NAME

bind - assign a local protocol address to a socket

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/socket.h>

int
bind(*int s*, *const struct sockaddr *addr*, *socklen_t addrlen*);

DESCRIPTION

The **bind**() system call assigns the local protocol address to a socket. When a socket is created with socket(2) it exists in an address family space but has no protocol address assigned. The **bind**() system call requests that *addr* be assigned to the socket.

NOTES

Binding an address in the UNIX domain creates a socket in the file system that must be deleted by the caller when it is no longer needed (using unlink(2)).

The rules used in address binding vary between communication domains. Consult the manual entries in section 4 for detailed information.

For maximum portability, you should always zero the socket address structure before populating it and passing it to **bind**().

RETURN VALUES

The **bind**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **bind**() system call will fail if:

[EAGAIN] Kernel resources to complete the request are temporarily unavailable.

[EBADF] The *s* argument is not a valid descriptor.

[EINVAL] The socket is already bound to an address, and the protocol does not support binding to a new address; or the socket has been shut down.

[EINVAL] The *addrlen* argument is not a valid length for the address family.

[ENOTSOCK] The *s* argument is not a socket.

[EADDRNOTAVAIL]
The specified address is not available from the local machine.

[EADDRINUSE] The specified address is already in use.

[EAFNOSUPPORT] Addresses in the specified address family cannot be used with this socket.

[EACCES] The requested address is protected, and the current user has inadequate permission to access it.

[EFAULT] The *addr* argument is not in a valid part of the user address space.

The following errors are specific to binding addresses in the UNIX domain.

[ENOTDIR] A component of the path prefix is not a directory.

[ENAMETOOLONG]
A component of a pathname exceeded 255 characters, or an entire path name exceeded 1023 characters.

[ENOENT] A prefix component of the path name does not exist.

[ELOOP] Too many symbolic links were encountered in translating the pathname.

[EIO] An I/O error occurred while making the directory entry or allocating the inode.

[EINTEGRITY] Corrupted data was detected while reading from the file system.

[EROFS] The name would reside on a read-only file system.

[EISDIR] An empty pathname was specified.

SEE ALSO

connect(2), getsockname(2), listen(2), socket(2)

HISTORY

The **bind**() system call appeared in 4.2BSD.