Subject: readcol procedure
Posted by shih I Chun on Tue, 10 Sep 2002 17:15:38 GMT
View Forum Message <> Reply to Message

Hi,

READCOL is a procedure in the IDL Astronomy Library. I usually use it to
read ASCII files for further analysis.

The program also indeicates the amount of lines in the file. My question
  then is how to retrieve this information so that I can use it in my
own program?

Thank you very much!

Subject: Re: readcol procedure
Posted by Don J Lindler on Tue, 10 Sep 2002 20:30:53 GMT
View Forum Message <> Reply to Message

"shih I Chun" <icshih@astro.soton.ac.uk> wrote in message
news:all9bs$9pp$1@aspen.sucs.soton.ac.uk...
> Hi,
>
> READCOL is a procedure in the IDL Astronomy Library. I usually use it to
> read ASCII files for further analysis.
>
> The program also indeicates the amount of lines in the file. My question
>   then is how to retrieve this information so that I can use it in my
> own program?
>

If you are only interested in how many valid lines are read, try:

```
readcol, 'filename', var1, ...
nlines = n_elements(var1)
```

Good luck,
Don

Subject: Re: readcol procedure
Posted by R.Bauer on Wed, 11 Sep 2002 06:33:24 GMT
View Forum Message <> Reply to Message

Don J Lindler wrote:

> "shih I Chun" <icshih@astro.soton.ac.uk> wrote in message
> news:all9bs$9pp$1@aspen.sucs.soton.ac.uk...
>
>> Hi,
>>
>> READCOL is a procedure in the IDL Astronomy Library. I usually use it to
>> read ASCII files for further analysis.
>>
>> The program also indeicates the amount of lines in the file. My question
>> then is how to retrieve this information so that I can use it in my
>> own program?
>>
>
>
>
> If you are only interested in how many valid lines are read, try:
>
> readcol, 'filename', var1, ...
> nlines = n_elements(var1)
>
> Good luck,
> Don
>
>

Dear all

this routine is platform dependent because it uses a unix shell command.
I like unix but it's not a problem for idl to determine this itselfs.

Perhaps you can try the routine file_line I have defined in 1996 and
which was improved later by Paul Krummel. You can find this routine by
David at http://www.dfanning.com/tip_examples/file_line.pro

Or the master one at our icg library.
The improvements are different. (Paul has added a lot of comments)

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/download/fileline.tar.gz
or as idl 5.5 binary
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/download/fileline.sav

If you have a look at the actual Ascii import thread you get much more
diskussions about this theme.

regards

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======

Subject: Re: readcol procedure
Posted by Wayne Landsman on Wed, 11 Sep 2002 14:54:10 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:

> this routine is platform dependent because it uses a unix shell command.
> I like unix but it's not a problem for idl to determine this itselfs.

numlines.pro (http://idlastro.gsfc.nasa.gov/ftp/pro/misc/numlines.pro called by readcol.pro) only spawns to the Unix 'wc' command if !VERSION.OS equals 'unix' (for speed). Otherwise it counts the
number of lines.

> Perhaps you can try the routine file_line I have defined in 1996 and
> which was improved later by Paul Krummel. You can find this routine by
> David at http://www.dfanning.com/tip_examples/file_line.pro

Hmm, I would have thought that reading the entire file into a byte array simply to count the number
of lines would be overkill. But in my quick tests, file_line.pro does seem to be faster than counting the number of lines, and almost as fast (on Unix) as spawning to 'wc'.

I have heard a rumor that there may be a standardized way of counting the number of lines in a

file
in the next release of IDL ;-)

--Wayne

Subject: Re: readcol procedure
Posted by David Fanning on Wed, 11 Sep 2002 21:12:23 GMT
View Forum Message <> Reply to Message

Wayne Landsman (landsman@mpb.gsfc.nasa.gov) writes:

> I have heard a rumor that there may be a standardized way of counting the number of lines in a file
> in the next release of IDL ;-)

And I hear it even has a name very similar to Riemer's little
File_Line program, which I think is too bad. Something like
COUNT_ROWS really makes more sense to me. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: readcol procedure
Posted by R.Bauer on Thu, 12 Sep 2002 06:42:22 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Wayne Landsman (landsman@mpb.gsfc.nasa.gov) writes:
>
>
>> I have heard a rumor that there may be a standardized way of counting the number of lines in a file
>> in the next release of IDL ;-)
>
>
> And I hear it even has a name very similar to Riemer's little
> File_Line program, which I think is too bad. Something like
> COUNT_ROWS really makes more sense to me. :-)

>
> Cheers,
>
> David

I don't know at the moment if I should be happy or not.
It's fine to see that's good routines would be implemented into the idl binary but always this is done I got the problem that's all of our sources using these routines need changes.
This happens last time by file_search. We have had nearly the same functionality in our routine but not the same parameters or keywords.
Internal routines are first called sources with the same name are ignored.

I believe they like to start with FILE_ because of the other file handling routines. I would prefer FILE_COUNT_ROWS if possible.
This gives more sense as the word I have choosen in the past.

I will have a look in the beta at this point. It's fine to be a beta tester if such new/old functions are implemented. So I have a bit more time to change our library.

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======

Subject: Re: readcol procedure
Posted by Liam E. Gumley on Thu, 12 Sep 2002 15:43:37 GMT
View Forum Message <> Reply to Message

Wayne Landsman wrote:
>
> Reimar Bauer wrote:
>

>> this routine is platform dependent because it uses a unix shell command.
>> I like unix but it's not a problem for idl to determine this itselfs.
>
> numlines.pro (http://idlastro.gsfc.nasa.gov/ftp/pro/misc/numlines.pro called by readcol.pro) only
> spawns to the Unix 'wc' command if !VERSION.OS equals 'unix' (for speed). Otherwise it counts the
> number of lines.
>
>> Perhaps you can try the routine file_line I have defined in 1996 and
>> which was improved later by Paul Krummel. You can find this routine by
>> David at http://www.dfanning.com/tip_examples/file_line.pro
>
> Hmm, I would have thought that reading the entire file into a byte array simply to count the number
> of lines would be overkill. But in my quick tests, file_line.pro does seem to be faster than
> counting the number of lines, and almost as fast (on Unix) as spawning to 'wc'.
>
> I have heard a rumor that there may be a standardized way of counting the number of lines in a file
> in the next release of IDL ;-)

I'm curious: Why does anyone need to count the number of lines in an
ASCII file? If it's to subsequently read the file, then the EOF function
can be used instead to tell you where the input file ends, and it
requires only one pass through the input file. There must be another
application that I don't know about. Or is it just easier to write code
that reads an ASCII file with a known number of lines?

Can anyone enlighten me?

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: readcol procedure
Posted by David Fanning on Thu, 12 Sep 2002 15:59:19 GMT
View Forum Message <> Reply to Message

Liam E. Gumley (Liam.Gumley@ssec.wisc.edu) writes:

> I'm curious: Why does anyone need to count the number of lines in an
> ASCII file? If it's to subsequently read the file, then the EOF function
> can be used instead to tell you where the input file ends, and it
> requires only one pass through the input file. There must be another
> application that I don't know about. Or is it just easier to write code
> that reads an ASCII file with a known number of lines?

>
> Can anyone enlighten me?

It's an aesthetic thing, Liam. That EOF stuff is
just...so...inelegant! :-)

Cheers,

DAvid

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: readcol procedure
Posted by R.Bauer on Thu, 12 Sep 2002 18:39:49 GMT
View Forum Message <> Reply to Message

Liam E. Gumley wrote:
> Wayne Landsman wrote:
>
>> Reimar Bauer wrote:
>>
>>
>>> this routine is platform dependent because it uses a unix shell command.
>>> I like unix but it's not a problem for idl to determine this itselfs.
>>
>> numlines.pro (http://idlastro.gsfc.nasa.gov/ftp/pro/misc/numlines.pro called by readcol.pro) only
>> spawns to the Unix 'wc' command if !VERSION.OS equals 'unix' (for speed). Otherwise it counts the
>> number of lines.
>>
>>
>>> Perhaps you can try the routine file_line I have defined in 1996 and
>>> which was improved later by Paul Krummel. You can find this routine by
>>> David at http://www.dfanning.com/tip_examples/file_line.pro
>>
>> Hmm, I would have thought that reading the entire file into a byte array simply to count the number
>> of lines would be overkill. But in my quick tests, file_line.pro does seem to be faster than
>> counting the number of lines, and almost as fast (on Unix) as spawning to 'wc'.
>>
>> I have heard a rumor that there may be a standardized way of counting the number of lines in a file

>> in the next release of IDL ;-)
>
>
> I'm curious: Why does anyone need to count the number of lines in an
> ASCII file? If it's to subsequently read the file, then the EOF function
> can be used instead to tell you where the input file ends, and it
> requires only one pass through the input file. There must be another
> application that I don't know about. Or is it just easier to write code
> that reads an ASCII file with a known number of lines?
>
> Can anyone enlighten me?
>
> Cheers,
> Liam.
> Practical IDL Programming
> http://www.gumley.com/

No it's not only asthetic.

if you use the eof method you have to read line by line. As you know idl
is an array orientated language so reading in an array is much faster.
It's really fast. If you have only 10 lines it doesn't matter but
sometimes we got datafiles of nearly 100.000 lines. In this case it is
very important.
The number of lines is one thing if you use some of our functions you
can determine how many columns the file has. Then it is quite easy to
define probably a float array[column,lines] and with one READF command
you get all the data at once.

The next trick is to determine the file itselfs about comments and data,
this all is done by the read_data_file itselfs. you have only to submit
a filename to this routine. The result is a structure of
header, separator, data, trailer.

The next version could return like read_Ascii a structure of the
parameters. But my routine determines itselfs the requiered minimum
datatypes of each column. e.g. if positive integer numbers less than 255
it will be defined as byte. If a decimal number has more than 6 digits
it must be double and so on. It needs no learn modus or other input
parameters as the datafile itself.

While read_ascii reads line by line it is extremly slow against this
routine.

More questions ?

best regards

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======

Subject: Re: readcol procedure
Posted by Liam E. Gumley on Thu, 12 Sep 2002 19:11:45 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:
[stuff deleted]
> if you use the eof method you have to read line by line. As you know idl
> is an array orientated language so reading in an array is much faster.
> It's really fast. If you have only 10 lines it doesn't matter but
> sometimes we got datafiles of nearly 100.000 lines. In this case it is
> very important.

How much time do you spend in determining the number of lines in the
file?

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: readcol procedure

Liam E. Gumley wrote:

> Reimar Bauer wrote:
> [stuff deleted]
>> if you use the eof method you have to read line by line. As you know idl
>> is an array orientated language so reading in an array is much faster.
>> It's really fast. If you have only 10 lines it doesn't matter but
>> sometimes we got datafiles of nearly 100.000 lines. In this case it is
>> very important.
>
> How much time do you spend in determining the number of lines in the
> file?

Dear Liam,

you are right there was a quite improvement which I missed in the past.
I did the following test to get no problems by internal caches.

I created on my USB Disk V1.1 which allows a max speed of 1MB/s
a file with transpose(sindgen(100000L)) characters.

Then I did a reboot so cache is empty
fileline needs 1.7 seconds to find 100000 lines

After this I rebooted the machine again
(or did someone know how to say linux to clear the filecache)

The following script needs only 1.63 seconds. So it's faster!!
(May be the difference comes from compiling two routines, fileline,filesize)

```
pro tr
openr,lun,'t1.txt',/get_lun
Z=''
count=0L
while not eof(lun) do begin
readf,lun,z
count=count+1
endwhile
print,count
end
```

I don't experimented if READS in addition to convert string to values
will need more time as reading again into rows and columns.
I believe reads takes more time.

and you are right too I am using the byte array in my read_data_file routine not only the number of lines which was calculated from fileline. The optional output is the bytarr.

There is another important routine bytes2strarr which converts the bytarr back into string. I have choosen this way to read the data only once. To get the routine faster is to read again the file because then the file is in cache and conversion could not be faster.

regards

Reimar

>
> Cheers,
> Liam.
> Practical IDL Programming
> http://www.gumley.com/

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html

---

Subject: Re: readcol procedure
Posted by Pavel A. Romashkin on Fri, 13 Sep 2002 15:39:11 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:
>
> if you use the eof method you have to read line by line. As you know idl
> is an array orientated language so reading in an array is much faster.
> It's really fast. If you have only 10 lines it doesn't matter but
> sometimes we got datafiles of nearly 100.000 lines. In this case it is
> very important.

I am sorry to disagree.
I routinely read large (60k-200k rows) ASCII files with unknown number of lines. I always use large arrays to read into and never ever use EOF with line by line reading.
All I have to do is to catch I/O error in case my buffer array is too

big as my reading approaches the end of file, then look up what size it
should have actually been, resize the buffer, then read the last portion
of file only. Reading a file with 80k lines using this method takes
about 0.1 s.
Take a look:
 http://www.ainaco.com/idl/idl_library/read_ascii_columns.pro
Cheers,
Pavel

Subject: Re: readcol procedure
Posted by David Fanning on Fri, 13 Sep 2002 15:50:05 GMT
View Forum Message <> Reply to Message

Pavel A. Romashkin (pavel_romashkin@hotmail.com) writes:

> I am sorry to disagree.
> I routinely read large (60k-200k rows) ASCII files with unknown number
> of lines. I always use large arrays to read into and never ever use EOF
> with line by line reading.
> All I have to do is to catch I/O error in case my buffer array is too
> big as my reading approaches the end of file, then look up what size it
> should have actually been, resize the buffer, then read the last portion
> of file only. Reading a file with 80k lines using this method takes
> about 0.1 s.
> Take a look:
> http://www.ainaco.com/idl/idl_library/read_ascii_columns.pro

Wow. You wrote this, Pavel!? :-)

Cheers,

David

P.S. Let's just say part of being an expert is knowing
who to steal code from. I'll be stealing some of this! :-)

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: readcol procedure
Posted by Pavel A. Romashkin on Fri, 13 Sep 2002 16:36:45 GMT

David Fanning wrote:
>
> You wrote this, Pavel!? :-)

If you don't have it on your site, where else would have I gotten it? I
think http://www.dfanning.com/ is the only site I steal from :-)
It is only 40 lines of a year-old code! I can type that much :-)
Cheers,
Pavel

Subject: Re: readcol procedure
Posted by R.Bauer on Fri, 13 Sep 2002 17:43:27 GMT

Pavel A. Romashkin wrote:
> Reimar Bauer wrote:
>
>> if you use the eof method you have to read line by line. As you know idl
>> is an array orientated language so reading in an array is much faster.
>> It's really fast. If you have only 10 lines it doesn't matter but
>> sometimes we got datafiles of nearly 100.000 lines. In this case it is
>> very important.
>
>
> I am sorry to disagree.
> I routinely read large (60k-200k rows) ASCII files with unknown number
> of lines. I always use large arrays to read into and never ever use EOF
> with line by line reading.
> All I have to do is to catch I/O error in case my buffer array is too
> big as my reading approaches the end of file, then look up what size it
> should have actually been, resize the buffer, then read the last portion
> of file only. Reading a file with 80k lines using this method takes
> about 0.1 s.
> Take a look:
> http://www.ainaco.com/idl/idl_library/read_ascii_columns.pro
> Cheers,
> Pavel

I don't understand where you are disagree.
I will try a comparison with the usb device and no file cache or
how should comparisons be done?

In principle you are using eof by an on_ioError condition.
This is nearly the same. Or not ?

You are reading portions and if you know the number of rows of the file
you can read this in one portions without an error.
Then it is nearly the same to my routine. The difference is only to my
routine that's it only need the filename and no column input and header
lines could be more or less then one line.

is this right ?

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======

Subject: Re: readcol procedure
Posted by Pavel A. Romashkin on Fri, 13 Sep 2002 19:30:44 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:
>
> I don't understand where you are disagree.

Well... Perhaps in that I think you don't have to know the number of
lines to use arrays for input? Or that EOF implies line by line reading?
Or in that I don't want to read the file more than once just to get
information other than file's contents?
Or, better yet, I think I just agree - as long as it works, who cares
how does it work. The least code and the faster, the better :-)
Cheers,
Pavel

Subject: Re: readcol procedure
Posted by R.Bauer on Sun, 15 Sep 2002 15:16:14 GMT

Reimar Bauer wrote:

> Pavel A. Romashkin wrote:
>> Reimar Bauer wrote:
>>
>>> if you use the eof method you have to read line by line. As you know idl
>>> is an array orientated language so reading in an array is much faster.
>>> It's really fast. If you have only 10 lines it doesn't matter but
>>> sometimes we got datafiles of nearly 100.000 lines. In this case it is
>>> very important.
>>
>>
>> I am sorry to disagree.
>> I routinely read large (60k-200k rows) ASCII files with unknown number
>> of lines. I always use large arrays to read into and never ever use EOF
>> with line by line reading.
>> All I have to do is to catch I/O error in case my buffer array is too
>> big as my reading approaches the end of file, then look up what size it
>> should have actually been, resize the buffer, then read the last portion
>> of file only. Reading a file with 80k lines using this method takes
>> about 0.1 s.
>> Take a look:
>> http://www.ainaco.com/idl/idl_library/read_ascii_columns.pro
>> Cheers,
>> Pavel
>
> I don't understand where you are disagree.
> I will try a comparison with the usb device and no file cache or
> how should comparisons be done?

I did a test today of both routines on my usb 1.1 device which could probaly
have a max speed of 1MByte/sec.
I learned that's umounting und remounting the device clears the cache.

To test only the reading speed I have both routines compiled by my compile
routine into a sav file which will loaded if the routine is called.
The testfile of 100000 lines by sindgen was altered in the first line with a
column name which is useable as a structure name for read_ascii_columns.

Result is:

read_ascii_columns: 2.048 seconds
read_data_file: 4.418 seconds

read_ascii_columns speed goes linear with the speed of the device.
by read_data_file the most time is used for the interpretation of the data

from bytearray to data.
I believe it is possible to improve the routine a bit but at the moment it's
for us fast enough.
It would be fine to see read_ascii_columns with an autodetection of headers
and columns and a translation of header description in useful tagnames.
e.g. H2(ppm) isn't possible to set as a tagname.

regards

Reimar

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html