

NAME

clearerr, feof, ferror – check and reset stream status

SYNOPSIS

```
#include <stdio.h>

void clearerr(FILE *stream);
int feof(FILE *stream);
int ferror(FILE *stream);
```

DESCRIPTION

The function **clearerr()** clears the end-of-file and error indicators for the stream pointed to by *stream*. The function **feof()** tests the end-of-file indicator for the stream pointed to by *stream*, returning nonzero if it is set. The end-of-file indicator can be cleared only by the function **clearerr()**. The function **ferror()** tests the error indicator for the stream pointed to by *stream*, returning nonzero if it is set. The error indicator can be reset only by the **clearerr()** function. For nonlocking counterparts, see **unlocked_stdio(3)**.

RETURN VALUE

The **feof()** function returns nonzero if the end-of-file indicator is set for *stream*; otherwise, it returns zero. The **ferror()** function returns nonzero if the error indicator is set for *stream*; otherwise, it returns zero.

ERRORS

These functions should not fail and do not set *errno*.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
clearerr() , feof() , ferror()	Thread safety	MT-Safe

CONFORMING TO

The functions **clearerr()**, **feof()**, and **ferror()** conform to C89, C99, POSIX.1-2001, and POSIX.1-2008.

NOTES

POSIX.1-2008 specifies that these functions shall not change the value of *errno* if *stream* is valid.

SEE ALSO

open(2), **fdopen(3)**, **fileno(3)**, **stdio(3)**, **unlocked_stdio(3)**

COLOPHON

This page is part of release 5.13 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.

NAME

fopen, fdopen, fileno – stream open functions

SYNOPSIS

```
#include <stdio.h>

FILE *fopen(const char *path, const char *mode);
FILE *fdopen(int fd, const char *mode);
int fileno(FILE *stream);
int fclose(FILE *stream);
```

DESCRIPTION

The **fopen** function opens the file whose name is the string pointed to by *path* and associates a stream with it. The argument *mode* points to a string beginning with one of the following sequences (Additional characters may follow these sequences.):

- r** Open text file for reading. The stream is positioned at the beginning of the file.
- r+** Open for reading and writing. The stream is positioned at the beginning of the file.
- w** Truncate file to zero length or create text file for writing. The stream is positioned at the beginning of the file.
- w+** Open for reading and writing. The file is created if it does not exist, otherwise it is truncated. The stream is positioned at the beginning of the file.
- a** Open for appending (writing at end of file). The file is created if it does not exist. The stream is positioned at the end of the file.
- a+** Open for reading and appending (writing at end of file). The file is created if it does not exist. The stream is positioned at the end of the file.

The **fdopen** function associates a stream with the existing file descriptor, *fdes*. The *mode* of the stream (one of the values "r", "r+", "w", "w+", "a", "a+") must be compatible with the mode of the file descriptor. The file position indicator of the new stream is set to that belonging to *fdes*, and the error and end-of-file indicators are cleared. Modes "w" or "w+" do not cause truncation of the file. The file descriptor is not dup'ed, and will be closed when the stream created by **fdopen** is closed. The result of applying **fdopen** to a shared memory object is undefined.

The function **fileno()** examines the argument *stream* and returns its integer descriptor.

The **fclose()** function flushes the stream pointed to by *stream* (writing any buffered output data using **flush(3)**) and closes the underlying file descriptor.

RETURN VALUE

Upon successful completion **fopen**, **fdopen** and **freopen** return a **FILE** pointer. Otherwise, **NULL** is returned and the global variable *errno* is set to indicate the error. Upon successful completion of **fclose**, 0 is returned. Otherwise, **EOF** is returned and *errno* is set to indicate the error.

ERRORS

EINVAL
The *mode* provided to **fopen**, **fdopen**, or **freopen** was invalid.

EBADF

The file descriptor underlying *stream* passed to **fclose** is not valid.

The **fopen**, **fdopen** and **freopen** functions may also fail and set *errno* for any of the errors specified for the routine **malloc(3)**.

The **fopen** function may also fail and set *errno* for any of the errors specified for the routine **open(2)**.

The **fdopen** function may also fail and set *errno* for any of the errors specified for the routine **fcntl(2)**.

fread/fwrite(3)

fread/fwrite(3)

NAME

fread, fwrite – binary stream input/output

SYNOPSIS

```
#include <stdio.h>

size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
size_t fwrite(const void *ptr, size_t size, size_t nmemb,
              FILE *stream);
```

DESCRIPTION

The function **fread()** reads *nmemb* elements of data, each *size* bytes long, from the stream pointed to by *stream*, storing them at the location given by *ptr*.

The function **fwrite()** writes *nmemb* elements of data, each *size* bytes long, to the stream pointed to by *stream*, obtaining them from the location given by *ptr*.

For non-locking counterparts, see **unlocked_stdio(3)**.

RETURN VALUE

fread() and **fwrite()** return the number of items successfully read or written (i.e., not the number of characters). If an error occurs, or the end-of-file is reached, the return value is a short item count (or zero).

fread() does not distinguish between end-of-file and error, and callers must use **feof(3)** and **ferror(3)** to determine which occurred.

CONFORMING TO

C89, POSIX.1-2001.

SEE ALSO

read(2), **write(2)**, **feof(3)**, **ferror(3)**, **unlocked_stdio(3)**