

explain\_writev(3)

explain\_writev(3)

**NAME**

explain\_writev – explain writev(2) errors

**SYNOPSIS**

```
#include <libexplain/writev.h>

const char *explain_writev(int fildes, const struct iovec *data, int data_size);
const char *explain_errno_writev(int errnum, int fildes, const struct iovec *data, int data_size);
void explain_message_writev(char *message, int message_size, int fildes, const struct iovec *data, int data_size);
void explain_message_errno_writev(char *message, int message_size, int errnum, int fildes, const struct iovec *data, int data_size);
```

**DESCRIPTION**

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

**explain\_writev**

```
const char *explain_writev(int fildes, const struct iovec *data, int data_size);
```

The **explain\_writev** function is used to obtain an explanation of an error returned by the *writev(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.

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

*data*     The original data, exactly as passed to the *writev(2)* system call.

*data\_size*

The original data\_size, exactly as passed to the *writev(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.

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

```
ssize_t result = writev(fildes, data, data_size);
if (result < 0)
{
    fprintf(stderr, "%s\n", explain_writev(fildes, data,
    data_size));
    exit(EXIT_FAILURE);
}
```

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

**explain\_errno\_writev**

```
const char *explain_errno_writev(int errnum, int fildes, const struct iovec *data, int data_size);
```

The **explain\_errno\_writev** function is used to obtain an explanation of an error returned by the *writev(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.

*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 *writev(2)* system call.

*data*     The original data, exactly as passed to the *writev(2)* system call.

*data\_size*

The original data\_size, exactly as passed to the *writev(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.



explain\_writev(3)

explain\_writev(3)

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

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

```
ssize_t result = writev(fildes, data, data_size);
if (result < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_writev(err, fildes,
        data, data_size));
    exit(EXIT_FAILURE);
}
```

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

### explain\_message\_writev

```
void explain_message_writev(char *message, int message_size, int fildes, const struct iovec *data, int data_size);
```

The **explain\_message\_writev** function is used to obtain an explanation of an error returned by the *writev*(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.

*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.

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

*data* The original data, exactly as passed to the *writev*(2) system call.

*data\_size*

The original *data\_size*, exactly as passed to the *writev*(2) system call.

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

```
ssize_t result = writev(fildes, data, data_size);
if (result < 0)
{
    char message[3000];
    explain_message_writev(message, sizeof(message), fildes,
        data, data_size);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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

### explain\_message\_errno\_writev

```
void explain_message_errno_writev(char *message, int message_size, int errnum, int fildes, const struct iovec *data, int data_size);
```

The **explain\_message\_errno\_writev** function is used to obtain an explanation of an error returned by the *writev*(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.

*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*.



explain\_writev(3)

explain\_writev(3)

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

*data*       The original data, exactly as passed to the *writev*(2) system call.

*data\_size*

The original *data\_size*, exactly as passed to the *writev*(2) system call.

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

```
ssize_t result = writev(fildev, data, data_size);
if (result < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_writev(message, sizeof(message), err,
    fildev, data, data_size);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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

## SEE ALSO

*writev*(2)

write data from multiple buffers

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

write data from multiple buffers and report errors

## COPYRIGHT

libexplain version 1.4

Copyright © 2009 Peter Miller

