

explain_ftruncate(3)

explain_ftruncate(3)

NAME

explain_ftruncate – explain ftruncate(2) errors

SYNOPSIS

#include <libexplain/ftruncate.h>

const char *explain_ftruncate(int fildes, long long length);

const char *explain_errno_ftruncate(int errnum, int fildes, long long length);

void explain_message_ftruncate(char *message, int message_size, int fildes, long long length);

void explain_message_errno_ftruncate(char *message, int message_size, int errnum, int fildes, long long length);

DESCRIPTION

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

explain_ftruncate

const char *explain_ftruncate(int fildes, long long length);

The **explain_ftruncate** function is used to obtain an explanation of an error returned by the *ftruncate(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 (ftruncate(fildes, length) < 0)
{
    fprintf(stderr, "%s\n", explain_ftruncate(fildes, length));
    exit(EXIT_FAILURE);
}
```

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

length The original length, exactly as passed to the *ftruncate(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_ftruncate

const char *explain_errno_ftruncate(int errnum, int fildes, long long length);

The **explain_errno_ftruncate** function is used to obtain an explanation of an error returned by the *ftruncate(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 (ftruncate(fildes, length) < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_ftruncate(err, fildes, length));
    exit(EXIT_FAILURE);
}
```

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 *ftruncate(2)* system call.

length The original length, exactly as passed to the *ftruncate(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



explain_ftruncate(3)

explain_ftruncate(3)

other functions in this library.

explain_message_ftruncate

```
void explain_message_ftruncate(char *message, int message_size, int fildes, long long length);
```

The **explain_message_ftruncate** function may be used to obtain an explanation of an error returned by the *ftruncate(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 (ftruncate(fildes, length) < 0)
{
    char message[3000];
    explain_message_ftruncate(message, sizeof(message), fildes, length);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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 *ftruncate(2)* system call.

length

The original length, exactly as passed to the *ftruncate(2)* system call.

explain_message_errno_ftruncate

```
void explain_message_errno_ftruncate(char *message, int message_size, int errnum, int fildes, long long length);
```

The **explain_message_errno_ftruncate** function may be used to obtain an explanation of an error returned by the *ftruncate(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 (ftruncate(fildes, length) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_ftruncate(message, sizeof(message), err,
        fildes, length);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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 *ftruncate(2)* system call.

length

The original length, exactly as passed to the *ftruncate(2)* system call.

SEE ALSO

ftruncate(2)

truncate a file to a specified length



explain_ftruncate(3)

explain_ftruncate(3)

explain_ftruncate_or_die(3)

truncate a file and report errors

COPYRIGHT

libexplain version 1.4

Copyright © 2008 Peter Miller

