Subject: Re: How to respond when memory is exceeded
Posted by mirek on Sun, 17 Nov 2024 15:07:08 GMT
View Forum Message <> Reply to Message

This is a request that sound reasonable, but in practice is hard to implement, impossible to test and might have less impact than you think.

I mean, there are two reasons for out-of-memory situation:

- a bug. Some parameter in input data wrongly indicates that you have to alloc insane amount of memory and the allocation fails because it exceeds virtual memory. Your example falls into this category. So far I believe it is easier to account for that before the allocation (check the parameter) that trying to make the whole code out-of-memory resistant (e.g. check how String Stream::GetAll(int size) is implemented).

- a real memory exhaustion. That in hosts that we are interested means exhaustion of virtual memory space. In that case it is equally likely that the user part of host will crash too or that the computer will get virtually deadlocked trying to shuffle virtual memory or the process gets ultimately killed by OOM killer. Plus, to make things worse, it does not even has to happen in alloc! (Sometimes to really alloc the memory, you need to alloc it and then write to it...) So the effort you have invested in fixing all the code wents exactly nowehere.

(Fun fact: Last time I tried to demontrate futility of checking for out of memory with simple C code in Windows 10, I got blue screen as a result :)

Mirek