

并行程序设计与算法第三次作业答案

April 6, 2024

1 简答题

习题 1

如果一个程序使用超过一个互斥量，并能够以不同的顺序来获取互斥量，程序可能会死锁。

(1) 用两个线程运行程序，假设发生了下列顺序的事件，会发生什么？

Table 1: 习题 1(1) 事件表

时间	线程 0	线程 1
0	pthread_mutex_lock (&mut0)	pthread_mutex_lock (&mut1)
1	pthread_mutex_lock (&mut1)	pthread_mutex_lock (&mut0)

(2) 如果程序使用忙等待（采用两个标志变量）替代互斥量，还会有问题吗？

(3) 如果程序使用信号量替代互斥量，还会有问题吗？

解答：

(1) 程序将陷入死锁，线程 0 获取了 mut0，等待获取 mut1，线程 1 获取了 mut1，等待获取 mut0，而由于线程 0 正在等待 mut1，无法放弃 mut0，同理线程 1 无法放弃 mut1，因此陷入死锁。

(2) 仍然会有问题，两个线程会互相等待对方释放标志变量。

(3) 仍然会有问题，两个线程会互相等待对方调用 sem_post() 函数。

习题 2

考虑一个链表以及对链表进行的操作，下列的哪些操作可能会导致问题，如果会导致问题，请举例说明：

(1) 两个 Delete 操作同时进行

(2) 一个 Insert 和一个 Delete 操作同时进行

(3) 一个 Member（查询一个节点是否存在）和一个 Delete 操作同时进行

(4) 两个 Insert 同时进行

(5) 一个 Insert 和一个 Member 同时进行

解答:

(1) 如图所示，当线程 0 要删除节点 2 时，会拥有节点 1 和节点 3 的信息，而线程 1 将节点 3 删除了，导致线程 0 会出问题。

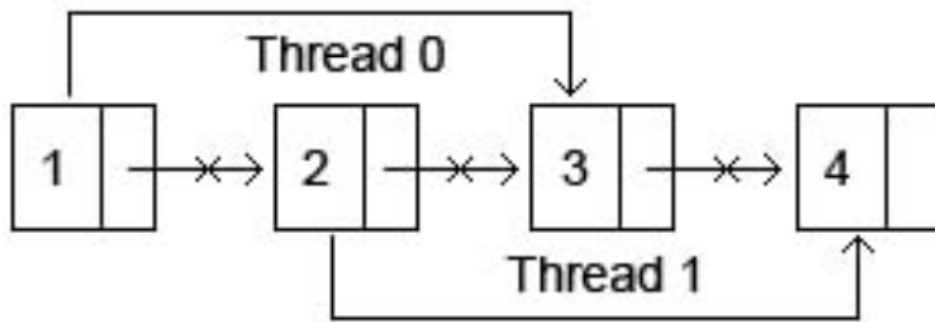


Figure 1: 习题 2(1) 解答

(2) 如图所示，线程 1 要在节点 1 和节点 3 之间插入节点 2，会拥有节点 1 和节点 3 的信息，而线程 0 将节点 3 删除了，导致线程 1 会出问题。

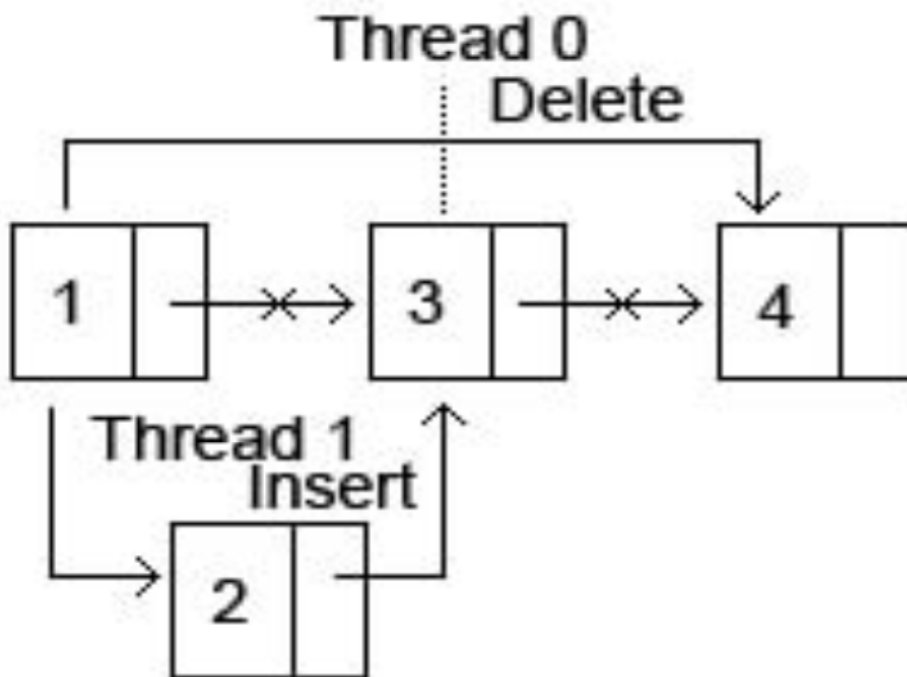


Figure 2: 习题 2(2) 解答

(3) 假设线程 0 要寻找节点 i，线程 1 要删除节点 i，在线程 0 找到节点 i 之后，返回结果之前，

线程 1 将节点 i 删掉了，这会导致线程 0 错误地返回节点 i 存在。

(4) 如图所示，线程 0 和线程 1 都想在节点 1 和节点 4 之间插入节点，会导致插入节点的顺序发生意外。

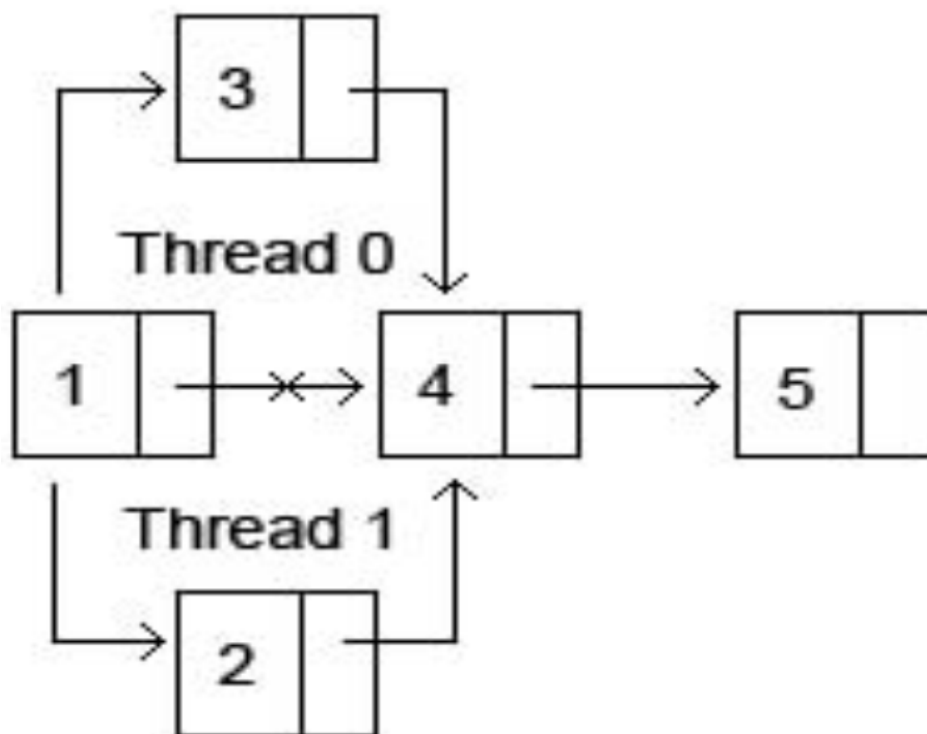


Figure 3: 习题 2(4) 解答

(5) 假设线程 0 要寻找节点 j，线程 1 要在节点 i 和节点 k 之间插入节点 j，在线程 0 搜索到节点 k 之后，返回结果之前，线程 1 插入了节点 j，这会导致线程 0 错误地返回节点 j 不存在。

习题 3

链表操作 Insert 和 Delete 可以拆成两个阶段，第一阶段两个操作都先找到要操作节点的位置，在第二阶段才插入或删除一个节点；也即第一阶段都只涉及对链表的读访问，只有第二阶段才写访问链表。如果在第一阶段使用一个读锁来锁链表，在第二阶段使用一个写锁来锁链表，假设该读写锁的实现是【获取写锁需要先释放读锁】，这样是否安全？

解答：

假设线程 0 想要删除一个节点，那么它先使用读锁搜索列表，找到了要删除的节点后需要获得写锁，但在放弃读锁和获得写锁之间的时间里，列表的状态可能会被另一个线程改变，例如线程 1 删除了线程 0 要删除的节点的前置节点，导致线程 0 所掌握的信息不再有效。

习题 4

在矩阵-向量乘法的例子中，采用 8000×8000 的输入，假设程序用 4 个线程运行，线程 0 和线程 2 被分配到不同的处理器上运行。如果一个缓存行大小为 64 字节或 8 个 double 数，在线程 0 和线程 2 之间会对向量 y 的任何一部分发生伪共享吗？如果线程 0 和线程 3 被分配到不同的处理器，会发生伪共享吗？

解答：

向量 y 的 8000 个元素会以线程 0 拥有 $y[0-1999]$ ；线程 1 拥有 $y[2000-3999]$ ；线程 2 拥有 $y[4000-5999]$ ；线程 3 拥有 $y[6000-7999]$ 的形式划分，要让线程 0 和线程 2 发生伪共享，就要让线程 0 拥有的其中一个 y 的元素和线程 2 拥有的其中一个 y 的元素处于同一缓存行中，然而一个缓存行只能缓存 8 个 y 的元素，显然线程 0 和线程 2 之间不会发生伪共享，同理线程 0 和线程 3 也不会。

习题 5

在矩阵-向量乘法的例子中，采用 $8 \times 8\,000\,000$ 的输入，假设一个缓存行的大小与习题 4 相同，同时假设系统有 2 个双核处理器，假设同一个处理器上的所有核共享一个缓存。

- (1) 如果只考虑一对线程共享一个处理器，可以有多少种不同的方式将 4 个线程分配到处理器上？
- (2) 在第 (1) 问的线程分配方式中，请找出有一种线程分配方式以及一种向量元素到缓存行的分配方式，使得没有伪共享的发生？

关于向量元素到缓存行的分配方式的说明：如下所示为两种不同的分配方式

Table 2: 分配方式 1

$y[0]$	$y[1]$	$y[2]$	$y[3]$	$y[4]$	$y[5]$	$y[6]$	$y[7]$
--------	--------	--------	--------	--------	--------	--------	--------

Table 3: 分配方式 2

?	?	?	?	?	?	?	$y[0]$
$y[1]$	$y[2]$	$y[3]$	$y[4]$	$y[5]$	$y[6]$	$y[7]$?

解答：

- (1) 如下图所示

- (2) 假设线程 0 和线程 1 分配在处理器 0，线程 2 和线程 3 分配在处理器 1， $y[0-3]$ 分配在同一个缓存行， $y[4-7]$ 分配在另一个缓存行，那么线程 0 和线程 1 的任何写都不会无效化线程 2 和线程 3 的缓存数据，反过来亦成立。

Table 4: 习题 5(1) 解答

处理器 0	处理器 1
0,1	2,3
0,2	1,3
0,3	1,2