



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'rm.1p' command***

**\$ man rm.1p**

RM(1P)                    POSIX Programmer's Manual                    RM(1P)

### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

### NAME

rm ? remove directory entries

### SYNOPSIS

```
rm [-iRr] file...
rm -f [-iRr] [file...]
```

### DESCRIPTION

The rm utility shall remove the directory entry specified by each file argument.

If either of the files dot or dot-dot are specified as the basename portion of an operand (that is, the final pathname component) or if an operand resolves to the root directory, rm shall write a diagnostic message to standard error and do nothing more with such operands.

For each file the following steps shall be taken:

1. If the file does not exist:
  - a. If the -f option is not specified, rm shall write a diagnostic message to standard error.
  - b. Go on to any remaining files.

2. If file is of type directory, the following steps shall be taken:
  - a. If neither the `-R` option nor the `-r` option is specified, `rm` shall write a diagnostic message to standard error, do nothing more with file, and go on to any remaining files.
  - b. If file is an empty directory, `rm` may skip to step 2d. If the `-f` option is not specified, and either the permissions of file do not permit writing and the standard input is a terminal or the `-i` option is specified, `rm` shall write a prompt to standard error and read a line from the standard input. If the response is not affirmative, `rm` shall do nothing more with the current file and go on to any remaining files.
  - c. For each entry contained in file, other than dot or dot-dot, the four steps listed here (1 to 4) shall be taken with the entry as if it were a file operand. The `rm` utility shall not traverse directories by following symbolic links into other parts of the hierarchy, but shall remove the links themselves.
  - d. If the `-i` option is specified, `rm` shall write a prompt to standard error and read a line from the standard input. If the response is not affirmative, `rm` shall do nothing more with the current file, and go on to any remaining files.
3. If file is not of type directory, the `-f` option is not specified, and either the permissions of file do not permit writing and the standard input is a terminal or the `-i` option is specified, `rm` shall write a prompt to the standard error and read a line from the standard input. If the response is not affirmative, `rm` shall do nothing more with the current file and go on to any remaining files.
4. If the current file is a directory, `rm` shall perform actions equivalent to the `rmdir()` function defined in the System Interfaces volume of POSIX.1-2017 called with a pathname of the current file used as the `path` argument. If the current file is not a directory, `rm` shall perform actions equivalent to the `unlink()` function defined in the System Interfaces volume of POSIX.1-2017 called with a path?

name of the current file used as the path argument.

If this fails for any reason, `rm` shall write a diagnostic message to standard error, do nothing more with the current file, and go on to any remaining files.

The `rm` utility shall be able to descend to arbitrary depths in a file hierarchy, and shall not fail due to path length limitations (unless an operand specified by the user exceeds system limitations).

## OPTIONS

The `rm` utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

- f Do not prompt for confirmation. Do not write diagnostic messages or modify the exit status in the case of no file operands, or in the case of operands that do not exist. Any previous occurrences of the `-i` option shall be ignored.
- i Prompt for confirmation as described previously. Any previous occurrences of the `-f` option shall be ignored.
- R Remove file hierarchies. See the DESCRIPTION.
- r Equivalent to `-R`.

## OPERANDS

The following operand shall be supported:

`file` A pathname of a directory entry to be removed.

## STDIN

The standard input shall be used to read an input line in response to each prompt specified in the STDOUT section. Otherwise, the standard input shall not be used.

## INPUT FILES

None.

## ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of `rm`:

`LANG` Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Vari?

ables for the precedence of internationalization variables used to determine the values of locale categories.)

**LC\_ALL** If set to a non-empty string value, override the values of all the other internationalization variables.

#### **LC\_COLLATE**

Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements used in the extended regular expression defined for the `yesexpr` locale keyword in the `LC_MESSAGES` category.

**LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments) and the behavior of character classes within regular expressions used in the extended regular expression defined for the `yesexpr` locale keyword in the `LC_MESSAGES` category.

#### **LC\_MESSAGES**

Determine the locale used to process affirmative responses, and the locale used to affect the format and contents of diagnostic messages and prompts written to standard error.

**NLSPATH** Determine the location of message catalogs for the processing of `LC_MESSAGES`.

#### **ASYNCHRONOUS EVENTS**

Default.

#### **STDOUT**

Not used.

#### **STDERR**

Prompts shall be written to standard error under the conditions specified in the `DESCRIPTION` and `OPTIONS` sections. The prompts shall contain the file pathname, but their format is otherwise unspecified. The standard error also shall be used for diagnostic messages.

#### **OUTPUT FILES**

None.

#### **EXTENDED DESCRIPTION**

None.

## EXIT STATUS

The following exit values shall be returned:

- 0 Each directory entry was successfully removed, unless its removal was canceled by a non-affirmative response to a prompt for confirmation.
- >0 An error occurred.

## CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

## APPLICATION USAGE

The `rm` utility is forbidden to remove the names `dot` and `dot-dot` in order to avoid the consequences of inadvertently doing something like:

```
rm -r .*
```

Some implementations do not permit the removal of the last link to an executable binary file that is being executed; see the `[EBUSY]` error in the `unlink()` function defined in the System Interfaces volume of POSIX.1?2017. Thus, the `rm` utility can fail to remove such files.

The `-i` option causes `rm` to prompt and read the standard input even if the standard input is not a terminal, but in the absence of `-i` the mode prompting is not done when the standard input is not a terminal.

## EXAMPLES

1. The following command:

```
rm a.out core
```

removes the directory entries: `a.out` and `core`.

2. The following command:

```
rm -Rf junk
```

removes the directory `junk` and all its contents, without prompting.

## RATIONALE

For absolute clarity, paragraphs (2b) and (3) in the DESCRIPTION of `rm` describing the behavior when prompting for confirmation, should be interpreted in the following manner:

```
if ((NOT f_option) AND
```

((not\_writable AND input\_is\_terminal) OR i\_option))

The exact format of the interactive prompts is unspecified. Only the general nature of the contents of prompts are specified because implementations may desire more descriptive prompts than those used on historical implementations. Therefore, an application not using the `-f` option, or using the `-i` option, relies on the system to provide the most suitable dialog directly with the user, based on the behavior specified.

The `-r` option is historical practice on all known systems. The synonym `-R` option is provided for consistency with the other utilities in this volume of POSIX.1?2017 that provide options requesting recursive descent through the file hierarchy.

The behavior of the `-f` option in historical versions of `rm` is inconsistent. In general, along with "forcing" the unlink without prompting for permission, it always causes diagnostic messages to be suppressed and the exit status to be unmodified for nonexistent operands and files that cannot be unlinked. In some versions, however, the `-f` option suppresses usage messages and system errors as well. Suppressing such messages is not a service to either shell scripts or users.

It is less clear that error messages regarding files that cannot be unlinked (removed) should be suppressed. Although this is historical practice, this volume of POSIX.1?2017 does not permit the `-f` option to suppress such messages.

When given the `-r` and `-i` options, historical versions of `rm` prompt the user twice for each directory, once before removing its contents and once before actually attempting to delete the directory entry that names it. This allows the user to "prune" the file hierarchy walk.

Historical versions of `rm` were inconsistent in that some did not do the former prompt for directories named on the command line and others had obscure prompting behavior when the `-i` option was specified and the permissions of the file did not permit writing. The POSIX Shell and Utilities `rm` differs little from historic practice, but does require that prompts be consistent. Historical versions of `rm` were also inconsis-

sistent in that prompts were done to both standard output and standard error. This volume of POSIX.1-2017 requires that prompts be done to standard error, for consistency with cp and mv, and to allow historical extensions to rm that provide an option to list deleted files on standard output.

The rm utility is required to descend to arbitrary depths so that any file hierarchy may be deleted. This means, for example, that the rm utility cannot run out of file descriptors during its descent (that is, if the number of file descriptors is limited, rm cannot be implemented in the historical fashion where one file descriptor is used per directory level). Also, rm is not permitted to fail because of path length restrictions, unless an operand specified by the user is longer than {PATH\_MAX}.

The rm utility removes symbolic links themselves, not the files they refer to, as a consequence of the dependence on the unlink() functionality, per the DESCRIPTION. When removing hierarchies with -r or -R, the prohibition on following symbolic links has to be made explicit.

## FUTURE DIRECTIONS

None.

## SEE ALSO

rmdir

The Base Definitions volume of POSIX.1-2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

The System Interfaces volume of POSIX.1-2017, remove(), rmdir(), unlink()

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard

is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

IEEE/The Open Group

2017

RM(1P)