NAME

shmop, shmat, shmdt – shared memory operations

SYNOPSIS

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

void ∗shmat(int *shmid*, **void** ∗*shmaddr*, **int** *shmflg***);**

int shmdt(void ∗*shmaddr***);**

DESCRIPTION

shmat( ) attaches the shared memory segment associated with the shared memory identifier specified by *shmid* to the data segment of the calling process.

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.

When (*shmflg***&SHM_SHARE_MMU**) is true, virtual memory resources in addition to shared memory itself are shared among processes that use the same shared memory.

The shared memory segment is attached to the data segment of the calling process at the address specified based on one of the following criteria:

- If *shmaddr* is equal to **(void ∗) 0**, the segment is attached to the first available address as selected by the system.
- If *shmaddr* is equal to **(void ∗) 0** and (*shmflg***&SHM_SHARE_MMU**) is true, then the segment is attached to the first available aligned address. See **NOTES** for the alignment requirement.
- If *shmaddr* is not equal to **(void ∗) 0** and (*shmflg***&SHM_RND**) is true, the segment is attached to the address given by (*shmaddr* **-** (*shmaddr* modulus **SHMLBA**)).
- If *shmaddr* is not equal to **(void ∗) 0** and (*shmflg***&SHM_RND**) is false, the segment is attached to the address given by *shmaddr*.

The segment is attached for reading if (*shmflg***&SHM_RDONLY**) is true {READ}, otherwise it is attached for reading and writing {READ/WRITE}.

When (*shmflg***&SHM_SHARE_MMU**) is set, however, the permission given by **shmget( )** determines whether the segment is attached for reading or reading and writing.

shmdt( ) detaches from the calling process's data segment the shared memory segment located at the address specified by *shmaddr*.

RETURN VALUES

Upon successful completion, the return value is as follows:

shmat( ) returns the data segment start address of the attached shared memory segment.

shmdt( ) returns a value of 0.

Otherwise, a value of −1 is returned and **errno** is set to indicate the error.

ERRORS

shmat( ) fails and does not attach the shared memory segment if one or more of the following are true:

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

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

EINVAL *shmaddr* is not equal to zero, and the value of (*shmaddr* **-** (*shmaddr* modulus **SHMLBA**)) is an illegal address.

EINVAL *shmaddr* is not equal to zero, (*shmflg*&**SHM_RND**) is false, and the value of *shmaddr* is an illegal address.

EINVAL shmaddr is not equal to zero, (*shmfg*&**SHM_SHARE_MMU**) is true, and *shmaddr* is not aligned properly.

EINVAL **shmdt( )** fails and does not detach the shared memory segment if *shmaddr* is not the data segment start address of a shared memory segment.

EINVAL **SHM_SHARE_MMU** is not supported in certain architectures.

EMFILE The number of shared memory segments attached to the calling process would exceed the system-imposed limit.

ENOMEM The available data space is not large enough to accommodate the shared memory segment.

SEE ALSO

intro(2), **exec**(2), **exit**(2), **fork**(2), **shmctl**(2), **shmget**(2)

NOTES

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

The alignment requirement, which varies on different machines, is determined by the mapping size of the virtual memory system.