
Subject: Re: Out of memory panic
Posted by [mdelfede](#) on Sat, 19 May 2012 20:41:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Sat, 19 May 2012 22:28: Actually swapping is not the problem in many cases. We have nowadays machines with 6-8 GB of RAM running 64-bit Windows editions. So, in many cases, the address space for a 32-bit application is really exhausted before physical RAM runs out.

I admit, that the machines with just 2-3 GB of RAM would experience severe swapping before getting to the limit.

I think your solution B1 (the manual one with calling a function to get the current allocation footprint of the process) would best fit my purposes. This way I could skip trying to allocate buffers for the large data sets after meeting the limit (-- which BTW for me is about 1GB less than available physical RAM.) Is there such a function in Core?

I don't know... you should ask Mirek. Probably there's something like that somewhere....

Quote:

What I would really love to have, is your Protect -package working with MSC 64-bit target compilation, which would take the address space pain out for good. Currently my application is limited to 32-bit exes...

eh... me too, but 64 bit MSC don't support inline assembler, so I don't see an easy way to support Protect package.

I'm forced to stay with 32 bit too because of that.

I could try with external assembler functions, but I'm afraid those would be by far easier to defeat. And, the other solution, encrypt the whole stuff using PE data to do it as professional solutions do would be too much work for me now.

I even don't know if you can mix 32 bit and 64 bit modules, but I guess not.... But if you app allows it you could separate it in a couple of executables, one (32 bit) protected and the other one (64 bit) unprotected.

Ciao

Max
