Subject: Re: need help reading binary FORTRAN data
Posted by Andy Heaps on Tue, 20 Oct 2009 17:38:36 GMT
View Forum Message <> Reply to Message

Hi Laura,
as you say this sounds a little strange. Do you know what type of machine the file was written out on? Was it Linux, 32/64bit and was it written as a FORTRAN unformatted file?

I would have thought that:

```
openr, ilun, filename, /get_lun, /f77_unformatted
model = '1111'
header = intarr(7)
readu, ilun, model, header
```

Would have read the file headers from what you've said.

Could you put the file somewhere on a web server so that we could have a look? If not, is there a web site you could point to where you got this sort of file from?

Cheers
Andy

Laura wrote:

> Hi, I'm trying to read a binary data file (seems FORTRAN oriented)
> into IDL, but got strange result. Maybe someone can help me to figure
> it out here.
>
> The file description is like this:
>
> Record #1
>
> * CHAR*4 Meteorological MODEL Identification
> * INT*4 Meteorological file starting time (YEAR, MONTH, DAY, HOUR,
> FORECAST-HOUR)
> * INT*4 NUMBER of starting locations
> * INT*4 Concentration packing flag (0=no 1=yes)
>
> Record #2
> ......
>
> However, when I tried using following IDL code,
>
> openr, ilun, filename, /get_lun
> model = '1111'

> header = lonarr(7)
> readu, ilun, model, header
>
> I get empty string in model, and weird long integer values in header.
>
> Then I looked at the binary file myself and found the first 4 bytes
> are "00 00 00 20', which seems indicating the length of "Record #1",
> the second four bytes are "4E 41 4D 53", seems to be the model string.
> So I use the following code
>
> openr, ilun, filename, /get_lun
> junk = 1L
> model = '1111'
> header = lonarr(7)
> readu, ilun, junk, model, header
>
> This time I got the string in "model" right, but the value in "junk"
> is still strange, not the "32" as I expected. If I use a bytarr(4)
> array to read each intr*4 value, the value I got matched what I saw in
> the binary file (e.g., I got "0 0 0 32" for the "00 00 00 20").
>
> Is there something between the FORTRAN int*4 and the IDL LONG needs
> special treatment?
>
> BTW, I also tried to open the file with a /F77_UNFORMATTED keyword,
> but IDL complained it's a corrupted f77 unformatted file. SWAP_ENDIAN
> doesn't work either.
>
> Any thoughts?
>
> Thanks,
>
> Laura
>

## Subject: Re: need help reading binary FORTRAN data
## Posted by Laura on Tue, 20 Oct 2009 18:13:33 GMT

View Forum Message <> Reply to Message

Hi, Andy, thanks for the help.

It seems the "swap_endian" finally worked (I don't know what happened
in my last several tries). Now I only need to manually disregard the
padding before and after each record since /f77_unformated keyword
still does not work. I'll see if I can find some place to upload the
file.

Cheers,

Laura

On Oct 20, 1:38 pm, Andy Heaps <a.j.he...@reading.ac.uk> wrote:
> Hi Laura,
>      as you say this sounds a little strange.  Do you know what type of
> machine the file was written out on?  Was it Linux, 32/64bit and was it
> written as a FORTRAN unformatted file?
>
> I would have thought that:
> openr, ilun, filename, /get_lun, /f77_unformatted
> model = '1111'
> header = intarr(7)
> readu, ilun, model, header
>
> Would have read the file headers from what you've said.
>
> Could you put the file somewhere on a web server so that we could have a
> look?  If not, is there a web site you could point to where you got this
> sort of file from?
>
>         Cheers
>            Andy
>
> Laura wrote:
>> Hi, I'm trying to read a binary data file (seems FORTRAN oriented)
>> into IDL, but got strange result. Maybe someone can help me to figure
>> it out here.
>
>> The file description is like this:
>
>> Record #1
>
>>     * CHAR*4 Meteorological MODEL Identification
>>     * INT*4 Meteorological file starting time (YEAR, MONTH, DAY, HOUR,
>> FORECAST-HOUR)
>>     * INT*4 NUMBER of starting locations
>>     * INT*4 Concentration packing flag (0=no 1=yes)
>
>> Record #2
>> ......
>
>> However, when I tried using following IDL code,
>
>> openr, ilun, filename, /get_lun
>> model = '1111'
>> header = lonarr(7)

>> readu, ilun, model, header
>
>> I get empty string in model, and weird long integer values in header.
>
>> Then I looked at the binary file myself and found the first 4 bytes
>> are "00 00 00 20', which seems indicating the length of "Record #1",
>> the second four bytes are "4E 41 4D 53", seems to be the model string.
>> So I use the following code
>
>> openr, ilun, filename, /get_lun
>> junk = 1L
>> model = '1111'
>> header = lonarr(7)
>> readu, ilun, junk, model, header
>
>> This time I got the string in "model" right, but the value in "junk"
>> is still strange, not the "32" as I expected. If I use a bytarr(4)
>> array to read each intr*4 value, the value I got matched what I saw in
>> the binary file (e.g., I got "0 0 0 32" for the "00 00 00 20").
>
>> Is there something between the FORTRAN int*4 and the IDL LONG needs
>> special treatment?
>
>> BTW, I also tried to open the file with a /F77_UNFORMATTED keyword,
>> but IDL complained it's a corrupted f77 unformatted file. SWAP_ENDIAN
>> doesn't work either.
>
>> Any thoughts?
>
>> Thanks,
>
>> Laura

### Subject: Re: need help reading binary FORTRAN data
### Posted by penteado on Tue, 20 Oct 2009 20:56:30 GMT

View Forum Message <> Reply to Message

Note that things like INT*4 do not universally tell how many bytes are used in that integer. The Fortran standard does not specify which kind corresponds to which size of variable, it is up to the implementation to define which kind number corresponds to which variable size. It is common that INT*4 means a 4-byte integer, but it does not have to be so. That is the reason why Fortran has functions like selected_int_kind, to pick the proper kind number.

Subject: Re: need help reading binary FORTRAN data
Posted by Paul Van Delst[1] on Wed, 21 Oct 2009 15:44:14 GMT
View Forum Message <> Reply to Message

That syntax, INTEGER*4, REAL*8, etc is, as you say, not in any Fortran standard but are ubiquitous extensions. But, where those Fortran extensions are allowed, the "4" and "8" above are indeed the length (read: number of bytes) and not the kind type. Kind types didn't exist at all in Fortran until Fortran90, and the INTEGER*4, etc extensions were available before that (and still are for those norty folks who continue to use them.... :o)

cheers,

paulv

pp wrote:
> Note that things like INT*4 do not universally tell how many bytes are
> used in that integer. The Fortran standard does not specify which kind
> corresponds to which size of variable, it is up to the implementation
> to define which kind number corresponds to which variable size. It is
> common that INT*4 means a 4-byte integer, but it does not have to be
> so. That is the reason why Fortran has functions like
> selected_int_kind, to pick the proper kind number.