Subject: Re: Thoughts about alternative approach to multithreading
Posted by Mindtraveller on Sun, 05 Jul 2009 19:43:45 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 03 July 2009 20:49I would like to, but right now I seem to be unable to describe it right. The problem was that it was user driven application and there are problems basically with "queue lag".

Maybe that the heart of problem is (was) the fact that it worked in "post" mode (not "execute") - messages (callbacks) being posted and not waiting for completition. Too often I ended with wrong events in the queue...

BTW, without posting, your method is equivalent to one mutex per instance and locking for any method call...

Mirek

1. What do you mean by "queue lag"? In my case case calling asynchronouse callback (this means: copy arguments, awake thread, post arguments and start execution), is almost as quick as calling U++ callback (I`ve posted these results above).
Personally I denied ANY types of events because I consider them absolutely artifical. The only thing which is really needed is executing some callback. So I had no problems identifying any types of events. I will appreciate any example where this approach fails (of course avoiding boundaries mentioned).

To be more precise, I takes more than single mutex per thread as there`s a queue and it really needs a semaphore.