Subject: String & strlen optimization, force_inline in defs.h
Posted by mirek on Fri, 02 Dec 2011 08:53:46 GMT
View Forum Message <> Reply to Message

When strlen is intrinsic and applied to string literal, both compilers are able to replace it with constant.

That has a huge impact when we do things like

String s = "Hello ";
s << "world!";

because it is possible that we completely avoid scanning literal for its length (second command is then equivalent to Cat("world!", 6)).

Now this always worked (without me actually knowing quite well in GCC, except some String methods that did have strlen hidden in .cpp file.

Not so easy with MSC, which stubbornly refuses to inline some things and needs strlen to be activated as intrinsics.

Anyway, by combination of techniques, both compiler should be now able to optimize out strlen in most common cases.

In the process, I have added "force_inline" macro (thanks for hinting that out, Novo) to hide differences between MSC (__force_inline) and GCC (declspace(always_inline) inline).

Mirek

Subject: Re: String & strlen optimization, force_inline in defs.h
Posted by Novo on Fri, 02 Dec 2011 18:57:39 GMT
View Forum Message <> Reply to Message

Another trick:

```
template <int N>
int strlen__(const char (&s)[N]) { return N; }

Cout() << strlen__("asdf") << EOL;
```