NAME

flopen, **flopenat** - Reliably open and lock a file

LIBRARY

System Utilities Library (libutil, -lutil)

SYNOPSIS

#include <sys/fcntl.h>
#include <libutil.h>

int
flopen(*const char *path*, *int flags*);

int
flopen(*const char *path*, *int flags*, *mode_t mode*);

int
flopenat(*int fd*, *const char *path*, *int flags*);

int
flopenat(*int fd*, *const char *path*, *int flags*, *mode_t mode*);

DESCRIPTION

The **flopen**() function opens or creates a file and acquires an exclusive lock on it. It is essentially equivalent with calling **open**() with the same parameters followed by **flock**() with an *operation* argument of LOCK_EX, except that **flopen**() will attempt to detect and handle races that may occur between opening / creating the file and locking it. Thus, it is well suited for opening lock files, PID files, spool files, mailboxes and other kinds of files which are used for synchronization between processes.

If *flags* includes O_NONBLOCK and the file is already locked, **flopen**() will fail and set *errno* to EWOULDBLOCK.

As with **open**(), the additional *mode* argument is required if *flags* includes O_CREAT.

The **flopenat**() function is equivalent to the **flopen**() function except in the case where the *path* specifies a relative path. In this case the file to be opened is determined relative to the directory associated with the file descriptor *fd* instead of the current working directory. If **flopenat**() is passed the special value AT_FDCWD in the *fd* parameter, the current working directory is used and the behavior is identical to a call to **flopen**().

RETURN VALUES

If successful, **flopen**() returns a valid file descriptor. Otherwise, it returns -1, and sets *errno* as described in flock(2) and open(2).

SEE ALSO

errno(2), flock(2), open(2)

AUTHORS

The **flopen** function and this manual page were written by Dag-Erling Smørgrav <*des@FreeBSD.org*>.