Subject: Re: reading dem
Posted by Sylvain Carette on Thu, 31 Aug 2000 02:08:49 GMT
View Forum Message <> Reply to Message

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<tt>Aaaaahh... Thank you so much</tt>

<tt>Whithout knowing it you answered a couple of unnasked questions
(I think I have one for each page of the documentation)</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> </blockquote>
<tt>Sorry if my requests sound confused; they are... I'm trying not to
trow all the questions floating in my head at the same time but probably
it will get better as time goes. Sorry also if a bit of frustration did
show up through the lines; I was getting anxious to get something done.
Now already feel better...</tt>
<blockquote TYPE=CITE><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></blockquote>
<tt>I did realize that the data was in ascii but my problem was to make
idl realize that...</tt>

<tt>I tryed readf but could get only the value of the ascii code instead
of the value itself; of course I tryed to use "format=" but couldnt guess
the right syntax - I think I was trying somthing like "format=(6(i))" so...
- again, the documentation could be lot more helpful by providing a couple
of (meaningful) examples.</tt>

<tt>So I end up thinking it was not the right function especially that
all the dem doc say it is a binary file and I though that formatted input
mean a real ascii file (one you can read). So I can do lot with readf as
long I know how to express the "format" syntax. Now, where I can find a
good table & rule describing formatting (I have absolutely no background
in fortran - completely alien to me. I have the impression I'll need to
refer to that all the time.</tt>
<blockquote TYPE=CITE><tt></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></blockquote>
<tt>Sure I will! I knew already that what I was not on the right track.
Now I will be able to "start" the right way so its a big improvement. Just
this small snippet tell me lot. It show me how to do things right. Thanks
also for the gtopo answer; now I have two goods starting point.</tt>
<blockquote TYPE=CITE><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>Sylvain Carette</tt>

<tt>VRML designer-composer</tt></html>