Потребность передачи файлов между серверами и компьютерами возникает довольно часто, особенно при администрировании нескольких устройств. Обычно для этих целей удобно использовать ssh и scp, но если файл очень большой, а изменена была только небольшая его часть, или вы хотите настроить постоянную автоматическую синхронизацию, то scp уже явно неидеальный вариант. Для этого существуют специальные утилиты. В этой статье мы рассмотрим одну из них. А именно будет рассмотрена rsync синхронизация в linux.

**Rsync** - это программное обеспечение с открытым исходным кодом, которое можно использовать для синхронизации файлов и папок с локального компьютера на удаленный и наоборот. Примечательная особенность Rsync - возможность передавать зашифрованные файлы с помощью SSH и SSL. Кроме того, здесь передача файлов выполняется в один поток, в отличие от других подобных программ, создающий отдельный поток для передачи каждого файла. Это увеличивает скорость и убирает дополнительные задержки, которые становятся проблемой при передаче большого количества маленьких файлов.

Возможно использование rsync для синхронизации файлов, каталогов, при этом может применяться сжатие и шифрование. Программа впервые была применена в июне 1996 года, она разработана Эндрю Тридгелом и Полом Маккерасом. Rsync синхронизация выполняется по протоколу RSYNC, который специально разработан не просто для передачи файлов между двумя компьютерами, а для их синхронизации. Если точнее, то передается не файл полностью, а только то, что было изменено.

Установка:

***sudo apt-get install rsync***

А теперь, уже по традиции подобных статей, рассмотрим синтаксис команды rsync:

**$ rsync опции источник приемник**

**ОПЦИИ RSYNC**

Теперь давайте кратко рассмотрим параметры rsync. Здесь перечислены не все опции. Для более подробной информации смотрите man rsync:

* **-v** - Выводить подробную информацию о процессе копирования;
* **-q** - Минимум информации;
* **-c** - Проверка контрольных сумм для файлов;
* **-a** - Режим архивирования, когда сохраняются все атрибуты оригинальных файлов;
* **-R** - Относительные пути;
* **-b** - Создание резервной копии;
* **-u** - Не перезаписывать более новые файлы;
* **-l** - Копировать символьные ссылки;
* **-L** - Копировать содержимое ссылок;
* **-H** - Копировать жесткие ссылки;
* **-p** - Сохранять права для файлов;
* **-g** - Сохранять группу;
* **-t** - Сохранять время модификации;
* **-x** - Работать только в этой файловой системе;
* **-e** - Использовать другой транспорт, например, ssh;
* **-z** - Сжимать файлы перед передачей;
* **--delete** - Удалять файлы которых нет в источнике;
* **--exclude** - Исключить файлы по шаблону;
* **--recursive** - Перебирать директории рекурсивно;
* **--no-recursive** - Отключить рекурсию;
* **--progress** - Выводить прогресс передачи файла;
* **--stat** - Показать статистику передачи;
* **--version** - Версия утилиты.

Для синхронизации на локальном компьютере:

Sudo rsync -abhz –progress /media/mint/878 /rsync – что я использовал

**DD :**

Довольно часто системным администраторам приходится копировать различные двоичные данные. Например, иногда может понадобиться сделать резервную копию жесткого диска, создать пустой файл, заполненный нулями для организации пространства подкачки или другой виртуальной файловой системы.

Для решения всех этих задач используется утилита dd linux, которая просто выполняет копирование данных из одного места в другое на двоичном уровне. Она может скопировать CD/DVD диск, раздел на диске или даже целый жесткий диск. В этой статье мы рассмотрим что из себя представляет команда dd linux, основные ее опции и параметры, а также как ею пользоваться.

## КАК РАБОТАЕТ КОМАНДА DD?

Сначала нужно понять как работает команда dd и что она делает. Фактически, это аналог утилиты [копирования файлов cp](https://losst.ru/kopirovanie-fajlov-v-linux)только для блочных данных. Утилита просто переносит по одному блоку данных указанного размера с одного места в другое. Поскольку в Linux все, в том числе, устройства, считается файлами, вы можете переносить устройства в файлы и наоборот.

С помощью различных опций утилиты можно повлиять на размер блока, а это, в свою очередь, уже влияет на скорость работы программы. Дальше мы рассмотрим основные опции утилиты и ее возможности.

## КОМАНДА DD

Синтаксис утилиты достаточно необычен, но в то же время очень прост, после того как вы его запомните и привыкнете:

**$ dd if=источник\_копирования of=место\_назначения параметры**

С помощью параметра if вам нужно указать источник, откуда будут копироваться блоки, это может быть устройство, например, /dev/sda или файл - disk.img. Дальше, с помощью параметра of необходимо задать устройство или файл назначения. Другие параметры имеют такой же синтаксис, как if и of.

Теперь давайте рассмотрим дополнительные параметры:

* **bs** - указывает сколько байт читать и записывать за один раз;
* **cbs** - сколько байт нужно записывать за один раз;
* **count** - скопировать указанное количество блоков, размер одного блока указывается в параметре bs;
* **conv** - применить фильтры к потоку данных;
* **ibs** - читать указанное количество байт за раз;
* **obs** - записывать указанное количество байт за раз;
* **seek** - пропустить указанное количество байт в начале устройства для чтения;
* **skip** - пропустить указанное количество байт в начале устройства вывода;
* **status** - указывает насколько подробным нужно сделать вывод;
* **iflag, oflag** - позволяет задать дополнительные флаги работы для устройства ввода и вывода, основные из них: nocache, nofollow.

**Команда** **df**, означает Disk Free, что означает использование дискового пространства в файловой системе. Она отображает объем дискового пространства, доступного в файловой системе, в системе **Linux**. **Команду** **DF** не следует путать с **командой** **df**. Они служит различным целям.

df -hT !

<https://ru.wikihow.com/создать-ISO-файл-в-Linux>

вот, что у меня получилось:

Изображение выглядит как текст, снимок экрана, монитор, компьютер

Автоматически созданное описание

# КАК ПОЛЬЗОВАТЬСЯ APT

[Инструкции](https://losst.ru/instruktsii) Обновлено:  17 июня, 2020  [5](https://losst.ru/kak-polzovatsya-apt#comments)  [admin](https://losst.ru/author/admin)

Пакетный менеджер Apt или Advanced Package Tool используется во множестве дистрибутивов, основанных на Debian или Ubuntu, а таких дистрибутивов сейчас очень много. Этот пакетный менеджер поддерживает все необходимые функции, вы можете устанавливать и удалять пакеты, обновлять то, что было уже установлено, искать пакеты, устанавливать их из файла или загружать без установки. При этом все зависимости будут разрешаться автоматически.

Утилита apt ничем не уступает пакетному менеджеру yum, используемому в RPM дистрибутивах, да, здесь все ещё не поддерживается частичное обновление пакетов, как в Zypper, но в целом всё выглядит очень неплохо. В этой статье мы подробно рассмотрим как пользоваться apt в Linux для решения задач работы с программами.

Содержание статьи:

* [Синтаксис и опции apt](https://losst.ru/kak-polzovatsya-apt#%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_%D0%B8_%D0%BE%D0%BF%D1%86%D0%B8%D0%B8_apt)
* [Как пользоваться apt](https://losst.ru/kak-polzovatsya-apt#%D0%9A%D0%B0%D0%BA_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D1%8C%D1%81%D1%8F_apt)
  + [1. Обновление пакетов](https://losst.ru/kak-polzovatsya-apt#1_%D0%9E%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%BE%D0%B2)
  + [2. Установка пакетов](https://losst.ru/kak-polzovatsya-apt#2_%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%BE%D0%B2)
  + [3. Удаление пакетов](https://losst.ru/kak-polzovatsya-apt#3_%D0%A3%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%BE%D0%B2)
* [Выводы](https://losst.ru/kak-polzovatsya-apt#%D0%92%D1%8B%D0%B2%D0%BE%D0%B4%D1%8B)

## СИНТАКСИС И ОПЦИИ APT

Синтаксис команды apt очень простой и похож на другие команды Linux:

**$ apt опции команда параметры\_команды**

Опции указывают общее поведение утилиты, команда - действие, которое надо выполнить, а в параметрах команды обычно передается имя пакета, с которым следует работать, например, установить или удалить. Вот основные опции утилиты:

* **-v, --version** - выводит версию утилиты;
* **-h, --help** - выводит справку по использованию утилиты;
* **-y, --yes** - автоматически отвечать "да" на все возникающие вопросы;
* **--assume-no** - автоматически отвечать "нет" на все возникающие вопросы;
* **-d, --download-only** - только скачать пакеты и больше ничего не делать;
* **-f, --fix-broken** - исправить недостающие зависимости;
* **--no-download** - ничего не загружать, использовать только пакеты из кэша;
* **-s, --simulate** - режим симуляции, никакие операции не выполняются, только выводится информация на экран;
* **--allow-unauthenticated** - позволяет установить пакеты, из репозиториев, для которых нет GPG подписи;
* **--no-install-recommends** - не устанавливать рекомендованные пакеты, по умолчанию будут установлены;
* **-m, --ignore-missing** - игнорировать пакеты, которые существует;
* **-q, --quiet** - выводить минимум информации, не показывать прогресс бар;
* **-V, --verbose-versions** - показывать полные версии обновленных пакетов;
* **--only-upgrade** - не устанавливать новые пакеты, только обновлять;
* **--allow-downgrades** - разрешить откатывать версию пакетов;
* **--reinstall** - переустановить пакет если он уже установлен;

А теперь давайте пройдёмся по командам apt, которые вы можете использовать:

* **install** - установить пакет;
* **remove** - удалить пакет, конфигурационные файлы, которые были изменены в вашей системе удалены не будут;
* **purge** - полностью удалить пакет, вместе со всеми его конфигурационными файлами;
* **autoremove** - очистить ненужные пакеты;
* **autoclean** - очистить кэш пакетов;
* **update** - обновить списки пакетов из репозиториев;
* **upgrade** - обновить версию пакета до последней, если пакет не указан будут обновлены все пакеты;
* **full-upgrade** - полное обновление системы, включая удаление несовместимых или больше ненужных пакетов;
* **list** - список установленных пакетов;
* **search** - поиск пакетов;
* **show** - посмотреть информацию о пакете;
* **download** - скачать пакет в текущую папку;
* **edit-sources** - открыть с настройками репозиториев в текстовом редакторе.
* **source** - скачать исходный код пакета в текущую папку;
* **build-dep** - установить зависимости необходимые для сборки выбранного пакета;

Дальше рассмотрим несколько примеров работы с apt, которые пригодятся как начинающим, так и более опытным пользователям.

## Команда aptitude - это ещё один пакетный менеджер для Ubuntu и Debian. Утилита позволяет делать всё то же самое, что и apt, но только кроме командной строки у неё есть псевдографический интерфейс на основе ncurses. С помощью этого инструмента вы можете устанавливать и удалять программы, обновлять систему, смотреть информацию о пакетах и их зависимости, выполнять поиск и многое другое.

## dd +Архивирование :

## gzip bzip2 xz – синтаксис одинаковый.

## gzip:

## Изображение выглядит как текст, снимок экрана, электроника Автоматически созданное описание



## 

## bzip2:

## Изображение выглядит как текст, электроника, снимок экрана, компьютер Автоматически созданное описание

## zstd

## Изображение выглядит как текст Автоматически созданное описаниеИзображение выглядит как текст Автоматически созданное описаниеИзображение выглядит как текст Автоматически созданное описание

## вот что было когда использовал dd без архиватора:

## 

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Еще вот такая ошибка была

Изображение выглядит как текст, электроника, снимок экрана, компьютер

Автоматически созданное описание