Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'semget.3p' command

$ man semget.3p

SEMGET(3P) POSIX Programmer's Manual SEMGET(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

semget ? get set of XSI semaphores

SYNOPSIS

#include <sys/sem.h>

int semget(key_t key, int nsems, int semflg);

DESCRIPTION

The semget() function operates on XSI semaphores (see the Base Defini? tions volume of POSIX.1?2017, Section 4.17, Semaphore). It is unspeci? fied whether this function interoperates with the realtime interprocess communication facilities defined in Section 2.8, Realtime.

The semget() function shall return the semaphore identifier associated with key.

A semaphore identifier with its associated semid_ds data structure and its associated set of nsems semaphores (see <sys/sem.h>) is created for key if one of the following is true:

* The argument key is equal to IPC_PRIVATE.
* The argument key does not already have a semaphore identifier asso?

ciated with it and (semflg &IPC_CREAT) is non-zero.

Upon creation, the semid_ds data structure associated with the new sem? aphore identifier is initialized as follows:

* In the operation permissions structure sem_perm.cuid, sem_perm.uid, sem_perm.cgid, and sem_perm.gid shall be set to the effective user ID and effective group ID, respectively, of the calling process.
* The low-order 9 bits of sem_perm.mode shall be set to the low-order 9 bits of semflg.
* The variable sem_nsems shall be set to the value of nsems.
* The variable sem_otime shall be set to 0 and sem_ctime shall be set to the current time, as described in Section 2.7.1, IPC General De? scription.
* The data structure associated with each semaphore in the set need not be initialized. The semctl() function with the command SETVAL or SETALL can be used to initialize each semaphore.

RETURN VALUE

Upon successful completion, semget() shall return a non-negative inte? ger, namely a semaphore identifier; otherwise, it shall return -1 and set errno to indicate the error.

ERRORS

The semget() function shall fail if:

EACCES A semaphore identifier exists for key, but operation permission as specified by the low-order 9 bits of semflg would not be granted; see Section 2.7, XSI Interprocess Communication.

EEXIST A semaphore identifier exists for the argument key but ((semflg &IPC_CREAT) &&(semflg &IPC_EXCL)) is non-zero.

EINVAL The value of nsems is either less than or equal to 0 or greater than the system-imposed limit, or a semaphore identifier exists for the argument key, but the number of semaphores in the set associated with it is less than nsems and nsems is not equal to 0.

ENOENT A semaphore identifier does not exist for the argument key and (semflg &IPC_CREAT) is equal to 0.

ENOSPC A semaphore identifier is to be created but the system-imposed limit on the maximum number of allowed semaphores system-wide would be exceeded.

The following sections are informative.

EXAMPLES

Refer to semop().

APPLICATION USAGE

The POSIX Realtime Extension defines alternative interfaces for inter? process communication. Application developers who need to use IPC should design their applications so that modules using the IPC routines described in Section 2.7, XSI Interprocess Communication can be easily modified to use the alternative interfaces.

RATIONALE

None.

FUTURE DIRECTIONS

A future version may require that the value of the semval, sempid, sem? ncnt, and semzcnt members of all semaphores in a semaphore set be ini? tialized to zero when a call to semget() creates a semaphore set. Many semaphore implementations already do this and it greatly simplifies what an application must do to initialize a semaphore set.

SEE ALSO

Section 2.7, XSI Interprocess Communication, Section 2.8, Realtime, ftok(), semctl(), semop(), sem_close(), sem_destroy(), sem_getvalue(), sem_init(), sem_open(), sem_post(), sem_trywait(), sem_unlink()

The Base Definitions volume of POSIX.1?2017, Section 4.17, Semaphore, <sys_sem.h>

COPYRIGHT

The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .