
Subject: Re: memory allocation on Macs
Posted by [pgrigis](#) on Mon, 05 May 2008 20:49:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Karl wrote:

> 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

OK, I guess I see the logic here: since the application cannot access this, the 4GB stands as a hard limit, and it makes more sense for the vendors to just move the application to 64 bits than implement fancy techniques.

Thanks for your explanations,
Paolo

> , 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
