Subject: Re: basic array-structure understanding question
Posted by Kenneth P. Bowman on Wed, 19 Nov 2008 15:10:06 GMT
View Forum Message <> Reply to Message

In article
<68a3f7b5-1c15-4dbb-bf83-b019eb7dd204@g17g2000prg.googlegroups.com>,
julia.walterspiel@gmail.com wrote:

> hi everybody
>
> This is what does not get into my head:
>
> I have one array containing my data.
> I have a second array containing the times when the data was
> collected.
> They have the same lenght.
>
> I put them into one structure because we all like structures and I
> learned about their advantages.
>
> now:
>
> IN SHORT are those arrays linked somehow? in other words: when working
> with a structure, does IDL know which value matches the corresponding
> date, given the data and the date array have the same length. Are
> those arrays somehow connected or are they completely independent from
> each other?
>
>
> IN LONG: if I want to get e.g. all the data from all Januarys from
> 2000 to 2007, can I do it somehow like
>
> all_jan = structure.data (where(structure.time EQ 200?01??????))..
> then it would automatically "select" only the right values. (and then
> I would have to make a separate array of the corresponding dates, make
> those two new arrays a structure so that I can plot data vs. date all
> in one?)
>
> or do I have to do it via some indexing, like:
> all_jan = where (structure.time EQ 200?01??????))
>
> and then apply the index to my array of data?
>
>
> jeez, i'm real slow on the uptake here.. sorry for bothering you with
> such simple questions, but can anybody help me undo this big knot in
> my brain?
> cheers,

> juls

An IDL structure is just a container, into which you can put a collection of (related) variables. Then you can pass the structure around, instead of having to pass all of the variables individually.

```
IDL> x = findgen(10)^3
IDL> i = lindgen(10)
IDL> str = {x:x, i:i}
IDL> help, str, /str
** Structure <1db1314>, 2 tags, length=80, data length=80, refs=1:
   X               FLOAT     Array[10]
   I               LONG      Array[10]
IDL> help, str.x
<Expression>    FLOAT     = Array[10]
IDL> help, str.i
<Expression>    LONG      = Array[10]
IDL> plot, str.i, str.x
IDL> even = where((str.i MOD 2) eq 0)    ;Find even indices
IDL> print, even
           0           2           4           6           8
IDL> oplot, str.i[even], str.x[even], psym = -1  ;Plot even indices only
```

Ken Bowman