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 'fdopen.3p' command

$ man fdopen.3p

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

fdopen ? associate a stream with a file descriptor

SYNOPSIS

#include <stdio.h>

FILE *fdopen(int fildes, const char *mode);

DESCRIPTION

The fdopen() function shall associate a stream with a file descriptor.

The mode argument is a character string having one of the following values:

r or rb Open a file for reading.

w or wb Open a file for writing.

a or ab Open a file for writing at end-of-file.

r+ or rb+ or r+b

Open a file for update (reading and writing).

w+ or wb+ or w+b

Open a file for update (reading and writing).

a+ or ab+ or a+b

Open a file for update (reading and writing) at end-of-file.

The meaning of these flags is exactly as specified in fopen(), except that modes beginning with w shall not cause truncation of the file.

Additional values for the mode argument may be supported by an imple? mentation.

The application shall ensure that the mode of the stream as expressed by the mode argument is allowed by the file access mode of the open file description to which fildes refers. The file position indicator associated with the new stream is set to the position indicated by the file offset associated with the file descriptor.

The error and end-of-file indicators for the stream shall be cleared.

The fdopen() function may cause the last data access timestamp of the underlying file to be marked for update.

If fildes refers to a shared memory object, the result of the fdopen() function is unspecified.

If fildes refers to a typed memory object, the result of the fdopen() function is unspecified.

The fdopen() function shall preserve the offset maximum previously set for the open file description corresponding to fildes.

RETURN VALUE

Upon successful completion, fdopen() shall return a pointer to a stream; otherwise, a null pointer shall be returned and errno set to indicate the error.

ERRORS

The fdopen() function shall fail if:

EMFILE {STREAM_MAX} streams are currently open in the calling process.

The fdopen() function may fail if:

EBADF The fildes argument is not a valid file descriptor.

EINVAL The mode argument is not a valid mode.

EMFILE {FOPEN_MAX} streams are currently open in the calling process.

ENOMEM Insufficient space to allocate a buffer.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

File descriptors are obtained from calls like open(), dup(), creat(), or pipe(), which open files but do not return streams.

RATIONALE

The file descriptor may have been obtained from open(), creat(), pipe(), dup(), fcntl(), or socket(); inherited through fork(), posix_spawn(), or exec; or perhaps obtained by other means.

The meanings of the mode arguments of fdopen() and fopen() differ. With fdopen(), open for write (w or w+) does not truncate, and append (a or a+) cannot create for writing. The mode argument formats that include a b are allowed for consistency with the ISO C standard function fopen(). The b has no effect on the resulting stream. Although not explicitly required by this volume of POSIX.1?2017, a good implementation of ap? pend (a) mode would cause the O_APPEND flag to be set.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5.1, Interaction of File Descriptors and Standard I/O Streams, fclose(), fmemopen(), fopen(), open(), open_memstream(), posix_spawn(), socket()

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

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?nel.org/doc/man-pages/reporting_bugs.html .