NAME

mq_receive, **mq_timedreceive** - receive a message from message queue (REALTIME)

LIBRARY

POSIX Real-time Library (librt, -lrt)

SYNOPSIS

#include <mqueue.h>

ssize_t
mq_receive(*mqd_t mqdes, char *msg_ptr, size_t msg_len, unsigned *msg_prio*);

ssize_t
mq_timedreceive(*mqd_t mqdes, char *msg_ptr, size_t msg_len, unsigned *msg_prio,*
*const struct timespec *abs_timeout*);

DESCRIPTION

The **mq_receive**() system call receives oldest of the highest priority message(s) from the message queue specified by *mqdes*. If the size of the buffer in bytes, specified by the *msg_len* argument, is less than the *mq_msgsize* attribute of the message queue, the system call will fail and return an error. Otherwise, the selected message will be removed from the queue and copied to the buffer pointed to by the *msg_ptr* argument.

If the argument *msg_prio* is not NULL, the priority of the selected message will be stored in the location referenced by *msg_prio*. If the specified message queue is empty and O_NONBLOCK is not set in the message queue description associated with *mqdes*, **mq_receive**() will block until a message is enqueued on the message queue or until **mq_receive**() is interrupted by a signal. If more than one thread is waiting to receive a message when a message arrives at an empty queue and the Priority Scheduling option is supported, then the thread of highest priority that has been waiting the longest will be selected to receive the message. Otherwise, it is unspecified which waiting thread receives the message. If the specified message queue is empty and O_NONBLOCK is set in the message queue description associated with *mqdes*, no message will be removed from the queue, and **mq_receive**() will return an error.

The **mq_timedreceive**() system call will receive the oldest of the highest priority messages from the message queue specified by *mqdes* as described for the **mq_receive**() system call. However, if O_NONBLOCK was not specified when the message queue was opened via the **mq_open**() system call, and no message exists on the queue to satisfy the receive, the wait for such a message will be terminated when the specified timeout expires. If O_NONBLOCK is set, this system call is equivalent to **mq_receive**().

The timeout expires when the absolute time specified by *abs_timeout* passes, as measured by the clock on which timeouts are based (that is, when the value of that clock equals or exceeds *abs_timeout*), or if the absolute time specified by *abs_timeout* has already been passed at the time of the call.

The timeout is based on the CLOCK_REALTIME clock.

RETURN VALUES

Upon successful completion, the **mq_receive**() and **mq_timedreceive**() system calls return the length of the selected message in bytes and the message is removed from the queue. Otherwise, no message is removed from the queue, the system call returns a value of -1, and the global variable *errno* is set to indicate the error.

ERRORS

The **mq_receive**() and **mq_timedreceive**() system calls will fail if:

[EAGAIN]	O_NONBLOCK flag is set in the message queue description associated with *mqdes*, and the specified message queue is empty.
[EBADF]	The *mqdes* argument is not a valid message queue descriptor open for reading.
[EMSGSIZE]	The specified message buffer size, *msg_len*, is less than the message size attribute of the message queue.
[EINTR]	The **mq_receive**() or **mq_timedreceive**() operation was interrupted by a signal.
[EINVAL]	The process or thread would have blocked, and the *abs_timeout* parameter specified a nanoseconds field value less than zero or greater than or equal to 1000 million.
[ETIMEDOUT]	The O_NONBLOCK flag was not set when the message queue was opened, but no message arrived on the queue before the specified timeout expired.

SEE ALSO

mq_open(2), mq_send(2), mq_timedsend(2)

STANDARDS

The **mq_receive**() and **mq_timedreceive**() system calls conform to IEEE Std 1003.1-2004 ("POSIX.1").

HISTORY

Support for POSIX message queues first appeared in FreeBSD 7.0.

COPYRIGHT