Subject: thread, virtual class and exception
Posted by bonami on Mon, 07 Apr 2008 08:48:28 GMT
View Forum Message <> Reply to Message

the old exception problem. i clarified the code and got different results after changing it between different senarios.
create a thread and pass an object's address. it is either a normal class or derived from a virtual class. the thread calls the object's function, which throws an exception. this exception sometimes causes the problem (program failure).

in main.cpp change lines below to see different behaviors. sometimes same code after changing and changing back gives different results.
line 23, virtual or not.
line 68, "new clss;" or "static_cast<clss *>(a);" for whether derived.
line 67, change it to "#if 0" usually makes the program run.

my project stalks here if this problem cannot be solved. thank you very much.

File Attachments
1) test.zip, downloaded 470 times

Subject: Re: thread, virtual class and exception
Posted by mrjt on Mon, 07 Apr 2008 10:53:37 GMT
View Forum Message <> Reply to Message

I'm afraid your example isn't terribly clear, but I think I can guess what you are trying to do.

You have a server, and various types of socket class. The server listens for a connection, creates the correct type of class to handle it and then runs a thread, passing the class object. These socket classes throw exceptions that you need to display on the GUI. Is this correct?

I think the main problem is that you are attempting to update the GUI from the thread. In Upp only the main thread (the server in this example) is allowed to do this. Also attempting to pass the messages up by exceptions is a bad idea, what you should do instead is pass a callback to a function in the main thread that can handle to GUI updates:
void fc(void *ss, Callback1<String> WhenException); // Thread function
void OnException(String exc); // Updates GUI
I'd also try and avoid creating objects with new, passing as void * and then recasting. you could for instance use the One<> container for this:

void fc(One<base_class> obj, Callback1<String> WhenException); // Thread function
..
One<base_class> obj;
obj = new inheritor(a, b, c);
thread.Run(THISBACK2(fc, obj, THISBACK(OnException));
This avoids the heap-leak as the One container takes ownership of the pointer and deletes it when the thread finishes.

One reason this may not have been working before is that you missed the virtual keyword from the vt() function decl. in the clss class.

Hope that helps, I rewrote your example to check that this all works so I've attached my test if it's any use.

James

File Attachments
1) test.zip, downloaded 438 times

Subject: Re: thread, virtual class and exception
Posted by bonami on Tue, 08 Apr 2008 06:00:15 GMT
View Forum Message <> Reply to Message

thank you so much, James. after using this callback feature this small program does not crash. but my real project remains same.

i renamed my testcase as testme. it runs well. but after changing line 99, 74 and 107("ss" to "ez"), it crashes (containing my real project's code). and after changing back, it crashes, too.
so i wonder whether there are more subtle problems. or have i some mistakes in understaning you into my code?

i was amazed you guessed my project's purpose, since i just hid the details in order to show my problem.
not yet studied the One, but because i need to track all the objects, i think i will probably not use it.

File Attachments
1) test.zip, downloaded 422 times

Subject: Re: thread, virtual class and exception
Posted by mrjt on Tue, 08 Apr 2008 08:40:25 GMT
View Forum Message <> Reply to Message

Sorry, I can't compile that, but looking at the code everything seems fine so the problem is probably in your socket code/exception throwing. Try opening the socket before starting the thread.

Are you compiling with MingW? If so try MSVC, the dubugging works a lot better and it might help.

Subject: Re: thread, virtual class and exception

Posted by bonami on Tue, 08 Apr 2008 09:23:48 GMT
View Forum Message <> Reply to Message

runs ok if i do not socket(). but wonder why i am restricted of networking b/t threads.