



Red Hat Enterprise Linux Release 9.2 Manual Pages on '___fwritable.3' command

\$ man ___fwritable.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-oriented streams the unit is wide characters. This function is undefined on buffers in reading mode, or opened read-only.

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

The `__freaddable()` 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 `__freading()` 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 `flock(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`.

`LOCKING_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.

For an explanation of the terms used in this section, see at?

tributes(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)