NAME

explain_setpriority − explain *setpriority*(2) errors

SYNOPSIS

#include <libexplain/setpriority.h>

const char *explain_setpriority(int which, int who, int prio);
const char *explain_errno_setpriority(int errnum, int which, int who, int prio);
void explain_message_setpriority(char *message, int message_size, int which, int who, int prio);
void explain_message_errno_setpriority(char *message, int message_size, int errnum, int which, int who, int prio);

DESCRIPTION

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

explain_setpriority

const char *explain_setpriority(int which, int who, int prio);

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

which The original which, exactly as passed to the *setpriority*(2) system call.

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

prio The original prio, exactly as passed to the *setpriority*(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 (setpriority(which, who, prio) < 0)
{
   fprintf(stderr, "%s\n", explain_setpriority(which, who,
   prio));
    exit(EXIT_FAILURE);
}
```

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

explain_errno_setpriority

const char *explain_errno_setpriority(int errnum, int which, int who, int prio);

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

which The original which, exactly as passed to the *setpriority*(2) system call.

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

prio The original prio, exactly as passed to the *setpriority*(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 (setpriority(which, who, prio) < 0)
{
    int err = errno;
   fprintf(stderr, "%s\n", explain_errno_setpriority(err,
   which, who, prio));
    exit(EXIT_FAILURE);
}
```

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

explain_message_setpriority

void explain_message_setpriority(char *message, int message_size, int which, int who, int prio);

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

which The original which, exactly as passed to the *setpriority*(2) system call.

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

prio The original prio, exactly as passed to the *setpriority*(2) system call.

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

```
if (setpriority(which, who, prio) < 0)
{
    char message[3000];
   explain_message_setpriority(message, sizeof(message),
   which, who, prio);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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

explain_message_errno_setpriority

void explain_message_errno_setpriority(char *message, int message_size, int errnum, int which, int who, int prio);

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

which The original which, exactly as passed to the *setpriority*(2) system call.

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

prio The original prio, exactly as passed to the *setpriority*(2) system call.

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

```
if (setpriority(which, who, prio) < 0)
{
```

```
        int err = errno;
        char message[3000];
        explain_message_errno_setpriority(message, sizeof(message),
        err, which, who, prio);
        fprintf(stderr, "%s\n", message);
        exit(EXIT_FAILURE);
}
```

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

SEE ALSO

setpriority(2)
set program scheduling priority

explain_setpriority_or_die(3)
set program scheduling priority and report errors

COPYRIGHT