

**NAME**

`fgetc`, `fgets`, `getc`, `getchar`, `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);

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)**.

**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 ('\0') 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.

**fgets()** returns *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.

**ATTRIBUTES**

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
<b>fgetc()</b> , <b>fgets()</b> , <b>getc()</b> , <b>getchar()</b> , <b>ungetc()</b>	Thread safety	MT-Safe

**CONFORMING TO**

POSIX.1-2001, POSIX.1-2008, C89, C99.

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)**, **gets(3)**, **getwchar(3)**, **puts(3)**, **scanf(3)**, **ungetwc(3)**, **unlocked\_stdio(3)**, **feature\_test\_macros(7)**

**COLOPHON**

This page is part of release 5.02 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/>.