Subject: Re: Using ASSOC
Posted by Peter Mason on Wed, 26 May 2004 00:50:36 GMT
View Forum Message <> Reply to Message

Jonathan Greenberg wrote:
> I was hoping to get some help/feedback on the use of ASSOC. I'm
> working with a large image file and, while I can extract subsections
> of the image (say 3 consequtive lines), I'd prefer to work with the
> image as an array, but not have it loaded into memory. Is this what
> ASSOC allows me to do? I'm having a problem with the following
> situation:
>
> Nl and ns = the dimensions of the byte image "someimage.dat".
> Openr,unit,'someimage.dat',/get_lun
> Testassoc=assoc(unit,bytarr(nl,ns))
>
> I'd like to be able to extract the value from position x,y from
> testassoc -- how do I do this? Testassoc[0] seems to be the entire
> image. But testassoc[0,1] gives me an end of file error, as does
> testassoc[0,1,1] (which I would think would give me the value at that
> position).
>
> Using the /packed keyword doesn't seem to do anything. Help!

What you have done is to set up ASSOC with a chunk size that's the same size as the whole image. (You don't want to do that as it's the same thing as reading the entire image into memory.) As such, your ASSOC variable can only give you one chunk, chunk number 0.

While ASSOC appears to implement memory mapping, it should really be thought of as "random access made simple". Here's the gist of it:

```
Testassoc=assoc(unit,bytarr(ns,nlchunk))
```

This means: "I want to access this file in chunks, where the size of one chunk is BYTARR(ns,nlchunk)."
(BTW, from your BYTARR(nl,ns), it would appear that your images is stored rotated?)

```
tile=testassoc[17]
```

This means: "Give me the eighteenth chunk." (And TILE ends up being a BYTARR(ns,nlchunk)'s worth of data.)

It's handy because you can whiz around your image, accessing chunks backwards and forwards as you please.

These days you can actually do *proper* memory mapping in IDL, using SHMMAP and SHMVAR (and ultimately SHMUNMAP). Using these functions, you can get your entire image mapped to memory... maybe. If it works, it'll look and function the same as an IDL array into which you have read the image, only

it won't necessarily consume as much of your computer's RAM + pagefile as a regular IDL array. I said "maybe" earlier because it might not work if your image is too big. Memory mapping consumes address space in the same way that regular memory allocation does, and on a 32-bit platform (certainly on a win32 platform) you might have even less address space than RAM + pagefile.

So, if your image isn't too large you might want to give proper memory-mapping a go, otherwise experiment with ASSOC using a smallish tile size. For instance, say you have a 1-band image that's NS samples wide and NL lines long, and it's stored "across then down". The simplest way to get pixels in it would be to set up a line-sized chunk:

```
linebuf=ASSOC( unit, BYTARR(ns) )
```

Then to get the pixel at, say sample X and line Y (counting from 0), you could do:

```
pixel_xy=(linebuf[y])[x]
```

...The simplest way but probably not the most efficient. To improve efficiency, try using a larger chunk size (more than 1 line) and caching it, updating whenever Y is "out of range".