Subject: Re: reading an ascii file efficiently
Posted by b_gom on Fri, 27 Nov 2009 18:34:03 GMT
View Forum Message <> Reply to Message

Hopefully someone else can provide enlightenment about what technique will provide the fastest read times for this situation. Since your text file does not have fixed column widths, you probably can't avoid using read_ascii\ascii_template or reads.

As for the format codes, you could do this instead if you only care about the last column:

```
aux=fltarr(6)
last_val=0.
READS, str, aux, last_val, FORMAT='(6(F0,",",),F0)'
```

On Nov 27, 11:11 am, nata <bernat.puigdomen...@gmail.com> wrote:

> Hi guys,
>
> I'm reading an ascii file and I can do that using different methods.
> Now, I'm trying to use the most efficiently method. I do something
> like this:
>
> ```
> lines=FILE_LINES(file)
> rr=STRARR(lines)
> OPENR, lun, file, /GET_LUN
> READF, lun, rr
> FREE_LUN, lun
> ```
>
> Now i have all the information in rr variable. Each line have the
> following information:
> 280 , 0 , 280 , 0 , -58.085 , -32.616 , -32.000
> or
> 15 , 1 , 15 , 1 , -60.908 , -32.603 , -32.000
>
> And I need to return only the last value, so -32.000. I can use
> STRSPLIT or STRMID with STRPOS but is not efficient so I'm trying to
> use READS for each line. Something like this:
>
> ```
> aux=0.
> result=FLTARR(lines)
> FOR i=0l, lines-1 DO BEGIN
> READS, rr[i], aux, aux, aux, aux, aux, aux, aux,
> FORMAT='(F0,",",F0,",",F0,",",F0,",",F0,",",F0,", ",F0,",",)'
> result[i]=aux
> ```

> ENDFOR
>
> You can see that I don't know how to use the FORMAT keyword properly
> so maybe you have an idea of how to skip the first 6 values.
> Using a template, strsplit, etc. I found that this is the most
> efficient way to read this *@#$* file.
>
> Thanks if you can help me with this format or if you have a
> suggestion.
>
> nata