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>

<tt>I got:</tt>

<tt>% Input line is too long for input buffer of 32767 characters.</tt>

<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>

<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>

<tt>On the command line, each line ok. In the loop:</tt>

<tt>% READS: End of input data encountered: <STRING
('   366   366   359 ...')></tt>

<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>

<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>

<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>

<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>

<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>

<tt>with what you are trying. Some, including me, offered some
general</tt>

<tt>advice, hoping that would help, but reading byte-level DEM's is
not</tt>

<tt>exactly common knowledge. Also realize that a topic so focussed
and</tt>

<tt>specific as yours is not likely to pique anybody's interst.
In the</tt>

<tt>future you would do better to describe what you need to do at a
higher</tt>

<tt>level, and help us by providing documentation, like I do now.</tt><tt></tt>

<p><tt>The documentation is here:</tt>

<tt><a
href="http://rockyweb.cr.usgs.gov/nmpstds/demstds.html">http://rockyweb.cr.usgs.gov/nmpstds/d
emstds.html</tt><tt></tt>
<p><tt>The DEM's are digital elevation maps, and are stored in 1024 byte</tt>

<tt>blocks. The first block contains a type "A" record.
The next blocks</tt>

<tt>contain a type "B" record which have the actual elevation data.
The</tt>

<tt>actual data start at offset 144 of the record (where offset 0 is
the</tt>

<tt>beginning); there are 1201x1201 elements stored, 146 in the first</tt>

<tt>1024-block, and then 170 in the successive 1024-blocks. Elevation</tt>

<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>

<tt>blanks. Therefore, while we could read each 1024-block in
turn, it's</tt>

<tt>better to just do a formatted READF. There is no need to
work at the</tt>

<tt>byte level, except at the beginning to get to the right file offset.</tt>

<tt>Unfortunately IDL can only read 32k elements at a time, so I read
a</tt>

<tt>row at a time as a compromise.</tt><tt></tt>
<p><tt>pro readdem250, file, im</tt>

<tt> m = 1201L & n = 1201L ;; Really get this from Type A,
element 16</tt>

<tt> im = lonarr(m, n)   ;; Formally defined
as INTEGER*4</tt><tt></tt>
<p><tt> openr, lun, file, /get_lun</tt>

<tt> ;; Skip past type A record and 144 bytes of Type B record</tt>

<tt> point_lun, lun, 1024L + 144L</tt><tt></tt>
<p><tt> ;; Read data, one row at a time</tt>

<tt> row = im(*,0)</tt>

<tt> for i = 0, 1200 do begin</tt>

<tt>   readf, lun, row, format='(1201(I6))'</tt>

<tt>   im(*,i) = row</tt>

<tt> endfor</tt>

<tt> free_lun, lun</tt>

<tt>end</tt><tt></tt>
<p><tt>Of course I haven't tested this, but you can use this as a starting</tt>

<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>

<tt>would read the blocks like this:</tt><tt></tt>
<p><tt>bb = bytarr(1024)*nblocks</tt>

<tt>readu, lun, bb</tt><tt></tt>

<p><tt>This avoids some extraneous STRING calls.</tt><tt></tt>
<p><tt>--</tt>

<tt> ---------- ---------- </tt>

<tt>Craig B. Markwardt, Ph.D.   & nbsp;
EMAIL: craigmnet@cow.physics.wisc.edu</tt>

<tt>Astrophysics, IDL, Finance, Derivatives | Remove "net" for better
response</tt>

<tt> ---------- ---------- </tt></blockquote>
<tt></tt></html>