## NAME

ipc − System V IPC system calls

## LIBRARY

Standard C library (*libc*, *−lc*)

## SYNOPSIS

```
#include <linux/ipc.h>      /* Definition of needed constants */
#include <sys/syscall.h>    /* Definition of SYS_* constants */
#include <unistd.h>

int syscall(SYS_ipc, unsigned int call, int first,
            unsigned long second, unsigned long third, void *ptr,
            long fifth);
```

*Note*: glibc provides no wrapper for **ipc**(), necessitating the use of **syscall**(2).

## DESCRIPTION

**ipc**() is a common kernel entry point for the System V IPC calls for messages, semaphores, and shared memory. *call* determines which IPC function to invoke; the other arguments are passed through to the appropriate call.

User-space programs should call the appropriate functions by their usual names. Only standard library implementors and kernel hackers need to know about **ipc**().

## STANDARDS

**ipc**() is Linux-specific, and should not be used in programs intended to be portable.

## NOTES

On some architectures—for example x86-64 and ARM—there is no **ipc**() system call; instead, **msgctl**(2), **semctl**(2), **shmctl**(2), and so on really are implemented as separate system calls.

## SEE ALSO

**msgctl**(2), **msgget**(2), **msgrcv**(2), **msgsnd**(2), **semctl**(2), **semget**(2), **semop**(2), **semtimedop**(2), **shmat**(2), **shmctl**(2), **shmdt**(2), **shmget**(2), **sysvipc**(7)