链滴

对 windows 互锁函数的补充

作者：localvar

原文链接：https://ld246.com/article/1525082201428

来源网站：链滴

互锁函数是多线程处理中最简单高效的手段之一，但这些函数的功能实在是太差劲了，要求稍微复杂点，就完成不了。比如if(n > 100) n++;这么简单的功能，它们就做不到。以前，为了达到互斥的目，我都要使用一个临界区，现在想想，这个方法简直太笨了！因为借助InterlockedCompareExchange甚至更复杂的功能都可以轻松实现，而这个函数曾经是我认为的最没用的互锁函数。例如前面的大于值才加1的功能就可以通过下面这个函数以原子的形式完成：

```
LONG InterlockedBiggerExchangeAdd( LONG volatile* Addend, LONG Value, LONG Compera
d )
{
    LONG lOrigin;
    do lOrigin = *Addend;
    while( (lOrigin > Comperand)
        && (::InterlockedCompareExchange(Addend, lOrigin + Value, lOrigin) != lOrigin) );
    return lOrigin;
}
```

这个函数比较*Addend和Comperand, 如果*Addend大于Comperand, 就给*Addend加上Value, 返值则是*Addend的初值。

仿照上面的例子，我们还可以写出InterlockedAnd、InterlockedOr和InterlockedXor的实现，这也我以前经常抱怨的，因为系统只在DDK中提供了它们！但话说回来，这三个函数还有更简单的实现方，因为汇编指令and、or、xor都支持lock前缀，如果直接用汇编实现的话，会更简单高效！