Subject: Required simple MT lesson
Posted by mubeta on Sat, 21 Sep 2013 11:01:03 GMT
View Forum Message <> Reply to Message

I'm trying to write a multithreaded program to make it quicker and easier to manage communication with peripheral BUS.
I have read on this forum that multithreading is not easy to do and in fact I have the first problems.
The application I'm writing has GUI interface, which, at startup, starts the communication thread.

The communication thread uses only local variables or global variables. There is only one shared buffer, declared as global variable.
Well. The application work properly, till, from some buttons I open an windonw, for example an file selector. After this action, in short time the application crash.

The application sources are included. I hope in some easy and solving lesson. Thanks.

File Attachments
1) First.png, downloaded 985 times
2) Second.png, downloaded 1226 times
3) PierHB.zip, downloaded 392 times

Subject: Re: Required simple MT lesson
Posted by dolik.rce on Sat, 21 Sep 2013 12:01:26 GMT
View Forum Message <> Reply to Message

Hi mubeta,

I can't compile your application (it is windows only, right?), but I noticed few things that are thread related. They might not cause your trouble, but you asked for a lesson so I tell you anyway

1) Have a look at Thread::GetCount(). It does exactly the same thing as the AtomicInc/Dec(threads) in your code.

2) You are accessing iniKeys Vector from various threads without any serialization. This might cause trouble when CommLoop modifies it's content while GetUpdates reads from it or vice versa. You should use INTERLOCKED or INTERLOCKED_ macros (or some other mechanism) to ensure no two threads access the variable at the same time. Note that this might cause some slowdown or even deadlocks, so one has to be careful about what is accessed from where.

3) Not really a code issue, but when posting example code on the forum 5000 lines of code is quite a lot It would be great if you could try to create a smaller testcase that exhibits the problem next time

Best regards,
Honza

Subject: Re: Required simple MT lesson
Posted by mubeta on Sat, 21 Sep 2013 15:05:58 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 21 September 2013 08:01
I can't compile your application (it is windows only, right?)

Yes.

dolik.rce wrote on Sat, 21 September 2013 08:01
1) Have a look at Thread::GetCount(). It does exactly the same thing as the
AtomicInc/Dec(threads) in your code.

Developing software is not myfirst interest. Maybe I am little far from the latest U++ features.

dolik.rce wrote on Sat, 21 September 2013 08:01
2) You are accessing iniKeys Vector from various threads without any serialization. This might
cause trouble when CommLoop modifies it's content while GetUpdates reads from it or vice versa.
You should use INTERLOCKED or INTERLOCKED_ macros (or some other mechanism) to
ensure no two threads access the variable at the same time. Note that this might cause some
slowdown or even deadlocks, so one has to be careful about what is accessed from where.

Yes. I am transforming this application from an more old one. The code that you see it's just the
first try.
About the INTERLOCKED: How and where I must use it? For example: in case of two thread, I
must use INTERLOCKED in one of them, or both?
And again, sharing variables in two thread cause a problem of data consistency when one thread
change the value when the second it's reading, or there is also other problems? The shared
variables must be interlocked also if the two threads are developed for read only?

dolik.rce wrote on Sat, 21 September 2013 08:01
3) Not really a code issue, but when posting example code on the forum 5000 lines of code is
quite a lot It would be great if you could try to create a smaller testcase that exhibits the problem
next time

I don't agree so much, but I will care. Some times a complete view it's better than many talks. In
my case also better, due to this unknown language, (for me).

Thaks

Subject: Re: Required simple MT lesson
Posted by mubeta on Sat, 21 Sep 2013 16:01:08 GMT
View Forum Message <> Reply to Message

So, maybe I have some news more.

The crash is caused bye an object used for load an external dll library

```
else
{
  RLOG("3");
  commLoop = 0;
  RLOG("3.1");
  RLOGHEXDUMP((const void *)&vBusHandle, sizeof(vBusHandle));
  if(vBusHandle != 0) if(Bus().IsPortOpen(vBusHandle)) Bus().ClosePort(&vBusHandle);
  RLOG("3.2");
}
```

The Object Bus() is used only in one hide thread, of the multithread applications. But it is "declared"? "build"? using dli utility of U++.
For now, I modify the code for don't call the IsPortOpen function if the handle is 0, (and the application don't crash). But I belive that the problem is not solved.

Subject: Re: Required simple MT lesson
Posted by dolik.rce on Sat, 21 Sep 2013 17:06:30 GMT
View Forum Message <> Reply to Message

mubeta wrote on Sat, 21 September 2013 18:01So, maybe I have some news more.

The crash is caused bye an object used for load an external dll library

```
else
{
  RLOG("3");
  commLoop = 0;
  RLOG("3.1");
  RLOGHEXDUMP((const void *)&vBusHandle, sizeof(vBusHandle));
  if(vBusHandle != 0) if(Bus().IsPortOpen(vBusHandle)) Bus().ClosePort(&vBusHandle);
  RLOG("3.2");
}
```

The Object Bus() is used only in one hide thread, of the multithread applications. But it is "declared"? "build"? using dli utility of U++.
For now, I modify the code for don't call the IsPortOpen function if the handle is 0, (and the application don't crash). But I belive that the problem is not solved.

This seems not directly related to multithreading. The error you describe seems as if you try to call the methods of Bus() before Bus().Load("something.dll") or Bus().SetLibName("something.dll") was called. If I'm looking correctly, you only call Bus().Load() inside the loop if openBus == true, while this code is in the else branch.

By the way: You can easily test if the object was loaded correctly simply by code like "if (Bus()) { ... }". Also, it seems like overkill to me to load the dll in each loop iteration, couldn't it be only done once before the loop even starts?

About the INTERLOCKED:

You should wrap it around every piece of code that might access the shared variable at the same time. It is usually safe (and often necessary) to guard all accesses to the variable. Example:

```
// global mutex protecting variable iniKeys
Mutex iniKeysMutex;

void function1() {
 int length;
 INTERLOCKED_(iniKeysMutex){
  length = iniKeys.GetPut("Application").GetPut("UsedCommDriver").GetLength();
 }
 // do something with length ...
}

void function2() {
 String driver;
 INTERLOCKED_(iniKeysMutex){
  driver = iniKeys.GetPut("Application").GetPut("UsedCommDriver")
 }
 // do something with driver ...
}
```
Here you are sure that you can call function1 and function2 from any thread and even if it happens simultaneously, it will be safe.

Honza

Subject: Re: Required simple MT lesson
Posted by mubeta on Sat, 21 Sep 2013 17:35:05 GMT
View Forum Message <> Reply to Message

Quote:The error you describe seems as if you try to call the methods of Bus() before Bus().Load("something.dll") or Bus().SetLibName("something.dll") was called. If I'm looking correctly, you only call Bus().Load() inside the loop if openBus == true, while this code is in the else branch.

Uhmmm... I understand what do you mean. It's possible. But using dli, the declared dll functions

are loaded by default when the application start. In the program I use Load() function only for change the used library.

Subject: Re: Required simple MT lesson
Posted by dolik.rce on Sat, 21 Sep 2013 18:15:22 GMT
View Forum Message <> Reply to Message

mubeta wrote on Sat, 21 September 2013 19:35Quote:The error you describe seems as if you try to call the methods of Bus() before Bus().Load("something.dll") or Bus().SetLibName("something.dll") was called. If I'm looking correctly, you only call Bus().Load() inside the loop if openBus == true, while this code is in the else branch.

Uhmmm... I understand what do you mean. It's possible. But using dli, the declared dll functions are loaded by default when the application start. In the program I use Load() function only for change the used library.
Yes, it can be loaded automatically, but if I'm not mistaken it can only be done if you give a full path when you define DLIFILENAME or when the dll can be found on system path.

Honza

Subject: Re: Required simple MT lesson
Posted by mubeta on Sat, 21 Sep 2013 20:40:50 GMT
View Forum Message <> Reply to Message

In the end everything works well and resolved other errors crashes that do not show up anymore.

Changing from single thread to multiple threads was hard to me, due to my low level of knowledge around this world.

Thanks a lot.

Subject: Re: Required simple MT lesson
Posted by dolik.rce on Sun, 22 Sep 2013 06:08:27 GMT
View Forum Message <> Reply to Message

mubeta wrote on Sat, 21 September 2013 22:40In the end everything works well and resolved other errors crashes that do not show up anymore.

Changing from single thread to multiple threads was hard to me, due to my low level of knowledge around this world.

Thanks a lot.
Glad to hear that you resolved it. Rewriting existing application into multiple threads is sometimes difficult even for skilled programmer

---

Honza