

Re: Why is Base3.exe using 700+ K in windows task manager ?

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004-01/0712.html>

From: Beth (*BethStone21_at_hotmail.NOSPICEDHAM.com*)

Date: 01/18/04

Date: Sun, 18 Jan 2004 05:27:52 -0000

The Half A Wannabee wrote:

> *Hey Rene or any other Spasm users, or maybe this is an Windows issue ?*

[It's called "RosAsm" these days but that's not particularly important, of course :)]

> *I am running the Spasm Base3.exe. This exe is 14K including source.*

But the

> *Window Taskmanager reports its using 700kb. How can this be ?*

Rene..if it

> *takes not much time, can you explain it to me ? Are there declared this much*

> *memory inside the base3.exe?*

No, not at all, the figure reported by Windows Task Manager is the _total_ amount of RAM an application uses...this figure is usually _greatly_ larger than the file size because it includes many things...

For example, if you call any Kernel32 API (which, basically, you more or less have to do, as "ExitProcess" and things like that are in Kernel32.DLL :) then Kernel32.DLL is _also_ loaded into RAM...if you have a GUI interface then User32.DLL is responsible for the user interface elements and GDI.DLL is needed for painting windows...hence, at least these three DLLs tend to be loaded into RAM for even the simplest GUI program...

If you're on an NT-based version of Windows (which includes 2K and XP, of course), then you can throw in "NTDLL.DLL" too...because, basically, those main DLLs like Kernel32.DLL on NT machines tend to just "call through" to NTDLL.DLL to do the actual work...not totally sure why Microsoft do it like this on NT machines but it's probably a "security" or "better OS design" thing or whatever...anyway, that one will automatically get loaded because – and this is another "biggie" towards that RAM size – loaded DLLs _themselves_ can load further DLLs (hence, on an NT-based machine, loading "Kernel32.DLL" – which is kind

alt.lang.asm: Re: Why is Base3.exe using 700+ K in windows task manager ?

of "mandatory", really – also brings along "NTDLL.DLL" because that's a dependent DLL of Kernel32 itself :)...so, in fact, it's quite possible when calling but a single API call that it starts off a "chain reaction" of DLL after DLL being loaded into your process's address space (the "usual" ones aren't too bad – because they are "sharable" (so, only one copy of Kernel32 is loaded to be shared by all the other DLLs which may, very likely, also be using those API :) – but the more "obscure" API Microsoft provide – you know, "Video for Windows" or creating an Internet Explorer window inside your application and that kind of thing – can often drag in a whole tree full of DLLs that, well, it can often be quite "silly" ...Microsoft, you see, don't particularly care about these things whatsoever – as you can probably tell – so they don't quite understand that some people might actually object to this kind of "laissez faire" who-gives-a-crap attitude of theirs)...

Plus, there's also "behind the scenes" memory stuff to also consider here because when you create a GUI window, Windows will, of course, need some RAM here and there to keep track of things like the window's title, position, class attributes and so forth...I'm not, I confess, too sure whether this is counted as "application RAM" or whether it's just thrown into the more general "system RAM" counters on the main Task Manager page...but, regardless, whether it's being counted as part of the application memory or Windows "system" memory, the point is that creating your windows and window classes and GDI resources and so forth is taking up some memory somewhere in the system...

Then, obviously, file size and RAM usage can differ wildly simply because the application itself allocates and deallocates memory as it is running...not that this, I think, is at all relevant to "Base3.exe" when I last looked at that example program because it doesn't do any of that...also, when using the stack – which is used an awful lot in Windows with lots of data – that also counts as part of your "run-time" RAM (although, again, not sure if Task Manager is actually included the amount of "committed" stack space used into its general "application RAM" figure :)...

Note that, in fact, 700KB is pretty good in Windows terms for a more or less "do nothing" application...it kind of tells me that you're probably not running XP here because I've tried a few tricks with this using real "do nothing" applications and you basically have lost at least 1MB to "overhead" and "OS fluff" and variables to keep track of windows and that kind of thing before you add any actual code to do anything...

Mind you, once all your DLLs are loaded in and your windows are created and so forth, then you only (usually) pay for this "overhead" once...that is to say, something like XP might automatically "eat" 1MB for an application before anything else actually happens in that program...BUT, once paid, the rest of the RAM usage does tend to be application RAM "proper", so to speak...stuff you know about

Re: Why is Base3.exe using 700+ K in windows task manager ?

alt.lang.asm: Re: Why is Base3.exe using 700+ K in windows task manager ?

directly like your allocations, stack use and, yes, raw file size...

If you've not already downloaded OllyDbg, then it's a thoroughly recommended program...and, also, you can use it to get a better picture of what's happening than Task Manager alone can provide...under OllyDbg's "view" menu, you can choose "Executable Modules" and that lists out what DLLs and stuff your program has loaded (plus, as I say, if there's any "chain reaction" of one DLL loading another DLL loading another, then you can see that happening when there's a whole bunch of DLLs you never asked for listed in the "modules" window :)...plus, OllyDbg has a "memory map" window that's also available from the "view" menu that gives a pretty complete picture of your application's "address space" (including, I notice, the automatic "system" stuff like the buffers for disk reads and the actual stack memory, right near the start of memory...note that every "address space" has this stuff at the bottom which is "reserved for system use"...all that area, though, is strictly "undefined" and an application shouldn't play with it (not least because I'd guess there's no "guarantee" given that memory there is used in the same way on different versions of Windows that playing around with it would make something "version specific" automatically...although, of course, OllyDbg can be more informative and look down there because it is a debugger and it's only reading it, anyway :)...)

But, after such "bad news", you probably now want to hear the "good news" (yes, there is "some", though not a great deal :)...open up your "Task Manager" once more and then go to the "Processes" panel, as you have already been doing...then, under "view", you'll find "select columns"...choose that and you should see a whole bunch of extra "statistics" you can ask about an application...choose "virtual memory size" as well as "memory usage"...you'll note that they differ (often substantially so, sometimes just by a small amount :)...)

And the reason for this is that Windows employs "virtual memory" throughout that "reserved memory" doesn't necessarily equate to actual physical RAM being used...as an example, so you get the basic idea, it was strictly not completely accurate to say "loading DLLs" above...in fact, they are "mapped"...and there is a subtle but useful difference there...what this means is that the area is "reserved" for that DLL – Kernel32.DLL, for instance – but Windows actually uses "demand loading" (like most virtual-memory-based modern OSes do :)...that is (this is simplified a little for clear example :), the area for the DLL is simply "reserved" to begin with and effectively nothing has been loaded...and then your application makes a call to "ExitProcess" or whatever in Kernel32.DLL, which triggers a "page fault" (because this memory is only "reserved" at the moment, it's not actually yet been loaded into actual physical RAM :) but this is a "page fault" that the virtual memory system automatically handles...upon trying to access this area, it realises that the application wants that part of the file and grabs a "page" of memory off the disk (this is a protected mode thing...memory is divided into

alt.lang.asm: Re: Why is Base3.exe using 700+ K in windows task manager ?

4KB (or 4MB can be selected on some later processors, but I don't think any Windows actually uses that possibility) "pages"...and, for simplicity, everything is just handled in 4KB chunks in most places and any "wastage" is just considered "acceptable loss" to Microsoft (although, to be fair, there are "trade-offs" involved where getting less "wastage" could end up consuming more "management" RAM and slowing things down...Microsoft aren't being too much their typical selves here – you know, throwing away 128MB on some pointless bitmap for no good reason – because the "trade off" here isn't actually that bad...I ain't saying it's perfect by any means but it's not as wasteful as would actually be more typical Microsoft, throwing away 1MB rather than the reasonably small 4KB...well, Intel set 4KB "pages" in the hardware itself...you know that if Microsoft made these chips, the page size would be 4MB or something and you'd need 1024GB just to load in the "loading screen" while it boots ;)...

This is why, as Randy notes in his "taking complete control of HLA" – when questioning how `_USEFUL_` it actually is to create "smallest programs" with HLA – if you create a 256 byte or 3KB ".com" or ".exe" file, if you click on "Properties" about that file, then (at least on all the newer Windows versions :), you'll see `_two_` figures...on is "size" and the other is "size on disk"...the "size on disk" figure is `_always a multiple of 4KB_` because of the "page size" and how the file system works with it and so forth...therefore, with the 3KB file, 1KB is "wasted" by Windows itself in just how it works and stores things...

Not that this, to my mind, means that one shouldn't make programs smaller...this is `_always_` a Good Thing...smaller programs: download faster, take up less disk space, take up less RAM, may `_run faster_` due to less "cache thrashing" (hey, I just made that term up and it's quite poetic and it rhymes too...a "cache thrash"...has a nice sound to it ;), can help with `_transport_` on removable media (a floppy's limited capacity is most obvious but it can also apply to CDs and Zip disks and DVDs when there's many, many files packaged together – as is often the case – that it shouldn't be totally forgotten...the arithmetic is quite simple, of course: if they are all half the size, then you can fit twice as much on a CD...or, alternatively, use `_one_` CD rather than two...or ten CDs rather than twenty when you've got an awfully big amount of data to move around ;)...blah–blah–blah...a small program that does the same job as a big program in far less space is always a better thing...just because it might be possible to always make things "byte perfect" or that people have "lots of RAM and hard drive" space is NOT really a valid excuse...indeed, if they have so much RAM and so much hard drive space then we can have `_millions_`, not just hundreds and thousands of things in the exact same space without having to "upgrade" anything in the slightest...

The only real reason not to at least attempt to "tidy up" a program to make better use of available resources is when we have a "no choice" situation...the boss wants it tomorrow and you just don't have

Re: Why is Base3.exe using 700+ K in windows task manager ?

time...well, if it can't be done in time then it can't be done in time (arguments that the boss is a bit of a short-sighted arsehole not to allow an extra day to halve the size which boosts so many _user-friendly_ attributes like faster downloading off the internet, _better multi-tasking_ with other applications (our programs no longer "own" the entire machine but it's amazing how many programs are written as if that wasn't the case because they once wrote too many DOS programs where, of course, you did have total "free reign" to use everything available and have no "other programs" to be worrying about...remembering that, on modern systems, a firewall, a virus-checker are almost _certainly_ sharing memory before we consider the amount of people running Messenger, ICQ, AIM and colleagues...then there's all those "tray" icons that practically everything you buy these days insists on installing to provide some useless "utility" ;), less cache and disk thrashing, the ability to store a lot more of these far smaller programs onto removable media...and so on and so forth...if you're months behind and you've got a bunch of programmers you can never shift to do any work and it's a _real_ deadline because, literally, the software _must_ be sent on that day to reach the CD writing people in time for Christmas, then you _might_ have a point...just there is an awful lot of "jobsworths" (as in "it's more than my job's worth to bend the rules", as coined by "that's life" on TV over here :) who are merely being "authoritative" for the sake of the feeling of wielding "power" over their "minions"...many things could be massively improved with but the smallest amount of very simple and easy "tweaking" to which some "boss" insists otherwise...the most obvious examples are when something gets a "critical update" or "patch" or whatever only a week or two after being released...there's 52 of those weeks in a year and if you're keeping people employed, you pay them for all those weeks (minus vacations)...one week is _nothing_ in the grand scheme of things...indeed, _if_ your programmers are "pulling a fast one" to be "lazy" then you should crack the whip a little...but if it's a 100% _genuine_: "look, give us a week _at most_ and we can work wonders on this program!" then it's actually _foolish_ not to consider it: "more haste, less speed", as the "patches" roll out and the users give up half-way through a download because it's taking too long and sales suffer because once put into the _reality_ of having to "multi-task" alongside ICQ and Outlook on an _ordinary_ machine that isn't the "bells and whistles" of the monster development machine, the thing falls apart and crawls like a snail...Aesop may have been around an awfully long time ago but his fables and their "slow and steady wins the race" morals _never_ go out of date ever...in fact, simply _ASKING_ your potential users one simple question like: "So what do you expect?" might take up a morning's worth of work to do the "surveys" but the _answers_ can end up meaning that you create a _targetted_ application that does only what is _actually useful_ and, yes, could potentially save _months_...think about, say, Rene working 5 years on some "feature" and then finding out that only he and 7 other people in the entire world were actually the only ones particularly interested in that "feature" (of course, Rene isn't

alt.lang.asm: Re: Why is Base3.exe using 700+ K in windows task manager ?

interested in "sales" or so he sometimes says and sometimes doesn't say, depending on who he's talking to)...a bit of "market research" _is_ boring and can eat into your coding time but it can help tell you _target_ exactly what you need to do...give a focus that helps strip off potentially weeks and even months from a project...users don't care much about your "retroactive superlinking pipelines" idea? Well, leave it out...or consider at least the _need_ to "tone it down a little"...if users are saying they don't want it – and you're convinced that they _would_ "if they only understood it" – then consider "starting small and gradually working big" with the idea to "introduce" them to it gradually until it dawns on them that, indeed, it's something they can't live without...

Basically, before doing anything, stop...take a deep breath...apply some "common sense" to whether it is or isn't a good idea...even a five minute stop can work wonders and, in the grand scheme of things, is not even a "blip" on the radar of your time (more time actually being wasted on going to the toilet or chatting around the water cooler about last night's TV "event" or smoking a sneaky cigarette when no-one's looking ;)...

There's no hard and fast rules (well, except, perhaps, for the rule that there's no hard and fast rules :)...sometimes taking your time is wasting time, sometimes it's the very difference between "good" and "excellent"...sometimes being in a hurry is _important_, sometimes it's the _very last thing_ you should be doing...and it has to ultimately boil down to "common sense" and "intuition"...which, indeed, are things that can't be easily put into some "flow chart" for the boss or "cashflow forecast" for the accountant...but not being able to apply numbers to things does not mean those things don't exist...although, of course, ultimately you equally won't get far just being "touchy-feely" about everything as well...like everything else – though people will insist on not seeing it, for some reason – the actual task at hand is _balancing_ the chaos of the universe from all sides to walk that tightrope to the objective...you don't want to go one _NOR_ the other too much or you'll simply just fall off...

Microsoft, for example, don't need to consistently win the "Hugi size contest" with every single thing that they write...but it would go an awful long way, in my opinion, if they simply, you know, just _TRIED_...just gave a crap and gave it a go...just to _TRY_ once...

"_ALL_ we have is the attempt"

[Socrates or Plato or Aristotle, I can never remember which...Socrates, probably ;)]

Beth :)