



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 'perror.3p' command

\$ man perror.3p

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

perror ? write error messages to standard error

SYNOPSIS

```
#include <stdio.h>

void perror(const char *s);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The `perror()` function shall map the error number accessed through the symbol `errno` to a language-dependent error message, which shall be written to the standard error stream as follows:

- * First (if `s` is not a null pointer and the character pointed to by `s` is not the null byte), the string pointed to by `s` followed by a `<colon>` and a `<space>`.
- * Then an error message string followed by a `<newline>`.

The contents of the error message strings shall be the same as those returned by `strerror()` with argument `errno`.

The `perror()` function shall mark for update the last data modification and last file status change timestamps of the file associated with the standard error stream at some time between its successful completion and `exit()`, `abort()`, or the completion of `fflush()` or `fclose()` on `stderr`.

The `perror()` function shall not change the orientation of the standard error stream.

On error, `perror()` shall set the error indicator for the stream to which `stderr` points, and shall set `errno` to indicate the error.

Since no value is returned, an application wishing to check for error situations should call `clearerr(stderr)` before calling `perror()`, then if `ferror(stderr)` returns non-zero, the value of `errno` indicates which error occurred.

RETURN VALUE

The `perror()` function shall not return a value.

ERRORS

Refer to `fputc()`.

The following sections are informative.

EXAMPLES

Printing an Error Message for a Function

The following example replaces `bufptr` with a buffer that is the necessary size. If an error occurs, the `perror()` function prints a message and the program exits.

```
#include <stdio.h>
#include <stdlib.h>
...
char *bufptr;
size_t szbuf;
...
if ((bufptr = malloc(szbuf)) == NULL) {
    perror("malloc"); exit(2);
}
```

}

...

APPLICATION USAGE

Application writers may prefer to use alternative interfaces instead of `perror()`, such as `strerror_r()` in combination with `fprintf()`.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`fprintf()`, `fputc()`, `psiginfo()`, `strerror()`

The Base Definitions volume of POSIX.1?2017, `<stdio.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

PERROR(3P)