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

============================================================ =======