File recovery

This article lists data recovery and undeletion options for Linux.

Related articles

/Post recovery tasks#Photorec

It is mostly intended to be used for educational purposes. If you have accidentally deleted or otherwise damaged your **valuable and irreplaceable** data on a technically healthy/ functional drive and have no previous experience with data recovery, **turn off** your computer immediately (just press and hold the off button or pull the plug; do not use the system shutdown function) and seek professional help.

> **Warning**
>
> It is quite possible and even probable that, if you follow any of the steps described below without fully understanding them, you will worsen your situation.

1 Special cases

1.1 Failing drives

In the area of data recovery, it is best to work on images of disks rather than physical disks themselves. Generally, a failing drive's condition worsens over time. The goal ought to be to first rescue as much data as possible as early as possible in the failure of the disk and to then abandon the disk. The **ddrescue (https://archlinux.org/packages/?name=ddrescue)** and **dd_rescue (https://archlinux.org/packages/?name=dd_rescue)** utilities, unlike `dd`, will repeatedly try to recover from errors and will read the drive front to back, then back to front, attempting to salvage data. They keep log files so that recovery can be paused and resumed without losing progress.

See **Disk cloning**.

The image files created from a utility like ddrescue can then be mounted like a physical device and can be worked on safely. Always make a copy of the original image so that you can revert if things go sour!

Do not attempt a filesystem check on a failing drive, as this will likely make the problem **worse**. Mount it read-only.

1.2 Backup flash media/small partitions

As an alternative to working with a 'live' partition (mounted or not), it is often preferable to work with an image, provided that the filesystem in question is not too large and that you have sufficient free HDD space to accommodate the image file. For example, flash memory devices like thumb drives, digital cameras, portable music players, cellular phones, etc. are likely to be small enough to image in many cases.

Be sure to read the man pages for the utilities listed below to verify that they are capable of working with image files.

To make an image, one can use `dd` as follows:

```
# dd if=/dev/target_partition of=/home/user/partition.image
```

1.3 Working with digital cameras

In order for some of the utilities listed in the next section to work with flash media, the device in question needs to be mounted as a block device (i.e., listed under /dev). Digital cameras operating in PTP (Picture Transfer Protocol) mode will not work in this regard. PTP cameras are transparently handled by libgphoto and/or libptp. In this case, "transparently" means that PTP devices do not get block devices. The alternative to PTP mode, USB Mass Storage (UMS) mode, is not supported by all cameras. Some cameras have a menu item that allows switching between the two modes; refer to your camera's user manual. If your camera does not support UMS mode and therefore cannot be accessed as a block device, your only alternative is to use a flash media reader and physically remove the storage media from your camera.

2 List of utilities

See also **Wikipedia:List of data recovery software#File Recovery**

- **ddrutility** — Compliment to GNU **ddrescue (https://archlinux.org/packages/?name=ddrescue)**. Find what files are related to the bad sectors and some special tools for NTFS. No longer actively supported.

  https://sourceforge.net/projects/ddrutility/ || **ddrutility (https://aur.archlinux.org/packages/ddrutility/)**[AUR]

- **dvdisaster** — Additional error protection for CD/DVD media.

  https://sourceforge.net/projects/dvdisaster/ || **dvdisaster (https://aur.archlinux.org/packages/dvdisaster/)**[AUR]

- **Foremost** — Console program to recover files based on their headers, footers, and internal data structures. This process is commonly referred to as data carving. The headers and footers can be specified by a configuration file or command line switches can be used to specify built-in file types.

  https://foremost.sourceforge.net/ || **foremost (https://archlinux.org/packages/?name=foremost)**

- **PhotoRec** — File data recovery software designed to recover lost files including video, documents and archives from hard disks, CD-ROMs, and lost pictures (thus the Photo Recovery name) from digital camera memory.

  https://www.cgsecurity.org/ || **testdisk (https://archlinux.org/packages/?name=testdisk)**

- **R-Linux** — A free file recovery utility for the Ext2/Ext3/Ext4 file systems.

  https://www.r-studio.com/free-linux-recovery/ || **r-linux (https://aur.archlinux.org/packages/r-linux/)**[AUR]

- **Scalpel** — File carving and indexing application originally based on **Foremost**, although significantly more efficient. It allows an examiner to specify a number of headers and footers to recover filetypes from a piece of media.

  https://github.com/sleuthkit/scalpel || **scalpel-git (https://aur.archlinux.org/packa**

ges/scalpel-git/)[AUR]

- **TestDisk** — Data recovery software primarily designed to help recover lost partitions and/or make non-booting disks bootable again when these symptoms are caused by faulty software: certain types of viruses or human error (such as accidentally deleting a Partition Table).

  https://www.cgsecurity.org/ || testdisk (https://archlinux.org/packages/?name=testdisk)

- **xfs_undelete** — Traverses the inode B+trees of each allocation group and tries to recover all files on an XFS filesystem marked as deleted.

  https://github.com/ianka/xfs_undelete || xfs_undelete-git (https://aur.archlinux.org/packages/xfs_undelete-git/)[AUR]

3 TestDisk and PhotoRec

TestDisk and Photorec are both open-source data recovery utilities licensed under the terms of the GNU Public License (https://www.gnu.org/licenses/gpl.html) (GPL).

TestDisk is primarily designed to help recover lost partitions and/or make non-booting disks bootable again when these symptoms are caused by faulty software, certain types of viruses, or human error, such as the accidental deletion of partition tables. TestDisk detects numerous filesystem including NTFS, FAT12, FAT16, FAT32, exFAT, ext2, ext3, ext4, btrfs, BeFS, CramFS, HFS, JFS, Linux Raid, Linux Swap, LVM, LVM2, NSS, ReiserFS, UFS, XFS. It can also undelete files from FAT, NTFS, exFAT and ext2 filesystem.

TestDisk allows to fix partition tables, recover deleted partitions, recover FAT32 boot sector from its backup, rebuild FAT12/FAT16/FAT32 boot sectors, fix FAT tables, rebuild NTFS boot sector and more.

PhotoRec is file recovery software designed to recover lost files including photographs (Hint: **Photo**graph**Rec**overy), videos, documents, archives from hard disks and CD-ROMs. PhotoRec ignores the filesystem and goes after the underlying data, so it will still work even with a re-formatted or severely damaged filesystems and/or partition tables.

3.1 Installation

Install the testdisk (https://archlinux.org/packages/?name=testdisk) package, which provides both TestDisk and PhotoRec.

3.2 Usage

After running e.g. ddrescue (https://archlinux.org/packages/?name=ddrescue) to create image.img, `photorec image.img` will open a terminal UI where you can select what file types to search for and where to put the recovered files. There is very good documentation on their wiki (https://www.cgsecurity.org/wiki/TestDisk) (step by step guide (https://www.cgsecurity.org/wiki/PhotoRec_Step_By_Step)).

3.3 Files recovered by photorec

The photorec utility stores recovered files with a random names(for most of the files) under a numbered directories, e.g. `./recup_dir.1/f872690288.jpg`, `./recup_dir.1/f864563104_wmclockmon-0.1.0.tar.gz`.

3.4 See also

- How to get the original filenames: **PhotoRec FAQ (https://www.cgsecurity.org/wiki/PhotoRec_FAQ#How_to_get_the_original_filenames_.3F)**
- How to add your own custom file signature: **CGSecurity Wiki (https://www.cgsecurity.org/wiki/Add_your_own_extension_to_PhotoRec)**
- Wiki (TestDisk): **https://www.cgsecurity.org/wiki/TestDisk**
- Wiki (Photorec): **https://www.cgsecurity.org/wiki/PhotoRec**
- Homepage: **https://www.cgsecurity.org/**

4 e2fsck

e2fsck is the ext2/ext3/ext4 filesystem checker included in the base install of Arch. *e2fsck* relies on a valid superblock. A superblock is a description of the entire filesystem's parameters. Because this data is so important, several copies of the superblock are distributed throughout the partition. With the `-b` option, *e2fsck* can take an alternate superblock argument; this is useful if the main, first superblock is damaged.

To determine where the superblocks are, run `dumpe2fs /dev/partition | grep superblock` as root on the target, unmounted partition. Superblocks are spaced differently depending on the filesystem's blocksize, which is set when the filesystem is created.

An alternate method to determine the locations of superblocks is to use the `-n` option with *mke2fs*. Be **sure** to use the `-n` flag, which, according to **mke2fs(8) (https://man.archlinux.org/man/mke2fs.8)**, "Causes mke2fs to not actually create a filesystem, but display what it would do if it were to create a filesystem. This can be used to determine the location of the backup superblocks for a particular filesystem, so long as the mke2fs parameters that were passed when the filesystem was originally created are used again. (With the `-n` option added, of course!)".

4.1 Installation

Both *e2fsck* and *dumpe2fs* are included in the base Arch install as part of **e2fsprogs (https://archlinux.org/packages/?name=e2fsprogs)**.

See also **e2fsck(8) (https://man.archlinux.org/man/e2fsck.8)** and **dumpe2fs(8) (https://man.archlinux.org/man/dumpe2fs.8)**.

5 Working with raw disk images

If you have backed up a drive using ddrescue or dd and you need to mount this image as a physical drive, see this section.

5.1 Mount the entire disk

To mount a complete disk image to the next free loop device, use the `losetup` command:

```
# losetup -f -P /path/to/image
```

Tip

- The `-f` flag mounts the image to the next available loop device.

- The `-P` flag creates additional devices for every partition.

See also **QEMU#With loop module autodetecting partitions**.

5.2 Mounting partitions

In order to be able to mount a partition of a whole disk image, follow **the steps above**.

Once the whole disk image is mounted, a normal `mount` command can be used on the loop device:

```
# mount /dev/loop0p1 /mnt/example
```

This command mounts the first partition of the image in loop0 to the folder to the mountpoint `/mnt/example`. Remember that the mountpoint directory must exist!

5.2.1 Getting disk geometry

Once the entire disk image has been mounted as a loopback device, its drive layout can be inspected.

5.3 Using QEMU to repair NTFS

With a disk image that contains one or more NTFS partitions that need to be `chkdsk`ed by Windows since no good NTFS filesystem checker for Linux exists, QEMU can use a raw disk image as a real hard disk inside a virtual machine:

```
# qemu -hda /path/to/primary.img -hdb /path/to/DamagedDisk.img
```

Then, assuming Windows is installed on `primary.img`, it can be used to check partitions on `/path/to/DamagedDisk.img`.

Warning

Do not use lower version of Windows to check NTFS partitions create by higher version of it, e.g. Windows XP can do damage to NTFS partitions created by Windows 8 by "fixing" **metadata** configuration that it does not support, resulting in damage/removal of these unsupported entries.

6 Text file recovery

It is possible to find deleted plain text files on a hard drive by directly searching on the block device. A preferably unique string from the file you are trying to recover is needed.

Use `grep` to search for fixed strings (`-F`) directly on the partition:

```
$ grep -a -C 200 -F 'Unique string in text file' /dev/sdXN > OutputFile
```

Hopefully, the content of the deleted file is now in *OutputFile*, which can be extracted from the surrounding context manually.

Note

The `-C 200` option tells grep to print 200 lines of context from before and after each match of the string. Alternatives are the `-A` and `-B` flags, which print context only from after and before each match, respectively. You may need to adjust the number of lines if the file you are looking for is very long.

7 See also

- **Data Recovery (https://help.ubuntu.com/community/DataRecovery)** on the Ubuntu wiki