NAME

msgctl - message control operations

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

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

DESCRIPTION

The **msgctl**() system call performs some control operations on the message queue specified by *msqid*.

Each message queue has a data structure associated with it, parts of which may be altered by **msgctl**() and parts of which determine the actions of **msgctl**(). The data structure is defined in <*sys/msg.h*> and contains (amongst others) the following members:

```
struct msqid_ds {
        struct      ipc_perm msg_perm;          /* msg queue permission bits */
        msglen_t msg_cbytes;        /* number of bytes in use on the queue */
        msgqnum_t msg_qnum;         /* number of msgs in the queue */
        msglen_t msg_qbytes;        /* max # of bytes on the queue */
        pid_t       msg_lspid;      /* pid of last msgsnd() */
        pid_t       msg_lrpid;      /* pid 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 msgctl() */
};
```

The *ipc_perm* structure used inside the *msqid_ds* structure is defined in <*sys/ipc.h*> and looks like this:

```
struct ipc_perm {
        uid_t               cuid;   /* creator user id */
        gid_t               cgid;   /* creator group id */
        uid_t               uid;    /* user id */
        gid_t               gid;    /* group id */
```

```
        mode_t          mode;   /* r/w permission */
        unsigned short  seq;    /* sequence # (to generate unique ipcid) */
        key_t           key;    /* user specified msg/sem/shm key */
};
```

The operation to be performed by **msgctl**() is specified in *cmd* and is one of:

IPC_STAT Gather information about the message queue and place it in the structure pointed to by *buf*.

IPC_SET Set the value of the *msg_perm.uid*, *msg_perm.gid*, *msg_perm.mode* and *msg_qbytes* fields in the structure associated with *msqid*. The values are taken from the corresponding fields in the structure pointed to by *buf*. This operation can only be executed by the super-user, or a process that has an effective user id equal to either *msg_perm.cuid* or *msg_perm.uid* in the data structure associated with the message queue. The value of *msg_qbytes* can only be increased by the super-user. Values for *msg_qbytes* that exceed the system limit (MSGMNB from <*sys/msg.h*>) are silently truncated to that limit.

IPC_RMID Remove the message queue specified by *msqid* and destroy the data associated with it. Only the super-user or a process with an effective uid equal to the *msg_perm.cuid* or *msg_perm.uid* values in the data structure associated with the queue can do this.

The permission to read from or write to a message queue (see msgsnd(2) and msgrcv(2)) is determined by the *msg_perm.mode* field in the same way as is done with files (see chmod(2)), but the effective uid can match either the *msg_perm.cuid* field or the *msg_perm.uid* field, and the effective gid can match either *msg_perm.cgid* or *msg_perm.gid*.

RETURN VALUES

The **msgctl**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **msgctl**() function will fail if:

[EPERM] The *cmd* argument is equal to IPC_SET or IPC_RMID and the caller is not the super-user, nor does the effective uid match either the *msg_perm.uid* or *msg_perm.cuid* fields of the data structure associated with the message queue.

An attempt is made to increase the value of *msg_qbytes* through IPC_SET but the

caller is not the super-user.

[EACCES] The command is IPC_STAT and the caller has no read permission for this message queue.

[EINVAL] The *msqid* argument is not a valid message queue identifier.

cmd is not a valid command.

[EFAULT] The *buf* argument specifies an invalid address.

SEE ALSO

msgget(2), msgrcv(2), msgsnd(2)

HISTORY

Message queues appeared in the first release of AT&T System V UNIX.