Red Hat Enterprise Linux Release 9.2 Manual Pages on 'shm_overview.7' command

$ man shm_overview.7

SHM_OVERVIEW(7)   Linux Programmer's Manual   SHM_OVERVIEW(7)

NAME

shm_overview - overview of POSIX shared memory

DESCRIPTION

The POSIX shared memory API allows processes to communicate information

by sharing a region of memory.

The interfaces employed in the API are:

shm_open(3)   Create and open a new object, or open an existing ob?

ject. This is analogous to open(2). The call returns a

file descriptor for use by the other interfaces listed

below.

ftruncate(2)   Set the size of the shared memory object. (A newly cre?

ated shared memory object has a length of zero.)

mmap(2)   Map the shared memory object into the virtual address

space of the calling process.

munmap(2)   Unmap the shared memory object from the virtual address

space of the calling process.

shm_unlink(3) Remove a shared memory object name.

close(2)   Close the file descriptor allocated by shm_open(3) when

it is no longer needed.

fstat(2)   Obtain a stat structure that describes the shared memory

object. Among the information returned by this call are

the object's size (st_size), permissions (st_mode), owner (st_uid), and group (st_gid).

fchown(2) To change the ownership of a shared memory object.

fchmod(2) To change the permissions of a shared memory object.

Versions

POSIX shared memory is supported since Linux 2.4 and glibc 2.2.

Persistence

POSIX shared memory objects have kernel persistence: a shared memory object will exist until the system is shut down, or until all processes have unmapped the object and it has been deleted with shm_unlink(3)

Linking

Programs using the POSIX shared memory API must be compiled with cc -lrt to link against the real-time library, librt.

Accessing shared memory objects via the filesystem

On Linux, shared memory objects are created in a (tmpfs(5)) virtual filesystem, normally mounted under /dev/shm. Since kernel 2.6.19, Linux supports the use of access control lists (ACLs) to control the permissions of objects in the virtual filesystem.

NOTES

Typically, processes must synchronize their access to a shared memory object, using, for example, POSIX semaphores.

System V shared memory (shmget(2), shmop(2), etc.) is an older shared memory API. POSIX shared memory provides a simpler, and better de? signed interface; on the other hand POSIX shared memory is somewhat less widely available (especially on older systems) than System V shared memory.

SEE ALSO

fchmod(2), fchown(2), fstat(2), ftruncate(2), mmap(2), mprotect(2), munmap(2), shmget(2), shmop(2), shm_open(3), shm_unlink(3), sem_over? view(7)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at

https://www.kernel.org/doc/man-pages/.