



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 'recvfrom.3p' command

\$ man recvfrom.3p

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

recvfrom ? receive a message from a socket

SYNOPSIS

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

ssize_t recvfrom(int socket, void *restrict buffer, size_t length,
                 int flags, struct sockaddr *restrict address,
                 socklen_t *restrict address_len);
```

DESCRIPTION

The `recvfrom()` 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 `recvfrom()` function takes the following arguments:

- `socket` Specifies the socket file descriptor.
- `buffer` Points to the 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 argument are formed by logically OR'ing zero or more of the following values:

MSG_PEEK Peeks at an incoming message. The data is treated as unread and the next `recvfrom()` 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-specific.

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.

address A null pointer, or points to a `sockaddr` structure in which the sending address is to be stored. The length and format of the address depend on the address family of the socket.

address_len Either a null pointer, if address is a null pointer, or a pointer to a `socklen_t` object which on input specifies the length of the supplied `sockaddr` structure, and on output specifies the length of the stored address.

The `recvfrom()` function shall return the length of the message written to the buffer pointed to by the buffer argument. For message-based sockets, such as SOCK_RAW, 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 excess 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.

Not all protocols provide the source address for messages. If the address argument is not a null pointer and the protocol provides the source address of messages, the source address of the received message shall be stored in the sockaddr structure pointed to by the address argument, and the length of this address shall be stored in the object pointed to by the address_len argument.

If the actual length of the address is greater than the length of the supplied sockaddr structure, the stored address shall be truncated.

If the address argument is not a null pointer and the protocol does not provide the source address of messages, the value stored in the object pointed to by address is unspecified.

If no messages are available at the socket and O_NONBLOCK is not set on the socket's file descriptor, recvfrom() shall block until a message arrives. If no messages are available at the socket and O_NONBLOCK is set on the socket's file descriptor, recvfrom() shall fail and set errno to [EAGAIN] or [EWOULDBLOCK].

RETURN VALUE

Upon successful completion, recvfrom() 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, recvfrom() shall return 0. Otherwise, the function shall return -1 and set errno to indicate the error.

ERRORS

The recvfrom() 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 A signal interrupted `recvfrom()` 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.

ETIMEDOUT

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

The `recvfrom()` 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()`, `read()`, `recv()`, `recvmsg()`, `send()`, `sendmsg()`,

sendto(), shutdown(), socket(), write()

The Base Definitions volume of POSIX.1-2017, <sys_socket.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

RECVFROM(3P)