Subject: Re: Structure Pass By Reference?
Posted by JD Smith on Mon, 25 Nov 2002 22:35:15 GMT
View Forum Message <> Reply to Message

On Fri, 15 Nov 2002 17:28:40 -0700, David Fanning wrote:

> Nidhi Kalra (nrk5@cornell.edu) writes:
>
>> I have a very deep structure that looks something like this:
>>
>> struct = $
>> {vmap, $
>> a: ptr_new(), $
>> b: ptr_new()$
>> }
>> }
>> struct = $
>> {rindex, $
>> x: 0, $
>> y: 0 $
>> }
>> }
>> where vmap.a is an array of structures of type RINDEX. vmap is an
>> actual defined object and rindex is just a structure
>>
>> in a vmap method i need to access individual elements of a. i tried
>> (for example):
>>
>> rindex1 = (*self.a)[2]
>> rindex1.x = 2
>>
>> I would think that since (*self.a)[2] is a structure of type rindex, it
>> would be passed by reference and therefore rindex1.x = 2 would mean
>> (self.a)[2].x =2, but this is not the case. Can somebody help me out?
>
> I guess I've been working with IDL for too long, because I don't find
> this result surprising at all. You are returning a subscripted array
> member. I would expect it to be returned by value. That is consistent
> with IDL's rules for this kind of thing.
>
> If you want to put it back after you have changed it, you just do this:
>
> rindex1 = (*self.a)[2]
> rindex1.x = 2
> (*self.a)[2] = rindex1

Hey Nidhi... couldn't get away from the clutches of IDL I see.

I'd probably just do it like this:

(*self.a)[2].x=2

It's more efficient, if uglier, but don't worry, there are much deeper structures (which much more arcane looking access calls) out there. The nice thing is, you can access all x's at once like:

mean_x=mean((*self.a).x)

Good luck,

JD
