NAME

sem_init − initialize an unnamed semaphore

LIBRARY

POSIX threads library (*libpthread*, *−lpthread*)

SYNOPSIS

#include <semaphore.h>

int sem_init(sem_t *****sem***, int** *pshared***, unsigned int** *value***);**

DESCRIPTION

sem_init() initializes the unnamed semaphore at the address pointed to by *sem*. The*value* argument specifies the initial value for the semaphore.

The *pshared* argument indicates whether this semaphore is to be shared between the threads of a process, or between processes.

If *pshared* has the value 0, then the semaphore is shared between the threads of a process, and should be located at some address that is visible to all threads (e.g., a global variable, or a variable allocated dynamically on the heap).

If *pshared* is nonzero, then the semaphore is shared between processes, and should be located in a region of shared memory (see **shm_open**(3), **mmap**(2), and **shmget**(2)). (Since a child created by**f ork**(2) inherits its parent's memory mappings, it can also access the semaphore.) Any process that can access the shared memory region can operate on the semaphore using **sem_post**(3), **sem_wait**(3), and so on.

Initializing a semaphore that has already been initialized results in undefined behavior.

RETURN VALUE

sem_init() returns 0 on success; on error, −1 is returned, and *errno* is set to indicate the error.

ERRORS

EINVAL

value exceeds **SEM_VALUE_MAX**.

ENOSYS

pshared is nonzero, but the system does not support process-shared semaphores (see **sem_overview**(7)).

ATTRIBUTES

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

Interface	Attribute	Value
sem_init()	Thread safety	MT-Safe

STANDARDS

POSIX.1-2001.

NOTES

Bizarrely, POSIX.1-2001 does not specify the value that should be returned by a successful call to **sem_init**(). POSIX.1-2008 rectifies this, specifying the zero return on success.

EXAMPLES

See **shm_open**(3) and **sem_wait**(3).

SEE ALSO

sem_destroy(3), **sem_post**(3), **sem_wait**(3), **sem_overview**(7)