Subject: Re: How to use pointers instead of arrays
Posted by David Fanning on Mon, 09 Dec 2002 14:56:44 GMT

View Forum Message <> Reply to Message

Murat Maga (maga@mail.utexas.edu) writes:

> The question is I need a 2 dimensional array to which new elements
> constantly appended during the execution. Currently I create a duplicate
> temp array, create a new one with right size, and finally transfer the
> values from the temp one to the actual one. I know people use pointers
> for things like this, but I never had an example. Can somebody post me a
> simple example? Do the things speed up if I use pointers?

People probably do use pointers for these sorts of things,
but if they do it doesn't solve their problem. :-)

The real problem is one of memory management. And continually
creating and recreating arrays is bad business no matter how
you do it, even with pointers.

What you want to do is allocate memory in big enough
"chunks" that it is efficient and meets the needs of
your program. For example, if the number of "things"
you are going to put into your array ranges from 10
to 1000, then you might allocate memory to your array
in chunks of 100.

In practice, this means that you have some kind of
counter to tell you where you are in your array. If the
counter gets above the "chunk" size, you allocate more
memory to the array:

```
array[counter] = value
counter = counter + 1
IF counter MOD 100 EQ 0 THEN array = [Temporary(array), Findgen(100)]
```

When you finish adding things to your array, you trim
it to the correct size:

```
array = array[0:counter-1]
```

This is both efficient and fast. But do learn about pointers.
They are incredibly useful creatures. They just won't do you
any good here.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155