



Full credit is given to the above companies including the OS that this PDF file was generated!

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

\$ man msgrcv.3p

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

msgrcv ? XSI message receive operation

SYNOPSIS

```
#include <sys/msg.h>

ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp,
               int msgflg);
```

DESCRIPTION

The msgrcv() function operates on XSI message queues (see the Base Definitions volume of POSIX.1?2017, Section 3.226, Message Queue). It is unspecified whether this function interoperates with the realtime interprocess communication facilities defined in Section 2.8, Realtime.

The msgrcv() function shall read a message from the queue associated with the message queue identifier specified by msqid and place it in the user-defined buffer pointed to by msgp.

The application shall ensure that the argument msgp points to a user-defined buffer that contains first a field of type long specifying the type of the message, and then a data portion that holds the data bytes

of the message. The structure below is an example of what this user-defined buffer might look like:

```
struct mymsg {
    long  mtype; /* Message type. */
    char  mtext[1]; /* Message text. */
}
```

The structure member `mtype` is the received message's type as specified by the sending process.

The structure member `mtext` is the text of the message.

The argument `msgsz` specifies the size in bytes of `mtext`. The received message shall be truncated to `msgsz` bytes if it is larger than `msgsz` and `(msgflg & MSG_NOERROR)` is non-zero. The truncated part of the message shall be lost and no indication of the truncation shall be given to the calling process.

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

The argument `msgtyp` specifies the type of message requested as follows:

- * If `msgtyp` is 0, the first message on the queue shall be received.
- * If `msgtyp` is greater than 0, the first message of type `msgtyp` shall be received.
- * If `msgtyp` is less than 0, the first message of the lowest type that is less than or equal to the absolute value of `msgtyp` shall be received.

The argument `msgflg` specifies the action to be taken if a message of the desired type is not on the queue. These are as follows:

- * If `(msgflg & IPC_NOWAIT)` is non-zero, the calling thread shall return immediately with a return value of -1 and `errno` set to `[ENOMSG]`.
- * If `(msgflg & IPC_NOWAIT)` is 0, the calling thread shall suspend execution until one of the following occurs:
 - A message of the desired type is placed on the queue.
 - The message queue identifier `msqid` is removed from the system; when this occurs, `errno` shall be set to `[EIDRM]` and -1 shall be

returned.

- The calling thread receives a signal that is to be caught; in this case a message is not received and the calling thread resumes execution in the manner prescribed in `sigaction()`.

Upon successful completion, the following actions are taken with respect to the data structure associated with `msqid`:

- * `msg_qnum` shall be decremented by 1.
- * `msg_lpid` shall be set to the process ID of the calling process.
- * `msg_rtime` shall be set to the current time, as described in Section 2.7.1, IPC General Description.

RETURN VALUE

Upon successful completion, `msgrcv()` shall return a value equal to the number of bytes actually placed into the buffer `mtext`. Otherwise, no message shall be received, `msgrcv()` shall return -1, and `errno` shall be set to indicate the error.

ERRORS

The `msgrcv()` function shall fail if:

E2BIG The value of `mtext` is greater than `msgsz` and `(msgflg & MSG_NOERROR)` is 0.

EACCES Operation permission is denied to the calling process; see Section 2.7, XSI Interprocess Communication.

EIDRM The message queue identifier `msqid` is removed from the system.

EINTR The `msgrcv()` function was interrupted by a signal.

EINVAL `msqid` is not a valid message queue identifier.

ENOMSG The queue does not contain a message of the desired type and `(msgflg & IPC_NOWAIT)` is non-zero.

The following sections are informative.

EXAMPLES

Receiving a Message

The following example receives the first message on the queue (based on the value of the `msgtyp` argument, 0). The queue is identified by the `msqid` argument (assuming that the value has previously been set). This call specifies that an error should be reported if no message is available.

able, but not if the message is too large. The message size is calculated directly using the sizeof operator.

```
#include <sys/msg.h>

...

int result;

int msqid;

struct message {
    long type;
    char text[20];
} msg;

long msgtyp = 0;

...

result = msgrcv(msqid, (void *) &msg, sizeof(msg.text),
               msgtyp, MSG_NOERROR | IPC_NOWAIT);
```

APPLICATION USAGE

The POSIX Realtime Extension defines alternative interfaces for interprocess communication (IPC). Application developers who need to use IPC should design their applications so that modules using the IPC routines described in Section 2.7, XSI Interprocess Communication can be easily modified to use the alternative interfaces.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.7, XSI Interprocess Communication, Section 2.8, Realtime, mq_close(), mq_getattr(), mq_notify(), mq_open(), mq_receive(), mq_send(), mq_setattr(), mq_unlink(), msgctl(), msgget(), msgsnd(), sigaction()

The Base Definitions volume of POSIX.1?2017, Section 3.226, Message Queue, <sys_msg.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

MSGRCV(3P)