Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tan.3p' command

$ man tan.3p

TAN(3P) POSIX Programmer's Manual TAN(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

tan, tanf, tanl ? tangent function

SYNOPSIS

```
#include <math.h>
double tan(double x);
float tanf(float x);
long double tanl(long double x);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

These functions shall compute the tangent of their argument x, measured in radians.

An application wishing to check for error situations should set errno to zero and call feclearexcept(FE_ALL_EXCEPT) before calling these functions. On return, if errno is non-zero or fetestexcept(FE_INVALID |

FE_DIVBYZERO | FE_OVERFLOW | FE_UNDERFLOW) is non-zero, an error has occurred.

RETURN VALUE

Upon successful completion, these functions shall return the tangent of x.

If the correct value would cause underflow, and is not representable, a range error may occur, and tan(), tanf(), and tanl() shall return 0.0, or (if IEC 60559 Floating-Point is not supported) an implementation-de? fined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

If x is NaN, a NaN shall be returned.

If x is ?0, x shall be returned.

If x is subnormal, a range error may occur and x should be returned.

If x is not returned, tan(), tanf(), and tanl() shall return an imple? mentation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

If x is ?Inf, a domain error shall occur, and either a NaN (if sup? ported), or an implementation-defined value shall be returned.

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

If the correct value would cause overflow, a range error shall occur and tan(), tanf(), and tanl() shall return ?HUGE_VAL, ?HUGE_VALF, and ?HUGE_VALL, respectively, with the same sign as the correct value of the function.

ERRORS

These functions shall fail if:

Domain Error

The value of x is ?Inf.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [EDOM]. If the in? teger expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the invalid floating-point exception shall

be raised.

Range Error The result overflows

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [ERANGE]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the overflow floating-point exception shall be raised.

These functions may fail if:

Range Error The result underflows, or the value of x is subnormal.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [ERANGE]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the underflow floating-point exception shall be raised.

The following sections are informative.

EXAMPLES

Taking the Tangent of a 45-Degree Angle

```
#include <math.h>
...
double radians = 45.0 * M_PI / 180;
double result;
...
result = tan (radians);
```

APPLICATION USAGE

There are no known floating-point representations such that for a nor? mal argument, tan(x) is either overflow or underflow.

These functions may lose accuracy when their argument is near a multi? ple of ?/2 or is far from 0.0.

On error, the expressions (math_errhandling & MATH_ERRNO) and (math_er? rhandling & MATH_ERREXCEPT) are independent of each other, but at least one of them must be non-zero.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

atan(), feclearexcept(), fetestexcept(), isnan()

The Base Definitions volume of POSIX.1?2017, Section 4.20, Treatment of Error Conditions for Mathematical Functions, <math.h>