

explain_chdir(3)

explain_chdir(3)

NAME

explain_chdir – explain chdir(2) errors

SYNOPSIS

```
#include <libexplain/chdir.h>
const char *explain_chdir(const char *pathname);
void explain_message_chdir(char *message, int message_size, const char *pathname);
const char *explain_errno_chdir(int errnum, const char *pathname);
void explain_message_errno_chdir(char *message, int message_size, int errnum, const char *pathname);
```

DESCRIPTION

These function may be used to obtain explanations of *chdir(2)* errors.

explain_chdir

```
const char *explain_chdir(const char *pathname);
```

The `explain_chdir` function is used to obtain an explanation of an error returned by the *chdir(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 (chdir(pathname) < 0)
{
    fprintf(stderr, '%s0, explain_chdir(pathname));
    exit(EXIT_FAILURE);
}
```

pathname

The original pathname, exactly as passed to the *chdir(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_chdir

```
const char *explain_errno_chdir(int errnum, const char *pathname);
```

The `explain_errno_chdir` function is used to obtain an explanation of an error returned by the *chdir(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 (chdir(pathname) < 0)
{
    int err = errno;
    fprintf(stderr, '%s0, explain_errno_chdir(err, pathname));
    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*.

pathname

The original pathname, exactly as passed to the *chdir(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_chdir(3)

explain_chdir(3)

explain_message_chdir

```
void explain_message_chdir(char *message, int message_size, const char *pathname);
```

The `explain_message_chdir` function is used to obtain an explanation of an error returned by the `chdir(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 (chdir(pathname) < 0)
{
    char message[3000];
    explain_message_chdir(message, sizeof(message), pathname);
    fprintf(stderr, '%s0', message);
    exit(EXIT_FAILURE);
}
```

message The location in which to store the returned message. Because a message return buffer has been supplied, this function is thread safe.

message_size

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

pathname

The original pathname, exactly as passed to the `chdir(2)` system call.

explain_message_errno_chdir

```
void explain_message_errno_chdir(char *message, int message_size, int errnum, const char *pathname);
```

The `explain_message_errno_chdir` function is used to obtain an explanation of an error returned by the `chdir(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 (chdir(pathname) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_chdir(message, sizeof(message), err,
                                pathname);
    fprintf(stderr, '%s0', message);
    exit(EXIT_FAILURE);
}
```

message The location in which to store the returned message. Because a message return buffer has been 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`.

pathname

The original pathname, exactly as passed to the `chdir(2)` system call.

COPYRIGHT

libexplain version 1.4

Copyright © 2008 Peter Miller

AUTHOR

Written by Peter Miller <pmiller AT opensource DOT org DOT au>

