

**NAME**

stdio – standard input/output library functions

**LIBRARY**

Standard C library (*libc*, *-lc*)

**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 **cc**(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
----------	-------------

<b>clearerr(3)</b>	check and reset stream status
<b>fclose(3)</b>	close a stream
<b>fdopen(3)</b>	stream open functions
<b>feof(3)</b>	check and reset stream status
<b>ferror(3)</b>	check and reset stream status
<b>fflush(3)</b>	flush a stream
<b>fgetc(3)</b>	get next character or word from input stream
<b>fgetpos(3)</b>	reposition a stream
<b>fgets(3)</b>	get a line from a stream
<b>fileno(3)</b>	return the integer descriptor of the argument stream
<b>fopen(3)</b>	stream open functions
<b>fprintf(3)</b>	formatted output conversion
<b>fpurge(3)</b>	flush a stream
<b>fputc(3)</b>	output a character or word to a stream
<b>fputs(3)</b>	output a line to a stream
<b>fread(3)</b>	binary stream input/output
<b>freopen(3)</b>	stream open functions
<b>fscanf(3)</b>	input format conversion
<b>fseek(3)</b>	reposition a stream
<b>fsetpos(3)</b>	reposition a stream
<b>ftell(3)</b>	reposition a stream
<b>fwrite(3)</b>	binary stream input/output
<b>getc(3)</b>	get next character or word from input stream
<b>getchar(3)</b>	get next character or word from input stream
<b>gets(3)</b>	get a line from a stream
<b>getw(3)</b>	get next character or word from input stream
<b>mktemp(3)</b>	make temporary filename (unique)
<b>perror(3)</b>	system error messages
<b>printf(3)</b>	formatted output conversion
<b>putc(3)</b>	output a character or word to a stream
<b>putchar(3)</b>	output a character or word to a stream
<b>puts(3)</b>	output a line to a stream
<b>putw(3)</b>	output a character or word to a stream
<b>remove(3)</b>	remove directory entry
<b>rewind(3)</b>	reposition a stream
<b>scanf(3)</b>	input format conversion
<b>setbuf(3)</b>	stream buffering operations
<b>setbuffer(3)</b>	stream buffering operations
<b>setlinebuf(3)</b>	stream buffering operations
<b>setvbuf(3)</b>	stream buffering operations
<b>sprintf(3)</b>	formatted output conversion
<b>sscanf(3)</b>	input format conversion
<b>strerror(3)</b>	system error messages
<b>sys_errlist(3)</b>	system error messages
<b>sys_nerr(3)</b>	system error messages
<b>tempnam(3)</b>	temporary file routines
<b>tmpfile(3)</b>	temporary file routines
<b>tmpnam(3)</b>	temporary file routines
<b>ungetc(3)</b>	un-get character from input stream
<b>vfprintf(3)</b>	formatted output conversion
<b>vscanf(3)</b>	input format conversion
<b>vprintf(3)</b>	formatted output conversion
<b>vscanf(3)</b>	input format conversion

**vsprintf(3)**       formatted output conversion

**vscanf(3)**        input format conversion

## STANDARDS

The *stdio* library conforms to C99.

## SEE ALSO

**close(2), open(2), read(2), write(2), stdout(3), unlocked\_stdio(3)**