

NAME

open_memstream, **open_wmemstream** - dynamic memory buffer stream open functions

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <stdio.h>

FILE *

open_memstream(*char **bufp*, *size_t *sizep*);

#include <wchar.h>

FILE *

open_wmemstream(*wchar_t **bufp*, *size_t *sizep*);

DESCRIPTION

The **open_memstream()** and **open_wmemstream()** functions create a write-only, seekable stream backed by a dynamically allocated memory buffer. The **open_memstream()** function creates a byte-oriented stream, while the **open_wmemstream()** function creates a wide-oriented stream.

Each stream maintains a current position and size. Initially, the position and size are set to zero. Each write begins at the current position and advances it the number of successfully written bytes for **open_memstream()** or wide characters for **open_wmemstream()**. If a write moves the current position beyond the length of the buffer, the length of the buffer is extended and a null character is appended to the buffer.

A stream's buffer always contains a null character at the end of the buffer that is not included in the current length.

If a stream's current position is moved beyond the current length via a seek operation and a write is performed, the characters between the current length and the current position are filled with null characters before the write is performed.

After a successful call to **fclose(3)** or **fflush(3)**, the pointer referenced by *bufp* will contain the start of the memory buffer and the variable referenced by *sizep* will contain the smaller of the current position and the current buffer length.

After a successful call to **fflush(3)**, the pointer referenced by *bufp* and the variable referenced by *sizep*

are only valid until the next write operation or a call to `fclose(3)`.

Once a stream is closed, the allocated buffer referenced by *bufp* should be released via a call to `free(3)` when it is no longer needed.

IMPLEMENTATION NOTES

Internally all I/O streams are effectively byte-oriented, so using wide-oriented operations to write to a stream opened via `open_wmemstream()` results in wide characters being expanded to a stream of multibyte characters in `stdio`'s internal buffers. These multibyte characters are then converted back to wide characters when written into the stream. As a result, the wide-oriented streams maintain an internal multibyte character conversion state that is cleared on any seek operation that changes the current position. This should have no effect as long as wide-oriented output operations are used on a wide-oriented stream.

RETURN VALUES

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

ERRORS

[EINVAL] The *bufp* or *sizep* argument was `NULL`.

[ENOMEM] Memory for the stream or buffer could not be allocated.

SEE ALSO

`fclose(3)`, `fflush(3)`, `fopen(3)`, `free(3)`, `fseek(3)`, `stdio(3)`, `sbuf(9)`

STANDARDS

The `open_memstream()` and `open_wmemstream()` functions conform to IEEE Std 1003.1-2008 ("POSIX.1").