

NAME

mq_receive, mq_timedreceive – receive a message from a message queue

LIBRARY

Real-time library (*librt*, *-lrt*)

SYNOPSIS

```
#include <mqqueue.h>
```

```
ssize_t mq_receive(mqd_t mqdes, char msg_ptr[msg_len],
                  size_t msg_len, unsigned int *msg_prio);
```

```
#include <time.h>
```

```
#include <mqqueue.h>
```

```
ssize_t mq_timedreceive(mqd_t mqdes, char *restrict msg_ptr[msg_len],
                       size_t msg_len, unsigned int *restrict msg_prio,
                       const struct timespec *restrict abs_timeout);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros(7)**):

```
mq_timedreceive():
    _POSIX_C_SOURCE >= 200112L
```

DESCRIPTION

mq_receive() removes the oldest message with the highest priority from the message queue referred to by the message queue descriptor *mqdes*, and places it in the buffer pointed to by *msg_ptr*. The *msg_len* argument specifies the size of the buffer pointed to by *msg_ptr*; this must be greater than or equal to the *mq_msgsize* attribute of the queue (see **mq_getattr(3)**). If *msg_prio* is not NULL, then the buffer to which it points is used to return the priority associated with the received message.

If the queue is empty, then, by default, **mq_receive()** blocks until a message becomes available, or the call is interrupted by a signal handler. If the **O_NONBLOCK** flag is enabled for the message queue description, then the call instead fails immediately with the error **EAGAIN**.

mq_timedreceive() behaves just like **mq_receive()**, except that if the queue is empty and the **O_NONBLOCK** flag is not enabled for the message queue description, then *abs_timeout* points to a structure which specifies how long the call will block. This value is an absolute timeout in seconds and nanoseconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC), specified in a **timespec(3)** structure.

If no message is available, and the timeout has already expired by the time of the call, **mq_timedreceive()** returns immediately.

RETURN VALUE

On success, **mq_receive()** and **mq_timedreceive()** return the number of bytes in the received message; on error, -1 is returned, with *errno* set to indicate the error.

ERRORS**EAGAIN**

The queue was empty, and the **O_NONBLOCK** flag was set for the message queue description referred to by *mqdes*.

EBADF

The descriptor specified in *mqdes* was invalid or not opened for reading.

EINTR

The call was interrupted by a signal handler; see **signal(7)**.

EINVAL

The call would have blocked, and *abs_timeout* was invalid, either because *tv_sec* was less than zero, or because *tv_nsec* was less than zero or greater than 1000 million.

EMSGSIZE

msg_len was less than the *mq_msgsize* attribute of the message queue.

ETIMEDOUT

The call timed out before a message could be transferred.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
mq_receive() , mq_timedreceive()	Thread safety	MT-Safe

STANDARDS

POSIX.1-2001, POSIX.1-2008.

NOTES

On Linux, **mq_timedreceive()** is a system call, and **mq_receive()** is a library function layered on top of that system call.

SEE ALSO

mq_close(3), **mq_getattr(3)**, **mq_notify(3)**, **mq_open(3)**, **mq_send(3)**, **mq_unlink(3)**, **timespec(3)**, **mq_overview(7)**, **time(7)**