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 'send.3p' command

$ man send.3p

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

send ? send a message on a socket

SYNOPSIS

#include <sys/socket.h>

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

DESCRIPTION

The send() function shall initiate transmission of a message from the specified socket to its peer. The send() function shall send a message only when the socket is connected. If the socket is a connectionless-mode socket, the message shall be sent to the pre-specified peer ad? dress.

The send() function takes the following arguments:

socket Specifies the socket file descriptor.

buffer Points to the buffer containing the message to send.

length Specifies the length of the message in bytes.

flags Specifies the type of message transmission. Values of this argument are formed by logically OR'ing zero or more of the

following flags:

MSG_EOR Terminates a record (if supported by the pro? tocol).

MSG_OOB Sends out-of-band data on sockets that sup? port out-of-band communications. The signifi? cance and semantics of out-of-band data are protocol-specific.

MSG_NOSIGNAL Requests not to send the SIGPIPE signal if an attempt to send is made on a stream-oriented socket that is no longer connected. The [EPIPE] error shall still be returned.

The length of the message to be sent is specified by the length argu? ment. If the message is too long to pass through the underlying proto? col, send() shall fail and no data shall be transmitted.

Successful completion of a call to send() does not guarantee delivery of the message. A return value of -1 indicates only locally-detected errors.

If space is not available at the sending socket to hold the message to be transmitted, and the socket file descriptor does not have O_NONBLOCK set, send() shall block until space is available. If space is not available at the sending socket to hold the message to be transmitted, and the socket file descriptor does have O_NONBLOCK set, send() shall fail. The select() and poll() functions can be used to determine when it is possible to send more data.

The socket in use may require the process to have appropriate privi? leges to use the send() function.

RETURN VALUE

Upon successful completion, send() shall return the number of bytes sent. Otherwise, -1 shall be returned and errno set to indicate the er? ror.

ERRORS

The send() function shall fail if:

EAGAIN or EWOULDBLOCK

The socket's file descriptor is marked O_NONBLOCK and the re?
quested operation would block.

EBADF The socket argument is not a valid file descriptor.

ECONNRESET

A connection was forcibly closed by a peer.

EDESTADDRREQ

The socket is not connection-mode and no peer address is set.

EINTR A signal interrupted send() before any data was transmitted.

EMSGSIZE

The message is too large to be sent all at once, as the socket
requires.

ENOTCONN

The socket is not connected.

ENOTSOCK

The socket argument does not refer to a socket.

EOPNOTSUPP

The socket argument is associated with a socket that does not
support one or more of the values set in flags.

EPIPE The socket is shut down for writing, or the socket is connec?
tion-mode and is no longer connected. In the latter case, and if
the socket is of type SOCK_STREAM or SOCK_SEQPACKET and the
MSG_NOSIGNAL flag is not set, the SIGPIPE signal is generated to
the calling thread.

The send() function may fail if:

EACCES The calling process does not have appropriate privileges.

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

ENETDOWN

The local network interface used to reach the destination is
down.

ENETUNREACH

No route to the network is present.

ENOBUFS

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

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

If the socket argument refers to a connection-mode socket, the send() function is equivalent to sendto() (with any value for the dest_addr and dest_len arguments, as they are ignored in this case). If the socket argument refers to a socket and the flags argument is 0, the send() function is equivalent to write().

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

connect(), getsockopt(), poll(), pselect(), recv(), recvfrom(), recvmsg(), sendmsg(), sendto(), setsockopt(), shutdown(), socket(), write()

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

COPYRIGHT

nel.org/doc/man-pages/reporting_bugs.html .