Subject: Re: Problem with structures getting padded to multiples of 4-bytes
Posted by thompson on Tue, 22 Feb 1994 17:38:25 GMT
View Forum Message <> Reply to Message

zawodny@arbd0.larc.nasa.gov (Joseph M Zawodny) writes:

> Howdy All,

> I've come a cross something a bit odd with large structures. I'm
> running IDL Version 3.5.1 (OSF alpha) on a DECStation 3000 Model 400 AXP.
> I've used a lot of small structures that are a non-integer number of (4-byte)
> words in length with little problem, however, the following structure
> should be 2618 bytes in length but comes up 2620. When written to a file,
> the record is indeed 2620 bytes long and the last two (extra) bytes appear to
> be padded with zero's. Is this a known problem and are there work arounds for
> it? Rumor has it that this does not happen on a SUN SPARC-10 running whatever
> they run for an OS this week.

> Thanks,

> The structure:

(rest deleted)

I've run into this sort of thing myself. A much simpler structure that exhibits the same behavior is the following:

TEST = {FLOAT: 0.0, BYTE: 0B}

This should, in principal, be five bytes long, but comes out as being eight bytes instead. The behavior is consistent on a Sparcstation 1, on a DEC 5000/240 and on an AXP 3000/400 running OpenVMS. Apparently, structures want to be an even number of "words" (four bytes) long.

Another factor to consider is the following. Consider the structure:

TEST2 = {FLOAT: 0.0, BYTE1: 0B, INTEGER: 0, BYTE2: 0B}

This should be eight bytes long, but instead it comes out as 12. However, by slightly modifying the structure to

TEST3 = {FLOAT: 0.0, BYTE1: 0B, BYTE2: 0B, INTEGER: 0}

then it comes out to 8 as it should. Apparently, integers want to be start on "half-word" (two bytes) boundaries.

I use the terms "word" and "half-word" in quotes because, as far as I understand it, the true wordsize on AXP platforms should be eight bytes.

However, IDL on AXPs doesn't seem to behave any different than on any other platform.

These considerations are important when doing reads or writes with structure variables. The only workarounds that I can suggest is either

1. Design your structures carefully to conform to the above restrictions, or

2. Read the data in as byte arrays, and then use the IDL data conversion functions--FIX, FLOAT, DOUBLE, COMPLEX, STRING--to convert this into a structure, or visa-versa for writes.

There is some information on this in the IDL manual, under "Unformatted Input/Output with Structures" in the chapter on structures.

Good luck,

Bill Thompson
