



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

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

\$ man open_memstream.3

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

NAME

open_memstream, open_wmemstream - open a dynamic memory buffer stream

SYNOPSIS

```
#include <stdio.h>
```

```
FILE *open_memstream(char **ptr, size_t *sizeloc);
```

```
#include <wchar.h>
```

```
FILE *open_wmemstream(wchar_t **ptr, size_t *sizeloc);
```

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

open_memstream(), open_wmemstream():

Since glibc 2.10:

```
_POSIX_C_SOURCE >= 200809L
```

Before glibc 2.10:

```
_GNU_SOURCE
```

DESCRIPTION

The `open_memstream()` function opens a stream for writing to a memory buffer. The function dynamically allocates the buffer, and the buffer automatically grows as needed. Initially, the buffer has a size of zero. After closing the stream, the caller should `free(3)` this buffer.

The locations pointed to by `ptr` and `sizeloc` are used to report, respectively, the current location and the size of the buffer. The locations referred to by these pointers are updated each time the stream is flushed (`fflush(3)`) and when the stream is closed (`fclose(3)`). These

values remain valid only as long as the caller performs no further output on the stream. If further output is performed, then the stream must again be flushed before trying to access these values.

A null byte is maintained at the end of the buffer. This byte is not included in the size value stored at `size`.

The stream maintains the notion of a current position, which is initially zero (the start of the buffer). Each write operation implicitly adjusts the buffer position. The stream's buffer position can be explicitly changed with `fseek(3)` or `fseeko(3)`. Moving the buffer position past the end of the data already written fills the intervening space with null characters.

The `open_wmemstream()` is similar to `open_memstream()`, but operates on wide characters instead of bytes.

RETURN VALUE

Upon successful completion, `open_memstream()` and `open_wmemstream()` return a `FILE` pointer. Otherwise, `NULL` is returned and `errno` is set to indicate the error.

VERSIONS

`open_memstream()` was already available in `glibc 1.0.x`. `open_wmemstream()` is available since `glibc 2.4`.

ATTRIBUTES

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

??

?Interface ? Attribute ? Value ?

??

?`open_memstream()`, ? Thread safety ? MT-Safe ?

?`open_wmemstream` ? ? ?

??

CONFORMING TO

POSIX.1-2008. These functions are not specified in POSIX.1-2001, and are not widely available on other systems.

NOTES

There is no file descriptor associated with the file stream returned by these functions (i.e., `fileno(3)` will return an error if called on the returned stream).

BUGS

In glibc before version 2.7, seeking past the end of a stream created by `open_memstream()` does not enlarge the buffer; instead the `fseek(3)` call fails, returning -1.

EXAMPLES

See `fmemopen(3)`.

SEE ALSO

`fmemopen(3)`, `fopen(3)`, `setbuf(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/>.

GNU 2020-06-09 OPEN_MEMSTREAM(3)