# 红色字体的为答案

设有两个事务T1，T2，其并发操作如下图所示，执行结果是什么？有什么问题？原因何在？

|  |  |
| --- | --- |
| T1 | T2 |
| 1. 读 A=8   ②  ③写 A=A-2  ④ | 读 A=8  写 A=A-3 |

答案：

结果为5， ---2分

存在丢失修改，T1对A的操作丢失，

根本原因是T1、T2的并发操作破坏了事务的隔离性。 ---3分

2 某停车场有多个入口和出口，车辆进入时从入口处由系统查询可用的停车位，从出口驶出时系统将其刚使用的车位标记为空车位。(5分)

|  |  |
| --- | --- |
| 指令 | 含义 |
| Get() | 返回一个空车位，如果没有空车位，则返回空值NULL；例如：x = Get()表示读取空车位到x中。 |
| Write(A,0) | 置停车位A为空 |
| Write(A,1) | 置停车位A为非空 |

假定入口处伪代码为：

x = Get();

IF( x ==NULL){ THEN return 0;}

Write(x, 1);

若两辆车在不同的入口处同时执行上述代码，会出现什么问题？如何解决？

答案：

答：如果没有停车位，或一个车的write操作在另一个车的get操作之前，或者两车的get操作在另个车的write操作之前，但get操作返回不同车位，不会产生冲突；若有两辆的get操作均在另一车的write位置之前并返回相同的停车位则冲突。(3分)

解决方案，采用封锁技术，给这段伪代码加锁。(2分)

Slock A

x = Get();

IF ( x ==NULL )

{

unlock A

return 0;

}

Xlock B

Write(x, 1);

unlock B

unlock A

如果学生用文字叙述也可以

3 设有两个事务T1，T2，其并发操作如下图所示，执行结果是什么？有什么问题？原因何在？

|  |  |
| --- | --- |
| T1 | T2 |
| **① R(C)=100** |  |
| **C←C\*2** |  |
| **W(C)=200** |  |
| **②** | **R(C)=200** |
|  |  |
|  |  |
| **③ROLLBACK** |  |
| **C恢复为100** |  |
|  |  |
|  |  |

T1将C值修改为200，T2读到C为200

T1由于某种原因撤销，其修改作废，C恢复原值100

这时T2读到的C为200，与数据库内容不一致，就是“脏”数据

并发操作带来的不一致性

4 设有两个事务T1，T2，其并发操作如下图所示，执行结果是什么？有什么问题？原因何在？

|  |  |
| --- | --- |
| T1 | T2 |
| ① R(A)=50 |  |
| R(B)=100 |  |
| 求和=150 |  |
| ② | R(B)=100 |
|  | B←B\*2 |
|  | W(B)=200 |
| ③ R(A)=50 |  |
| R(B)=200 |  |
| 和=250 |  |
| (验算不对) |  |

T1读取B=100进行运算

T2读取同一数据B，对其进行修改后将B=200写回数据库。

T1为了对读取值校对重读B，B已为200，与第一次读取值不一致

不可重复读

5 已知事务T1的封锁序列为：

LOCK S(A)…LOCK S(B)…LOCK X(C) …

UNLOCK(B) …UNLOCK (A) …UNLOCK (C)

事务T2的封锁序列为：

LOCK S(A) …UNLOCK (A) …LOCK S(B) …

LOCK X(C) …UNLOCK (C) …UNLOCK (B)

请判断T1和T2是否遵守两段锁协议，并说明理由。

T1遵守两段锁协议

LOCK S(A)…LOCK S(B)…LOCK X(C) …UNLOCK(B) …UNLOCK (A) …UNLOCK (C)

收缩阶段

扩张阶段

T2不遵守两段锁协议

6 为什么要先写日志文件？

1）写数据库和写日志文件是两个不同的操作

2）在这两个操作之间可能发生故障

3）如果先写了数据库修改，而在日志文件中没有登记下这个修改，则以后就无法恢复这个修改了

4）如果先写日志，但没有修改数据库，按日志文件恢复时只不过是多执行一次不必要的UNDO操作，并不会影响数据库的正确性

7 数据库在Tf时刻发生故障时，T1、T2、T3三个事务的状态如下图所示，数据库的日志带有检查点，请给出故障恢复的策略。

**Tc (检查点)**

**Tf(系统故障)**

**T1**

**T2**

**T3**

T1和T2在检查点之后才提交，它们对数据库所做的修改在故障发生时可能还在缓冲区中，尚未写入数据库，所以要REDO；T3在故障发生时还未完成，所以予以撤销，UNDO

8 设T1，T2是如下二个事务：

T1：A= A+2；B=B+1；

T2：A=A\*2；B=B\*2；

A的初始值为1，B的初始值为0

若这两个事务允许并行执行，请给出一个串行化的调度，并给出执行结果。

T1: A=A+2;B=B+1; T2:A=A\*2;B=B\*2; 结果为：A=6,B=2