Subject: Re: bug in latest svn
Posted by mdelfede on Sat, 03 May 2008 22:07:26 GMT
View Forum Message <> Reply to Message

Well.... found at least a dirty workaround.

In 'heap.cpp', line 148 :

```
//#ifdef _DEBUG
#if 1 // @@@@@ WORKAROUND !!!!!
void FreeFill(dword *ptr, int count)
{
 while(count--)
  *ptr++ = 0x65657246;
}

void FreeCheck(dword *ptr, int count)
{
 int c = count;
 while(c--)
  if(*ptr++ != 0x65657246)
   HeapPanic("Writes to freed blocks detected", ptr, sizeof(dword) * count);
}

#endif
```

And, (that's the real workaround....), in file String.cpp, line 22 :

```
static inline void MFree_S(void *ptr)
{
 MCache& m = mcache[1];
 ((FreeLink *)ptr)->next = m.list;
 m.list = (FreeLink *)ptr;
//#ifdef _DEBUG
#if 1 // @@@@ WORKAROUND !!!!!
#ifdef CPU_64
 FreeFill((dword *)ptr + 2, 32 / 4 - 2);
#else
 FreeFill((dword *)ptr + 1, 32 / 4 - 1);
#endif
#endif
 if(++m.count > CACHEMAX)
  MFree_Reduce(m, 1);
}
```

This 'hides' the bug, but I guess it's still there.
Theide run fine, and also my (few) test apps.

For Mirek : the bug should be OR in string functions, OR in Vector stuff, in particular in VectorGrow stuffs.

Here a small app that shows the behaviour....BUT, to test it you have to :
1) Build in Optimal mode (NO DEBUG flag set)
2) Change optimal mode debug flag to -O0 (otherwise it's undebuggable because of optimizations...)
3) Enable full debug info even in optimal mode
4) Enable DEBUG just in heap.cpp, and change allocation stuffs to the debugger ones (some changes in core.h, heapdbg.cpp and others...)
5) The hard stuff... the bug arises before main(), so you've got to disable some initialization stuffs (quite long work). If not, you'll have to follow some hundreds of calls before main().

After point 5 done, the app will execute the first loop, then show the error. Following this, you'll find mostly string and vectormap calls. Quite long to follow, yet, but better than debugging theide
I stopped here because your string stuffs are quite un-understandable (and uncommented....).
Here the test app :

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 char key[200];

 VectorMap<String, char *> aMap;

 int nKeys = 100;

 for(int i = 0; i < nKeys; i++)
 {
  sprintf(key, "Chiave di prova n.%d", i);
  RLOG("\n\n============================================================\
nAdding key " << key);
  RLOG("AllocTest before....");
//  AllocTest();
  aMap.Add(key);
  RLOG("Key added, AllocTest after");
//  AllocTest();
 }

 exit(0);
}
```

Commented AllocTest() stuffs are from this routine :

```
void AllocTest(void)
{
 RLOG("AllocTest -- Entering");
 const int numAllocs = 100;
 size_t sizeAlloc = 32;

 void *p[numAllocs];
 for(int i = 0; i < numAllocs; i++)
  p[i] = MemoryAllocSz(sizeAlloc);
 RLOG("AllocTest -- Memory allocated");
 for(int i = 0; i < numAllocs; i++)
  MemoryFree(p[i]);
 RLOG("AllocTest -- Memory freed");

}
```

These just do some dummy alloc/frees to trigger the bug.

BTW... I've got a suggestion here... just to ease the debugging.
I guess we should have alternate memory allocation stuffs that could be used (independently from DEBUG flag) to check the heap on demand. MemoryCheck() and memoryCheckDebug() doesn't do the job, even when enabled by hand. And, the best would be to have a switchable pointer-checking functions for all Upp containers, called entering and leaving each container's method.
All that could be switched by a CHECKPOINTERS macro, and should check container's internal pointers and free heap.

Another 'stylish' stuff... many functions that resides in heap.cpp should (IMHO) belong to heapdbg.cpp.

Ciao

Max

Well, after some tests on 32 bit (thanks Bytefield), I've seen that workarount don't work for 32 bit builds... so, better to stay on SVN 219 build (for 32 bit) and SVN218 build (for 64) up to the stuff is fixed.

Max