

NAME

atan2, **atan2f**, **atan2l**, **carg**, **cargf**, **cargl** - arc tangent and complex phase angle functions

LIBRARY

Math Library (libm, -lm)

SYNOPSIS

#include <math.h>

double

atan2(*double* y, *double* x);

float

atan2f(*float* y, *float* x);

long double

atan2l(*long double* y, *long double* x);

#include <complex.h>

double

carg(*double complex* z);

float

cargf(*float complex* z);

long double

cargl(*long double complex* z);

DESCRIPTION

The **atan2**(), **atan2f**(), and **atan2l**() functions compute the principal value of the arc tangent of y/x , using the signs of both arguments to determine the quadrant of the return value.

The **carg**(), **cargf**(), and **cargl**() functions compute the complex argument (or phase angle) of z . The complex argument is the number θ such that $z = r * e^{i * \theta}$, where $r = \text{cabs}(z)$. The call **carg**(z) is equivalent to **atan2**(**cimag**(z), **creal**(z)), and similarly for **cargf**() and **cargl**().

RETURN VALUES

The **atan2**(), **atan2f**(), and **atan2l**() functions, if successful, return the arc tangent of y/x in the range $[-\pi, +\pi]$ radians. Here are some of the special cases:

atan2 (y, x) :=	atan (y/x)	if $x > 0$,
	$\text{sign}(y) * (\pi - \mathbf{atan}(y/x))$	if $x < 0$,
	0	if $x = y = 0$, or
	$\text{sign}(y) * \pi/2$	if $x = 0 \neq y$.

NOTES

The function **atan2**() defines "if $x > 0$," **atan2**(0, 0) = 0 despite that previously **atan2**(0, 0) may have generated an error message. The reasons for assigning a value to **atan2**(0, 0) are these:

1. Programs that test arguments to avoid computing **atan2**(0, 0) must be indifferent to its value. Programs that require it to be invalid are vulnerable to diverse reactions to that invalidity on diverse computer systems.
2. The **atan2**() function is used mostly to convert from rectangular (x,y) to polar (r,theta) coordinates that must satisfy $x = r * \cos \theta$ and $y = r * \sin \theta$. These equations are satisfied when (x=0,y=0) is mapped to (r=0,theta=0). In general, conversions to polar coordinates should be computed thus:

```
r      := hypot(x,y); ... := sqrt(x*x+y*y)
theta  := atan2(y,x).
```

3. The foregoing formulas need not be altered to cope in a reasonable way with signed zeros and infinities on a machine that conforms to IEEE 754; the versions of hypot(3) and **atan2**() provided for such a machine are designed to handle all cases. That is why **atan2**(+0, -0) = +pi for instance. In general the formulas above are equivalent to these:

```
r := sqrt(x*x+y*y); if r = 0 then x := copysign(1,x);
```

SEE ALSO

acos(3), asin(3), atan(3), cabs(3), cos(3), cosh(3), math(3), sin(3), sinh(3), tan(3), tanh(3)

STANDARDS

The **atan2**(), **atan2f**(), **atan2l**(), **carg**(), **cargf**(), and **cargl**() functions conform to ISO/IEC 9899:1999 ("ISO C99").