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.3p' command

$ man open_memstream.3p

OPEN_MEMSTREAM(3P) POSIX Programmer's Manual OPEN_MEMSTREAM(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

open_memstream, open_wmemstream ? open a dynamic memory buffer stream

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 shall create an I/O stream associated with a dynamically allocated memory buffer. The stream shall be opened for writing and shall be seekable.

The stream associated with a call to open_memstream() shall be byte-oriented.

The stream associated with a call to open_wmemstream() shall be wide-oriented.

The stream shall maintain a current position in the allocated buffer and a current buffer length. The position shall be initially set to

zero (the start of the buffer). Each write to the stream shall start at the current position and move this position by the number of success? fully written bytes for open_memstream() or the number of successfully written wide characters for open_wmemstream(). The length shall be initially set to zero. If a write moves the position to a value larger than the current length, the current length shall be set to this posi? tion. In this case a null character for open_memstream() or a null wide character for open_wmemstream() shall be appended to the current buf? fer. For both functions the terminating null is not included in the calculation of the buffer length.

After a successful fflush() or fclose(), the pointer referenced by bufp shall contain the address of the buffer, and the variable pointed to by sizep shall contain the smaller of the current buffer length and the number of bytes for open_memstream(), or the number of wide characters for open_wmemstream(), between the beginning of the buffer and the cur? rent file position indicator.

After a successful fflush() the pointer referenced by bufp and the variable referenced by sizep remain valid only until the next write op? eration on the stream or a call to fclose().

After a successful fclose(), the pointer referenced by bufp can be passed to free().

RETURN VALUE

Upon successful completion, these functions shall return a pointer to the object controlling the stream. Otherwise, a null pointer shall be returned, and errno shall be set to indicate the error.

ERRORS

These functions shall fail if:

EMFILE {STREAM_MAX} streams are currently open in the calling process.

These functions may fail if:

EINVAL bufp or sizep are NULL.

EMFILE {FOPEN_MAX} streams are currently open in the calling process.

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

The following sections are informative.

EXAMPLES

```
#include <stdio.h>
#include <stdlib.h>
int
main (void)
{
   FILE *stream;
   char *buf;
   size_t len;
   off_t eob;
   stream = open_memstream (&buf, &len);
   if (stream == NULL)
      /* handle error */ ;
   fprintf (stream, "hello my world");
   fflush (stream);
   printf ("buf=%s, len=%zu\n", buf, len);
   eob = ftello(stream);
   fseeko (stream, 0, SEEK_SET);
   fprintf (stream, "good-bye");
   fseeko (stream, eob, SEEK_SET);
   fclose (stream);
   printf ("buf=%s, len=%zu\n", buf, len);
   free (buf);
   return 0;
}
```

This program produces the following output:

```
buf=hello my world, len=14
buf=good-bye world, len=14
```

APPLICATION USAGE

The buffer created by these functions should be freed by the applica? tion after closing the stream, by means of a call to free().

RATIONALE

These functions are similar to fmemopen() except that the memory is al?

ways allocated dynamically by the function, and the stream is opened only for output.

FUTURE DIRECTIONS

None.

SEE ALSO

fclose(), fdopen(), fflush(), fmemopen(), fopen(), free(), freopen()

The Base Definitions volume of POSIX.1?2017, <stdio.h>, <wchar.h>

COPYRIGHT