Subject: Re: CTRL + C = 659 Heap leaks
Posted by Novo on Thu, 11 Jul 2019 14:42:29 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 11 July 2019 04:22
So, this is console application in POSIX and you are pressing Ctrl+C to terminate it. AFAIK, Ctrl+C just calls exit, which is what is causing the leaks, as 'mybot' never gets destroyed.

Ctrl+C sends SIGINT, which causes "soft" termination of an app (unlike SIGKILL). It does all cleanup job for the process including stack unwinding (at least in x64 Linux).

Proof:
Memory leak report is a result of a call to UPP::MemoryDumpLeaks(), and it is called by

```
MemDiagCls::~MemDiagCls()
{
 if(--sMemDiagInitCount == 0)
  UPP::MemoryDumpLeaks();
}
```

So, stack gets unwinded, but in case of Xemuth memory doesn't get deallocated.
This is a bug with Xemuth's code. :blush: