

Re: how long is double

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-01/0848.html

From: John Potter (jpotter_at_falcon.lhup.edu)

Date: 01/07/04

Date: 7 Jan 2004 13:04:01 -0500

On 7 Jan 2004 03:50:05 -0500, Martijn Lievaart
<m@remove.this.part.rtij.nl> wrote:

> On Tue, 06 Jan 2004 06:35:26 -0500, kanze wrote:

>> *I think that this was Francis' point. According to the standard, $a+b+c$
>> is $(a+b)+c$. The compiler is free to rearrange this any way it pleases,
>> as long as the results are the same as if it had done $(a+b)+c$. On most
>> machines, with integer arithmetic, there is no problem. On no machine
>> that I know of, however, can the compiler legally rearrange floating
>> point, unless it absolutely knows the values involved.*

>> *There was quite a lot of discussion about this when the C standard was
>> first being written. K&R explicitly allowed rearrangement, even when it
>> would result in different results. In the end, the C standard decided
>> not to allow this.*

> *Then this seems a place where C and C++ differ, see the answer and quote
> from the C++ standard from Ron Natalie.*

> *Anyone who can confirm this?*

His quote is about order of evaluation. It relates to $a*b+c*d$ where either of $a*b$ or $c*d$ may be evaluated first. In the code under discussion, $a+b+c$, it is $(a+b)+c$. C may be evaluated before $a+b$ but changing it to $a+(b+c)$ or $(a+c)+b$ is only allowed by the as-if rule. The non-normative note in 1.9/15 amplifies this. It seems that the C++ rules are the same as the C rules.

I have no idea how Ron based his other post that claimed an expression could be reordered giving different results. There is no support in either of the cited verses.

John

[See <http://www.gotw.ca/resources/clcm.htm> for info about]
[comp.lang.c++.moderated. First time posters: Do this!]