

explain\_getrusage(3)

explain\_getrusage(3)

**NAME**explain\_getrusage – explain *getrusage*(2) errors**SYNOPSIS**

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

const char *explain_getrusage(int who, struct rusage *usage);
const char *explain_errno_getrusage(int errnum, int who, struct rusage *usage);
void explain_message_getrusage(char *message, int message_size, int who, struct rusage *usage);
void explain_message_errno_getrusage(char *message, int message_size, int errnum, int who, struct rusage *usage);
```

**DESCRIPTION**

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

**explain\_getrusage**

```
const char *explain_getrusage(int who, struct rusage *usage);
```

The **explain\_getrusage** function is used to obtain an explanation of an error returned by the *getrusage*(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.

*who* The original who, exactly as passed to the *getrusage*(2) system call.

*usage* The original usage, exactly as passed to the *getrusage*(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 (getrusage(who, usage) < 0)
{
    fprintf(stderr, "%s\n", explain_getrusage(who, usage));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_getrusage\_or\_die*(3) function.

**explain\_errno\_getrusage**

```
const char *explain_errno_getrusage(int errnum, int who, struct rusage *usage);
```

The **explain\_errno\_getrusage** function is used to obtain an explanation of an error returned by the *getrusage*(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*.

*who* The original who, exactly as passed to the *getrusage*(2) system call.

*usage* The original usage, exactly as passed to the *getrusage*(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 (getrusage(who, usage) < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_getrusage(err, who,
```



explain\_getrusage(3)

explain\_getrusage(3)

```

        usage) );
        exit(EXIT_FAILURE);
    }

```

The above code example is available pre-packaged as the *explain\_getrusage\_or\_die(3)* function.

### explain\_message\_getrusage

```
void explain_message_getrusage(char *message, int message_size, int who, struct rusage *usage);
```

The **explain\_message\_getrusage** function is used to obtain an explanation of an error returned by the *getrusage(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.

*who* The original who, exactly as passed to the *getrusage(2)* system call.

*usage* The original usage, exactly as passed to the *getrusage(2)* system call.

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

```

if (getrusage(who, usage) < 0)
{
    char message[3000];
    explain_message_getrusage(message, sizeof(message), who,
    usage);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain\_getrusage\_or\_die(3)* function.

### explain\_message\_errno\_getrusage

```
void explain_message_errno_getrusage(char *message, int message_size, int errnum, int who, struct rusage *usage);
```

The **explain\_message\_errno\_getrusage** function is used to obtain an explanation of an error returned by the *getrusage(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*.

*who* The original who, exactly as passed to the *getrusage(2)* system call.

*usage* The original usage, exactly as passed to the *getrusage(2)* system call.

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

```

if (getrusage(who, usage) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_getrusage(message, sizeof(message),
    err, who, usage);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```



`explain_getrusage(3)``explain_getrusage(3)`

The above code example is available pre-packaged as the *explain\_getrusage\_or\_die(3)* function.

**SEE ALSO***getrusage(2)*

get resource usage

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

get resource usage and report errors

**COPYRIGHT**

libexplain version 1.4

Copyright © 2013 Peter Miller

