

A proposal to add sincos to the standard library

P0081R0

Paul Dreik <paul dot dreik at dreik dot se>

2015-09-25

Introduction

Sometimes one needs to calculate sine and cosine for the same argument x . This is made with one call to `std::sin(x)` and one to `std::cos(x)`. This paper proposes adding a new function `std::sincos(x)`, calculating both at the same time.

Problem

In case the argument x is a complicated expression, it is easy to introduce errors at the time of writing or during maintenance, erroneously calculating sine and cosine for different input.

Using a `sincos` function naturally expresses the intent of the programmer (“sin and cos should be calculated for the same argument”).

Discussion

Gnu libc provides the following c `sincos` functions [gnu]:

```
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);
```

Using pointer arguments for the results is c style and requires deciding if the pointers should be allowed to alias and/or be null. It is proposed that return by value is used instead.

Newer x86 cpus seem to have support for calculating sin and cos simultaneously [wikipedia] which can be more efficient[intel]. Having a `sincos` function may simplify exploiting hardware support without having to rely on compiler optimization.

Solution

It is proposed that a `std::sincos` function is added to the standard library, with overloads on all builtin floating point types.

Demonstration of intended usage:

```
#include <sincos>
#include <tuple>

double s, c;
void doit() {
    std::tie(s, c) = std::sincos(1.047);
}
```

Proposed wording

26.X sincos

26.X.1 Header sincos synopsis

```
#include <utility>
namespace std {
    pair<double, double> sincos(double x) noexcept;
    // overloads on other floating point types, see below
}
```

In addition to the double version, there should be overloads on all other builtin floating point types.

26.X.2 sincos functions

Preconditions: none

Returns: the sine and cosine of the input

Remark: The sincos function shall behave as if it had the following definition:

```
std::pair<double, double> std::sincos(double x) {
    return std::pair<double, double>(std::sin(x), std::cos(x));
}
// similarly for the overloaded versions
```

References

[gnu] Gnu libc manual: https://www.gnu.org/software/libc/manual/html_node/Trig-Functions.html

[intel] Intel® 64 and IA-32 Architectures Optimization Reference Manual, table C-20: <https://www-ssl.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html>

[wikipedia] X86 instruction listings

https://en.wikipedia.org/wiki/X86_instruction_listings#Added_with_80387