Subject: passing structure elements... by value?
Posted by Randall Skelton on Fri, 13 Jul 2001 16:03:39 GMT
View Forum Message <> Reply to Message

Hi all,

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...

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/

## Subject: Re: passing structure elements... by value?
## Posted by david[2] on Fri, 13 Jul 2001 16:53:41 GMT

View Forum Message <> Reply to Message

Randall Skelton writes:

```
>  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])'

You could easily add the field you want to change as
a second positional parameter:

```
 pro test, mod_struct, fieldno
   IF N_Elements(filedno) EQ 0 THEN BEGIN
    for I=0, n_tags(mod_struct)-1 do begin
      mod_struct.(I) = mod_struct.(I) + 1
    endfor
   ENDIF ELSE mod_struct.(fieldno) = mod_struct.(fieldno) + 1
 end
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

## Subject: Re: passing structure elements... by value?
## Posted by Jaco van Gorkom on Mon, 16 Jul 2001 09:45:29 GMT

View Forum Message <> Reply to Message

Randall Skelton wrote:

> 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])'
```

Note that in your example you are passing one *array* element of an array of structures. Indexed array elements are also passed by value - see Liam's list.
One workaround would be to copy the array element(s) into a temporary variable:

```
working_elements = big[range]
test, working_elements
big[range] = working_elements
```

, with range being a vector of indices, just [0] in the example.
Another workaround involves passing the indices-of-interest into the routine.

cheers,
  Jaco

 ----------------
Jaco van Gorkom                                gorkom@rijnh.nl
FOM-Instituut voor Plasmafysica `Rijnhuizen', The Netherlands

---

## Subject: Re: passing structure elements... by value?
## Posted by Vince Hradil on Tue, 17 Jul 2001 15:54:16 GMT

View Forum Message <> Reply to Message

---

Not sure this is a structure issue, but instead an array issue!
Array elements are passed by value.  Instead you have to do:
IDL> struct = {A:0, B:0} & big = replicate(struct,3)

```
IDL> tmpbig = big[0]
IDL> test, tmpbig
IDL> big[0] = tmpbig
IDL> print, big
{       1       1}{       0       0}{       0       0}
```

Or how about changing your pro to a function?

```
function test, mod_struct
  for i=0, n_tags(mod_struct)-1 do begin
    mod_struct.(i) = mod_struct.(i) + 1
  endfor
  return, mod_struct
end
```

then do:

```
IDL> struct = {A:0, B:0} & big = replicate(struct,3)
IDL> big[0] = test(big[0])
IDL> print, big
{       1       1}{       0       0}{       0       0}
```

NB I haven't tested any of the above code, since I am away from my "IDL computer" at  the moment ;^)

On Fri, 13 Jul 2001 17:03:39 +0100, Randall Skelton <rhskelto@atm.ox.ac.uk> wrote:

```
>  Hi all,
>
>  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...
>
```