

explain\_bind(3)

explain\_bind(3)

**NAME**

explain\_bind – explain bind(2) errors

**SYNOPSIS**

#include &lt;libexplain/bind.h&gt;

```

const char *explain_bind(int fildes, const struct sockaddr *sock_addr, int sock_addr_size);
const char *explain_errno_bind(int errnum, int fildes, const struct sockaddr *sock_addr, int
sock_addr_size);
void explain_message_bind(char *message, int message_size, int fildes, const struct sockaddr
*sock_addr, int sock_addr_size);
void explain_message_errno_bind(char *message, int message_size, int errnum, int fildes, const struct
sockaddr *sock_addr, int sock_addr_size);

```

**DESCRIPTION**

These functions may be used to obtain explanations for errors returned by the *bind(2)* system call.

**explain\_bind**

```
const char *explain_bind(int fildes, const struct sockaddr *sock_addr, int sock_addr_size);
```

The **explain\_bind** function is used to obtain an explanation of an error returned by the *bind(2)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

This function is intended to be used in a fashion similar to the following example:

```

if (bind(fildes, sock_addr, sock_addr_size) < 0)
{
    fprintf(stderr, "%s\n",
        explain_bind(fildes, sock_addr, sock_addr_size));
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain\_bind\_or\_die(3)* function.

*fildes*     The original *fildes*, exactly as passed to the *bind(2)* system call.

*sock\_addr*

The original *sock\_addr*, exactly as passed to the *bind(2)* system call.

*sock\_addr\_size*

The original *sock\_addr\_size*, exactly as passed to the *bind(2)* system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

**Note:** This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

**explain\_errno\_bind**

```
const char *explain_errno_bind(int errnum, int fildes, const struct sockaddr *sock_addr, int
sock_addr_size);
```

The **explain\_errno\_bind** function is used to obtain an explanation of an error returned by the *bind(2)* system call. The least the message will contain is the value of `strerror(errnum)`, but usually it will do much better, and indicate the underlying cause in more detail.

This function is intended to be used in a fashion similar to the following example:

```

if (bind(fildes, sock_addr, sock_addr_size) < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_bind(err,
        fildes, sock_addr, sock_addr_size));
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain\_bind\_or\_die(3)* function.



explain\_bind(3)

explain\_bind(3)

*errnum* The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

*fildev* The original *fildev*, exactly as passed to the *bind(2)* system call.

*sock\_addr*

The original *sock\_addr*, exactly as passed to the *bind(2)* system call.

*sock\_addr\_size*

The original *sock\_addr\_size*, exactly as passed to the *bind(2)* system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

**Note:** This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

### explain\_message\_bind

```
void explain_message_bind(char *message, int message_size, int fildev, const struct sockaddr
*sock_addr, int sock_addr_size);
```

The **explain\_message\_bind** function may be used to obtain an explanation of an error returned by the *bind(2)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

This function is intended to be used in a fashion similar to the following example:

```
if (bind(fildev, sock_addr, sock_addr_size) < 0)
{
    char message[3000];
    explain_message_bind(message, sizeof(message),
        fildev, sock_addr, sock_addr_size);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_bind\_or\_die(3)* function.

*message* The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

*message\_size*

The size in bytes of the location in which to store the returned message.

*fildev* The original *fildev*, exactly as passed to the *bind(2)* system call.

*sock\_addr*

The original *sock\_addr*, exactly as passed to the *bind(2)* system call.

*sock\_addr\_size*

The original *sock\_addr\_size*, exactly as passed to the *bind(2)* system call.

### explain\_message\_errno\_bind

```
void explain_message_errno_bind(char *message, int message_size, int errnum, int fildev, const struct
sockaddr *sock_addr, int sock_addr_size);
```

The **explain\_message\_errno\_bind** function may be used to obtain an explanation of an error returned by the *bind(2)* system call. The least the message will contain is the value of `strerror(errnum)`, but usually it will do much better, and indicate the underlying cause in more detail.

This function is intended to be used in a fashion similar to the following example:

```
if (bind(fildev, sock_addr, sock_addr_size) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_bind(message, sizeof(message), err,
```



explain\_bind(3)

explain\_bind(3)

```
        fildes, sock_addr, sock_addr_size);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_bind\_or\_die*(3) function.

*message* The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

*message\_size*

The size in bytes of the location in which to store the returned message.

*errnum* The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

*fildes* The original fildes, exactly as passed to the *bind*(2) system call.

*sock\_addr*

The original sock\_addr, exactly as passed to the *bind*(2) system call.

*sock\_addr\_size*

The original sock\_addr\_size, exactly as passed to the *bind*(2) system call.

## SEE ALSO

*bind*(2) bind a name to a socket

*explain\_bind\_or\_die*(3)

bind a name to a socket and report errors

## COPYRIGHT

libexplain version 1.4

Copyright © 2008 Peter Miller

