Subject: Re: CTRL + C = 659 Heap leaks
Posted by Novo on Mon, 15 Jul 2019 04:15:32 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 12 July 2019 03:44
In win32, if you run this in console and press Ctrl+C:

PS C:\xxx> ./Console
Here
Exit

Heap leaks detected:

--memory-breakpoint__ 736 : Memory at 0x0000000006b1d570, size 0x4CC = 1228
  +0 0x0000000006B1D570 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
 +16 0x0000000006B1D580 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
 +32 0x0000000006B1D590 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
 +48 0x0000000006B1D5A0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
PS C:\xxx>

This explains everything.
On Windows stack doesn't get unwinded, but descructors of global objects are still called. This is not a correct behavior.
In the old 32-bit Win ABI CRT was responsible for calling descructors of global objects. I do not know how this is implemented in the x64 ABI, most likely the same way.

When I strace my app in Linux I do see an exit_group(0) call on normal run (no CTRL-C).
I do not see any system calls when I press CTRL-C ...
This is an interesting topic ...
I guess, SIGINT is handled completely by glibc in Linux ...
Sending SIGINT directly to a process from another shell doesn't change anything ...