**2019年秋季学期/并行和分布式计算**

**分布式计算的作业**

**说明：作业分成两类，每个人独立完成巩固类作业(占10分)，4-5人形成一个小组，完成上机类作业(占10分，同组人同分)。**

**巩固类作业/10分**

1. 下图中，直线上小黑点给出了时钟计数，请分别用Lamport 逻辑时钟和向量时钟给图上的事件设置时间戳，并给出一致割集和非一致割集的例子。



1. 考虑在异步分布式系统中使用的两个通信服务。在服务A中，消息可能丢失、被复制或延迟，校验和仅应用到消息头。在服务B中，消息可能丢失、延迟或发送地太快以致接收方无法处理它，但到达目的地的消息其内容一定正确。描述每个服务会有的故障类型。根据对有效性和完整性的影响将故障分类。服务B能被描述成一个可靠的通信服务吗？
2. 请证明Lamport的互斥算法满足ME1、ME2和ME3。其中，ME1指在临界区一次最多有一个进程可以执行，ME2指进入和离开临界区的请求最终成功执行，ME3（→顺序）指如果一个进入临界区的请求发生在先，那么进入临界区任按此顺序。
3. 请证明Ricart-Agrawala的互斥算法满足ME2和ME3。
4. 在Ricart-Agrawala的互斥算法中，原始假定系统的进程是不出故障的。请修改算法增加处理一个进程崩溃的情况。
5. 改进基于环的互斥算法使得它能检测权标的丢失并重新生成权标。
6. 基于环的选举算法是建立在单向环的假设之上的，为了获得更快的选举速度，现采用双向环结构，即每个节点可以同时向顺时针和逆时针两个方向发送选举消息，请列出新算法的高层描述，并用一个四节点的双向环来说明你的方法。
7. 节点之间按照生成树方式连接，仅有边相连的节点能通信，请基于此网络拓扑，设计一个选举算法，给出其伪码。当仅有一个进程发起选举，你的选举算法所需的消息量是多少？
8. 一个服务器管理对象a1, a2, ... an ，它为客户提供下面两种操作：read (i)返回对象ai的值。write(i, Value)将对象ai设置为值Value。

事务T和U定义如下：

*T*: *x* = *read*(*j*); *y* = *read* (*i*); *write*(*j*, 44); *write*(*i*, 33)

*U*: *x* = *read*(*k*); *write*(*i*, 55); *y* = *read* (*j*); *write*(*k*, 66)

请给出事务T和U的3个串行化等价的交错执行。

1. 考虑将乐观并发控制应用于下列事务T和U的情况：

*T*: *x* = *read*(*i*); *write*(*j*, 44);

*U*: *write*(*i*, 55); *write*(*j*, 66);

如果事务T和U同时处于活动状态，试描述以下几种情况的结果如何：

1. 服务器首先处理T的提交请求，使用向后验证方式。
2. 服务器首先处理U的提交请求，使用向后验证方式。
3. 服务器首先处理T的提交请求，使用向前验证方式。
4. 服务器首先处理U的提交请求，使用向前验证方式。

对于上面的每种情况，描述事务T和U的操作顺序，注意写操作在验证通过之后才真正起作用。

**上机类作业/10分**

1. (5分)实现全局状态的快照算法，并监控下列程序：两个进程P和Q用两个通道连成一个环，它们不断地轮转消息m。在任何一个时刻，系统中仅有一份m的拷贝。每个进程的状态是指由它接收到m的次数。P首先发送m。在某一点，P得到消息且它的状态是101。在发送m之后，P启动快照算法，要求记录由快照算法报告的全局状态。
2. (5分) 在Linux平台，利用开源RPC代码，实现锁服务(Lock Service)，锁服务包括两个模块：锁客户，锁服务器，两者通过RPC通信。需要实现两个功能：(1)客户发acquire请求，从锁服务器请求一个特定的锁，用release释放锁，锁服务器一次将锁授予一个客户。(2) 扩充RPC库，实现at-most-once执行语义，即消除重复的RPC请求。