Subject: Re: Use same variable in different threads
Posted by dolik.rce on Fri, 10 Dec 2010 10:34:24 GMT
View Forum Message <> Reply to Message

Here is a little sample code for you, demonstrating the Mutex and Atomic. It is stupidly written (on purpose, of course) to show how bad it can go if you don't use the locking. See for yourself by uncommenting the "#define NOMUTEX".

In this example it is easy to see what is wrong with the code and it might be possible to fix it so it works better even without locking, but in many cases the errors are more subtle and harder to find. Also remember that if you access the shared variable from multiple places in you code, you should use the mutex around each of them.

Here is the code: #include <Core/Core.h>
using namespace UPP;

```
struct mySharedData {
 int a,b;
 String c;
};

Mutex m;
mySharedData data;
Atomic threadnum;

//uncomment this to see how it ends up when not using mutex
//#define NOMUTEX

void ThreadFunction(){
 int thisthread=AtomicInc(threadnum); //thisthread serves as an unique identifier of thread, just for the sake of the examples clarity
 for(int i = 0; i < 10; i++){
  Thread::Sleep(Random(10)); // Pretend some work...
#ifndef NOMUTEX
  m.Enter();     // Enter the section that accesses the shared data
#endif
  data.a++;
  data.c<<data.a<<": ";
  Thread::Sleep(20); //Let's pretend that some slow operation happens here (for example file access)
  data.b=data.a;
  data.c<<data.b<<" ["<<thisthread<<"]\n";
#ifndef NOMUTEX
  m.Leave();    // we don't need the exclusive access to data anymore, release the mutex
#endif
 }
}
```

```
CONSOLE_APP_MAIN {
 //initialize variables (no threads yet, so it is save to make it without mutex);
 threadnum=0;
 data.a=0;
 data.b=0;
 data.c="";

 // start the threads
 Thread::Start(callback(ThreadFunction));
 Thread::Start(callback(ThreadFunction));
 Thread::Start(callback(ThreadFunction));

 while(Thread::GetCount()); // the main thread should wait for other to finish, so we can see the
results
  Thread::Sleep(100);

 Cout()<<data.c<<"\n";
}
```

Honza
