NAME

rm — remove files and directory trees

SYNOPSIS

rm [**-rRdvx**]
[**-f**| **-iI**| **--interactive**[={**yes**|**always**}|{**validate**|**once**}|{**no**|**never**|**none**}]]
[**--no-preserve-root**| **--preserve-root**[=**all**]] *file*...

DESCRIPTION

Removes each *file*. With **-d**, removes empty directories (à la rmdir(1)), too; with **-r** — directory trees. Symbolic links are *never* followed.

A file ceases to exist when it has no links, open descriptions, or mappings. Write credentials are required only to the directory that contains a file, not the file itself.

. and **..** may not be deleted (and are not recursed down (or up, as the case may be) — removing [path/]**.*** *only* removes hidden files from *path*).

Interactive Modes

By default (no **-fi**-family flags or **-I**), if write permission is denied for a given file *and* the standard input stream is a teletype, a prompt is issued to the standard error stream with the file's path and metadata, asking to confirm deletion. If the user agrees, or the prompt wasn't issued, the file is removed (if a directory with **-r** — entered); if not, it's skipped.

With **-i**, the same happens to *all* files, *regardless* of the standard input stream's type. Additionally, with **-r**, a separate prompt is issued for entering a directory and removing it (provided all of its entries were removed).

With **-f** and **--interactive**=**no**, the prompt is never issued, and removal always proceeds.

-I is different entirely: per-file prompting happens as in the default mode, but there is an additional prompt issued before any processing is done, provided **-r** or more than **3** *file*s were specified. Unless the user confirms the arguments were as-expected, **rm** exits **0**.

OPTIONS

-r, **-R**, **--recursive** Descend down directory trees.

-d, **--dir** Remove *file*s which are empty directories.

-v, **--verbose** Write paths to deleted files to the standard output stream.

-x, **--one-file-system** Don't remove or descend to filesystems (mount-points) different than their corresponding *file*.

-f, **--force** **--interactive**=**no**, ignore nonexistent *file*s, and allow an empty *file* list.

-i, **--interactive**[=**yes**|**always**] Disables **-f**, asks for every path (see above).

-I, **--interactive**=**validate**|**once** Disables **-f**, asks once before the entire operation (see above).

--interactive=**no**|**never**|**none** Disables **-f**, never asks for confirmation (see above).
All **--interactive** values are prefix-matched (**--interactive**=*n* is equivalent to **--interactive**=**none**, &c.).

--no-preserve-root Allow removing *file*s equivalent to /.

--preserve-root Forbid removing *file*s equivalent to /. This is the default.

--preserve-root=**all** Forbid removing any *file*s which are mount-points.

EXIT STATUS

1 if a *file* didn't exist without **-f**, was a **.** or **..** entry, or was preserved, or a file couldn't be deleted (incl. if it was a (non-empty) directory without **-r**/**-d**, or a mount-point with **-x**), or a directory couldn't be read or accessed if **-r**.

EXAMPLES

```
$ rm -vdI *.log
rm: 6 files: proceed? y
rm: build.2231.log: removed
rm: build.2232.log: removed
rm: build.2233.log: removed
rm: build.2234.log: removed
rm: ingest.log (-rw-r--r--, cicada/vore, 198.0k): unwritable, remove? n
    # An empty answer is equivalent here
rm: old.log: Directory not empty

# Clean out temporary directories without disrupting mounts underneath them
# rm -rfx /tmp /var/tmp
$ ls /var/tmp
schroot-0e7364148696.upper      schroot-41676a5d5fbe.upper      # &c.
```

SEE ALSO

rmdir(1), unlink(1), faccessat(2), isatty(3), rmdir(3), rpmatch(3), unlink(3)

STANDARDS

Conforms to IEEE Std 1003.1-2024 (“POSIX.1”); **-rRfi** are the only flags specified by the standard. **-I**, **--one-file-system**, **--interactive** (except **validate**), and **--**[no-]**preserve-root** are extensions, compatible from the GNU system. Short **-x** and **--interactive**=**validate** (but not **once**) are extensions.

HISTORY

Research UNIX

Appears in the first edition of the UNIX Programmer’s Manual as rm(I):

NAME rm -- remove (unlink) files
SYNOPSIS rm name$_1$ ...
BUGS rm probably should ask whether a read-only file is really to be removed.

And, indeed, Version 3 AT&T UNIX sees

rm [-f] [-r] name$_1$ ...

With default prompts regardless of standard input type and present-day **-f** — the prompts are simple, consisting of the name and mode, issued to the standard output stream, and affirmative responses are ones which begin with "**y**". **-r** recursess using chdir(II)/glob(VII), losing the full paths for diagnostics, and doesn’t remove the directories themselves. The flags can also appear at any point in the arguments.

Version 7 AT&T UNIX merges **rmdir** into the same page and evicts flags to the first argument, seeing:

rm [**-fri**] file ...

It ignores . and .., citing "antisocial consequences" of **rm -r** .*, with additional explicit handling of .. when passed as an argument (that being the only diagnostic to go to the standard error stream). **-r** recurses on the stack, with a filename of up to **99** bytes (note, however, that the maximum filename is **14** bytes long), and now removing the directories themselves (by forking out to **rmdir**). The new **-i** behaves as present-day, except still prompting on the standard output stream and not prompting for recursing into the arguments themselves, with **-f** being the default if the standard input stream is not a teletype.

The BSD

4.1BSD sees

rm [**-f**] [**-r**] [**-i**] [**-**] file ...

freeing the flags from the confines of the first argument and **-** ("the null option") terminating the argument list, much like **--** of today, and increasing the maximum path length to BUFSIZ (**1024**).

4.2BSD sees `rmdir`(2), meaning **rm** no longer forks, `access`(2), replacing the manual permission check, and `lstat`(2) to pair with the equally new `symlink`(2).

4.3BSD adds **−R** as an alias for **−r**, validates that paths are no longer than a global **malloc**()ed buffer of MAXNAMLEN + MAXPATHLEN (**255** + **1024**) bytes with **−r**, instead of razing its stack, and only allows no *file*s if **−f** was specified.

4.3BSD−Reno drops **−** in favour of the standard **−−** as part of compatibility with IEEE Std 1003.2-1992 (“POSIX.2”), and favours asking for permission overrides over the generic prompt in **−i** mode, if applicable.

4.4BSD adds **−d** and a `fts`(3)-based enumeration strategy that does away with depth limits.

4.4BSD−Lite2 adds **−W**, undeleting white-outed files, and **−P**, overwriting them consecutively with **0xFF**, **0x00**, and **0xFF** before removing (then immediately points out that this doesn’t work on journaled filesystems like its native LFS).

System V

AT&T System III UNIX directs Version 7 AT&T UNIX’s diagnostics (but not prompts) to the standard error stream and properly handles **−f**/**−i**/`isatty`(3).

AT&T System V Release 3 UNIX validates the path length to be no more than **1024** (it bases that number on DIRBUF ("buffer size for fs-indep. dirs" according to <`dirent.h`>), despite the maximum filename length being **512** according to that same header and official maximum path length (PATH_MAX) being **256**; the true maximum path length is **1024**).

AT&T System V Release 4 UNIX ignores a lone "**−**" after flags for compatibility with 4.1BSD, sees `rmdir`(2) and `lstat`(2) (despite symlinks appearing in AT&T System V Release 3 UNIX).

Standards

X/Open Portability Guide Issue 2 (“XPG2”) standardises the AT&T System III UNIX implementation, and X/Open Portability Guide Issue 3 (“XPG3”) adds localisation, to a singular effect:

> If that line begins with *y*, or the locale’s equivalent of a *y*,

IEEE Std 1003.2-1992 (“POSIX.2”) adds **−R** for consistency with other utilities.

IEEE Std 1003.1-2001 (“POSIX.1”) adds `/` preservation.

IEEE Std 1003.1-2024 (“POSIX.1”) adds **−dv**, as present-day.