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

Jason Li wrote in message <6rdfig$756@post.gsfc.nasa.gov>...
> 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?
>
> On UNIX, I can pass number of lines information back from wc command. Of
> course the same code won't work a on Mac. Please help.

Below are two functions that I use regularly. I have used them on both
Windows and UNIX and they work fine. They should be platform independent and
I find them to be efficient!
They were originally written by R. Bauer and I have modified them.

Hope this helps

Cheers Paul Krummel

~- ~-~-~-~
CSIRO Atmospheric Research
Private Bag #1 Aspendale Victoria 3195 Australia
e-mail: paul.krummel@dar.csiro.au
ph: +61 3 9239 4568 fax: +61 3 9239 4444
~- ~-~-~-~

```
;+
; NAME:
; FILE_LINE
;
; PURPOSE:
; This function finds the number of lines in an ASCII data file.
; It should be platform independent (well Windows and UNIX at least!).
;
; CATEGORY:
```

```
; Read/Write OR Input/Output.
;
; CALLING SEQUENCE:
;
; Result = FILE_LINE(File_name)
;
; INPUTS:
; File_name: The name of the file to find the number of lines in.
;    This can now be an array of filenames!!
;
; OUTPUTS:
; This function returns the number of lines in a file. If the input is
; an array of filenames then the output is a long array with length of
; the input array plus 1. This array will contain the number of lines
; in each of the files plus the total number of lines for all the files
; combined.
;
; PROCEDURE:
; Calls FILE_SIZE.
;
; EXAMPLE:
; To find the number of lines in the file test.dat enter:
; IDL> out=FILE_LINE('test.dat')
;   OR
; IDL> files=['test1.dat','test2.dat','test3.dat']
; IDL> print,FILE_LINE(files)
;  15 20 30 65
;
; MODIFICATION HISTORY:
; Copyright    R.Bauer 2. Jan. 1996
;
; Modified by Paul Krummel, 12 February 1997, CSIRO Division of Atmospheric
; Research. Changed error messages to english and modified them. Added
complete
; header information and usage information (help keyword).
; Added some more comments and a check to see if filename is a string.
;
; Modified by Paul Krummel, 8 January 1998. Has been considrably modified
; and can now take in an array of filenames or just one file name.
;-

FUNCTION FILE_LINE, filename, help=help
;
; =====>> HELP
;
on_error,2
if (N_PARAMS(0) lt 1) or keyword_set(help) then begin
   doc_library,'FILE_LINE'
```

```
   if N_PARAMS(0) ne 1 and not keyword_set(help) then $
              message,'Incorrect number of parameters, see above for
usage.'
   return,-1
;
   ioerr:
   message,'Error reading file, '+filename+', does not exist' ,/inform
   return,-1
   filerr:
   message,'Filename(s) must be of type string' ,/inform
   return,-1
endif
;
; ++++
; Check the filename to see if it is a string, if not display error message.
if type(filename) ne 7L then goto, filerr
;
; ++++
; Find if the number of elements in the input (filename)
num=n_elements(filename)
;
; ++++
; If the input "filename" is an array loop around each file else just
; process as single filename.
CASE 1 of
;
; ****** Just a string ******
 num eq 1: BEGIN
; ++++
; Use the filesize function to find the number of bytes in the file. If
there
; are no bytes then the file does not exist, print error message.
  byt=file_size(filename)
  if byt eq -1 then goto, ioerr
;
; ++++
; Set up byte array to length of the file.
  bytes=bytarr(byt)
;
; ++++
; Open the file name, if it does not exist, display error message.
  openr,lun,filename,/get_lun,error=err
  if err ne 0 then goto, ioerr
;
; ++++
; Read the file into the byte array.
  readu,lun,bytes
  free_lun,lun
```

```
;
; ++++
; Find where we have a line feeds and count them.
  line=where(bytes eq 10B,count_line)
;
; ++++
   END
;
; ++++
; ****** An array of strings ******
 num gt 1: BEGIN
; ++++
; Set up the ouput array, one extra that will contain the total of all the
files.
  count_line=lonarr(num+1)
;
; Loop around the filenames
  For i=0,num-1 do begin
;
; ++++
; Use the filesize function to find the number of bytes in the file. If
there
; are no bytes then the file does not exist, print error message.
   byt=file_size(filename[i])
   if byt eq -1 then goto, ioerr
;
; ++++
; Set up byte array to length of the file.
   bytes=bytarr(byt)
;
; ++++
; Open the file name, if it does not exist, display error message.
   openr,lun,filename[i],/get_lun,error=err
   if err ne 0 then goto, ioerr
;
; ++++
; Read the file into the byte array.
   readu,lun,bytes
   free_lun,lun
;
; ++++
; Find where we have a line feeds and count them.
   line=where(bytes eq 10B,cnt_line)
   count_line[i]=cnt_line
; ++++
  endfor
;
; ++++
```

```
; Now total all the file lines
  count_line[num]=total(count_line[0:num-1])
;
; ++++
   END
;
ENDCASE
;
; ++++
; Return the number of lines in the file.
return,count_line
;
; ++++
END
```

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

```
;+
; NAME:
; FILE_SIZE
;
; PURPOSE:
; This function finds the number of bytes in an ASCII data file.
; It should be platform independent (well Windows and UNIX at least!).
;
; CATEGORY:
; Read/Write OR Input/Output.
;
; CALLING SEQUENCE:
;
; Result = FILE_SIZE(File_name)
;
; INPUTS:
; File_name: The name of the file to find the number of bytes in.
;
; OUTPUTS:
; This function returns the number of bytes in a file.
;
; PROCEDURE:
; Uses fstat to find information about the opened unit number.
;
; EXAMPLE:
; To find the size in bytes of the file test.dat enter:
; IDL> out=FILE_SIZE('test.dat')
;
; MODIFICATION HISTORY:
; Copyright    R.Bauer 2. Jan. 1996
```

```
; The idea to use fstat instead of spawn ls -l was given by Phil Williams.
;
; Modified by Paul Krummel, 12 February 1997, CSIRO Division of Atmospheric
; Research. Changed error messages to english and modified them. Added
complete
; header information and usage information (help keyword).
; Added some more comments and a check to see if filename is a string.
;-

FUNCTION FILE_SIZE, filename, help=help
;
; =====>> HELP
;
on_error,2
if (N_PARAMS(0) lt 1) or keyword_set(help) then begin
   doc_library,'FILE_SIZE'
   if N_PARAMS(0) ne 1 and not keyword_set(help) then $
            message,'Incorrect number of parameters, see above for
usage.'
   return,-1
;
   ioerr:
   message,'Error reading file, '+filename+', does not exist' ,/inform
   return,-1
   filerr:
   message,'Filename must be of type string' ,/inform
   return,-1
endif
;
; ++++
; Check the filename to see if it is a string, if not display error message.
if type(filename) ne 7L then goto, filerr
;
; ++++
; Open the file name, if it does not exist, display error message.
openr, lun, filename, /get_lun, error=err
if err ne 0 then goto, ioerr
;
; ++++
; Use the fstat function to find information about the opened unit.
stats = fstat(lun)
free_lun, lun
;
; ++++
; Return the file size!
return, stats.size
;
; ++++
```

end