Subject: Saving structure variables
Posted by Bernard Puc on Fri, 02 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I need some insight into how structure variables are saved in a binary file. I have data files written on an SGI which now need to be read on Linux. It seems that just using the swap_endian keyword in the OPENR command doesn't get me where I want to go. The structure tagnames come out correctly, however, the values are garbage.

Any help appreciated!

P.S. - The data file consists of an array of structures written out unformatted binary output.

--
Bernard Puc AETC, INC.
bpuc@va.aetc.com 1225 Jefferson Davis Highway #800
(703) 413-0500 Arlington, VA 22202

Subject: Re: Saving structure variables
Posted by Liam Gumley on Tue, 06 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Bernard Puc wrote:
> Ah, but unfortunately, the files were stored as unformatted binary.
> I need to find a way of reading them on a Linux platform. This is
> part of moving an archive of data one time to a new platform. I
> guess I'll have to read in the structures, resave as IDL save files,
> and transfer to the new platform.

Bernard,

I went back and read your original message again, and I think I understand your problem a little better. I assume the data was saved to disk on the SGI something like this:

```
record = {name:'Test', value:indgen(10)}
data = replicate(record, 100)
openw, 1, 'test.dat'
writeu, 1, data
close, 1
```

Then to read the data in Linux, the code should look like this:

```
record = {name:'    ', value:intarr(10)}
```

```
data = replicate(record, 100)
openr, 1, 'test.dat'
readu, 1, data
close, 1
data = swap_endian(data)
```

Note that only the data values are stored in the file on disk; the structure tag names are set by the IDL statements in the read procedure which define the structure.

I notice that you use the word 'archive' in your second post. In my experience, data archives tend to live longer than we expect, so I recommend putting at least a little effort into making sure the archived data can be read on both little-endian and big-endian platforms.

For example, if you stay with the method shown above, you could make the reader more intelligent, so that it queries the *data* itself to see whether the byte order needs to be swapped (this assumes that your data format stays constant). You can pick a couple of items from each record whose values you expect to be in a certain range, e.g.

HOUR must be in the range 0 to 24,
DATE must be in the range 19950101 to 20101231,

then your reader would work like this:

(1) Define the record structure
(2) Read the first record from the data file
(3) If HOUR and DATE are within limits, turn swap flag OFF and goto step (7)
(4) If HOUR and DATE are out of limits, use SWAP_ENDIAN on the entire record
(5) If HOUR and DATE are within limits, turn swap flag ON and goto step (7)
(6) The file is not in the expected format, so stop with an error message
(7) Read the rest of the data file, using SWAP_ENDIAN is swap flag is on

This will give you a reader that handles byte-swapping transparently on SGI or Linux.

Note that the IDL SAVE format could potentially change at some point in a future version of IDL. If I really wanted a safe *archive* format, I would not use SAVE. Rather, I would use a binary format as described above, or a portable self describing format like netCDF.

Cheers,
Liam.

--
Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: Saving structure variables
Posted by davidf on Tue, 06 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Bernard Puc (bpuc@va.aetc.com) writes:

> Ah, but unfortunately, the files were stored as unformatted binary.
> I need to find a way of reading them on a Linux platform. This is
> part of moving an archive of data one time to a new platform. I
> guess I'll have to read in the structures, resave as IDL save files,
> and transfer to the new platform.

I'm afraid in terms of time and effort, this is *by far* the
easiest solution. :-(

SAVE files will work, but you might have to change your
code less if you just save the data as XDR binary files. This
is mostly routine for those of us who have to work with
binary files on multiple platforms. :-)

Cheers,

David

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