

Re: The problems in comp.lang.c

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2008-03/msg04869.html

- *From:* jxh <jxh@xxxxxxxxxxxxxx>
 - *Date:* Fri, 28 Mar 2008 11:59:47 -0700 (PDT)
-

On Mar 23, 9:40 am, "Jeff P. Bailey" <nos...@xxxxxxxxxxx> wrote:
[... laments about C topicality and fingers Richard H. for
bullying Jacob N. ...]

If I remember right, Richard had started a thread with suggestions
on expanding the topics discussed in this newsgroup.

I actually don't mind for **any** C programming question to be asked
here (since the name of the newsgroup sort of invites it), but the
answers to the questions should be maximally portable. If a
maximally portable solution isn't possible, then the answer should
explain that, and redirect the poster to an appropriate resource.

Threads has been given as an example. Here is a possible scenario:

Q: I have a multi-threaded application written in C. One thread is
spinning in a loop checking for the value of a flag to change that
will be changed by a different thread. However, even after the 2nd
thread changes the flag, the 1st thread doesn't detect the change.
Why is this happening, and how can I fix it?

Given this kind of question, there is a Standard C component that
may be useful to the developer: declaring the flag variable to
be volatile, and this should be provided as part of the answer.
But, the complete answer is that even volatile may be insufficient
(particularly in a multi-processor environment), and so some sort
of barrier synchronization is likely to be required to propagate
a change made in one processor to another one. However, barrier
synchronization APIs will be system specific, and the poster
must be redirected to some other resource (perhaps POSIX,
or perhaps a CPU architecture specific resource) to find the
best answer for their application.

The C FAQ already provides some precedence for giving answers to
platform specific solutions for one of the most commonly available
environments at the time the FAQ was written (e.g., DOS), so it
would not be such a big stretch to extend a helping hand to at least
point the user in the right direction. Certainly the poster of a
question can understand that a superior solution specific to their

Re: The problems in comp.lang.c

platform may be available from a more suitable resource than comp.lang.c.

This would be more helpful than dismissing the question out of hand because of topicality concerns.

I have read many of Jacob's posts, and he is at times overzealous about providing platform specific solutions (particularly, solutions that his implementation of lcc provides), and he does at times get the specifics of the C Standard wrong. I think everyone who has posted solutions here have had their answers nitpicked. What usually happens is that you get better at creating answers that have fewer nits. Someone that wants to help others will usually want to provide as accurate a statement as possible. So nits that are picked should be taken gratefully, and then used for providing better answers next time.

— James

.