How to completely delete a file in Linux so that it cannot be restored

In Linux, files can be deleted but can still be recovered. These are things to do when you really want them to disappear.

In Linux, files can be deleted but can still be recovered. These are things to do when you really want them to disappear.

The **rm** command easily makes the files disappear from your file list, but what will it actually do and how can you ensure that deleted files cannot be recovered anymore?

Basic concepts about removing a file completely in Linux

To understand what happens when you delete a file from a Linux system using rm, first think of inodes - attractive data structures that track all file attributes (often called "metadata"). ") - description of the file. This includes the name, owner and group of the file, which permissions are set and where the content of the file can be found on the drive.

Next, learn about Linux directories. Although they have the look and properties of the directories (ie, containing only files), they are actually just files and include only the name and the inode number of the files they contain. Therefore, it is better to consider directories and files in Linux like folders and paperwork in your filing cabinet.

The contents of the actual files are stored, usually in blocks, elsewhere on your drive. So when you remove a file, the inode of the file is released and the directory file adjusts to remove its references to the file that has just been deleted. The file's data will remain on the drive location and will then be provided for reuse.

How and when to delete files in Linux?

In most cases, it is not important that the contents of deleted files remain on the drive. They are usually files you don't need anymore, like last month's weekly reports, previous versions of scripts that have been replaced with better scripts, etc. When these files can generate problems There are a number of tools that you can use to reduce the possibility of others recovering them.

shred

The shred command will overwrite a file and also delete it. Overwriting will ensure that data blocks containing old content now have other content. Note, by default, content is overwritten in multiple writes. The options used below include **z** (add the last override with the number 0), **v** (show what is going on), and **u** (truncate and then

delete the file after overwriting is finished) . These options make the contents of the file completely overwritten.

```
$ shred -zvu passwords-save
shred: passwords-save: pass 1/4 (random) .
shred: passwords-save: pass 2/4 (random) .
shred: passwords-save: pass 3/4 (random) .
shred: passwords-save: pass 4/4 (000000) .
shred: passwords-save: remove
shred: passwords-save: renamed to 00000000000000
shred: passwords-save: removed
```

wipe

The **wipe** command works only for **magnetic media** , not hard drives. It works with the so-called " **magnetic force microscopy** ", allowing others to recover two or three "data" in the end, which can be written to your drive, but as stated. above it only works with magnetic media, not hard drives and not all drives are eligible, you can determine the type of system drive you have and see if wipe can work with them. by looking at the result from this command, place **0 = SSD** (magnetic media) and **1 = HDD** (hard drive):

```
$ cat / sys / block / sda / queue / rotational
0
```

Below is an example of a wipe command when operating:

```
$ wipe -rfi temp
Entering directory 'temp'
Wiping mno, pass 34 (34)
Mno file (340 bytes) wiped
Wiping fileA, pass 34 (0)
FileA file (808 bytes) wiped
Wiping klm, pass 34 (0)
File klm (1056 bytes) wiped
Wiping lmn, pass 34 (0)
File lmn (3827 bytes) wiped
Wiping fileC, pass 34 (0)
File file (842 bytes) wiped
Wiping myfiles.tar, pass 34 (0)
File myfiles.tar (122880 bytes) wiped
Wiping fileB, pass 34 (0)
FileB file (5092 bytes) wiped
Going back to directory / home / shs
Operation finished.
7 các t?p tin wiped và 0 t?p tin này b? b? qua trong m?t th? m?
c, 0 symlinks removed, nh?ng không theo sau, 0 l?i ?ã th?c hi?n
```

In this example, r will receive a wipe command to recurse into directories if they exist, f will avoid having to verify that each file is corrupted, and i make the command run in detail ("i" is " **informative** "). ").

secure-delete

Another tool to completely delete files is called " **secure-delete** ", although the command that the package will add to your system is called srm as in " **secure rm** ".

Here's an example of how to use this tool:

```
$ srm -vz BoD_meeting
?ang dùng / dev / urandom cho d? li?u nh?p vào.
Wipe mode is secure (38 special passes)
Wiping BoD_meeting ************************************ Removed the BoD_meeti
```

Note the number of passes that have been made to ensure safe file removal.

Check your file deletion

Because there are several options on how to safely remove files from the Linux system, the following example will run a simple test - create some copies of the same file and safely delete them each The tool is discussed above. Here's how to use a tool called foremost to try to recover files of the same type, from the affected partition.

```
$ shred -zvu penguin1.png
$ wipe -fi penguin2.png
$ srm -vz penguin3.png
$ sudo foremost -i / dev / sda1 -t png -o / root / rescued
Processing: / dev / sda1
| ************************************************* ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************************************** ****************************
************************** |
```

The **foremost** command will take a few minutes to run. This command found more than 51,000 **.png** files in the example. In any case, it should be noted that the files taken from your drive during the recovery process will not have the original name, as they are lost when the affected folder file has been modified when deleting the file. Instead, your restored files will have the same name as **105210720.png** .

The foremost command has options to select the type of file you want to recover. Check the **man** page for details. And remember that it cannot limit your search to a specific directory because, at this time, folders are no longer relevant.

An important note

If you want to test the complete deletion of files from the system and then use the foremost tool to see what can be recovered, consider restoring files to extract media or at least not simply delete files , when you are testing them, or basically, you will double the number of recovery files in the next test. The restored files are independent of the original files, even though they have the same content.

See more:

1. How to copy and rename files in Linux
2. 7 commands to manipulate the most basic files and folders everyone must know
3. Search for files and directories in Linux using the command line interface