Subject: Re: how to find number of lines in an ASCII file?
Posted by LC's No-Spam Newsread on Mon, 31 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Jason Li wrote:

>> I have an ASCII text file that contains data in a nice tabular form,
>> 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?

I don't know if it is the most efficient (I doubt it), but I find easy the
following way.

(1) I use a csh script to append one line to the top of the file telling
how many header lines, how many data lines and columns there are

xasasc filename

(2) I use an IDL procedure to read the data in a structure of arrays, one
array being an entire column, one optionally can name the columns

xasasc,'filename',strucname or
xasasc,'filename',strucname,['name1','name2'.....]

Use of the software below is free, adapt as you wish, just remember I did
it first, no warranties implied, etc. etc.

Lucio Chiappetti - IFCTR/CNR - via Bassini 15 - I-20133 Milano (Italy)
For more info : http://www.ifctr.mi.cnr.it/~lucio/personal.html
---

This is the xasasc shell script

```
#! /bin/csh -f
#
# transform an ASCII tabular file into a "XAS ASCII file"
# this is mainly for reading into IDL
# a XAS ASCII file is defined as having a first record structured as
#
# on OSF Alpha changed grep -s to grep -s -q to suppress all echo
#
# XAS1ASC2GEN31234 n_header_records n_data_records n_columns
#
# followed by some (or none) header records
# and some data records in free-format containing only numbers in columnar arrangement
#
```

```
# check if file already contains magic number in first line
  head -1 $1 | grep -s 'XAS1ASC2GEN31234'
  if ($status == 0) exit 1
#
# determine number of records in file ($nlines[1])
  set nlines = `wc -l $1`
#
# loop on all lines trying to identify header lines
# an header line is defined as a non data line
# a data line is defined as one containing only numbers in the form 1 1.1 +1.1 -1.1 1.1e2 1.2e-2
etc.
# cannot do tail | head | grep otherwise sometimes grep will inherit the wrong $status code
  set i = 1
  startloop:
       set temp = `tail +$i $1 | head -1`
       echo $temp | grep -s -q '[+\-]*[0-9]\.*[0-9]*[eE]*[+-]*[0-9]*'
#      echo $temp | grep -s '[+\-]*[0-9]\.[0-9][eE]*[+-]*[0-9]*'      ????
       if ( $status == 0 ) then
#         make sure line does not contain any other alphanumeric character
          echo $temp | grep -s -q '[a-df-zA-DF-Z]'
          if( $status != 0 ) goto endloop
       endif
       @ i = $i + 1
       goto startloop
  endloop:
  @ nhead = $i - 1
  @ ndata = $nlines[1] - $nhead
#
# in first data line try to identify how many (blank separated) columns there are
  set line = `tail +$i $1 | head -1`
  set ncol = $#line
#
  echo XAS1ASC2GEN31234 $nhead $ndata $ncol > $$.tmp
  cat $1 >> $$.tmp
  cp -f $$.tmp $1
  rm -f $$.tmp
```

---

and this is the IDL procedure

```
pro xasasc,file,structure,colnames
;
;
; open file and check it is XAS ASCII
; this reads magic number AND entire content of first line
;
  openr,1,file
```

```
  magic=' '
  readf,1,magic
  magic=string(magic,format='(A16)')
  if (magic ne 'XAS1ASC2GEN31234') then return
;
; reposition to 17-th character in first line and read numbers
;
  point_lun,1,16
  readf,1,nhead,nrec,ncol
;
; skip header records
;
  hdr=' '
  for i=1,nhead do readf,1,hdr
;
; read entire set of data
;
  a=fltarr(ncol,nrec)
  readf,1,a
  a=transpose(a)
  close,1
;
; create the structure to be returned
; if no array of names passed
;
  s='structure = { '
  if (n_params() gt 2) then begin
     for i=1,ncol do begin
     b=string(format='(A,": fltarr(",I6.6,"), ")',colnames(i-1),nrec)
     s=s+b
     endfor
  endif else begin
     for i=1,ncol do begin
     b=string(format='("col",I2.2,": fltarr(",I6.6,"), ")',i,nrec)
     s=s+b
     endfor
  endelse
  strput,s,'}',strlen(s)-2
  test=execute(s)
;
; fill the structure
; executing assignment like  structure.col01=a(*,0) etc.
;
  for i=1,ncol do begin
  b=string(format='("structure.(",I2,")=a(*,",I2,")")',i-1,i-1)
  test=execute(b)
  endfor
return
```

end

---------- ----------