Subject: Re: pointer & structure
Posted by penteado on Tue, 12 Jan 2010 15:50:49 GMT
View Forum Message <> Reply to Message

On Jan 12, 1:42 pm, bing999 <thibaultga...@gmail.com> wrote:
> Hi,
>
> i have created a structure S which contains a few elements (arrays)
> a,b,c : S={a,b,c}
> Then i replicate it N times.
> Now i want to create iteratively 10 other structures with the same
> skeleton.
> Since, i did not see any other way to do, i used a pointer p to stock
> the structures:
>
> for k=0,10 do begin
> *p(k) = S
> endfor
>
> By typing help,*p(k) i know *p(k) is indeed a structure BUT *p(k).a
> (for instance) prints "Expression must be a structure in this context:
> P"
>
> So my question is: how can I extract the information stored in *p(k) ?

The issue is the precedence of the operators. Use

(*p[k]).a

*p[k].a means *(p[k].a)

You do not need pointers for that array, you could have directly put
that in a structure array with p=replicate(S,N), which makes an array
with N copies of S. Then you can change the values of individual
elements by their indexes.

Subject: Re: pointer & structure
Posted by Thibault Garel on Tue, 12 Jan 2010 16:05:48 GMT
View Forum Message <> Reply to Message

On Jan 12, 4:50 pm, pp <pp.pente...@gmail.com> wrote:
> On Jan 12, 1:42 pm, bing999 <thibaultga...@gmail.com> wrote:
>
>
>
>> Hi,
>

>> i have created a structure S which contains a few elements (arrays)
>> a,b,c : S={a,b,c}
>> Then i replicate it N times.
>> Now i want to create iteratively 10 other structures with the same
>> skeleton.
>> Since, i did not see any other way to do, i used a pointer p to stock
>> the structures:
>
>> for k=0,10 do begin
>> *p(k) = S
>> endfor
>
>> By typing help,*p(k) i know *p(k) is indeed a structure BUT *p(k).a
>> (for instance) prints "Expression must be a structure in this context:
>> P"
>
>> So my question is: how can I extract the information stored in *p(k) ?
>
> The issue is the precedence of the operators. Use
>
> (*p[k]).a
>
> *p[k].a means *(p[k].a)

OK, thank you, that works !

>
> You do not need pointers for that array, you could have directly put
> that in a structure array with p=replicate(S,N), which makes an array
> with N copies of S. Then you can change the values of individual
> elements by their indexes.

Actually, i replicated S first: S=replicate(S,N) but then, i need to
iterate that process 10 times. And i used pointers (k=0 => 10) for
this because i do not know any other way to do it...

Subject: Re: pointer & structure
Posted by penteado on Tue, 12 Jan 2010 16:30:22 GMT
View Forum Message <> Reply to Message

On Jan 12, 2:05 pm, bing999 <thibaultga...@gmail.com> wrote:
> Actually, i replicated S first: S=replicate(S,N) but then, i need to
> iterate that process 10 times. And i used pointers (k=0 => 10) for
> this because i do not know any other way to do it...

It is not very clear what is going on. Could you copy all the relevant
lines, including where you define S and p, and where you get the

problem?