NAME

shmctl – shared memory control operations

SYNOPSIS

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
```

int shmctl(int *shmid***, int** *cmd***, struct shmid_ds** **buf***);**

DESCRIPTION

shmctl( ) provides a variety of shared memory control operations as specified by *cmd*. The permission required for a shared memory control 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	WRITE by user
00040	READ by group
00020	WRITE by group
00004	READ by others
00002	WRITE by others

See the *Shared Memory Operation Permissions* section of **intro**(2) for more information.

The following operations require the specified tokens:

IPC_STAT Place the current value of each member of the data structure associated with *shmid* into the structure pointed to by *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 *shmid* to the corresponding value found in the structure pointed to by *buf*:

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

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

IPC_RMID Remove the shared memory identifier specified by *shmid* from the system and destroy the shared memory segment and data structure associated with it. This command can be executed only by a process that has an effective user ID equal to that of super-user, or to the value of **shm_perm.cuid** or **shm_perm.uid** in the data structure associated with *shmid*.

SHM_LOCK Lock the shared memory segment specified by *shmid* in memory. This command can be executed only by a process that has an effective user ID equal to super-user.

SHM_UNLOCK Unlock the shared memory segment specified by *shmid*. This command can be executed only by a process that has an effective user ID equal to super-user.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned and **errno** is set to indicate the error.

ERRORS

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

EACCES *cmd* is equal to **IPC_STAT** and {READ} operation permission is denied to the calling process.

EFAULT *buf* points to an illegal address.

EINVAL *shmid* is not a valid shared memory identifier.

EINVAL *cmd* is not a valid command.

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

ENOMEM *cmd* is equal to **SHM_LOCK** and there is not enough memory.

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

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 **shm_perm.cuid** or **shm_perm.uid** in the data structure associated with *shmid*.

EPERM *cmd* is equal to **SHM_LOCK** or **SHM_UNLOCK** and the effective user ID of the calling process is not equal to that of super-user.

SEE ALSO

ipcs(1), **intro**(2), **shmget**(2), **shmop**(2)

NOTES

The user must explicitly remove shared memory segments after the last reference to them has been removed.