



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

Linux Ubuntu 26.04 Manual Pages on command 'atanhl.3'

\$ man atanh.3

atanh(3) Library Functions Manual atanh(3)

NAME

atanh, atanhf, atanh_l - inverse hyperbolic tangent function

LIBRARY

Math library (libm, -lm)

SYNOPSIS

```
#include <math.h>

double atanh(double x);

float atanhf(float x);

long double atanhl(long double x);
```

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

atanh():

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
|| _XOPEN_SOURCE >= 500
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
|| /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

atanhf(), atanh_l():

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
|| /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

DESCRIPTION

These functions calculate the inverse hyperbolic tangent of x ; that is the value whose hyperbolic tangent is x .

RETURN VALUE

On success, these functions return the inverse hyperbolic tangent of x .

If x is a NaN, a NaN is returned.

If x is +0 (-0), +0 (-0) is returned.

If x is +1 or -1, a pole error occurs, and the functions return HUGE_VAL, HUGE_VALF, or HUGE_VALL, respectively, with the mathematically correct sign.

If the absolute value of x is greater than 1, a domain error occurs, and a NaN is returned.

ERRORS

See `math_error(7)` for information on how to determine whether an error has occurred when calling these functions.

The following errors can occur:

Domain error: x less than -1 or greater than $+1$

errno is set to EDOM. An invalid floating-point exception (FE_INVALID) is raised.

Pole error: x is $+1$ or -1

errno is set to ERANGE (but see BUGS). A divide-by-zero floating-point exception (FE_DIVBYZERO) is raised.

ATTRIBUTES

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

[illegible]

?	Interface	?	Attribute	?	Value	?
---	-----------	---	-----------	---	-------	---

[illegible]

? atanh(), atanhf(), atanhll() ? Thread safety ? MT-Safe ?

[illegible]

STANDARDS

C11, POSIX.1-2008.

HISTORY

C99, POSIX.1-2001.

The variant returning double also conforms to SVr4, 4.3BSD.

BUGS

In glibc 2.9 and earlier, when a pole error occurs, `errno` is set to `EDOM` instead of the POSIX-mandated `ERANGE`. Since glibc 2.10, glibc does the right thing.

SEE ALSO

acosh(3), asinh(3), catanh(3), cosh(3), sinh(3), tanh(3)

