



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 'fgetc.3p' command

\$ man fgetc.3p

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

fgetc ? get a byte from a stream

SYNOPSIS

```
#include <stdio.h>

int fgetc(FILE *stream);
```

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.

If the end-of-file indicator for the input stream pointed to by stream is not set and a next byte is present, the fgetc() function shall obtain the next byte as an unsigned char converted to an int, from the input stream pointed to by stream, and advance the associated file position indicator for the stream (if defined). Since fgetc() operates on bytes, reading a character consisting of multiple bytes (or ``a multi-byte character'') may require multiple calls to fgetc().

The `fgetc()` function may mark the last data access timestamp of the file associated with stream for update. The last data access timestamp shall be marked for update by the first successful execution of `fgetc()`, `fgets()`, `fread()`, `fscanf()`, `getc()`, `getchar()`, `getdelim()`, `getline()`, `gets()`, or `scanf()` using stream that returns data not supplied by a prior call to `ungetc()`.

RETURN VALUE

Upon successful completion, `fgetc()` shall return the next byte from the input stream pointed to by stream. If the end-of-file indicator for the stream is set, or if the stream is at end-of-file, the end-of-file indicator for the stream shall be set and `fgetc()` shall return EOF. If a read error occurs, the error indicator for the stream shall be set, `fgetc()` shall return EOF, and shall set `errno` to indicate the error.

ERRORS

The `fgetc()` function shall fail if data needs to be read and:

EAGAIN The `O_NONBLOCK` flag is set for the file descriptor underlying stream and the thread would be delayed in the `fgetc()` operation.

EBADF The file descriptor underlying stream is not a valid file descriptor open for reading.

EINTR The read operation was terminated due to the receipt of a signal, and no data was transferred.

EIO A physical I/O error has occurred, or the process is in a background process group attempting to read from its controlling terminal, and either the calling thread is blocking `SIGTTIN` or the process is ignoring `SIGTTIN` or the process group of the process is orphaned. This error may also be generated for implementation-defined reasons.

E_OVERFLOW

The file is a regular file and an attempt was made to read at or beyond the offset maximum associated with the corresponding stream.

The `fgetc()` function may fail if:

ENOMEM Insufficient storage space is available.

ENXIO A request was made of a nonexistent device, or the request was outside the capabilities of the device.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

If the integer value returned by `fgetc()` is stored into a variable of type `char` and then compared against the integer constant `EOF`, the comparison may never succeed, because sign-extension of a variable of type `char` on widening to integer is implementation-defined.

The `ferror()` or `feof()` functions must be used to distinguish between an error condition and an end-of-file condition.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5, Standard I/O Streams, `feof()`, `ferror()`, `fgets()`, `fread()`, `fscanf()`, `getchar()`, `getc()`, `gets()`, `ungetc()`

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.ker?>

