Subject: Re: Array Subscripting Memory Usage (watch out!)
Posted by JD Smith on Fri, 04 Oct 2002 20:58:44 GMT
View Forum Message <> Reply to Message

On Fri, 04 Oct 2002 13:13:14 -0700, Dick Jackson wrote:

>> STRING:
>> .reset
>> s=memory()&x=indgen(20,/string)&e=memory()& print,e[3]-s[0]
>> 748
>> .reset
>> s=memory()&x=indgen(10,/string)&e=memory()& print,e[3]-s[0]
>> 458
>> The difference is 290. How to explain, the string length is 12 of each
>> element. Normally I believe 1 char is 1 byte. So first result should
> be
>> 12 byte * 20 = 240 and the second one 120. The differnce should be
> 120.
>> So I have no explanation why it needs so much more memory. Any ideas?
>
> I think each string would be terminated by a null character, making it
> 13 bytes each, then increase that to 16 because of 4-byte words, but
> that still doesn't match... I give up at this point!

Actually, if you look through export.h, you can see that IDL strings are not only null-terminated (like C), but also have other information saved in the IDL_STRING structure, like the length and type of the string. You typically need 12 bytes just for an empty string! Also of interest in solving the puzzle:

```
IDL> print,'*'+indgen(1,/string)+'*'
*       0*
```

These strings are 12 characters long. That's 25 bytes so far, counting the terminating null. One more 4-byte word will do it. I'll leave that one as an exercise for the reader ;)

JD