Wait, you never said that I had to initialize the object before I used it!

devblogs.microsoft.com/oldnewthing/20120823-00

August 23, 2012

Raymond Chen

A customer reported that they were having trouble creating slim reader/writer locks at runtime. They simplified the issue to a short program:

```
#include <windows.h>
#include <iostream>
using namespace std; // this is just a quick test app
int a = 10;
// This version works
int working_version()
{
 SRWLOCK lock;
 AcquireSRWLockExclusive(&lock);
 cout<<"Acquired exclusively"<<endl;
 a++;
 ReleaseSRWLockExclusive(&lock);
}
// This one doesn't
int broken_version_1()
{
 SRWLOCK *lock = new SRWLOCK;
 AcquireSRWLockExclusive(lock);
 cout<<"Acquired exclusively"<<endl;
 a++;
 ReleaseSRWLockExclusive(lock);
 // ignore the memory leak - this is just a quick test app
}
// This one doesn't either
int broken_version_2()
{
 SRWLOCK *lock = new SRWLOCK[2];
 AcquireSRWLockExclusive(&lock[0]);
 cout<<"Acquired exclusively"<<endl;
 a++;
 ReleaseSRWLockExclusive(&lock[0]);
 // ignore the memory leak - this is just a quick test app
}
int main(int argc, char **argv)
{
 switch (argv[1][0]) {
 case '0': working_version(); break;
 case '1': broken_version_1(); break;
 case '2': broken_version_2(); break;
 }
 cout<<"a="<<a<<endl;
 return 0;
}
```

“What is the correct way of creating an `SRWLOCK` via the `new` operator?”

It wasn’t long before somebody noted that nowhere in the code is the function `InitializeSRWLock` called.

“Oh, yeah, thanks for catching that. It looks like one needs to initialize SRW locks which are created via the `new` operator. Otherwise it’s not required.”

No, the function is *always* required. It’s just that you got lucky in the local variable case and the initial stack garbage looks enough like an initialized SRW lock that you don’t notice the problem.

MSDN doesn’t say “You must initialize an SRW lock before using it” because the statement was believed to be so obvious that it never occurred to anybody that somebody would think the opposite was true. I mean, what’s the point of having an `InitializeSRWLock` function if initialization is not required? Think of it as one of the ground rules for programming: If an object has an initialization method, you must initialize the object before using it.

But just to be sure, I’ve submitted a documentation change request to add the requirement.

Bonus chatter: A common coding pattern is to wrap the low-level C-style object inside a C++style RAII-style object.

Bonus chatter 2: If you’re creating a highly-concurrent system, then you should probably put each lock on its own cache line.

Raymond Chen

Follow