

explain_fchmod(3)

explain_fchmod(3)

NAMEexplain_fchmod – explain *fchmod(2)* errors**SYNOPSIS**

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

const char *explain_fchmod(int fildes, mode_t mode);
const char *explain_errno_fchmod(int errnum, int fildes, mode_t mode);
void explain_message_fchmod(char *message, int message_size, int fildes, mode_t mode);
void explain_message_errno_fchmod(char *message, int message_size, int errnum, int fildes, mode_t mode);
```

DESCRIPTION

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

explain_fchmod

```
const char *explain_fchmod(int fildes, mode_t mode);
```

The **explain_fchmod** function is used to obtain an explanation of an error returned by the *fchmod(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 *fchmod(2)* system call.

mode The original mode, exactly as passed to the *fchmod(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:

```
if (fchmod(fildes, mode) < 0)
{
    fprintf(stderr, "%s\n", explain_fchmod(fildes, mode));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain_fchmod_or_die(3)* function.

explain_errno_fchmod

```
const char *explain_errno_fchmod(int errnum, int fildes, mode_t mode);
```

The **explain_errno_fchmod** function is used to obtain an explanation of an error returned by the *fchmod(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 *fchmod(2)* system call.

mode The original mode, exactly as passed to the *fchmod(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:

```
if (fchmod(fildes, mode) < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_fchmod(err, fildes,
```



explain_fchmod(3)

explain_fchmod(3)

```

        mode) );
        exit(EXIT_FAILURE);
    }

```

The above code example is available pre-packaged as the *explain_fchmod_or_die(3)* function.

explain_message_fchmod

```
void explain_message_fchmod(char *message, int message_size, int fildes, mode_t mode);
```

The **explain_message_fchmod** function is used to obtain an explanation of an error returned by the *fchmod(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 *fchmod(2)* system call.

mode The original mode, exactly as passed to the *fchmod(2)* system call.

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

```

if (fchmod(fildes, mode) < 0)
{
    char message[3000];
    explain_message_fchmod(message, sizeof(message), fildes,
    mode);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain_fchmod_or_die(3)* function.

explain_message_errno_fchmod

```
void explain_message_errno_fchmod(char *message, int message_size, int errnum, int fildes, mode_t mode);
```

The **explain_message_errno_fchmod** function is used to obtain an explanation of an error returned by the *fchmod(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*.

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

mode The original mode, exactly as passed to the *fchmod(2)* system call.

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

```

if (fchmod(fildes, mode) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_fchmod(message, sizeof(message), err,
    fildes, mode);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```



`explain_fchmod(3)``explain_fchmod(3)`

The above code example is available pre-packaged as the *explain_fchmod_or_die(3)* function.

SEE ALSO*fchmod(2)*

change permissions of an open file

explain_fchmod_or_die(3)

change permissions of an open file and report errors

COPYRIGHT

libexplain version 1.4

Copyright © 2013 Peter Miller

