NAME

modf, modff, modfl - extract signed integral and fractional values from floating-point number

SYNOPSIS

#include <math.h>

double modf(double *x***, double ***iptr***);**
float modff(float *x***, float ***iptr***);**
long double modfl(long double *x***, long double ***iptr***);**

Link with *-lm*.

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

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

DESCRIPTION

The **modf**() function breaks the argument *x* into an integral part and a fractional part, each of which has the same sign as *x*. The integral part is stored in the location pointed to by *iptr*.

RETURN VALUE

The **modf**() function returns the fractional part of *x*.

If *x* is a NaN, a NaN is returned, and **iptr* is set to a NaN.

If *x* is positive infinity (negative infinity), +0 (-0) is returned, and **iptr* is set to positive infinity (negative infinity).

ERRORS

No errors occur.

ATTRIBUTES

Multithreading (see pthreads(7))

The **modf**(), **modff**(), and **modfl**() functions are thread-safe.

CONFORMING TO

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

SEE ALSO

frexp(3), ldexp(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/.