

---

Subject: Re: A couple of issues with a test app  
Posted by [mirek](#) on Sat, 05 Jan 2008 09:34:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

cbpporter wrote on Sat, 05 January 2008 04:06 I'm not quite sure what you mean by "So in the end U++ does never return the memory to the system.". I thought that U++ solved memory allocation issues. My objects are on the stack, and when they are destroyed, they should be able to free all the hidden objects. Especially in the case of XML, where the data is allocated sequentially, and could be free in the same way. I never had problems with freeing block. I've written compilers which allocated huge amount of tokens and during their runtime heap sizes rapidly fluctuated between 10 and 400MB.

So if you could give some details why that memory can't be freed, I'm sure I'll be able to do something so that at least my app will have normal memory consumption. I don't want my app which with all data loaded will take up about 30MB to have 600 in the end, all because I had to parse 60MB of XML. Heck, gc would give a lot better results.

PS: I don't believe in swap

Well, heap size is one thing, comited memory another.

OK, details: Most of blocks average U++ code uses are concentrated in 4KB pages. It quite normal for such page to become free and it would seem that it could be returned to Win32.

ALAS, Win32 can only provide limited number of \*separate\* blocks. For 4KB blocks, that would mean at most 128MB allocated.

Therefore, we need to allocated these 4KB blocks in groups (the size of group in allocator gradually rises). Anyway, to release the group, it would require ALL pages in group to be free, which is statistically quite unlikely.

Sure, there is still possible that we have overlooked something in development, but it is not very likely. Actually, we have found this limit hard way a couple of years back, because originally U++ was capable of returning the memory back.

And, as far as I know, this policy (never return the memory) is quite standard (based on review of other allocators).

(Note that there is exception - very large blocks, >64KB, are actually returned to system on free - simply because they are allocated directly as single system block).

Mirek

---