

**NAME**

open\_memstream, open\_wmemstream – open a dynamic memory buffer stream

**LIBRARY**

Standard C library (*libc*, *-lc*)

**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 *sizeloc*.

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
<b>open_memstream()</b> , <b>open_wmemstream()</b>	Thread safety	MT-Safe

**STANDARDS**

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**

Before glibc 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)