

一、考虑下面的三个事务和它们的一个调度 S（时间从上往下依次增大）。判断 S 是否是冲突可串行化的调度？要求画出优先图并给出判断依据。

Transaction $T_0$	Transaction $T_1$	Transaction $T_2$
$r_0[A]$ $w_0[A]$		$r_2[A]$ $w_2[A]$
$r_0[B]$ $w_0[B]$	$r_1[A]$	$r_2[B]$ $w_2[B]$
	$r_1[B]$ $c_1$	
$c_0$		$c_2$

二、设 T1、T2、T3 是如下三个事务：

T1:  $A:=A+2$ ;  
T2:  $A:=A*2$ ;  
T3:  $A:=A^2$ ;  
设 A 的初值=0。

设三个事务都遵守两段锁协议，按 T3-T1-T2 的顺序执行，请给出一个不产生死锁的可串行化调度（在下表中填写相应代码即可），并给出最终 A 的结果。

时间	T1	T2	T3
1			
2			
3			
4			
5			
.....			

三、考虑两个事务 T1, T2。其中, T1 显示账户 A 与 B 的内容:

```
T1: Read(B);
    Read(A);
    Display(A+B).
```

T2 表示从账户 B 转 50 美元到账户 A, 然后显示两个账户的内容:

```
T2: Read(B);
    B := B-50;
    Write(B);
    Read(A);
    A := A+50;
    Write(A);
    Display(A+B).
```

令 TS (T1) ,TS (T2) 分别是事物 T1 和 T2 开始的时间戳, 并且 TS (T2) <TS (T1)。现有如下一个调度:

T1	T2
	Read(B);
	B := B-50;
	Write(B);
Read(B);	
	Read(A);
	A := A+50;
Read(A);	
	Write(A);
Display(A+B).	
	Display(A+B).

请给出每一步骤中 W-TS(A), R-TS(A), W-TS(B), R-TS(B) 的取值。(注: Display(A+B)仅表示显示账户 A 和账户 B 的内容)

四、基于日志的恢复技术

1. 有如下两个事务 T1、T2

T1:  Read(B); B:= B+50; Write(B)	T2: Read(A); A:= A-100; Write(A); Read(B); B:= B+100; Write(B)
--	---

A 的初值为 1000, B 的初值为 600。

考虑如下调度:

r2(A); r1(B); w2(A); w1(B); r2(B); w2(B);

基于推迟更新策略, 向磁盘输出日志文件和更新数据信息的过程如下: (忽略了 flush log 的动作)

步骤	动作	日志
1	r2(A, t)	<T2 start>
2	t:=t-100	
3	r1(B, t)	<T1 start>
4	t:=t+50	
5	w2(A, t)	<T2,A,1000,900>
6	w1(B, t)	<T1,B,600,650>
7		<T1, commit>
8	Output(B)	
9	r2(B, t)	
10	t:=t+100	
11	w2(B, t)	<T2,B,650,750>
12		<T2, commit>
13	Output(B)	

请按照上表形式, 给出基于即时更新策略, 向磁盘输出日志文件和更新数据信息的过程。

2. 有一个日志内容如下(注: 更新日志记录的格式为< T<sub>i</sub>, X, V<sub>old</sub>, V<sub>new</sub>>):

- (a) <T1 start>
- (b) <T1,A,1000,950>
- (c) <T2, start>
- (d) <T2,A,950,850>
- (e) <T1,B,300,350>
- (f) <T1, commit>
- (g) <T2,B,350,450>
- (h) <T2, commit>

(1) 如果故障点发生在 (e) 处, 采用推迟更新技术, 哪些事务需要重做, 哪些事务要撤销, 数据库恢复后 A, B 值各是多少?

(2) 如果故障点发生在 (g) 处, 且此时已完成 output (B), 采用即时更新策略, 哪些事务需要重做, 哪些事务需要撤销, 数据库恢复后 A, B 值各是多少?

五、有一个带检查点的日志内容如下, 结束处<T3,D,-30,10>发生了故障, 请简述恢复算法的过程, 并给出 Undo-List 和 Redo-List, 以及恢复后数据库中 A, B, C, D 的值。(注: 更新日志记录的格式为< T<sub>i</sub>, X, V<sub>old</sub>, V<sub>new</sub>>)

```
<T0 start> //T0 事务开始
<T0,A,-10,10>
<T0, commit> //T0 事务提交
<T1, start> //T1 事务开始
<T1,B,-10,10>
<T2, start> //T2 事务开始
<T2,A,10,20>
<T2,A,20,-30>
<checkpoint{T1,T2}> //检查点标记
<T2, commit> //T2 事务提交
<T3, start> //T3 事务开始
<T3,C,10,-20>
<T3,D,-30,10>
```