Subject: Re: Matching Structure Tag Lengths
Posted by R.Bauer on Fri, 28 Jun 2002 07:12:36 GMT
View Forum Message <> Reply to Message

Nate Doyle wrote:
>
> Hi there everyone. I've got a kind of confusing question that I can't
> quite figure out.
>
> I've got a bunch of data that is stored in two different structures,
> structure P and structure S. Structure P has 16 tags each of which is
> approximately an eighth of size of the tags in structure S (Structure
> S has 21 tags but that's pretty inconsequential).
>
> What I need to do is do some sort of interpolation between the two
> structures to make the arrays in Structure P the same size as the
> arrays in Structure S. I have an array in each structure that I know
> starts and ends at the same number so I'm trying to use this to do the
> interpolation. My code is below. I'm trying to use the replicate
> function but it doesn't seem to do the trick. I think I'm doing
> something wrong. Any help would be appreciated.
>
> p2=replicate(P[0],n_elements(S)) ; n_elements is a function that
> determines the number of elements
>
> for i=0,n_tags(P)-4 do begin
> temp=interp1(P.ms,P.(i),S.ms)
> P2.(i)=temp
> endfor
>
> the last three elements of P are meaningless hence the -4. The .ms is
> the array that I know to be equal.
>
> This is as far as I've gotten. Unfortunately I cannot do this at work
> as my computer won't connect to the board and allow me to post.
>
> Thanks in advance.
>
> Nate Doyle

My suggestion.

P has less tags and this tags are all given in S.

result=struct2ptr_struct(P)
tn=tag_names(p)
pos=tag_position(S,tn)

n=n_elements(tn)
FOR I=0,n-1 do $
*result.(I)=interp1(P.(i),S.(pos[i]))
; I don't know this routine interp1

result=ptr_struct2struct(result,/free)

needed routines:

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl
_html/dbase/download/struct2ptr_struct.tar.gz
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/download/tag_position.tar.gz
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl
_html/dbase/download/ptr_struct2struct.tar.gz

for further routines and licensing please have a look at

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

regards

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
========== =======

Subject: Re: Matching Structure Tag Lengths
Posted by doyle on Mon, 08 Jul 2002 18:43:34 GMT
View Forum Message <> Reply to Message

> My suggestion.
>
> P has less tags and this tags are all given in S.
>
> result=struct2ptr_struct(P)

> tn=tag_names(p)
> pos=tag_position(S,tn)
> n=n_elements(tn)
> FOR I=0,n-1 do $
> *result.(I)=interp1(P.(i),S.(pos[i]))
> ; I don't know this routine interp1
>
> result=ptr_struct2struct(result,/free)

Thanks for the reply. I'm not sure that I communicated my problem quite right so I think your solution is for a slightly different problem. I want the number of tags in the arrays to stay the same, it is the length of the tags that I want to change. Perhaps I just don't understand your code though. It's very possible.

Nate Doyle

Subject: Re: Matching Structure Tag Lengths
Posted by R.Bauer on Thu, 11 Jul 2002 20:33:39 GMT
View Forum Message <> Reply to Message

Nate Doyle wrote:
>
>> My suggestion.
>>
>> P has less tags and this tags are all given in S.
>>
>> result=struct2ptr_struct(P)
>> tn=tag_names(p)
>> pos=tag_position(S,tn)
>> n=n_elements(tn)
>> FOR I=0,n-1 do $
>> *result.(I)=interp1(P.(i),S.(pos[i]))
>> ; I don't know this routine interp1
>>
>> result=ptr_struct2struct(result,/free)
>
> Thanks for the reply. I'm not sure that I communicated my problem
> quite right so I think your solution is for a slightly different
> problem. I want the number of tags in the arrays to stay the same, it
> is the length of the tags that I want to change. Perhaps I just don't
> understand your code though. It's very possible.
>
> Nate Doyle

Dear Nate,

I try to explain what I did.

At the moment structure P has less tags as structure S. But
all tags of P are defined in S too.
The tags which are in both structures could be used for an
interpolation.
The function struct2ptr_struct duplicates structure P but all values
of the tags are defined as pointer values.
This means the result structure is defined by this command.
With tag_names a list of all tags given from structure P is stored.
Now it is necessary to know where are these tags defined in structure S
The positions of the tags in structure S is returned with tag_position.
At least a FOR loop runs through each tag of the structure result
with the depending structure P.(i) and the S.(pos[i]) tags.
You have used interp1 in your code. I don't know the syntax from this.
We have some other routines e.g. calc_linear_with_limits in our lib.

While you are not using pointers yourself the result Pointer
Structure is changed by ptr_struct2struct(result,/free) in
a normal structure without pointers.

regards

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
 ---------- -------
       a IDL library at ForschungsZentrum Juelich
  http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
 ========== =======