NAME

IPC::SysV - System V IPC constants and system calls

SYNOPSIS

```
use IPC::SysV qw(IPC_STAT IPC_PRIVATE);
```

DESCRIPTION

`IPC::SysV` defines and conditionally exports all the constants defined in your system include files which are needed by the SysV IPC calls. Common ones include

```
IPC_CREATE IPC_EXCL IPC_NOWAIT IPC_PRIVATE IPC_RMID IPC_SET IPC_STAT
GETVAL SETVAL GETPID GETNCNT GETZCNT GETALL SETALL
SEM_A SEM_R SEM_UNDO
SHM_RDONLY SHM_RND SHMLBA
```

and auxiliary ones

```
S_IRUSR S_IWUSR S_IRWXU
S_IRGRP S_IWGRP S_IRWXG
S_IROTH S_IWOTH S_IRWXO
```

but your system might have more.

ftok(PATH)

ftok(PATH, ID)

>Return a key based on PATH and ID, which can be used as a key for `msgget`, `semget` and `shmget`. See *ftok*.

>If ID is omitted, it defaults to `1`. If a single character is given for ID, the numeric value of that character is used.

shmat(ID, ADDR, FLAG)

>Attach the shared memory segment identified by ID to the address space of the calling process. See *shmat*.

>ADDR should be `undef` unless you really know what you're doing.

shmdt(ADDR)

>Detach the shared memory segment located at the address specified by ADDR from the address space of the calling process. See *shmdt*.

memread(ADDR, VAR, POS, SIZE)

>Reads SIZE bytes from a memory segment at ADDR starting at position POS. VAR must be a variable that will hold the data read. Returns true if successful, or false if there is an error. memread() taints the variable.

memwrite(ADDR, STRING, POS, SIZE)

>Writes SIZE bytes from STRING to a memory segment at ADDR starting at position POS. If STRING is too long, only SIZE bytes are used; if STRING is too short, nulls are written to fill out SIZE bytes. Returns true if successful, or false if there is an error.

SEE ALSO

IPC::Msg, *IPC::Semaphore*, *IPC::SharedMem*, *ftok*, *shmat*, *shmdt*

AUTHORS

Graham Barr <gbarr@pobox.com>, Jarkko Hietaniemi <jhi@iki.fi>, Marcus Holland-Moritz <mhx@cpan.org>

COPYRIGHT

Version 2.x, Copyright (C) 2007, Marcus Holland-Moritz.

Version 1.x, Copyright (c) 1997, Graham Barr.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.