

NAME

`linkat` – create a file link relative to directory file descriptors

SYNOPSIS

```
#define _ATFILE_SOURCE
#include <fcntl.h> /* Definition of AT_* constants */
#include <unistd.h>
```

```
int linkat(int olddirfd, const char *oldpath,
           int newdirfd, const char *newpath, int flags);
```

DESCRIPTION

The `linkat()` system call operates in exactly the same way as `link(2)`, except for the differences described in this manual page.

If the pathname given in *oldpath* is relative, then it is interpreted relative to the directory referred to by the file descriptor *olddirfd* (rather than relative to the current working directory of the calling process, as is done by `link(2)` for a relative pathname).

If *oldpath* is relative and *olddirfd* is the special value `AT_FDCWD`, then *oldpath* is interpreted relative to the current working directory of the calling process (like `link(2)`).

If *oldpath* is absolute, then *olddirfd* is ignored.

The interpretation of *newpath* is as for *oldpath*, except that a relative pathname is interpreted relative to the directory referred to by the file descriptor *newdirfd*.

By default, `linkat()`, does not dereference *oldpath* if it is a symbolic link (like `link(2)`). Since Linux 2.6.18, the flag `AT_SYMLINK_FOLLOW` can be specified in *flags* to cause *oldpath* to be dereferenced if it is a symbolic link. Before kernel 2.6.18, the *flags* argument was unused, and had to be specified as 0.

RETURN VALUE

On success, `linkat()` returns 0. On error, `-1` is returned and *errno* is set to indicate the error.

ERRORS

The same errors that occur for `link(2)` can also occur for `linkat()`. The following additional errors can occur for `linkat()`:

EBADF

olddirfd or *newdirfd* is not a valid file descriptor.

ENOTDIR

oldpath is relative and *olddirfd* is a file descriptor referring to a file other than a directory; or similar for *newpath* and *newdirfd*

VERSIONS

`linkat()` was added to Linux in kernel 2.6.16.

CONFORMING TO

POSIX.1-2008.

NOTES

See `openat(2)` for an explanation of the need for `linkat()`.

SEE ALSO

`link(2)`, `openat(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/>.