

NAME

cosh, coshf, coshl – hyperbolic cosine function

LIBRARY

Math library (*libm*, *-lm*)

SYNOPSIS

```
#include <math.h>
```

```
double cosh(double x);
```

```
float coshf(float x);
```

```
long double coshl(long double x);
```

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

```
coshf(), coshl():
```

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

```
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
```

```
|| /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

DESCRIPTION

These functions return the hyperbolic cosine of x , which is defined mathematically as:

$$\cosh(x) = (\exp(x) + \exp(-x)) / 2$$

RETURN VALUE

On success, these functions return the hyperbolic cosine of x .

If x is a NaN, a NaN is returned.

If x is +0 or −0, 1 is returned.

If x is positive infinity or negative infinity, positive infinity is returned.

If the result overflows, a range error occurs, and the functions return +**HUGE_VAL**, +**HUGE_VALF**, or +**HUGE_VALL**, respectively.

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:

Range error: result overflow

errno is set to **ERANGE**. An overflow floating-point exception (**FE_OVERFLOW**) is raised.

ATTRIBUTES

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

Interface	Attribute	Value
cosh(), coshf(), coshl()	Thread safety	MT-Safe

STANDARDS

C99, POSIX.1-2001, POSIX.1-2008.

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

BUGS

In glibc 2.3.4 and earlier, an overflow floating-point (**FE_OVERFLOW**) exception is not raised when an overflow occurs.

SEE ALSO

acosh(3), **asinh(3)**, **atanh(3)**, **ccos(3)**, **sinh(3)**, **tanh(3)**