NAME

umount − unmount filesystems

SYNOPSIS

umount −a [**−dflnrv**] [**−t** *fstype*] [**−O** *option*...]

umount [**−dflnrv**] {*directory*|*device*}

umount −h|**−V**

DESCRIPTION

The **umount** command detaches the mentioned filesystem(s) from the file hierarchy. A filesystem is specified by giving the directory where it has been mounted. Giving the special device on which the filesystem lives may also work, but is obsolete, mainly because it will fail in case this device was mounted on more than one directory.

Note that a filesystem cannot be unmounted when it is 'busy' − for example, when there are open files on it, or when some process has its working directory there, or when a swap file on it is in use. The offending process could even be **umount** itself − it opens libc, and libc in its turn may open for example locale files. A lazy unmount avoids this problem, but it may introduce other issues. See **−−lazy** description below.

OPTIONS

−a, **−−all**

All of the filesystems described in */proc/self/mountinfo* (or in deprecated */etc/mtab*) are unmounted, except the proc, devfs, devpts, sysfs, rpc_pipefs and nfsd filesystems. This list of the filesystems may be replaced by **−−types** umount option.

−A, **−−all−targets**

Unmount all mountpoints in the current mount namespace for the specified filesystem. The filesystem can be specified by one of the mountpoints or the device name (or UUID, etc.). When this option is used together with **−−recursive**, then all nested mounts within the filesystem are recursively unmounted. This option is only supported on systems where */etc/mtab* is a symlink to */proc/mounts*.

−c, **−−no−canonicalize**

Do not canonicalize paths. The paths canonicalization is based on **stat**(2) and **readlink**(2) system calls. These system calls may hang in some cases (for example on NFS if server is not available). The option has to be used with canonical path to the mount point.

This option is silently ignored by **umount** for non−root users.

For more details about this option see the **mount**(8) man page. Note that **umount** does not pass this option to the **/sbin/umount.***type* helpers.

−d, **−−detach−loop**

When the unmounted device was a loop device, also free this loop device. This option is unnecessary for devices initialized by **mount**(8), in this case "autoclear" functionality is enabled by default.

−−fake

Causes everything to be done except for the actual system call or umount helper execution; this 'fakes' unmounting the filesystem. It can be used to remove entries from the deprecated */etc/mtab* that were unmounted earlier with the **−n** option.

−f, **−−force**

Force an unmount (in case of an unreachable NFS system).

Note that this option does not guarantee that umount command does not hang. It’s strongly recommended to use absolute paths without symlinks to avoid unwanted **readlink**(2) and **stat**(2) system calls on unreachable NFS in **umount**.

−i, **−−internal−only**

Do not call the **/sbin/umount.***filesystem* helper even if it exists. By default such a helper program is called if it exists.

−l, **−−lazy**

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.

A system reboot would be expected in near future if you’re going to use this option for network filesystem or local filesystem with submounts. The recommended use−case for **umount −l** is to prevent hangs on shutdown due to an unreachable network share where a normal **umount** will hang due to a downed server or a network partition. Remounts of the share will not be possible.

−N, **−−namespace** *ns*

Perform **umount** in the mount namespace specified by *ns*. *ns* is either PID of process running in that namespace or special file representing that namespace.

umount switches to the namespace when it reads */etc/fstab*, writes */etc/mtab* (or writes to */run/mount*) and calls **umount**(2) system call, otherwise it runs in the original namespace. It means that the target mount namespace does not have to contain any libraries or other requirements necessary to execute **umount**(2) command.

See **mount_namespaces**(7) for more information.

−n, **−−no−mtab**

Unmount without writing in */etc/mtab*.

−O, **−−test−opts** *option...*

Unmount only the filesystems that have the specified option set in */etc/fstab*. More than one option may be specified in a comma−separated list. Each option can be prefixed with **no** to indicate that no action should be taken for this option.

−q, **−−quiet**

Suppress "not mounted" error messages.

−R, **−−recursive**

Recursively unmount each specified directory. Recursion for each directory will stop if any unmount operation in the chain fails for any reason. The relationship between mountpoints is determined by */proc/self/mountinfo* entries. The filesystem must be specified by mountpoint path; a recursive unmount by device name (or UUID) is unsupported. Since version 2.37 it umounts also all over−mounted filesystems (more filesystems on the same mountpoint).

−r, **−−read−only**

When an unmount fails, try to remount the filesystem read−only.

−t, **−−types** *type...*

Indicate that the actions should only be taken on filesystems of the specified *type*. More than one type may be specified in a comma−separated list. The list of filesystem types can be prefixed with **no** to indicate that no action should be taken for all of the mentioned types. Note that **umount** reads information about mounted filesystems from kernel (*/proc/mounts*) and filesystem names may be different than filesystem names used in the */etc/fstab* (e.g., "nfs4" vs. "nfs").

−v, **−−verbose**
Verbose mode.

−h, **−−help**
Display help text and exit.

−V, **−−version**
Print version and exit.

NON−SUPERUSER UMOUNTS

Normally, only the superuser can umount filesystems. However, when *fstab* contains the **user** option on a line, anybody can umount the corresponding filesystem. For more details see **mount**(8) man page.

Since version 2.34 the **umount** command can be used to perform umount operation also for fuse filesystems if kernel mount table contains user's ID. In this case *fstab* **user=** mount option is not required.

Since version 2.35 **umount** command does not exit when user permissions are inadequate by internal **libmount** security rules. It drops suid permissions and continue as regular non−root user. This can be used to support use−cases where root permissions are not necessary (e.g., fuse filesystems, user namespaces, etc).

LOOP DEVICE

The **umount** command will automatically detach loop device previously initialized by **mount**(8) command independently of */etc/mtab*.

In this case the device is initialized with "autoclear" flag (see **losetup**(8) output for more details), otherwise it's necessary to use the option **−−detach−loop** or call **losetup −d** *device*. The autoclear feature is supported since Linux 2.6.25.

EXTERNAL HELPERS

The syntax of external unmount helpers is:

umount.*suffix* {*directory*|*device*} [**−flnrv**] [**−N** *namespace*] [**−t** *type.subtype*]

where *suffix* is the filesystem type (or the value from a **uhelper=** or **helper=** marker in the mtab file). The **−t** option can be used for filesystems that have subtype support. For example:

umount.fuse −t fuse.sshfs

A **uhelper=***something* marker (unprivileged helper) can appear in the */etc/mtab* file when ordinary users need to be able to unmount a mountpoint that is not defined in */etc/fstab* (for example for a device that was mounted by **udisks**(1)).

A **helper=***type* marker in the *mtab* file will redirect all unmount requests to the **/sbin/umount.***type* helper independently of UID.

Note that */etc/mtab* is currently deprecated and **helper=** and other userspace mount options are maintained by **libmount**.

ENVIRONMENT

LIBMOUNT_FSTAB=<path>
overrides the default location of the *fstab* file (ignored for **suid**)

LIBMOUNT_DEBUG=all
enables **libmount** debug output

FILES

/etc/mtab

table of mounted filesystems (deprecated and usually replaced by symlink to */proc/mounts*)

/etc/fstab

table of known filesystems

/proc/self/mountinfo

table of mounted filesystems generated by kernel.

HISTORY

A **umount** command appeared in Version 6 AT&T UNIX.

SEE ALSO

umount(2), **losetup**(8), **mount_namespaces**(7), **mount**(8)

REPORTING BUGS

For bug reports, use the issue tracker at https://github.com/util−linux/util−linux/issues.

AVAILABILITY

The **umount** command is part of the util−linux package which can be downloaded from Linux Kernel Archive <https://www.kernel.org/pub/linux/utils/util−linux/>.