

Lab 04 ---- Lock Manager Simulator

本实验要求实现一个支持 S 锁、X 锁的锁管理器模拟程序。完整的锁管理器实现须依赖事务、事务管理以及并发控制的实现，因而过于复杂。在本实验中，我们模拟实现一个简单的锁管理器。该锁管理器从 stdin 获取事务的锁请求（实际中应来自事务管理子系统以及并发控制子系统），然后将事务的锁信息输出到 stdout。

输入命令的格式为：<request type> <transaction ID> <object>

其中<request type>是锁管理器接收的处理请求。本实验要求模拟 S 锁和 X 锁（其它锁类型可以自行扩充），因此输入命令至少包括 Start、End、XLock、SLock、Unlock 等五种命令，分别表示开始事务、结束事务(abort or commit)、请求 X 锁、请求 S 锁、释放锁。<transaction ID>是事务标识，我们规定其为 0 到 255 之间的一个整数；<object>是请求加锁的数据库对象，以单个的大写字母来表示。

以下是几种可能的输入命令例子：

1. *Start transID*: 开始一个事务 transID.
2. *End transID*: 结束事务 transID。此时需要释放该事务所持有的所有锁。如果有别的事务在等待 transID 上的锁，则需要根据给定的策略将锁授予等待的事务。
3. *SLock transID object*: 事务 transID 请求 object 上的一个 S 锁。该请求可能的输出有两种：如果锁请求被批准，则更新锁表信息同时输出“Lock granted”；如果锁请求不能被批准，则将该事务放入等待队列并输出“Waiting for lock (X-lock held by: <trans_ID>)”。
4. *XLock transID object*: 事务 transID 请求 object 上的一个 X 锁。该请求同样有两种可能的执行结果，即批准或者等待。如果事务之前已经持有了 object 上的 S 锁，则该操作将 S 锁升级为 X 锁。
5. *Unlock transID object*: 事务 transID 释放对象 object 上的锁。释放锁之后，如果有别的事务在等待该锁，则需要将锁按给定的策略授予等待的事务。你可以自定义等待事务获得锁的规则，例如 FIFO、LIFO 等，但需要在实验报告中明确解释你的规则。

每种输入命令有相应的输出信息。下表给出了每一种输入命令可能的输出信息：

输入命令	可能的输出
Start	Transaction started
SLock	Lock granted Waiting for lock (X-lock held by: <trans_ID>)
XLock	Lock granted Upgrade to X-lock granted Waiting for lock (S-lock held by: <transID> . . . <transID>) Waiting for lock (X-lock held by: <trans_ID>)
End	Transaction ended Lock released Lock granted to <transID> . . . <transID>
Unlock	Lock released Lock granted to <transID> . . . <transID>

下面是输入输出信息的一个示例：

输入	输出
Start 100	Start 100 : Transaction 100 started
Start 200	Start 200 : Transaction 200 started
SLock 100 A	SLock 100 A: Lock granted
XLock 200 A	XLock 200 A: Waiting for lock (S-lock held by: 100)
Unlock 100 A	Unlock 100 A: Lock released X-Lock granted to 200
XLock 100 B	XLock 100 B: Lock granted
XLock 200 B	XLock 200 B: Waiting for lock (X-lock held by: 100)
XLock 100 A	XLock 100 A: Waiting for lock (X-lock held by: 200)
End 100	End 100: Transaction 100 ended Release X-lock on B X-Lock on B granted to 200
Unlock 200 A	Unlock 200 A: Lock released
End 200	End 200: Transaction 200 ended

说明：

1. 编程语言首选使用 C/C++；
2. 输入输出可以直接使用标准输入输出，但也允许使用窗体。
3. 除了要求的输入命令外，其它命令可以自行增加，例如打印整个锁表、输出特定数据对象上当前持有锁的事务列表、输出特定对象上等待锁的事务列表等等。

提交要求：

1. 完成锁管理器模拟程序设计与实现，并能够演示；
2. 提交设计报告，格式和要求参考[报告模板](#)。