Subject: Re: C Alignment/IDL structures
Posted by Nigel Wade on Thu, 31 Mar 2005 10:46:57 GMT
View Forum Message <> Reply to Message

joey@swri.edu wrote:

> Nigel Wade <nmw@ion.le.ac.uk> wrote:
>>>
>>> // 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;
>>> }
>>>
>
>> 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
>
> _dataIDL is actually a C++ vector of unsigned char *'s. This group of
> char * is created from a C++ map of data. The map is a tagname and data
> value. Complex, isn't it? I'd like to think its elegant since I can very
> simply wrap any C/C++ and interface it with IDL. However, if one of the
> items in the map is a structure itself, I think it has problem.

Perhaps it's overly complicated?

You have an array of tags called tags which you pass into IDL_MakeStruct. These are the tag names which will be set in the structure it creates. Why does the data structure _dataIDL also have tag names in it? I'm confused.

Are you trying to create a structure which contains a structure?. If so, what I do is set the tag type in the IDL_STRUCT_TAG_DEF array for the parent structure to IDL_TYP_STRUCT. For the child structure I use IDL_MakeStruct to create it's IDL structure, and then set the parent structure type for that tag to be this value.

E.g. this creates a structure containing a structure. The display_tags structure contains a structure with protocol_tags.

```
IDL_STRUCT_TAG_DEF protocol_tags[] = {
  { "VERSION", 0, (void *)IDL_TYP_LONG },
  { "REVISION", 0, (void *)IDL_TYP_LONG },
  { 0 },
};

IDL_STRUCT_TAG_DEF display_tags[] = {
   { "NAME", 0, (void *)IDL_TYP_STRING },
   { "VENDOR", 0, (void *)IDL_TYP_STRING },
   { "PROTOCOL", 0, (void *)IDL_TYP_STRUCT },
                /* set this to protocol_s later */
   { 0 },
};
void *protocol_s, *display_s;

protocol_s = IDL_MakeStruct(0, protocol_tags);
display_tags[2].type = protocol_s;

display_s = IDL_MakeStruct(0, display_tags);
```

IDL_TYP_STRUCT in the tags array is really just a place-holder and comment,
it doesn't do anything as it's replaced before the tags array is used to
create the parent structure.

--
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