Subject: Re: C Alignment/IDL structures
Posted by Nigel Wade on Tue, 22 Mar 2005 10:34:18 GMT
View Forum Message <> Reply to Message

joey@swri.edu wrote:

> Randall Skelton <randall.skelton@gmail.com> wrote:
>> I'm not sure I completely understand what you are doing... can you post
>> a snipit of code? I trust that you are passing back an unnamed
>
> Here's my code that does the IDL/C interaction. I have a vector
(_dataIDL)
> which has all the values I want into IDL within it as a malloc'ed sets of
> memory.
>
> // Copy the real data
>
> unsigned long pos = 0;
> unsigned char *myStructureThatLooksLikeTags = malloc
(_totalSpaceNeeded
> * _dataIDL.size
());
> for (unsigned int i = 0; i < _dataIDL.size (); i++) {
> memcpy (&(myStructureThatLooksLikeTags [pos]), _dataIDL [i],
> _totalSpaceNeeded);
> pos += _totalSpaceNeeded;
> }
>
> char idl_struct_name [100];
> sprintf (idl_struct_name, "%s_K%ld_V%d", dbVirtualName (data_key),
> data_key, version);
>
> void *idl_struct = IDL_MakeStruct (idl_struct_name, tags);
> IDL_LONG dims;
> dims = _dataIDL.size (); // number of elment in the array of
structures
> IDL_VPTR ivReturn = IDL_ImportArray (1, &dims, IDL_TYP_STRUCT,
> myStructureThatLooksLikeTags,
> cleanUpIDL, idl_struct);
>
>> In general, when I have to pass C structures from existing code back to
>> IDL I do it by creating a shadow structure (in C) that uses all the
>> defined IDL types and copying the data. You really cannot rely on
>> generic C variables having the same size as thier IDL counterparts
>> (take a look at the definition of IDL_ALLTYPES in idl_export.h).
>
> Ok, this probably answers my question. I was hoping I could create an
array

> of structs, but this is maybe not so memory efficient so I might try to
> create one structure with multiple arrays.
>
> Joey

It is possible to do what you want to do. But you have to realize that there are some caveats, and they are pretty serious ones.

The padding within a C structure is not defined by the language. It is an implementation issue, and is entirely at the discretion of the compiler developer. There is no guarantee that the padding inserted by one C compiler will be the same as the padding inserted by another compiler. You can't even guarantee that the next minor release of your compiler will use the same padding as the current release. It's a pretty safe bet, but not guaranteed.

In your code you copy memory from two disimilar data structures using memcpy() taking no account of alignment. Your memory allocation uses the construct _dataIDL.size(), what is that? Since you haven't shown us the definition of the storage for _dataIDL or your IDL tags, then it's impossible to say what's wrong. But my guess would be your data types or alignment in _dataIDL doesn't match what you've specified in your IDL structure tags.

It is possible to generate arrays of structures in C and pass them back to IDL, but you have to be very careful and you have to undertand its limitations.

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555