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 'recvmsg.3p' command

$ man recvmsg.3p

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

recvmsg ? receive a message from a socket

SYNOPSIS

#include <sys/socket.h>

ssize_t recvmsg(int socket, struct msghdr *message, int flags);

DESCRIPTION

The recvmsg() function shall receive a message from a connection-mode or connectionless-mode socket. It is normally used with connectionless-mode sockets because it permits the application to retrieve the source address of received data.

The recvmsg() function takes the following arguments:

socket Specifies the socket file descriptor.

message Points to a msghdr structure, containing both the buffer to store the source address and the buffers for the incoming message. The length and format of the address depend on the address family of the socket. The msg_flags member is ig? nored on input, but may contain meaningful values on out?

put.

flags Specifies the type of message reception. Values of this ar?
gument are formed by logically OR'ing zero or more of the
following values:

MSG_OOB Requests out-of-band data. The significance and
semantics of out-of-band data are protocol-spe?
cific.

MSG_PEEK Peeks at the incoming message.

MSG_WAITALL On SOCK_STREAM sockets this requests that the
function block until the full amount of data
can be returned. The function may return the
smaller amount of data if the socket is a mes?
sage-based socket, if a signal is caught, if
the connection is terminated, if MSG_PEEK was
specified, or if an error is pending for the
socket.

The recvmsg() function shall receive messages from unconnected or con?
nected sockets and shall return the length of the message.

The recvmsg() function shall return the total length of the message.

For message-based sockets, such as SOCK_DGRAM and SOCK_SEQPACKET, the entire message shall be read in a single operation. If a message is too long to fit in the supplied buffers, and MSG_PEEK is not set in the flags argument, the excess bytes shall be discarded, and MSG_TRUNC shall be set in the msg_flags member of the msghdr structure. For stream-based sockets, such as SOCK_STREAM, message boundaries shall be ignored. In this case, data shall be returned to the user as soon as it becomes available, and no data shall be discarded.

If the MSG_WAITALL flag is not set, data shall be returned only up to the end of the first message.

If no messages are available at the socket and O_NONBLOCK is not set on the socket's file descriptor, recvmsg() shall block until a message ar?
rives. If no messages are available at the socket and O_NONBLOCK is set on the socket's file descriptor, the recvmsg() function shall fail and

set errno to [EAGAIN] or [EWOULDBLOCK].

In the msghdr structure, the msg_name member may be a null pointer if the source address is not required. Otherwise, if the socket is uncon? nected, the msg_name member points to a sockaddr structure in which the source address is to be stored, and the msg_namelen member on input specifies the length of the supplied sockaddr structure and on output specifies the length of the stored address. If the actual length of the address is greater than the length of the supplied sockaddr struc? ture, the stored address shall be truncated. If the socket is con? nected, the msg_name and msg_namelen members shall be ignored. The msg_iov and msg_iovlen fields are used to specify where the received data shall be stored. The msg_iov member points to an array of iovec structures; the msg_iovlen member shall be set to the dimension of this array. In each iovec structure, the iov_base field specifies a storage area and the iov_len field gives its size in bytes. Each storage area indicated by msg_iov is filled with received data in turn until all of the received data is stored or all of the areas have been filled.

Upon successful completion, the msg_flags member of the message header shall be the bitwise-inclusive OR of all of the following flags that indicate conditions detected for the received message:

MSG_EOR End-of-record was received (if supported by the protocol).

MSG_OOB Out-of-band data was received.

MSG_TRUNC Normal data was truncated.

MSG_CTRUNC Control data was truncated.

RETURN VALUE

Upon successful completion, recvmsg() shall return the length of the message in bytes. If no messages are available to be received and the peer has performed an orderly shutdown, recvmsg() shall return 0. Oth? erwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The recvmsg() function shall fail if:

EAGAIN or EWOULDBLOCK

The socket's file descriptor is marked O_NONBLOCK and no data is

waiting to be received; or MSG_OOB is set and no out-of-band data is available and either the socket's file descriptor is marked O_NONBLOCK or the socket does not support blocking to await out-of-band data.

EBADF The socket argument is not a valid open file descriptor.

ECONNRESET

A connection was forcibly closed by a peer.

EINTR This function was interrupted by a signal before any data was available.

EINVAL The sum of the iov_len values overflows a ssize_t, or the MSG_OOB flag is set and no out-of-band data is available.

EMSGSIZE

The msg_iovlen member of the msghdr structure pointed to by mes? sage is less than or equal to 0, or is greater than {IOV_MAX}.

ENOTCONN

A receive is attempted on a connection-mode socket that is not connected.

ENOTSOCK

The socket argument does not refer to a socket.

EOPNOTSUPP

The specified flags are not supported for this socket type.

ETIMEDOUT

The connection timed out during connection establishment, or due to a transmission timeout on active connection.

The recvmsg() function may fail if:

EIO An I/O error occurred while reading from or writing to the file system.

ENOBUFS

Insufficient resources were available in the system to perform the operation.

ENOMEM Insufficient memory was available to fulfill the request.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The select() and poll() functions can be used to determine when data is available to be received.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

poll(), pselect(), recv(), recvfrom(), send(), sendmsg(), sendto(), shutdown(), socket()

The Base Definitions volume of POSIX.1?2017, <sys_socket.h>