



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'getwc_unlocked.3'

\$ man getwc_unlocked.3

UNLOCKED_STDIO(3) Linux Programmer's Manual UNLOCKED_STDIO(3)

NAME

getc_unlocked, getchar_unlocked, putc_unlocked, putchar_unlocked - non?

locking stdio functions

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, size_t size, size_t n,
```

```

        FILE *stream);
size_t fwrite_unlocked(const void *ptr, size_t size, size_t n,
        FILE *stream);
char *fgets_unlocked(char *s, int n, FILE *stream);
int fputs_unlocked(const char *s, FILE *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, int n, FILE *stream);
int fputws_unlocked(const wchar_t *ws, FILE *stream);

```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

```

getc_unlocked(), getchar_unlocked(), putc_unlocked(), putchar_un?
locked():

```

```

/* Since glibc 2.24: */ _POSIX_C_SOURCE >= 199309L

```

```

|| /* Glibc versions <= 2.23: */ _POSIX_C_SOURCE

```

```

|| /* Glibc versions <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE

```

```

clearerr_unlocked(), feof_unlocked(), ferror_unlocked(), fileno_un?
locked(), fflush_unlocked(), fgetc_unlocked(), fputc_unlocked(),
fread_unlocked(), fwrite_unlocked():

```

```

/* Glibc since 2.19: */ _DEFAULT_SOURCE

```

```

|| /* Glibc versions <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE

```

```

fgets_unlocked(), fputs_unlocked(), getwc_unlocked(), getwchar_un?
locked(), fgetwc_unlocked(), fputwc_unlocked(), putwchar_unlocked(),
fgetws_unlocked(), fputws_unlocked():

```

```

_GNU_SOURCE

```

DESCRIPTION

Each of these functions has the same behavior as its counterpart with? out 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 at?

tributes(7).

??

?Interface ? Attribute ? Value ?

??

?getc_unlocked(), ? Thread safety ? MT-Safe race:stream ?

?putc_unlocked(), ? ? ?

?clearerr_unlocked(), ? ? ?

?fflush_unlocked(), ? ? ?

?fgetc_unlocked(), ? ? ?

?fputc_unlocked(), ? ? ?

?fread_unlocked(), ? ? ?

?fwrite_unlocked(), ? ? ?

?fgets_unlocked(), ? ? ?

?fputs_unlocked(), ? ? ?

?getwc_unlocked(), ? ? ?

?fgetwc_unlocked(), ? ? ?

?fputwc_unlocked(), ? ? ?

?putwc_unlocked(), ? ? ?

?fgetws_unlocked(), ? ? ?

?fputws_unlocked() ? ? ?

??

?getchar_unlocked(), ? Thread safety ? MT-Unsafe race:stdin ?

?getwchar_unlocked() ? ? ?

??

?putchar_unlocked(), ? Thread safety ? MT-Unsafe race:stdout ?

?putwchar_unlocked() ? ? ?

??

?feof_unlocked(), ? Thread safety ? MT-Safe ?

?ferror_unlocked(), ? ? ?

?fileno_unlocked() ? ? ?

