Subject: Re: Dereferencing a large array in a structure
Posted by Braedley on Wed, 27 Sep 2006 12:20:52 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:
> 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).
>>
>
> Hello,
>
> How about storing a pointer to your data in your UVALUE instead of a
> copy of the data itself.
>
> In the widget contructor you can ...
>
> pData = PTR_NEW(myBigData)
> myStuff = {DATANAME: 'baggage',PDATA :pDATA, FLAVOR: 'vanilla'}
> WIDGET_CONTROL, myBaseWidgetID, SET_UVALUE = myStuff
>
>
> In the event handler you can then ...
>
> WIDGET_CONTROL, ev.TOP, GET_UVALUE = myStuff
>
> *(myStuff).pData = doSomething(*(myStuff).pData, FLAVOR = myStuff.flavor)
>
> WIDGET_CONTROL, ev.TOP, SET_UVALUE = myStuff
>
>

> Don't forget to free the pointer memory when the widget hierarchy dies
> (or when ever you want it to.)
>
> The common block stuff is throwing me off a bit. This isn't an IDL EPA
> quiz is it? (http://www.dfanning.com/misc_tips/iepa.html)
>
> Cheers,
> Name Withheld Just In Case It Is A Quiz

No, this is not a quiz.

Using a pointer is definately an option, and will help with memory management as well. The use of the common blocks is due to fact that many subwidgets use values stored in the common blocks, and it's easier if those widgets just grab what they need from the common blocks, instead of passing them in the call. In some cases, passing the values would actually break the code, as most calls are made through call_procedure. In all cases, using call_procedure is unavoidable.

Thanks
Braedley
