Subject: Re: how to find number of lines in an ASCII file?
Posted by R. Bauer on Fri, 28 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Jason Li wrote:

> 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?
>
> 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.
>
> many thanks
>

Hi Jason,

a few months ago I answered someone else in this way.
We have a lot more routines for file handling.

Reimar

;

;+
; NAME:

```
;  filesize
;
; PURPOSE:
;   The result of this function is 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


   IF N_PARAMS(0) LT 1 THEN BEGIN
      HELP:   MESSAGE, 'result=filesize()',/cont
       MESSAGE,'---------------------------------------------------- ----',/cont
      RETURN,-1
      help_open: MESSAGE,'file: '+filename+' not found',/cont
      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
```

---

```
;
; Copyright (c) 1997, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made.  This
; routine is provided as is without any express or implied warranties
; whatsoever.
;
;+
; NAME:
;  fileline
;
; PURPOSE:
;   This function returns 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
;
; EXAMPLE:
;   Result=fileline('test.asc')
;
; MODIFICATION HISTORY:
;   Written by: R.Bauer (ICG-1), Oct. 1996
;-


FUNCTION fileline, filename

  IF n_params(0) LT 1 THEN BEGIN
    HELP: message, "result=fileline('test.dat')",/cont

 message,'-------------------------------------------------- ----',/cont
    RETURN,-1
    help_open: message,'file: '+filename+' not found.',/cont
    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

  FREE_LUN,lun
  if lesefeld(byt-1) ne 10b then lesefeld=[lesefeld,10b]
  line=where(lesefeld EQ 10B,count_line)

  RETURN,count_line
END
```

---

Example:

```
fltarr=fileline('test.dat')
readf,10,fltarr
```

---



```
;+
; NAME:
;   file_exist
;
; PURPOSE:
;   The result of this function is 1 if a file exist and 0 if not
;
; CATEGORY:
;   DATAFILES
;
; CALLING SEQUENCE:
;   Result=file_exist(file_name)
;
; INPUTS:
;   file_name: The name of the File
```

```
;
; OUTPUTS:
;   This function returns 1 if the file exist and 0 if not
;
; EXAMPLE:
;   result=file_exist('otto.nc')
;
; MODIFICATION HISTORY:
;       Written by:     R.Bauer (ICG-1),  1998-May-18
;-
FUNCTION file_exist,file_name
  OPENR,lun,file_name,err=err,/GET_LUN
  IF n_elements(lun) GT 0 THEN FREE_LUN,lun
  IF err NE 0 THEN RETURN,0 ELSE RETURN,1
END

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

```
; seperator: the seperator between the numbers
; comments:  the number of comment lines before the data
;
; RESTRICTIONS:
; All inputs by extra: this means no shorter inputs possible
;
; EXAMPLE:
;    if a file is given like
;    1 2
;    3 4
;    result=get_columns('file.dat')
;
;
; MODIFICATION HISTORY:
;       Written by:     R.Bauer (ICG-1), 1998-Jun-05
;-


FUNCTION get_columns, file,_extra=extra

   IF (N_PARAMS(0) LT 1)  THEN BEGIN
      PRINT,'<get_columns> result=get_columns(file)'
      RETURN, -1
   ENDIF
   IF not is_structure(extra) THEN extra={not_defined:1}
   IF is_tag(extra,'seperator') eq 0  THEN seperator=' ' else
seperator=extra.seperator

   if is_tag(extra,'comments') then if extra.comments gt 0 then
comments=strarr(extra.comments)
   first_line=''

   IF file_exist(file) THEN BEGIN
      OPENR, lun, file,/GET_LUN
      if n_elements(comments) gt 0 then READF,lun,comments
      READF,lun,first_line
      FREE_LUN, lun
      result=N_ELEMENTS(STR_SEP(first_line,seperator))
      RETURN, result
      ENDIF ELSE BEGIN
      PRINT,'File: '+file +''doesn't exist''
      RETURN,-1
   ENDELSE

END
```

---

```
;
; Copyright (c) 1997, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made.  This
; routine is provided as is without any express or implied warranties
; whatsoever.
;
;+
; NAME:
;  is_tag
;
; PURPOSE:
;   This function returns 1 if a tagname is defined in a structure
;
; CATEGORY:
;   PROG_TOOLS/STRUCTURES
;
; CALLING SEQUENCE:
;   Result=is_tag(structure,tagname)
;
; INPUTS:
;   structure:  the structure
;   tagname:    the tagname as string which should be searched in structure
;
; OUTPUTS:
;   Result will be 1 or 0
;
;
; EXAMPLE:
;   print,is_tag(inhalt,'param')
;   1
;
; MODIFICATION HISTORY:
;       Written by:     R.Bauer (ICG-1) , Sep. 2 1996
;               F.Rohrer (ICG-3), Mai 15 1997 downgrade to idl 3.6.1
;               R.Bauer 1998-Jul-05 previously named as find_tag now renamed for
better consistens
;               R.Bauer 1998-Jul-05 upgraded to idl 5.1
;
;-


FUNCTION is_tag,struct,tag_such

  IF N_PARAMS(0) LT 2 THEN BEGIN
    HELP:   message, " PRINT,result=is_tag(inhalt,'file')",/cont
```

```
    RETURN,-1
    help_struct: message,'structure not defined',/cont
    RETURN,-1
  ENDIF

  count = 0
  tag_such=STRUPCASE(tag_such)
  IF N_ELEMENTS(struct) GT 0 THEN BEGIN
    tags=TAG_NAMES(struct)
    a=WHERE(tags EQ tag_such,count)
  ENDIF

  IF N_ELEMENTS(struct) LT 1 THEN GOTO, help_struct

  RETURN, count
END
```

---