Subject: Re: Numbers from nowhere?
Posted by elwood on Mon, 25 Feb 2008 17:57:14 GMT
View Forum Message <> Reply to Message

Thanks,
very succinct and clear explanation.
Fun with Binary and Computer Precision is what I summarize
the topic as ;-)

-Elisha

On Feb 21, 7:06 am, Sven Utcke <utcke+n...@informatik.uni-hamburg.de>
wrote:
> David Fanning <n...@dfanning.com> writes:
>> elwood writes:
>
>>> But my question is more pointed: if you assign x=3.3 and you know
>>> apriori that the floating point data type will not have enough bits
>>> to store this number precisely, why does "print" show this number
>>> as 3.3?
>
>> I presume it is because whatever number *is* stored, when rounded to
>> the 7-8 significant figures a float can accurately represent, comes
>> out to 3.300000.
>
> What number _is_ stored, actually? Assuming we are talking ieee, we
> have one bit for the sign, 8 for the exponent, and 23 for the
> mantissa. So what is 3.3?
>
> 3 = 11 = 1.1 * 2^1
> 0.3 = 0.0100110011001100110011001100110011001100...
> which we see from
>
> 0.3*2 = _0_.6
> 0.6*2 = _1_.2
> 0.2*2 = _0_.4
> 0.4*2 = _0_.8
> 0.8*2 = _1_.6
> 0.6*2 = ...
>
> so we get, combined,
>
> 3.3 = 1.10100110011001100110011 * 2^1
>
> or
>
> S | Exp + 127 | Mantissa without leading 1
> 0 | 1000000 | 1010011 00110011 00110011

>
> which, if we recombine it, turns out to be 3.2999999523162841796875
>
> We can actually see this in IDL too:
>
> IDL> print, byte(3.3,0,4)
> 51 51 83 64
> Which, if we rewrite it appropriately, turns out to be:
>
> 01000000 01010011 00110011 T
> which, recombined differently, is the above number :-)
>
> Sven
> --
> ___ _ _____ ___ Dr.-Ing. Sven Utcke ___ ___ _____ __
