NAME

fcntl – file control options

SYNOPSIS

#include <fcntl.h>

DESCRIPTION

The **<fcntl.h>** header defines the following requests and arguments for use by the functions **fcntl**(2) and **open**(2).

Values for *cmd* used by **fcntl** (the following values are unique):

F_DUPFD	Duplicate file descriptor
F_GETFD	Get file descriptor flags
F_SETFD	Set file descriptor flags
F_GETFL	Get file status flags
F_SETFL	Set file status flags
F_GETLK	Get record locking information
F_SETLK	Set record locking information
F_SETLKW	Set record locking information; wait if blocked

File descriptor flags used for **fcntl**:

FD_CLOEXEC	Close the file descriptor upon execution of an exec function (see **exec**(2))

Values for **l_type** used for record locking with **fcntl** (the following values are unique):

F_RDLCK	Shared or read lock
F_UNLCK	Unlock
F_WRLCK	Exclusive or write lock

The following three sets of values are bitwise distinct: Values for **oflag** used by **open**:

O_CREAT	Create file if it does not exist
O_EXCL	Exclusive use flag
O_NOCTTY	Do not assign controlling tty
O_TRUNC	Truncate flag

File status flags used for **open** and **fcntl**:

O_APPEND	Set append mode
O_NDELAY	Non-blocking mode
O_NONBLOCK	Non-blocking mode (POSIX)
O_DSYNC	Write I/O operations on the file descriptor complete as defined by synchronized I/O data integrity completion
O_RSYNC	Read I/O operations on the file descriptor complete at the same level of integrity as specified by the the **O_DSYNC** and **O_SYNC** flags. If both **O_DSYNC** and **O_RSYNC** are set in *oflag*, all I/O operations on the file descriptor complete as defined by synchronized I/O data integrity completion. If both **O_SYNC** and **O_RSYNC** are set in *oflag*, all I/O operations on the file descriptor complete as defined by synchronized I/O file

integrity completion.

O_SYNC When opening a regular file, this flag affects subsequent writes. If set, each **write**(2) will wait for both the file data and file status to be physically updated. Write I/O operations on the file descriptor complete as defined by synchronized I/O file integrity completion.

Mask for use with file access modes:

O_ACCMODE Mask for file access modes

File access modes used for **open** and **fcntl**:

O_RDONLY	Open for reading only
O_RDWR	Open for reading and writing
O_WRONLY	Open for writing only

The structure **flock** describes a file lock. It includes the following members:

```
short   l_type;     /* Type of lock */
short   l_whence;   /* Flag for starting offset */
off_t   l_start;    /* Relative offset in bytes */
off_t   l_len;      /* Size; if 0 then until EOF */
long    l_sysid;    /* Returned with F_GETLK */
pid_t   l_pid;      /* Returned with F_GETLK */
```

SEE ALSO

creat(2), **exec**(2), **fcntl**(2), **open**(2), **fsync**(3C), **fdatasync**(3R)

NOTES

Data is successfully transferred for a write operation to a regular file when the system ensures that all data written is readable on any subsequent open of the file (even one that follows a system or power failure) in the absense of a failure of the physical storage medium.

Data is successfully transferred for a read operation when an image of the data on the physical storage medium is available to the requesting process.

Synchronized I/O data integrity completion (see **fdatasync**(3R)):

For reads, the operation has been completed or diagnosed if unsuccessful. The read is complete only when an image of the data has been successfully transferred to the requesting process. If there were any pending write requests affecting the data to be read at the time that the synchronized read operation was requested, these write requests will be successfully transferred prior to reading the data.

For writes, the operation has been completed or diagnosed if unsuccessful. The write is complete only when the data specified in the write request is successfully transferred, and all file system information required to retrieve the data is successfully transferred.

File attributes that are not necessary for data retrieval (access time, modification time, status change time) need not be successfully transferred prior to returning to the calling process.

Synchronized I/O file integrity completion (see **fsync**(3C)):

Identical to a synchronized I/O data integrity completion with the addition that all file attributes relative tothe I/O operation (including access time, modification time, status change time) will be successfully transferred prior to returning to the calling process.