

work 9

3. 登记日志文件时为什么必须先写日志文件，后写数据库

如果先写了数据库修改，之间出现了故障，而日志文件中没有记录这个修改，则以后无法恢复这个修改。而对于先写日志文件再写数据库，在恢复时只不过是多执行了一次恢复操作，不会对数据库有任何正确性的影响

4. 考虑下图所示的日志记录

(1)

T1，T3重做；T2，T4回滚

(2)

T1重做；T2，T3回滚

(3)

T1重做；T2，T3回滚

(4)

T1重做；T2回滚

5. 考虑题 4 所示的日志记录，假设开始时A、B、C的值都是 0

\	A	B	C
1	8	7	11
2	10	0	11
3	10	0	11
4	10	0	11
5	10	0	11
6	0	0	0

6. 针对不同的故障，试给出恢复的策略和方法

- 事务故障恢复：
 - 反向扫描文件日志，查找该事务的更新操作

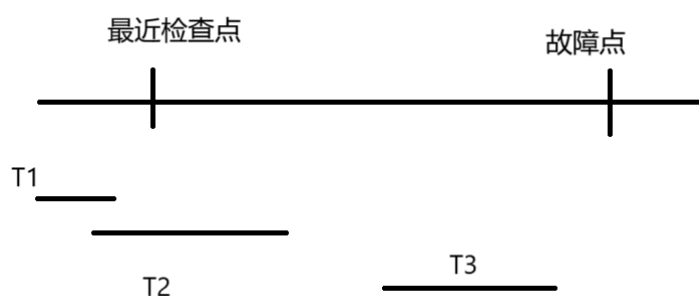
- 对该事务的更新操作执行逆操作
- 直到读到此事务的开始标记，该事务故障的恢复完成
- **系统故障恢复：**
 - 正向扫描日志文件，找出在故障发生前已经提交的事务队列和未完成的事务队列
 - 对于未完成的事务队列中各个事务进行UNDO操作
 - 对于已经提交的事务队列进行REDO操作
- **介质故障恢复：**
 - 装入最新的数据库后备副本，使数据库恢复到最近一次转储时的一致性状态
 - 装入转储结束时刻的日志文件副本
 - 启动系统恢复命令，由DBMS完成恢复功能，即重做已完成的事务

7. 什么是检查点记录？检查点记录包括哪些内容

- 检查点记录：一类新的日志记录
- 包括建立检查点时刻所有正在执行的事务清单，以及这些事务的最近一个日志记录的地址

8. 具有检查点的恢复技术有什么优点？试举一个具体例子加以说明

- 优点：利用日志技术进行数据库恢复时，恢复子系统必须搜索整个日志，这将耗费大量的时间；对于需要REDO处理的事务实际上已经将它们的更新操作结果写道数据库中了，恢复子系统又重新执行了这些操作，浪费了大量时间
- 对于下图的情况，T1在检查点前已经将结果写到了数据库中，不需要再进行REDO操作，可以节省大量时间



9. 试述使用检查点方法进行恢复的步骤

- 从日志末尾向前扫描，找到最后一个有效的检查点记录。
- 从检查点记录中获取活跃事务列表，记为 `UNDO-LIST`（需撤销的事务候选）。
- 从检查点位置开始，扫描日志到末尾：
 - 遇到新事务：将其加入 `UNDO-LIST`。

- 遇到事务提交：将其从 `UNDO-LIST` 移至 `REDO-LIST`（需重做的事务）。
- 遇到事务中止：直接移出 `UNDO-LIST`。
- 从检查点之前最早的未持久化操作开始，对 `REDO-LIST` 中的所有已提交事务，重新执行其日志中的操作（即使数据已写入磁盘，幂等性确保正确性）。
- 从日志末尾逆向扫描，对 `UNDO-LIST` 中的每个未提交事务，执行补偿操作（如写回旧值），直到事务开始记录。