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 'recv.3p' command

$ man recv.3p

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

recv ? receive a message from a connected socket

SYNOPSIS

#include <sys/socket.h>

ssize_t recv(int socket, void *buffer, size_t length, int flags);

DESCRIPTION

The recv() function shall receive a message from a connection-mode or connectionless-mode socket. It is normally used with connected sockets because it does not permit the application to retrieve the source ad? dress of received data.

The recv() function takes the following arguments:

socket Specifies the socket file descriptor.

buffer Points to a buffer where the message should be stored.

length Specifies the length in bytes of the buffer pointed to by the buffer argument.

flags Specifies the type of message reception. Values of this argu? ment are formed by logically OR'ing zero or more of the fol?

lowing values:

MSG_PEEK Peeks at an incoming message. The data is treated as unread and the next recv() or similar function shall still return this data.

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

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 message-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 recv() function shall return the length of the message written to the buffer pointed to by the buffer argument. For message-based sock? ets, 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 buffer, and MSG_PEEK is not set in the flags argument, the ex? cess bytes shall be discarded. 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, recv() 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, recv() shall fail and set errno to [EAGAIN] or [EWOULDBLOCK].

RETURN VALUE

Upon successful completion, recv() shall return the length of the mes? sage in bytes. If no messages are available to be received and the peer

has performed an orderly shutdown, recv() shall return 0. Otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The recv() 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 file descriptor.

ECONNRESET

A connection was forcibly closed by a peer.

EINTR The recv() function was interrupted by a signal that was caught, before any data was available.

EINVAL The MSG_OOB flag is set and no out-of-band data is available.

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 or protocol.

ETIMEDOUT

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

The recv() 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 recv() function is equivalent to recvfrom() with null pointer ad?dress and address_len arguments, and to read() if the socket argument refers to a socket and the flags argument is 0.

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(), read(), recvmsg(), recvfrom(), send(), sendmsg(), sendto(), shutdown(), socket(), write()

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

COPYRIGHT