

NAME

lfs — Logical File System operations

SYNOPSIS

```
#include <lfsh.h>
lfmount (name, bname, flag)
lfumount (name)
lfopen (name, mode)
lfclose ( )
lfcreate (fd, lfn, nsectors)
lfdelete (fd, lfn, nsectors)
lfrread (fd, lfn, sector, buf, size)
lfrwrite (fd, lfn, sector, buf, size)
lfrswitch (fd, lfn, lfn2)
lfrsize (fd, lfn)
lfrffde(fd, lfn, buf)
lfrufde(fd, lfn, buf)
lfrstat ( )
lfrfistat ( )
lfrupdate( )

unsigned short lfn, lfn2;
char *name, *bname, *buf;
```

DESCRIPTION

The Logical File System is a pseudo device which provides a fast-access file system implemented through the UNIX raw I/O facility. It uses the *ioctl(2)*, *open(2)*, and *close(2)* system calls to interface to the user process level.

The file system provided by *lfs* consists of a set of contiguously allocated files, specified by their file number (*lfn*), which are manipulated by the Logical File System operations. The *lfn* is used as an index into a file definition table stored on disk whose entries record the physical location and size of each file. File definition entries are read into memory when referenced and remain there if frequently used. The 512 byte block (i.e. one disk sector) is the basic unit of size for operations with the Logical File System.

The calling parameter definitions are:

<i>name</i>	The name of the character-special file by which the LFS driver is accessed.
<i>bname</i>	The name of the block-special file for the LFS disk area desired.
<i>flag</i>	If non-zero, indicates that the file system is to be mounted read-only.
<i>mode</i>	The mode to be used when the Logical File System is opened (see <i>open(2)</i>); the user must have the requested access permissions with respect to the UNIX file <i>name</i> .
<i>lfn</i>	The file number of the file within the logical file system to be manipulated. Up to 65,535 files are permitted per file system.
<i>nsectors</i>	The number of 512 byte blocks.
<i>sector</i>	Position within a logical file in terms of 512 byte disk blocks.
<i>buf</i>	Buffer address for reading or writing data.
<i>size</i>	Buffer size in 512 byte blocks.

The *lfs* operations are:

lfmount Associates character device *name* used to access the LFS driver with the block device *bname* corresponding to the LFS disk area. This operation can be thought of as

- "mounting" *bname* on *name*. If *flag* is non-zero, *bname* is mounted read-only. If either name is already in the internal mount table, or if the LFS header for *bname* cannot be accessed or contains the wrong magic number, a -1 is returned.
- lfmount** Disassociates *name* with its current block device. If the device is not in the mount table, or is currently open, a -1 is returned.
- lfopen** Opens the Logical File System associated with *name*. Normally, a file descriptor is returned; however, if *name* does not exist or does not appear in the mount table, a -1 is returned.
- lfclose** Closes the passed file descriptor previously returned by *lfopen*. All modified file definition entries for the specified device are flushed to disk.
- lfcreate** Creates logical file *lfn* in the LFS specified by *fd*. *Nsectors* is the initial size in sectors of the file. If *nsectors* is 0, a file 0 sectors long is created (whereas *lfdelete* deletes the file given the same input). All storage is allocated to *lfn* and remains so (even if never written to) until *lfcreate* or *lfdelete* is called to change the file size. Legal *lfn*'s start with 1; if *lfn* is 0 in an *lfcreate* call, the number of a 'scratch' logical file is returned. By convention, scratch files are allocated starting from the end of the range of legal *lfn*'s and proceed downward. *Lfcreate* may be used to decrease or increase the size of an existing file. Because all *lfn*'s are contiguous, increasing the size of an *lfn* may force a copy of the entire file. Increasing or decreasing file sizes may also lead to fragmentation of the storage freelist. Normally, *Lfcreate* returns the *lfn* of the file created; a -1 returned indicates an error.
- lfdelete** Changes the size of logical file *lfn* in the LFS specified by *fd* to be *nsectors*. If *nsectors* is 0 the file is deleted. Normally, the *lfn* of the file deleted is returned; a -1 returned indicates an error.
- lfread** *Size* sectors are read from logical file *lfn* in the LFS specified by *fd* and placed into *buf*. *Sector* is the starting position within *lfn* for the read. Normally, the number of sectors read is returned; this can be less than *size* if the end of the file is encountered. A -1 returned indicates an error.
- lfwrite** *Size* sectors are written from *buf* to logical file *lfn* in the LFS specified by *fd*. *Sector* is the starting position within *lfn* for the write. Normally the number of blocks written is returned; this can be less than *size* if the end of the file is encountered. A -1 returned indicates an error, e.g., if *sector* is beyond the end of the file.
- lfswitch** Switches the physical storage allocated for logical files *lfn* and *lfn2* in the LFS specified by *fd* without copying. This enables files to be replaced on-line very quickly. A -1 returned indicates an error.
- lfsize** Returns the size in sectors of *lfn*. A -1 returned indicates an error.
- lffde** Retrieves the file definition entry for logical file *lfn* in the LFS specified by *fd*. This includes flag bits, the file starting location and size, and user information stored therein.
- lfufde** Stores the user portion of the file definition entry for logical file *lfn* in the LFS specified by *fd*. This information is arbitrary and may be changed by the user.
- lfstat** Prints several statistics about LFS operations. These counts are summed over all mounted LFS. The statistics printed are: the number of *lfs* calls, the number of raw read and write calls, the number of raw read and write blocks transferred, the number of file definition entries found in memory, the number of such entries retrieved from disk, the hit ratio for retrieving such entries, and the number of calls to the UNIX block I/O system for administering the raw disk area.
- lffstat** Zeroes all *lfs* statistics counts.

SEE ALSO

mkfs(1M), close(2), ioctl(2), open(2), lfs(5)

The Logical File System — A Fast-Access File System Using UNIX Raw I/O, TM 79-9471-1,
J. R. McSkimin.

AUTHOR

J. C. Kaufeld Jr.
J. R. McSkimin