
Subject: Re: Naive pointer question ?

Posted by [Bhautik Joshi](#) on Wed, 23 Jan 2002 04:51:58 GMT

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

> From my understanding, pointers are useful in C where you have the
> option of passing by value or by reference, but in IDL it seems one
> *always* passes by reference (insofar as any variable passed to a
> function and then changed within that function is also changed in the
> calling function).
> So I was wondering, what is the benefit of explicitly using pointers
> in IDL ?

Couple of reasons:

- * speed & efficiency (you are passing only a reference to a variable, but we've covered this already)
- * flexibility (easy to create and change dynamic data types)
- * future expansion (if you want to temporarily change the data type of something in a structure during the lifetime of a program while it is running, create that bit of data as a pointer and simply change where it points when you want to change the data)

and what I think is the coolest:

- * double, triple or higher (!!) dereferencing - a pointer pointing to a pointer pointing to a pointer pointing to a pointer pointing to a pointer blah blah *foam at mouth & fall over*

They allow you to make complex data structures and types that are good for many excellent and useful algorithms (such as ones based on linked lists or trees etc.).

However, on the flipside, if you want to effectively use pointers, you need to design your program with 'em in mind. Also, it runs against the method of passing data via common blocks (which are EVIL! EVIL! EVIL!) - data is instead passed down a heirachy of functions.

Well, anyway, thats my insane rant justifying their use, feel free to correct me if I'm wrong about anything :)

Cheers,
Bhautik

--

```
/------(__)------\
| nbj@imag.wsahs.nsw.gov.au | phone: 0404032617 |..|--\ -moo |
| ICQ #: 2464537           | http://cow.mooh.org | |--|
|-----+-----\OO/|| -----/
```

| international |
| roast. my sanity has gone |
its lost forever
