The Single UNIX ® Specification, Version 2
Copyright © 1997 The Open Group

NAME

sys/ipc.h - interprocess communication access structure

SYNOPSIS

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

DESCRIPTION

The *<sys/ipc.h>* header is used by three mechanisms for interprocess communication (IPC): messages, semaphores and shared memory. All use a common structure type, **ipc_perm** to pass information used in determining permission to perform an IPC operation.

The structure **ipc_perm** contains the following members:

```
uid_t     uid     owner's user ID
gid_t     gid     owner's group ID
uid_t     cuid    creator's user ID
gid_t     cgid    creator's group ID
mode_t    mode    read/write permission
```

The **uid_t, gid_t, mode_t** and **key_t** types are defined as described in *<sys/types.h>*.
Definitions are given for the following constants:
Mode bits:
IPC_CREAT
Create entry if key does not exist.
IPC_EXCL
Fail if key exists.
IPC_NOWAIT
Error if request must wait.
Keys:
IPC_PRIVATE
Private key.
Control commands:
IPC_RMID
Remove identifier.
IPC_SET
Set options.
IPC_STAT
Get options.
The following is declared as a function and may also be defined as a macro.
Function prototypes must be provided for use with an ISO C compiler.

1

```
key_t   ftok(const char *, int);
```

APPLICATION USAGE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

ftok(), *<sys/types.h>*.

2

The Single UNIX ® Specification, Version 2
Copyright © 1997 The Open Group

NAME

sys/msg.h - message queue structures

SYNOPSIS

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

DESCRIPTION

The <**sys/msg.h**> header defines the following constant and members of the structure **msqid_ds**.

The following data types are defined through **typedef**:

msgqnum_t
Used for the number of messages in the message queue.

msglen_t
Used for the number of bytes allowed in a message queue.

These types are unsigned integer types that are able to store values at least as large as a type **unsigned short**.

Message operation flag:

MSG_NOERROR
No error if big message.

The structure **msqid_ds** contains the following members:

```
struct ipc_perm msg_perm    operation permission structure
msgqnum_t         msg_qnum    number of messages currently on queue
msglen_t          msg_qbytes  maximum number of bytes allowed on
queue
pid_t             msg_lspid   process ID of last msgsnd()
pid_t             msg_lrpid   process ID of last msgrcv()
time_t            msg_stime   time of last msgsnd()
time_t            msg_rtime   time of last msgrcv()
time_t            msg_ctime   time of last change
```

The **pid_t, time_t, key_t** and **size_t** types are defined as described in
<u>*<sys/types.h>*</u>.
The following are declared as functions and may also be defined as macros.
Function prototypes must be provided for use with an ISO C compiler.

```
int       msgctl(int, int, struct msqid_ds *);
int       msgget(key_t, int);
ssize_t   msgrcv(int, void *, size_t, long int, int);
int       msgsnd(int, const void *, size_t, int);
```

In addition, all of the symbols from <u>*<sys/ipc.h>*</u> will be defined when this header

3

is included.

APPLICATION USAGE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

msgctl(), *msgget()*, *msgrcv()*, *msgsnd()*, *<sys/types.h>*.

NAME

sys/sem.h - semaphore facility

SYNOPSIS

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

DESCRIPTION

The **<sys/sem.h>** header defines the following constants and structures.

Semaphore operation flags:
SEM_UNDO
Set up adjust on exit entry.
Command definitions for the function *semctl()*:
GETNCNT
Get **semncnt**.
GETPID
Get **sempid**.
GETVAL
Get **semval**.
GETALL
Get all cases of **semval**.
GETZCNT
Get **semzcnt**.
SETVAL
Set **semval**.
SETALL
Set all cases of **semval**.
The structure **semid_ds** contains the following members:

```
struct ipc_perm     sem_perm  operation permission structure
unsigned short int  sem_nsems number of semaphores in set
time_t              sem_otime last semop^) time
time_t              sem_ctime last time changed by semctl()
```

The **pid_t, time_t, key_t** and **size_t** types are defined as described in
<sys/types.h>.
A semaphore is represented by an anonymous structure containing the following members:

```
unsigned short int semval    semaphore value
pid_t              sempid    process ID of last operation
unsigned short int semncnt   number of processes waiting for
```

5

```
semval
                                    to become greater than current value
unsigned short int semzcnt    number of processes waiting for
semval
                                    to become 0
```

The structure **sembuf** contains the following members:

```
unsigned short int sem_num    semaphore number
short int            sem_op    semaphore operation
short int            sem_flg   operation flags
```

The following are declared as functions and may also be defined as macros.
Function prototypes must be provided for use with an ISO C compiler.

```
int    semctl(int, int, int, ...);
int    semget(key_t, int, int);
int    semop(int, struct sembuf *, size_t);
```

In addition, all of the symbols from _<sys/ipc.h>_ will be defined when this header
is included.

APPLICATION USAGE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

semctl(), _semget()_, _semop()_, _<sys/types.h>_.

6

The Single UNIX ® Specification, Version 2
Copyright © 1997 The Open Group

NAME

sys/shm.h - shared memory facility

SYNOPSIS

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

DESCRIPTION

The **<sys/shm.h>** header defines the following symbolic constants and structure:

Symbolic constants:

SHM_RDONLY

Attach read-only (else read-write).

SHMLBA

Segment low boundary address multiple.

SHM_RND

Round attach address to SHMLBA.

The following data types are defined through **typedef**:

shmatt_t

Unsigned integer used for the number of current attaches that must be able to store values at least as large as a type **unsigned short**.

The structure **shmid_ds** contains the following members:

```
struct ipc_perm shm_perm   operation permission structure
size_t          shm_segsz  size of segment in bytes
pid_t           shm_lpid   process ID of last shared memory operation
pid_t           shm_cpid   process ID of creator
shmatt_t        shm_nattch number of current attaches
time_t          shm_atime  time of last shmat()
time_t          shm_dtime  time of last shmdt()
time_t          shm_ctime  time of last change by shmctl()
```

7

The **pid_t**, **time_t**, **key_t** and **size_t** types are defined as described in *<sys/types.h>*. The following are declared as functions and may also be defined as macros. Function prototypes must be provided for use with an ISO C compiler.

```
void *shmat(int, const void *, int);
int   shmctl(int, int, struct shmid_ds *);
int   shmdt(const void *);
int   shmget(key_t, size_t, int);
```

In addition, all of the symbols from *<sys/ipc.h>* will be defined when this header is included.

APPLICATION USAGE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

shmat(), *shmctl()*, *shmdt()*, *shmget()*, *<sys/types.h>*.