Subject: Re: Solve memory problems
Posted by pgrigis on Wed, 14 Jan 2009 15:38:50 GMT
View Forum Message <> Reply to Message

But probably this is important only on windows 32-bit systems?

Ciao,
Paolo

Jean H. wrote:
> Corinne,
>
> setting var=0 or *ptr_var = 0 will have the same effect on memory.
> Now, let's say you have 10 bands, 4000*5000. If you try to create an
> array like this data = bytarr(4000,5000,10), you might run out of
> memory, or, for the same reason, can not create any other variable (not
> enough contiguous space in memory). With this example, you would need
> about 8 bits * 4000 * 5000 * 10 = 1 600 000 000 bits of contiguous
> memory. Now, if you use points, you can create an array of 10 pointers,
> each holding a band.
> ptr_data = ptrarr(10)
> ptr_data[0] = ptr_new(bytarr(4000,5000)
> ptr_data[1] = ptr_new(bytarr(4000,5000)
> ....
>
> so now, the contiguous memory you need is only 8 bits * 4000 * 5000 =
> 160 000 000 bits.
>
> If band 2 has a different size, no problem:
> ptr_data[2] = ptr_new(bytarr(12,25)
>
> Jean
>
> Corinne wrote:
>> hi jean,
>>
>> i have never used pointers, so my question is: do you use separate
>> pointers or whole pointer-arrays for your bands? what do you do, if
>> your bands have different sizes due to different spatial resolutions
>> (which you might interpolate later in the programme)?
>>
>> i'm still trying to figure out, what the advantage of pointers is.
>>
>> example: i have created an float array a with 4000x5000 elements and
>> don't need it anymore. so i want to get rid of it. does it make a
>> difference, if i put a=0 or if i set the value of the pointer of the
>> array to zero?
>>

>> regards,
>> corinne
>>
>>
>>> In my own program, I do all the analysis on a modified version of my
>>> original image (a classified land-use map, with the background values
>>> removed so the data is a 1D array), then, at the very end, I
>>> re-transform it to save and display it. I save a lot of memory!
>>> Moreover, all bands are saved in pointers, allowing the program to run
>>> on almost any computer, while the original version, which did not use
>>> much pointer, was making my work-beast run out of memory fairly quickly!
>>>
>>> Jean
>>
