NAME

sem_open, **sem_close**, **sem_unlink** - named semaphore operations

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <semaphore.h>

*sem_t **
sem_open(*const char *name*, *int oflag*, *...*);

int
sem_close(*sem_t *sem*);

int
sem_unlink(*const char *name*);

DESCRIPTION

The **sem_open**() function creates or opens the named semaphore specified by *name*. The returned semaphore may be used in subsequent calls to sem_getvalue(3), sem_wait(3), sem_trywait(3), sem_post(3), and **sem_close**().

This implementation places strict requirements on the value of *name*: it must begin with a slash (‘/’) and contain no other slash characters.

The following bits may be set in the *oflag* argument:

O_CREAT Create the semaphore if it does not already exist.

The third argument to the call to **sem_open**() must be of type *mode_t* and specifies the mode for the semaphore. Only the S_IWUSR, S_IWGRP, and S_IWOTH bits are examined; it is not possible to grant only "read" permission on a semaphore. The mode is modified according to the process’s file creation mask; see umask(2).

The fourth argument must be an *unsigned int* and specifies the initial value for the semaphore, and must be no greater than SEM_VALUE_MAX.

O_EXCL Create the semaphore if it does not exist. If the semaphore already exists, **sem_open**() will fail. This flag is ignored unless O_CREAT is also specified.

The **sem_close**() function closes a named semaphore that was opened by a call to **sem_open**().

The **sem_unlink**() function removes the semaphore named *name*. Resources allocated to the semaphore are only deallocated when all processes that have the semaphore open close it.

RETURN VALUES

If successful, the **sem_open**() function returns the address of the opened semaphore. If the same *name* argument is given to multiple calls to **sem_open**() by the same process without an intervening call to **sem_close**(), the same address is returned each time. If the semaphore cannot be opened, **sem_open**() returns SEM_FAILED and the global variable *errno* is set to indicate the error.

The **sem_close**() and **sem_unlink**() functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **sem_open**() function will fail if:

[EACCES] The semaphore exists and the permissions specified by *oflag* at the time it was created deny access to this process.

[EACCES] The semaphore does not exist, but permission to create it is denied.

[EEXIST] O_CREAT and O_EXCL are set but the semaphore already exists.

[EINTR] The call was interrupted by a signal.

[EINVAL] The **sem_open**() operation is not supported for the given *name*.

[EINVAL] The *value* argument is greater than SEM_VALUE_MAX.

[ENAMETOOLONG]
The *name* argument is too long.

[ENFILE] The system limit on semaphores has been reached.

[ENOENT] O_CREAT is not set but the named semaphore does not exist.

[ENOSPC] There is not enough space to create the semaphore.

The **sem_close**() function will fail if:

[EINVAL] The *sem* argument is not a valid semaphore.

The **sem_unlink**() function will fail if:

[EACCES] Permission is denied to unlink the semaphore.

[ENAMETOOLONG]
The specified *name* is too long.

[ENOENT] The named semaphore does not exist.

SEE ALSO

close(2), open(2), umask(2), unlink(2), sem_getvalue(3), sem_post(3), sem_trywait(3), sem_wait(3)

STANDARDS

The **sem_open**(), **sem_close**(), and **sem_unlink**() functions conform to ISO/IEC 9945-1:1996 ("POSIX.1").

HISTORY

Support for named semaphores first appeared in FreeBSD 5.0.