



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'dlerror.3p' command

\$ man dlerror.3p

DLERROR(3P) POSIX Programmer's Manual DLERROR(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

dlerror ? get diagnostic information

SYNOPSIS

```
#include <dlfcn.h>

char *dlerror(void);
```

DESCRIPTION

The `dlerror()` function shall return a null-terminated character string (with no trailing `<newline>`) that describes the last error that occurred during dynamic linking processing. If no dynamic linking errors have occurred since the last invocation of `dlerror()`, `dlerror()` shall return `NULL`. Thus, invoking `dlerror()` a second time, immediately following a prior invocation, shall result in `NULL` being returned.

It is implementation-defined whether or not the `dlerror()` function is thread-safe. A thread-safe implementation shall return only errors that occur on the current thread.

RETURN VALUE

If successful, `dlerror()` shall return a null-terminated character

string; otherwise, NULL shall be returned.

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to `dlerror()` in the same thread (if `dlerror()` is thread-safe) or in any thread (if `dlerror()` is not thread-safe). The returned pointer might also be invalidated if the calling thread is terminated.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

The following example prints out the last dynamic linking error:

```
...
#include <dlfcn.h>
char *errstr;
errstr = dlerror();
if (errstr != NULL)
    printf ("A dynamic linking error occurred: (%s)\n", errstr);
...
```

APPLICATION USAGE

Depending on the application environment with respect to asynchronous execution events, such as signals or other asynchronous computation sharing the address space, conforming applications should use a critical section to retrieve the error pointer and buffer.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`dlclose()`, `dlopen()`, `dlsym()`

The Base Definitions volume of POSIX.1?2017, `<dlfcn.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form

from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

DLERROR(3P)