Subject: Re: little and big endian -- once more
Posted by Martin Schultz on Wed, 12 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Martin Schultz (mgs@io.harvard.edu) writes:
>
> [...]
>
> Here is a little function, Martin, that will make sure
> your code works without having to track every machine
> architecture down. Found it on my web page. :-)
>
> Cheers,
>
> David

well, that's not all you need. Of course I replaced my string('x86')
thing with the inline code of little_endian() and at first it seemed to
work fine. But I encountered one situation when the SWAP_IF keyword
would not detect things properly which is reading data from a mounted
network drive with different endian (i.e. a generic unix drive mounted
to a linux box). So, I guess, I should give Liam's suggestion a try.
Unfortunately, it seems that you must know what's in the file in order
to test it, but that's probably a good idea anyway when you are dealing
with binary files ;-) Just for clarification: is it true that the endian
matters for integers as well as floats?

Thanks again,
Martin.

--

||||||||||||||\\\\\\\\\\\\\\----------////////////// //||||||||||||||
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

Subject: Re: little and big endian -- once more
Posted by Liam Gumley on Wed, 12 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:

> took me a while to realize that it is the machine architecture not
> the OS that determines the byte swapping -- in fact I needed to have IDL
> on linux (after using it on an SGI and with Windows) to figure that out
> ... Anyway, I now devised the following test for byte swapping which is
> applied in my open_file routine and handles everything transparently.
> Please tell me if there are any other machine architectures that are big
> endian and what their !version.arch tag would be.

The way I usually handle this is to let the *data* tell me whether
byte-swapping is required. For example, my colleagues generate datafiles
in host order using the same code on SGI and Linux boxes. Rather than me
keeping track of what platform was used to create the data, when I open
the datafile in IDL my code interrogates the data itself to see if
appears to be swapped. There is usually some data item (e.g. a date
and/or time) that can be checked to see if it's in the correct range. If
it's in the correct range, then proceed. If not, check swap_endian(date)
to see if it's in the right range. If it looks ok, swap everything. If
it doesn't look ok, then the data is not in the expected format.

I usually only apply this method to files that others supply to me. If I
want to create platform-independent datafiles, I use netCDF or HDF.

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

Subject: Re: little and big endian -- once more
Posted by davidf on Wed, 12 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (mgs@io.harvard.edu) writes:

> took me a while to realize that it is the machine architecture not
> the OS that determines the byte swapping -- in fact I needed to have IDL
> on linux (after using it on an SGI and with Windows) to figure that out
> ... Anyway, I now devised the following test for byte swapping which is
> applied in my open_file routine and handles everything transparently.
> Please tell me if there are any other machine architectures that are big
> endian and what their !version.arch tag would be.

Here is a little function, Martin, that will make sure
your code works without having to track every machine
architecture down. Found it on my web page. :-)

Cheers,

David

```
************************************************
FUNCTION Little_Endian

; Returns 1 if Little-Endian, returns 0 if Big-Endian.

  little_endian = (BYTE(1, 0, 1))[0]
  RETURN, little_endian
END
************************************************
```

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

Subject: Re: little and big endian -- once more
Posted by Dick Jackson on Thu, 13 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hi,

Martin Schultz wrote:
>
> Hi all,
>
> took me a while to realize that it is the machine architecture not
> the OS that determines the byte swapping -- in fact I needed to have IDL
> on linux (after using it on an SGI and with Windows) to figure that out
> ... Anyway, I now devised the following test for byte swapping which is
> applied in my open_file routine and handles everything transparently.

I'm not sure if I'm missing some subtlety here, but wouldn't the (newish) keywords /SWAP_IF_[BIG|LITTLE]_ENDIAN to IDL's OPEN and BYTEORDER routines work in all cases? I've used them to good effect for cross-platform work.

From the help file:

[...] it only takes effect if the current system has [big|little] endian byte ordering. This keyword does not refer to the byte ordering of the input data, but to the computer hardware.

Cheers,
--
-Dick

Dick Jackson Fanning Software Consulting, Canadian Office
djackson@dfanning.com Calgary, Alberta Voice/Fax: (403) 242-7398
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: little and big endian -- once more
Posted by Liam Gumley on Thu, 13 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
> So, I guess, I should give Liam's suggestion a try.
> Unfortunately, it seems that you must know what's in the file in order
> to test it, but that's probably a good idea anyway when you are dealing
> with binary files ;-)

Yeah, it's pretty hard to read a binary file if you don't know the format.

> Just for clarification: is it true that the
> endian matters for integers as well as floats?

Yes, it's true.

Here at SSEC we've messed with porting binary datafiles for years. However in the last year or so, netCDF has become very widely used in our applications. Some of the reasons are:
(1) No more byte-swapping worries, ever.
(2) The same IDL code can read the same datafile and give the same results on any IDL platform (Unix, Linux, Windows, Mac).
(3) If you give someone the file and tell them it's netCDF, they can then easily figure out what it contains. You don't need to tell them the format details.
(4) The netCDF API is available for Fortran77/90, C, C++, IDL, Matlab, perl, Python, and Java.
(5) The API is very simple and concise: the only things you can store in a netCDF file are arrays, attributes, and dimensions. However, you can create structures using these building blocks which are as simple or as complex as you like.
(6) netCDF supports the following IDL datatypes: BYTE, STRING, INT, LONG, FLOAT, DOUBLE.

More information is available at
http://www.unidata.ucar.edu/packages/netcdf/faq.html

Liam E. Gumley
Space Science and Engineering Center, UW-Madison