NAME
       sem_open − initialize and open a named semaphore

SYNOPSIS
       #include <fcntl.h>   /* For O_* constants */
       #include <sys/stat.h>   /* For mode constants */
       #include <semaphore.h>

       sem_t *sem_open(const char **name**, int** *oflag***);**
       sem_t *sem_open(const char **name**, int** *oflag***,**
               mode_t *mode***, unsigned int** *value***);**

       Link with −*pthread*.

DESCRIPTION
       sem_open() creates a new POSIX semaphore or opens an existing semaphore. The semaphore is identified
       by *name*. For details of the construction of *name*, see **sem_overview**(7).

       The *oflag* argument specifies flags that control the operation of the call. (Definitions of the flags values can
       be obtained by including <*fcntl.h*>.) If **O_CREAT** is specified in *oflag*, then the semaphore is created if it
       does not already exist. The owner (user ID) of the semaphore is set to the effective user ID of the calling
       process. The group ownership (group ID) is set to the effective group ID of the calling process. If both
       O_CREAT and **O_EXCL** are specified in *oflag*, then an error is returned if a semaphore with the given
       name already exists.

       If **O_CREAT** is specified in *oflag*, then two additional arguments must be supplied. The *mode* argument
       specifies the permissions to be placed on the new semaphore, as for **open**(2). (Symbolic definitions for the
       permissions bits can be obtained by including <*sys/stat.h*>.) The permissions settings are masked against
       the process umask. Both read and write permission should be granted to each class of user that will access
       the semaphore. The *value* argument specifies the initial value for the new semaphore. If **O_CREAT** is
       specified, and a semaphore with the given *name* already exists, then *mode* and *value* are ignored.

RETURN VALUE
       On success, **sem_open**() returns the address of the new semaphore; this address is used when calling other
       semaphore-related functions. On error, **sem_open**() returns **SEM_FAILED**, with *errno* set to indicate the
       error.

ERRORS
       EACCES
              The semaphore exists, but the caller does not have permission to open it.

       EEXIST
              Both **O_CREAT** and **O_EXCL** were specified in *oflag*, but a semaphore with this *name* already
              exists.

       EINVAL
              value was greater than **SEM_VALUE_MAX**.

       EINVAL
              name consists of just "/", followed by no other characters.

       EMFILE
              The per-process limit on the number of open file descriptors has been reached.

       ENAMETOOLONG
              name was too long.

       ENFILE
              The system-wide limit on the total number of open files has been reached.

       ENOENT
              The **O_CREAT** flag was not specified in *oflag* and no semaphore with this *name* exists; or,
              O_CREAT was specified, but *name* wasn't well formed.

ENOMEM
　　　Insufficient memory.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes**(7).

Interface	Attribute	Value
sem_open()	Thread safety	MT-Safe

CONFORMING TO

POSIX.1-2001, POSIX.1-2008.

SEE ALSO

sem_close(3), **sem_getvalue**(3), **sem_post**(3), **sem_unlink**(3), **sem_wait**(3), **sem_overview**(7)

COLOPHON

This page is part of release 5.05 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.