# How to use dd in Linux without destroying your disk

# Safely and reliably make perfect copies of drives, partitions, and filesystems with the Linux dd tool.

Whether you're trying to rescue data from a dying storage drive, backing up archives to remote storage, or making a perfect copy of an active partition somewhere else, you'll need to know how to safely and reliably copy drives and filesystems. Fortunately, dd is a simple and powerful image-copying tool that's been around, well, pretty much forever. And in all that time, nothing's come along that does the job better.

#### Basic dd operations

Now that you've been suitably warned, we'll start with something straightforward. Suppose you want to create an exact image of an entire disk of data that's been designated as /dev/ sda. You've plugged in an empty drive (ideally having the same capacity as your /dev/sda system).

The syntax is simple: if= defines the source drive and of= defines the file or location where you want your data saved:

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

The next example will create an .img archive of the /dev/sda drive and save it to the home directory of your user account:

#### dd if=/dev/sda of=/home/username/sdadisk.img

Those commands created images of entire drives. You could also focus on a single partition from a drive. The next example does that and also uses bs to set the number of bytes to copy at a single time (4,096, in this case). Playing with the bs value can have an impact on the overall speed of a dd operation, although the ideal setting will depend on your hardware profile and other considerations.

#### dd if=/dev/sda2 of=/home/username/partition2.img bs=4096

Restoring is simple: Effectively, you reverse the values of if and of. In this case, if= takes the image you want to restore, and of= takes the target drive to which you want to write the image:

#### dd if=sdadisk.img of=/dev/sdb

You can also perform both the create and copy operations in one command. This example, for instance, will create a compressed image of a remote drive using SSH and save the resulting archive to your local machine:

#### ssh username@54.98.132.10 "dd if=/dev/sda | gzip -1 -" | dd of=backup.gz

You should always test your archives to confirm they're working. If it's a boot drive you've created, stick it into a computer and see if it launches as expected. If it's a normal data partition, mount it to make sure the files both exist and are appropriately accessible.

# Wiping disks with dd

Years ago, I had a friend who was responsible for security at his government's overseas embassies. He once told me that each embassy under his watch was provided with an official government-issue hammer. Why? In case the facility was ever at risk of being overrun by unfriendlies, the hammer was to be used to destroy all their hard drives. What's that? Why not just delete the data? You're kidding, right? Everyone knows that deleting files containing sensitive data from storage devices doesn't actually remove the data. Given enough time and motivation, nearly anything can be retrieved from virtually any digital media, with the possible exception of the ones that have been well and properly hammered.

You can, however, use dd to make it a whole lot more difficult for the bad guys to get at your old data. This command will spend some time writing millions and millions of zeros over every nook and cranny of the /dev/sda1 partition:

# dd if=/dev/zero of=/dev/sda1

But it gets better. Using /dev/urandom file as your source, you can write over a disk with random characters:

## dd if=/dev/urandom of=/dev/sda1

## Monitoring dd operations

Since disk or partition archiving can take a very long time, you might want to add a progress monitor to your command. Install Pipe Viewer (sudo apt install pv on Ubuntu) and insert it into dd. With pv, that last command might look something like this:

#### sudo apt install pv dd if=/dev/urandom | pv | dd of=/dev/sda1

4,14MB 0:00:05 [ 98kB/s] [ <=> ]

Putting off backups and disk management? With dd, you aren't left with too many excuses. It's really not difficult, but be careful.

Good luck!