

Next: [shred invocation](#), Previous: [mv invocation](#), Up: [Basic operations](#) [[Contents](#)][[Index](#)]

11.5 rm: Remove files or directories

rm removes each given *file*. By default, it does not remove directories. Synopsis:

```
rm [option]... [file]...
```

If the `-I` or `--interactive=once` option is given, and there are more than three files or the `-r`, `-R`, or `--recursive` are given, then rm prompts the user for whether to proceed with the entire operation. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the `-f` or `--force` option is not given, or the `-i` or `--interactive=always` option *is* given, rm prompts the user for whether to remove the file. If the response is not affirmative, the file is skipped.

Any attempt to remove a file whose last file name component is `.` or `..` is rejected without any prompting, as mandated by POSIX.

Warning: If you use rm to remove a file, it is usually possible to recover the contents of that file. If you want more assurance that the contents are unrecoverable, consider using shred.

The program accepts the following options. Also see [Common options](#).

`'-d'`

`'--dir'`

Remove the listed directories if they are empty.

`'-f'`

`'--force'`

Ignore nonexistent files and missing operands, and never prompt the user. Ignore any previous `--interactive (-i)` option.

`'-i'`

Prompt whether to remove each file. If the response is not affirmative, the file is skipped. Ignore any previous `--force (-f)` option. Equivalent to `--interactive=always`.

`'-I'`

Prompt once whether to proceed with the command, if more than three files are named or if a recursive removal is requested. Ignore any previous `--force (-f)` option. Equivalent to `--interactive=once`.

`'--interactive [=when]'`

Specify when to issue an interactive prompt. *when* may be omitted, or one of:

- `never` - Do not prompt at all.
- `once` - Prompt once if more than three files are named or if a recursive removal is requested. Equivalent to `-I`.
- `always` - Prompt for every file being removed. Equivalent to `-i`.

`--interactive` with no *when* is equivalent to `--interactive=always`.

`'--one-file-system'`

When removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument. This option is useful when removing a build “chroot” hierarchy, which normally contains no valuable data. However, it is not uncommon to bind-mount `/home` into such a hierarchy, to make it easier to use one’s start-up file. The catch is that it’s easy to forget to unmount `/home`. Then, when you use `rm -rf` to remove your normally throw-away chroot, that command will remove everything under `/home`, too. Use the `--one-file-system` option, and it will warn about and skip directories on other file systems. Of course, this will not save your `/home` if it and your chroot happen to be on the same file system. See also `--preserve-root=all` to protect command line arguments themselves.

`'--preserve-root [=all]'`

Fail upon any attempt to remove the root directory, `/`, when used with the `--recursive` option. This is the default behavior. See [Treating / specially](#). When ‘all’ is specified, reject any command line argument that is not on the same file system as its parent.

`'--no-preserve-root'`

Do not treat `/` specially when removing recursively. This option is not recommended unless you really want to remove all the files on your computer. See [Treating / specially](#).

`'-r'`

`'-R'`

`'--recursive'`

Remove the listed directories and their contents recursively.

`'-v'`

‘--verbose’

Print the name of each file before removing it.

One common question is how to remove files whose names begin with a ‘-’. GNU `rm`, like every program that uses the `getopt` function to parse its arguments, lets you use the ‘--’ option to indicate that all following arguments are non-options. To remove a file called `-f` in the current directory, you could type either:

```
rm -- -f
```

or:

```
rm ./-f
```

The Unix `rm` program’s use of a single ‘-’ for this purpose predates the development of the `getopt` standard syntax.

An exit status of zero indicates success, and a nonzero value indicates failure.

Next: [shred invocation](#), Previous: [mv invocation](#), Up: [Basic operations](#) [[Contents](#)][[Index](#)]