Subject: Problem with reading IDL written binary file....corrupt values?
Posted by Dave[2] on Fri, 20 Jan 2006 14:04:35 GMT
View Forum Message <> Reply to Message

Hi Group...I'm having a problem reading some binary data I write to a file in IDL. Everything looks great until I try to read data from the file after I close it. What I am doing is as follows:

I'm spawning to my UNIX command shell to run a C program that creates a 5 column text file that looks like this:

-121.375491	39.049994	14.722300	836	201826173
-121.375254	39.049994	15.247306	595	201831399
-121.369433	39.049994	15.551452	836	201831589
-121.374392	39.049994	17.324328	668	201832516
-121.369364	39.049994	14.978454	676	201831695
-121.378077	39.049994	10.458236	484	201827779
-121.368601	39.049994	14.737473	841	201832916

It's approximately 1 million records in length (X,Y,Z, intensity, GMT packed time) and the program is called numerous times to create a series of txt files. each is done in pieces given the data is eventually hundreds of millions of records.

> From there I generate a binary file in IDL using the code below from the text file above.

```
; Initialize the type for some variables:
    lat_e = 0.0D
    lon_e = 0.0D
    zval_e = 0.0D
    int_e = 0L
    time_e = 0L
    lon = lonarr(wtch)
    lat = lonarr(wtch)
    zval = 0L
    wtch=1

    ; Define single record header:
    numwords  = 8
    numwords0 = 5
```

```
    numwords1 = 4
    header    = lonarr(numwords)
    header[0] = numwords*4L ;set record length

    for i = 1, numwords-1 do begin
    header[i] = 0    ;default other header words to zero
    endfor

    final_rq_file = '  '

    for j = 0, wtch-1 do begin
       ; Set text file to be written to rq format file:
        openr, j+1, ofilname_NAVD88_txt(j)
    endfor;

   ; Open final binary file to be written:
   ftype = '_elv_'
   fname_ext = '_navd88.rq'
   final_rq_file = outpath + fname_noext + ftype + area_code +
fname_ext
   openw, 3, final_rq_file
   writeu, 3, header

   WHILE NOT eof(1) DO BEGIN
       readf, 1, lon_e, lat_e, zval_e, int_e, time_e,
format='(f15.6,f15.6,f12.3,i10,i20)'
       elv_recs_rq = elv_recs_rq+1
         lon_e = ((lon_e)*1e6)
         lon[0] = long(lon_e)+360000000L
         lat_e = (lat_e)*1e6
         lat[0] = long(lat_e)
         zval_e = zval_e*1e3
         zval = long(zval_e)
       writeu, 3, lat[0], lon[0], zval, int_e[0], lat[0], lon[0], 0,
time_e
       ;print, 'lat[0], lon[0], zval, int_e[0], lat[0], lon[0], 0,
time_e'
   ENDWHILE

   free_lun, 1
   close, 1
   free_lun, 2
   close, 2
   free_lun, 3
   close, 3

;-------------------------------------
;_____________________________________________________________
```

;_____

The data being written to the binary file looks great if I look at it
row by row when it's being written but when I try to access the data
using the code below it comes out strange.
Data results using the code below:

First two rows of data:
32 0 0 0 0
0 0 0
39049994 238624509 43618 836 39049994
238624509 0 3079

;Everything looks great in this first two rows except for the last
column. The first row is a header, the second is the data?
;This second rows last column should be the same as the last column in
the text file above. column 5, and 6 are a clone of 1 and 2.

Number of Records in file : 852510

opening integer file

outfile data type:
DATA LONG = Array[8, 852510]

This call to get_file_chunk.pro is complete

;When I print the data after reading it in it gets worse:
print, data(*,1:2)
-1619197357 -620097991 535429120 -1402077184 38994515 -620097991
535429120 0
201831399 39049994 238630567 44419 836 39049994
238630567 0

Anyone have a suggestion on what might be happening?
;_____ _____
;_____ _____
;----------code to read in the IDL Written binary file

pro extract_rq, file = fname
;this procedure extracts level2 data ;David Finnegan
;keywords:
;file = input path and file name
;Procedures called get_file_chunk.pro

IF NOT Keyword_set(file) then begin

```
fname=dialog_PICKFILE(GROUP=b,/READ,PATH=path,FILTER=["*"],TITLE='select
data file',/MUST_EXIST)
ENDIF

fname1 = file_basename(fname)
ipath=file_dirname(fname, /mark_directory)
ipath_fname= ipath+fname1

offset_l2 = 0LL
while offset_l2 ne -1LL do begin
    offset_l2 = get_file_chunk(data,'integer', offset_l2,
file=ipath_fname, maxrecs=10000000L)
endwhile

return
END ; extract_rq
;*********************************************************
;***********
; get_file_chunk.pro
; IDL routine to read data from Airborne Oceanographic Lidar
;
; modified to read data from file in chunks
;
;usage:  a=get_file(data,type,offset,FILE=filename,MAXRECS=maxrecs)
;   a= offset value (in bytes) of data read. Set to -1 on end of file.
;   data= array into which file data is read. the type and size are set
by the get_file.
;   type= 'float' or 'integer'
;   offset= defines number of bytes into file to begin getting data
(excludes header bytes)
;     - set equal zero on first call to function.
;   FILE= optional keyword specifying file to be read.  If absent then
the
;     PICKFILE routine is called to interactively specify the file to
be read.
;   SUBSAMP= optional keyword specifying the amount of subsampling.
every Nth point is read.
;   MAXRECS= defines number of records to return per get (if not
specified in parameter list
;     - default = 1e4
; Note: header data in the data file is stripped automatically.
;
;*************************************************
FUNCTION
 get_file_chunk,data,datatyp,offset,FILE=filnam,SUBSAMP=subsa mp,MAXRECS=maxrecs

if NOT Keyword_set(filnam) then begin
```

```
  filnam=pickfile(GROUP=b,/READ,PATH=path,FILTER=["*"],TITLE='select
data file',/MUST_EXIST)
  if filnam EQ '' then Return, 0
  endif

if keyword_set(maxrecs) then numrecsperchunk = maxrecs ELSE
numrecsperchunk = 10000L

;%%%%%%%%% read the selected data file %%%%%%%%%%%%%%%%%%
;; if first pass open file else reset pointer to start of file

;open the file
openr,lun,filnam,/get_lun

 ;initialize a variable for number of records in each line
 ;and create a data array call DATA
 reclen=0L
 readu,lun,reclen
 point_lun,lun,0
 data=lonarr(reclen/4,2 ,/NOZERO)
 readu,lun,data

print, ''
print,'First two rows of data:  '
print, data(*,0)
print, data(*,1)
print, ''

;Check the AOL Header to determine what the nskip should be
     IF (data(0,1) EQ -9000008L) THEN nskip=data(1,1) ELSE nskip=reclen

;Set filinfo with the fstat structure of the file opened
;and determine the number of records in the file for numrecs
 filinfo=Fstat(lun)
 numrecs= (filinfo.size-nskip)/reclen
 reclen=reclen/4
 nskip = nskip + offset

;note: reclen is the number of words in each record
if (numrecs - (offset/(reclen*4))) LT numrecsperchunk then begin
    numrecsperchunk    = numrecs - (offset/(reclen * 4))
    offset = -1
endif else begin
    offset = offset + (numrecsperchunk * (reclen * 4))
endelse


IF (DATATYP EQ 'integer') THEN  begin
```

```
    a=assoc(lun,lonarr(reclen,numrecsperchunk),nskip) & print,'opening
integer file'; & Info,"opening INTEGER data file"
ENDIF ELSE begin
    a=assoc(lun,fltarr(reclen,numrecsperchunk),nskip) & print,'opening
float file'; & Info,"opening FLOAT data file"
ENDELSE
data=a(0)

free_lun,lun

print, 'This call to get_file_chunk.pro is complete'

Return, offset

end ; get_file
```