NAME

VOP_LOCK, **VOP_UNLOCK**, **VOP_ISLOCKED**, **vn_lock** - serialize access to a vnode

SYNOPSIS

#include <sys/param.h>
#include <sys/lock.h>
#include <sys/vnode.h>

int
VOP_LOCK(*struct vnode *vp, int flags*);

int
VOP_UNLOCK(*struct vnode *vp*);

int
VOP_ISLOCKED(*struct vnode *vp*);

int
vn_lock(*struct vnode *vp, int flags*);

DESCRIPTION

These calls are used to serialize access to the file system, such as to prevent two writes to the same file from happening at the same time.

The arguments are:

vp The vnode being locked or unlocked.

flags One of the lock request types:

LK_SHARED	Shared lock.
LK_EXCLUSIVE	Exclusive lock.
LK_UPGRADE	Shared-to-exclusive upgrade.
LK_DOWNGRADE	Exclusive-to-shared downgrade.
LK_RELEASE	Release any type of lock.
LK_DRAIN	Wait for all lock activity to end.

The lock type may be *or*'ed with these lock flags:

LK_NOWAIT	Do not sleep to wait for lock.

LK_SLEEPFAIL	Sleep, then return failure.
LK_CANRECURSE	Allow recursive exclusive lock.
LK_NOWITNESS	Instruct witness(4) to ignore this instance.

The lock type may be *or*'ed with these control flags:

LK_INTERLOCK	Specify when the caller already has a simple lock (**VOP_LOCK**() will unlock the simple lock after getting the lock).
LK_RETRY	Retry until locked.

Kernel code should use **vn_lock**() to lock a vnode rather than calling **VOP_LOCK**() directly. **vn_lock**() also does not want a thread specified as argument but it assumes curthread to be used.

RETURN VALUES

Zero is returned on success, otherwise an error is returned.

SEE ALSO

vnode(9)

AUTHORS

This manual page was written by Doug Rabson.