

System Calls

close(2)

NAME

close - close a file descriptor

SYNOPSIS

```
#include <unistd.h>

int close(int fildes);
```

DESCRIPTION

The close() function will deallocate the file descriptor indicated by fildes. To deallocate means to make the file descriptor available for return by subsequent calls to open(2) or other functions that allocate file descriptors. All outstanding record locks owned by the process on the file associated with the file descriptor will be removed (that is, unlocked).

If close() is interrupted by a signal that is to be caught, it will return -1 with errno set to EINTR and the state of fildes is unspecified.

When all file descriptors associated with a pipe or FIFO special file are closed, any data remaining in the pipe or FIFO will be discarded.

When all file descriptors associated with an open file description have been closed the open file description will be freed.

If the link count of the file is 0, when all file descriptors associated with the file are closed, the space occupied by the file will be freed and the file will no longer be accessible.

If a STREAMS-based (see intro(2)) fildes is closed and the calling process was previously registered to receive a SIGPOLL signal (see signal(3C)) for events associated with that STREAM (see I_SETSIG in streamio(7I)), the calling process will be unregistered for events associated with the STREAM. The last close() for a STREAM causes the STREAM associated with fildes to be dismantled. If O_NONBLOCK and O_NDELAY are not set and there have been no signals posted for the STREAM, and if there is data on the module's write queue, close() waits up to 15 seconds (for each module and driver) for any output to drain before dismantling the STREAM. The time delay can be changed via an I_SETCLTIME ioctl(2) request (see streamio(7I)). If the O_NONBLOCK or O_NDELAY flag is set, or if there are any pending signals, close() does not wait for output to drain, and dismantles the STREAM immediately.

If fildes is associated with one end of a pipe, the last close() causes a hangup to occur on the other end of the pipe. In addition, if the other end of the pipe has been named by fattach(3C), then the last close() forces the named end to be detached by fdetach(3C). If the named end has no open file descriptors associated with it and gets detached, the STREAM associated with that end is also dismantled.

If fildes refers to the master side of a pseudo-terminal, a SIGHUP signal is sent to the process group, if any, for which the slave side of the pseudo-terminal is the controlling terminal. It is unspecified whether closing the master side of the pseudo-terminal flushes all queued input and output.

If fildes refers to the slave side of a STREAMS-based pseudo-terminal, a zero-length message may be sent to the master.

If fildes refers to a socket, close() causes the socket to be destroyed. If the socket is connection-mode, and the SOCK_LINGER option is set for the socket, and the socket has untransmitted data, then close() will block for up to the current linger interval until all data is transmitted.

RETURN VALUES

Upon successful completion, 0 is returned. Otherwise, -1 is returned and `errno` is set to indicate the error.

ERRORS

The `close()` function will fail if:

EBADF The `fd` argument is not a valid file descriptor.

EINTR The `close()` function was interrupted by a signal.

ENOLINK

The `fd` argument is on a remote machine and the link to that machine is no longer active.

ENOSPC

There was no free space remaining on the device containing the file.

The `close()` function may fail if:

EIO An I/O error occurred while reading from or writing to the file system.

USAGE

An application that used the `stdio` function `fopen(3C)` to open a file should use the corresponding `fclose(3C)` function rather than `close()`.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Async-Signal-Safe

SEE ALSO

`intro(2)`, `creat(2)`, `dup(2)`, `exec(2)`, `fcntl(2)`, `ioctl(2)`, `open(2)`, `pipe(2)`, `fattach(3C)`, `fclose(3C)`, `fdetach(3C)`, `fopen(3C)`, `signal(3C)`, `attributes(5)`, `signal(3HEAD)`, `streamio(7I)`

SunOS 5.9

Last change: 4 Apr 1997