strerror(3) - Linux man page

Name

strerror, strerror_r - return string describing error number

Synopsis

```
#include <string.h>

char *strerror(int errnum);

int strerror_r(int errnum, char *buf, size_t buflen);
            /* XSI-compliant */

char *strerror_r(int errnum, char *buf, size_t buflen);
            /* GNU-specific */
```

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

The XSI-compliant version of **strerror_r**() is provided if:
(_POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE >= 600) && ! _GNU_SOURCE
Otherwise, the GNU-specific version is provided.

Description

The **strerror**() function returns a pointer to a string that describes the error code passed in the argument *errnum*, possibly using the **LC_MESSAGES** part of the current locale to select the appropriate language. (For example, if *errnum* is **EINVAL**, the returned description will "Invalid argument".) This string must not be modified by the application, but may be modified by a subsequent call to **strerror**(). No library function, including **perror**(3), will modify this string.

The **strerror_r**() function is similar to **strerror**(), but is thread safe. This function is available in two versions: an XSI-compliant version specified in POSIX.1-2001 (available since glibc 2.3.4, but not POSIX-compliant until glibc 2.13), and a GNU-specific version (available since glibc 2.0). The XSI-compliant version is provided with the feature test macros settings shown in the SYNOPSIS; otherwise the GNU-specific version is provided. If no feature test macros are explicitly defined, then (since glibc 2.4) **_POSIX_SOURCE** is defined by default with the value 200112L, so that the XSI-compliant version of **strerror_r**() is provided by default.

The XSI-compliant **strerror_r**() is preferred for portable applications. It returns the error string in the user-supplied buffer *buf* of length *buflen*.

The GNU-specific **strerror_r**() returns a pointer to a string containing the error message. This may be either a pointer to a string that the function stores in *buf*, or a pointer to some (immutable) static string (in which case *buf* is unused). If the function stores a string in *buf*, then at most *buflen* bytes are stored (the string may be truncated if *buflen* is too small and *errnum* is unknown). The string always includes a terminating null byte.

Return Value

The **strerror**() and the GNU-specific **strerror_r**() functions return the appropriate error description string, or an "Unknown error nnn" message if the error number is unknown.

POSIX.1-2001 and POSIX.1-2008 require that a successful call to **strerror**() shall leave *errno* unchanged, and note that, since no function return value is reserved to indicate an error, an application that wishes to check for errors should initialize *errno* to zero before the call, and then check *errno* after the call.

The XSI-compliant **strerror_r**() function returns 0 on success. On error, a (positive) error number is returned (since glibc 2.13), or -1 is returned and *errno* is set to indicate the error (glibc versions before 2.13).

Errors

EINVAL

The value of *errnum* is not a valid error number.

ERANGE

Insufficient storage was supplied to contain the error description string.

Conforming To

strerror() is specified by POSIX.1-2001, C89, C99. **strerror_r**() is specified by POSIX.1-2001.

The GNU-specific **strerror_r**() function is a nonstandard extension.

POSIX.1-2001 permits **strerror**() to set *errno* if the call encounters an error, but does not specify what value should be returned as the function result in the event of an error. On some systems, **strerror**() returns NULL if the error number is unknown. On other systems, **strerror**() returns a string something like "Error nnn occurred" and sets *errno* to **EINVAL** if the error number is unknown. C99 and POSIX.1-2008 require the return value to be non-NULL.

See Also

err(3), **errno**(3), **error**(3), **perror**(3), **strsignal**(3)

Referenced By

archive_util(3), **assert_perror**(3), **explain**(3), **explain_lca2010**(1), **lwres_gai_strerror**(3), **pcap_strerror**(3)