Subject: Re: Multi-Thread Critial Section Problem
Posted by r1kon on Sat, 10 Dec 2011 12:58:43 GMT
View Forum Message <> Reply to Message

..actually...

I didn't know about mutexes when I made this post. But afterwards I did an hour or so of browsing and I think I got it figured out

The code example..is kind of irrelevant at this point. I know I'm doing it horribly wrong. We're talking stuff like...

Thread #1:
```
 myBuff[i][0] = "STRING";
```

Thread #2:
```
 myBuff[i][0] = "STRING TWO";
```

I'm just straight up reading it, willy nilly.

So here's what I found to do...created some test programs, I think I got it right...

Create some global mutex's (Mutex m1, m2, m3, etc)..I should need one for each buffer and (correct me if I'm wrong here) one for each array ctrl that I'm reading to/writing from (as to avoid reading/writing to the array ctrl at the same time..unless GuiLock __; does this?)

Then, in my code, change (as an example):

```
 myBuff[i][0] = "STRING";
```

to something more like this:

```
void myProgram::setMyBuff(String val)
{
  INTERLOCKED_(m1)
  {
    myBuff[i][0] = val;
  }
}


..

setMyBuff(t_("STRING"));
```

and similarly:

```
String myProgram::readMyBuff(int pos1, int pos2)
{
  String retVal;
  INTERLOCKED_(m1)
  {
    retVal = myBuff[pos1][pos2];
  }
  return(retVal);
}

...

String currentValue = readMyBuff(i,0);
```

And if I'm correct..then I need to do these two "functions" (read/write) for each of my threads..and use them EXCLUSIVELY to read/write that way they are ALWAYS syncronized no matter what. Is this correct? And if so, can I use different mutex's (m1, m2, m3, etc) for different buffers?

One more thing -- is it necessary for me to do this with array controls as well? Like so:

```
void myProgram::writeArrayValue(int pos1, int pos2, String val)
{
  INTERLOCK_(m2)
  {
    tabMainTab.arr.Set(pos1,pos2,val);
  }
}
```

So I'm serializing my access to the controls? Or is it just necessary to do it with the data?

Thanks so much in advance! YOu guys, and the other posts on this forum, have been SO helpful!

-Kevin