NAME

getpeername − get name of connected peer socket

SYNOPSIS

#include <sys/socket.h>

int getpeername(int *sockfd*, **struct sockaddr ****addr*, **socklen_t ****addrlen*);

DESCRIPTION

getpeername() returns the address of the peer connected to the socket *sockfd*, in the buffer pointed to by *addr*. The *addrlen* argument should be initialized to indicate the amount of space pointed to by *addr*. On return it contains the actual size of the name returned (in bytes). The name is truncated if the buffer provided is too small.

The returned address is truncated if the buffer provided is too small; in this case, *addrlen* will return a value greater than was supplied to the call.

RETURN VALUE

On success, zero is returned. On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EBADF
The argument *sockfd* is not a valid file descriptor.

EFAULT
The *addr* argument points to memory not in a valid part of the process address space.

EINVAL
addrlen is invalid (e.g., is negative).

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

ENOTCONN
The socket is not connected.

ENOTSOCK
The file descriptor *sockfd* does not refer to a socket.

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, SVr4, 4.4BSD (**getpeername**() first appeared in 4.2BSD).

NOTES

For background on the *socklen_t* type, see **accept**(2).

For stream sockets, once a **connect**(2) has been performed, either socket can call **getpeername**() to obtain the address of the peer socket. On the other hand, datagram sockets are connectionless. Calling **connect**(2) on a datagram socket merely sets the peer address for outgoing datagrams sent with **write**(2) or **recv**(2). The caller of **connect**(2) can use **getpeername**() to obtain the peer address that it earlier set for the socket. However, the peer socket is unaware of this information, and calling **getpeername**() on the peer socket will return no useful information (unless a **connect**(2) call was also executed on the peer). Note also that the receiver of a datagram can obtain the address of the sender when using **recvfrom**(2).

SEE ALSO

accept(2), **bind**(2), **getsockname**(2), **ip**(7), **socket**(7), **unix**(7)

COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.