

How to Repair a Corrupted USB Drive in Linux

MTE maketecheasier.com/repair-corrupted-usb-drive-linux

Everyone has a flash drive. They're great little things, and they make safeguarding data easy. However, sometimes flash drives can get corrupted or just flat out quit working. If you're a Linux user and this has happened to you, there's an easy fix to all of this. In this article we'll go over a few really simple tricks on how you can get your flash drive back in working order.

Note: the tricks outlined in this article aren't necessarily exclusive to USB drives and can be used on hard drives too.

Removing bad blocks from the USB drive with fsck

An easy way to repair a flash drive, or any drive really, is to use the `fsck` tool. This tool is great for removing bad file blocks, as most (if not all) corruption and unreadability comes from problems like this. To remove the corrupted file blocks from the USB flash drive, open a terminal window and enter the following commands.

Users must figure out what the drive designation is on the system before things can move forward. Do this by entering the `lsblk` command. This will list all of the attached disks on your system.

`lsblk`

Note: the `lsblk` command lists all disks, not just USB drives. Pay close attention to the output, as it is easy to mistake a hard drive for a flash drive.

```
fish /home/derrik
fish /home/derrik 72x21
derrik@Arch-Linux-Desktop -> lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0      0 298.1G  0 disk
├─sda1       8:1      0   450M  0 part
├─sda2       8:2      0   100M  0 part
├─sda3       8:3      0    16M  0 part
├─sda4       8:4      0 297.6G  0 part
sdb          8:16     0 931.5G  0 disk
├─sdb1       8:17     0   512M  0 part /boot
├─sdb2       8:18     0   931G   0 part /
sdc          8:32     1   7.6G   0 disk
└─sdc1       8:33     1   7.6G   0 part
```

To remove the bad file block, run the `fsck` command on either a specific partition (e.g. `/dev/sdc1`), or the entire disk (e.g. `/dev/sdc`). Once completed, the USB drive will have a healthy partition again and be fully operational on Linux.

```
sudo fsck /dev/sdc1
```

Note: this tutorial assumes that the flash drive is `/dev/sdc` (or `/dev/sdc1`). Users may have different labels for their flash drive on their system.

```
usage: fsck.vfat [-aAbflrtvVwy] [-d path -d ...] [-u path -u ...]
              device
  -a          automatically repair the filesystem
  -A          toggle Atari filesystem format
  -b          make read-only boot sector check
  -c N        use DOS codepage N to decode short file names (default: 437)
  -d path     drop that file
  -f          salvage unused chains to files
  -l          list path names
  -n          no-op, check non-interactively without changing
  -p          same as -a, for compat with other *fsck
  -r          interactively repair the filesystem (default)
  -t          test for bad clusters
  -u path     try to undelete that (non-directory) file
  -v          verbose mode
  -V          perform a verification pass
  -w          write changes to disk immediately
  -y          same as -a, for compat with other *fsck
```

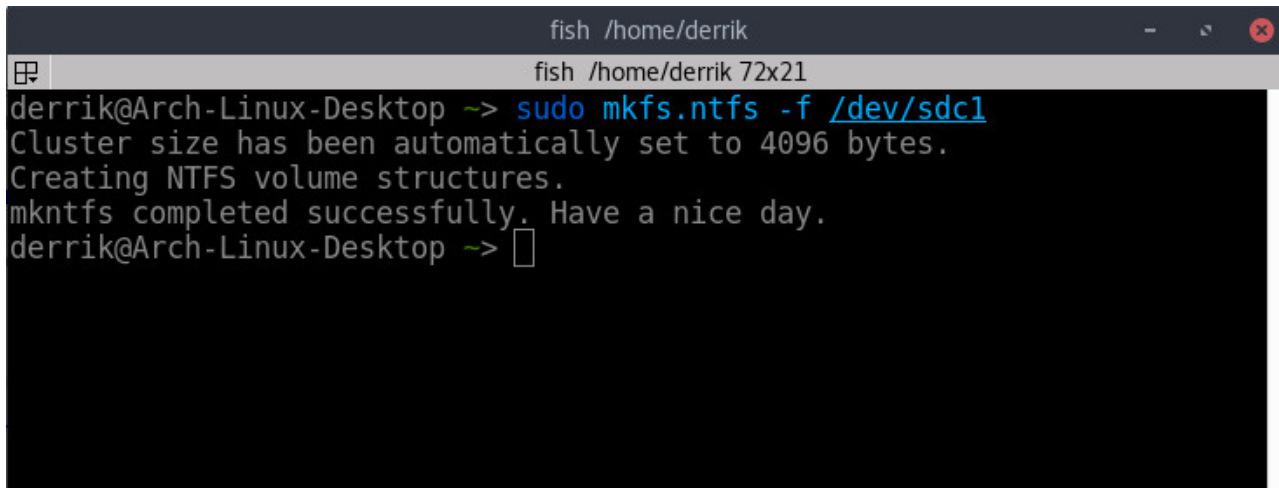
Zeroing the USB drive

Sometimes a USB drive can be totally unreadable to the point where it is no longer worth saving. When this happens the best route is often to just zero out the data and start over. The best tool for the job in this situation is `dd`, and it works quite well.

Start by taking the drive label that was found earlier with the `lsblk` command, and apply the same logic (remember that `/dev/sdc1` is a partition, and `/dev/sdc` is an entire device).

```
sudo dd if=/dev/zero of=/dev/sdc
```

Making a new file system

A terminal window titled 'fish /home/derrik' with a subtitle 'fish /home/derrik 72x21'. The prompt is 'derrik@Arch-Linux-Desktop ~->'. The command 'sudo mkfs.ntfs -f /dev/sdc1' has been entered. The output shows: 'Cluster size has been automatically set to 4096 bytes.', 'Creating NTFS volume structures.', 'mkntfs completed successfully. Have a nice day.', and the prompt 'derrik@Arch-Linux-Desktop ~->' with a cursor.

```
fish /home/derrik
fish /home/derrik 72x21
derrik@Arch-Linux-Desktop ~-> sudo mkfs.ntfs -f /dev/sdc1
Cluster size has been automatically set to 4096 bytes.
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
derrik@Arch-Linux-Desktop ~-> 
```

Zeroing a USB drive (or any device for that matter) renders the data on it totally useless. This means that you'll need to create a new data partition. Choose a file system, and then run the command!

Fat32

```
sudo mkfs.msdos -f 32 /dev/sdc1
```

Ext4

```
sudo mkfs.ext4 -f /dev/sdc1
```

NTFS

```
sudo mkfs.ntfs -f /dev/sdc1
```

Conclusion

USB flash drives are useful devices. They make it easy for people to easily transfer data from one computer to the other, regardless of the operating system it's running. That's why it's so important to know what to do when the drive is no longer accessible. Luckily, Linux ships with some really useful tools that make saving a flash drive quite easy.