NAME

fopen, freopen, fdopen – open a stream

SYNOPSIS

#include <stdio.h>

FILE ∗**fopen(const char** ∗*filename*, **const char** ∗*type*);

FILE ∗**freopen(const char** ∗*filename*, **const char** ∗*type*, **FILE** ∗*stream*);

FILE ∗**fdopen(int** *fildes*, **const char** ∗*type*);

MT-LEVEL

MT-Safe

DESCRIPTION

fopen( ) opens the file named by *filename* and associates a *stream* with it. **fopen( )** returns a pointer to the **FILE** structure associated with the *stream*.

filename points to a character string that contains the name of the file to be opened.

type is a character string beginning with one of the following sequences:

"r" or **"rb"**	open for reading
"w" or **"wb"**	truncate to zero length or create for writing
"a" or **"ab"**	append; open for writing at end of file, or create for writing
"r+", **"r+b"** or **"rb+"**	open for update (reading and writing)
"w+", **"w+b"** or **"wb+"**	truncate or create for update
"a+", **"a+b"** or **"ab+"**	append; open or create for update at end-of-file

The "**b**" is ignored in the above *type*s. The "**b**" exists to distinguish binary files from text files. However, there is no distinction between these types of files on a UNIX system.

freopen( ) substitutes the named file in place of the open *stream*. A flush is first attempted, and then the original *stream* is closed, regardless of whether the open ultimately succeeds. Failure to flush or close *stream* successfully is ignored. **freopen( )** returns a pointer to the **FILE** structure associated with *stream*.

freopen( ) is typically used to attach the preopened *streams* associated with **stdin**, **stdout**, and **stderr** to other files. **stderr** is by default unbuffered, but the use of **freopen( )** will cause it to become buffered or line-buffered.

fdopen( ) associates a *stream* with a file descriptor. File descriptors are obtained from **open**(2), **dup**(2), **creat**(2), or **pipe**(2), which open files but do not return pointers to a **FILE** structure *stream*. Streams are necessary input for almost all of the Section 3S library routines. The *type* of *stream* must agree with the mode of the open file. The file position indicator associated with *stream* is set to the position indicated by the file offset associated with *fildes*.

When a file is opened for update, both input and output may be done on the resulting *stream*. However, output may not be directly followed by input without an intervening **fflush( )**, **fseek( )**, **fsetpos( )**, or **rewind( )**, and input may not be directly followed by output without an intervening **fseek( )**, **fsetpos( )**, or **rewind( )**, or an input operation that encounters end-of-file.

When a file is opened for append (that is, when *type* is **"a"**, **"ab"**, **"a+"**, or **"ab+"**), it is impossible to overwrite information already in the file. **fseek( )** may be used to reposition the file pointer to any position in the file, but when output is written to the file, the current file pointer is disregarded. All output is written at the end of the file and causes the file pointer to be repositioned at the end of the output. If two separate processes open the same file for append, each process may write freely to the file without fear of destroying output being written by the other. The output from the two processes will be intermixed in the file in the order in which it is written.

When opened, a *stream* is fully buffered if and only if it can be determined not to refer to an interactive device. The error and end-of-file indicators are cleared for the *stream*.

RETURN VALUES

The functions **fopen( )** and **freopen( )** return a null pointer if *path* cannot be accessed, or if *type* is invalid, or if the file cannot be opened.

The function **fdopen( )** returns a null pointer if *fildes* is not an open file descriptor, or if *type* is invalid, or if the file cannot be opened.

The functions **fopen( )** or **fdopen( )** may fail and not set **errno** if there are no free **stdio** streams.

File descriptors used by **fdopen( )** must be less than 255.

SEE ALSO

close(2), **creat**(2), **dup**(2), **open**(2), **pipe**(2), **write**(2), **fclose**(3S), **fseek**(3S), **setbuf**(3S), **stdio**(3S)