

# 第十五章作业

## 15.2

15.2 考虑下面两个事务：

```
 $T_{34}$ : read( $A$ );  
      read( $B$ );  
      if  $A=0$  then  $B:=B+1$ ;  
      write( $B$ );  
  
 $T_{35}$ : read( $B$ );  
      read( $A$ );  
      if  $B=0$  then  $A:=A+1$ ;  
      write( $A$ ).
```

给事务  $T_{34}$  与  $T_{35}$  增加加锁、解锁指令，使它们遵从两阶段封锁协议。这两个事务会引起死锁吗？

$T_{34}$

```
1 lock-S( $A$ );  
2 lock-X( $B$ );  
3 read( $A$ );  
4 read( $B$ );  
5 if  $A=0$  then  $B:=B+1$ ;  
6 write( $B$ );  
7 unlock( $A$ );  
8 unlock( $B$ );
```

$T_{35}$

```
1 lock-S( $A$ );  
2 lock-X( $B$ );  
3 read( $B$ );  
4 read( $A$ );  
5 if  $B=0$  then  $A:=A+1$ ;  
6 write( $A$ );  
7 unlock( $A$ );  
8 unlock( $B$ );
```

不会死锁，因为只要一个事务拿到A的锁，另一个事务就无法开始

## 15.3

15.3 强两阶段封锁协议带来什么好处？它与其他形式的两阶段封锁协议相比有何异同？

- 强两阶段封锁协议要求事务提交之前不得释放任何锁
- 在强两阶段封锁条件下，事务可以按其提交的顺序串行化
- 严格两阶段封锁协议要求事务持有的所有排他锁必须在事务提交后方可释放
- 在严格两阶段封锁条件下，未提交事务所写的任何数据在该事务提交之前均已排他方式加锁，防止其他事务读这些数据

## 15.20

15.20 严格两阶段封锁协议带来什么好处？会产生哪些弊端？

- 好处：严格两阶段封锁协议只会产生无级联调度，恢复更加容易
- 弊端：调度的并发性相较于最优解可能更差