NAME
      stdio - standard input/output library functions

SYNOPSIS
      #include <stdio.h>

      FILE *_stdin_**;**
      FILE *_stdout_**;**
      FILE *_stderr_**;**

DESCRIPTION
      The standard I/O library provides a simple and efficient buffered stream I/O interface. Input and output is mapped into logical data streams and the physical I/O characteristics are concealed. The functions and macros are listed below; more information is available from the individual man pages.

      A stream is associated with an external file (which may be a physical device) by _opening_ a file, which may involve creating a new file. Creating an existing file causes its former contents to be discarded. If a file can support positioning requests (such as a disk file, as opposed to a terminal), then a _file position indicator_ associated with the stream is positioned at the start of the file (byte zero), unless the file is opened with append mode. If append mode is used, it is unspecified whether the position indicator will be placed at the start or the end of the file. The position indicator is maintained by subsequent reads, writes and positioning requests. All input occurs as if the characters were read by successive calls to the fgetc(3) function; all output takes place as if all characters were written by successive calls to the fputc(3) function.

      A file is disassociated from a stream by _closing_ the file. Output streams are flushed (any unwritten buffer contents are transferred to the host environment) before the stream is disassociated from the file. The value of a pointer to a _FILE_ object is indeterminate after a file is closed (garbage).

      A file may be subsequently reopened, by the same or another program execution, and its contents reclaimed or modified (if it can be repositioned at the start). If the main function returns to its original caller, or the exit(3) function is called, all open files are closed (hence all output streams are flushed) before program termination. Other methods of program termination, such as abort(3) do not bother about closing files properly.

      At program startup, three text streams are predefined and need not be opened explicitly: _standard input_ (for reading conventional input), _standard output_ (for writing conventional output), and _standard error_ (for writing diagnostic output). These streams are abbreviated _stdin_,_stdout_ and _stderr_. When opened, the standard error stream is not fully buffered; the standard input and output streams are fully buffered if and only if the streams do not refer to an interactive device.

      Output streams that refer to terminal devices are always line buffered by default; pending output to such streams is written automatically whenever an input stream that refers to a terminal device is read. In cases where a large amount of computation is done after printing part of a line on an output terminal, it is necessary to fflush(3) the standard output before going off and computing so that the output will appear.

      The _stdio_ library is a part of the library **libc** and routines are automatically loaded as needed by the compilers **cc(1)** and **pc(1)**. The SYNOPSIS sections of the following manual pages indicate which include files are to be used, what the compiler declaration for the function looks like and which external variables are of interest.

      The following are defined as macros; these names may not be reused without first removing their current definitions with **#undef**: **BUFSIZ**, **EOF**, **FILENAME_MAX**, **FOPEN_MAX**, **L_cuserid**, **L_ctermid**, **L_tmpnam**, **NULL**, **SEEK_END**, **SEEK_SET**, **SEEK_CUR**, **TMP_MAX**, **clearerr**, **feof**, **ferror**, **fileno**, **getc**, **getchar**, **putc**, **putchar**, **stderr**, **stdin**, **stdout**. Function versions of the macro functions **feof**, **ferror**, **clearerr**, **fileno**, **getc**, **getchar**, **putc**, and **putchar** exist and will be used if the macros definitions are explicitly removed.

List of functions

Function	Description
clearerr	check and reset stream status

fclose	close a stream
fdopen	stream open functions
feof	check and reset stream status
ferror	check and reset stream status
fflush	flush a stream
fgetc	get next character or word from input stream
fgetpos	reposition a stream
fgets	get a line from a stream
fileno	return the integer descriptor of the argument stream
fopen	stream open functions
fprintf	formatted output conversion
fpurge	flush a stream
fputc	output a character or word to a stream
fputs	output a line to a stream
fread	binary stream input/output
freopen	stream open functions
fscanf	input format conversion
fseek	reposition a stream
fsetpos	reposition a stream
ftell	reposition a stream
fwrite	binary stream input/output
getc	get next character or word from input stream
getchar	get next character or word from input stream
gets	get a line from a stream
getw	get next character or word from input stream
mktemp	make temporary filename (unique)
perror	system error messages
printf	formatted output conversion
putc	output a character or word to a stream
putchar	output a character or word to a stream
puts	output a line to a stream
putw	output a character or word to a stream
remove	remove directory entry
rewind	reposition a stream
scanf	input format conversion
setbuf	stream buffering operations
setbuffer	stream buffering operations
setlinebuf	stream buffering operations
setvbuf	stream buffering operations
sprintf	formatted output conversion
sscanf	input format conversion
strerror	system error messages
sys_errlist	system error messages
sys_nerr	system error messages
tempnam	temporary file routines
tmpfile	temporary file routines
tmpnam	temporary file routines
ungetc	un-get character from input stream
vfprintf	formatted output conversion
vfscanf	input format conversion
vprintf	formatted output conversion
vscanf	input format conversion
vsprintf	formatted output conversion

> **vsscanf**   input format conversion

CONFORMING TO

The *stdio* library conforms to C89.

SEE ALSO

close(2), open(2), read(2), write(2), stdout(3), unlocked_stdio(3)

COLOPHON

This page is part of release 4.10 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.