
Subject: Re: Move data to structure fields without copying
Posted by [Russell\[1\]](#) on Wed, 07 Mar 2012 17:31:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mar 4, 9:35 am, David Fanning <n...@idlcoyote.com> wrote:

```
> Wox writes:
>> Is there a way of getting an array in a structure without copying it?
>
>> data=fltarr(89,1192,1192)
>> struct={data:temporary(data),...}
>
>> Using temporary doesn't seem to work in this instance on a 32-bit
>> machine (not enough memory). I also want to avoid using pointers
>
>> data=fltarr(89,1192,1192)
>> struct={data:ptr_new(temporary(data)),...}
>
>> This does work, but I will need to adapt all further use of the
>> structure (god forbid) and it doesn't seem worth the effort because
>> one day, I will have an array that can't be allocated on 32bit at all.
>> Any suggestions?
>
> Use the pointer. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
```

Is there any reason you don't simply create the structure and populate? I suppose, I'm imagining you created some floating point thing, then filled it, then stored it in the structure --- presumably for later use somewhere else. Why not skip that, and just create the data within the structure a priori:

```
struct={data:fltarr(89,1192,1192)}
```

Then, if you fill it with data, loop over the first index, or do whatever you planned to do? Then you should only need one copy of the data?

Of course, you should use pointers. That's what they're there for...

