Subject: Reading binary data (written using XDR) into IDL??
Posted by khan on Wed, 09 Aug 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Is there any way to read platform independent binary data written using
XDR into IDL? I'd like to get rid of the Bin->Ascii translators that we
have to use just for IDL.

thanks in advance
mumit -- khan@xraylith.wisc.edu

Subject: Re: Reading Binary
Posted by greg michael on Tue, 11 Jul 2006 12:43:10 GMT
View Forum Message <> Reply to Message

You can't do that. You can only read the file in units of a byte. If
your data is really in 12/13-bit units (which seems highly unusual) -
you'll have to extract them afterwards (paying close attention to
putting them back in the right order!). Where did you get this data?

regards,
Greg

Dirk1106@googlemail.com wrote:
> Hi everyone,
>
> I have a problem reading a binary file.
> The file begins with a Header of 6 unsingned Int.
> After this I have to dicive, if I want to read 12 bits oder 13 bits
> (not byte) as one record.
>
> Has soneone an idea, how to read the 12 or 13 bits best.
> After i read the bits I have so split them in the units, they stand
> for.
>
> Thanks
> .
> Greetz Dirk

Subject: Re: Reading Binary
Posted by Dirk1106@googlemail.c on Wed, 12 Jul 2006 08:39:09 GMT
View Forum Message <> Reply to Message

I get the Data from a collegue.
It is an LMF-File. I don`t know, why they made such un anusual format.

So there ist no chance, of getting the data directly right in IDL?

Subject: Re: Reading Binary
Posted by Peter Clinch on Wed, 12 Jul 2006 09:14:12 GMT
View Forum Message <> Reply to Message

Dirk1106@googlemail.com wrote:
> I get the Data from a collegue.
> It is an LMF-File. I don`t know, why they made such un anusual format.
>
> So there ist no chance, of getting the data directly right in IDL?

If there is it'll certainly be very awkward: even in C you're limited to
reading in byte sized chunks.

I imagine the best way is to read /everything/ that you might need into
memory in one go in a language that does bit operations fairly well
(like C) and then work through the data stream looking at fractions of
the bytes. And export from that into something readable by Normal
Computer Programs!

Pete.
--
Peter Clinch Medical Physics IT Officer
Tel 44 1382 660111 ext. 33637 Univ. of Dundee, Ninewells Hospital
Fax 44 1382 640177 Dundee DD1 9SY Scotland UK
net p.j.clinch@dundee.ac.uk http://www.dundee.ac.uk/~pjclinch/

Subject: Re: Reading Binary
Posted by greg michael on Wed, 12 Jul 2006 11:12:12 GMT
View Forum Message <> Reply to Message

I don't know what an LMF file is. Are you certain it's stored the way
you think? Even if they are 12/13-bit values, it's likely they'd be
stored in 16-bit chunks with the rest empty. But if your colleague made
it, he'll know how to read it...

Subject: Re: Reading Binary
Posted by Dirk1106@googlemail.c on Wed, 12 Jul 2006 13:01:51 GMT
View Forum Message <> Reply to Message

He reads it with C, but I want to work on it with IDL.
Now I have the C-Code and he always reads 12 or 13 bits.

So I also have to use C to read it and import it in IDL with
Call_External.

Subject: Re: Reading Binary
Posted by Peter Clinch on Wed, 12 Jul 2006 13:30:23 GMT
View Forum Message <> Reply to Message

Dirk1106@googlemail.com wrote:
> He reads it with C, but I want to work on it with IDL.
> Now I have the C-Code and he always reads 12 or 13 bits.

How does he do that in C?

It's been a while since I wrote any C and I'm more of a sysadmin these days so I'm wary of talking Rubbish here, but I've written a fair bit in my time and I can't recall anything in the stdio library that reads anything less than a single byte at a time.

Furthermore, a [unsigned] char is (or was, IIRC) a single byte and the smallest data type available in any flavour of C I've worked with, so even if you /can/ read 12 or 13 bits you'll have to *put* it in a short int at the very least. Although C has fairly comprehensive bitwaise operation support at the language level, I've never seen it read or store anything other than whole bytes.

Can you post the C fragment where he's reading 12/13 bits please?

Pete.
--
Peter Clinch Medical Physics IT Officer
Tel 44 1382 660111 ext. 33637 Univ. of Dundee, Ninewells Hospital
Fax 44 1382 640177 Dundee DD1 9SY Scotland UK
net p.j.clinch@dundee.ac.uk http://www.dundee.ac.uk/~pjclinch/

Subject: Re: Reading Binary
Posted by Dirk1106@googlemail.c on Wed, 12 Jul 2006 14:13:44 GMT
View Forum Message <> Reply to Message

for 12 bits

```
fread(&t0,   3, 1, fp);   byteswap(&t0,3);
fread(&tof,  1, 1, fp);
```

```
    fread(&pos1, 2, 1, fp);   byteswap(&pos1,2);
    fread(&pos2, 2, 1, fp);   byteswap(&pos2,2);
    fread(&ang,  2, 1, fp);   byteswap(&ang,2);
    fread(&e1,   1, 1, fp);
    fread(&e2,   1, 1, fp);
```

for 13 bits.

```
fread(&t,   8, 1, fp);   byteswap(&t,8);
    fread(&pos, 2, 1, fp);   byteswap(&pos,2);
    fread(&ang, 2, 1, fp);   byteswap(&ang,2);
    fread(&e,   1, 1, fp);
```

Subject: Re: Reading Binary
Posted by Paul Van Delst[1] on Wed, 12 Jul 2006 14:21:54 GMT
View Forum Message <> Reply to Message

Dirk1106@googlemail.com wrote:

```
>  for 12 bits
>
>       fread(&t0,   3, 1, fp);   byteswap(&t0,3);
>       fread(&tof,  1, 1, fp);
>       fread(&pos1, 2, 1, fp);   byteswap(&pos1,2);
>       fread(&pos2, 2, 1, fp);   byteswap(&pos2,2);
>       fread(&ang,  2, 1, fp);   byteswap(&ang,2);
>       fread(&e1,   1, 1, fp);
>       fread(&e2,   1, 1, fp);
>
>
>
>  for 13 bits.
>
>  fread(&t,   8, 1, fp);   byteswap(&t,8);
>       fread(&pos, 2, 1, fp);   byteswap(&pos,2);
>       fread(&ang, 2, 1, fp);   byteswap(&ang,2);
>       fread(&e,   1, 1, fp);
>
```

I thought fread worked in units of (8-bit) bytes? My C reference double check confirmed that also (from http://www.elook.org/programming/c/fread.html)

---
fread()

Syntax:

#include <stdio.h>
int fread(void *buffer, size_t size, size_t num, FILE *stream);

Description:
The function fread() reads num number of objects (where each object is size bytes) and places them into the array pointed to by buffer. The data comes from the given input stream. The return value of the function is the number of things read...use |feof()| or |ferror()| to figure out if an error occurs.
---

paulv

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx
Ph: (301)763-8000 x7748
Fax:(301)763-8545

Subject: Re: Reading Binary
Posted by greg michael on Wed, 12 Jul 2006 14:25:36 GMT
View Forum Message <> Reply to Message

I haven't used C in many years, but those look like bytes and not bits to me. You can't store positions and angles in 2 bits. And byteswapping bits?

If so, you can use normal data structures, defined for either of your two cases.

regards,
Greg

Dirk1106@googlemail.com wrote:
> for 12 bits
>
> fread(&t0, 3, 1, fp); byteswap(&t0,3);
> fread(&tof, 1, 1, fp);
> fread(&pos1, 2, 1, fp); byteswap(&pos1,2);
> fread(&pos2, 2, 1, fp); byteswap(&pos2,2);
> fread(&ang, 2, 1, fp); byteswap(&ang,2);
> fread(&e1, 1, 1, fp);
> fread(&e2, 1, 1, fp);
>
>

>
> for 13 bits.
>
> fread(&t, 8, 1, fp); byteswap(&t,8);
>   fread(&pos, 2, 1, fp); byteswap(&pos,2);
>   fread(&ang, 2, 1, fp); byteswap(&ang,2);
>   fread(&e, 1, 1, fp);

Subject: Re: Reading Binary
Posted by Peter Clinch on Wed, 12 Jul 2006 15:28:30 GMT
View Forum Message <> Reply to Message

Dirk1106@googlemail.com wrote:
> for 12 bits
>
>   fread(&t0, 3, 1, fp); byteswap(&t0,3);
>   fread(&tof, 1, 1, fp);
>   fread(&pos1, 2, 1, fp); byteswap(&pos1,2);
>   fread(&pos2, 2, 1, fp); byteswap(&pos2,2);
>   fread(&ang, 2, 1, fp); byteswap(&ang,2);
>   fread(&e1, 1, 1, fp);
>   fread(&e2, 1, 1, fp);

As others have pointed out, C's fread works in bytes, not bits.

From man fread on my linux box:
****
NAME
       fread, binary stream input/output

SYNOPSIS
       #include <stdio.h>
       size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

DESCRIPTION
       The function fread reads nmemb elements of data, each size bytes
  long, from the stream pointed to by stream, storing them at the
location given by ptr.

****

Note that the location given by ptr is cast to void* above but in
practice will be a holding space for a valid C data type, the smallest
of which is an 8 bit byte (char or unsigned char).

Pete.
--

Peter Clinch Medical Physics IT Officer
Tel 44 1382 660111 ext. 33637 Univ. of Dundee, Ninewells Hospital
Fax 44 1382 640177 Dundee DD1 9SY Scotland UK
net p.j.clinch@dundee.ac.uk http://www.dundee.ac.uk/~pjclinch/