

**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 <i>oflag</i> 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 <b>sem_open()</b> operation is not supported for the given <i>name</i> .   |
| [EINVAL]       | The <i>value</i> argument is greater than SEM_VALUE_MAX.   |
| [ENAMETOOLONG] | The <i>name</i> 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.