NAME

getc, getc_unlocked, getchar, getchar_unlocked, fgetc, getw – get character or word from a stream

SYNOPSIS

#include <stdio.h>

int getc(FILE **stream* **);**

int getc_unlocked(FILE **stream* **);**

int getchar(void);

int getchar_unlocked(void);

int fgetc(FILE **stream* **);**

int getw(FILE **stream* **);**

MT-LEVEL

See the **NOTES** section of this page.

DESCRIPTION

getc( ) returns the next character (that is, byte) from the named input *stream* (see **intro**(3)) as an **unsigned char** converted to an **int**. It also moves the file pointer, if defined, ahead one character in *stream*. **getchar( )** is defined as **getc(stdin)**. **getc( )** and **getchar( )** are macros.

getc_unlocked( ) and **getchar_unlocked( )** are respectively variants of **getc( )** and **getchar( )** that do not lock the stream. It is the caller's responsibility to acquire the stream lock before calling these functions and releasing the lock afterwards; see **flockfile**(3S) and **stdio**(3S).

fgetc( ) behaves like **getc( )**, but is a function rather than a macro. **fgetc( )** runs more slowly than **getc( ) ,** but it takes less space per invocation and its name can be passed as an argument to a function.

getw( ) returns the next word (that is, integer) from the named input *stream*. **getw( )** increments the associated file pointer, if defined, to point to the next word. The size of a word is the size of an integer and varies from machine to machine. **getw( )** assumes no special alignment in the file.

RETURN VALUES

These functions return the constant **EOF** at end-of-file or upon an error and set the **EOF** or error indicator of *stream*, respectively. Because is a valid integer, **ferror( )** should be used to detect **getw( )** errors.

SEE ALSO

intro(3), **fclose**(3S), **ferror**(3S), **flockfile**(3S), **fopen**(3S), **fread**(3S), **gets**(3S), **putc**(3S), **scanf**(3S), **stdio**(3S), **ungetc**(3S)

NOTES

If the integer value returned by **getc( )**, **getchar( )**, or **fgetc( )** is stored into a character variable and then compared against the integer constant **EOF**, the comparison may never succeed, because sign-extension of a character on widening to integer is implementation dependent.

The macro version of **getc( )** evaluates a *stream* argument more than once and may treat side effects incorrectly. In particular, **getc(*f++)** does not work sensibly. Use **fgetc( )** instead.

Because of possible differences in word length and byte ordering, files written using **putw( )** are implementation dependent, and may not be read using **getw( )** on a different processor.

Functions exist for all the above-defined macros. To get the function form, the macro name must be undefined (for example, **#undef getc**).

fgetc( ), getc( ), getchar( ), getw( ), and **ungetc( )** are MT-Safe in multi-thread applications. **getc_unlocked( )** and **getchar_unlocked( )** are unsafe in multi-thread applications.