NAME

ln – make hard or symbolic links to files

SYNOPSIS

/usr/ucb/ln [**–fs**] *filename* [*linkname*]
/usr/ucb/ln [**–fs**] *pathname*. . . *directory*

AVAILABILITY

SUNWscpu

DESCRIPTION

/usr/ucb/ln creates an additional directory entry, called a link, to a file or directory. Any number of links can be assigned to a file. The number of links does not affect other file attributes such as size, protections, data, etc.

filename is the name of the original file or directory. *linkname* is the new name to associate with the file or filename. If *linkname* is omitted, the last component of *filename* is used as the name of the link.

If the last argument is the name of a directory, symbolic links are made in that directory for each *pathname* argument; **/usr/ucb/ln** uses the last component of each *pathname* as the name of each link in the named *directory*.

A hard link (the default) is a standard directory entry just like the one made when the file was created. Hard links can only be made to existing files. Hard links cannot be made across file systems (disk partitions, mounted file systems). To remove a file, all hard links to it must be removed, including the name by which it was first created; removing the last hard link releases the inode associated with the file.

A symbolic link, made with the **–s** option, is a special directory entry that points to another named file. Symbolic links can span file systems and point to directories. In fact, you can create a symbolic link that points to a file that is currently absent from the file system; removing the file that it points to does not affect or alter the symbolic link itself.

A symbolic link to a directory behaves differently than you might expect in certain cases. While an **ls**(1) on such a link displays the files in the pointed-to directory, an '**ls –l**' displays information about the link itself:

```
example% /usr/ucb/ln –s dir link
example% ls link
file1 file2 file3 file4
example% ls –l link
lrwxrwxrwx  1 user          7 Jan 11 23:27 link → dir
```

When you **cd**(1) to a directory through a symbolic link, you wind up in the pointed-to location within the file system. This means that the parent of the new working directory is not the parent of the symbolic link, but rather, the parent of the pointed-to directory. For instance, in the following case the final working directory is **/usr** and not **/home/user/linktest**.

```
example% pwd
/home/user/linktest
example% /usr/ucb/ln –s /var/tmp symlink
example% cd symlink
```

```
example% cd ..
example% pwd
/usr
```

C shell user's can avoid any resulting navigation problems by using the **pushd** and **popd** built-in commands instead of **cd**.

OPTIONS

–f Force a hard link to a directory. This option is only available to the super-user, and should be used with extreme caution.

–s Create a symbolic link or links.

EXAMPLES

The commands below illustrate the effects of the different forms of the **/usr/ucb/ln** command:

```
example% /usr/ucb/ln file link
example% ls –F file link
file   link
example% /usr/ucb/ln –s file symlink
example% ls –F file symlink
file   symlink@
example% ls –li file link symlink
 10606 -rw-r--r--  2 user      0 Jan 12 00:06 file
 10606 -rw-r--r--  2 user      0 Jan 12 00:06 link
 10607 lrwxrwxrwx  1 user      4 Jan 12 00:06 symlink → file
example% /usr/ucb/ln –s nonesuch devoid
example% ls –F devoid
devoid@
example% cat devoid
devoid: No such file or directory
example% /usr/ucb/ln –s /proto/bin/* /tmp/bin
example% ls –F /proto/bin /tmp/bin
/proto/bin:
x*     y*     z*

/tmp/bin:
x@     y@     z@
```

SEE ALSO

cp(1), **ls**(1), **mv**(1), **rm**(1), **link**(2), **readlink**(2), **stat**(2), **symlink**(2)

NOTES

When the last argument is a directory, simple basenames should not be used for *pathname* arguments. If a basename is used, the resulting symbolic link points to itself:

```
example% /usr/ucb/ln –s file /tmp
example% ls –l /tmp/file
lrwxrwxrwx  1 user      4 Jan 12 00:16 /tmp/file → file
example% cat /tmp/file
/tmp/file: Too many levels of symbolic links
```

To avoid this problem, use full pathnames, or prepend a reference to the **PWD** variable to files in the working directory:

```
example% rm /tmp/file
example% /usr/ucb/ln –s $PWD/file /tmp
lrwxrwxrwx  1 user 4       Jan 12 00:16 /tmp/file → /home/user/subdir/file
```