
Subject: Re: CoWork and shared memory
Posted by [mirek](#) on Sun, 26 Feb 2012 08:56:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

peek wrote on Sat, 25 February 2012 14:05Hello Mirek

So this way have Cowork processes to end fast because they work in any way in the same thread that main program?

I mean, to do:

```
App::~~App() {  
    finish = true;  
    ...
```

```
// and in Cowork function
```

```
if (finish)  
    return;
```

is useless as destructor is not accessible until Cowork variable is destructed and that is done when all Cowork calls have ended.

No, this still works - CoWork has no clue what the processing is and this one simply makes the processing "nop" and it ends quickly. The loop would still go through all iterations, thread will still perform them, but each iteration job will end immediately...

Quote:

A question: If the Cowork jobs lasts time a solution could be to run Cowork in a separate thread?

Well, not really, if you want a clean exit from the app...

The problematic thing is that for clean exit, you need cannot use any form on thread abortion - it has to exit through its normal exit point; otherwise it would result in resource leaks.

Mirek
