NAME

stdio − standard buffered input/output package

SYNOPSIS

#include <stdio.h>

FILE *stdin;
FILE *stdout;
FILE *stderr;

DESCRIPTION

The functions in the standard I/O library constitute a user-level buffering scheme. The in-line macros **getc** and **putc**(3) handle characters quickly. The higher level routines **gets**, **fgets**, **scanf**, **fscanf**, **fread**, **puts**, **fputs**, **printf**, **fprintf**, **fwrite** all use **getc** and **putc**; they can be freely intermixed.

A file with associated buffering is called a *stream*, and is declared to be a pointer to a defined type **FILE**. **Fopen**(3) creates certain descriptive data for a stream and returns a pointer to designate the stream in all further transactions. There are three normally open streams with constant pointers declared in the include file and associated with the standard open files:

stdin	standard input file
stdout	standard output file
stderr	standard error file

A constant 'pointer' **NULL** (0) designates no stream at all.

An integer constant **EOF** (−1) is returned upon end of file or error by integer functions that deal with streams.

Any routine that uses the standard input/output package must include the header file *<stdio.h>* of pertinent macro definitions. The functions and constants mentioned in the standard I/O manual pages are declared in the include file and need no further declaration. The constants, and the following 'functions' are implemented as macros; redeclaration of these names is perilous: **clearerr**, **getc**, **getchar**, **putc**, **putchar**, **feof**, **ferror**, **fileno**.

SEE ALSO

open(2), **close**(2), **read**(2), **write**(2), **fclose**(3), **ferror**(3), **fopen**(3), **fread**(3), **fseek**(3), **getc**(3), **gets**(3), **printf**(3), **putc**(3), **puts**(3), **scanf**(3), **setbuf**(3), **ungetc**(3).

DIAGNOSTICS

The value **EOF** is returned uniformly to indicate that a **FILE** pointer has not been initialized with **fopen**, input (output) has been attempted on an output (input) stream, or a **FILE** pointer designates corrupt or otherwise unintelligible **FILE** data.

For purposes of efficiency, this implementation of the standard library has been changed to line buffer output to a terminal by default and attempts to do this transparently by flushing the output whenever a **read**(2) from the standard input is necessary. This is almost always transparent, but may cause confusion or malfunctioning of programs which use standard i/o routines but use **read**(2) themselves to read from the standard input.

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.

BUGS

The standard buffered functions do not interact well with certain other library and system functions, especially **fork** and **abort**.

LIST OF FUNCTIONS

Name	**Appears on Page**	**Description**
clearerr	ferror(3)	stream status inquiries
fclose	fclose(3)	close or flush a stream
fdopen	fopen(3)	open a stream

feof	ferror(3)	stream status inquiries
ferror	ferror(3)	stream status inquiries
fflush	fclose(3)	close or flush a stream
fgetc	getc(3)	get character or word from stream
fgets	gets(3)	get a string from a stream
fileno	ferror(3)	stream status inquiries
fopen	fopen(3)	open a stream
fprintf	printf(3)	formatted output conversion
fputc	putc(3)	put character or word on a stream
fputs	puts(3)	put a string on a stream
fread	fread(3)	buffered binary input/output
freopen	fopen(3)	open a stream
fscanf	scanf(3)	formatted input conversion
fseek	fseek(3)	reposition a stream
ftell	fseek(3)	reposition a stream
fwrite	fread(3)	buffered binary input/output
getc	getc(3)	get character or word from stream
getchar	getc(3)	get character or word from stream
gets	gets(3)	get a string from a stream
getw	getc(3)	get character or word from stream
printf	printf(3)	formatted output conversion
putc	putc(3)	put character or word on a stream
putchar	putc(3)	put character or word on a stream
puts	puts(3)	put a string on a stream
putw	putc(3)	put character or word on a stream
rewind	fseek(3)	reposition a stream
scanf	scanf(3)	formatted input conversion
setbuf	setbuf(3)	assign buffering to a stream
setvbuf	setbuf(3)	assign buffering to a stream
snprintf	printf(3)	formatted output conversion
sprintf	printf(3)	formatted output conversion
sscanf	scanf(3)	formatted input conversion
ungetc	ungetc(3)	push character back into input stream
vfprintf	printf(3)	formatted output conversion
vfscanf	scanf(3)	formatted input conversion
vprintf	printf(3)	formatted output conversion
vscanf	scanf(3)	formatted input conversion
vsnprintf	printf(3)	formatted output conversion
vsprintf	printf(3)	formatted output conversion
vsscanf	scanf(3)	formatted input conversion