How to unmount a busy device

Asked 9 years, 1 month ago Active 1 month ago Viewed 680k times

284

92

I've got some samba drives that are being accessed by multiple users daily. I already have code to recognize shared drives (from a SQL table) and mount them in a special directory where all users can access them.

I want to know, if I remove a drive from my SQL table (effectively taking it offline) how, or even is, there a way to unmount a busy device? So far I've found that any form of `umount` does not work.

Ignoring the possibility of destroying data - is it possible to unmount a device that is currently being read?

linux umount

edited Mar 12 '19 at 10:52

Javier Arias
1,371 ● 2 ● 10 ● 24

asked Oct 24 '11 at 16:22

Max
3,919 ● 2 ● 18 ● 27

3 A more general answer addressing more causes for failing umount is found here oletange.blogspot.dk/2012/04/umount-device-is-busy-why.html – Ole Tange Jun 22 '14 at 11:13

3 Hello, probably you `cd` to mounted dir, then you became root or login again then the other shell is trapped. Do `exit` on all shells. – Smeterlink May 5 at 23:24

12 Answers

Active Oldest Votes

522

YES!! There is a way to detach a busy device immediately (even if it is busy and cannot be unmounted forcefully). You may cleanup all later:

```
umount -l /PATH/OF/BUSY-DEVICE
umount -f /PATH/OF/BUSY-NFS (NETWORK-FILE-SYSTEM)
```

NOTE:

1. These commands can disrupt a running process, cause data loss OR corrupt open files. Programs accessing target DEVICE/NFS files may throw errors OR could not work properly after force unmount.
2. Do execute these commands when **not** inside mounted Folder/Drive/Device.

edited Aug 7 at 20:29

answered Nov 14 '13 at 4:17
Amit Verma


5,922 ●5 ●28 ●35

28 Note: `-l` here is a lowercase `L` (for "lazy unmounting"). (See this related answer.) – ジョージ Feb 21 '14 at 7:47

4 Worked. One nuance, if you are logged in through FTP client, you have to logout in order to successfully unmount folder. – Alexander Kim Oct 22 '14 at 16:47

1 `-l` / `--lazy` won't corrupt open files, but on Linux it seems you can't know when the device is actually unmounted and can be removed – Tom Hale Aug 12 '17 at 6:14

1 Kinda scary. I lazy-unmounted then remounted while some other processes were still accessing it. So I guess I mounted it twice in the end to the same location? Not sure what that did. – sudo Sep 8 '17 at 0:42

1 Hi, it's very likely some process has some file descriptor open on your mount partition, follow the thread you have to use 'lsof' command to diagnostic and see if some process has an open file. In the case of the data base you should kill the processes or close the database before umount the partition. Use -L or -F can hide the real problem about the umount is failing and is not advisable. – Javier Gutiérrez-Maturana Sánc Jun 16 at 11:02

138

If possible, let us locate/identify the busy process, kill that process and then `unmount` the samba share/ drive to minimize damage:

- `lsof | grep '<mountpoint of /dev/sda1>'` (or whatever the mounted device is)
- `pkill target_process` (kills busy proc. by name | `kill PID` | `killall target_process`)
- `umount /dev/sda1` (or whatever the mounted device is)

edited Oct 4 at 21:02

Inspired_Blue
1,563 ●3 ●8 ●17

answered Oct 24 '11 at 16:27

Frank Tudor
3,196 ●2 ●20 ●42

8 That doesn't return anything. I'm assuming its because its a network drive and I can't see the processes of other computers accessing the drive. Same deal with the "fuser" commands. – Max Oct 24 '11 at 16:40

oh hell... you need the samba commands... /usr/bin/smbclient service <password>: See if this gets you started...tldp.org/HOWTO/SMB-HOWTO-8.html – Frank Tudor Oct 24 '11 at 17:17

2 The smb commands have actually been deprecated and replaced by "umount.cifs" .... which also doesn't work. It appears that I'm stuck with not being able to umount while its busy. – Max Oct 24 '11 at 18:00

If you are using Asuswrt-Merlin, you need to install `lsof`: `# opkg install lsof` – Tonatio Jul 24 '19 at 17:18

1 you need to sudo lsof to get some results – aheigins Apr 25 at 10:14

88

Make sure that you aren't still in the mounted device when you are trying to umount.

answered Apr 28 '17 at 19:05

Luci
889 ●5 ●2

5 Exactly, simply having current folder (located on target device) opened in your terminal (through e.g. cd command) is enaugh to stop the unmnounting process :) – jave.web Jan 28 '18 at 14:13

2 Yep, I had a shell running in a directory on the device. Closed the terminal window and voila – sh78 Dec 31 '18 at 20:31

Also, make sure there are not any other mount points inside the one you are trying to `umount`. – victe Jul 19 '19 at 8:19

@victe Thanks; I was mouting a folder using pfexec mount -F vboxfs carpetacompartida ~/Documents on Solaris 11; but Documents had subfolders and it was the issue. – Dani Aya Jan 29 at 20:44

44

Try the following, but before running it **note that the `-k` flag will kill any running processes keeping the device busy.**

The `-i` flag makes `fuser` ask before killing.

```
fuser -kim /address  # kill any processes accessing file
unmount /address
```

edited Aug 24 '16 at 17:14 — Arel — 962 ● 11 ● 19

answered Jun 30 '14 at 7:44 — user3751769 — 549 ● 4 ● 3

5 `lsof | grep '/dev/<my-device>` didn't return anything, but this works great! May want to also suggest `fuser -m /dev/<my-device>` in case you want to find out the process before killing it. – modulitos Feb 10 '16 at 8:26

3 Running the fuser command immediately disconnected me from the VPS. – giorgio79 May 30 '16 at 13:06

There's no fuser in ubuntu 18.04 – Freedo Oct 2 at 22:22

36

Avoid `umount -l`

At the time of writing, the top-voted answer recommends using `umount -l`.

`umount -l` **is dangerous or at best unsafe**. In summary:

- It doesn't actually unmount the device, it just removes the filesystem from the namespace. Writes to open files can continue.
- It can cause btrfs filesystem corruption

Work around / alternative

The useful behaviour of `umount -l` is hiding the filesystem from access by *absolute* pathnames, thereby minimising further moutpoint usage.

This same behaviour can be achieved by mounting an empty directory with permissions `000` over the directory to be unmounted.

Then any new accesses to filenames in the below the mountpoint will hit the newly overlaid directory with zero permissions - new blockers to the unmount are thereby prevented.

First try to `remount,ro`

The major unmount achievement to be unlocked is the read-only remount. When you gain the `remount,ro` badge, you know that:

1. All pending data has been written to disk
2. All future write attempts will fail
3. The data is in a consistent state, should you need to physcially disconnect the device.

`mount -o remount,ro /dev/device` is guaranteed to fail if there are files open for writing, so try that straight up. You may be feeling lucky, punk!

If you are unlucky, focus only on processes with files open for writing:

```
lsof +f -- /dev/<devicename> | awk 'NR==1 || $4~/[0-9]+[uw -]/'
```

You should then be able to remount the device read-only and ensure a consistent state.

If you can't remount read-only at this point, investigate some of the other possible causes listed here.

Read-only re-mount achievement unlocked 🔒☑

Congratulations, your data on the mountpoint is now consistent and protected from future writing.

Why `fuser` is inferior to `lsof`

Why not use use `fuser` earlier? Well, you could have, but `fuser` operates upon a *directory*, not a *device*, so if you wanted to remove the mountpoint from the file name space and still use `fuser`, you'd need to:

1. Temporarily duplicate the mountpoint with `mount -o bind /media/hdd /mnt` to another location
2. Hide the original mount point and block the namespace:

Here's how:

```
null_dir=$(sudo mktemp --directory --tmpdir empty.XXXXX")
sudo chmod 000 "$null_dir"

# A request to remount,ro will fail on a `-o bind,ro` duplicate if there are
# still files open for writing on the original as each mounted instance is
```

```
# checked.  https://unix.stackexchange.com/a/386570/143394
# So, avoid remount, and bind mount instead:
sudo mount -o bind,ro "$original" "$original_duplicate"

# Don't propagate/mirror the empty directory just about hide the original
sudo mount --make-private "$original_duplicate"

# Hide the original mountpoint
sudo mount -o bind,ro "$null_dir" "$original"
```

You'd then have:

1. The original namespace hidden (no more files could be opened, the problem can't get worse)
2. A duplicate bind mounted directory (as opposed to a device) on which to run `fuser`.

This is more convoluted[1], but allows you to use:

```
fuser -vmMkiw <mountpoint>
```

which will interactively ask to kill the processes with files open for writing. Of course, you could do this without hiding the mount point at all, but the above mimicks `umount -l`, without any of the dangers.

The `-w` switch restricts to writing processes, and the `-i` is interactive, so after a read-only remount, if you're it a hurry you could then use:

```
fuser -vmMk <mountpoint>
```

to kill all remaining processes with files open under the mountpoint.

Hopefully at this point, you can unmount the device. (You'll need to run `umount` on the mountpoint twice if you've bind mounted a mode `000` directory on top.)

Or use:

```
fuser -vmMki <mountpoint>
```

to interactively kill the remaining read-only processes blocking the unmount.

Dammit, I still get `target is busy`!

Open files aren't the only unmount blocker. See here and here for other causes and their remedies.

Even if you've got some lurking gremlin which is preventing you from fully unmounting the device, you have at least got your filesystem in a consistent state.

You can then use `lsof +f -- /dev/device` to list all processes with open files on the device containing the filesystem, and then kill them.

---

[1] It is less convoluted to use `mount --move`, but that requires `mount --make-private /parent-mount-point` which has implications. Basically, if the mountpoint is mounted under the `/` filesystem, you'd want to avoid this.

edited Jan 24 at 15:33

answered Sep 26 '19 at 16:41

Tom Hale
21.1k ● 10 ● 111 ● 155

3 If `--lazy` is so dangerous, why is there not so much as a warning in the `umount` man page? All it says is "*Lazy unmount. Detach the filesystem from the file hierarchy now, and clean up all references to this filesystem as soon as it is not busy anymore.*" – bitinerant Jan 23 at 14:26

7

Check for exported NFS file systems with exportfs -v. If found, remove with exportfs -d share:/directory. These don't show up in the fuser/lsof listing, and can prevent umount from succeeding.

answered Apr 20 '14 at 15:02

numberer6
101 ● 1 ● 1

1 Thanks for this advice. I had to use exportfs -ua to remove the lock. – FuePi Apr 14 '15 at 8:36

6

Check out `umount2`:

> Linux 2.1.116 added the umount2() system call, which, like umount(), unmounts a target, but allows additional flags controlling the behaviour of the operation:
>
> MNT_FORCE (since Linux 2.1.116) Force unmount even if busy. (Only for NFS mounts.) MNT_DETACH (since Linux 2.4.11) Perform a lazy unmount: make the mount point unavailable for new accesses, and actually perform the unmount when the mount point ceases to be busy. MNT_EXPIRE (since Linux 2.6.8) Mark the mount point as expired. If a mount point is not currently in use, then an initial call to umount2() with this flag fails with the error EAGAIN, but marks the mount point as expired. The mount point remains expired as long as it isn't accessed by any process. A second umount2() call specifying MNT_EXPIRE unmounts an expired mount point. This flag cannot be specified with either MNT_FORCE or MNT_DETACH. Return Value
>
> On success, zero is returned. On error, -1 is returned, and errno is set appropriately.

answered Oct 24 '11 at 16:26

chown
47.9k ● 16 ● 127 ● 165

Unforunately these aren't NFS mounts, but CIFS. I will try the MNT_DETACH though. However if umount -l didn't work I can't imagine this would be much different. Thanks though! – Max Oct 24 '11 at 16:35

3

Someone has mentioned that if you are using terminal and your current directory is inside the path which you want to unmount, you will get the error.
As a complementary, in this case, your `lsof | grep path-to-be-unmounted` must have below output:

```
bash ... path-to-be-unmounted
```

answered Mar 30 at 8:17

Eugene
7,927 ● 3 ● 30 ● 50

1

Another alternative when anything works is editing `/etc/fstab`, adding `noauto` flag and rebooting the machine. The device won't be mounted, and when you're finished doing whatever, remove flag and reboot again.

answered May 31 '18 at 7:55

jesjimher
999 ● 1 ● 8 ● 15

1

Just in case someone has the same pb. :

I couldn't unmount the mount point (here `/mnt`) of a chroot jail.

Here are the commands I typed to investigate :

```
$ umount /mnt
umount: /mnt: target is busy.
$ df -h | grep /mnt
/dev/mapper/VGTout-rootFS  4.8G  976M  3.6G  22% /mnt
$ fuser -vm /mnt/
                     USER        PID ACCESS COMMAND
/mnt:                root     kernel mount /mnt
$ lsof +f -- /dev/mapper/VGTout-rootFS
$
```

As you can notice, even `lsof` returns nothing.

Then I had the idea to type this :

```
$ df -ah | grep /mnt
/dev/mapper/VGTout-rootFS  4.8G  976M  3.6G  22% /mnt
dev                        2.9G     0  2.9G   0% /mnt/dev
$ umount /mnt/dev
$ umount /mnt
$ df -ah | grep /mnt
$
```

Here it was a `/mnt/dev` bind to `/dev` that I had created to be able to repair my system inside from the chroot jail.

After umounting it, my pb. is now solved.

edited Oct 1 at 23:50

answered Oct 1 at 22:19

SebMa

1,857 ● 11 ● 21

0

Niche Answer:

If you have a zfs pool on that device, at least when it's a file-based pool, `lsof` will not show the usage. But you can simply run

```
sudo zpool export mypoo
```

and then unmount.

answered May 17 at 6:35

lucidbrot

3,192 ● 2 ● 24 ● 52

0

Multiple mounts inside a folder

An additional reason could be a *secondary mount inside your primary mount folder*, e.g. after you worked on an SD card for an embedded device:

```
# mount /dev/sdb2 /mnt       # root partition which contains /boot
# mount /dev/sdb1 /mnt/boot  # boot partition
```

Unmounting /mnt will fail:

```
# umount /mnt
umount: /mnt: target is busy.
```

First we have to unmount the boot folder and then the root:

```
# umount /mnt/boot
# umount /mnt
```

answered Aug 30 at 15:13

gökçe
1