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

Subject: Re: reading an ascii file efficiently
Posted by jeanh on Fri, 27 Nov 2009 18:39:08 GMT
View Forum Message <> Reply to Message

nata 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
>

what about reading your data in a float array directly?
data = fltarr(nbCol,nbLines)
readf,lun,data

dataToKeep = data[nbCol-1,*]

Jean

Subject: Re: reading an ascii file efficiently
Posted by natha on Fri, 27 Nov 2009 19:02:45 GMT
View Forum Message <> Reply to Message

Hi all,

Thanks for your suggestions. Finally, the fastest read time for this
situation is the following one:

```
lines=FILE_LINES(file)
rr=STRARR(lines)
OPENR, lun, file, /GET_LUN
READF, lun, rr
FREE_LUN, lun

result=FLTARR(lines)
FOR i=0l, lines-1 DO BEGIN
  str_arr=STRSPLIT(rr[i],',',/EXTRACT)
  result[i]=FLOAT(str_arr[6])
ENDFOR
```

It's just 0.4 seconds faster than the previous solution. Thanks,
anyway

nata

Subject: Re: reading an ascii file efficiently
Posted by penteado on Fri, 27 Nov 2009 22:45:42 GMT
View Forum Message <> Reply to Message

On Nov 27, 5:02 pm, nata <bernat.puigdomen...@gmail.com> wrote:
> Hi all,
>
> Thanks for your suggestions. Finally, the fastest read time for this
> situation is the following one:
>
> lines=FILE_LINES(file)
> rr=STRARR(lines)
> OPENR, lun, file, /GET_LUN
> READF, lun, rr
> FREE_LUN, lun
>
> result=FLTARR(lines)
> FOR i=0l, lines-1 DO BEGIN
> str_arr=STRSPLIT(rr[i],',',/EXTRACT)
> result[i]=FLOAT(str_arr[6])
> ENDFOR
>
> It's just 0.4 seconds faster than the previous solution. Thanks,
> anyway
>
> nata

Another option, in IDL 7.1, is read_csv. It would take just one line
to write:

result=(read_csv(file)).(6)

With no need to open or close the file yourself, find out its length,
or declare the result variable.

Subject: Re: reading an ascii file efficiently
Posted by Heinz Stege on Sat, 28 Nov 2009 03:16:35 GMT
View Forum Message <> Reply to Message

On Fri, 27 Nov 2009 11:02:45 -0800 (PST), nata wrote:

> Hi all,
>
> Thanks for your suggestions. Finally, the fastest read time for this
> situation is the following one:
>
> lines=FILE_LINES(file)

> rr=STRARR(lines)
> OPENR, lun, file, /GET_LUN
> READF, lun, rr
> FREE_LUN, lun
>
> result=FLTARR(lines)
> FOR i=0l, lines-1 DO BEGIN
> str_arr=STRSPLIT(rr[i],',',/EXTRACT)
> result[i]=FLOAT(str_arr[6])
> ENDFOR
>
> It's just 0.4 seconds faster than the previous solution. Thanks,
> anyway
>
> nata

What do you think about replacing the for-loop (and the initial array definition). I would expect the following method to be significantly faster:

pos=strpos(rr,',',/reverse_search)
result=float(strmid(rr,transpose(pos)))

Have fun, Heinz

Subject: Re: reading an ascii file efficiently
Posted by Heinz Stege on Sat, 28 Nov 2009 03:27:57 GMT
View Forum Message <> Reply to Message

On Sat, 28 Nov 2009 04:16:35 +0100, Heinz Stege wrote:

> pos=strpos(rr,',',/reverse_search)
> result=float(strmid(rr,transpose(pos)))

Oh no, it doesn't work this way. We have to exclude the comma from the string:

pos=strpos(rr,',',/reverse_search)
result=float(strmid(rr,transpose(pos)+1))

Heinz

Subject: Re: reading an ascii file efficiently
Posted by penteado on Sat, 28 Nov 2009 03:58:37 GMT
View Forum Message <> Reply to Message

On Nov 28, 1:27 am, Heinz Stege <public.215....@arcor.de> wrote:
> On Sat, 28 Nov 2009 04:16:35 +0100, Heinz Stege wrote:
>> pos=strpos(rr,',',/reverse_search)
>> result=float(strmid(rr,transpose(pos)))
>
> Oh no, it doesn't work this way. We have to exclude the comma from
> the string:
>
> pos=strpos(rr,',',/reverse_search)
> result=float(strmid(rr,transpose(pos)+1))
>
> Heinz

Cool. I thought about it before, but thought that kind of thing would
not work, because strmid would take the pos array to mean to extract
multiple arrays, instead of associating each element of pos with each
element of rr. I had not realised that it is its first dimension that
is taken as the stride, until I saw your post, and wondered how that
could work, and why you transposed pod.

Nice to know it, I probably have some loops that I get rid of now.
Looking at strmid's documentation, I say it should have been a bit
more clear about this point.

Subject: Re: reading an ascii file efficiently
Posted by David Fanning on Sat, 28 Nov 2009 04:56:44 GMT
View Forum Message <> Reply to Message

pp writes:

> Nice to know it, I probably have some loops that I get rid of now.
> Looking at strmid's documentation, I say it should have been a bit
> more clear about this point.

Yeah, it's probably easier to understand this:

http://www.dfanning.com/code_tips/strmidvec.html

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.")

Subject: Re: reading an ascii file efficiently
Posted by penteado on Sat, 28 Nov 2009 05:04:13 GMT
View Forum Message <> Reply to Message

On Nov 28, 2:56 am, David Fanning <n...@dfanning.com> wrote:
> Yeah, it's probably easier to understand this:
>
> http://www.dfanning.com/code_tips/strmidvec.html
>
> 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.")

Thanks. So I was not the only one to be fooled by strmid and its
documentation.

Subject: Re: reading an ascii file efficiently
Posted by David Fanning on Sat, 28 Nov 2009 19:00:34 GMT
View Forum Message <> Reply to Message

pp writes:

> Thanks. So I was not the only one to be fooled by strmid and its
> documentation.

No, not at all. You are in EXTREMELY good company. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Subject: Re: reading an ascii file efficiently
Posted by natha on Mon, 30 Nov 2009 14:39:37 GMT
View Forum Message <> Reply to Message

On Nov 27, 10:27 pm, Heinz Stege <public.215....@arcor.de> wrote:
> On Sat, 28 Nov 2009 04:16:35 +0100, Heinz Stege wrote:
>> pos=strpos(rr,',',/reverse_search)
>> result=float(strmid(rr,transpose(pos)))
>
> Oh no, it doesn't work this way. We have to exclude the comma from
> the string:
>
> pos=strpos(rr,',',/reverse_search)
> result=float(strmid(rr,transpose(pos)+1))

It doesn't work... Check a simple example :
rr=['a,b,c','a,b,c']
pos=strpos(rr,',',/reverse_search)
print, strmid(rr,pos+1,99)

You have to do it for each component of the string array... Is not the
best solution, thanks anyway
nata

Subject: Re: reading an ascii file efficiently
Posted by jeanh on Mon, 30 Nov 2009 14:50:58 GMT
View Forum Message <> Reply to Message

nata wrote:
> On Nov 27, 10:27 pm, Heinz Stege <public.215....@arcor.de> wrote:
>> On Sat, 28 Nov 2009 04:16:35 +0100, Heinz Stege wrote:
>>> pos=strpos(rr,',',/reverse_search)
>>> result=float(strmid(rr,transpose(pos)))
>> Oh no, it doesn't work this way. We have to exclude the comma from
>> the string:
>>
>> pos=strpos(rr,',',/reverse_search)
>> result=float(strmid(rr,transpose(pos)+1))
>
> It doesn't work... Check a simple example :
> rr=['a,b,c','a,b,c']
> pos=strpos(rr,',',/reverse_search)

> print, strmid(rr,pos+1,99)
>
> You have to do it for each component of the string array... Is not the
> best solution, thanks anyway
> nata

you forgot to transpose the pos array

print, strmid(rr,transpose(pos+1),99)

Jean
