Subject: Re: Static Variables in IDL
Posted by rivers on Thu, 14 Mar 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article <4i8geg$5fv@vixen.cso.uiuc.edu>, santanu@eehpx22.cen.uiuc.edu
(S Bhattacharyya) writes:
> rivers@cars3.uchicago.edu (Mark Rivers) writes:
>
> Pardon me if I appear a bit obtuse, but I am still a little confused. I am
> under the impression that the common block declaration is equivalent to
> C's global declaration. What I would like to have is a bit of non re-entrant
> code in a standalone function (.pro). I want an IDL .pro that does the
> following::
>
> main()
> {
>  for(;;) non_rEntrant();
> }
> non_rEntrant()
> {
>  static int block=1;
>
>  if (block == 1){
>    puts("This is executed only once");
>    block=0;
>        }
>  puts("And this is done over and over again");
> }
>
> The following does exactly the same thing ----------
>
> pro test
> common SHARE,block
> block=1
>
> repeat begin
>   call_procedure,'non_rEntrant'
> endrep until block eq 1
> end
>
> pro non_rEntrant
> common SHARE,block
> if block eq 1 then begin print,'This is executed only once' & block=0 & endif
> print,'And this is done over and over again'
> end
>
>  But the two lower level function/pro's are not the same, the
> C version is standalone, how do I make the IDL pro non_rEntrant behave

> in the same way ?

Change your IDL example to the following:

```
pro test
while (1) call_procedure,'non_rEntrant'
end

pro non_rEntrant
common SHARE,block
if (n_elements(block) eq 0) then begin
   print,'This is executed only once'
   block=0
endif
print,'And this is done over and over again'
end
```

The key is n_elements(). If non_rEntrant has never been called then block is undefined and n_elements(block)=0.


Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)