Subject: Re: crush of the program
Posted by exolon on Wed, 28 Mar 2007 19:58:21 GMT
View Forum Message <> Reply to Message

BTW, a quick note which may be of help to those debugging these kind of problems - on linux, you can use a program called valgrind which will give a report like this:

==18964== Conditional jump or move depends on uninitialised value(s)
==18964== at 0x80947F0: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x8094DEB: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x809C7AC: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x809CCF8: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x809A004: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x809A592: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x8098C19: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x804CD68: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x804CDB6: (within /home/oisin/upp/out/GCC32.Gui.Shared/AnimatedHello)
==18964== by 0x4294EA1: __libc_start_main (in /lib/tls/i686/cmov/libc-2.3.6.so)
(many times, then, after a long while... for me, about 3 minutes (!) program appears, I let it run for a moment and quit)

==18964==
==18964== ERROR SUMMARY: 100 errors from 19 contexts (suppressed: 39 from 1)
==18964== malloc/free: in use at exit: 2,144,858 bytes in 30,435 blocks.
==18964== malloc/free: 492,498 allocs, 462,063 frees, 51,400,138 bytes allocated.
==18964== For counts of detected errors, rerun with: -v
==18964== searching for pointers to 30,435 not-freed blocks.
==18964== checked 2,146,892 bytes.
==18964==
==18964== LEAK SUMMARY:
==18964== definitely lost: 697 bytes in 42 blocks.
==18964== possibly lost: 543,752 bytes in 90 blocks.
==18964== still reachable: 1,600,409 bytes in 30,303 blocks.
==18964== suppressed: 0 bytes in 0 blocks.
==18964== Use --leak-check=full to see details of leaked memory.

It takes a huge amount of time though - for me running on a not too fast Celeron 2ghz or so in Ubuntu linux, loading AnimatedHello normally in Shared libs mode took about 10 seconds...

Apparently gcc has a memory debugging library called mudflap, but it seems to generate a lot of spurious errors in C++.