Subject: Re: How does one read in simple binary files?
Posted by thompson on Sat, 13 Jan 1996 08:00:00 GMT
View Forum Message <> Reply to Message

bmac@igpp.llnl.gov (Bruce Macintosh) writes:

> This is an incredibly basic question: what's the simplest way
> to read in a simple binary file (ie an file of 16384 2-byte integers
> representing a 128x128 pixel image, for example), or slightly
> more complicated binary files (an array of 16384 IEEE 32-bit reals
> with a 100-byte header preceeding the pixel values, for example.)
> Are there any standard packages/routines for handling this kind of i/o,
> with or without byte and word swapping? All
> I can find in the manuals are routines for reading in ascii text, or
> various specialized formats, but nothing generic. Does "unformatted i/o"
> (readu, etc.) do this sort of input?

Yep.

All you would need to do for your example of a 128x128 array would be the following:

```
 IDL> OPENR, UNIT, 'filename1', /GET_LUN
 IDL> I_ARRAY = INTARR(128,128)
 IDL> READU, UNIT, I_ARRAY
 IDL> FREE_LUN, UNIT
```

or for your more complicated example

```
 IDL> OPENR, UNIT, 'filename2', /GET_LUN
 IDL> HEADER = BYTARR(100)
 IDL> READU, UNIT, HEADER
 IDL> F_ARRAY = FLTARR(16384) ;Or did you mean (128,128) again?
 IDL> READU, UNIT, F_ARRAY
 IDL> FREE_LUN, UNIT
```

You can also use ASSOC to do the same thing--check the documentation on the use of ASSOC.

If the data is written in a binary format different from that of the host computer, then it's a little more complicated. The BYTEORDER routine can be used to convert between standard and host-specific byte ordering, and even between IEEE and host floating point representations. The above two examples might also include the following lines after the read statements.

```
 IDL> BYTEORDER, I_ARRAY, /NTOHS

 IDL> BYTEORDER, F_ARRAY, /XDRTOF
```

Personally, I recommend using the routines IEEE_TO_HOST and HOST_TO_IEEE from
the Astronomy User's Library. That way, you don't have to figure out what the
proper keyword to use with BYTEORDER.

William Thompson

Subject: Re: How does one read in simple binary files?
Posted by kennealy on Sat, 13 Jan 1996 08:00:00 GMT

View Forum Message <> Reply to Message

Bruce, with IDL, reading binary files is quite simple. For example,
your case of a 128x128 image of 2-byte pixels could be handled as:

```
  image=intarr(128,128)
  openr,1,filename
  readu,1,image
  close,1
```

and that takes care of it. Your second case could be handled as

```
  header=bytarr(100)
  image=fltarr(128,128)
  openr,1,filename
  readu,1,header,image
  close,1
```

Hope this helps. In this newsgroup, you'll find people are very
helpful on all levels of questions.

Regards,
    Jack

Subject: Re: How does one read in simple binary files?
Posted by hahn on Mon, 15 Jan 1996 08:00:00 GMT

View Forum Message <> Reply to Message

Several good examples were given but ....

thompson@achilles.nascom.nasa.gov (William Thompson) wrote:

> Yep.

> All you would need to do for your example of a 128x128 array would be the
> following:

```
>  IDL> OPENR, UNIT, 'filename1', /GET_LUN
```

As the poster of the original question didn't not mention which
programing language was used to write the binary data I want to
point to an additional keyword for IDL's open procedures. It is
/fortran_77 which takes care of extra words the run time system
of Fortran adds for binary files. Binary Fortran files have the
record length of transferred record prepended and appended
to the record, *if the operating system is some Unix flavor*.
In following Fortran chunk will read 10 elements of a real array
from standard input and write it as one binary record on unit 27:

```
      real x(10)
      read(*,*) x
      write(27) x
      end
```

will result in 48 bytes being written: 4 bytes contain the record
length (40) as integer value, 40 data bytes follow, and then the
header is repeated.

> William Thompson

Norbert Hahn