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 'fputc.3p' command

$ man fputc.3p

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

fputc ? put a byte on a stream

SYNOPSIS

#include <stdio.h>

int fputc(int c, 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.

The fputc() function shall write the byte specified by c (converted to an unsigned char) to the output stream pointed to by stream, at the po? sition indicated by the associated file-position indicator for the stream (if defined), and shall advance the indicator appropriately. If the file cannot support positioning requests, or if the stream was opened with append mode, the byte shall be appended to the output stream.

The last data modification and last file status change timestamps of the file shall be marked for update between the successful execution of fputc() and the next successful completion of a call to fflush() or fclose() on the same stream or a call to exit() or abort().

RETURN VALUE

Upon successful completion, fputc() shall return the value it has writ? ten. Otherwise, it shall return EOF, the error indicator for the stream shall be set, and errno shall be set to indicate the error.

ERRORS

The fputc() function shall fail if either the stream is unbuffered or the stream's buffer needs to be flushed, and:

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

EBADF The file descriptor underlying stream is not a valid file de? scriptor open for writing.

EFBIG An attempt was made to write to a file that exceeds the maximum file size.

EFBIG An attempt was made to write to a file that exceeds the file size limit of the process.

EFBIG The file is a regular file and an attempt was made to write at or beyond the offset maximum.

EINTR The write operation was terminated due to the receipt of a sig? nal, and no data was transferred.

EIO A physical I/O error has occurred, or the process is a member of a background process group attempting to write to its control? ling terminal, TOSTOP is set, the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned. This error may also be re? turned under implementation-defined conditions.

ENOSPC There was no free space remaining on the device containing the file.

EPIPE An attempt is made to write to a pipe or FIFO that is not open for reading by any process. A SIGPIPE signal shall also be sent

to the thread.

The fputc() 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

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5, Standard I/O Streams, ferror(), fopen(), getrlimit(), putc(), puts(), setbuf(), ulimit()

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

COPYRIGHT