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.

Это были все основные опции, которые вам могут понадобиться. Теперь перейдем ближе к практике и рассмотрим несколько примеров как пользоваться утилитой dd linux.

**КАК ПОЛЬЗОВАТЬСЯ DD?**

Обычные пользователи используют команду dd чаще всего для создания образов дисков DVD или CD. Например, чтобы сохранить образ диска в файл можно использовать такую команду:

 sudo dd if=/dev/sr0 of=~/CD.iso bs=2048 conv=noerror

Фильтр noerror позволяет отключить реагирование на ошибки. Дальше, вы можете создать образ жесткого диска или раздела на нем и сохранить этот образ на диск. Только смотрите не сохраните на тот же жесткий диск или раздел, чтобы не вызвать рекурсию:

 dd if=/dev/sda of=~/disk.img

В вашей домашней папке будет создан файл с именем disk1.img, который в будущем можно будет развернуть и восстановить испорченную систему. Чтобы записать образ на жесткий диск или раздел достаточно поменять местами адреса устройств:

 dd if=~/disk.img of=/dev/sda

Очень важная и полезная опция - это bs. Она позволяет очень сильно влиять на скорость работы утилиты. Этот параметр позволяет установить размер одного блока при передаче данных. Здесь нужно задать цифровое значение с одним из таких модификаторов формата:

* **с** - один символ;
* **b** - 512 байт;
* **kB** - 1000 байт;
* **K** - 1024 байт;
* **MB** - 1000 килобайт;
* **M** - 1024 килобайт;
* **GB** - 1000 мегабайт;
* **G** - 1024 мегабайт.