Subject: reading dem
Posted by Sylvain Carette on Wed, 30 Aug 2000 06:07:03 GMT
View Forum Message <> Reply to Message

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<tt>I'm trying to get started with IDL .</tt>
<br><tt>All their examples work fine. But of course, working with my data
didnt work at all...</tt>
<br><tt>Need to read usgs1:250k dem. I'm not sure if I understand the format
but I succeed to</tt>
<br><tt>get the right value using '' reads, string(filechunk), var ''. Maybe
there is other ways</tt>
<br><tt>with assoc, format, xdr stuff?</tt>
<br><tt>Anyhow, the problem here is that although if I execute each line
of a small procedure</tt>
<br><tt>separately from the console and it work as expected, it doesnt
when i execute the</tt>
<br><tt>procedure. It doesnt even finish the first loop.</tt>
<br><tt>% READS: End of input data encountered: <STRING  
('   366   366   359  ...')></tt>
<br><tt>% Execution halted at:   READDEM250     
   
25 C:\RSI\IDL53\project\readdem250.pro</tt>
<br><tt>Why can I execute the first loop cycle from the console and not
from inside the loop?</tt>
<br><tt>What did I miss?</tt>
<br><tt>Below listing -  btw, the reason the statements are all written
inline instead of</tt>
<br><tt>inside another loop is that I coulnt get nested loop to work thanks
to wonderful IDL documentation.</tt><tt></tt>
<p><tt>Thanks</tt><tt></tt>
<p><tt>Sylvain Carette</tt>
<br><tt>VRML designer-composer</tt><tt></tt>
<p><tt>pro readdem250</tt>
<br><tt>openr, lun, dialog_pickfile(), /get_lun</tt>
<br><tt>block = assoc( lun, bytarr(1024)) ;the file is divided in 1024
chunk</tt>
<br><tt>im = intarr(1201,1201) ; dem is 1201x1201</tt>
<br><tt>col = intarr(1201) ; one column</tt>
<br><tt>ba = intarr(146) ; to hold one profile stored trough 8*1024 recordB
chunk</tt>
<br><tt>bb = intarr(170) ; first = 146 bytes, last = 35 bytes,</tt>
<br><tt>bc = intarr(170) ; others = 170 bytes, total = 1201 bytes</tt>
<br><tt>bd = intarr(170)</tt>
<br><tt>be = intarr(170)</tt>
<br><tt>bf = intarr(170)</tt>
<br><tt>bg = intarr(170)</tt>

<br><tt>bh = intarr(35)</tt><tt></tt>
<p><tt>for i=1, 1201 do begin ; skip first 1024 chunk, "recordA"</tt>
<br><tt> reads, string(block[146:*,i]), ba ; recordB profile value
begin at byte 146</tt>
<br><tt> reads, string(block[i+1]), bb</tt>
<br><tt> reads, string(block[i+2]), bc</tt>
<br><tt> reads, string(block[i+3]), bd</tt>
<br><tt> reads, string(block[i+4]), be</tt>
<br><tt> reads, string(block[i+5]), bf</tt>
<br><tt> reads, string(block[i+6]), bg</tt>
<br><tt> reads, string(block[0:210,i+7]), bh  ; 6 byte/value</tt>
<br><tt> </tt>
<br><tt> col = [ba, bb, bc, bd, be, bf, bg, bh] ; concatenate all
segment in one profile</tt>
<br><tt> im[i,*] = col</tt>
<br><tt>endfor</tt>
<br><tt>imlr = bytscl(im)</tt>
<br><tt>end</tt></html>

---

Subject: Re: reading dem
Posted by Sylvain Carette on Fri, 01 Sep 2000 10:28:02 GMT
View Forum Message <> Reply to Message

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<tt>Well it didnt work.. :-(</tt>
<br><tt>I got:</tt>
<br><tt>% Input line is too long for input buffer of 32767 characters.</tt>
<br><tt>% READF: Error encountered reading from file. Unit: 100, File:
D:\gis\DEM250\montrealw.dem</tt><tt></tt>
<p><tt>Now I'm stuck... is there a way to make readf read only no more
than 32767 char?</tt>
<br><tt>I though to split the file in little chunks but if I do that I
cant use anymore readf since it need a lun if I understand correctly. This
bring me back to my first post in which I divide one record b into 8 1024
chunks and use "reads, string(chunk), var".</tt>
<br><tt>On the command line, each line ok. In the loop:</tt>
<br><tt>% READS: End of input data encountered: <STRING  
('   366   366   359  ...')></tt>
<br><tt>Either too many or not enough.... :-(</tt><tt></tt>
<p><tt>I noticed that you used long instead of int, what is the reason
for this?</tt>
<br><tt>Wait a minute: do I have more chance if I use assoc subdividing
the file in 1024 byte arrays before calling readf? Do assoc effectively
will make readf read only 1024 byte chunk at a time without complain?(I'll
try anyway but comment welcome)</tt>
<br><tt>And about the file pointer, once you place it what happen after?

I mean, do you have any control over it or it just drive crazy by itself
up to EOF?</tt>
<br><tt>I begin to have a fuzzy feel that I begin to understand.. or its
because its already 6am and I need to sleep</tt><tt></tt>
<p><tt>Sylvain Carette</tt>
<br><tt>VRML designer-composer</tt><tt></tt>
<p><tt>Craig Markwardt wrote:</tt>
<blockquote TYPE=CITE><tt>Sylvain,</tt><tt></tt>
<p><tt>Part of the problem is that very few people have any direct experience</tt>
<br><tt>with what you are trying.  Some, including me, offered some
general</tt>
<br><tt>advice, hoping that would help, but reading byte-level DEM's is
not</tt>
<br><tt>exactly common knowledge.  Also realize that a topic so focussed
and</tt>
<br><tt>specific as yours is not likely to pique anybody's interst. 
In the</tt>
<br><tt>future you would do better to describe what you need to do at a
higher</tt>
<br><tt>level, and help us by providing documentation, like I do now.</tt><tt></tt>
<p><tt>The documentation is here:</tt>
<br><tt><a
href="http://rockyweb.cr.usgs.gov/nmpstds/demstds.html">http://rockyweb.cr.usgs.gov/nmpstds/d
emstds.html</a></tt><tt></tt>
<p><tt>The DEM's are digital elevation maps, and are stored in 1024 byte</tt>
<br><tt>blocks.  The first block contains a type "A" record. 
The next blocks</tt>
<br><tt>contain a type "B" record which have the actual elevation data. 
The</tt>
<br><tt>actual data start at offset 144 of the record (where offset 0 is
the</tt>
<br><tt>beginning); there are 1201x1201 elements stored, 146 in the first</tt>
<br><tt>1024-block, and then 170 in the successive 1024-blocks.  Elevation</tt>
<br><tt>values are stored as ASCII format=(I6).</tt><tt></tt>
<p><tt>The key thing to realize that all the data is in ASCII, separated
by</tt>
<br><tt>blanks.  Therefore, while we could read each 1024-block in
turn, it's</tt>
<br><tt>better to just do a formatted READF.  There is no need to
work at the</tt>
<br><tt>byte level, except at the beginning to get to the right file offset.</tt>
<br><tt>Unfortunately IDL can only read 32k elements at a time, so I read
a</tt>
<br><tt>row at a time as a compromise.</tt><tt></tt>
<p><tt>pro readdem250, file, im</tt>
<br><tt>  m = 1201L & n = 1201L ;; Really get this from Type A,
element 16</tt>
<br><tt>  im = lonarr(m, n)     ;; Formally defined

as INTEGER*4</tt><tt></tt>
<p><tt>  openr, lun, file, /get_lun</tt>
<br><tt>  ;; Skip past type A record and 144 bytes of Type B record</tt>
<br><tt>  point_lun, lun, 1024L + 144L</tt><tt></tt>
<p><tt>  ;; Read data, one row at a time</tt>
<br><tt>  row = im(*,0)</tt>
<br><tt>  for i = 0, 1200 do begin</tt>
<br><tt>    readf, lun, row, format='(1201(I6))'</tt>
<br><tt>    im(*,i) = row</tt>
<br><tt>  endfor</tt>
<br><tt>  free_lun, lun</tt>
<br><tt>end</tt><tt></tt>
<p><tt>Of course I haven't tested this, but you can use this as a starting</tt>
<br><tt>point.</tt><tt></tt>
<p><tt>Craig</tt><tt></tt>
<p><tt>P.S.  If you are still interested in the block structure, then
you</tt>
<br><tt>would read the blocks like this:</tt><tt></tt>
<p><tt>bb = bytarr(1024)*nblocks</tt>
<br><tt>readu, lun, bb</tt><tt></tt>
<p><tt>This avoids some extraneous STRING calls.</tt><tt></tt>
<p><tt>--</tt>
<br><tt> ---------- ---------- </tt>
<br><tt>Craig B. Markwardt,  Ph.D.     & nbsp;  
EMAIL:    craigmnet@cow.physics.wisc.edu</tt>
<br><tt>Astrophysics, IDL, Finance, Derivatives | Remove "net" for better
response</tt>
<br><tt> ---------- ---------- </tt></blockquote>
<tt></tt></html>

---

## Subject: Re: reading dem
Posted by Craig Markwardt on Fri, 01 Sep 2000 18:34:34 GMT
View Forum Message <> Reply to Message

Sylvain Carette <sylvainc@total.net> writes:

> ----------A4C74D0ACADA4F254E47B596
> Content-Type: text/plain; charset=us-ascii
> Content-Transfer-Encoding: 7bit
>
> Well it didnt work.. :-(
> I got:
> % Input line is too long for input buffer of 32767 characters.
> % READF: Error encountered reading from file. Unit: 100, File:
> D:\gis\DEM250\montrealw.dem
>
> Now I'm stuck... is there a way to make readf read only no more than 32767

> char?

Okay, so you can read <31 1024-blocks at a time, unformatted. Then you convert to a string and use reads.

```
ii = 0L
bb = bytarr(1024L*31) && ll = lonarr(170L*31)
while ii LT nvals do begin
  readu, lun, bb          ;; Read blocks
  reads, string(bb), ll   ;; Change to a string and parse into LONGs
  im(ii:ii+170-1) = ll    ;; Insert into image array
  ii = ii + n_elements(ll) ;; Advance array index
end
```

This doesn't handle the case of the first block, which only has 135 elements, but you should get the idea. Now why not try a little experimentation on your own...

Good luck,
Craig

P.S. I used a long integer because the values are defined in the spec as INTEGER*4.

--


---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------