

7 of the Best Data Recovery Tools for Linux - Make Tech Easier

Did your screen just turn black or your laptop freeze without warning? Maybe your hard drive has started chirping. Worse yet, maybe you suddenly can't save to your home partition.

All these signs of a corrupted or failing drive can cause you to sweat, but there's no reason to immediately throw away your computer. Keep reading to learn about six Linux tools that can help you retrieve your data and get your digital life back on track.

1. Trinity Rescue Kit

This live Linux distro made specifically for data recovery seemed to have been left for dead years ago, but a couple of years ago its developer turned up with an update out of nowhere that makes it run more effectively on modern computers.

```
Trinity Rescue Kit easy menu
| Welcome
| TRK Help -->
| Keyboard layout selection -->
| Windows password resetting -->
| Mount all local filesystems
| Unmount all local filesystems
| Virus scanning -->
| Windows junkfile cleaning -->
| Mclone: computer replication over the network -->
| Backup and restore utilities-->
| Run a windows fileserver -->
| Run an ssh server
| Set an ip-address on the first adapter
| TRK Network boot server
| Trinity Remote Support (contact us first)
| Ethernet packet sniffing -->
| Try detecting more harddisk controllers
| Try detecting more USB and PCMCIA network adapters
| Midnight Commander
| Go to a shell
| Go to a shell and save all output to /tmp/terminal.out
| Quit this menu
| Poweroff computer
| Reboot without ejecting CD / usb stick

Welcome to Trinity Rescue Kit 3.4 build 365
This simple menu allows you to perform simple actions that don't require any Linux knowledge and
gets things done in a quick and rather dirty way.
If you need a command line you can switch to the 5 other consoles by pressing ALT+<F2 to F6> or use
'Go to a shell' below
```

So what is [Trinity Rescue Kit](#)? Essentially, if your main Linux OS (or Windows or macOS for that matter) becomes unusable due to hard drive failure or some other catastrophe, you can create a TRK USB drive, then boot to it to pull data from your main OS using a command line. It's a rescue disk, and a really rather good one that has extra features like password recovery and virus scanning.

There is, of course, a bulk-undelete utility here too that scours your damaged or inaccessible hard drives for anything that can be recovered, and a couple of options for master boot record repair too. Be warned though, that there's quite a steep learning curve to really take advantage of all these features.

2. Testdisk

Certain Linux recovery tools attempt to save your data by copying whatever is retrievable from a damaged partition. This can work, but isn't guaranteed as issues with the partition itself may be obscuring the data you want to retrieve.

```
parallels@parallels-vm: ~
TestDisk 7.0, Data Recovery Utility, April 2015
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /dev/sdc - 4000 GB / 3726 GiB - 7814037168 sectors
Current partition structure:
    Partition              Start          End      Size in sectors
 1 P Linux Swap             30832         30847          16
 2 P Linux Raid            1032192       5031935     3999744 [md1]
 3 P Unknown                5031936       9031551     3999616
 4 P Unknown                9428990      7814035453  7804606464

                                P=Primary  D=Deleted
>[Quick Search]  [ Backup ]
                        Try to locate partition
```

Testdisk comes in handy as it can actually fix partition-level errors, making it easier to recover data using Testdisk itself or another data recovery tool. It may even outright fix a corrupted hard drive and make your Linux PC work again, so you can access the data directly.

As well as carrying out deep-diving operations like restoring boot sectors and filesystem tables, Testdisk is also capable of recovering data – even deleted data – from hard drives in the NTFS, FAT, exFAT and ext formats.

To install Testdisk on Linux, go to the Terminal and enter

```
sudo apt-get install testdisk
```

If you want to know more about using Testdisk, you can [read our Testdisk guide here](#).

3. ddrescue

While [ddrescue](#) isn't a data recovery tool in title, it should stand as your first step in the journey to retrieving your files. Ddrescue creates an image of your corrupt drive or partition so you can analyze a *copy* of your broken disk.

Always copy your disk to a separate image before you begin file recovery operations with the tools listed below. The more you use your actual failing drive, the more damage you can cause it.

```

1 2 3 4 5 6 7 8 9 [2] st st
[test@lappy ~]$ sudo ddrescue -f -n /dev/sdb1 ~/disks/backup.img ~/disks/rescue.log
GNU ddrescue 1.23
Press Ctrl-C to interrupt
      ipos: 247398 kB, non-trimmed:      0 B, current rate: 5160 kB/s
      opos: 247398 kB, non-scraped:      0 B, average rate: 9517 kB/s
non-tried:      0 B, bad-sector:      0 B, error rate: 0 B/s
      rescued: 247447 kB, bad areas:      0, run time: 25s
pct rescued: 100.00%, read errors:      0, remaining time: n/a
                                time since last successful read: n/a

Finished
[test@lappy ~]$ sudo ddrescue -d -f -r3 /dev/sdb1 ~/disks/backup.img ~/disks/rescue.log
GNU ddrescue 1.23
Press Ctrl-C to interrupt
Initial status (read from mapfile)
rescued: 247447 kB, tried: 0 B, bad-sector: 0 B, bad areas: 0

Current status
      ipos:      0 B, non-trimmed:      0 B, current rate:      0 B/s
      opos:      0 B, non-scraped:      0 B, average rate:      0 B/s
non-tried:      0 B, bad-sector:      0 B, error rate:      0 B/s
      rescued: 247447 kB, bad areas:      0, run time: 0s
pct rescued: 100.00%, read errors:      0, remaining time: n/a
                                time since last successful read: n/a

Finished

```

What you see here is some output of ddrescue in action. In the first command, it copies the entire disk to an image named “backup.img.” The second command then copies only the bad blocks to that same image, passing over those blocks three times each to try to read them.

When you run these same commands, always use a logfile. Backups can take hours or days to complete, and without a logfile, any interruptions will make you start the process again from the beginning.

When this process is complete for your own disk or partition, you can mount the copied image and use the following utilities to retrieve files from it. Further use of our other Linux recovery tools in this article will grab data from the same “backup.img” created here.

4. Foremost

[Foremost](#) uses the data structures of common file types to retrieve data. You can either scrape an entire disk image for all its files or specify certain file types you’re most interested in.

```

1 2 3 4 5 6 7 8 9 [2] st st
[test@lappy ~]$ foremost -v -t jpg -i ~/disks/backup.img -o ~/disks/foremost_recover/
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Wed Feb 21 13:35:36 2018
Invocation: foremost -v -t jpg -i /home/test/disks/backup.img -o /home/test/disks/foremos
Output directory: /home/test/disks/foremost_recover
Configuration file: /etc/foremost.conf
Processing: /home/test/disks/backup.img
|-----
File: /home/test/disks/backup.img
Start: Wed Feb 21 13:35:36 2018
Length: 235 MB (247447552 bytes)

Num      Name (bs=512)      Size      File Offset      Comment
0:      00165672.jpg      555 KB      84824240
1:      00166792.jpg      558 KB      85397504
2:      00167912.jpg       98 KB      85971118
3:      00168112.jpg       98 KB      86073344
4:      00168320.jpg      333 KB      86179840
5:      00168992.jpg      348 KB      86523904
6:      00169696.jpg      269 KB      86884352
7:      00171272.jpg      835 KB      87691264
8:      00173024.jpg      837 KB      88588288
*9:      00234520.jpg      597 KB     120074240
**|

Finish: Wed Feb 21 13:35:37 2018

10 FILES EXTRACTED

jpg:= 10
-----

```

Foremost finished at Wed Feb 21 13:35:37 2018

What you can see here is the output of Foremost in verbose mode (the `-v` option). The `-t` option searches for file types of jpg, and the `-i` and `-o` options mark the respective input file and output directory.

You can see that Foremost analyzes the image ddrescue created in the previous step; that image has a number of JPEGs in it. Foremost was able to find ten such files, and when it was done scraping the image, it copied those ten files to the stated output folder.

5. scalpel

[Scalpel](#), originally based on Foremost, aims to be frugal in its operation. It uses multi-threading and asynchronous input/output to search through images in an efficient manner. Moreover, it gives users the power to specify the number of footers and headers they wish to use for file recovery.

Users can also specify the types of files they want to recover by editing scalpel's configuration file. The default configuration produces a lot of output, even without verbose mode (`-v` parameter) turned on.

```

1 2 3 4 5 6 7 8 9 [2] st st
Adding /home/test/disks/scalpel_recover/zip-21-11/00011684.zip to queue
Adding /home/test/disks/scalpel_recover/zip-21-11/00011685.zip to queue
Adding /home/test/disks/scalpel_recover/zip-21-11/00011686.zip to queue
Adding /home/test/disks/scalpel_recover/zip-21-11/00011687.zip to queue
Adding /home/test/disks/scalpel_recover/zip-21-11/00011688.zip to queue
Adding /home/test/disks/scalpel_recover/zip-21-11/00011689.zip to queue
Work queues built. Workload:
gif with header "\x47\x49\x46\x38\x37\x61" and footer "\x00\x3b" --> 0 files
gif with header "\x47\x49\x46\x38\x39\x61" and footer "\x00\x00\x3b" --> 0 files
jpg with header "\xff\xd8\xff\xe0\x00\x10" and footer "\xff\xd9" --> 7 files
png with header "\x50\x4e\x47?" and footer "\xff\xfc\xfd\xfe" --> 30 files
tif with header "\x49\x49\x2a\x00" and footer "" --> 6 files
tif with header "\x4d\x4d\x00\x2a" and footer "" --> 6 files
avi with header "RIFF???AVI" and footer "" --> 0 files
mpg with header "\x00\x00\x01\xba" and footer "\x00\x00\x01\xb9" --> 116 files
mpg with header "\x00\x00\x01\xb3" and footer "\x00\x00\x01\xb7" --> 91 files
fws with header "FWS" and footer "" --> 0 files
wav with header "RIFF???WAVE" and footer "" --> 0 files
doc with header "\xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00" and footer "\xd0\xcf\x11\xe0\x"
doc with header "\xd0\xcf\x11\xe0\xa1\xb1" and footer "" --> 0 files
pst with header "\x21\x42\x4e\xa5\x6f\xb5\xa6" and footer "" --> 0 files
ost with header "\x21\x42\x44\x4e" and footer "" --> 0 files
dbx with header "\xcf\xad\x12\xfe\xc5\xfd\x74\x6f" and footer "" --> 0 files
idx with header "\x4a\x4d\x46\x39" and footer "" --> 0 files
mbx with header "\x4a\x4d\x46\x36" and footer "" --> 0 files
htm with header "<html" and footer "</html>" --> 0 files
pdf with header "%PDF" and footer "%EOF\x0d" --> 7 files
pdf with header "%PDF" and footer "%EOF\x0a" --> 5 files
zip with header "PK\x03\x04" and footer "\x3c\xac" --> 11422 files
java with header "\xca\xfe\xba\xbe" and footer "" --> 0 files
tgz with header "\x1f\x8b\x08\x08" and footer "" --> 0 files
ogg with header "x4fx67x67x53x00x02" and footer "x4fx67x67x53x00x02" --> 0 files
Carving files from image.
Image file pass 2/2.
/home/test/disks/backup.img: 100.0% |*****
Processing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 11690, elapsed = 326 secs.
[test@lappy ~]$ scalpel -o ~/disks/scalpel_recover/ ~/disks/backup.img

```

In that screenshot you can see the final output of scalpel's analysis of "backup.img." The basic command (listed at the bottom of the screenshot) requires only an output directory and an image for analysis.

6. PhotoRec

[PhotoRec](#) strays away from its competitors by focusing on the recovery of photographs, videos, and text documents. It also works as an interactive utility within the console itself. Check out all its majesty

```
1 2 3 4 5 6 7 8 9 [2] st st
```

```
PhotoRec 7.0, Data Recovery Utility, April 2015  
Christophe GRENIER <grenier@cgsecurity.org>  
http://www.cgsecurity.org
```

PhotoRec is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):

```
>Disk /home/test/disks/backup.img - 247 MB / 235 MiB (RO)
```

```
>[Proceed ] [ Sudo ] [ Quit ]
```

Note: **Some disks won't appear unless you're root user.**

Disk capacity must be correctly detected for a successful recovery.

If a disk listed above has incorrect size, check HD jumper settings, BIOS
detection, and install the latest OS patches and disk drivers.

The initial PhotoRec command must specify a desired image (our backup.img) and output folder. PhotoRec then drops the user into its graphical environment. The screenshot here shows the image size. In further screens it asks for the disk's partition type and whether or not you want to search the entire image for files.

7. grep

Finally, we come to [grep](#). This may also not seem like the most straightforward of recovery options, but grep has the power to [find deleted or lost text files](#) by searching for strings present on a block device or disk image.

A file exists in backup.img called "myfile." It contains only one line of text: "This is the file I will try to recover."

Grep uses that string as its starting point for file recovery. Alongside a few other parameters, you can see that, in this example, it dumps the found string into a new binary file named "foundtext."

```
1 2 3 4 5 6 7 8 9 [2] st st
[test@lappy disks]$ grep -a -C 1 -F 'This is the file' ~/disks/backup.img > foundtext
```

In particular, you will want to pay attention to – and modify – the `-c` parameter which prints extra context surrounding the string in the initial command. This example command tells `grep` to find one line of text before and one after the provided string.

Starting at `-c 200`, `grep` would find 200 lines both before and after a string. Such reach is unnecessary here, but it could be important for larger text files with hundreds of lines. You will, of course, need to know the text of your own files so `grep` has a starting point to begin its search.

`Grep` will generate a binary file as its output. Still, some parts will be human-readable, such as this example's desired line of text near the bottom of this screenshot. It will be your job to manually scrape out the data you need. It's tough work, for sure, but it beats the alternative of having no file at all.

`grep`

Conclusion

In summary, be sure to first copy your drive or partition with `ddrescue`, then work on that copy with any of the other Linux recovery tools you need. Don't be afraid to try more than one tool, especially if your first choice didn't find the data you wanted.

Be patient. With any luck, you'll have your precious files back before you know it.

This article was first published in Feb 2018 and was updated in March 2019.

Is this article useful?