Subject: Re: array concatenation
Posted by Joost Aan de Brugh on Fri, 03 Oct 2008 15:15:12 GMT
View Forum Message <> Reply to Message

On Oct 3, 1:19 pm, lecacheux.al...@wanadoo.fr wrote:

```
>  On 3 oct, 13:12, lecacheux.al...@wanadoo.fr wrote:
>   I got a strange error when using the array concatenation construct
>   within a large loop (a few thousands).
>   something like:
>     b = 0B
>     for i=0,999 do begin
>       ... compute a = array of bytes (a few 100) ...
>       b = [b, a]
>       endfor
>  This could crash IDL and even, randomly, crash the system (Win2K).
>  Nothing found by catching errors.
>  Is there some limits in the implicit addressing of such arrays ?
>  The error disappeared when compiling with COMPILE_OPT IDL2.
```

Hello,

Maybe it has something to do with the array descriptor. Anyway, in a large group concatenation is not the most elegant way. In Matlab (a language similar to IDL), you get a warning if you use such a construction. It has to do with the fact that if your array grows, you ask your system for more space. A safer way is to ask for enough space at once.

```
afh = a few 100

b = BytArr(afh*1000) ; Here is where you ask for a lot of space.
for i=0,999 do begin
  ... compute a = array of bytes (a few 100) ...
  b[i*afh:(i+1)*afh-1] = a ; Now b does not grow in the loop
end
```

Or use a 2D array

```
b = BytArr(afh,1000) ; Here, you ask for the space again.
for i=0,999 do begin
  ... compute a = array of bytes (a few 100) ...
  b[*,i] = a ; Now b does not grow in the loop
end
```

Cheers,
Joost

```
b = Reform(b,afh*1000) ; Or b = Reform(b,N_Elements(b))
```

It is a bit harder if you have different 'a few 100's for each
iteration