



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

### ***Rocky Enterprise Linux 9.2 Manual Pages on command '\_\_fpending.3'***

**\$ man \_\_fpending.3**

STDIO\_EXT(3)                      Linux Programmer's Manual                      STDIO\_EXT(3)

#### NAME

\_\_fbufsize, \_\_flbf, \_\_fpending, \_\_fpurge, \_\_freadable, \_\_freading, \_\_fsetlocking, \_\_fwritable, \_\_fwriting, \_flushlbf - interfaces to stdio FILE structure

#### SYNOPSIS

```
#include <stdio.h>

#include <stdio_ext.h>

size_t __fbufsize(FILE *stream);

size_t __fpending(FILE *stream);

int __flbf(FILE *stream);

int __freadable(FILE *stream);

int __fwritable(FILE *stream);

int __freading(FILE *stream);

int __fwriting(FILE *stream);

int __fsetlocking(FILE *stream, int type);

void _flushlbf(void);

void __fpurge(FILE *stream);
```

#### DESCRIPTION

Solaris introduced routines to allow portable access to the internals of the FILE structure, and glibc also implemented these.

The \_\_fbufsize() function returns the size of the buffer currently used by the given stream.

The \_\_fpending() function returns the number of bytes in the output buffer. For wide-ori?

ented streams the unit is wide characters. This function is undefined on buffers in read? ing mode, or opened read-only.

The `__flbf()` function returns a nonzero value if the stream is line-buffered, and zero otherwise.

The `__freadable()` function returns a nonzero value if the stream allows reading, and zero otherwise.

The `__fwritable()` function returns a nonzero value if the stream allows writing, and zero otherwise.

The `__freanding()` function returns a nonzero value if the stream is read-only, or if the last operation on the stream was a read operation, and zero otherwise.

The `__fwriting()` function returns a nonzero value if the stream is write-only (or append-only), or if the last operation on the stream was a write operation, and zero otherwise.

The `__fsetlocking()` function can be used to select the desired type of locking on the stream. It returns the current type. The type argument can take the following three values:

#### FSETLOCKING\_INTERNAL

Perform implicit locking around every operation on the given stream (except for the \*\_unlocked ones). This is the default.

#### FSETLOCKING\_BYCALLER

The caller will take care of the locking (possibly using flockfile(3) in case there is more than one thread), and the stdio routines will not do locking until the state is reset to FSETLOCKING\_INTERNAL.

#### FSETLOCKING\_QUERY

Don't change the type of locking. (Only return it.)

The `_flushlbf()` function flushes all line-buffered streams. (Presumably so that output to a terminal is forced out, say before reading keyboard input.)

The `__fpurge()` function discards the contents of the stream's buffer.

#### ATTRIBUTES

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

??

?Interface                    ? Attribute    ? Value            ?

??

?\_\_fbufsize(), \_\_fpending(), ? Thread safety ? MT-Safe race:stream ?

?\_\_fpurge(), \_\_fsetlocking() ? ? ?

??

?\_\_flbf(), \_\_freadable(), ? Thread safety ? MT-Safe ?

?\_\_freading(), \_\_fwritable(), ? ? ?

?\_\_fwriting(), \_flushlbf() ? ? ?

??

SEE ALSO

flockfile(3), fpurge(3)

COLOPHON

This page is part of release 5.10 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/>.

2015-03-02

STDIO\_EXT(3)