NAME

symlink − make a new name for a file

SYNOPSIS

#include <unistd.h>

int symlink(const char ****oldpath*, const char *****newpath***);**

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

symlink(): _BSD_SOURCE || _XOPEN_SOURCE >= 500 || _POSIX_C_SOURCE >= 200112L

DESCRIPTION

symlink() creates a symbolic link named *newpath* which contains the string *oldpath*.

Symbolic links are interpreted at run time as if the contents of the link had been substituted into the path being followed to find a file or directory.

Symbolic links may contain .. path components, which (if used at the start of the link) refer to the parent directories of that in which the link resides.

A symbolic link (also known as a soft link) may point to an existing file or to a nonexistent one; the latter case is known as a dangling link.

The permissions of a symbolic link are irrelevant; the ownership is ignored when following the link, but is checked when removal or renaming of the link is requested and the link is in a directory with the sticky bit (**S_ISVTX**) set.

If *newpath* exists it will *not* be overwritten.

RETURN VALUE

On success, zero is returned. On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EACCES
Write access to the directory containing *newpath* is denied, or one of the directories in the path prefix of *newpath* did not allow search permission. (See also **path_resolution**(7).)

EEXIST
newpath already exists.

EFAULT
oldpath or *newpath* points outside your accessible address space.

EIO An I/O error occurred.

ELOOP
Too many symbolic links were encountered in resolving *newpath*.

ENAMETOOLONG
oldpath or *newpath* was too long.

ENOENT
A directory component in *newpath* does not exist or is a dangling symbolic link, or *oldpath* is the empty string.

ENOMEM
Insufficient kernel memory was available.

ENOSPC
The device containing the file has no room for the new directory entry.

ENOTDIR
A component used as a directory in *newpath* is not, in fact, a directory.

EPERM
The file system containing *newpath* does not support the creation of symbolic links.

EROFS
newpath is on a read-only file system.

CONFORMING TO
SVr4, 4.3BSD, POSIX.1-2001.

NOTES
No checking of *oldpath* is done.

Deleting the name referred to by a symlink will actually delete the file (unless it also has other hard links). If this behavior is not desired, use **link**(2).

SEE ALSO
ln(1), **lchown**(2), **link**(2), **lstat**(2), **open**(2), **readlink**(2), **rename**(2), **symlinkat**(2), **unlink**(2), **path_resolution**(7), **symlink**(7)

COLOPHON
This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.