NAME

fopen, freopen – open a stream

SYNOPSIS

/usr/ucb/cc [*flag* . . .] *file* . . .

#include <stdio.h>

FILE ∗**fopen(***file***,** *mode***)**
const char ∗*file***,** ∗*mode***;**

FILE ∗**freopen(***file, mode, iop***)**
const char ∗*file,* ∗*mode***;**
register FILE ∗*iop***;**

DESCRIPTION

fopen( ) opens the file named by *file* and associates a stream with it. If the open succeeds, **fopen( )** returns a pointer to be used to identify the stream in subsequent operations.

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

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

- **r** open for reading
- **w** truncate or create for writing
- **a** append: open for writing at end of file, or create for writing
- **r+** open for update (reading and writing)
- **w+** truncate or create for update
- **a+** append; open or create for update at EOF

freopen( ) opens the file named by *file* and associates the stream pointed to by *iop* with it. The *mode* argument is used just as in **fopen( )**. The original stream is closed, regardless of whether the open ultimately succeeds. If the open succeeds, **freopen( )** returns the original value of *iop*.

freopen( ) is typically used to attach the preopened streams associated with **stdin**, **stdout**, and **stderr** to other files.

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 **fseek**(3S) or **rewind**(3S), and input may not be directly followed by output without an intervening **fseek**(3S) or **rewind**(3S). An input operation which encounters EOF will fail.

RETURN VALUES

fopen( ) and **freopen( )** return a **NULL** pointer on failure.

SEE ALSO

open(2), **fclose**(3S), **fopen**(3S), **freopen**(3S), **fseek**(3S), **malloc**(3C), **rewind**(3S)

NOTES

Use of these interfaces should be restricted to only applications written on BSD platforms. Use of these interfaces with any of the system libraries or in multi-thread applications is unsupported.

In order to support the same number of open files that the system does, **fopen( )** must allocate additional memory for data structures using **malloc**(3C) after 64 files have been opened. This confuses some programs which use their own memory allocators.

The interfaces of **fopen( )** and **freopen( )** differ from the Standard I/O Functions **fopen**(3S) and **freopen**(3S). The Standard I/O Functions distinguish binary from text files with an additional use of '**b**' as part of the *mode.* This enables portability of **fopen**(3S) and **freopen**(3S) beyond SunOS 4.*X* systems.