Subject: Re: memory allocation on Macs
Posted by Brian Larsen on Thu, 01 May 2008 17:54:10 GMT
View Forum Message <> Reply to Message

Paolo,

I get the same behavior but different error codes and memory usage
info.

IDL> a=bytarr(1000,1000,1000)
IDL> b=bytarr(1000,1000,1000)
IDL> c=bytarr(1000,1000,1000)
IDL> d=bytarr(1000,1000,1000)
idl(344,0xa0488fa0) malloc: *** mmap(size=1000001536) failed (error
code=12)
*** error: can't allocate region
*** set a breakpoint in malloc_error_break to debug
% Unable to allocate memory: to make array.
  Cannot allocate memory
% Execution halted at: $MAIN$

but my activity monitor only shows 131MB free

I am on IDL 6.4.1 and OSX 10.5.2.

Is this an activity monitor difference only?

Cheers,

Brian

---------- ----------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: memory allocation on Macs
Posted by Rick Towler on Thu, 01 May 2008 18:15:49 GMT
View Forum Message <> Reply to Message

pgrigis wrote:
> Hi folks,
>
> we have pretty much exhausted the topic of memory

> allocation on Windows and Linux, but I don't remember
> any discussion abut this on Mac OS.
>
> So, I am using IDL 6.3 on Mac OS X 10.4.11.
>
> I tried allocating as many 1GB array as possible,
> and it failed after 3 successful allocations.
> Now, the "Activity Monitor" indicates that at this point
> I have 3.6 GB of memory used and 3.4 GB free.
> So I am wondering why cant'I allocate a couple more
> of 1GB arrays?

I'm not a macatista, but a quick google search reveals that as of 10.3, the per process memory limit in OS X is 4GB. That squares with what you're seeing. Someone more in the know might be able to tell you if/how this can be tuned. For instance using "setrlimit".

-Rick

Subject: Re: memory allocation on Macs
Posted by pgrigis on Thu, 01 May 2008 18:25:51 GMT
View Forum Message <> Reply to Message

Brian Larsen wrote:
> Paolo,
>
> I get the same behavior but different error codes and memory usage
> info.
>
> IDL> a=bytarr(1000,1000,1000)
> IDL> b=bytarr(1000,1000,1000)
> IDL> c=bytarr(1000,1000,1000)
> IDL> d=bytarr(1000,1000,1000)
> idl(344,0xa0488fa0) malloc: *** mmap(size=1000001536) failed (error
> code=12)
> *** error: can't allocate region
> *** set a breakpoint in malloc_error_break to debug
> % Unable to allocate memory: to make array.
> Cannot allocate memory
> % Execution halted at: $MAIN$
>
> but my activity monitor only shows 131MB free
>
> I am on IDL 6.4.1 and OSX 10.5.2.
>
> Is this an activity monitor difference only?
Well, the machine as 7GB total real memory...

How much you have? If it is 4GB, that would
explain it I guess, although I don't see why it
wouldn't try to use virtual memory first.

Ciao,
Paolo

>
> Cheers,
>
> Brian
>
>
>
> ---------- ----------
> Brian Larsen
> Boston University
> Center for Space Physics
> http://people.bu.edu/balarsen/Home/IDL

Subject: Re: memory allocation on Macs
Posted by pgrigis on Thu, 01 May 2008 19:06:25 GMT
View Forum Message <> Reply to Message

Rick Towler wrote:
> pgrigis wrote:
>> Hi folks,
>>
>> we have pretty much exhausted the topic of memory
>> allocation on Windows and Linux, but I don't remember
>> any discussion abut this on Mac OS.
>>
>> So, I am using IDL 6.3 on Mac OS X 10.4.11.
>>
>> I tried allocating as many 1GB array as possible,
>> and it failed after 3 successful allocations.
>> Now, the "Activity Monitor" indicates that at this point
>> I have 3.6 GB of memory used and 3.4 GB free.
>> So I am wondering why cant'I allocate a couple more
>> of 1GB arrays?
>
> I'm not a macatista, but a quick google search reveals that as of 10.3,
> the per process memory limit in OS X is 4GB. That squares with what
> you're seeing. Someone more in the know might be able to tell you
> if/how this can be tuned. For instance using "setrlimit".
I'll investigate this.

Thanks,
Paolo

>
> -Rick

Subject: Re: memory allocation on Macs
Posted by Brian Larsen on Thu, 01 May 2008 20:08:40 GMT
View Forum Message <> Reply to Message

Rick,

you may be onto something here...

Paolo what do you get from typing "limit" at a terminal prompt?

I ran "ulimit" which unsets the limit but forgot to look at the limit
beforehand (oops)

[balarsen@comm725-0401-dhcp-226]~> limit
cputime unlimited
filesize unlimited
datasize 6144 kbytes
stacksize 8192 kbytes
coredumpsize 0 kbytes
memoryuse unlimited
descriptors 256
memorylocked unlimited
maxproc 266

However I had the same IDL result, error on array d.

Yes I have 4GB of RAM in my macbook pro.

Cheers,

Brian

---------- ----------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: memory allocation on Macs

Posted by pgrigis on Thu, 01 May 2008 20:56:22 GMT
View Forum Message <> Reply to Message

Brian Larsen wrote:
> Rick,
>
> you may be onto something here...
>
> Paolo what do you get from typing "limit" at a terminal prompt?
Limit shows the same output as yours, so that shouldn't be
a problem.

Paolo

>
> I ran "ulimit" which unsets the limit but forgot to look at the limit
> beforehand (oops)
>
> [balarsen@comm725-0401-dhcp-226]~> limit
> cputime unlimited
> filesize unlimited
> datasize 6144 kbytes
> stacksize 8192 kbytes
> coredumpsize 0 kbytes
> memoryuse unlimited
> descriptors 256
> memorylocked unlimited
> maxproc 266
>
> However I had the same IDL result, error on array d.
>
> Yes I have 4GB of RAM in my macbook pro.
>
> Cheers,
>
> Brian
>
> ---------- ----------
> Brian Larsen
> Boston University
> Center for Space Physics
> http://people.bu.edu/balarsen/Home/IDL

Subject: Re: memory allocation on Macs
Posted by pgrigis on Thu, 01 May 2008 21:00:26 GMT

Yes, I found this on one of apple's webpages:

Unlike earlier versions of Mac OS, Mac OS X includes a fully-integrated virtual memory system that you cannot turn off. It is always on, providing up to 4 gigabytes of addressable space per 32-bit process and approximately 18 exabytes of addressable space for 64-bit processes.

So if this is true, 32 bit processes cannot access more than 4GB of memory....

Ciao,
Paolo

Rick Towler wrote:
> pgrigis wrote:
>> Hi folks,
>>
>> we have pretty much exhausted the topic of memory
>> allocation on Windows and Linux, but I don't remember
>> any discussion abut this on Mac OS.
>>
>> So, I am using IDL 6.3 on Mac OS X 10.4.11.
>>
>> I tried allocating as many 1GB array as possible,
>> and it failed after 3 successful allocations.
>> Now, the "Activity Monitor" indicates that at this point
>> I have 3.6 GB of memory used and 3.4 GB free.
>> So I am wondering why cant'I allocate a couple more
>> of 1GB arrays?
>
> I'm not a macatista, but a quick google search reveals that as of 10.3,
> the per process memory limit in OS X is 4GB. That squares with what
> you're seeing. Someone more in the know might be able to tell you
> if/how this can be tuned. For instance using "setrlimit".
>
> -Rick

Subject: Re: memory allocation on Macs
Posted by Karl[1] on Fri, 02 May 2008 16:36:06 GMT

Yep, a process on a 32-bit OS can only address 4GB of memory. The long and complicated discussions about being able to allocate less

memory on Windows had to do with how Windows partitioned the 32-bit virtual address space and virtual address space fragmentation issues.

Is a machine with 7GB of RAM making use of the 7GB, even if the OS is 32-bit?

Yes, I think, for OS X with a G5. Note that on this machine with 7GB of RAM, you could probably start a second instance of IDL and allocate 3 more 1 GB arrays and use them WITHOUT paging.

Some OS's, I dunno about OS X, will cache file I/O in this "extra" memory, which greatly speeds up file reads if you read files over and over.

I did find an article (2003) that says the G5 can support more than 4GB RAM and probably uses it as I have noted above. Note that the story may be different for Intel processors. I know that the Xeon can address more than 4GB when running a Server version of Windows and that's why you see Windows servers built on the Xeon and tons of RAM. I don't know if any of this is true for any versions of the P4.

You can also start as many instances of IDL that you want and allocate more arrays, but then you'll be subject to a drop in performance due to paging and any upper limit placed on the paging file.

The performance hit depends on the memory Working Set of the applications that are involved. If these large processes are only touching a few pages of memory (unlikely), the performance will be very good since all the needed pages fit into RAM. But increase, the working set to the point where paging occurs, and the performance drops by 2 orders of magnitude, due to paging.

I think Wikipedia has some decent articles on virtual memory OS's.

In any case, if you need a single process size to exceed 4GB, use a 64-bit OS.

Karl

On May 1, 3:00 pm, pgri...@gmail.com wrote:
> Yes, I found this on one of apple's webpages:
>
> Unlike earlier versions of Mac OS, Mac OS X includes a fully-
> integrated virtual memory system that you cannot turn off. It is
> always on, providing up to 4 gigabytes of addressable space per 32-bit
> process and approximately 18 exabytes of addressable space for 64-bit
> processes.

>
> So if this is true, 32 bit processes cannot access more than 4GB of
> memory....
>
> Ciao,
> PaoloRick Towler wrote:
>> pgrigis wrote:
>>> Hi folks,
>
>>> we have pretty much exhausted the topic of memory
>>> allocation on Windows and Linux, but I don't remember
>>> any discussion abut this on Mac OS.
>
>>> So, I am using IDL 6.3 on Mac OS X 10.4.11.
>
>>> I tried allocating as many 1GB array as possible,
>>> and it failed after 3 successful allocations.
>>> Now, the "Activity Monitor" indicates that at this point
>>> I have 3.6 GB of memory used and 3.4 GB free.
>>> So I am wondering why cant'I allocate a couple more
>>> of 1GB arrays?
>
>> I'm not a macatista, but a quick google search reveals that as of 10.3,
>> the per process memory limit in OS X is 4GB. That squares with what
>> you're seeing. Someone more in the know might be able to tell you
>> if/how this can be tuned. For instance using "setrlimit".
>
>> -Rick

Subject: Re: memory allocation on Macs
Posted by Brian Larsen on Fri, 02 May 2008 17:56:07 GMT
View Forum Message <> Reply to Message

Rick,

On this point

> In any case, if you need a single process size to exceed 4GB, use a 64-
> bit OS.

at the risk of being a little ignorant here. I ran this test on
Leopard (10.5.2) which is reported to be a 64-bit OS...
http://www.apple.com/macosx/technology/64bit.html
Making this an issue of having 32-bit IDL not a 32-bit OS. Right?

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: memory allocation on Macs
Posted by pgrigis on Fri, 02 May 2008 19:13:04 GMT
View Forum Message <> Reply to Message

Yes, you're right that I can allocate all the 7 GB (and more) in different IDL
sessions. So there seems to be a limit indeed on how much memory one single
IDL session (or in general , one process) can use up, but there isn't a limit for
total usage (which, though I am sure there are a number of technical reason
for it, seems a bit silly, after all if the system as a whole can access more
than 4 GB, why shouldn't parts of the system be allowed to do the same?)

FYI, this is a Xeon machine in Mac OS X 10.4, so it is a 64 bit processor
in a 32 bit OS running a 32 bit application.

Anyway, thanks to all. I can cope with reading a few arrays off the disk
from time to time.

Ciao,
Paolo

Karl wrote:
> Yep, a process on a 32-bit OS can only address 4GB of memory. The
> long and complicated discussions about being able to allocate less
> memory on Windows had to do with how Windows partitioned the 32-bit
> virtual address space and virtual address space fragmentation issues.
>
> Is a machine with 7GB of RAM making use of the 7GB, even if the OS is
> 32-bit?
>
> Yes, I think, for OS X with a G5. Note that on this machine with 7GB

> of RAM, you could probably start a second instance of IDL and allocate
> 3 more 1 GB arrays and use them WITHOUT paging.
>
> Some OS's, I dunno about OS X, will cache file I/O in this "extra"
> memory, which greatly speeds up file reads if you read files over and
> over.
>
> I did find an article (2003) that says the G5 can support more than
> 4GB RAM and probably uses it as I have noted above. Note that the
> story may be different for Intel processors. I know that the Xeon can
> address more than 4GB when running a Server version of Windows and
> that's why you see Windows servers built on the Xeon and tons of RAM.
> I don't know if any of this is true for any versions of the P4.
>
> You can also start as many instances of IDL that you want and allocate
> more arrays, but then you'll be subject to a drop in performance due
> to paging and any upper limit placed on the paging file.
>
> The performance hit depends on the memory Working Set of the
> applications that are involved. If these large processes are only
> touching a few pages of memory (unlikely), the performance will be
> very good since all the needed pages fit into RAM. But increase, the
> working set to the point where paging occurs, and the performance
> drops by 2 orders of magnitude, due to paging.
>
> I think Wikipedia has some decent articles on virtual memory OS's.
>
> In any case, if you need a single process size to exceed 4GB, use a 64-
> bit OS.
>
> Karl
>
>
> On May 1, 3:00 pm, pgri...@gmail.com wrote:
>> Yes, I found this on one of apple's webpages:
>>
>> Unlike earlier versions of Mac OS, Mac OS X includes a fully-
>> integrated virtual memory system that you cannot turn off. It is
>> always on, providing up to 4 gigabytes of addressable space per 32-bit
>> process and approximately 18 exabytes of addressable space for 64-bit
>> processes.
>>
>> So if this is true, 32 bit processes cannot access more than 4GB of
>> memory....
>>
>> Ciao,
>> PaoloRick Towler wrote:
>>> pgrigis wrote:

>>>> Hi folks,
>>
>>>> we have pretty much exhausted the topic of memory
>>>> allocation on Windows and Linux, but I don't remember
>>>> any discussion abut this on Mac OS.
>>
>>>> So, I am using IDL 6.3 on Mac OS X 10.4.11.
>>
>>>> I tried allocating as many 1GB array as possible,
>>>> and it failed after 3 successful allocations.
>>>> Now, the "Activity Monitor" indicates that at this point
>>>> I have 3.6 GB of memory used and 3.4 GB free.
>>>> So I am wondering why cant'I allocate a couple more
>>>> of 1GB arrays?
>>
>>> I'm not a macatista, but a quick google search reveals that as of 10.3,
>>> the per process memory limit in OS X is 4GB. That squares with what
>>> you're seeing. Someone more in the know might be able to tell you
>>> if/how this can be tuned. For instance using "setrlimit".
>>
>>> -Rick

Subject: Re: memory allocation on Macs
Posted by Kenneth P. Bowman on Fri, 02 May 2008 19:39:48 GMT
View Forum Message <> Reply to Message

In article
<c76c7777-b3ab-4645-b22a-c90be59a1979@t54g2000hsg.googlegroups.com>,
Brian Larsen <balarsen@gmail.com> wrote:

> Rick,
>
> On this point
>
>> In any case, if you need a single process size to exceed 4GB, use a 64-
>> bit OS.
>
> at the risk of being a little ignorant here. I ran this test on
> Leopard (10.5.2) which is reported to be a 64-bit OS...
> http://www.apple.com/macosx/technology/64bit.html
> Making this an issue of having 32-bit IDL not a 32-bit OS. Right?

Yes, I'm sure I'm not the only one waiting for ITTVIS
to make a 64-bit version for Mac OS X.

Ken

Subject: Re: memory allocation on Macs
Posted by ekvys101 on Fri, 02 May 2008 23:57:14 GMT
View Forum Message <> Reply to Message

> FYI, this is a Xeon machine in Mac OS X 10.4, so it is a 64 bit
> processor
> in a 32 bit OS running a 32 bit application.

We asked ITTVIS a month or two ago when they would support 64 bits on Mac OS X and they said sometime this calendar year... I think it will require OS X 10.5+ since 10.4.etc was not fully 64 bit.

Corinne

Subject: Re: memory allocation on Macs
Posted by Karl[1] on Mon, 05 May 2008 20:23:57 GMT
View Forum Message <> Reply to Message

On May 2, 1:13 pm, pgri...@gmail.com wrote:
> Yes, you're right that I can allocate all the 7 GB (and more) in
> different IDL
> sessions. So there seems to be a limit indeed on how much memory one
> single
> IDL session (or in general , one process) can use up, but there isn't
> a limit for
> total usage (which, though I am sure there are a number of technical
> reason
> for it, seems a bit silly, after all if the system as a whole can
> access more
> than 4 GB, why shouldn't parts of the system be allowed to do the
> same?)

Because it is a 32-bit application. One key difference between 32-bit and 64-bit applications is that the pointers maintained by a 32-bit application are 32 bits in size, and the pointers maintained by a 64-bit application are 64 bits in size. This happens at compile time. So, your 32-bit application simply cannot address more than 4GB at a time due to its fundamental pointer size. Note that a 64-bit application will have a larger storage requirement due to the larger pointers.

The memory management unit on the 32-bit CPU, something that you cannot directly access, can address more than 4GB worth of RAM since it can map more than 4GB among several processes. Here, it is probably mapping larger chunks of memory, or pages, rather than individual bytes, so it isn't as hard as it sounds. But it is the MMU

that locates the memory pages assigned to a 4GB process among the 7GB of installed memory and translates their physical addresses to 32-bit virtual addresses for the 4GB process.

While there are lots of ways to emulate bigger address spaces and ways to fit bigger problems onto small machines, it may often be much easier to move to a 64-bit address space.

Karl

> FYI, this is a Xeon machine in Mac OS X 10.4, so it is a 64 bit
> processor
> in a 32 bit OS running a 32 bit application.
>
> Anyway, thanks to all. I can cope with reading a few arrays off the
> disk
> from time to time.
>
> Ciao,
> Paolo
