**NAME**

rm, rmdir – remove directory entries

**SYNOPSIS**

**/usr/bin/rm** [**−f**] [**−i**] *file*. . .
**/usr/bin/rm** **−rR** [**−f**] [**−i**] *dirname*. . . [*file*. . . ]

**/usr/xpg4/bin/rm** [ **−fiRr** ] *file* . . .

**/usr/bin/rmdir** [**−ps**] *dirname*. . .

**AVAILABILITY**

**/usr/bin/rm** SUNWcsu
**/usr/bin/rmdir**

**/usr/xpg4/bin/rm** SUNWxcu4

**DESCRIPTION**

**rm**

The **rm** command removes the directory entry specified by each *file* argument. If a *file* has no write permission and the standard input is a terminal, the full set of permissions (in octal) for the file are printed followed by a question mark. This is a prompt for confirmation. If the answer begins with **y** (for yes), the file is deleted, otherwise the file remains.

If *file* is a symbolic link, the link will be removed, but the file or directory to which it refers will not be deleted. Users do not need write permission to remove a symbolic link, provided they have write permissions in the directory.

If multiple *file*s are specified and removal of a *file* fails for any reason, **rm** will write a diagnostic message to standard error, do nothing more to the current *file*, and go on to any remaining *file*s.

If the standard input is not a terminal, the command will operate as if the **−f** option is in effect.

**rmdir**

The **rmdir** command will remove the directory entry specified by each *dirname* operand, which must refer to an empty directory.

Directories will be processed in the order specified. If a directory and a subdirectory of that directory are specified in a single invocation of **rmdir**, the subdirectory must be specified before the parent directory so that the parent directory will be empty when **rmdir** tries to remove it.

**OPTIONS**

The following options apply to **rm**:

**/usr/bin/rm**

**−f** Remove all files (whether write-protected or not) in a directory without prompting the user. In a write-protected directory, however, files are never removed (whatever their permissions are), but no messages are displayed. If the removal of a write-protected directory is attempted, this option will not suppress an error message.

**/usr/xpg4/bin/rm**

**–f** Do not prompt for confirmation. Do not write diagnostic messages or modify the exit status in the case of non-existent operands. Any previous occurences of the **–i** option will be ignored.

**/usr/xpg4/bin/rm**

**–i** Interactive. With this option, **rm** prompts for confirmation before removing any files. It overrides the **–f** option and remains in effect even if the standard input is not a terminal.

**/usr/xpg4/bin/rm**

**–i** Prompt for confirmation. Any occurences of the **–f** option will be ignored.

**–r** Recursively remove directories and subdirectories in the argument list. The directory will be emptied of files and removed. The user is normally prompted for removal of any write-protected files which the directory contains. The write-protected files are removed without prompting, however, if the **–f** option is used, or if the standard input is not a terminal and the **–i** option is not used.

Symbolic links that are encountered with this option will not be traversed.

If the removal of a non-empty, write-protected directory is attempted, the command will always fail (even if the **–f** option is used), resulting in an error message.

**–R** Same as **–r** option.

The following options apply to **rmdir**:

**–p** Allow users to remove the directory *dirname* and its parent directories which become empty. A message is printed on the standard error about whether the whole path is removed or part of the path remains for some reason.

**–s** Suppress the message printed on the standard error when **–p** is in effect.

## OPERANDS

The following operand is supported:

*file* A path name of a directory entry to be removed.

*dirname* A path name of an empty directory to be removed.

## EXAMPLES

### rm

The following command:

**example% rm a.out core**

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

The following command:

**example% rm -rf junk**

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

### rmdir

If a directory **a** in the current directory is empty except it contains a directory **b** and **a/b** is empty except it contains a directory **c**,

**example% rmdir -p a/b/c**

will remove all three directories.

## ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **rm** and **rmdir**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

## EXIT STATUS

The following exit values are returned:

**0** If the −**f** option was not specified, all the named directory entries were removed; otherwise, all the existing named directory entries were removed.

**>0** An error occurred.

## SEE ALSO

**rmdir**(2), **unlink**(2), **environ**(5)

## DIAGNOSTICS

All messages are generally self-explanatory.

It is forbidden to remove the files "**.**" and "**..**" in order to avoid the consequences of inadvertently doing something like the following:

**rm** −**r .***

## NOTES

A −− permits the user to mark explicitly the end of any command line options, allowing **rm** to recognize file arguments that begin with a −. As an aid to BSD migration, **rm** will accept − as a synonym for −−. This migration aid may disappear in a future release. If a −− and a − both appear on the same command line, the second will be interpreted as a file.