NAME

mq_send – send a message to a message queue

SYNOPSIS

cc [*flag* ...] *file* ... **–lposix4** [*library* ...]

#include <mqueue.h>

int mq_send(mqd_t *mqdes*, **const char** **msg_ptr*,
    size_t *msg_len*, **unsigned int** *msg_prio***);**

```
struct mq_attr {
        long    mq_flags;       /* message queue flags */
        long    mq_maxmsg;      /* maximum number of messages */
        long    mq_msgsize;     /* maximum message size */
        long    mq_curmsgs;     /* number of messages currently queued */
        ...
};
```

MT-LEVEL

MT-Safe

DESCRIPTION

mq_send( ) adds the message pointed to by *msg_ptr* to the message queue specified by *mqdes. msg_len* specifies the length of the message in bytes pointed to by *msg_ptr*. The value of *msg_len* must be less than or equal to the **mq_msgsize** attribute of the message queue, or **mq_send( )** will fail.

If the specified message queue is not full, **mq_send( )** behaves as if the message is inserted into the message queue at the position indicated by *msg_prio*. A message with a larger numeric value of *msg_prio* is inserted before messages with lower values of *msg_prio*. A message is inserted after other messages in the queue, if any, with equal *msg_prio*priority. The value of *msg_prio* must be greater than **0**, and less than or equal to **{MQ_PRIO_MAX}**.

If the specified message queue is full and if **O_NONBLOCK** is not set in the message queue description associated with *mqdes* (see **mq_open**(3R) and **mq_setattr**(3R)), **mq_send( )** blocks, waiting until space becomes available to enqueue the message, or until **mq_send( )** is interrupted by a signal. If more than one process (or thread) is waiting to send when space becomes available in the message queue, then the process of the highest priority which has been waiting the longest is unblocked to send its message. If the specified message queue is full and **O_NONBLOCK** is set in the message queue description associated with *mqdes*, the message is not queued, and **mq_send( )** returns an error.

RETURN VALUES

Upon successful completion, **mq_send( )** returns a value of **0**; otherwise, no message is enqueued, the function returns **-1**, and sets **errno** to indicate the error condition.

ERRORS

EAGAIN   The **O_NONBLOCK** flag is set in the message queue description associated with *mqdes*, and the specified message queue is full.

EBADF   *mqdes* is not a valid message queue descriptor open for writing.

EINTR   A signal interrupted the call to **mq_send( )**

EMSGSIZE The specified message length, *msg_len*, exceeds the message size attribute of the message queue.

ENOSYS **mq_send( )** is not supported by this implementation.

SEE ALSO

mq_open(3R), **mq_receive**(3R), **mq_setattr**(3R), **sysconf**(3C)

BUGS

In Solaris 2.5, these functions always return –**1** and set **errno** to **ENOSYS,** because this release does not support the Message Passing option. It is our intention to provide support for these interfaces in future releases.