

explain\_ustat(3)

explain\_ustat(3)

**NAME**

explain\_ustat – explain ustat(2) errors

**SYNOPSIS**

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

const char *explain_ustat(dev_t dev, struct ustat *ubuf);
const char *explain_errno_ustat(int errnum, dev_t dev, struct ustat *ubuf);
void explain_message_ustat(char *message, int message_size, dev_t dev, struct ustat *ubuf);
void explain_message_errno_ustat(char *message, int message_size, int errnum, dev_t dev, struct ustat *ubuf);
```

**DESCRIPTION**

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

**explain\_ustat**

```
const char *explain_ustat(dev_t dev, struct ustat *ubuf);
```

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

*dev*        The original dev, exactly as passed to the *ustat(2)* system call.

*ubuf*       The original ubuf, exactly as passed to the *ustat(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 (ustat(dev, ubuf) < 0)
{
    fprintf(stderr, "%s\n", explain_ustat(dev, ubuf));
    exit(EXIT_FAILURE);
}
```

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

**explain\_errno\_ustat**

```
const char *explain_errno_ustat(int errnum, dev_t dev, struct ustat *ubuf);
```

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

*dev*        The original dev, exactly as passed to the *ustat(2)* system call.

*ubuf*       The original ubuf, exactly as passed to the *ustat(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 (ustat(dev, ubuf) < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_ustat(err, dev,
```



explain\_ustat(3)

explain\_ustat(3)

```

        ubuf ) );
        exit (EXIT_FAILURE);
    }

```

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

### explain\_message\_ustat

```
void explain_message_ustat(char *message, int message_size, dev_t dev, struct ustat *ubuf);
```

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

*dev* The original dev, exactly as passed to the *ustat(2)* system call.

*ubuf* The original ubuf, exactly as passed to the *ustat(2)* system call.

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

```

if (ustat(dev, ubuf) < 0)
{
    char message[3000];
    explain_message_ustat(message, sizeof(message), dev, ubuf);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

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

### explain\_message\_errno\_ustat

```
void explain_message_errno_ustat(char *message, int message_size, int errnum, dev_t dev, struct ustat *ubuf);
```

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

*dev* The original dev, exactly as passed to the *ustat(2)* system call.

*ubuf* The original ubuf, exactly as passed to the *ustat(2)* system call.

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

```

if (ustat(dev, ubuf) < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_ustat(message, sizeof(message), err,
    dev, ubuf);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

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



`explain_ustat(3)``explain_ustat(3)`**SEE ALSO***ustat(2)* get file system statistics*explain\_ustat\_or\_die(3)*

get file system statistics and report errors

**COPYRIGHT**

libexplain version 1.4

Copyright © 2009 Peter Miller

