Subject: Critical section in PteBase::PtrAdd
Posted by qapko on Mon, 11 Jan 2016 11:18:06 GMT
View Forum Message <> Reply to Message

Hello,

out of curiosity I have studied the implementation of Ptr template and noticed, that in PteBase::PtrAdd the critical section sPteLock is leaved at the beginning of the else block, just before writing to "prec" and its subitems. Is it OK? If yes, why? At first glance I would suppose, that because of some risk of race condition here, the sPteLock should be leaved at the end of the else block. Am I missing something?

Gabi

Subject: Re: Critical section in PteBase::PtrAdd
Posted by mirek on Mon, 11 Jan 2016 12:52:38 GMT
View Forum Message <> Reply to Message

qapko wrote on Mon, 11 January 2016 12:18Hello,

out of curiosity I have studied the implementation of Ptr template and noticed, that in PteBase::PtrAdd the critical section sPteLock is leaved at the beginning of the else block, just before writing to "prec" and its subitems. Is it OK? If yes, why? At first glance I would suppose, that because of some risk of race condition here, the sPteLock should be leaved at the end of the else block. Am I missing something?

Gabi

Hi,

I believe that the original reasoning was:

You should use some other pointee related mutex when taking pointer to pointee, because pointer is R/W and that is why this is a non-const operation. And in that case, it is clear that this is the first pointer and we know (because of external mutex), that no other "first" pointer will race with us. Therefore it is possible and better to release mutex early, as 'new' can be lengthy operation and we are using single mutex for ALL pointers.

Anyway... The optimization is very small and now, years later, I am not sure it is worth risking that above reasoning is correct in all cases... I have changed the code to use the mutex for the whole block.

Thanks,

Mirek

P.S.: Funny. Previous change to that file was 8 years ago... :)

Subject: Re: Critical section in PteBase::PtrAdd
Posted by qapko on Mon, 11 Jan 2016 14:01:07 GMT
View Forum Message <> Reply to Message

Hello,

your original reasoning makes sense to me. Maybe it would be better not to "fix" it at all :)

Thanks for your reply.

Gabi
