Subject: Help: Countlines on Win 95 version of IDL
Posted by PDW on Tue, 09 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

I am converting programs from Unix version of IDL to Windows 95, and have a problem with routines using SPAWN and "wc -l" to get the number of lines in a file in order to dimension an array. How can this be done on the PC version?

Subject: Re: Help: Countlines on Win 95 version of IDL
Posted by R. Bauer on Wed, 17 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

PDW wrote:

> I am converting programs from Unix version of IDL to Windows 95, and have a
> problem with routines using SPAWN and "wc -l" to get the number of lines in
> a file in order to dimension an array. How can this be done on the PC
> version?

Hy Paul,

I have included two of my routines.
filesize uses fstat to get the byte size of the file
This is used by fileline to read the file at once in a byte array.
Then the 10b are counted and returned.

This routines works on win95 and unix.

R.Bauer

```
; $Id: filesize.pro,v 1. 1997/06/13 13:31:13 RwB ICG-1 $
;
; Copyright (c) 1997, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
;+
; NAME:
; filesize
;
; PURPOSE:
; This function founds the bytelength of an ascii file
;
; CATEGORY:
;   DATAFILES/FILE
```

```
;
; CALLING SEQUENCE:
;   Result=filesize(file_name)
;
; INPUTS:
; file_name: the name of an ascii file
;
;
; OUTPUTS:
; This function returns the number of bytes of an ascii file
;
;
; EXAMPLE:
;   Result=filesize('test.asc')
;
; MODIFICATION HISTORY:
;  Written by: R.Bauer (ICG-1), Oct. 1996
;-

function filesize, filename


;debug,'<filesize> 1.1 RB  1997-Sep-16'

if n_params(0) lt 1 then begin
   help:   print, ' Diese Hilfe kommt mit a=filesize().'
   print,' '
   print,' stellt fest wieviele Bytes in einer Datei sind.'
   print,''
   print,'Example'
   print,"a=filesize('testfile.asc')"
    print,'-------------------------------------------------- --'
   return,-1
   help_open: print,'(filesize) Das File: ',filename,' gibt es nicht.'
   return,-1
ENDIF

openr, lun, filename, /get_lun,error=err
if err  ne 0 then goto, help_open
stats = fstat(lun)
free_lun, lun

return, stats.size
end



; $Id: fileline.pro,v 1. 1997/06/13 13:31:13 RwB ICG-1 $
```

```
;
; Copyright (c) 1997, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
;+
; NAME:
; fileline
;
; PURPOSE:
; This function founds the number of lines of an ascii file
;
; CATEGORY:
; DATAFILES/FILE
;
;
; CALLING SEQUENCE:
;   Result=fileline(file_name)
;
; INPUTS:
; file_name: the name of an ascii file
;
;
; OUTPUTS:
; This function returns the number of lines of an asii file
;
;
; EXAMPLE:
;   Result=fileline('test.asc')
;
; MODIFICATION HISTORY:
;  Written by: R.Bauer (ICG-1), Oct. 1996
;-


function fileline, filename


;debug,'<fileline> 1.1 RB  1997-Sep-16'

if n_params(0) lt 1 then begin
   help:   print, ' Diese Hilfe kommt mit a=fileline().'
   print,' '
   print,' stellt fest wieviele Zeilen in einer ASCII Datei sind.'
   print,''
   print,'Example'
   print,"a=fileline('testfile.asc')"
    print,'-------------------------------------------------- --'
   return,-1
```

```
   help_open: print,'(fileline) Das File: ',filename,' gibt es nicht.'
   return,-1
ENDIF

 byt=filesize(filename)


 if byt eq -1 then goto, help_open

lesefeld=bytarr(byt)

openr,lun,filename,/get_lun,error=err
if err  ne 0 then goto, help_open
 readu,lun,lesefeld
 close,lun

 free_lun,lun
 line=where(lesefeld eq 10B,count_line)

return,count_line
END
```

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: Help: Countlines on Win 95 version of IDL
Posted by Michael Werger on Wed, 17 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

PDW wrote:

> I am converting programs from Unix version of IDL to Windows 95, and have a
> problem with routines using SPAWN and "wc -l" to get the number of lines in
> a file in order to dimension an array. How can this be done on the PC
> version?

Try this:

```
;+
; Name:
; nlines
```

```
; Purpose:
; Return the number of lines in a file
; Usage:
; nl = nlines(file)
; Inputs:
; file = file to scan
; Optional Inputs or Keywords:
; help = flag to print header
; Outputs:
; nl = number of lines in the file.
; Common blocks:
; none
; Procedure:
; Assume ASCII data and read through file.
; Modification history:
; write, 24 Feb 92, F.K.Knight
;-
function nlines,file,help=help
;
; =====>> HELP
;
on_error,2
if keyword_set(help) then begin & doc_library,'nlines' & return,0 & endif
;
; =====>> LOOP THROUGH FILE COUNTING LINES
;
tmp = ' '
nl = 0
on_ioerror,NOASCII
if n_elements(file) eq 0 then file = pickfile()
openr,lun,file,/get_lun
while not eof(lun) do begin
  readf,lun,tmp
  nl = nl + 1
  endwhile
close,lun
free_lun,lun
NOASCII:
return,nl
end
```

--
Michael Werger ESA ESTEC & Praesepe B.V.
Astrophysics Division mwerger@estec.esa.nl
Postbus 299 http://astro.estec.esa.nl
2200 AG Noordwijk +31 71 565 3783 (Voice)
The Netherlands +31 71 565 4690 (FAX)

Subject: Re: Help: Countlines on Win 95 version of IDL
Posted by mgs on Thu, 18 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <34981500.572F0480@fz-juelich.de>, "R. Bauer"
<r.bauer@fz-juelich.de> wrote:

> PDW wrote:
>
>> I am converting programs from Unix version of IDL to Windows 95, and have a
>> problem with routines using SPAWN and "wc -l" to get the number of lines in
>> a file in order to dimension an array. How can this be done on the PC
>> version?

Here's another "Fileline". Same purpose and name, different author. Works
cross-platform (UNIX, Mac and PC) and shouldn't gag on huge files.

```
 ;############################################################## ################
; File Name:  %M%
; Version:    %I%
; Author:     Mike Schienle
; Orig Date:  97-03-09
; Delta Date: %G% @ %U%
 ;############################################################## ################
; Purpose:
; History:
 ;############################################################## ################
; %W%
 ;############################################################## ################

FUNCTION FileLine, file, LINETERM=lineTerm
   ;   open the file and check for errors
   OpenR, lun, file, /Get_Lun, Error=err, BufSize=2L^20
   IF NOT (err) THEN BEGIN
      ;   if line terminator is not specified
      ;   assume the file is native to the OS
      ;   UNIX uses line feed
      ;   Mac uses carriage return
      ;   PC uses both
      IF (N_Elements(lineTerm) EQ 0L) THEN $
         IF (StrLowCase(!Version.OS_Family) EQ 'unix') THEN $
            lineTerm = 10b  $
         ELSE $
            lineTerm = 13b

      ;   use FStat function to get access to file size
      fInfo = FStat(lun)

      ;   set an arbitrary size threshold
```

```
lThresh = 2L^20

;   compare the file size to the threshold
IF (fInfo.Size LT lThresh) THEN BEGIN
   ;   allocate array to read entire file into memory
   aData = BytArr(fInfo.Size, /NoZero)
   ReadU, lun, aData

   ;   find the number of characters matching line terminator
   ;   Provided in lCount. Not interested in positions.
   aPos = Where(aData EQ lineTerm, lCount)
ENDIF ELSE BEGIN
   ;   file arbitrarily too large to read into a single array
   ;   determine the number of loops to be processed
   lLoops = fInfo.Size / lThresh

   ;   associate a variable for easy access
   mVar = Assoc(lun, BytArr(lThresh, /Nozero))

   ;   init a var to hold the terminator count
   lCount = 0L

   ;   loop through the data file one threshold chunk at a time
   FOR i = 0L, (lLoops - 1) DO BEGIN
      ;   find the number of characters matching line terminator
      ;   Provided in lPos. Not interested in positions.
      aPos = Where(mVar(i) EQ lineTerm, lPos)

      ;   running sum of terminator counts
      lCount = lCount + lPos
   ENDFOR

   ;   get amount of file remainder
   lModSize = (fInfo.Size MOD lThresh)

   ;   read remainder of file if necessary
   IF (lModSize NE 0) THEN BEGIN
      ;   create an array to read in remaining data
      aData = BytArr(lModSize, /NoZero)

      ;   put the data in the file
      ReadU, lun, aData

      ;   find the number of characters matching line terminator
      ;   Provided in lPos. Not interested in positions.
      aPos = Where(aData EQ lineTerm, lPos)

      ;   running sum of terminator counts
```

```
            lCount = lCount + lPos
          ENDIF
        ENDELSE

        ;   free and close the lun
        Free_Lun, lun
    ENDIF ELSE BEGIN
        ;   an error occurred
        lCount = -1
    ENDELSE

    ;   return the lCount to the calling program
    Return, lCount
END
```

--
Mike Schienle Interactive Visuals
mgs@sd.cybernex.net http://ww2.sd.cybernex.net/~mgs/

Subject: Re: Help: Countlines on Win 95 version of IDL
Posted by David Foster on Thu, 18 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

R. Bauer wrote:
>
> PDW wrote:
>
>> I am converting programs from Unix version of IDL to Windows 95, and have a
>> problem with routines using SPAWN and "wc -l" to get the number of lines in
>> a file in order to dimension an array. How can this be done on the PC
>> version?
>

If you need the number of the lines to dimension an array, do you
plan to read the file into a STRARR? If so, you might want to
try FILE_STRARR.PRO, which reads an ASCII file into a STRARR:

```
========== FILE_STRARR.PRO ============================================

;  FILE_STRARR.PRO    5-27-94
;
; Routine to read text file and return a STRARR containing
; the lines of text. Returns a STRARR(2) containing the null
; string '' and the value of !err_string if an I/O error is encountered
;
; This code adapted from XDISPLAYFILE
```

```
FUNCTION file_strarr, fname

ON_IOERROR, IO_ERROR

openr, unit, fname, /get_lun, error=err
if (err ne 0) then begin
   return, ['ERROR', !err_string]
endif else begin
   max_lines = 1000
   a = strarr(max_lines)
   i = 0
   c = ''
   while (not eof(unit)) do begin
      readf, unit, c
      a(i) = c
      i = i + 1
      if (i eq max_lines - 2) then begin
         a = [a, strarr(1000)]
         max_lines = max_lines + 1000
      endif
   endwhile
   a = a(0:i-1)
   free_lun, unit
   return, a
endelse

IO_ERROR:
   message, 'Error reading from file ' + fname, /continue
   return, ['ERROR', !err_string]

END

======== .doc file ============================================
 FILE_STRARR

 Use this routine to read a text file and return
 a STRARR variable containing the lines of text.
 This is useful when displaying text-files in
 text widgets. If an I/O error such as file-not-
 found occurs then returns a STRARR(2) containing
 the null string '' and the value of the IDL
 system variable !ERR_STRING.


 Calling Sequence

   Text = FILE_STRARR(Filename)
```

Arguments

Filename

The name of the file containing the text
you wish to return as a STRARR.

Outputs

Returns the lines of text from the file as
a STRARR variable. If an I/O error occurs
then this will contain two elements: the
string 'ERROR' and the value of the IDL
system variable !ERR_STRING when the error
occured.

Example

```
Text = FILE_STRARR(Filename)

if (Text(0) eq 'ERROR' and $
      n_elements(Text) eq 2) then $
   message, 'Error reading file ' + Filename
```

--

~~~~~~ ~~~~~~~
David S. Foster Univ. of California, San Diego
Programmer/Analyst Brain Image Analysis Laboratory
foster@bial1.ucsd.edu Department of Psychiatry
(619) 622-5892 8950 Via La Jolla Drive, Suite 2240
La Jolla, CA 92037
~~~~~~ ~~~~~~~