
Subject: Re: intarr speed in structure

Posted by [Dr. G. Scott Lett](#) on Mon, 16 Feb 1998 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

c.c.mclean@ed.ac.uk wrote:

> Hi,
>
> When I allocate an array, as part of a structure, it
> takes far longer to perform the operation than normal.
>
> ie. a) is much quicker than b) where:-
>
> a) array=intarr(x,y,z)
> b) array={volume:intarr(x,y,z)}
>
> if I use x=2048,y=240,z=5 a) takes 0.1s, b) takes 0.4s
> if I use x=2048,y=240,z=15 a) takes 0.4s,b) takes 2mins
>
> The same thing happens with IDL 5.03 under win95 (P200, 32MB)
> and with IDL 4 on an HP workstation.
>
> Can anybody explain why there is this difference!
>
> TIA
>
> Calum...
>
> ===== Posted via Deja News, The Leader in Internet Discussion =====
> <http://www.dejanews.com/> Now offering spam-free web-based newsreading

In the case of the structure, IDL does the following:

Allocate enough memory for INTARR(x,y,z), and fill with zeroes.

Allocate enough memory for the structure, including enough to hold the array.

Copy the (original) array into the structure.

Free the memory holding the (original) array.

I'm just guessing, but this looks like at least 3 times as much work as simply allocating memory for a single array, not counting for special hardware and such.

Since 2 copies of the array exist temporarily, IDL uses (at least) twice as much

memory to build the structure as the array. In your latter case,

intarr(2048,240,15)

requires about 14MB, so {volume:intarr(2048,240,15)} requires 28MB to

build.

Adding the memory for the IDL, the operating system, and any other memory use,

you've probably exceeded the 32MB your system has, and you're using virtual

memory, which is many times slower than RAM. This probably explains why your second example is so much slower than the first.

--

=====

Dr. G. Scott Lett

slett@holisticmath.com

<http://holisticmath.com/>

=====