

explain_getc(3)

explain_getc(3)

NAME

explain_getc – explain getc(3) errors

SYNOPSIS

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

const char *explain_getc(FILE *fp);
const char *explain_errno_getc(int errnum, FILE *fp);
void explain_message_getc(char *message, int message_size, FILE *fp);
void explain_message_errno_getc(char *message, int message_size, int errnum, FILE *fp);
```

DESCRIPTION

These functions may be used to obtain explanations for errors returned by the *getc(3)* system call.

explain_getc

```
const char *explain_getc(FILE *fp);
```

The **explain_getc** function is used to obtain an explanation of an error returned by the *getc(3)* 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:

```
int c = getc(fp);
if (c == EOF && ferror(fp))
{
    fprintf(stderr, "%s\n", explain_getc(fp));
    exit(EXIT_FAILURE);
}
```

fp The original *fp*, exactly as passed to the *getc(3)* 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_getc

```
const char *explain_errno_getc(int errnum, FILE *fp);
```

The **explain_errno_getc** function is used to obtain an explanation of an error returned by the *getc(3)* 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:

```
int c = getc(fp);
if (c == EOF && ferror(fp))
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_getc(err, fp));
    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*.

fp The original *fp*, exactly as passed to the *getc(3)* 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_getc(3)

explain_getc(3)

explain_message_getc

```
void explain_message_getc(char *message, int message_size, FILE *fp);
```

The **explain_message_getc** function may be used to obtain an explanation of an error returned by the *getc(3)* 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:

```
int c = getc(fp);
if (c == EOF && ferror(fp))
{
    char message[3000];
    explain_message_getc(message, sizeof(message), fp);
    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.

fp

The original *fp*, exactly as passed to the *getc(3)* system call.

explain_message_errno_getc

```
void explain_message_errno_getc(char *message, int message_size, int errnum, FILE *fp);
```

The **explain_message_errno_getc** function may be used to obtain an explanation of an error returned by the *getc(3)* 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:

```
int c = getc(fp);
if (c == EOF && ferror(fp))
{
    int err = errno;
    char message[3000];
    explain_message_errno_getc(message, sizeof(message), err, fp);
    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*.

fp

The original *fp*, exactly as passed to the *getc(3)* system call.

SEE ALSO

getc(3) input of characters

explain_getc_or_die(3)

input of characters and report errors

COPYRIGHT

libexplain version 1.4

Copyright © 2008 Peter Miller

