Subject: Re: Sharing and Locking
Posted by gridem on Thu, 01 Apr 2010 07:14:04 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 21 March 2010 16:21
Well, this is the exact tradeoff of GC - you have lost the capability of destructors to manage resources.

I just want to show that you can manage resources carefully when you use one additinal inderection. While you cannot predict the lifetime of the object wrapper (class FileObject), you can manipulate with file resources in a predictable manner (class File). Class File will be destoyed on any invocation of Close method.
luzr wrote on Sun, 21 March 2010 16:21
Do not get me wrong. What you present is the 'mainstream' approach. In that case, however, the question is why not to use some real GC language instead...

I'm not quite sure about mainstream approach. I see 2 differences:

1. Mainstream doesn't use weak_ptr as reference to the object. In my programming life I see only shared_ptr in the production code.
2. Mainstream doesn't use additional layer for resource manipulation.

Also I cannot see autolocking in production code, but it's not relevant to discussion.
luzr wrote on Sun, 21 March 2010 16:21
What we are trying to do is exactly oposite. End of block closes the file (pipe, stream, whatever). That is why shared ownership (at interface level) is not recommended...

In my example I have another additional option: if someone wants to manipulate with the object and he grants the object than it can manipulate with it without any restrictions. The File will be closed implicitly if the FileObject will be destroyed and noone has access to it. I can also close file and release file handle or any resource handles explicitly even if someone tries to use it (the example demonstrates such behavior).
luzr wrote on Sun, 21 March 2010 16:21
(P.S.: Not quite sure "try/catch" is not needed there. Who will close the file if the exception leaves the block?)

Mirek

Try/catch is not needed in SetterThread because:
1. Other clients don't use Close method.
2. Setter doesn't convert from weak_ptr to shared_ptr.

Only if one of the statements will be incorrect, the exception can be thrown.

Also file will be closed automatically even Close method will not be invoked.

Regards,

Grigory.