Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by David Fanning on Thu, 03 Dec 2009 17:09:13 GMT
View Forum Message <> Reply to Message

nata wites:

>
> My application needs a lot of RAM so I have to improve a solution in
> order to to use less resources.
> Normally, when I've an object I use declarations like this:
>
> pro myobject__define
> struct = { myobject, $
> data: ptr_new() }
> end
>
> Now for example if my object has to save an array like FLTARR
> (400,400,24,97) I will use the data pointer to store this array. The
> problem is that if I do this I take a lot of computer resources. With
> this example:
> help, /mem
> aa=fltarr(400,400,24,97)
> myobject->SetProperty, DATA=aa
> aa=0l
> help, /mem
>
> IDl returns:
> heap memory used: 658906, max: 805215874, gets: 1195,
> frees: 387
> heap memory used: 1490578946, max: 1490579037, gets: 1207,
> frees: 398
>
> So, only for this example I'm using 1.4 Gb aprox. I tried to used
> ASSOC procedure but I didn't succeed.... Some suggestions or comments
> about how to reduce the memory ? There is a method to store compressed
> data or something similar ?

I think I would learn how to put data into a pointer without
making a copy of it:

http://www.dfanning.com/misc_tips/pointers.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by ben.bighair on Thu, 03 Dec 2009 17:09:24 GMT
View Forum Message <> Reply to Message

On Dec 3, 10:44 am, nata <bernat.puigdomen...@gmail.com> wrote:
> Hi gurus,
>
> My application needs a lot of RAM so I have to improve a solution in
> order to to use less resources.
> Normally, when I've an object I use declarations like this:
>
> pro myobject__define
> struct = { myobject, $
> data: ptr_new() }
> end
>
> Now for example if my object has to save an array like FLTARR
> (400,400,24,97) I will use the data pointer to store this array. The
> problem is that if I do this I take a lot of computer resources. With
> this example:
> help, /mem
> aa=fltarr(400,400,24,97)
> myobject->SetProperty, DATA=aa
> aa=0l
> help, /mem
>
> IDl returns:
> heap memory used: 658906, max: 805215874, gets: 1195,
> frees: 387
> heap memory used: 1490578946, max: 1490579037, gets: 1207,
> frees: 398
>
> So, only for this example I'm using 1.4 Gb aprox. I tried to used
> ASSOC procedure but I didn't succeed.... Some suggestions or comments
> about how to reduce the memory ? There is a method to store compressed
> data or something similar ?

Hi,

You don't show how you assign your array to a pointer reference, but

you might see a some gain in using the /NO_COPY keyword for PTR_NEW()

Another thought is to allow your SetProperty method to accept a pointer to start with so that you reduce the transfer costs passing the data into the method call. Maybe something like the following will work as a switch hitter for you. Keep in mind I have written an pointer handler in a long time*, so I might have some of the logic askew.

```
PRO MyObject::SetProperty, data = data, ...

if N_ELEMENTS(data) GT 0 then begin
  if (SIZE(data, /TYPE)) EQ 10) then begin
    if PTR_VALID(self.data) then PTR_FREE, self.data
    self.data = data
  endif else begin
    if PTR_VALID(self.data) then $
      *self.data = data else $
      self.data = PTR_NEW(data, /NO_COPY)
  endif
endif

END; SetProperty
```

Then you could do

```
aa = PTR_NEW(fltarr(400,400,24,97), /NO_COPY)
myobject->SetProperty, DATA=aa
```

Some folks might squeak (rightly so) that this approach might allows you sneaky and direct access to an object property - thus exposing the innards when you shouldn't and breaking the spirit of encapsulation. But, why should we be confined by well-reasoned and established common practice?

I am quite sure that there are other places for you to gain more on memory management but that gets above my pay grade.

Cheers,
Ben

* In fact, my IDL is so rusty that I kept trying to type the program blocks using { }. Oops!

Subject: Re: The best way to keep data in RAM / object-oriented programming

Posted by David Fanning on Thu, 03 Dec 2009 17:40:33 GMT
View Forum Message <> Reply to Message

Ben wites:

> But, why should we be confined by well-reasoned and established common
> practice?

Exactly! :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by natha on Thu, 03 Dec 2009 18:31:21 GMT
View Forum Message <> Reply to Message

I'm sorry guys but I don't see the difference.
I understand what are you explaining and the functionality of the
NO_COPY KEYWORD but the result is the same...

If I've to store an array fltarr(400,400,24,97) in a pointer, the
result, in heap memory usage, is the same if I do:

a=fltarr(400,400,24,97)
b=ptr_new(a)
a=0l
help, /heap

or

a=fltarr(400,400,24,97)
b=ptr_new(a,/no_copy)
help, /heap

I´ll learn about COMMONs
Cheers,
nata

So, that's not what I'm looking for. I need to keep the arrays in
memory but using less memory resources. Is it possible?

Cheers,
nata

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by Kenneth P. Bowman on Thu, 03 Dec 2009 18:44:24 GMT
View Forum Message <> Reply to Message

In article
<f9ac2203-ccfc-4917-b300-4db3750b6e37@m38g2000yqd.googlegroups.com>,
 nata <bernat.puigdomenech@gmail.com> wrote:

> I'm sorry guys but I don't see the difference.
> I understand what are you explaining and the functionality of the
> NO_COPY KEYWORD but the result is the same...
>
> If I've to store an array fltarr(400,400,24,97) in a pointer, the
> result, in heap memory usage, is the same if I do:
>
> a=fltarr(400,400,24,97)
> b=ptr_new(a)
> a=0l
> help, /heap
>
> or
>
> a=fltarr(400,400,24,97)
> b=ptr_new(a,/no_copy)
> help, /heap
>

> Cheers,
> nata
>
>
> So, that's not what I'm looking for. I need to keep the arrays in
> memory but using less memory resources. Is it possible?

400 x 400 x 24 x 97 x (4 bytes) is approx. 1.5 GB.

The only way to use less memory is to use a smaller type
(e.g., INTs) and give up both precision and convenience.

Memory is very cheap nowadays.

Ken Bowman

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by penteado on Thu, 03 Dec 2009 19:27:41 GMT
View Forum Message <> Reply to Message

On Dec 3, 4:31 pm, nata <bernat.puigdomen...@gmail.com> wrote:
> I'm sorry guys but I don't see the difference.
> I understand what are you explaining and the functionality of the
> NO_COPY KEYWORD but the result is the same...
>
> If I've to store an array fltarr(400,400,24,97) in a pointer, the
> result, in heap memory usage, is the same if I do:
>
> a=fltarr(400,400,24,97)
> b=ptr_new(a)
> a=0l
> help, /heap
>
> or
>
> a=fltarr(400,400,24,97)
> b=ptr_new(a,/no_copy)
> help, /heap
>
> I´ll learn about COMMONs
> Cheers,
> nata
>
> So, that's not what I'm looking for. I need to keep the arrays in
> memory but using less memory resources. Is it possible?

The difference is that in the first way you had, for a while (between
b=ptr_new(a) and a=0l), two copies of the same large array in memory,
one in a, and another in *b. Yes, you free (nearly all) the memory
used by a with a=0l, but before that you wasted all the memory and
time to make a copy of a. Which is particularly relevant if the extra
memory use means having to use disk cache. So use no_copy instead.

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by natha on Thu, 03 Dec 2009 19:47:02 GMT
View Forum Message <> Reply to Message

Ok thank you to all of you !

nata

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by Juggernaut on Fri, 04 Dec 2009 19:00:09 GMT
View Forum Message <> Reply to Message

On Dec 3, 10:44 am, nata <bernat.puigdomen...@gmail.com> wrote:
> Hi gurus,
>
> My application needs a lot of RAM so I have to improve a solution in
> order to to use less resources.
> Normally, when I've an object I use declarations like this:
>
> pro myobject__define
> struct = { myobject, $
> data: ptr_new() }
> end
>
> Now for example if my object has to save an array like FLTARR
> (400,400,24,97) I will use the data pointer to store this array. The
> problem is that if I do this I take a lot of computer resources. With
> this example:
> help, /mem
> aa=fltarr(400,400,24,97)
> myobject->SetProperty, DATA=aa
> aa=0l
> help, /mem
>
> IDl returns:
> heap memory used: 658906, max: 805215874, gets: 1195,
> frees: 387
> heap memory used: 1490578946, max: 1490579037, gets: 1207,
> frees: 398
>
> So, only for this example I'm using 1.4 Gb aprox. I tried to used
> ASSOC procedure but I didn't succeed.... Some suggestions or comments
> about how to reduce the memory ? There is a method to store compressed
> data or something similar ?
>
> Thanks in advance,
> nata

When you say you tried to use ASSOC but didn't succeed does that mean
you used it and it didn't work out for you or you don't know how to
use it? I've used the ASSOC command to open massive arrays and store
them to disk. As long as you're willing to take a bit of a
performance hit it does the job.

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by natha on Fri, 04 Dec 2009 20:25:52 GMT
View Forum Message <> Reply to Message

First of all, I suppose that for use the ASSOC function I need to
store my arrays into a binary file. That's not very efficient...
After that, I tried to accede to the created files to get the desired
information and I didn't succeed. Maybe I don't know how to use this
function properly but I always have errors.
I don't know if I can use the ASSOC function in order to save
compressed data using the PACKED keyword and then have access to the
file efficiently. If it's possible and this is a good solution to keep
information, please tell me how to do that.

Subject: Re: The best way to keep data in RAM / object-oriented programming
Posted by Juggernaut on Mon, 07 Dec 2009 17:42:41 GMT
View Forum Message <> Reply to Message

On Dec 4, 3:25 pm, nata <bernat.puigdomen...@gmail.com> wrote:
> First of all, I suppose that for use the ASSOC function I need to
> store my arrays into a binary file. That's not very efficient...
> After that, I tried to accede to the created files to get the desired
> information and I didn't succeed. Maybe I don't know how to use this
> function properly but I always have errors.
> I don't know if I can use the ASSOC function in order to save
> compressed data using the PACKED keyword and then have access to the
> file efficiently. If it's possible and this is a good solution to keep
> information, please tell me how to do that.

```
; Create a blank file ready to receive data
data = FLTARR(256,256)
OPENW, lun, writename, /GET_LUN
FOR i = 0, 99 DO WRITEU, lun, data
FREE_LUN, lun
OPENU, datalun, writename, /GET_LUN
A = ASSOC(dataLun, data)

; Perform some genius processing of your choosing
FOR i = 0, 99 DO BEGIN
  newFrame = FLTARR(256,256) + i
  A[i] = newFrame
ENDFOR
```

I believe this is the general way I did things.