

explain\_getchar(3)

explain\_getchar(3)

**NAME**

explain\_getchar – explain getchar(3) errors

**SYNOPSIS**

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

const char *explain_getchar(void);
const char *explain_errno_getchar(int errnum, void);
void explain_message_getchar(char *message, int message_size);
void explain_message_errno_getchar(char *message, int message_size, int errnum);
```

**DESCRIPTION**

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

**explain\_getchar**

```
const char *explain_getchar(void);
```

The **explain\_getchar** function is used to obtain an explanation of an error returned by the *getchar(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 = getchar();
if (c == EOF && ferror(stdin))
{
    fprintf(stderr, "%s\n", explain_getchar());
    exit(EXIT_FAILURE);
}
```

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\_getchar**

```
const char *explain_errno_getchar(int errnum);
```

The **explain\_errno\_getchar** function is used to obtain an explanation of an error returned by the *getchar(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 = getchar();
if (c == EOF && ferror(stdin))
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_getchar(err, ));
    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*.

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\_message\_getchar**

```
void explain_message_getchar(char *message, int message_size);
```

The **explain\_message\_getchar** function may be used to obtain an explanation of an error returned by



explain\_getchar(3)

explain\_getchar(3)

the *getchar(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 = getchar();
if (c == EOF && ferror(stdin))
{
    char message[3000];
    explain_message_getchar(message, sizeof(message), );
    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.

### explain\_message\_errno\_getchar

```
void explain_message_errno_getchar(char *message, int message_size, int errnum);
```

The **explain\_message\_errno\_getchar** function may be used to obtain an explanation of an error returned by the *getchar(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 = getchar();
if (c == EOF && ferror(stdin))
{
    int err = errno;
    char message[3000];
    explain_message_errno_getchar(message, sizeof(message), err, );
    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*.

### SEE ALSO

*getchar(3)*

input of characters

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

input of characters and report errors

### COPYRIGHT

libexplain version 1.4

Copyright © 2008 Peter Miller

