Subject: Re: WhenAction() -> ProcessEvents() -> WhenAction() -> hang, crash :-)
Posted by Oblivion on Mon, 09 Dec 2019 11:00:59 GMT
View Forum Message <> Reply to Message

Hello Frank,

Adding to what Koldo and Mirek said,

Quote:
What is the most appropriate container in U++ to implement a FIFO queue? T

IME, BiVector for FIFO/LIFO queue of moveable types, and BiArray for FIFO/LIFO queue of non-moveable/quirky types.
BiVector and BiArray are bi-directional random access containers, allowing adding or removing elements from both ends in a constant amortized time.

Quote:
Hmm... I can see that U++ has its own class Mutex and ConditionVariable. And they appear to be less convoluted (objectified) than the C++11 std::mutex and std::condition_variable (geez... if I didn't know the bare C libpthread version, I would probably just shake my head in disbelief).

IMHO, as I said elsewhere, you should really look into Upp::AsyncWork or Job. I'd suggest using Job here, not because I am the author of it :lol: , but from what you describe, it seems that it suits your needs better:

- Unlike Upp::AsyncWork, it does not require a pre-allocated thread-pool. It is a single, scope-bound worker thread that can be used as a dedicated thread.

- Unlike Upp::AsyncWork, it is guaranteed to run the desired function in another thread.

- Unlike Upp::Asyncwork, it has an internal latch: if a work is already in progress, then calls to Job::Do method will simply return false.

- Just like AsyncWork it has a result gathering, thread cancellation and exception propagation mechanisms.

I've attached the Job package and the multithreaded version of U++'s AnimatedHello example, running on Job. It demonstrates a one way of constantly updating the display from another thread, and handling gui locks/threads. Hope it helps.

(Note it can also be written almost identically using Upp::AsyncWork. But you'll have to provide a latch yourself, and also it will bring in the overhead of a pre-allocated thread pool)

Use CTRL + D to start animation, and CTRL + C to stop animation.

And let me know if you encounter any problem or have any questions.

Best regards,
Oblivion

File Attachments
1) Example.zip, downloaded 259 times