Subject: Probable nasty bug with StringBuffer
Posted by mdelfede on Thu, 12 Aug 2021 16:07:59 GMT
View Forum Message <> Reply to Message

This snippet:

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

using namespace Upp;

String test()
{
 StringBuffer sb(10);
 sb[0] = 'a';
 sb[1] = 'b';
 sb[2] = 'c';
 sb[3] = 'd';
 sb[4] = 'e';
 sb[5] = 0;
 sb[6] = 'f';
 sb[7] = 'g';
 sb[8] = 'h';
 sb[9] = 'i';
 return sb;
}

CONSOLE_APP_MAIN
{
 String s = test();
 Cerr() << "len(s)=" << s.GetCount() << "\n";
}
```

prints 10 in Linux with GCC and 5 in windows10 with msvs19 compiler, 32 bit.
Tracing it I found that converting sb to string just before test() return uses strlen(), which obviously stops at '0' byte in string.
It used to behave correctly before (msvs15 compiler) but now because of a problem in SIMD_SSE2.h I can't compile
with it anymore.
This problem appeared in Cypher bazaar (and also UppHub) package, which make encrypted strings containint '0' bytes to be
truncated.

Ciao

Massimo

p.s.: quick solution is to do a 'return String(sb)' instead of 'return sb', but IMHO this is a dangerous

bug, as used way should be correct anyways...

p.p.s.: clang compiler brings correct result.
