计算机二级 C++辅导:旋转锁转自

InterlockedExchange((volatile long*)&bNoThreadUsing, TRUE);

}

虽然 InterlockedExchange 看起来是原子操作，但是这个比较也有可能插入其他的代码啊

下面是 while 部分的反汇编代码

while (InterlockedExchange((volatile long*)&bNoThreadUsing, TRUE) == TRUE)

013A13EE mov esi,esp

013A13F0 push 1

013A13F2 push offset bNoThreadUsing (13A7000h)

013A13F7 call dword ptr [__imp__InterlockedExchange@8 (13A819Ch)]

以前听说过“互斥旋转锁”这个东西，名字听起来很牛逼啊，让我心生敬仰。夜晚翻书，在 windows 核心编程里面看到了对这个东西的详细解释，记录在这里。

旋转锁的原型：

//线程之间进行互斥

bool bNoThreadUsing = true;

//旋转互斥锁

void Locker()

{

//以原子操作的方式来进行判断

while (InterlockedExchange((volatile long*)&bNoThreadUsing, TRUE) == TRUE)

{

Sleep(0);

}

/**

在这里写业务逻辑

**/

//这个地方是不是有可能另外一个线程也在 call dword ptr [__imp__InterlockedExchange@8 (13A819Ch)]？假如那个线程给执行成功了，这个时候，此线程又

//进行了下面这句比较，岂不是让这个线程给得利了？疑惑中

013A13FD cmp esi,esp

013A13FF call @ILT+320(__RTC_CheckEsp) (13A1145h)

013A1404 cmp eax,1

013A1407 jne Locker+4Ch (13A141Ch)

转自:计算机培训网