

---

Subject: Re: Simple class to handle variables used by different threads

Posted by [Mindtraveller](#) on Mon, 24 Aug 2015 17:08:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Of course, making variable 'volatile' actually doesn't guarantee anything when we discuss multithreading issues. It just says to compiler 'please, don't optimize it as it may be changed outside your source code'.

The reason it 'sort of' works under modern x86/x64 systems is that modern CPUs change char/int variable with single CPU instruction which effectively avoids multithreading issues. If your variable is anything but char/int (+int64 under x64) you may have multithreading issues when variable is accessed while being updated in another thread.

You should also notice that under arm/mips systems you may have issues even with int32/int64 depending on particular CPU used.

I guess we have nice solution in U++. There's Atomic type which guarantees no multithreading issues with possible support with native OS/CPU commands. So why should we really duplicate its functionality?

---