ftok()

convert a path name and a project identifier to system v ipc

#include <sys/types.h>

#inlucde <sys/ipc.h>

key_t ftok(const char *pathname, int proj id);

use "pathname" for identity of the file name.

least significant 8 bits of proj_id (which must be nonzero) to generate a key_t type System V IPC key.

If same proj_id, resulting value is all the same.

return the generated key_t value. On fail -1 msgget

NAME

msgget - get a System V message queue identifier

SYNOPSIS

```
#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

int msgget(key_t key, int msgflg);
```

return queue identifier. -1 if fail

If a new message queue is created, then its associated data structure

msqid_ds (see msgctl(2)) is initialized as follows:

msg_perm.cuid and msg_perm.uid are set to the effective user ID

of the calling process

msg_perm.cgid and msg_perm.gid are set to the effective group ID

of the calling process.

The least significant 9 bits of msg_perm.mode are set to the

least significant 9 bits of msgflg.

msg_qnum, msg_lspid, msg_lrpid, msg_stime, and msg_rtime are set

to 0.

msg_ctime is set to the current time.

msg_qbytes is set to the system limit MSGMNB.

msgrcv, msgsnd

NAME

msgrcv, msgsnd - System V message queue operations

SYNOPSIS

```
#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);

ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp,

               int msgflg);
```

On failure both functions return -1 with errno indicating the error, otherwise msgsnd() returns 0 and msgrcv() returns the number of bytes

Flags:

IPC_NOWAIT

Return immediately if no message of the requested type is in the queue.

MSG_COPY

Nondestructively fetch a copy of the message at the ordinal position in the queue specified by msgtyp

MSG_EXCEPT

Used with msgtyp greater than 0 to read the first message in the queue with message type that differs from msgtyp.

MSG_NOERROR

To truncate the message text if longer than msgsz bytes.

msgctl

NAME

msgctl - System V message control operations

SYNOPSIS

```
#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

int msgctl(int msqid, int cmd, struct msqid_ds *buf);
```

cmd valid values

IPC_STAT

Copy information from the kernel data structure associated with

msqid into the msqid_ds structure pointed to by buf

IPC_SET

Write the values of some members of the msqid_ds structure

pointed to by buf to the kernel data structure associated with

this message queue, updating also its msg_ctime member.

IPC_RMID

Immediately remove the message queue, awakening all waiting

reader and writer processes

IPC_INFO

Return information about system-wide message queue limits and

parameters in the structure pointed to by buf.

MSG_INFO

Return a msginfo structure containing the same information as for

IPC_INFO, except that the following fields are returned with

information about system resources consumed by message queues

MSG_STAT

Return a msqid_ds structure as for IPC_STAT.

Shmget

NAME

shmget - allocates a System V shared memory segment

SYNOPSIS

```
#include <sys/ipc.h>

#include <sys/shm.h>

int shmget(key_t key, size_t size, int shmflg);
```

value of shmflg:

IPC_CREAT

Create a new segment.

IPC_EXCL

This flag is used with IPC_CREAT to ensure that this call creates the segment. If the segment already exists, the call fails.

SHM_HUGETLB

SHM_HUGE_2MB

Used in conjunction with SHM_HUGETLB to select alternative

hugetlb page sizes (respectively, 2 MB and 1 GB) on systems

that support multiple hugetlb page sizes.

SHM_NORESERVE

This flag serves the same purpose as the mmap(2) MAP_NORE-

SERVE flag.

Shmat, shmdt

NAME

shmat, shmdt - System V shared memory operations

SYNOPSIS

```
#include <sys/types.h>

#include <sys/shm.h>

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

int shmdt(const void *shmaddr);
```

shm-flg bit-mask argument:

SHM_EXEC (Linux-specific; since Linux 2.6.9)

Allow the contents of the segment to be executed. The caller must have execute permission on the segment.

SHM_RDONLY

Attach the segment for read-only access. The process must have read permission for the segment.

SHM_REMAP (Linux-specific)

This flag specifies that the mapping of the segment should replace any existing mapping in the range starting at shmaddr and continuing for the size of the segment.

shmctl

NAME

shmctl - System V shared memory control

SYNOPSIS

```
#include <sys/ipc.h>

#include <sys/shm.h>

int shmctl(int shmid, int cmd, struct shmid_ds *buf);
```

Valid values for cmd are:

IPC_STAT

Copy information from the kernel data structure associated with
shmid into the shmid_ds structure pointed to by buf

IPC_SET

Write the values of some members of the shmid_ds structure
pointed to by buf to the kernel data structure associated with
this shared memory segment, updating also its shm_ctime mem-
ber.

IPC_RMID

Mark the segment to be destroyed.

IPC_INFO

Return information about system-wide shared memory limits and
parameters in the structure pointed to by buf.

SHM_INFO

Return a shm_info structure whose fields contain information
about system resources consumed by shared memory.

SHM_STAT

Return a shmid_ds structure as for IPC_STAT.