Subject: Re: passing structure elements... by value?
Posted by Liam E. Gumley on Fri, 13 Jul 2001 16:39:57 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
> I am seem to be somewhat confused on passing structure... this *almost*
> seems like one of those dreaded 'pass by reference/pass by value'
> problems...
>
> I have a simple routine that increments the value of a structure:
>
> pro test, mod_struct
>   for i=0, n_tags(mod_struct)-1 do begin
>   mod_struct.(i) = mod_struct.(i) + 1
>   endfor
> end
>
> As expected, this will increment all of the values in a passed structure.
>
> IDL> struct = {A:0, B:0} & big = replicate(struct,3)
> IDL> print, big
> {   0   0}{   0   0}{   0   0}
> IDL> test, big
> IDL> print, big
> {   1   1}{   1   1}{   1   1}
>
> But, when I try and increment a single element in the structure it fails?
>
> IDL> struct = {A:0, B:0} & big = replicate(struct,3)
> IDL> test, big[0]
> IDL> print, big
> {   0   0}{   0   0}{   0   0}
> ^^^^^^^^^^^^^^^^
>
> I expected the ^^ element above to be ones? Is there any way to force IDL
> to pass this by reference instead of passing by value? It would be nice if
> you could put brackets around the thing you want to pass by reference...
> something like '(big[0])'
>
> Have a good weekend all!
> Randall
>
> PS: I'm sure Liam's new book answers this...

It sure does: see pages 114-115.

Structure elements are passed by value. Therefore they cannot be
modified by a called procedure or function. To be more precise, the

called routine makes a copy of all input arguments, and the copies may be modified during the lifetime of the called routine. When control is returned to the caller, the copies of arguments that were passed by value are copied back to the corresponding calling arguments. The copies of arguments that were passed by value are destroyed.

The following entities are passed by reference:
Scalars
Arrays
Structures
Undefined variables

The following entities are passed by value:
Constants
Indexed subarrays
Structure elements
System variables
Expressions

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/