Subject: Re: Extract a sub-structure from a structure
Posted by David Fanning on Thu, 26 Jan 2006 15:42:52 GMT
View Forum Message <> Reply to Message

L. Testut writes:

> ; it works ok until now
> ; Then as I am very optimistic I want to do that:
> tag   = tag_names(stp)
> FOR i=1,N_ELEMENTS(tag)-1 DO *stp.(i)=*stp.(i)[IDP]
>
> ;but it doesnt' work ? is there somebody that can tell me what is the
> ;right way to replace in my structure the total region with the
> ;sub-region for all the parameter of course !

You have to remember that pointer dereferencing has the *lowest* order of precedence in all the things you are doing, *including* structure dereferencing. You code would probably work with a few more parentheses:

```
  FOR I=1,N_ELEMENTS(tag)-1 DO (*stp).(I)=(*stp).(I)[IDP]
```

But, personally, I would go with Solution 1. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Extract a sub-structure from a structure
Posted by L. Testut on Thu, 26 Jan 2006 16:15:21 GMT
View Forum Message <> Reply to Message

Hi David,

Even with the parantheses it doesn't work ?
With solution 1 it becomes more tricky to change the type of each field parameter in the structure ? and what about the memory occupied by the old complete structure (which I don't need anymore)...or was it a joke ?

Then

```
  toto=findgen(100)
  print,toto[where(toto lt 50)]
  toto=toto[where(toto lt 50)]
```

  this sequence works and I've read on this newsgroup that a
de-referenced pointer is lik a variable
  so

```
  toto=ptr_new(findgen(100))
  print,*toto[where(*toto lt 50)]
  *toto=*toto[where(*toto lt 50)]
```

   why this last sequence doesn't work ? even with parantheses ?

Cheers,
Laurent
PS: what is the local time in Fort Collins ?

Subject: Re: Extract a sub-structure from a structure

Posted by David Fanning on Thu, 26 Jan 2006 16:32:22 GMT
View Forum Message <> Reply to Message

L. Testut writes:

>   Even with the parantheses it doesn't work ?
>   With solution 1 it becomes more tricky to change the type of each
> field parameter in the structure ? and what about the memory occupied
> by the old complete structure (which I don't need anymore)...or was it
> a joke ?

Have you *ever* known me to make a joke!? I'm
always deadly serious when it comes to IDL. :-)

(BTW, I've been told by my son that his girlfriend is
terrified of me, primarily because she can never tell
if I am joking or not. Must be an acquired taste.)

>   this sequence works and I've read on this newsgroup that a
> de-referenced pointer is lik a variable
>   so
>
>   toto=ptr_new(findgen(100))
>   print,*toto[where(*toto lt 50)]
>   *toto=*toto[where(*toto lt 50)]
>
>   why this last sequence doesn't work ? even with parantheses ?

Well, what parentheses? If I put parentheses in the right
place, it works for me:

```
  toto=ptr_new(findgen(100))
  print,(*toto)[where(*toto lt 50)]
  *toto=(*toto)[where(*toto lt 50)]
  help, *toto

     <PtrHeapVar1>   FLOAT     = Array[50]
```

> PS: what is the local time in Fort Collins ?

I don't know. I've been up since 2AM with a dying dog. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Extract a sub-structure from a structure
Posted by L. Testut on Thu, 26 Jan 2006 17:47:57 GMT

View Forum Message <> Reply to Message

That's great it works !
thanks a lot and sorry for your dog
Cheers,
Laurent

Subject: Re: Extract a sub-structure from a structure
Posted by btt on Fri, 27 Jan 2006 15:29:53 GMT

View Forum Message <> Reply to Message

L. Testut wrote:
> ; Dear all,
> ; I have a big structure st={lat:FLTARR(N), lon:FLTARR(N),
> para1:FLTARR(N),..., paraN:INTARR(N)} with many parameters
> ; Lat and lon represent coordinates of an altimetric point above the
> sea surface and concern let's the entire globe.
> ; What I want to do is to extract a sub-region from this struture, I've
> seen that there is at least two solutions :
> ; 1) I create a new structure of the new size

```
> ; 2) or I use pointer to change the size of my field whitout defining a
> new structure
> ;
> ;
> ; I prefer the second solution but I only manage to do the first one
> !!!
>
>    ;*******SOLUTION 1***************
>    ; I define stn "normal" structure
> stn = {x:findgen(100),y:findgen(100),z:(indgen(100)+500.),
> para1:LONARR(100), para2:INTARR(100)}
> help,stn,/str
>
>    ; I select a sub-region of interest
> IDN  = WHERE((stn.x LT 50) AND (stn.y LT 50),count)
> print,count
> plot,stn.x[ID],stn.y[ID]
>
>    ; Now if I want to work with a struture representative of the
> sub-region which is smaller
>    ; and needs less computing time I have to do some contorsions like :
> tag    = tag_names(stn)
> NewStn = CREATE_STRUCT(tag[0], FINDGEN(count))
> FOR i=1,N_ELEMENTS(tag)-1 DO NewStn=CREATE_STRUCT(NewStn, tag[i],
> FINDGEN(count))
>
> help,NewStn,/str
>
> ; ! It seems to be OK ! (unless the field type have changed ! ) Now I
> tried the second solution
>
>    ;*******SOLUTION 2***************
>    ; I define stp "pointer" struture
>
> stp =
>  {x:ptr_new(findgen(100)),y:ptr_new(findgen(100)),z:ptr_new(( indgen(100)+500.)),
> para1:ptr_new(LONARR(100)), para2:ptr_new(INTARR(100))}
>
> help,stp, /str
>
> ; I select a sub-region of interest
> IDP  = WHERE((*stp.x LT 50) AND (*stp.y LT 50),count)
> print,count
>
> ; it works ok until now
> ; Then as I am very optimistic I want to do that:
> tag    = tag_names(stp)
> FOR i=1,N_ELEMENTS(tag)-1 DO *stp.(i)=*stp.(i)[IDP]
```

>
> ;but it doesnt' work ? is there somebody that can tell me what is the
> ;right way to replace in my structure the total region with the
> ;sub-region for all the parameter of course !
> ; Thanks a lot !
> ; Laurent
>

Hi,

It sounds like you have been steered down a good path by David, but there is another way to handle your data from the get-go. You have defined a structure whose fields are vectors of the same length. An alternative is to define a simple structure whose fields are scalars, then bundle the individual structures into a vector. Note the difference: "a structure of vectors" versus "a vector of structures".

If you adopted the latter, then subsetting is easier in terms of working with the data (at least for a slug like me.)

```
;make a 100 element vector of simple strutures
n = 100
s = REPLICATE({T: 0.0d, Y: 0.0}, n)
s.T = TIMEGEN(n, START = SYSTIME(/JUL), UNIT = 'DAY', STEP = 0.5)
s.Y = RANDOMN(seed, n)

;plot the T and Y values for ALL the elements
;in your vector of structures
Plot, s.T, s.Y, XTICKUNIT = 'time'

;find a subset of the field
;and plot it
A = WHERE((s.Y LE 0.5) AND (s.Y GE -0.5), nA)
if nA GT 0 then OPLOT, s[A].T, s[A].Y, psym = 6
```

This style of holding your data gives it the feel of records in a database.

Cheers,
Ben

Subject: Re: Extract a sub-structure from a structure
Posted by JD Smith on Sat, 28 Jan 2006 02:01:24 GMT

View Forum Message <> Reply to Message

On Thu, 26 Jan 2006 08:15:21 -0800, L. Testut wrote:

> Hi David,
>
> Even with the parantheses it doesn't work ?
> With solution 1 it becomes more tricky to change the type of each
> field parameter in the structure ? and what about the memory occupied
> by the old complete structure (which I don't need anymore)...or was it
> a joke ?

You don't need to place parens at random until it works. There is a very easy to remember (if not obvious) rule, which I've discussed a few times, including in the operator precedence tutorial. Here's the most recent discussion:

http://groups.google.com/group/comp.lang.idl-pvwave/msg/3ea7 242603c7f040

JD