MIT6.824lab2 lab log

2019 2 25：

test2A一直过不去，就连最基本的选举都会出问题，选不出leader，但是第一次选举又是可以的。我认为错误出在了Request和send Request。通过输出我知道了在leader死亡之后，follower变成candidate后会锁死，于是我在所有的加锁节点都打上一个log，最后发现在appendEntries中加锁后未解锁，在函数中自习排查发现，是中间一个channel忘记初始化大小，那边没有接受者，卡死了

2019 2 26：

test2B前几个就出现了小问题，找到原因是在appendEntries必须找到与leader前一个相同的log，leader才好发送新的，改变follower的log使之与leader相同。

但是我只是机械的匹配与leader最新的log相同的那一个，如果follower中没有那一个log就会永远找不到，陷入一个死循环，最后修改能在windows上通过所有的测试

2019 2 27凌晨:

训完练回来，继续完成lab2B最后部分。由于在linux环境和windows环境下运行结果不同，且windows环境下运行成功而linux环境下运行失败，所以我认为应该是某个地方出了点问题导致。在大的地方打了许多log之后，我发现并没有找到错误地址，只是知道最后commit出错，应该是日志提交的时候复制错误。

于是我在sendAppendEntries和AppendEntries两个函数上面都打了许多细节上的log，最后终于知道，是最开始的disconnect的leader再次发送AppendEntries给后面比较新的raft节点的时候，raft节点并没有拒绝，而是接受并且把自己的committed log

改成了old leader的log，这是违反raft协议的，按照raft协议和我的代码，节点应该拒绝。所以我再细致的排查，发现是在传送的途中，leader的上一个appendEntries被拒绝时，leader将自己的state改为follower并且将自己的current Term改为最新的。导致后一个接受。windows平台下没有出错的原因也应该是巧合，最后在发送消息的位置再次检测了leader节点的state，如果不是leader，则退出函数。

修改完成之后，终于通过test2B

3月4日：

在做lab3a的时候发现follower一直不提交，最后知道我是在心跳的时候设置提交，但是如果一直有client在start的话就会失败，一直不提交，后面我把提交的位置换了就好了