NAME

semctl – semaphore control operations

SYNOPSIS

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

int semctl(int *semid*, **int** *semnum*, **int** *cmd*, **. . .);**

DESCRIPTION

semctl( ) provides a variety of semaphore control operations as specified by *cmd*. The fourth argument is optional, depending upon the operation requested. If required it is of type **union semun**, which must be explicitly declared by the application program.

```
union semun {
        int val;
        struct semid_ds *buf;
        ushort *array
} arg;
```

The permission required for a semaphore operation is given as {*token*}, where *token* is the type of permission needed. The types of permission are interpreted as follows:

00400	READ by user
00200	ALTER by user
00040	READ by group
00020	ALTER by group
00004	READ by others
00002	ALTER by others

See the **Semaphore Operation Permissions** subsection of the **DEFINITIONS** section of **intro**(2) for more information. The following semaphore operations as specified by *cmd* are executed with respect to the semaphore specified by *semid* and *semnum*.

GETVAL Return the value of semval (see **intro**(2)). {READ}

SETVAL Set the value of **semval** to *arg*.**val**. {ALTER}. When this command is successfully executed, the **semadj** value corresponding to the specified semaphore in all processes is cleared.

GETPID Return the value of **(int) sempid**. {READ}

GETNCNT Return the value of **semncnt**. {READ}

GETZCNT Return the value of **semzcnt**. {READ}

The following operations return and set, respectively, every **semval** in the set of semaphores.

GETALL Place **semval**s into array pointed to by *arg*.**array**. {READ}

SETALL Set **semval**s according to the array pointed to by *arg*.**array**. {ALTER}. When this cmd is successfully executed, the **semadj** values corresponding to each specified semaphore in all processes are cleared.

The following operations are also available.

IPC_STAT Place the current value of each member of the data structure associated with *semid* into the structure pointed to by *arg*.**buf**. The contents of this structure are defined in **intro**(2). {READ}

IPC_SET Set the value of the following members of the data structure associated with *semid* to the corresponding value found in the structure pointed to by *arg*.**buf**:

sem_perm.uid
sem_perm.gid
sem_perm.mode /* only access permission bits */

This command can be executed only by a process that has an effective user ID equal to either that of super-user, or to the value of **sem_perm.cuid** or **sem_perm.uid** in the data structure associated with *semid*.

IPC_RMID Remove the semaphore identifier specified by *semid* from the system and destroy the set of semaphores and data structure associated with it. This command can only be executed by a process that has an effective user ID equal to either that of super-user, or to the value of **sem_perm.cuid** or **sem_perm.uid** in the data structure associated with *semid*.

RETURN VALUES

Upon successful completion, the value returned depends on *cmd* as follows:

GETVAL the value of **semval**

GETPID the value of **(int) sempid**

GETNCNT the value of **semncnt**

GETZCNT the value of **semzcnt**

All other successful completions return **0**; otherwise, −**1** is returned and **errno** is set to indicate the error.

ERRORS

semctl( ) fails if one or more of the following are true:

EACCES Operation permission is denied to the calling process (see **intro**(2)).

EINVAL *semid* is not a valid semaphore identifier.

EINVAL *semnum* is less than **0** or greater than **sem_nsems** −**1**.

EINVAL *cmd* is not a valid command.

EINVAL *cmd* is **IPC_SET** and **sem_perm.uid** or **sem_perm.gid** is not valid.

EPERM *cmd* is equal to **IPC_RMID** or **IPC_SET** and the effective user of the calling process is not super-user, or to the value of **sem_perm.cuid** or **sem_perm.uid** in the data structure associated with *semid*.

EOVERFLOW *cmd* is **IPC_STAT** and *uid* or *gid* is too large to be stored in the structure pointed to by *arg.buf*.

ERANGE *cmd* is **SETVAL** or **SETALL** and the value to which **semval** is to be set is greater than the system imposed maximum.

SEE ALSO **ipcs**(1), **intro**(2), **semget**(2), **semop**(2)