Subject: Re: crush of the program
Posted by mirek on Wed, 28 Feb 2007 08:25:39 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 27 February 2007 17:09
- usage of uninitialized/corrupted memory - in debug mode you get all kind of those helpers like 0xCCCCCCCC for allocated memory or 0xFDFDFDFD for deleted one, same allocation addresses, etc... in release you get random garbage and never really knows what to expect. Also allocated memory in debug mode has some guardians space which may catch occasional memory overruns, in release it's much easier to corrupt your memory.

BTW, the most troublesome bug is "read past end of buffer". While there is a huge chance that U++ heap allocator catches writes, reads, especially one byte past end, are impossible to catch. Plus, the chance that you read byte in area that causes exception is very very low. Means it crashes once a week or so.

Once I was dealing with mysterious crashes of one of my application for 6 months, before identifying this.

Quote:
Also if you manage to get stack/memory/code dump of crash, it may be worth to compare it with symbol table to see if it does crash always on the same place, and examine the exact reason of crash.

In U++/Win32 you can call "InstallCrashDump" at the start of program. This will create the core dump that can be later analyzed in "Crash" utility (you should be able to compile it from uppsrc). So if you have any difficulity analyzing, you can try this. Crash will need "*.crash" dump file and map file of executable.

Mirek