



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'mq_receive.3p' command

\$ man mq_receive.3p

MQ_RECEIVE(3P) POSIX Programmer's Manual MQ_RECEIVE(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

mq_receive, mq_timedreceive ? receive a message from a message queue
(REALTIME)

SYNOPSIS

```
#include <mqueue.h>

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

#include <mqueue.h>
#include <time.h>

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

DESCRIPTION

The mq_receive() function shall receive the 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 function shall

fail and return an error. Otherwise, the selected message shall be removed from the queue and copied to the buffer pointed to by the `msg_ptr` argument.

If the value of `msg_len` is greater than `{SSIZE_MAX}`, the result is implementation-defined.

If the argument `msg_prio` is not `NULL`, the priority of the selected message shall 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()` shall 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 shall 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 shall be removed from the queue, and `mq_receive()` shall return an error.

The `mq_timedreceive()` function shall receive the oldest of the highest priority messages from the message queue specified by `mqdes` as described for the `mq_receive()` function. However, if `O_NONBLOCK` was not specified when the message queue was opened via the `mq_open()` function, and no message exists on the queue to satisfy the receive, the wait for such a message shall be terminated when the specified timeout expires.

If `O_NONBLOCK` is set, this function is equivalent to `mq_receive()`.

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

The timeout shall be based on the `CLOCK_REALTIME` clock. The resolution of the timeout shall be the resolution of the clock on which it is based. The `timespec` argument is defined in the `<time.h>` header.

Under no circumstance shall the operation fail with a timeout if a message can be removed from the message queue immediately. The validity of the `abstime` parameter need not be checked if a message can be removed from the message queue immediately.

RETURN VALUE

Upon successful completion, the `mq_receive()` and `mq_timedreceive()` functions shall return the length of the selected message in bytes and the message shall be removed from the queue. Otherwise, no message shall be removed from the queue, the functions shall return a value of `-1`, and set `errno` to indicate the error.

ERRORS

These functions shall fail if:

EAGAIN `O_NONBLOCK` was set in the message 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 `abstime` 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.

These functions may fail if:

EBADMSG

The implementation has detected a data corruption problem with the message.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`mq_open()`, `mq_send()`, `msgctl()`, `msgget()`, `msgrcv()`, `msgsnd()`, `time()`

The Base Definitions volume of POSIX.1-2017, `<mqueue.h>`, `<time.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.

IEEE/The Open Group

2017

MQ_RECEIVE(3P)