NAME

fmod, fmodf, fmodl − floating-point remainder function

SYNOPSIS

#include <math.h>

double fmod(double *x*, **double** *y***);**
float fmodf(float *x*, **float** *y***);**
long double fmodl(long double *x*, **long double** *y***);**

Link with *−lm*.

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

fmodf(), **fmodl**():
_BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600 || _ISOC99_SOURCE ||
_POSIX_C_SOURCE >= 200112L;
or *cc -std=c99*

DESCRIPTION

The **fmod**() function computes the floating-point remainder of dividing *x* by *y*. The return value is *x* − *n* * *y*, where *n* is the quotient of *x* / *y*, rounded toward zero to an integer.

RETURN VALUE

On success, these functions return the value *x* − *n***y*, for some integer *n*, such that the returned value has the same sign as *x* and a magnitude less than the magnitude of *y*.

If *x* or *y* is a NaN, a NaN is returned.

If *x* is an infinity, a domain error occurs, and a NaN is returned.

If *y* is zero, a domain error occurs, and a NaN is returned.

If *x* is +0 (−0), and *y* is not zero, +0 (−0) 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* is an infinity
errno is set to **EDOM** (but see BUGS). An invalid floating-point exception (**FE_INVALID**) is raised.

Domain error: *y* is zero
errno is set to **EDOM**. An invalid floating-point exception (**FE_INVALID**) is raised.

CONFORMING TO

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

BUGS

Before version 2.10, the glibc implementation did not set *errno* to **EDOM** when a domain error occurred for an infinite *x*.

SEE ALSO

remainder(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/.