

# HW10

姓名: 陈锐林, 学号:21307130148

2023 年 11 月 10 日

## Question1:

线程的顺序可能会改变。

## Question2:

加上-d 标志后, 也会看到执行顺序的不同; 可能会看到死锁的发生。如把-l 提高到 1000, 经过多次试验都不会死锁; 但是提高到 100000. 就经常会死锁。

## Question3:

显而易见的, 随着-n 的增大, 即线程数的增加, 死锁会发生得更频繁。能保证不发生死锁的 n 只有 0 和 1 了, 因为再大的 2 在 Q2 中, 只要循环数多了一样会死锁。

## Question4:

(1) 避免死锁是因为采用了全局的顺序判断, 对 dst 和 src 进行判断, 觉得 lock 的顺序。(2) 特殊情况是因为不能对一个锁尝试上锁两次, 会死锁。

## Question5:

(1) 由下图看出来, ”./vector-global-order -t -n 2 -l 100000 -d” 命令, 用时为 0.05s。(2) 而随着线程数和循环数的增大, 所用时也会增加; 从这几组数据来看, 增加线程数的提升是非线性的、增加循环数的提升是线性的。

```
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 10 -l 100000 -d
Time: 0.98 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 10 -l 200000 -d
Time: 1.95 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 5 -l 100000 -d
Time: 0.27 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 5 -l 200000 -d
Time: 0.48 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 2 -l 100000 -d
Time: 0.05 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 4 -l 100000 -d
Time: 0.24 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 8 -l 100000 -d
Time: 0.76 seconds
```

## Question6:

(1) 用时会减短。(2) 我预测表现会更好, 因为不同的线程不再需要一直等待两个 vector。下图也表明如此:

```
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 10 -l 200000 -d -p
Time: 0.05 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 10 -l 100000 -d -p
Time: 0.03 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 5 -l 100000 -d -p
Time: 0.02 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 5 -l 200000 -d -p
Time: 0.05 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 2 -l 100000 -d -p
Time: 0.02 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 4 -l 100000 -d -p
Time: 0.02 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-global-order -t -n 8 -l 100000 -d -p
Time: 0.03 seconds
```

## Question7:

(1) 这里的 pthread\_mutex\_trylock 是没有必要的; 它在这里不会进入休眠, 而是持续跳转到 top, 这是浪费时间的。(2) 如果采用 -p 标签, 能看到 Retries 都是 0, 用时更少; 但如果不用 -p, 能看到表现并没有前几问的好。(3)(不 -p 情况下) Retries 个数是呈指数上涨的。

```
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 2 -l 100000 -d -p
Retries: 0
Time: 0.01 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 4 -l 100000 -d -p
Retries: 0
Time: 0.02 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 5 -l 100000 -d -p
Retries: 0
Time: 0.02 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 5 -l 200000 -d -p
Retries: 0
Time: 0.04 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 8 -l 100000 -d -p
Retries: 0
Time: 0.03 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 10 -l 100000 -d -p
Retries: 0
Time: 0.03 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 10 -l 200000 -d -p
Retries: 0
Time: 0.05 seconds
```

```
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 2 -l 100000 -d
Retries: 1333926
Time: 0.18 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 4 -l 100000 -d
Retries: 2622549
Time: 0.48 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 5 -l 100000 -d
^[[ARetries: 3262652
Time: 0.66 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 5 -l 200000 -d
Retries: 6308982
Time: 1.28 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 8 -l 100000 -d
Retries: 9909762
Time: 2.31 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 10 -l 100000 -d
^[[ARetries: 14710000
Time: 4.06 seconds
• chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-try-wait -t -n 10 -l 200000 -d
Retries: 29309218
Time: 7.95 seconds
```

### Question8:

(1) 主要问题在于，全局锁的存在导致，即使两个线程在不同向量上操作，也要等待。(2) 取两个情况对-p 和不带-p 进行测试，能看到：带-p 的 vector-avoid-hold-and-wait 是比较慢的，但是不带-p 的情况下，还是会比 vecotr-global-order 略快 (或差不多)，比 vector-try-wait 快得多。

```
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-avoid-hold-and-wait -t -n 10 -l 100000 -d
Time: 0.46 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-avoid-hold-and-wait -t -n 10 -l 100000 -d -p
Time: 0.17 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-avoid-hold-and-wait -t -n 2 -l 100000 -d -p
Time: 0.03 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-avoid-hold-and-wait -t -n 2 -l 100000 -d
Time: 0.07 seconds
```

### Question9:

首先这个版本没有调用任何关于 lock 的函数，而是使用了 fetch\_and\_add 来解决问题。但是这个版本在语义上应该是一样的，asm volatile("lock; xaddl %%eax, %2;") 表明了 fetch\_and\_add 的内部实现仍然是用到了锁相关的知识，可能只是更底层而已。

### Question10:

能看到这个版本倒也不是那么快，带-p 和不带都比 vector-avoid-hold-and-wait 慢；和 vector-global-order 和 vector-try-wait 比，也只有不带-p 是快。

```
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-nolock -t -n 10 -l 100000 -d
Time: 0.76 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-nolock -t -n 10 -l 100000 -d -p
Time: 0.15 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-nolock -t -n 2 -l 100000 -d
Time: 0.19 seconds
chenr1959@SK-20210701MSSI:~/NEWHW/ostep-homework/threads-bugs$ ./vector-nolock -t -n 2 -l 100000 -d -p
Time: 0.05 seconds
```