



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 'sincos.3'

\$ man sincos.3

sincos(3) Library Functions Manual sincos(3)

NAME

sincos, sincosf, sincosl - calculate sin and cos simultaneously

LIBRARY

Math library (libm, -lm)

SYNOPSIS

```
#define _GNU_SOURCE      /* See feature_test_macros(7) */
#include <math.h>

void sincos(double x, double *sin, double *cos);

void sincosf(float x, float *sin, float *cos);

void sincosl(long double x, long double *sin, long double *cos);
```

DESCRIPTION

Several applications need sine and cosine of the same angle x . These functions compute both at the same time, and store the results in $*sin$ and $*cos$. Using this function can be more efficient than two separate calls to $\sin(3)$ and $\cos(3)$.

If x is a NaN, a NaN is returned in $*sin$ and $*cos$.

If x is positive infinity or negative infinity, a domain error occurs, and a NaN is returned in $*sin$ and $*cos$.

RETURN VALUE

These functions return void.

ERRORS

See $\text{math_error}(7)$ 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.

ATTRIBUTES

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

[illegible]

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

[illegible]

? sincos(), sincosf(), sincosl()	? Thread safety ? MT-Safe ?
----------------------------------	-----------------------------

[illegible]

STANDARDS

GNU.

HISTORY

glibc 2.1.

NOTES

To see the performance advantage of `sincos()`, it may be necessary to disable `gcc(1)` built-in optimizations, using flags such as:

```
cc -O -lm -fno-builtin prog.c
```

BUGS

Before glibc 2.22, the glibc implementation did not set `errno` to `EDOM` when a domain error occurred.

SEE ALSO

 $\cos(3), \sin(3), \tan(3)$