Subject: Re: bug in CoWork since C++11
Posted by mirek on Sun, 07 Aug 2016 18:36:31 GMT
View Forum Message <> Reply to Message

crydev wrote on Sun, 07 August 2016 11:50mirek wrote on Thu, 04 August 2016 21:58"Quit thread"

- this is interesting. It looks like you are quiting the thread that spawned jobs before it has the chance to finish.

Is not it possible that you are calling Finish (perhaps via ~CoWork) from other thread than the one that scheduled the work? Or in other words, have CoWork instance shared between threads?

I had two instances of Finish in my code, and I was sharing the CoWork instance for different purposes. Is that not a good idea? It used to work fine.

mirek wrote on Thu, 04 August 2016 21:58
It is true that between 'classic' and 'C++11' I have changed the logic so that each 'master thread' has its own pool of worker threads, to avoid work stealing. Perhaps it was not a good idea after all..

Has the way CoWork should be used changed with the C++11 way of U++? I see a thread pool as an object that makes sure tasks you give it are dispatched into other threads, regardless of where the tasks come from. Does it work differently with CoWork? I also saw a CoWork changed commit this morning. Did anything important change?

Thanks

crydev

Well, I rethought the whole thing... I guess new iteration will work just fine again.

It is not 100% finished yet, maybe wait till tomorrow.

Mirek