Subject: Re: Heap errors behavior is dependent on target machine.
Posted by mirek on Mon, 30 Nov 2015 09:03:01 GMT
View Forum Message <> Reply to Message

Getting out of options.

The main hypothesis here is that we are detecting leaks too early.

Still, we can check this:

In the file with those std::string globals, put something like

```
struct MyInitChecker {
    MyInitChecker() { printf("Module initialized"); }
    ~MyInitChecker() { printf("Module deinitialized"); }
};

static const MyInitChecker myinitchecker;
```

then at the end of Core/heapdbg.cpp change destructor:

```
MemDiagCls::~MemDiagCls()
{
 if(--sMemDiagInitCount == 0) {
                printf("Now checking for leaks");
  UPP::MemoryDumpLeaks();
        }
}
```

Also, there are some details not yet provided:

- what is that "compatible" CPU?
- is the system updated to current version and it is exactly the same?
- are there any peripherals using serial communication that are not on Dell?
- what is that shared library?

Last but not least, it is entirely possible that the library leaks by design. In that case, it can be just bad luck and not really fixable. Well, in reality, leaving some global leaks is still considered "normal" in mainstream C++.