

Re: question about random generator

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-07/msg02642.html

- *From:* "Antonio" <anconor@xxxxxxxxx>
 - *Date:* 26 Jul 2005 07:34:35 -0700
-

Robert Gamble wrote:

> Antonio wrote:
>> pete wrote:
>>> Does "pseudo-random" mean the same thing as "uniformly distributed" ?
>>
>> Completely OT, but anyway... No, pseudo-random means that it looks like
>> it's random but it really isn't. There is no way to generate trully
>> random numbers with a computer, everything you do is deterministic, but
>> you can generate sequences that look like they're random but that
>> aren't. Hence the term `_pseudo_`-random.
>
> Yes, we know, you didn't say anything that wasn't completely obvious.
> The C Standard specifies that `rand()` generates pseudo-random numbers,
> the questions is whether a conforming implementation could generate a
> series of normal distributed numbers via the `rand()` function or if the
> term pseudo-random implies that the numbers must be generated with a
> uniform distribution. I was wondering the same thing myself, I think
> the intention is that the numbers be uniform but that may be debatable.

It may be completely obvious to you, and to many people (including myself), but it doesn't seem to be obvious to "pete", since he asked. What I was trying to explain is that pseudo-random does not imply anything about the distribution of the numbers. You may get pseudo-random numbers that look like a uniform distribution, or pseudo-random numbers that look like a poisson distribution, or a gaussian distribution, or anything you want. In fact if you are able to generate pseudo-random numbers with any distribution, you can operate with them to obtain any other distribution you want.

On the topic of wether the standard requires the distribution to be uniform. Well I don't know the standard by hard, but many people here seem to have a copy of it, so it should be a simple matter to check it. Every implementation I've seen of C and almost any other language/tool generates pseudo-random number using a multiplicative seed, simply because it's a good enough method and requires relatively few operations.

And finally, I don't know if an implementation that generated normally distributed n