fgetc(3) - Linux man page

Name

fgetc, fgets, getc, getchar, gets, ungetc - input of characters and strings

Synopsis

```
#include <stdio.h>
int fgetc(FILE *stream);char *fgets(char *s, int size, FILE *stream);int
getc(FILE *stream);int getchar(void);char *gets(char *s);int ungetc(int
c, FILE *stream);
```

Description

fgetc() reads the next character from *stream* and returns it as an *unsigned* char cast to an *int*, or **EOF** on end of file or error.

getc() is equivalent to **fgetc**() except that it may be implemented as a macro which evaluates *stream* more than once.

getchar() is equivalent to **getc(***stdin***)**.

gets() reads a line from *stdin* into the buffer pointed to by *s* until either a terminating newline or **EOF**, which it replaces with a null byte (aq\0aq). No check for buffer overrun is performed (see BUGS below).

fgets() reads in at most one less than *size* characters from *stream* and stores them into the buffer pointed to by *s*. Reading stops after an **EOF** or a newline. If a newline is read, it is stored into the buffer. A terminating null byte (aq\0aq) is stored after the last character in the buffer.

ungetc() pushes *c* back to *stream*, cast to *unsigned char*, where it is available for subsequent read operations. Pushed-back characters will be returned in reverse order; only one pushback is guaranteed.

Calls to the functions described here can be mixed with each other and with calls to other input functions from the *stdio* library for the same input stream.

For nonlocking counterparts, see ***unlocked_stdio***(3).

Return Value

fgetc(), **getc**() and **getchar**() return the character read as an *unsigned* char cast to an *int* or **EOF** on end of file or error.

gets() and **fgets**() return *s* on success, and NULL on error or when end of file occurs while no characters have been read.

ungetc() returns *c* on success, or **EOF** on error.

Conforming to

C89, C99, POSIX.1-2001.

LSB deprecates **gets**(). POSIX.1-2008 marks **gets**() obsolescent. ISO C11 removes the specification of **gets**() from the C language, and since version 2.16, glibc header files don't expose the function declaration if the **_ISOC11_SOURCE** feature test macro is defined.

Bugs

Never use **gets**(). Because it is impossible to tell without knowing the data in advance how many characters **gets**() will read, and because **gets**() will continue to store characters past the end of the buffer, it is extremely dangerous to use. It has been used to break computer security. Use **fgets**() instead.

It is not advisable to mix calls to input functions from the *stdio* library with low-level calls to ***read****(2)* for the file descriptor associated with the input stream; the results will be undefined and very probably not what you want.

See Also

read*(2)*, ***write****(2)*, ***ferror****(3)*, ***fgetwc****(3)*, ***fgetws****(3)*, ***fopen****(3)*, ***fread****(3)*, ***fseek****(3)*, ***getline****(3)*, ***getwchar****(3)*, ***puts****(3)*, ***scanf****(3)*, ***ungetwc****(3)*, ***unlocked_stdio****(3)*, ***feature_test_macros****(7)*

Referenced By

explain(1), **explain**(3), **explain_fgetc**(3), **explain_fgetc_or_die**(3), **stdio**(3)