



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

Rocky Enterprise Linux 9.2 Manual Pages on command 'svipc.7'

\$ man svipc.7

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

NAME

sysvipc - System V interprocess communication mechanisms

DESCRIPTION

System V IPC is the name given to three interprocess communication mechanisms that are widely available on UNIX systems: message queues, semaphore, and shared memory.

Message queues

System V message queues allow data to be exchanged in units called messages. Each messages can have an associated priority, POSIX message queues provide an alternative API for achieving the same result; see mq_overview(7).

The System V message queue API consists of the following system calls:

msgget(2)

Create a new message queue or obtain the ID of an existing message queue. This call returns an identifier that is used in the remaining APIs.

msgsnd(2)

Add a message to a queue.

msgrcv(2)

Remove a message from a queue.

msgctl(2)

Perform various control operations on a queue, including deletion.

Semaphore sets

System V semaphores allow processes to synchronize their actions System V semaphores are allocated in groups called sets; each semaphore in a set is a counting semaphore. POSIX

semaphores provide an alternative API for achieving the same result; see `sem_overview(7)`.

The System V semaphore API consists of the following system calls:

`semget(2)`

Create a new set or obtain the ID of an existing set. This call returns an identifier that is used in the remaining APIs.

`semop(2)`

Perform operations on the semaphores in a set.

`semctl(2)`

Perform various control operations on a set, including deletion.

Shared memory segments

System V shared memory allows processes to share a region of memory (a "segment"). POSIX shared memory is an alternative API for achieving the same result; see `shm_overview(7)`.

The System V shared memory API consists of the following system calls:

`shmget(2)`

Create a new segment or obtain the ID of an existing segment. This call returns an identifier that is used in the remaining APIs.

`shmat(2)`

Attach an existing shared memory object into the calling process's address space.

`shmdt(2)`

Detach a segment from the calling process's address space.

`shmctl(2)`

Perform various control operations on a segment, including deletion.

IPC namespaces

For a discussion of the interaction of System V IPC objects and IPC namespaces, see `ipc_namespaces(7)`.

SEE ALSO

`ipcrm(1)`, `ipcrm(1)`, `ipcs(1)`, `lsipc(1)`, `ipc(2)`, `msgctl(2)`, `msgget(2)`, `msgrcv(2)`, `msgsnd(2)`, `semctl(2)`, `semget(2)`, `semop(2)`, `shmat(2)`, `shmctl(2)`, `shmdt(2)`, `shmget(2)`, `ftok(3)`, `ipc_namespaces(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/>.

