Visual Studio 2005 gives you acquire and release semantics for free on volatile memory access

devblogs.microsoft.com/oldnewthing/20110419-00

April 19, 2011

Raymond Chen

If you are using Visual Studio 2005 or later, then you don't need the weird `InterlockedReadAcquire` function because Visual Studio 2005 and later automatically impose acquire semantics on reads from volatile locations. It also imposes release semantics on writes to volatile locations. In other words, you can replace the old `InterlockedReadAcquire` function with the following:

```
#if _MSC_VER >= 1400
LONG InterlockedReadAcquire(__in volatile LONG *pl)
{
    return *pl; // Acquire imposed by volatility
}
#endif
```

This is a good thing because it expresses your intentions more clearly to the compiler. The old method that overloaded `InterlockedCompareExchangeAcquire` forced the compiler to perform the actual compare-and-exchange even though we really didn't care about the operation; we just wanted the side effect of the Acquire semantics. On some architectures, this forces the cache line dirty even if the comparison fails.

Raymond Chen

Follow