Subject: Re: Pointers in IDL
Posted by Antonio Santiago on Tue, 13 Apr 2004 17:08:35 GMT
View Forum Message <> Reply to Message

Well, first of all sorry about my poor english (Perhaps i not understand well your message).

A pointer is a type of variable that points to another variable.
For example, i have an struct "image":

```
{table, $
 parent:0L, $
 boxtable:0L, $
 n_rows:0, $
 n_columns:0, $
 type: 0, $
 data: ptr_new() $
}
```

At first time i dont know number of columns, rows and data. Because this i create 'data' as a pointer (a NULL pointer).

After this when user specifies me a file a restore the data and fill the struct fiels with values and initialize the pointer:

```
IF PTR_VALID( self.child ) THEN $
 PTR_FREE, self.child

self.child = ptr_new( REPLICATE({children}, n_columns, n_rows) )
```

If you use ptr_new() you get a NULL pointer that dont points to anything.
If you use ptr_new(/allocate_heap) yo obtain a valid pointer taht points to a undefined variable at heap.

If you use the second form you must to free the data before to re-reference the pointer other time.

If you do:

```
a=ptr_new(bytarr(10))
```

you must free 'a' before do: a=ptr_new(bytarr(50)), if not you will have a lake.

Benjamin Hornberger wrote:
> Hi all,
>
> I still don't understand all aspects of pointers in IDL. 2 Questions:
>
> 1. What are null pointers for? I read that they can't be dereferenced. What
> is their purpose then? The Gumley book writes (pg. 61): "Null pointers are
> used when a pointer must be created, but the variable ... does not yet
> exist." What would I do then when the variable does exist later and I want
> the pointer to point to it? Wouldn't I use ptr_new(/allocate_heap) in the
> first place, i.e. not create a null pointer but a pointer to an undefined
> variable? Can anyone give an example when I would use ptr_new()?
>
> 2. If I point a pointer to a variable (e.g. *ptr=indgen(100)) and later
> point it to a smaller variable (*ptr=indgen(50)), do I have a memory leak?
> I.e., do I have to free it before I re-reference it?
>
> I want to write a GUI which can open files which contain arrays of varying
> size. Is it ok to define a pointer in the GUI to hold these arrays
> (ptr=ptr_new(/allocate_heap)), and then whenever I open a new file, just
> dereference to the new array (*ptr=array)? Or do I have to free the pointer
> when I close one file and open another one?
>
> Thanks for your help,
> Benjamin
