NAME

link, **ln** — make links

SYNOPSIS

ln [**-Ffhinsv**] *source_file* [*target_file*]
ln [**-Ffhinsv**] *source_file ... target_dir*
link *source_file target_file*

DESCRIPTION

The **ln** utility creates a new directory entry (linked file) which has the same modes as the original file. It is useful for maintaining multiple copies of a file in many places at once without using up storage for the "copies"; instead, a link "points" to the original copy. There are two types of links; hard links and symbolic links. How a link "points" to a file is one of the differences between a hard and symbolic link.

The options are as follows:

-F If the target file already exists and is a directory, then remove it so that the link may occur. The **-F** option should be used with either **-f** or **-i** options. If none is specified, **-f** is implied. The **-F** option is a no-op unless **-s** option is specified.

-h If the *target_file* or *target_dir* is a symbolic link, do not follow it. This is most useful with the **-f** option, to replace a symlink which may point to a directory.

-f If the target file already exists, then unlink it so that the link may occur. (The **-f** option overrides any previous **-i** options.)

-i Cause **ln** to write a prompt to standard error if the target file exists. If the response from the standard input begins with the character 'y' or 'Y', then unlink the target file so that the link may occur. Otherwise, do not attempt the link. (The **-i** option overrides any previous **-f** options.)

-n Same as **-h**, for compatibility with other **ln** implementations.

-s Create a symbolic link.

-v Cause **ln** to be verbose, showing files as they are processed.

By default, **ln** makes *hard* links. A hard link to a file is indistinguishable from the original directory entry; any changes to a file are effectively independent of the name used to reference the file. Hard links may not normally refer to directories and may not span file systems.

A symbolic link contains the name of the file to which it is linked. The referenced file is used when an open(2) operation is performed on the link. A stat(2) on a symbolic link will return the linked-to file; an lstat(2) must be done to obtain information about the link. The readlink(2) call may be used to read the contents of a symbolic link. Symbolic links may span file systems and may refer to directories.

Given one or two arguments, **ln** creates a link to an existing file *source_file*. If *target_file* is given, the link has that name; *target_file* may also be a directory in which to place the link; otherwise it is placed in the current directory. If only the directory is specified, the link will be made to the last component of *source_file*.

Given more than two arguments, **ln** makes links in *target_dir* to all the named source files. The links made will have the same name as the files being linked to.

When the utility is called as **link**, exactly two arguments must be supplied, neither of which may specify a directory. No options may be supplied in this simple mode of operation, which performs a link(2) operation using the two passed arguments.

COMPATIBILITY

The **-h**, **-i**, **-n** and **-v** options are non-standard and their use in scripts is not recommended. They are provided solely for compatibility with other **ln** implementations.

The **-F** option is FreeBSD extention and should not be used in portable scripts.

SEE ALSO

`link`(2), `lstat`(2), `readlink`(2), `stat`(2), `symlink`(2), `symlink`(7)

STANDARDS

The **`ln`** utility conforms to IEEE Std 1003.2-1992 (“POSIX.2”).

The simplified **`link`** command conforms to Version 2 of the Single UNIX Specification (“SUSv2”).

HISTORY

An **`ln`** command appeared in Version 1 AT&T UNIX.