Subject: Re: pointer to structures
Posted by John-David T. Smith on Tue, 04 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> eeeyler@my-deja.com (eeeyler@my-deja.com) writes:
>
>> suppose I wish to create a structure and wish to reference that
>> structure and its contents via a pointer:
>> filter=ptr_new({points:['a','b'],pt1_value:200, pt2_value:'X_WHYLOG'})
>> how do I reference the points array? I thought it would be as:
>> print, *filter.points
>> but I get the message
>> %Expression must be a structure in this context: Filter
>
> The problem here is that a pointer dereference has the lowest
> order of precedence. Lower, even, than a structure dereference.
> So, you must first dereference the structure (by using
> parentheses), and *then* dereference the pointer, like this:
>
>   filter=ptr_new({points:['a','b'],pt1_value:200, pt2_value:'X_WHYLOG'})
>   Print, *(filter.points)
>

Uh oh... forgot to test your code I'm afraid. You have the precendence correct,
but the solution reversed. You must force the pointer dereference to occur
before structure dereference... see the other posts.

If you had something like this:

filter={points:ptr_new(['a','b']),pt1_value:200, pt2_value:'X_WHYLOG'}

You can use the precendence to your advantage, visa vis:

print,*filter.points

Keeping the precendence in mind can eliminate extraneous parentheses, which, for
everyone except lisp programmers, often add confusion.

JD

--
 J.D. Smith |*| WORK: (607) 255-5842
 Cornell University Dept. of Astronomy |*| (607) 255-6263
 304 Space Sciences Bldg. |*| FAX: (607) 255-5875
 Ithaca, NY 14853 |*|