

1. 考虑下面表中给出了 DBMS 对某四个事务操作五个数据项的调度按时间分布的表, 其中 $R(X)$ 和 $W(X)$ 分别代表对数据项 X 的“读”和“写”, T_i 表示事务 i , t_n 表示第 n 个时刻, 请阅读下表, 并回答以下几个问题:

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
T_1	R(A)			R(C)			R(B)		W(C)		
T_2						R(C)				W(D)	W(E)
T_3		R(E)			W(A)						
T_4			W(B)					R(D)			

1.1 这是否是一个串行的调度?

1.2 请仿照课上幻灯片中的画法, 画出上表中所示事务的优先图 (请在箭头上标明事务操作的数据项)

1.3 根据你画出的事务优先图, 该调度是否是个冲突可串行的调度?

2. 设 $r_i(X)$ 与 $w_i(X)$ 分别表示事物 T_i 读、写数据单元 X , 则一个并发调度可以抽象为读、写串。基于上述表示, 画出优先图, 并判断下面两个并发调度是否是冲突可串行化的, 为什么?

调度 S_1 :

$r_2(A); r_1(B); w_2(A); r_2(B); r_3(A); w_1(B); w_3(A); w_2(B)$

调度 S_2 :

$r_2(A); r_1(B); w_2(A); r_3(A); w_1(B); w_3(A); r_2(B); w_2(B)$

3. 假设我校淘乐果园水果店架设了一套数据库系统，同学在超市挑选好商品后，带商品到结算处结算付款，结算处有多名收银员使用多台机器进行结算。收银员负责扫同学购买水果的种类和数量，由系统后台结算程序计算出同学购买商品的总金额，修改商品表的水果库存量，并将销售信息写入销售表。请根据上述描述，回答以下问题。

假设有两位同学同时购买同一种类的水果，结算事务修改该水果的库存量（记为数据项）所产生的部分调度如下表所示：

<i>T1</i>	<i>T2</i>
$a \leftarrow Read(X)$	
	$a \leftarrow Read(X)$
$a = a - 1$	
$Write(X, a)$	
	$a = a - 2$
	$Write(X, a)$

3.1 如果数据项*X*购买前的初值为114514，则上述调度执行完成后，*X*的值是多少？这属于哪一类不一致性？

3.2 引入独占锁指令*lock()*和解锁指令*Unlock()*，对"问题 1"中存在并发调度进行重写，要求满足两段锁协议，且事务 *T1*、*T2* 首条指令的相对请求时间与"问题 1"中的相同。

4、已知某数据库采用即时更新方法（undo-redo 方法）记录 WAL 日志。设故障发生时 WAL 日志文件内容如下：

<T1, begin>
<T1, A, 114, 114514>
<T2, begin>
<T1, B, “hit”, “hits”>
<T1, commit>
<T3, begin>
<T3, B, “hits”, “hitsdb”>
<T2, A, 114514, 1919810>

当系统重启后，DBMS 基于该 WAL 日志文件进行故障恢复。回答下列问题：

- 4.1 该数据库系统的日志恢复策略为 Undo/Redo 型，那么其对应的缓冲区处理策略是什么（Steal? No Steal + Force? No Force）？该策略的每一项的具体内容都有什么？
- 4.2 当 DBMS 进行故障恢复时，需要对那些事务进行 undo？对那些事务进行 redo？给出具体理由
- 4.3 当故障恢复完成时，对象 A 和 B 的值分别是什么？描述故障恢复的具体过程。

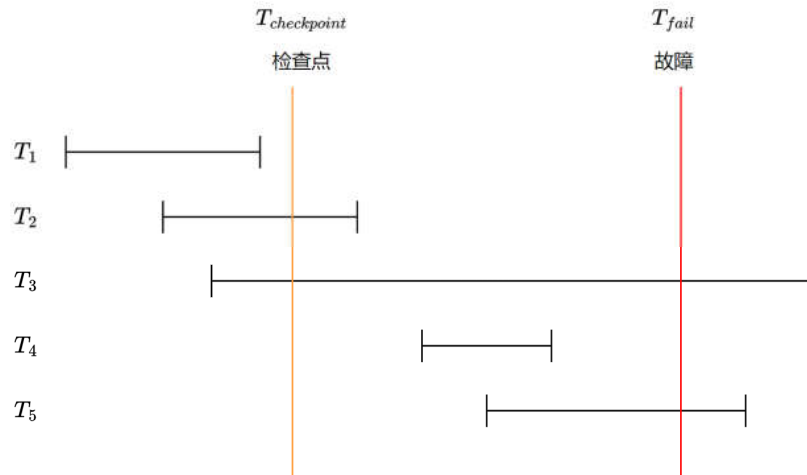
5. 当前有一个支持故障恢复技术的 DBMS，其采用了 STEAL 和 NO-FORCE 缓冲区策略，假设每当 DBMS 执行检查点时会将数据缓冲区里面所有的脏数据块写入相应的数据文件，保证数据库的一致性。

以下是数据库发生故障后的一份 WAL 日志，请阅读该日志并回答以下几个问题，

LSN	WAL Record
1	<T1 BEGIN>
2	<T1, X, 1, 2>
3	<T2 BEGIN>
4	<T2, Y, 1, 2>
5	<T1 COMMIT>
6	<T2, Y, 2, 3>
7	<T3 BEGIN>
8	<T3, Z, 1, 2>
9	<T2, X, 2, 3>
10	<CHECKPOINT>
11	<T2, Y, 3, 4>
12	<T3, Z, 2, 3>
13	<T3 COMMIT>
14	<T2, Z, 3, 4>

- 其中< T_i , X, A, B >分别代表着事务 T_i ，数据项X，更新前数据项的值A，更新后数据项的值B
- 5.1 请问 DBMS 需要进行恢复数据的时候，数据库的磁盘文件上数据项 X Y Z 的值分别是多少？
- 5.2 在对该 WAL 日志进行故障恢复的时候，事务 T_1, T_2, T_3 都应如何处理？请详述该过程与原因
- 5.3 假设恢复作业完成后，DBMS 会将所有的数据写入磁盘，DBMS 从 WAL 恢复数据库状态后数据库磁盘中数据项 X、Y、Z 的值分布是多少？

6. 使用检查点的数据库恢复系统将根据事务的状态和检查点的关系采取相对应的恢复策略，现在有事务 T_1-T_5 其执行过程如下图所示（线段左端和右端分别表示事务开始与提交），其执行过程中数据库系统发生如图所示的故障，请回答下列问题



- 6.1 请问在故障恢复时事务 T_1-T_5 那些需要撤销，那些需要重做，那些不需要操作？
- 6.2 事务 T_6-T_8 的日志文件如下图所示,< T_i , begin>表示事务 T_i 开始执行，< T_i , commit>表示事务 T_i 提交，< T_i , D, V_1 , V_2 >表示事务 T_i 将数据项 D 的值由 V_1 修改为 V_2 ，<crash>表示数据库发生故障

```

<T6, begin>
<T6, X, 100, 1>
<T7, begin>
<T7, X, 1,3>
<T8, begin>
<T7, Y, 50, 6>
<T8, Y, 6, 8>
<T8, Z, 10, 9>
<checkpoint>
<T6,commit>
<T8,Z,9,10>
<crash>

```

数据库系统发生故障时，请给出恢复子系统时需要 undo 的事务列表和需要 redo 的事务列表

- 6.3 请简述事务 T_6-T_8 在系统故障后，基于检查点的故障恢复过程

1. 考虑下面表中给出了 DBMS 对某四个事务操作五个数据项的调度按时间分布的表,

其中 $R(X)$ 和 $W(X)$ 分别代表对数据项 X 的“读”和“写”, T_i 表示事务 i , t_n 表示第 n 个时刻,

请阅读下表,并回答以下几个问题:

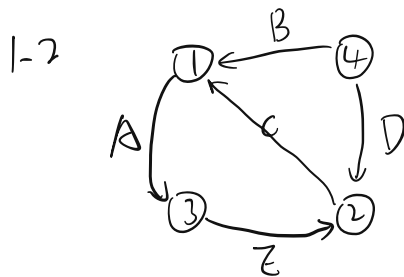
time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
T_1	R(A)			R(C)			R(B)		W(C)		
T_2						R(C)				W(D)	W(E)
T_3		R(E)			W(A)						
T_4			W(B)					R(D)			

1.1 这是否是一个串行的调度?

1.2 请仿照课上幻灯片中的画法,画出上表中所示事务的优先图(请在箭头上标明事务操作的数据项)

1.3 根据你画出的事务优先图,该调度是否是个冲突可串行的调度?

1-1 不是,是并发调度



1.3 不是,有环路

2. 设 $r_i(X)$ 与 $w_i(X)$ 分别表示事物 T_i 读、写数据单元 X , 则一个并发调度可以抽象为读、

写串。基于上述表示, 画出优先图, 并判断下面两个并发调度是否是冲突可串行化的, 为

什么?

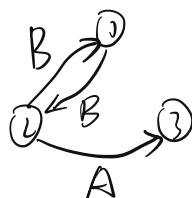
调度 S_1 :

$r_2(A); r_1(B); w_2(A); r_2(B); r_3(A); w_1(B); w_3(A); w_2(B)$

调度 S_2 :

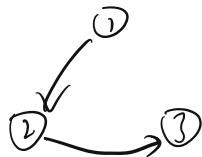
$r_2(A); r_1(B); w_2(A); r_3(A); w_1(B); w_3(A); r_2(B); w_2(B)$

S_1 优先图:



不是冲突可串行化的

S_2 优先图:



是冲突可串行化的

3. 假设我校淘乐果园水果店架设了一套数据库系统，同学在超市挑选好商品后，带商品到结算处结算付款，结算处有多名收银员使用多台机器进行结算。收银员负责扫描同学购买水果的种类和数量，由系统后台结算程序计算出同学购买商品的总金额，修改商品表的水果库存量，并将销售信息写入销售表。请根据上述描述，回答以下问题。

假设有两位同学同时购买同一种类的水果，结算事务修改该水果的库存量（记为数据项）所产生的部分调度如下表所示：

T1	T2
$a \leftarrow Read(X)$	
	$a \leftarrow Read(X)$
$a = a - 1$	
$Write(X, a)$	
	$a = a - 2$
	$Write(X, a)$

3.1 如果数据项X购买前的初值为114514，则上述调度执行完成后，X的值是多少？这属于哪一类不一致性？

3.2 引入独占锁指令 $lock()$ 和解锁指令 $unlock()$ ，对“问题 1”中存在并发调度进行重写，要求满足两段锁协议，且事务 T1、T2 首条指令的相对请求时间与“问题 1”中的相同。

3.1 X 的值为 114512，属于丢失修改

3.2

T1	T2
$lock(X)$	
$a \leftarrow Read(X)$	
$a = a - 1$	
$Write(X, a)$	
$unlock(X)$	

	$lock(X)$
	$a \leftarrow Read(X)$
	$a = a - 2$
	$Write(X, a)$
	$unlock(X)$

4. 已知某数据库采用即时更新方法 (undo-redo 方法) 记录 WAL 日志。设故障发生时 WAL 日志文件内容如下:

```
<T1, begin>
<T1, A, 114, 114514>
<T2, begin>
<T1, B, "hit", "hits">
<T1, commit>
<T3, begin>
<T3, B, "hits", "hitsdb">
<T2, A, 114514, 1919810>
```

当系统重启后, DBMS 基于该 WAL 日志文件进行故障恢复。回答下列问题:

4.1 该数据库系统的日志恢复策略为 Undo/Redo 型, 那么其对应的缓冲区处理策略是什么 (Steal? No Steal + Force? No Force)? 该策略的每一项的具体内容都有什么?

4.2 当 DBMS 进行故障恢复时, 需要对那些事务进行 undo? 对那些事务进行 redo? 给出具体理由

4.3 当故障恢复完成时, 对象 A 和 B 的值分别是什么? 描述故障恢复的具体过程。

4.1 steal + No-Force

steal: 允许将来提交事务所做的修改写回磁盘

No Force: 已提交事务不强制写回磁盘

4.2 对 T1 进行 redo, 因为 T1 已经提交

对 T2, T3 进行 undo, 因为 T2, T3 都还没提交

4.3 A 为 114514 B 变为 hits

redo 阶段:

log

redo action

```
<T1, begin>
<T1, A, 114, 114514>
<T2, begin>
<T1, B, "hit", "hits">
<T1, commit>
<T3, begin>
<T3, B, "hits", "hitsdb">
<T2, A, 114514, 1919810>
```

```
A ← 114
T2 无需 redo
B ← hits
T1 redo 完毕
T3 无需 redo
```

undo 阶段

log

undo action

```
<T1, begin>
<T1, A, 114, 114514>
<T2, begin>
<T1, B, "hit", "hits">
<T1, commit>
<T3, begin>
<T3, B, "hits", "hitsdb">
<T2, A, 114514, 1919810>
```

T2 undo 完毕, 写 T2, ABORT

```
T1 无需 undo
T3 undo 完毕, 写 T3, ABORT
B ← hits
A ← 114514
```


5. 当前有一个支持故障恢复技术的 DBMS，其采用了 STEAL 和 NO-FORCE 缓冲区策略，假设每当 DBMS 执行检查点时会将数据缓冲区里面所有的脏数据块写入相应的数据文件，保证数据库的一致性。

以下是数据库发生故障后的一份 WAL 日志，请阅读该日志并回答以下几个问题，

LSN	WAL Record
1	<T1 BEGIN>
2	<T1, X, 1, 2>
3	<T2 BEGIN>
4	<T2, Y, 1, 2>
5	<T1 COMMIT>
6	<T2, Y, 2, 3>
7	<T3 BEGIN>
8	<T3, Z, 1, 2>
9	<T2, X, 2, 3>
10	<CHECKPOINT>
11	<T2, Y, 3, 4>
12	<T3, Z, 2, 3>
13	<T3 COMMIT>
14	<T2, Z, 3, 4>

其中 $\langle T_i, X, A, B \rangle$ 分别代表着事务 T_i ，数据项 X ，更新前数据项的值 A ，更新后数据项的值 B

5.1 请问 DBMS 需要进行恢复数据的时候，数据库的磁盘文件上数据项 X Y Z 的值分别是多少？

5.2 在对该 WAL 日志进行故障恢复的时候，事务 T_1, T_2, T_3 都应如何处理？请详述该过程与原因

5.3 假设恢复作业完成后，DBMS 会将所有的数据写入磁盘，DBMS 从 WAL 恢复数据库状态后数据库磁盘中数据项 X、Y、Z 的值分布是多少？

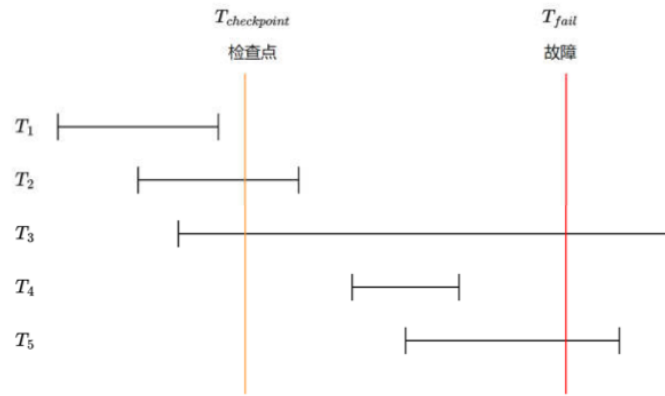
5.1 $X=3$ $Y=3$ $Z=2$

5.2 T_1 无需处理， T_2 进行 undo 操作， T_3 进行 redo 操作
 因为 T_1 的 commit 在 checkpoint 之前，
 而 T_2 没有 commit，需要 undo。
 T_3 有 commit 且在 checkpoint 之后，需要 redo

redo 阶段	log	redo action	undo 阶段	log	undo action
10	<CHECKPOINT>	开始 redo	3	<T2 BEGIN>	T_2 undo 完成，从 T_2 开始
11	<T2, Y, 3, 4>	T_2 无需 redo	4	<T2, Y, 1, 2>	$Y \leftarrow 1$
12	<T3, Z, 2, 3>	$Z \leftarrow 3$	5	<T1 COMMIT>	T_1 无需 undo
13	<T3 COMMIT>	T_3 redo 完成	6	<T2, Y, 2, 3>	$Y \leftarrow 2$
14	<T2, Z, 3, 4>		7	<T3 BEGIN>	
			8	<T3, Z, 1, 2>	
			9	<T2, X, 2, 3>	$X \leftarrow 2$
			10	<CHECKPOINT>	
			11	<T2, Y, 3, 4>	$Y \leftarrow 3$
			12	<T3, Z, 2, 3>	
			13	<T3 COMMIT>	T_3 无需 undo
			14	<T2, Z, 3, 4>	$Z \leftarrow 3$

5.3 $X=2$ ， $Y=1$ ， $Z=3$

6. 使用检查点的数据库恢复系统将根据事务的状态和检查点的关系采取相对应的恢复策略，现在有事务 T_1-T_5 其执行过程如下图所示（线段左端和右端分别表示事务开始与提交），其执行过程中数据库系统发生如图所示的故障，请回答下列问题



6.1 请问在故障恢复时事务 T_1-T_5 那些需要撤销，那些需要重做，那些不需要操作？

6.2 事务 T_6-T_8 的日志文件如下图所示,< T_i , begin>表示事务 T_i 开始执行, < T_i , commit>表示事务 T_i 提交, < T_i , D, V_1 , V_2 >表示事务 T_i 将数据项 D 的值由 V_1 修改为 V_2 , <crash>表示数据库发生故障

```
<T6, begin>
<T6, X, 100, 1>
<T7, begin>
<T7, X, 1,3>
<T8, begin>
<T7, Y, 50, 6>
<T8, Y, 6, 8>
<T8, Z, 10, 9>
<checkpoint>
<T6,commit>
<T8,Z,9,10>
<crash>
```

数据库系统发生故障时，请给出恢复子系统时需要 undo 的事务列表和需要 redo 的事务列表

6.3 请简述事务 T_6-T_8 在系统故障后，基于检查点的故障恢复过程

6.1 撤销: T_3, T_5
重做: T_2, T_4
不操作: T_1

6.2 undo: T_7, T_8
redo: T_6

6.3 redo阶段
log
<checkpoint>
<T6,commit>
<T8,Z,9,10>
<crash>
redo开始
 T_6 redo结束
 T_8 无需 redo

undo阶段
log

```
<T7, begin>
<T7, X, 1,3>
<T8, begin>
<T7, Y, 50, 6>
<T8, Y, 6, 8>
<T8, Z, 10, 9>
<checkpoint>
<T6,commit>
<T8,Z,9,10>
<crash>
```

undo action

T_7 undo完成, 写<T7.ABORT>
 $X \leftarrow 1$
 T_8 undo完成, 写<T8.ABORT>
 $Y \leftarrow 50$
 $Y \leftarrow 6$
 $Z \leftarrow 10$
 T_6 无需 undo
 $Z \leftarrow 9$