NAME

bind – bind a name to a socket

SYNOPSIS

cc [*flag* ...] *file* ... **–lsocket –lnsl** [*library* ...]

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

int bind(int *s***, const struct sockaddr** **name***, int** *namelen***);**

MT-LEVEL

Safe

DESCRIPTION

bind( ) assigns a name to an unnamed socket. When a socket is created with **socket**(3N), it exists in a name space (address family) but has no name assigned. **bind( )** requests that the name pointed to by *name* be assigned to the socket.

RETURN VALUES

If the bind is successful, **0** is returned. A return value of –**1** indicates an error, which is further specified in the global **errno**.

ERRORS

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

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

EADDRINUSE The specified address is already in use.

EADDRNOTAVAIL The specified address is not available on the local machine.

EBADF *s* is not a valid descriptor.

EINVAL *namelen* is not the size of a valid address for the specified address family.

EINVAL The socket is already bound to an address.

ENOSR There were insufficient STREAMS resources for the operation to complete.

ENOTSOCK *s* is a descriptor for a file, not a socket.

The following errors are specific to binding names in the UNIX domain:

EACCES Search permission is denied for a component of the path prefix of the pathname in *name*.

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

EISDIR A null pathname was specified.

ELOOP Too many symbolic links were encountered in translating the pathname in *name*.

ENOENT A component of the path prefix of the pathname in *name* does not exist.

ENOTDIR A component of the path prefix of the pathname in *name* is not a directory.

EROFS The inode would reside on a read-only file system.

SEE ALSO

unlink(2), **socket**(3N)

NOTES

Binding a name 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 name binding vary between communication domains.