Subject: Re: Philosophy of for loops
Posted by John-David T. Smith on Tue, 29 Aug 2000 05:14:37 GMT
View Forum Message <> Reply to Message

Todd Clements wrote:
>
> Hello, again, everyone!
>
> I was just wondering what the general concensus of the "IDL Expert
> Programmers" was on the use of for loops. When I first learned IDL, I
> remember getting from someone or somewhere the mantra "for loops are evil"
> because they take up so much time. Of course, as I learn more and watch
> what goes on in this group, it seems like "for loops are sometimes evil"
> would be a better mantra. The question then becomes, when do they become
> evil?
>
> In response to my thread on summing diagonal elements, Craig said that for
> loops aren't always bad if you can do a lot at once, and his code proves
> that you can have some fast loops.
>
> So what defines a slow loop? Is it having a bunch of accesses to
> sub-elements of arrays? Is it just having a bunch of statments? I suppose
> I could do some tests of my own, and I have a little, but it's much more
> fun to hear what you all have to say on the subject. I wouldn't have seen
> any IDL-ku if I just kept my thoughts to myself!

Despite what many supposedly learned men of the cloth will say to the contrary, for loops are without question the embodiment of pure evil. It is only the demonic influences to which these unfortunate souls have yielded which speak, and we must take their lamentable and unholy descent as frightening examples of the ever-burgeoning power of the dark, unoptimized side.

The only for loops you should ever sully your hands with are those involving the occasional pointer or object arrays, and then only disdainfully and with a careful eye to preserving your hard fought IDL sanctity and virtue, lest you slip into the evaluation of individual rows or columns at a time, or even, may the gods curse it, *nested* for loops. And yes, the dark side can be seductive with its adding of matrix diagonals in a few simple lines, its promise to iterate "just a few times".

But simply compare the snake-like slitheriness and so-called "readability" of:

```
****************************************
tt = fltarr(nx+ny-1)
ll = lindgen(nx>ny)
for i = 0, ny-1 do tt(i)      = total((a(0+ll,i-ll))(0:i<(nx-1)))
for i = 1, nx-1 do tt(i+ny-1) = total((a(i+ll,ny-1-ll))(0:(nx-1-i)<(ny-1)))
****************************************
```

to the most properly considered and well-balanced:

```
****************************************
tt=total(a[(((dy=((di=lindgen(((n=nx<ny)),nx+ny-1)))/n))*(nx gt ny?1:nx)+ $
           (nx gt ny?1:-1)*((dx=di mod n))*(nx-1))>0<(nx*ny-1)]* $
         (dy ge dx AND (dy-dx) lt nx>ny),1)
****************************************
```

which would stand in your code like a pillar of sunlight, shining firmly through the firmament. Oh yes, the wretched demon seducters of the nether-codeland will pass before your eyes many beautiful visions, but a closer look will reveal a false beauty, marred to the point of hateful grotesqueness. But despair not! If you take upon yourself a solemn and unquavering vow to reveal and combat these foul and insidious creatures on each journey you undertake, down every darkened lane of inquest; if you stand firmly and resolutely in steadfast devotion to the good Vector; if you read aloud daily from the holy book of Histogram; than shall you triumph over IDL with all the powers of the right and just at your aid.

Your humble guide,

JD