CSS, JavaScript и другие файлы. Полезен для загрузки страниц с сохранением их внешнего вида. По умолчанию такие файлы не загружаются. Параметры `-r` и `-l`, указанные вместе могут помочь, но так как `wget` не различает внешние и внутренние документы, то нет гарантии, что загрузится все требуемое.

-nr, --no-parent – Запрещает переход в родительские каталоги при рекурсивном скачивании. Это ограничивает загрузку файлов только в пределах указанного каталога и его подкаталогов. Полезно для избежания загрузки лишних данных.

-A, --accept – Указывает список файлов, которые следует загружать. Имена файлов разделяются запятыми. Например, `-A ".html,.css"` загружает только HTML и CSS файлы.

-R, --reject – Указывает список файлов, которые не следует загружать. Имена файлов разделяются запятыми. Например, `-R ".jpg,.png"` исключает загрузку изображений.

Протоколирование и ключи входного файла.

-o logfile, --output-file=logfile – Сохраняет все сообщения `wget` в указанный файл `logfile`. По умолчанию сообщения выводятся в стандартный поток ошибок (`stderr`).

-a logfile, --append-output=logfile – Дописывает сообщения в указанный файл `logfile`. Аналогично ключу `-o`, но файл не перезаписывается, а дополняется. Если файл не существует, он будет создан.

-q, --quiet – Отключает вывод сообщений `wget`. Используйте этот ключ, если хотите, чтобы программа работала без вывода информации в терминал.

-v, --verbose – Включает подробный вывод с максимальной информацией. Этот режим включён по умолчанию.

-nv, --non-verbose – Уменьшает уровень детализации вывода, но не отключает его полностью. Отображаются только основные сообщения и ошибки. Для полного отключения вывода используйте ключ `-q`.

-i file, --input-file=file – Прочитать URL из файла. Читать URL из входного файла `file`, в этом случае URL не обязательно указывать в командной строке. Если адреса URL указаны в командной строке и во входном файле, первыми будут запрошены адреса из командной строки. Файл не должен (но может) быть документом HTML – достаточно последовательного списка адресов URL. Однако, при указании `--force-html` входной файл будет считаться html. В этом случае могут

возникнуть проблемы с относительными ссылками, которые можно решить указанием `<base href="url">` внутри входного файла или `--base=url` в командной строке.

-F, --force-html – Указывает, что входной файл, заданный через `-i`, является HTML-документом. При чтении списка адресов из файла устанавливает формат файла как HTML. Это позволяет обрабатывать относительные ссылки в файле. Это позволяет организовать загрузку по относительным ссылкам в локальном HTML-файле при указании `<base href="url">` внутри входного файла или `--base=url` в командной строке.

-B URL, --base=URL – Используется совместно с ключом `-F` для добавления указанного URL к началу относительных ссылок во входном файле. Это полезно для корректной обработки относительных ссылок в HTML-документах.

Ключи скачивания.

--bind-address=ADDRESS – Привязывает TCP/IP соединения к указанному адресу (ADDRESS) на локальной машине. ADDRESS может быть именем хоста или IP-адресом. Полезно, если на машине настроено несколько IP-адресов.

-t number, --tries=number – Устанавливает количество попыток скачивания. Значение 0 или `inf` означает бесконечное число попыток. По умолчанию – 20 попыток, за исключением критических ошибок (например, “в соединении отказано” или “файл не найден” (404)), при которых попытки не возобновляются.

-O file, --output-document=file – Сохраняет скачанные данные в указанный файл file. Если файл уже существует, он будет перезаписан. Если указано `-`, данные выводятся в стандартный поток вывода (stdout). Ключ `-k` игнорируется при использовании этого параметра. Помните, что комбинация с `-k` нормально определена только для скачивания одного документа.

-nc, --no-clobber – Предотвращает перезапись существующих файлов. Если файл уже существует, `wget` откажется скачивать его снова. Это полезно для избежания дублирования файлов.

-c, --continue – Продолжает загрузку частично скачанного файла. Полезно для возобновления прерванных загрузок. `Wget` запрашивает сервер о продолжении загрузки с места, где она была прервана.

--progress=type – Выбирает тип индикатора хода загрузки.

Возможные значения:

- `bar` (по умолчанию) – отображает ASCII-полосу прогресса.

- `dot` – отображает прогресс в виде точек, где каждая точка представляет фиксированный объём данных.
- `dot:style` – позволяет настроить стиль отображения точек (например, `binary`, `mega`).

-N, --timestamping – Включает проверку временных меток файлов. **Wget** скачивает файл только в том случае, если он был изменён на сервере.

-S, --server-response – выводит заголовки HTTP-ответов сервера и ответы FTP-серверов. Полезно для отладки и анализа.

--spider – Режим “паука”. **Wget** проверяет доступность URL, но не скачивает файлы. Полезно для проверки работоспособности ссылок на сайте.

-T seconds, --timeout=seconds – Устанавливает общее время ожидания для сетевых операций (в секундах). Включает время ожидания DNS, соединения и чтения. Значение 0 отключает тайм-ауты, то есть установка времени ожидания в 0 отменяет проверки. Эквивалентно одновременному указанию `--dns-timeout`, `--connect-timeout` и `--read-timeout`. / Когда **Wget** соединяется или читает с удалённого хоста, он проверяет время ожидания и прерывает операцию при его истечении. Это предотвращает возникновение аномалий, таких как повисшее чтение или бесконечные попытки соединения. Единственное время ожидания, установленное по умолчанию, – это время ожидания чтения в 900 секунд. / Если вы не знаете точно, что вы делаете, лучше не устанавливать никаких значений для ключей времени ожидания.

--dns-timeout=seconds – Устанавливает время ожидания для DNS-запросов. Незавершённые запросы отменяются по истечении указанного времени.

--connect-timeout=seconds – Устанавливает время ожидания для установки TCP-соединения. Соединения, требующие больше времени, отменяются.

--read-timeout=seconds – Устанавливает время ожидания для чтения данных. По умолчанию – **900 секунд**.

--limit-rate=amount – Ограничивает скорость скачивания. Значение может быть указано в байтах, килобайтах (k) или мегабайтах (m). Например, `--limit-rate=20k` ограничивает скорость до 20 КБ/с.

-w seconds, --wait=seconds – Устанавливает паузу между закачками (в секундах). Полезно для снижения нагрузки на сервер. Время можно указать в минутах (m), часах (h) или днях (d).

--waitretry=seconds – Устанавливает паузу между попытками скачивания для прерванных закачек. Пауза увеличивается линейно с каждой

попыткой.

--random-wait – Добавляет случайную паузу между загрузками. Полезно для избежания блокировки автоматических скачиваний.

-Y on/off, --proxy=on/off – Включает или отключает использование прокси. По умолчанию включено, если настроены переменные окружения.

-Q quota, --quota=quota – Устанавливает квоту на объём скачиваемых данных. Значение может быть указано в байтах, килобайтах (**k**) или мегабайтах (**m**). Квота не применяется к отдельным файлам.

--dns-cache=off – Отключает кэширование DNS-запросов. Полезно при работе с серверами, использующими динамические IP-адреса.

--restrict-file-names=mode – Ограничивает набор символов, используемых в именах файлов.

Возможные значения:

- **unix** – экранирует символы, запрещённые в Unix.
- **windows** – экранирует символы, запрещённые в Windows.
- **nocontrol** – отключает экранирование управляющих символов.

Пример:

```
--restrict-file-names=windows,nocontrol.
```

Ключи управления каталогами.

-nd, --no-directories – Не создаёт структуру каталогов при рекурсивном скачивании. Все файлы сохраняются в текущий каталог. Если имена файлов повторяются, к ним добавляются суффиксы **.n** (например, **file.1, file.2**).

-x, --force-directories – Создаёт структуру каталогов, даже если она не была бы создана по умолчанию. Например, команда **wget -x http://htmlweb.ru/robots.txt** сохранит файл в каталог **htmlweb.ru/robots.txt**.

-nH, --no-host-directories – Отключает создание каталога с именем хоста. По умолчанию, при рекурсивном скачивании (например, **wget -r http://htmlweb.ru/**) создаётся каталог **htmlweb.ru/**. Этот ключ отменяет такое поведение.

--protocol-directories – Использует название протокола (например, **http, ftp**) как часть структуры каталогов. Например, команда **wget -r http://host/** сохранит файлы в каталог **http/host/...** вместо **host/...**.

--cut-dirs=number – Игнорирует указанное количество уровней

вложенности каталогов при сохранении файлов. Полезно для управления структурой каталогов при рекурсивном скачивании.

Например:

Без ключей: ftp.htmlweb.ru/pub/xxx/

C -nH – это pub/xxx/;

C -nH --cut-dirs=1 – это xxx/;

C -nH --cut-dirs=2: – это . (текущий каталог).

Этот ключ сохраняет подкаталоги, в отличие от -nd.

-P prefix, --directory-prefix=prefix – Устанавливает корневой каталог для сохранения файлов. Все файлы и подкаталоги будут сохранены в указанный каталог prefix. По умолчанию используется текущий каталог (.).

Ключи для работы с HTTP.

-E, --html-extension – Добавляет расширение .html к имени локального файла, если скачиваемый URL имеет тип application/xhtml+xml или text/html, а его имя не заканчивается на .html или .htm. Полезно для зеркалирования сайтов, использующих динамические страницы (например, .asp). Например, URL http://site.com/article.cgi?25 будет сохранён как article.cgi?25.html. Для предотвращения повторной загрузки используйте ключи -k и -K.

--http-user=user, --http-passwd=password – Указывает имя пользователя и пароль для доступа к HTTP-серверу. Wget использует базовую или дайджест-аутентификацию в зависимости от сервера. Для безопасности рекомендуется хранить пароли в файлах .wgetrc или .netrc с ограниченными правами доступа.

--no-cache – Отключает кеширование на стороне сервера. Wget отправляет заголовок Pragma: no-cache, чтобы получить обновлённую версию файла, а не кешированную. Полезно для обхода устаревших данных на прокси-серверах.

--no-cookies – Отключает использование cookies. По умолчанию wget использует cookies, но не сохраняет их. Этот ключ полезен для повышения конфиденциальности, так как cookies могут использоваться для отслеживания пользователей.

--load-cookies file – Загружает cookies из указанного файла перед выполнением HTTP-запросов. Файл должен быть в формате cookies.txt, используемом Netscape. Полезно для зеркалирования сайтов, требующих авторизации.

--save-cookies file – Сохраняет cookies в указанный файл перед завершением работы wget. По умолчанию не сохраняются сессионные cookies и cookies с истёкшим сроком действия.

--keep-session-cookies – Сохраняет сессионные cookies при использовании --save-cookies. Обычно сессионные cookies не сохраняются, так как они предназначены для временного использования.

--ignore-length – Игнорирует заголовок Content-Length, если сервер отправляет некорректное значение. Полезно, если wget повторно скачивает один и тот же файл из-за ошибок в заголовке.

--header=additional-header – Добавляет пользовательский заголовок к HTTP-запросу.

Например:

```
$ wget --header='Accept-Charset: iso-8859-2' --header='Accept-Language: hr' http://example.com/
```

Пустая строка в качестве заголовка очищает все ранее добавленные заголовки.

--proxy-user=user, --proxy-passwd=password – Указывает имя пользователя и пароль для аутентификации на прокси-сервере. Используется базовая схема аутентификации.

--referer=url – Добавляет заголовок Referer: url к HTTP-запросу. Полезно, если сервер проверяет источник запроса.

--save-headers – Сохраняет заголовки HTTP-ответа в файл перед содержимым. Заголовки отделяются от содержимого пустой строкой.

-U agent-string, --user-agent=agent-string – Изменяет значение заголовка User-Agent на указанное. По умолчанию wget использует wget/версия. Полезно для обхода ограничений на определённые браузеры.

--post-data=string, --post-file=file – Использует метод POST для отправки данных на сервер. --post-dat отправляет строку, а --post-file отправляет содержимое файла.

Пример:

```
$ wget --post-data='user=foo&password=bar' http://example.com/login
```

Если сервер перенаправляет запрос, данные POST не отправляются на новый URL.

Примеры использования.

Проверить работоспособность ссылок.

Не скачивать страницы, а лишь проверить их наличие.

Полезно для проверки работоспособности ссылок на сайте.

Например, с помощью `wget` можно проверить закладки, то есть ссылки из файла:

```
$ wget --spider --force-html -i bookmarks.html
```

или факт работы ссылок на всём сайте:

```
$ wget --spider --recursive --no-check-certificate  
https://example.org
```

Загрузка страницы.

Скачать страницу с глубиной следования 10, записывая протокол в файл `log`:

```
$ wget -r -l 10 https://example.org/ -o log
```

Скачать содержимое каталога.

Скачать содержимое каталога `http://example.org/~luzer/my-archive/` и всех его подкаталогов, при этом не поднимаясь по иерархии каталогов выше:

```
$ wget -r --no-parent http://example.org/~luzer/my-archive/
```

Загрузка файла.

Команда `wget` скачает один файл и сохранит его в текущем каталоге.

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

Загрузка файла с логином и паролем.

Использование имени пользователя и пароля на FTP/HTTP (вариант 1):

```
$ wget ftp://login:password@ftp.example.org/some_file.iso
```

Использование имени пользователя и пароля на FTP/HTTP (вариант 2):

```
$ wget --user=login --password=password  
ftp://ftp.example.org/some_file.iso
```

Сохранить файл с другим именем.

Опция `-O` 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
```

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
```

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
```

Продолжить загрузку.

Утилита `wget` 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
```

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
```

Ограничение скорости загрузки.

Утилита `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
```

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=vasia --http-password=password
http://mirrors.hns.net.in/centos/6.3/isos/x86_64/CentOS-6.3-x86_64-L
iveDVD.iso
```

или:

```
$ wget --ftp-user=vasia --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
```

Залогиниться и скачать файл ключа.

```
$ sudo mcedit download_key.sh
```

Помещаем в него содержимое:

```
#!/bin/bash
# Залогиниться и скачать файл ключа

# Сохраняем куки и выполняем вход
wget --save-cookies cookies.txt --post-data
"login=ТВОЙ_ЛОГИН&password=ТВОЙ_ПАРОЛЬ" http://beta.drweb.com/files/
-O /dev/null

# Загружаем файл ключа, используя сохранённые куки
wget --load-cookies cookies.txt
```

```
"http://beta.drweb.com/files/?p=win%2Fdrweb32-betatesting.key&t=f" -  
0 drweb32-betatesting.key
```

```
# Удаляем временный файл с куками (опционально)  
rm cookies.txt
```

Сделаем его исполняемым:

```
$ sudo chmod +x download_key.sh  
$ sudo ./download_key.sh
```

Загрузить и выполнить.

Wget позволяет сразу же выполнять скачанные скрипты:

```
$ wget -O - http://www.example.ru/скрипт.sh | bash
```

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

Сохранить файл в каталоге.

По умолчанию `wget` сохраняет файл в текущем каталоге, но это очень легко изменить с помощью опции `-P`:

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

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" //example.ru
```

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 так далее.

Скачать сайт целиком.

Wget позволяет не только скачивать одиночные файлы, но и целые сайты, чтобы вы могли их потом просматривать в **off-line** режиме локально.

Внимание! Некоторые сайты имеют механизмы защиты от больших нагрузок. При слишком наглой загрузке сайта, ваши запросы могут быть заблокированы системой безопасности сайта. Сайт станет недоступным. Поэтому требуется действовать очень аккуратно и не провоцировать систему безопасности на ограничительные меры.

Примечание! Имеются вариации работы программы в зависимости от версий программного обеспечения. В случае возникновения предостережений от программы, просто выполните её требования из подсказки. Удалите лишние или конфликтующие ключи.

Рекомендации:

- Если вы не доверяете сертификату сайта, используйте **--no-check-certificate**, но помните, что это может быть небезопасно.
- Убедитесь, что у вас есть права на запись в каталог `/home/user/web-sites`.
- Если вы хотите скачать только определённые типы файлов, добавьте ключи **--accept** или **--reject**.

Подготовим каталог хранения копии ресурса в домашнем каталоге пользователя:

```
$ mkdir ~/web-sites
```

Использование wget, чтобы скачать сайт в GNU/Linux выглядит вот так:

```
$ wget --mirror --no-parent --recursive --timestamping --random-wait
```

```
--convert-links --page-requisites --html-extension --restrict-file-  
names=windows --domains hamsterden.ru --limit-rate=100k --wait=10 -P  
/home/hamster/web-sites https://hamsterden.ru
```

```
$ wget --mirror \  
--no-parent \  
--recursive \  
--timestamping \  
--random-wait \  
--convert-links \  
--page-requisites \  
--html-extension \  
--restrict-file-names=windows \  
--domains site.ru \  
--limit-rate=100k \  
--wait=10 \  
-P /home/user/web-sites \  
https://site.ru
```

Что делает эта команда:

1. **--mirror** – Создает зеркало сайта, включая рекурсивное скачивание и сохранение структуры каталогов.
2. **--no-parent** – Запрещает переход в родительские каталоги при рекурсивном скачивании.
3. **--recursive** – Включает рекурсивное скачивание всех связанных страниц и файлов.
4. **--timestamping** – Проверяет временные метки файлов и скачивает только обновлённые версии.
5. **--random-wait** – Добавляет случайные паузы между запросами для снижения нагрузки на сервер.
6. **--convert-links** – Преобразует ссылки в загруженных файлах для локального использования.
7. **--page-requisites** – Скачивает все ресурсы, необходимые для отображения страниц (изображения, CSS, JavaScript и так далее).
8. **--html-extension** – Добавляет расширение .html к файлам, которые имеют тип text/html или application/xhtml+xml.
9. **--restrict-file-names=windows** – Ограничивает имена файлов символами, допустимыми в Windows.
10. **--domains site.ru** – Ограничивает скачивание файлов только в пределах домена www.example.org.
11. **--limit-rate=100k** – Ограничивает скорость скачивания до 100 КБ/с.
12. **--wait=10** – Устанавливает паузу в 10 секунд между запросами.
13. **-P /home/user/web-sites** – Сохраняет все файлы в каталог /home/user/web-sites.
14. **https://site.ru** – URL сайта, который нужно скачать.

Если загрузка прервалась или вы её отменили, то догрузить сайт можно вот так:

```
$ wget --mirror --no-parent --recursive --timestamping --random-wait
--convert-links --page-requisites --html-extension --restrict-file-
names=windows --domains hamsterden.ru --limit-rate=100k --wait=10 -P
/home/hamster/web-sites https://hamsterden.ru --no-clobber
```

Добавили ключ:

1. **--no-clobber** – Не перезаписывает существующие файлы.

Скачать страничку целиком.

Для скачивания одиночной страницы все параметры примерно такие же:

```
$ wget --no-parent --recursive --timestamping --convert-links --
page-requisites --html-extension --restrict-file-names=windows --
domains timeweb.cloud --limit-rate=2000k -P /home/hamster/web-sites
https://timeweb.cloud/tutorials/linux/komanda-chmod-v-linux-cto-del
aet-i-kak-ispolzovat
```

```
$ wget --no-parent \
--recursive \
--timestamping \
--convert-links \
--page-requisites \
--html-extension \
--restrict-file-names=windows \
--domains timeweb.cloud \
--limit-rate=2000k \
-P /home/hamster/web-sites \
https://timeweb.cloud/tutorials/linux/komanda-chmod-v-linux-cto-del
aet-i-kak-ispolzovat
```

Если загрузка прервалась или вы её отменили, то догрузить сайт можно вот так:

```
$ wget --no-parent --recursive --timestamping --convert-links --
page-requisites --html-extension --restrict-file-names=windows --
domains timeweb.cloud --limit-rate=2000k -P /home/hamster/web-sites
https://timeweb.cloud/tutorials/linux/komanda-chmod-v-linux-cto-del
aet-i-kak-ispolzovat --no-clobber
```

Оригиналы источников информации.

1. losst.pro «Команда wget Linux – Losst».
2. goubuntu.ru «Как скачать сайт целиком в Ubuntu»
3. linuxhandbook.com «Ignore SSL Certificate Error with Wget».
4. habr.com «Изучаем команду wget на 12 примерах».
5. htmlweb.ru «WGet – программа для загрузки файлов и скачивания сайта целиком».
6. help.ubuntu.ru «wget»