

---

Subject: Re: Core 2019

Posted by [mirek](#) on Fri, 21 Jun 2019 07:16:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I think you cannot deduce too much about efficiency looking at "down" number. That will depend a lot on overall load of system, more the load, less this number as system will page out unused pages from your apps address space. So the current philosophy of U++ allocator is that this does not matter.

Further explanation. The function that any allocator is using to obtain address space from system is mmap and there is munmap that returns address space to system. Normally there is a threshold - if block is too big, its allocation is simply handled by mmap / munmap calls, meaning it is returned to the system at MemoryFree. If it is less than threshold, bigger chunk is mmaped from the system and then divided to smaller chunks (somehow).

Now what is different is that standard GCC allocator has threshold at 4MB. U++ allocator at 224MB. In practice, this means that if you alloc / free 5MB block in std, it gets released back to system immediately. With U++, blocks up to 224 MB are not returned to the system immediately. If they are really unused, this just means that system will retrieve them when there is a need for more physical memory.

Mirek

---