NAME

ln, **link** - link files

SYNOPSIS

ln [**-L** | **-P** | **-s** [**-F**]] [**-f** | **-iw**] [**-hnv**] *source_file* [*target_file*]
ln [**-L** | **-P** | **-s** [**-F**]] [**-f** | **-iw**] [**-hnv**] *source_file ... target_dir*
link *source_file target_file*

DESCRIPTION

The **ln** utility creates a new directory entry (linked file) for the file name specified by *target_file*. The *target_file* will be created with the same file modes as the *source_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 neither **-f** nor **-i** is specified, **-f** is implied. The **-F** option is a no-op unless **-s** is specified.

-L When creating a hard link to a symbolic link, create a hard link to the target of the symbolic link. This is the default. This option cancels the **-P** option.

-P When creating a hard link to a symbolic link, create a hard link to the symbolic link itself. This option cancels the **-L** option.

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

-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.

-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.

-w Warn if the source of a symbolic link does not currently exist.

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. Directories may not be hardlinked, and hard links 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.

EXAMPLES

Create a symbolic link named */home/src* and point it to */usr/src*:

```
# ln -s /usr/src /home/src
```

Hard link */usr/local/bin/fooprog* to file */usr/local/bin/fooprog-1.0*:

```
# ln /usr/local/bin/fooprog-1.0 /usr/local/bin/fooprog
```

As an exercise, try the following commands:

```
# ls -i /bin/[
11553 /bin/[
# ls -i /bin/test
11553 /bin/test
```

Note that both files have the same inode; that is, */bin/[* is essentially an alias for the test(1) command. This hard link exists so test(1) may be invoked from shell scripts, for example, using the if [ ] construct.

In the next example, the second call to **ln** removes the original *foo* and creates a replacement pointing to *baz*:

```
# mkdir bar baz
# ln -s bar foo
# ln -shf baz foo
```

Without the **-h** option, this would instead leave *foo* pointing to *bar* and inside *foo* create a new symlink *baz* pointing to itself. This results from directory-walking.

An easy rule to remember is that the argument order for **ln** is the same as for cp(1): The first argument needs to exist, the second one is created.

COMPATIBILITY

The **-h**, **-i**, **-n**, **-v** and **-w** 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 a FreeBSD extension 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.