Subject: Re: how to find number of lines in an ASCII file?
Posted by Martin Schultz on Thu, 20 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Robert S. Mallozzi wrote:
>
> In article <6rdfig$756@post.gsfc.nasa.gov>,
> jyli@redback.gsfc.nasa.gov (Jason Li) writes:
>> Hi,
>>
>> I have an ASCII text file that contains data in a nice tabular form,
>>
>> 0 28660 1827.1 72.7705 -158.8828 3388.0 22.3846 10.8545
>> 1 28661 1827.7 72.7701 -158.8752 3391.0 21.1213 10.6029
>> 2 28662 1828.3 72.7698 -158.8677 3394.0 19.8743 10.3546
>> .
>> .
>> .
>>
>> I want to read them all and save into an array:
>> data[8, numberOfLines]. But
>> I don't know numberOfLines in the file before hand.
>> What is the most efficient way to find that out?
>
> Here is yet another method:
>
> IDL does not need to know the number of lines in the file. It
> will dynamically increase the array for you. Assuming you know
> how many columns are in the file, I would read it into an array of
> structures as follows:
>
> data = {c1: 0L, c2: 0L, c3: 0.0, ..., c8: 0.0}
> data_in = data
>
> OPENR, FL, file, /GET_LUN
>
> READF, FL, data
> WHILE (NOT EOF (FL)) DO BEGIN
> READF, FL, data_in
> data = [data, data_in] <<<<<
> ENDWHILE
> FREE_LUN, FL
>
> Now data is an array of structures. The array length
> is the number of lines in the column. One caveat: this
> method won't work if any of the columns are STRING data.
>
Hi Robert,

As Kevin pointed out before, there may be some trouble with the marked line (although I must admit that I use this kind of dynamically increasing array quite often myself). Has anyone ever investigated the actual cost of this type of assignment? I imagine it increases more than linearily with the sized of the data (the number of lines) since the data block that has to be copied increases with each step.

In my readdata routine, I therefore allocate a very large array at the beginning (e.g. 20000 lines), and then truncate it to the actual number of lines in the end. Of course, one could become somewhat more sophisticated and alloacte blocks of, say 4000, entries at a time, read line by line, store it into the array, and allocate a new data block whenever you reach your line limit. Something like this (yeah, I couldn't let that pass ...):

```
 ------------------------------------------------------------ -------

pro dynalloc,maxc

  if (n_elements(maxc) eq 0) then maxc = 501

  MAXLINES = 100
  data = fltarr(MAXLINES,10)
  sample = data

  for i=0,maxc do begin   ; <<< replace loop by WHILE not eof()
     tmp = findgen(10)+i
     count = i
     ; see if new block must be allocated
     if (count mod MAXLINES eq 0) then $
        data = [ data, sample ]
     ; store one data line
     data[count,*] = transpose(tmp)
  endfor                 ; <<<

  data = data[0:count-1,*]
  help,data

end



pro slowalloc,maxc

  if (n_elements(maxc) eq 0) then maxc = 501

  for i=1,maxc do begin   ; <<< replace loop by WHILE not eof()
```

```
    tmp = findgen(10)+i
    count = i
    if (count eq 1) then data = transpose(tmp) $
    else data = [ data, transpose(tmp) ]
  endfor                ; <<<

  help,data

end



pro testalloc,maxc

  if (n_elements(maxc) eq 0) then maxc = 501

  t0 = systime(1)
  dynalloc,maxc
  t1 = systime(1)
  slowalloc,maxc
  t2 = systime(1)

  print,'DYNALLOC: ',t1-t0,'  SLOWALLOC: ',t2-t1

end
```

Here are a few test results:

```
IDL> testalloc,500
DATA          FLOAT     = Array[500, 10]
DATA          FLOAT     = Array[500, 10]
DYNALLOC:      0.022094965  SLOWALLOC:      0.039510012
IDL> testalloc,5000
DATA          FLOAT     = Array[5000, 10]
DATA          FLOAT     = Array[5000, 10]
DYNALLOC:       0.26451409  SLOWALLOC:       6.1116600
```

Martin.

--

Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
---------- -------