
Subject: Re: compile a routine wich include a comun
Posted by [Paul Van Delst\[1\]](#) on Tue, 24 Jan 2006 16:09:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter Albert wrote:

>> No, it does not. Whether or not you point to the "complete" variable, or just a part of
>> it, a new heap "variable" is created.

```
>>
>> IDL> x=indgen(4,4)
>> IDL> print, x
>>   0   1   2   3
>>   4   5   6   7
>>   8   9  10  11
>>  12  13  14  15
>> IDL> p=ptr_new(x)
>> IDL> print, *p
>>   0   1   2   3
>>   4   5   6   7
>>   8   9  10  11
>>  12  13  14  15
>> IDL> print, (*p)[1:2,1:2]
>>   5   6
>>   9  10
>> IDL> (*p)[1:2,1:2] = (*p)[1:2,1:2] + 100
>> IDL> print, *p
>>   0   1   2   3
>>   4 105 106   7
>>   8 109 110  11
>>  12  13  14  15
>> IDL> print, x
>>   0   1   2   3
>>   4   5   6   7
>>   8   9  10  11
>>  12  13  14  15
```

>
>

> Damn, I should have tried it before. I was so sure :-(
>

> But then, this `_is_` a polite group. Nobody murmuring something like
> rtfm, while a look in the manual reveals :

>

> "If `InitExpr` is provided, `PTR_NEW` uses it to initialize the newly
> created heap variable. Note that the new heap variable does not point
> at the `InitExpr` variable in any sense-the new heap variable simply
> contains a copy of its value."

>

> Well, here is is, black on white. So you `_can't_` alias variables or
> subarrays.

Huh, how 'bout that? It's in the manual. (ehem)

I wonder why RSI decided to implement pointers this way? Or, to rephrase, why they called the implementation they chose a pointer?

paulv

--

Paul van Delst
CIMSS @ NOAA/NCEP/EMC
