NAME

explain_mkdtemp − explain *mkdtemp*(3) errors

SYNOPSIS

#include <libexplain/mkdtemp.h>

const char *explain_mkdtemp(char *pathname);
const char *explain_errno_mkdtemp(int errnum, char *pathname);
void explain_message_mkdtemp(char *message, int message_size, char *pathname);
void explain_message_errno_mkdtemp(char *message, int message_size, int errnum, char *pathname);

DESCRIPTION

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

explain_mkdtemp

const char *explain_mkdtemp(char *pathname);

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

pathname
The original pathname, exactly as passed to the *mkdtemp*(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.

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

```
char *result = mkdtemp(pathname);
if (!result)
{
    fprintf(stderr, "%s\n", explain_mkdtemp(pathname));
     exit(EXIT_FAILURE);
}
```

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

explain_errno_mkdtemp

const char *explain_errno_mkdtemp(int errnum, char *pathname);

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

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*.

pathname
The original pathname, exactly as passed to the *mkdtemp*(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.

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

```
char *result = mkdtemp(pathname);
if (!result)
{
    int err = errno;
   fprintf(stderr, "%s\n", explain_errno_mkdtemp(err,
```

```
        pathname));
         exit(EXIT_FAILURE);
    }
```

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

explain_message_mkdtemp

void explain_message_mkdtemp(char *message, int message_size, char *pathname);

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

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.

pathname
The original pathname, exactly as passed to the *mkdtemp*(3) system call.

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

```
char *result = mkdtemp(pathname);
if (!result)
{
    char message[3000];
    explain_message_mkdtemp(message, sizeof(message), path-
    name);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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

explain_message_errno_mkdtemp

void explain_message_errno_mkdtemp(char *message, int message_size, int errnum, char *pathname);

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

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*.

pathname
The original pathname, exactly as passed to the *mkdtemp*(3) system call.

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

```
char *result = mkdtemp(pathname);
if (!result)
{
    int err = errno;
    char message[3000];
    explain_message_errno_mkdtemp(message, sizeof(message),
    err, pathname);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

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

SEE ALSO

mkdtemp(3)
create a unique temporary directory

explain_mkdtemp_or_die(3)
create a unique temporary directory and report errors

COPYRIGHT