NAME

close – relinquish access to a device

SYNOPSIS

Block and Character

#include <sys/types.h>
#include <sys/file.h>
#include <sys/errno.h>
#include <sys/open.h>
#include <sys/cred.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

int *prefix***close(dev_t** *dev*, **int** *flag*, **int** *otyp*, **cred_t** **cred_p***);**

STREAMS

#include <sys/types.h>
#include <sys/stream.h>
#include <sys/file.h>
#include <sys/errno.h>
#include <sys/open.h>
#include <sys/cred.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

int *prefix***close(queue_t** **q*, **int** *flag*, **cred_t** **cred_p***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI). This entry point is **required** for block devices.

ARGUMENTS

Block and Character

dev Device number.

flag File status flag, as set by the **open**(2) or modified by the **fcntl**(2) system calls. The flag is for information only—the file should always be closed completely. Possible values are: **FEXCL**, **FNDELAY**, **FREAD, FKLYR,** and **FWRITE**. Refer to **open**(9E) for more information.

otyp Parameter supplied so that the driver can determine how many times a device was opened and for what reasons. The flags assume the **open**( ) routine may be called many times, but the **close**( ) routine should only be called on the last **close** of a device.

OTYP_BLK	close was through block interface for the device
OTYP_CHR	close was through the raw/character interface for the device
OTYP_LYR	close a layered process (a higher-level driver called the **close( )** routine of the device)

**cred_p* Pointer to the user credential structure.

STREAMS

q* Pointer to **queue(9S) structure used to reference the read side of the driver. (A queue is the central node of a collection of structures and routines pointed to by a queue.)

flag File status flag.

**cred_p* Pointer to the user credential structure.

DESCRIPTION

For STREAMS drivers, the **close( )** routine is called by the kernel through the **cb_ops**(9S) table entry for the device. (Modules use the **fmodsw** table.) A non-null value in the **d_str** field of the **cb_ops** entry points to a **streamtab** structure, which points to a **qinit**(9S) containing a pointer to the **close( )** routine. Non-STREAMS **close( )** routines are called directly from the **cb_ops** table.

close( ) ends the connection between the user process and the device, and prepares the device (hardware and software) so that it is ready to be opened again.

A device may be opened simultaneously by multiple processes and the **open( )** driver routine is called for each open, but the kernel will only call the **close( )** routine when the last process using the device issues a **close**(2) or **umount**(2) system call or exits. (An exception is a close occurring with the *otyp* argument set to **OTYP_LYR**, for which a close (also having *otyp* = **OTYP_LYR**) occurs for each open.)

In general, a **close( )** routine should always check the validity of the minor number component of the *dev* parameter. The routine should also check permissions as necessary, by using the user credential structure (if pertinent), and the appropriateness of the *flag* and *otyp* parameter values.

close( ) could perform any of the following general functions:

- disable interrupts
- hang up phone lines
- rewind a tape
- deallocate buffers from a private buffering scheme
- unlock an unsharable device (that was locked in the **open( )** routine)
- flush buffers
- notify a device of the close
- deallocate any resources allocated on open

The **close( )** routines of STREAMS drivers and modules are called when a stream is dismantled or a module popped. The steps for dismantling a stream are performed in the following order. First, any multiplexor links present are unlinked and the lower streams are closed. Next, the following steps are performed for each module or driver on the stream, starting at the head and working toward the tail:

1. The write queue is given a chance to drain.
2. The **close( )** routine is called.
3. The module or driver is removed from the stream.

RETURN VALUES

close( ) should return 0 for success, or the appropriate error number. Return errors rarely occur, but if a failure is detected, the driver should decide whether the severity of the problem warrants either displaying a message on the console or, in worst cases, triggering a system panic. Generally, a failure in a **close( )** routine occurs because a problem occurred in the associated device.

SEE ALSO

close(2), **detach**(9E), **open**(9E)

Writing Device Drivers

STREAMS Programming Guide