Subject: Re: reading in a long line of data
Posted by Spon on Fri, 09 May 2008 14:31:43 GMT
View Forum Message <> Reply to Message

On May 9, 2:28 pm, David Fanning <n...@dfanning.com> wrote:
> Spon writes:
>> you once wrote a nice little programme called Read_Tab_Lines which may
>> be of some use here:
>
> Humm, I don't think so, since the problem is that the file
> is filling up the character buffer.
>
> I think the usual solution it to open this file in some
> kind of word processor, turn on word wrapping, and save
> the file with some reasonable length lines. But some word
> processors won't allow you to load a line of that length
> either.
>
> In that case, I think you have to make a large byte
> array and try to read the data that way, then process
> the byte array to obtain the data. I guess I would
> try something like this.
>
>   OpenR, lun, 'myfile.dat', /Get_Lun
>   info = FSTAT(lun)
>   data = BytArr(info.size)
>   ReadU, lun, data
>   Free_lun, lun
>   actualData = Float(StrSplit(String(data), /EXTRACT))
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Is there any reason not to use the following?

```
OpenR, lun, 'myfile.dat', /Get_Lun
info = FSTAT(lun)
data = FLTARR(Info.Size / 4L)
readu, lun, data
Free_lun, lun
```

There's something fundamental about this input buffer that I'm not getting here :-(

I have a file that looks like this:

```
8.06"89676e-002 9.0884782e-002 5.1953532e-002 8.1742041e-002
5.8772590e-002 6.7513607e-002 5.1806606e-002 "...
```

And I'm easily able to read 1.8 million floats from it at once, like this:

```
OPENR, Lun, File, /GET_LUN
FileSize = (FSTAT(Lun)).Size / 4L
Data = FLTARR(FileSize)
READU, Lun, Data
FREE_LUN, Lun
HELP, Data
IDL> DATA           FLOAT     = Array[1800720]
PRINT, Data[1800710:*]
IDL> 1.16934e-019 2.69980e-006 1.67846e-007 6.30809e-010 4.93642e-031
4.14105e-011 1.04315e-008 5.46591e+022 1.02555e-008 1.16934e-019
```

Surely, this is equivalent to (1800720 * 4) bytes, or (1800720 * 4 * 8) bits, or (1800720 * 13) ASCII chars, and therefore massively bigger than the buffer. Yet IDL has no problem doing this. What am I missing? Does the buffer only come into play with ReadF? How come?

Regards,
Chris