Atitit。Cas机制 软件开发 编程语言 无锁机制 java c# php

[1. 为什么需要无锁操作 1](#_Toc15873)

[2. 硬件支持 cas atomic 2](#_Toc23660)

[3. 无锁编程（Lock-Free）就是在某些应用场景和领域下解决以上基于锁机制的并发编程的一种方案。 2](#_Toc24263)

[4. Volatile 内存屏障（Memory Barriers），就是它让一个处理器内的内存状态对其他处理器可见。 3](#_Toc1997)

# 为什么需要无锁操作

在某些时刻，你给这个变量赋一个64位的值。

|  |  |
| --- | --- |
| 1  2  3  4 | void storeValue()  {       sharedValue = 0x100000002;  } |

当你在32位的x86环境下使用GCC来编译这个函数时，将会生成如下机器码。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $ gcc -O2 -S -masm=intel test.c  $ cat test.s        ...        mov DWORD PTR sharedValue, 2        mov DWORD PTR sharedValue+4, 1        ret        ... |

这个时候你就会看到，编译器会使用两个单独的机器指令来完成这个64位的赋值。第一条指令设置低32位的0×00000002，第二条指令设置高32位的0×00000001.非常明显，这个赋值操作是非原子的。如果共享变量同时被不同的线程存取，就会出现很多错误：

同时读取sharedValue会带给它一系列的问题：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | uint64\_t loadValue()  {        return sharedValue;  }    $ gcc -O2 -S -masm=intel test.c  $ cat test.s        ...        mov eax, DWORD PTR sharedValue        mov edx, DWORD PTR sharedValue+4        ret        ... |

这里也一样，编译器会使用两条机器指令来执行这个加载操作：第一条读取低32位到eax，第二条读取高32位到edx。在这种情况下，如果对于sharedValue进行同时存储则会发现，它将导致一个**读撕裂**——即使这个同时存储是原子的。

众所周知，在x86环境下，如果内存操作数是自然对齐的，那么一个32位的mov指令就是原子的，但如果不是自然对齐，那么将是非原子的。换句话说，原子性的保证仅仅是当一个32位整数的地址正好是4的倍数的时候。

# 硬件支持 cas atomic

原子性不可能由软件单独保证--必须需要硬件的支持，因此是和架构相关的。在x86 平台上，CPU提供了在指令执行期间对总线加锁的手段。CPU芯片上有一条引线#HLOCK pin，如果汇编语言的程序中在一条指令前面加上前缀"LOCK"，经过汇编以后的机器代码就使CPU在执行这条指令的时候把#HLOCK pin的电位拉低，持续到这条指令结束时放开，从而把总线锁住，这样同一总线上别的CPU就暂时不能通过总线访问内存了，保证了这条指令在多处理器环境中的原子性。

软件级的原子操作，包括两大类系统调用，一类是基于对整数进行操作的atomic\_set/and/inc，一类是针对单独的位进行操作的set/clear/change\_bit，它们大部分都是基于硬件层面的CAS的指令实现的。

各种开发语言中（c,c++,java）基于操作系统提供的接口也都封装实现了对应的原子操作api，所以开发者完全可以直接调用各个开发语言提供的接口实现无锁程序。

# 无锁编程（Lock-Free）就是在某些应用场景和领域下解决以上基于锁机制的并发编程的一种方案。

# ****Volatile**** 内存屏障（Memory Barriers），就是它让一个处理器内的内存状态对其他处理器可见。

**volatile**

用volatile修饰的变量，线程在每次使用变量的时候，都会读取变量修改后的最的值。volatile很容易被误用，用来进行原子性操作

# 参考

原子操作 vs 非原子操作 - 博客 - 伯乐在线.htm

内存屏障（Memory Barriers） - 博客 - 伯乐在线.htm

【原创】无锁编程技术及实现-黑夜路人-微头条(wtoutiao.com).htm

java中volatile关键字的含义 - God Is Coder - 博客园.htm