Asprintf

Generated by Doxygen 1.8.9.1

Tue Feb 24 2015 20:01:29

Contents

Chapter 1

Main Page

1.1 Asprintf manpage

NAME

asprintf, **vasprintf** — print to allocated string

SYNOPSIS

```
#include "asprintf.h"

int asprintf(char **strp, const char *format, ...);
int vasprintf(char **strp, const char *format, va_list ap);
```

MOTIVATION

From **asprintf**(3) in Linux Manual Pages:

> If memory allocation wasn't possible, or some other error occurs, these functions will return `-1`, and the contents of `strp` is undefined.

Therefore, when **asprintf** returns `-1`, no one knows whether `*strp` should be passed to **free** or not.

DESCRIPTION

Same as in Linux Manual Pages.

RETURN VALUE

When successful, same as in Linux Manual Pages.

If memory allocation wasn't possible, or some other error occurs, these functions will return `-1` and set `*strp` to be `NULL`, and no memory is allocated.

If an error occurs after memory is allocated successfully, these function calls **free** for the allocated memory.

SEE ALSO

malloc(3), **printf**(3), **stdarg**(3)

BUGS

The current implementation calls **vsnprintf** twice; first call is for getting size of memory allocation, and second call is for getting output string. When a return value of the second **vsnprintf** call is different from the first, the current implementation considers that some error occurred.

1.2 License

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

File Documentation

3.1 src/asprintf.c File Reference

Implementation of **asprintf** and **vasprintf**.

Macros

- #define mockable_vsnprintf1 vsnprintf

  *Function pointer for mocking first **vsnprintf** call.*
- #define mockable_vsnprintf2 vsnprintf

  *Function pointer for mocking second **vsnprintf** call.*
- #define mockable_malloc malloc

  *Function pointer for mocking **malloc** call.*
- #define mockable_free free

  *Function pointer for mocking **free** call.*

Functions

- int vasprintf (char **strp, const char *format, va_list ap)

  Print to allocated string.
- int asprintf (char **strp, const char *format,...)

  Print to allocated string.

3.1.1 Detailed Description

This is an implementation of **asprintf** and **vasprintf**.

The function **asprintf** and **vasprintf** are similar to **sprintf** and **vsprintf** respectively, except that they allocate memory to store output string, and return a pointer to it via the first parameter.

3.1.2 Macro Definition Documentation

3.1.2.1 #define mockable_free free

If macro `UNIT_TEST` is defined, this is a function pointer for mocking **free** call, whose type is as same as **free** and initial value is `free`. Otherwise, this is replaced with `free` simply.

3.1.2.2 #define mockable_malloc malloc

If macro `UNIT_TEST` is defined, this is a function pointer for mocking **`malloc`** call, whose type is as same as **`malloc`** and initial value is `malloc`. Otherwise, this is replaced with `malloc` simply.

3.1.2.3 #define mockable_vsnprintf1 vsnprintf

If macro `UNIT_TEST` is defined, this is a function pointer for mocking first **`vsnprintf`** call, whose type is as same as **`vsnprintf`** and initial value is `vsnprintf`. Otherwise, this is replaced with `vsnprintf` simply.

3.1.2.4 #define mockable_vsnprintf2 vsnprintf

If macro `UNIT_TEST` is defined, this is a function pointer for mocking second **`vsnprintf`** call, whose type is as same as **`vsnprintf`** and initial value is `vsnprintf`. Otherwise, this is replaced with `vsnprintf` simply.

3.1.3 Function Documentation

3.1.3.1 int asprintf (char ** *strp*, const char * *format*, ...)

This function is similar to **`sprintf`**, except that this function allocates memory to store output string, and returns a pointer to it via the first parameter.

Parameters

`out`	*strp*	Same as **`vasprintf`**.
`in`	*format*	Same as **`vasprintf`**.
`in`	*...*	Same as **`sprintf`** variable argument list.

Returns

Same as **`vasprintf`**.

3.1.3.2 int vasprintf (char ** *strp*, const char * *format*, va_list *ap*)

This function is similar to **`vsprintf`**, except that this function allocates memory to store output string, and returns a pointer to it via the first parameter.

Parameters

`out`	*strp*	If no error occurs, this function sets `*strp` to be a pointer to output string. Otherwise, this function sets `*strp` to be `NULL`. If an error occurs after memory is allocated successfully, this function calls **`free`** for the allocated memory.
`in`	*format*	Same as **`sprintf`** and/or **`vsprintf`** format string.
`in`	*ap*	Same as **`vsprintf`** variable argument list.

Returns

If no error occurs, this function returns the number of bytes of output string, not including the final NUL character. Otherwise, this function returns `-1`.

3.2 src/asprintf.h File Reference

Header file for **`asprintf`** and **`vasprintf`**.

Functions

- int vasprintf (char **strp, const char *format, va_list ap)

  Print to allocated string.
- int asprintf (char **strp, const char *format,...)

  Print to allocated string.

3.2.1 Detailed Description

This is a header file for **asprintf** and **vasprintf**.

The function **asprintf** and **vasprintf** are similar to **sprintf** and **vsprintf** respectively, except that they allocate memory to store output string, and return a pointer to it via the first parameter.

3.2.2 Function Documentation

3.2.2.1 int asprintf (char ** *strp*, const char * *format*, ...)

This function is similar to **sprintf**, except that this function allocates memory to store output string, and returns a pointer to it via the first parameter.

Parameters

out	*strp*	Same as **vasprintf**.
in	*format*	Same as **vasprintf**.
in	...	Same as **sprintf** variable argument list.

Returns

Same as **vasprintf**.

3.2.2.2 int vasprintf (char ** *strp*, const char * *format*, va_list *ap*)

This function is similar to **vsprintf**, except that this function allocates memory to store output string, and returns a pointer to it via the first parameter.

Parameters

out	*strp*	If no error occurs, this function sets `*strp` to be a pointer to output string. Otherwise, this function sets `*strp` to be `NULL`. If an error occurs after memory is allocated successfully, this function calls **free** for the allocated memory.
in	*format*	Same as **sprintf** and/or **vsprintf** format string.
in	*ap*	Same as **vsprintf** variable argument list.

Returns

If no error occurs, this function returns the number of bytes of output string, not including the final NUL character. Otherwise, this function returns `-1`.

3.3 test/test_1_normal_case.c File Reference

Tests for normal case.

Functions

- void test_normal_case_0_arg (void)

  Test for normal case, no argument in variable argument list.
- void test_normal_case_1_arg (void)

  Test for normal case, one argument in variable argument list.
- void test_normal_case_2_args (void)

  Test for normal case, two arguments in variable argument list.

3.3.1 Detailed Description

These are tests for normal case of **asprintf** and **vasprintf**.

3.3.2 Function Documentation

3.3.2.1 void test_normal_case_0_arg (void)

This is a test for normal case, no argument in variable argument list.

3.3.2.2 void test_normal_case_1_arg (void)

This is a test for normal case, one argument in variable argument list.

3.3.2.3 void test_normal_case_2_args (void)

This is a test for normal case, two arguments in variable argument list.

3.4 test/test_2_vsnprintf1_error_case.c File Reference

Test for first **vsnprintf** call error case.

Functions

- void cut_setup (void)

  *Setup for first **vsnprintf** call error case.*
- void cut_teardown (void)

  *Teardown for first **vsnprintf** call error case.*
- void test_vsnprintf1_error_case (void)

  *Test for first **vsnprintf** call error case.*

3.4.1 Detailed Description

This is test for first **vsnprintf** call error case of **asprintf** and **vasprintf**.

3.4.2 Function Documentation

3.4.2.1 void cut_setup (void)

This is a setup function for first **vsnprintf** call error case.

This function sets `mockable_vsnprintf1` to be mock function `mock_vsnprintf1`.

The mock function `mock_vsnprintf1` always returns `-1`.

3.4.2.2 void cut_teardown (void)

This is a teardown function for first **vsnprintf** call error case.

This function resets `mockable_vsnprintf1`.

3.4.2.3 void test_vsnprintf1_error_case (void)

This is a test for first **vsnprintf** call error case.

In this case, **asprintf** should return `-1` and set pointer to be `NULL`.

3.5 test/test_3_malloc_error_case.c File Reference

Test for **malloc** call error case.

Functions

- void cut_setup (void)
  *Setup for **malloc** call error case.*
- void cut_teardown (void)
  *Teardown for **malloc** call error case.*
- void test_malloc_error_case (void)
  *Test for **malloc** call error case.*

3.5.1 Detailed Description

This is test for **malloc** call error case of **asprintf** and **vasprintf**.

3.5.2 Function Documentation

3.5.2.1 void cut_setup (void)

This is a setup function for **malloc** call error case.

This function sets `mockable_malloc` to be mock function `mock_malloc`.

The mock function `mock_malloc` always returns `NULL`.

3.5.2.2 void cut_teardown (void)

This is a teardown function for **malloc** call error case.

This function resets `mockable_malloc`.

3.5.2.3 void test_malloc_error_case (void)

This is a test for **malloc** call error case.

In this case, **asprintf** should return `-1` and set pointer to be `NULL`.

3.6 test/test_4_vsnprintf2_error_case.c File Reference

Test for second **vsnprintf** call error case.

Functions

- void cut_setup (void)

  *Setup for second **vsnprintf** call error case.*
- void cut_teardown (void)

  *Teardown for second **vsnprintf** call error case.*
- void test_vsnprintf2_error_case (void)

  *Test for second **vsnprintf** call error case.*

3.6.1 Detailed Description

This is test for second **vsnprintf** call error case of **asprintf** and **vasprintf**.

3.6.2 Function Documentation

3.6.2.1 void cut_setup (void)

This is a setup function for second **vsnprintf** call error case.

This function sets `mockable_vsnprintf2` and `mockable_free` to be mock function `mock_vsnprintf2` and `mock_free` respectively, and resets `free_count`.

The mock function `mock_vsnprintf2` always returns its second parameter, that is not equal to return value of first **vsnprintf** call.

The mock function `mock_free` counts `free_count` up, and calls **free**.

3.6.2.2 void cut_teardown (void)

This is a teardown function for second **vsnprintf** call error case.

This function resets `mockable_vsnprintf2` and `mockable_free`.

3.6.2.3 void test_vsnprintf2_error_case (void)

This is a test for second **vsnprintf** call error case.

In this case, **vasprintf** should call **free** for allocated memory, **asprintf** should return `-1` and set pointer to be `NULL`.

3.7 test/test_5_vsnprintf1_parameter_check_case.c File Reference

Test for first **vsnprintf** call parameter check case.

Functions

- void cut_setup (void)

  *Setup for first **vsnprintf** call parameter check case.*
- void cut_teardown (void)

  *Teardown for first **vsnprintf** call parameter check case.*
- void test_vsnprintf1_parameter_check_case (void)

  *Test for first **vsnprintf** call parameter check case.*

3.7.1 Detailed Description

This is test for first **vsnprintf** call parameter check case of **asprintf** and **vasprintf**.

3.7.2 Function Documentation

3.7.2.1 void cut_setup (void)

This is a setup function for first **vsnprintf** call parameter check case.

This function sets `mockable_vsnprintf1` to be mock function `mock_vsnprintf1`, and resets all save area.

The mock function `mock_vsnprintf1` saves its parameters, and returns `-1`.

3.7.2.2 void cut_teardown (void)

This is a teardown function for first **vsnprintf** call parameter check case.

This function resets `mockable_vsnprintf1`.

3.7.2.3 void test_vsnprintf1_parameter_check_case (void)

This is a test for first **vsnprintf** call parameter check case.

3.8 test/test_6_malloc_parameter_check_case.c File Reference

Test for **malloc** call parameter check case.

Functions

- void cut_setup (void)

  *Setup for **malloc** call parameter check case.*
- void cut_teardown (void)

  *Teardown for **malloc** call parameter check case.*
- void test_malloc_parameter_check_case (void)

  *Test for **malloc** call parameter check case.*

3.8.1 Detailed Description

This is test for **malloc** call parameter check case of **asprintf** and **vasprintf**.

3.8.2 Function Documentation

3.8.2.1 void cut_setup (void)

This is a setup function for **malloc** call parameter check case.

This function sets `mockable_vsnprintf1` and `mockable_malloc` to be mock function `mock_vsnprintf1` and `mock_malloc` respectively, and resets all save area.

The mock function `mock_vsnprintf1` saves and returns **vsnprintf** return value.

The mock function `mock_malloc` saves its parameter, and returns `NULL`.

3.8.2.2 void cut_teardown (void)

This is a teardown function for **malloc** call parameter check case.

This function resets `mockable_vsnprintf1` and `mockable_malloc`.

3.8.2.3 void test_malloc_parameter_check_case (void)

This is a test for **malloc** call parameter check case.

3.9 test/test_7_vsnprintf2_parameter_check_case.c File Reference

Test for second **vsnprintf** call parameter check case.

Functions

- void cut_setup (void)

  *Setup for second **vsnprintf** call parameter check case.*
- void cut_teardown (void)

  *Teardown for second **vsnprintf** call parameter check case.*
- void test_vsnprintf2_parameter_check_case (void)

  *Test for second **vsnprintf** call parameter check case.*

3.9.1 Detailed Description

This is test for second **vsnprintf** call parameter check case of **asprintf** and **vasprintf**.

3.9.2 Function Documentation

3.9.2.1 void cut_setup (void)

This is a setup function for second **vsnprintf** call parameter check case.

This function sets `mockable_malloc` and `mockable_vsnprintf2` to be mock function `mock_malloc` and `mock_vsnprintf2` respectively, and resets all save area.

The mock function `mock_malloc` saves its parameter, calls **malloc**, saves and returns **malloc** return value.

The mock function `mock_vsnprintf2` saves its parameters, calls and returns **vsnprintf** return value.

3.9.2.2 void cut_teardown (void)

This is a teardown function for second **vsnprintf** call parameter check case.

This function resets `mockable_malloc` and `mockable_vsnprintf2`.

3.9.2.3 void test_vsnprintf2_parameter_check_case (void)

This is a test for second **vsnprintf** call parameter check case.

3.10 test/test_8_free_parameter_check_case.c File Reference

Test for **free** call parameter check case.

Functions

- void cut_setup (void)

  *Setup for **free** call parameter check case.*
- void cut_teardown (void)

  *Teardown for **free** call parameter check case.*
- void test_free_parameter_check_case (void)

  *Test for **free** call parameter check case.*

3.10.1 Detailed Description

This is test for **free** call parameter check case of **asprintf** and **vasprintf**.

3.10.2 Function Documentation

3.10.2.1 void cut_setup (void)

This is a setup function for **free** call parameter check case.

This function sets `mockable_vsnprintf2` and `mockable_free` to be mock function `mock_vsnprintf2` and `mock_free` respectively, and resets all save area.

The mock function `mock_vsnprintf2` saves its first parameter, and returns `-1`.

The mock function `mock_free` saves its parameter, and calls **free**.

3.10.2.2 void cut_teardown (void)

This is a teardown function for **free** call parameter check case.

This function resets `mockable_vsnprintf2` and `mockable_free`.

3.10.2.3 void test_free_parameter_check_case (void)

This is a test for **free** call parameter check case.

Index