Subject: Converting Byte Arrays
Posted by popprice on Mon, 06 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I am struggling with a file i/o problem. I have a data file which comes from a PC. I know the data structure in terms of how many bytes correspond to each entry in the file and what file type for each of these entries (i.e. integer, real, etc). Generally, I two bytes into a variable for a integer and 4 bytes for a real. I can convert the integers simply enough. For a variable named id which is a bytarr(2) the integer is simply 256*id(1)+id(0). However, how can I convert the 4 byte arrays which are floats into the proper numbers? I seem to come up with gibberish if I do a simple float(val) where val=byytarr(4).

Any thoughts or tricks? All help much appreciated.

Subject: Re: Converting Byte Arrays
Posted by David Foster on Wed, 08 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Michael Slameczka wrote:
>
> Liam Gumley wrote:
> ....
>> record = swap_endian(record)
> ^^^^^^^^^^
>
> where is this function? It is not mentioned in my help-file of PV-Wave!
>
> cheers
> michael

It's an IDL function. But perhaps you have BYTEORDER().

Dave
--

~~~~~~ ~~~~~~
David S. Foster Univ. of California, San Diego
Programmer/Analyst Brain Image Analysis Laboratory
foster@bial1.ucsd.edu Department of Psychiatry
(619) 622-5892 8950 Via La Jolla Drive, Suite 2240
La Jolla, CA 92037
~~~~~~ ~~~~~~

Subject: Re: Converting Byte Arrays
Posted by davis on Wed, 08 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 06 Oct 1997 16:54:55 -0500, Liam Gumley <liam.gumley@ssec.wisc.edu>
wrote:
> One way to read a data file with mixed variable types (integers, floats) is by
> using an anonymous structure, e.g.
>
> record = { var1:0, var2:0L, var3:0.0 }
> openr, lun, file, /get_lun
> readu, lun, record
> free_lun, lun
> help, record.var1, record.var2, record.var3

This is not really portable because you need to know how the structure
and data are padded. For example, consider the C code:

```
#include <stdio.h>
typedef struct
{
  short s;
  long l;
}
Short_Long_Type;

int main (int argc, char **argv)
{
  Short_Long_Type sl;
  FILE *fp;

  sl.s = 1234;
  sl.l = 2147483647;

  fp = fopen ("sl.dat", "wb");
  fwrite ((char *) &sl, sizeof (Short_Long_Type), 1, fp);
  fclose (fp);
  return 0;
}
```

In general, the SIZE of data file that this program produces will
vary. Under Linux, it produces an 8 byte file, whereas under DOS the
resulting file is 6 bytes. If you want portable datafiles, never
directly write out the structures.

--John

Subject: Re: Converting Byte Arrays
Posted by popprice on Thu, 09 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Just a note. Tried the suggestions and the following worked best.

For integer data the 'BYTEORDER,var' worked perfectly. However, for float
you need 'swap_endian(var)'.

Thanks for the suggestions.

Subject: Re: Converting Byte Arrays
Posted by pit on Mon, 13 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

In article <popprice-ya023580000910971434490001@news.tiac.net>,
popprice@xinetics.com (Thomas Price) writes:
> Just a note. Tried the suggestions and the following worked best.
>
> For integer data the 'BYTEORDER,var' worked perfectly. However, for float
> you need 'swap_endian(var)'.

No, just byteorder, f_var, /lswap

Peter

--
~~~~~~
~~~~~~~~~~~~~~
Peter "Pit" Suetterlin http://www.uni-sw.gwdg.de/~pit
Universitaets-Sternwarte Goettingen
Tel.: +49 551 39-5048 pit@uni-sw.gwdg.de
-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * --
Come and see the stars! http://www.kis.uni-freiburg.de/~ps/SFB
Sternfreunde Breisgau e.V. Tel.: +49 7641 3492
_____ _____