

Re: track positions in arrays= index variables || pointers to elements?

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-02/2038.html

From: dandelion (*dandelion_at_meadow.net*)

Date: 02/14/05

Date: Mon, 14 Feb 2005 12:03:23 +0100

"Luke Wu" <LookSkywalker@gmail.com> wrote in message
news:1108362538.603608.303290@141g2000cwc.googlegroups.com...

<snip>

> > *Method 1: Pointers*
> > =====

<snip>

> *This instrument for holding a position holds a lot of information. It
> has a reference type equal to the type of each member of the buffer,
> and it holds an address of a member of the buffer. It does not require
> a "base" address for use, or any other variables/values.*

However, on the downside, this version is not relocatable and cannot be used
to denote specific entries in a common data-block between two systems or two
runs of the same program.

> > *Method 2: Index Ints*
> > =====

<snip>

> *This instrument for holding a position holds much less information.
> It's type is unrelated to the type of the array. It does not hold a
> value that has any meaning on it's own. It can't do it's work without
> being part of a larger expression that provides the base address for
> the array.*

On the upside, this version can readily be used to denote a position in a
block of data between two systems or two runs of the same program.

> *I prefer the pointer, because it can be readily passed around as
> arguments to functions and hold enough information on it's own for*

comp.lang.c: Re: track positions in arrays= index variables || pointers to elements?

> *those functions to access the array.*

Which is a perfectly good reason, but not the exclusive criterion. There are other uses in which a integer offset is much preferable, especially if the reference needs to be interchangeable between system or different runs of the same program. Since we can readily switch between the two, the choice for the actual version to be used is not very critical.