Subject: Re: Memory management
Posted by isaacman on Fri, 12 Jun 1992 19:29:00 GMT
View Forum Message <> Reply to Message

In article <12JUN199210563303@stars.gsfc.nasa.gov>, landsman@stars.gsfc.nasa.gov (Wayne Landsman (301)-286-3625) writes...
> In article <12JUN199209141539@stars.gsfc.nasa.gov>, isaacman@stars.gsfc.nasa.gov (Subvert the Dominant Paradigm! (301) 513-7769) writes...
>> There seems to be some peculiarity in the way IDL does its memory
>> management in the TRANSPOSE function. Running on a VAXstation 3100/76,
>> the following statements cause the process to hang:
>>
>> IDL> A = FINDGEN(3,200000)
>> IDL> A = TRANSPOSE(A)
>>
>> The problem can be made to disappear as the arry size is made smaller,
>> presumably depending on one's page file quota and so forth. (On our
>> system, using A = FINDGEN(3,100000) works fine.)
>>
>> Rich Isaacman
>> COBE Project
>> NASA/Goddard Space Flight Center
>>
>
> I also have problems using TRANSPOSE with a 3 by 200000 array on both SUN
> V2.3.1 and VAX V2.2.2. (Note that TRANSPOSE works properly for much larger
> arrays if they are closer to being square, so, as Rich noted, the problem is not
> one of simple virtual memory limits.)
> Presumably, TRANSPOSE has been optimized for a near square array, but the
> code should be fixed up so that it doesn't hang up an IDL session.
>
> Meanwhile, the following IDL code accomplishes the TRANSPOSE fairly quickly:
>
> a = findgen(3,200000)
> b = fltarr(200000,3,/nozero)
> for i=0,2 do b(0,i) = REFORM(a(i,*))
>
> **********
>
> Wayne Landsman
> landsman@stars.gsfc.nasa.gov
>

RSI has responded to my original posting and reported that there is indeed a bug in TRANSPOSE. The fix has been made and will be in the next release after Version 2.3. Wayne's hypothesis, it turns out, is correct: TRANSPOSE is optimized for a square array and will fall into an infinite loop if (larger_dimension / smaller_dimension) * element_size

is greater than 250 Kbytes.

Rich Isaacman