

NAME

`floor`, `floorf`, `floorl` - largest integral value not greater than argument

SYNOPSIS

```
#include <math.h>

double floor(double x);
float floorf(float x);
long double floorl(long double x);
```

Link with `-lm`.

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

```
floorf(), floorl():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600 || _ISOC99_SOURCE ||
    _POSIX_C_SOURCE >= 200112L;
    or cc -std=c99
```

DESCRIPTION

These functions return the largest integral value that is not greater than x .

For example, `floor(0.5)` is 0.0, and `floor(-0.5)` is -1.0.

RETURN VALUE

These functions return the floor of x .

If x is integral, +0, -0, NaN, or an infinity, x itself is returned.

ERRORS

No errors occur. POSIX.1-2001 documents a range error for overflows, but see NOTES.

ATTRIBUTES

Multithreading (see [pthreads\(7\)](#))

The `floor()`, `floorf()`, and `floorl()` functions are thread-safe.

CONFORMING TO

C99, POSIX.1-2001. The variant returning *double* also conforms to SVr4, 4.3BSD, C89.

NOTES

SUSv2 and POSIX.1-2001 contain text about overflow (which might set *errno* to **ERANGE**, or raise an **FE_OVERFLOW** exception). In practice, the result cannot overflow on any current machine, so this error-handling stuff is just nonsense. (More precisely, overflow can happen only when the maximum value of the exponent is smaller than the number of mantissa bits. For the IEEE-754 standard 32-bit and 64-bit floating-point numbers the maximum value of the exponent is 128 (respectively, 1024), and the number of mantissa bits is 24 (respectively, 53).)

SEE ALSO

[ceil\(3\)](#), [lrint\(3\)](#), [nearbyint\(3\)](#), [rint\(3\)](#), [round\(3\)](#), [trunc\(3\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.