

NAME

fhlink, **fhlinkat** - make a hard file link

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <unistd.h>
```

int

```
fhlink(fhandle_t *fhp, const char *to);
```

int

```
fhlinkat(fhandle_t *fhp, int tofd, const char *to);
```

DESCRIPTION

The **fhlink()** system call atomically creates the specified directory entry (hard link) *to* with the attributes of the underlying object pointed at by *fhp*. If the link is successful: the link count of the underlying object is incremented; *fhp* and *to* share equal access and rights to the underlying object.

If *fhp* is removed, the file *to* is not deleted and the link count of the underlying object is decremented.

The object pointed at by the *fhp* argument must exist for the hard link to succeed and both *fhp* and *to* must be in the same file system. The *fhp* argument may not be a directory.

The **fhlinkat()** system call is equivalent to *fhlink* except in the case where *to* is a relative paths. In this case a relative path *to* is interpreted relative to the directory associated with the file descriptor *tofd* instead of the current working directory.

If **fhlinkat()** is passed the special value `AT_FDCWD` in the *tofd* parameter, the current working directory is used for the *to* argument. If *tofd* has value `AT_FDCWD`, the behavior is identical to a call to **link()**. Unless *flag* contains the `AT_SYMLINK_FOLLOW` flag, if *fhp* names a symbolic link, a new link is created for the symbolic link *fhp* and not its target.

RETURN VALUES

The **link()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **fhlink()** system call will fail and no link will be created if:

[ENOTDIR]	A component of <i>to</i> prefix is not a directory.
[ENAMETOOLONG]	A component of <i>to</i> exceeded 255 characters, or entire length of <i>to</i> name exceeded 1023 characters.
[ENOENT]	A component of <i>to</i> prefix does not exist.
[EOPNOTSUPP]	The file system containing the file pointed at by <i>fhp</i> does not support links.
[EMLINK]	The link count of the file pointed at by <i>fhp</i> would exceed 32767.
[EACCES]	A component of <i>to</i> prefix denies search permission.
[EACCES]	The requested link requires writing in a directory with a mode that denies write permission.
[ELOOP]	Too many symbolic links were encountered in translating one of the pathnames.
[ENOENT]	The file pointed at by <i>fhp</i> does not exist.
[EEXIST]	The link named by <i>to</i> does exist.
[EPERM]	The file pointed at by <i>fhp</i> is a directory.
[EPERM]	The file pointed at by <i>fhp</i> has its immutable or append-only flag set, see the <i>chflags(2)</i> manual page for more information.
[EPERM]	The parent directory of the file named by <i>to</i> has its immutable flag set.
[EXDEV]	The link named by <i>to</i> and the file pointed at by <i>fhp</i> are on different file systems.
[ENOSPC]	The directory in which the entry for the new link is being placed cannot be extended because there is no space left on the file system containing the directory.
[EDQUOT]	The directory in which the entry for the new link is being placed cannot be extended because the user's quota of disk blocks on the file system containing the directory has been exhausted.
[EIO]	An I/O error occurred while reading from or writing to the file system to make the

directory entry.

- [EINTEGRITY] Corrupted data was detected while reading from the file system.
- [EROFS] The requested link requires writing in a directory on a read-only file system.
- [EFAULT] One of the pathnames specified is outside the process's allocated address space.
- [ESTALE] The file handle *fhp* is no longer valid

In addition to the errors returned by the **fhlink()**, the **fhlinkat()** system call may fail if:

- [EBADF] The *fhp* or *to* argument does not specify an absolute path and the *tofd* argument, is not `AT_FDCWD` nor a valid file descriptor open for searching.
- [EINVAL] The value of the *flag* argument is not valid.
- [ENOTDIR] The *fhp* or *to* argument is not an absolute path and *tofd* is not `AT_FDCWD` nor a file descriptor associated with a directory.

SEE ALSO

`fhopen(2)`, `fthreadlink(2)`, `fhstat(2)`

HISTORY

The **fhlink()** and **fhlinkat()** system calls first appeared in FreeBSD 12.1.