# 'dd' command in Linux - GeeksforGeeks

**dd** is a command-line utility for Unix and Unix-like operating systems whose primary purpose is to convert and copy files.

- On Unix, device drivers for hardware (such as hard disk drives) and special device files (such as /dev/zero and /dev/random) appear in the file system just like normal files.
- dd can also read and/or write from/to these files, provided that function is implemented in their respective drivers
- As a result, dd can be used for tasks such as backing up the boot sector of a hard drive, and obtaining a fixed amount of random data.
- The dd program can also perform conversions on the data as it is copied, including byte order swapping and conversion to and from the ASCII and EBCDIC text encodings.

**Usage :** The command line syntax of dd differs from many other Unix programs, in that it uses the syntax *option=value* for its command line options, rather than the more-standard *-option value* or *- option=value* formats. By default, dd reads from stdin and writes to stdout, but these can be changed by using the if (input file) and of (output file) options.

### Some practical examples on dd command :

- 1. **To backup the entire harddisk :** To backup an entire copy of a hard disk to another hard disk connected to the same system, execute the dd command as shown. In this dd command example, the UNIX device name of the source hard disk is /dev/hda, and device name of the target hard disk is /dev/hdb.
  - # dd if = /dev/sda of = /dev/sdb
    - *"if"* represents inputfile, and *"of"* represents output file. So the exact copy of */dev/sda* will be available in */dev/sdb*.
    - If there are any errors, the above command will fail. If you give the parameter *"conv=noerror"* then it will continue to copy if there are read errors.
    - Input file and output file should be mentioned very carefully. Just in case, you mention source device in the target and vice versa, you might loss all your data.
    - To copy, hard drive to hard drive using dd command given below, sync option allows you to copy everything using synchronized I/O.

# dd if = /dev/sda of = /dev/sdb conv=noerror, sync

- 2. **To backup a Partition :** You can use the device name of a partition in the input file, and in the output either you can specify your target path or image file as shown in the dd command.
  - # dd if=/dev/hda1 of=~/partition.img
- 3. **To create an image of a Hard Disk :** Instead of taking a backup of the hard disk, you can create an image file of the hard disk and save it in other storage devices. There are many advantages of backing up your data to a disk image, one being the ease of use. This method is typically faster than other types of backups, enabling you to quickly restore data following an unexpected catastrophe.It creates the image of a hard disk /dev/hda.
  - # dd if = /dev/hda of = ~/hdadisk.img
- 4. **To restore using the Hard Disk Image :** To restore a hard disk with the image file of an another hard disk, the following dd command can be used

# dd if = hdadisk.img of = /dev/hdb

The image file hdadisk.img file, is the image of a /dev/hda, so the above command will restore the image of /dev/hda to /dev/hdb.

5. **To create CDROM Backup :** dd command allows you to create an iso file from a source file. So we can insert the CD and enter dd command to create an iso file of a CD content.

```
# dd if = /dev/cdrom of = tgsservice.iso bs = 2048
```

dd command reads one block of input and process it and writes it into an output file. You can specify the block size for input and output file. In the above dd command example, the parameter "*bs*" specifies the block size for the both the input and output file. So dd uses **2048bytes** as a block size in the above command.

#### **References :**

- dd command in Unix.
- <u>dd man page.</u>

The dd command stands for "**data duplicator**" and used for copying and converting data. It is very powerful low level utility of Linux which can do much more like;

• Backup and restore the entire hard disk or partition.

• Backup of MBR (Master Boot Record)

• It can copy and convert magnetic tape format, convert between ASCII and EBCDIC formats, **swap** bytes and can also convert lower case to upper case.

• It can also be used by Linux kernel make files to make boot images.

Only superuser can run this command because you can face a big data loss due to it's improper usage, so you should be very careful while working with this utility. At that moment data loss can convert the dd utility as a "**data destroyer**" for you. That's why it is recommended that beginners should not use this command on a production machine until they get familiarity on this. You must make sure that target location must have sufficient space while running this command.

# Syntax of dd command

Before we start with some practical work we need to talk about it's syntax.

#### dd if=<source file name> of=<target file name> [Options]

We normally do not explain about syntax but this command syntax require some explanation. The syntax is totally different when compared to many Linux commands we know. In this syntax dd is followed by two things

**if**=<**source**> –This is a source from where you want to copy data and 'if' stands for input-file.

**of**=<**destination**> –This is a source from where you want to write/paste data and 'of' stands for outputfile.

[options] – These options include, how fast data should be written, what format etc.

Input(source file name) and Output(target file name) in syntax are disks, partitions, files and devices to which you want to write and read data from. There are many options which we will discuss in examples.

# Learn Linux dd command with examples

**Example 1:** Clone one hard disk to another hard disk. This is useful when we are building many machines with same configuration. We no need to install OS on all the machines. Just <u>install OS</u> and required software on machine then clone with below example.

```
dd if=/dev/sda of=/dev/sdb
```

Example 2: We can take backup of a partition/complete HDD for future restoration.

Backing up a partition to a file(to my home directory as hdadisk.img)

```
dd if =<u>/dev</u>/sda2 of=~/hdadisk.img
```

Restoring this image file in to other machine dd if=hdadisk.img of=/dev/sdb3

**Example 3:** Do you feel hdadisk.img is bit big? <u>Use gzip</u> or bzip2 to compress when creating image.

```
dd if =/dev/sda2 | bzip2 hdadisk.img.bz2
```

**Example 4:** Do you know dd command can be used as file copier as well? Yes, if you don't have cp command use dd command to copy a file from one location to other.

```
dd if=/home/imran/abc.txt of=/mnt/abc.txt
```

Ok, that is fine for basic use of dd command. But the beauty of dd command lies in advanced usages like wiping of disks, complete wiping of disks, MBR backups etc.

# Advanced usage of Linux dd command

From here you should be careful when using this command and you should first try these commands on a test machine before getting familiarity.

Example 5: Wipe/delete content of a disk so that it will be empty for some one to use it.

```
dd if=/dev/zero of=/dev/sdb
```

This will wipe out your second hard disk and every bit is written with zero. you may be interested in **learning /dev/null and /dev/zero files** which do similar stuff but there is a bit of difference.

# How writing of a file on hard disk with data happen?

Normally what ever you write on to a disk at the block level it will write combination of zeros and ones. Hope you know this and what we are doing here is that we are just writing zeros which will clear all 1's from the hard disk. This eventually makes your disk empty.

**Example 6:** What to hide your ass by deleting your personal data. Many people think if we do rm -rf /<your data> will do the needful. But we can recover those deletion by using disk recovery tools like **Photorec** or some **forensic tools**. But if you want some not to recover your data you have to write random data on your partition where you data resides.

#### dd if=/dev/random of=/dev/sdb

Do above command multiple times so that it is real hard to recover data. If I am in your place, I will write below shell script to do that.

#### for i in {1..10};do dd if=/dev/random of=/dev/sdb;done

This will execute dd command 10 times in a row one after the other.

**Example 7:** We can create virtual file system with dd command which can be used as swap. To know more about you should know on **how to create virtual swap space in Linux**.

#### dd if=/dev/zero of=/swapfile bs=1024 count=200000

where **bs** stands for block size and count is nothing but number of such blocks used to crate this swap file.

Make sure you use **block sizes** in multiples of 1024 bytes which is equal to 1KB. Ff you do not specify block size, dd use a default block size of 512 bytes. Below conventions will work for block sizes.

N and BYTES may be followed by the following multiplicative suffixes: c =1, w =2, b =512,

```
kB =1000, K =1024,
Mb =1000*1000, MB=1024*1024,
```

Gb =1000\*1000\*1000, GB =1024\*1024\*1024,

```
and so on for T, P, E, Z, Y.
```

```
Example 8: We can even create ISO files from a CD-ROM or DVD-ROM using dd command.
dd if=/dev/dvd of=/opt/my_linux_image.iso
```

or with more

dd if=/dev/sr0 of=/home/\$user/mycd\_image.iso bs=2048 conv=sync

Some other examples:

```
dd if=/dev/sda1 of=/dev/sdb1 bs=4096 conv=noerror,sync
```

This will make clone of one partition sda1 to other sdb1 partition, also used sync option to synchronize the partition

#### dd if=/dev/sdx of=/dev/sdy bs=64k conv=noerror,sync

This will clone the entire drive, including MBR, all partitions and data where noerrr instructs dd to ignore all read errors while continuing operations. The snyc data offsets stay in sync And bs=sets block size which is set to 64k.

**Example 9:** We can even check disk quota using dd command by creating huge files which eats up HDD in no time.

#### dd if=/dev/zero of=/usr/disk-img/disk-quota.ext3 count=40960

This will create 20MB file (disk image) at said path.

#### Example 10: We can even create bootable USB's using dd command.

dd if=/home/\$user/bootimage.img of=/dev/sdc

This will create boot-able USB drive where /dev/sdc is an USB drive.

#### Example 11: Data recovery using ddrescue command.

ddrescue tool is used for cloning and recovering data. this is not installed by default in many Linux machines. Below are the commands to install for respective distributions.

## Installing ddrescue in Debian based machines

apt-get install gddrescue

## Installing ddrescue in Redhat based machines

#### yum install gddrescue

It can copy data from one file/block device(hard disk or CD-ROM) to another while trying to rescue the data i.e read error to maximizing the recovered data. To recover data ddrescue command needs to run in two steps followed by running fsck command.

Step 1: It copies each and every block without reading errors and logging these errors in log file.

**Step 2:** It will copy only bad block and try to read 3 times to source before it gives up. Than you can run command to check file system for corruption.

Step 3: Running fsck command.

```
ddrescue -f -n /dev/sdX /dev/sdY rescue.log
  ddrescue -d -f -r3 /dev/sdX /dev/sdY rescue.log
  fsck -f /dev/sdY
```

#### Example 12: Computer forensics using dcfldd

**dcfldd** is the enhanced version of dd having useful features used for computer forensics and security as well. It takes dd's parameters. You can use dcfldd command when you need to know that a copy and subsequent copies are identical to the original. dcfldd has some additional features like;

- Hashing on-the-fly
- Status output
- Image/wipe verify
- Split output
- hashing

Again We have to install this command as well, if that is not present.