Subject: Re: Finding a value in a array efficiently
Posted by JD Smith on Fri, 14 Apr 2006 18:05:02 GMT
View Forum Message <> Reply to Message

On Fri, 14 Apr 2006 01:58:29 +0000, Wayne Landsman wrote:

>
> "JD Smith" <jdsmith@as.arizona.edu> wrote in message
> news:pan.2006.04.13.23.00.45.990566@as.arizona.edu...
>> As an aside, do you avoid using BREAK instead of goto for compatibility
>> with old IDL versions?
>>
>
> Actually, today I decided to set the minimum IDL version for my code to V5.4
> instead of V5.3. V5.4 is when both BREAK and ARRAY_EQUAL() were
> introduced. So after first updating the the GOTOs to BREAK, I started
> to look if the code could be more efficient using ARRAY_EQUAL(). (I
> didn't immediately realize, though, that BREAK could be safely placed
> inside of an inner ENDIF clause, and it would still break out of the FOR
> loop.)

Yep, it works much like C's break.

>>> found = 1 - array_equal( array EQ value, 0)
>
>> Just negate your ARRAY_EQUAL. I use this all the time:
>>
>> in_array=~array_equal(array NE value,1b)
>
> This is still not quite optimal since the term (array NE value) is always
> evaluated for the entire array. That is why I was looking for a function,
> say, ARRAY_OR for which one could write
>
> in_array = array_or(array,value)
>
> which would return 1 as soon as it found an element of array equal to value,
> and not perform any operations on the remaining elements of the array.

I decided to check, and at least for my machine, evaluating "array NE value" is sufficiently slower than "array EQ value" (about 1.8x, not sure why), that the hypothetical speedup never materializes. In fact, now that I think about it a little more (and contrary to what you said), I see your method *does* stop as soon as it finds a match as well (and the timing confirms that). Both do require doing a boolean operation on the entire array first though. So, I think:

```
 ~array_equal(array EQ value,0b)
```

is your best bet.

I think you want an ARRAY_NE.  I'm sure RSI would be happy to add
that, but then you'd need to require IDL 7.0 ;).

JD