NAME

link – link to a file

SYNOPSIS

#include <unistd.h>

int link(const char **existing*, **const char** **new***);**

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

link( ) creates a new link (directory entry) for the existing file and increments its link count by one. *existing* points to a path name naming an existing file. *new* points to a path name naming the new directory entry to be created.

To create hard links, both files must be on the same file system. Both the old and the new link share equal access and rights to the underlying object. The super-user may make multiple links to a directory. Unless the caller is the super-user, the file named by *existing* must not be a directory.

Upon successful completion, **link( )** marks for update the **st_ctime** field of the file. Also, the **st_ctime** and **st_mtime** fields of the directory that contains the new entry are marked for update.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of −1 is returned and **errno** is set to indicate the error.

ERRORS

link( ) will fail and no link will be created if one or more of the following are true:

EACCES	A component of either path prefix denies search permission.
EACCES	The requested link requires writing in a directory with a mode that denies write permission.
EEXIST	The link named by *new* exists.
EFAULT	*existing* or *new* points to an illegal address.
EINTR	A signal was caught during the **link( )** function.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMLINK	The maximum number of links to a file would be exceeded.
EMULTIHOP	Components of *existing* or *new* require hopping to multiple remote machines and the file system type does not allow it.
ENAMETOOLONG	The length of the *existing* or *new* argument exceeds {**PATH_MAX**}, or the length of a *existing* or *new* component exceeds {**NAME_MAX**} while {**_POSIX_NO_TRUNC**} is in effect.
ENOENT	*existing* or *new* is a null path name.
ENOENT	A component of either path prefix does not exist.
ENOENT	The file named by *existing* does not exist.
ENOLINK	*existing* or *new* points to a remote machine and the link to that machine is no longer active.

ENOSPC	the directory that would contain the link cannot be extended.
ENOTDIR	A component of either path prefix is not a directory.
EPERM	The file named by *existing* is a directory and the effective user of the calling process is not super-user.
EROFS	The requested link requires writing in a directory on a read-only file system.
EXDEV	The link named by *new* and the file named by *existing* are on different logical devices (file systems).

SEE ALSO **symlink**(2), **unlink**(2)