

**NAME**

getc\_unlocked, getchar\_unlocked, putc\_unlocked, putchar\_unlocked – nonlocking stdio functions

**LIBRARY**

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

**SYNOPSIS**

```
#include <stdio.h>

int getc_unlocked(FILE *stream);
int getchar_unlocked(void);
int putc_unlocked(int c, FILE *stream);
int putchar_unlocked(int c);

void clearerr_unlocked(FILE *stream);
int feof_unlocked(FILE *stream);
int ferror_unlocked(FILE *stream);
int fileno_unlocked(FILE *stream);
int fflush_unlocked(FILE *stream);

int fgetc_unlocked(FILE *stream);
int fputc_unlocked(int c, FILE *stream);

size_t fread_unlocked(void ptr[restrict], size_t size, size_t n,
FILE *restrict stream);
size_t fwrite_unlocked(const void ptr[restrict], size_t size, size_t n,
FILE *restrict stream);

char *fgets_unlocked(char s[restrict], int n, FILE *restrict stream);
int fputs_unlocked(const char *restrict s, FILE *restrict stream);

#include <wchar.h>

wint_t getwc_unlocked(FILE *stream);
wint_t getwchar_unlocked(void);
wint_t fgetwc_unlocked(FILE *stream);

wint_t fputwc_unlocked(wchar_t wc, FILE *stream);
wint_t putwc_unlocked(wchar_t wc, FILE *stream);
wint_t putwchar_unlocked(wchar_t wc);

wchar_t *fgetws_unlocked(wchar_t ws[restrict], int n,
FILE *restrict stream);
int fputws_unlocked(const wchar_t *restrict ws,
FILE *restrict stream);
```

Feature Test Macro Requirements for glibc (see **feature\_test\_macros(7)**):

```
getc_unlocked(), getchar_unlocked(), putc_unlocked(), putchar_unlocked():
/* glibc >= 2.24: */ _POSIX_C_SOURCE >= 199309L
|| /* glibc <= 2.23: */ _POSIX_C_SOURCE
|| /* glibc <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE

clearerr_unlocked(), feof_unlocked(), ferror_unlocked(), fileno_unlocked(), fflush_unlocked(),
fgetc_unlocked(), fputc_unlocked(), fread_unlocked(), fwrite_unlocked():
/* glibc >= 2.19: */ _DEFAULT_SOURCE
|| /* glibc <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE

fgets_unlocked(), fputs_unlocked(), getwc_unlocked(), getwchar_unlocked(), fgetwc_unlocked(),
fputwc_unlocked(), putwchar_unlocked(), fgetws_unlocked(), fputws_unlocked():
_GNU_SOURCE
```

**DESCRIPTION**

Each of these functions has the same behavior as its counterpart without the "\_unlocked" suffix, except that they do not use locking (they do not set locks themselves, and do not test for the presence of locks set by others) and hence are thread-unsafe. See **flockfile(3)**.

**ATTRIBUTES**

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

Interface	Attribute	Value
<b>getc_unlocked()</b> , <b>putc_unlocked()</b> , <b>clearerr_unlocked()</b> , <b>fflush_unlocked()</b> , <b>fgetc_unlocked()</b> , <b>fputc_unlocked()</b> , <b>fread_unlocked()</b> , <b>fwrite_unlocked()</b> , <b>fgets_unlocked()</b> , <b>fputs_unlocked()</b> , <b>getwc_unlocked()</b> , <b>fgetwc_unlocked()</b> , <b>fputwc_unlocked()</b> , <b>putwc_unlocked()</b> , <b>fgetws_unlocked()</b> , <b>fputws_unlocked()</b>	Thread safety	MT-Safe race:stream
<b>getchar_unlocked()</b> , <b>getwchar_unlocked()</b>	Thread safety	MT-Unsafe race:stdin
<b>putchar_unlocked()</b> , <b>putwchar_unlocked()</b>	Thread safety	MT-Unsafe race:stdout
<b>feof_unlocked()</b> , <b>ferror_unlocked()</b> , <b>fileno_unlocked()</b>	Thread safety	MT-Safe

**STANDARDS**

The four functions **getc\_unlocked()**, **getchar\_unlocked()**, **putc\_unlocked()**, **putchar\_unlocked()** are in POSIX.1-2001 and POSIX.1-2008.

The nonstandard \*\_**unlocked()** variants occur on a few UNIX systems, and are available in recent glibc. They should probably not be used.

**SEE ALSO**

**flockfile(3)**, **stdio(3)**