Subject: Re: Dereferencing a large array in a structure
Posted by greg michael on Wed, 27 Sep 2006 12:43:06 GMT
View Forum Message <> Reply to Message

In my widget programs, I settled for the option of putting everything into a single state variable, s - a sometimes huge structure containing all kinds of stuff, and frequently pointers to variable-sized or large items, which gets passed around via the tlb uvalue (extract and set it as the first and last things in the event handler). Then nearly every routine passes s as its first parameter. Seems to me much cleaner than using common blocks. In all other widget uvalues, I pass only its name (e.g. 'tab3.some_button'), and where necessary decompose that to call the right routine.

I think this gets the points for the speed and memory items. But you'd have to mess with your common blocks... (although it might not be so hard to pull out all the variables and put them in a state variable)

Greg

Braedley wrote:
> In one of my widget programs, I have a tab widget, and the bases that
> belong to it store a fair amout of data concerning that tab in their
> uvalues. When a tab is selected, all the data in the uvalue must be
> dereferenced and loaded into common block variables. This was done so
> that other widgets within the program have quick and easy access to the
> data of the currently open tab.
>
> Enter the problem: Some of the data can be very large. One field in
> the uvalue structure can be as large as a 20 by 50000 double array (or
> larger), and it obviously can take some time to copy, especially with
> memory running low with other similarly large structures for other
> tabs. I've already set the no_copy keyword in calls to widget_control
> when setting and getting the uvalues to help reduce the load on memory
> and help out with some speed, but the million element array still needs
> referencing into the structure to be set, and dereferenced from the
> structure of the newly selected tab, which is taking up a large portion
> of the time spent. Are the any faster ways of doing this? Bonus
> points for reducing the load on memory and not making me rewrite every
> subwidget (ie not messing with the common blocks).