**华东师范大学数据科学与工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**： 区块链 | **年级**：19 | **上机实践成绩**： |
| **指导教师**：张召 | **姓名**：赵煜硕 |  |
| **上机实践名称**：第十二次实验 | **学号**：10195501415 | **上机实践日期**： |
| **上机实践编号**： | **组号**： | **上机实践时间**： |

1. **实验目的**

PBFT共识算法简单实现

1. **实验任务**

了解PBFT算法的原理及共识流程，并能对给出的不完整代码进行补充。

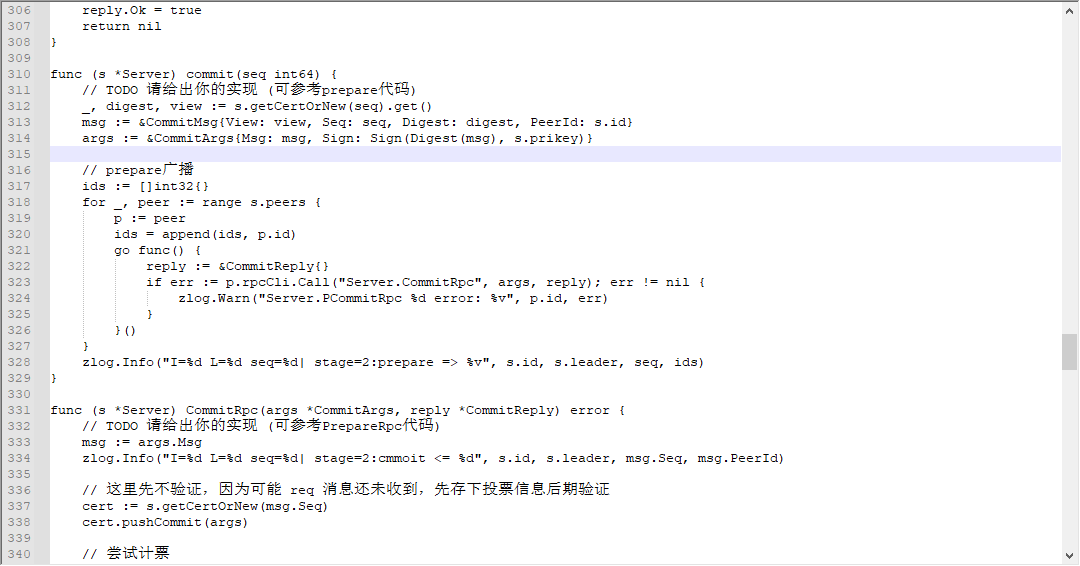
1. **使用环境**

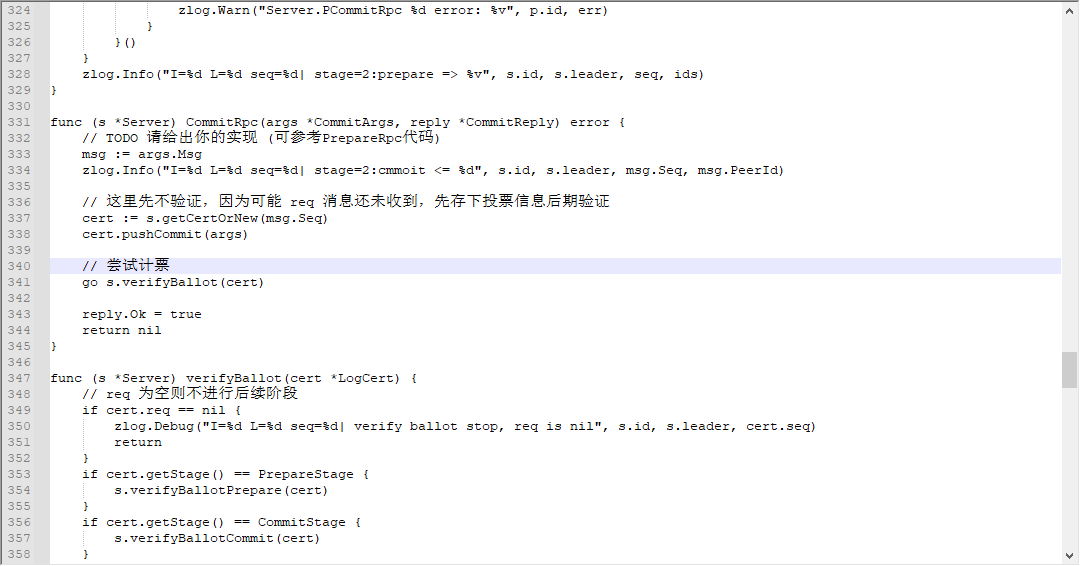
**Go1.7**

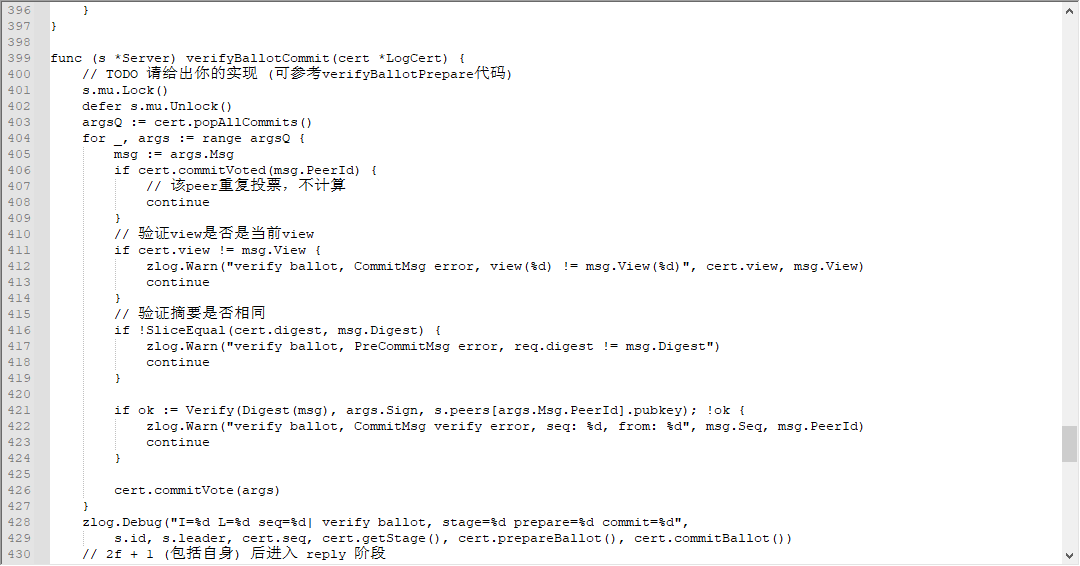
1. **实验过程**

先切换到go1.7，之前使用的go1.1在执行go build naive.go时会显示版本不足

然后补全代码







执行go build naive.go

执行./naive -role master -maddr :8000 启动master



再另起四个窗口

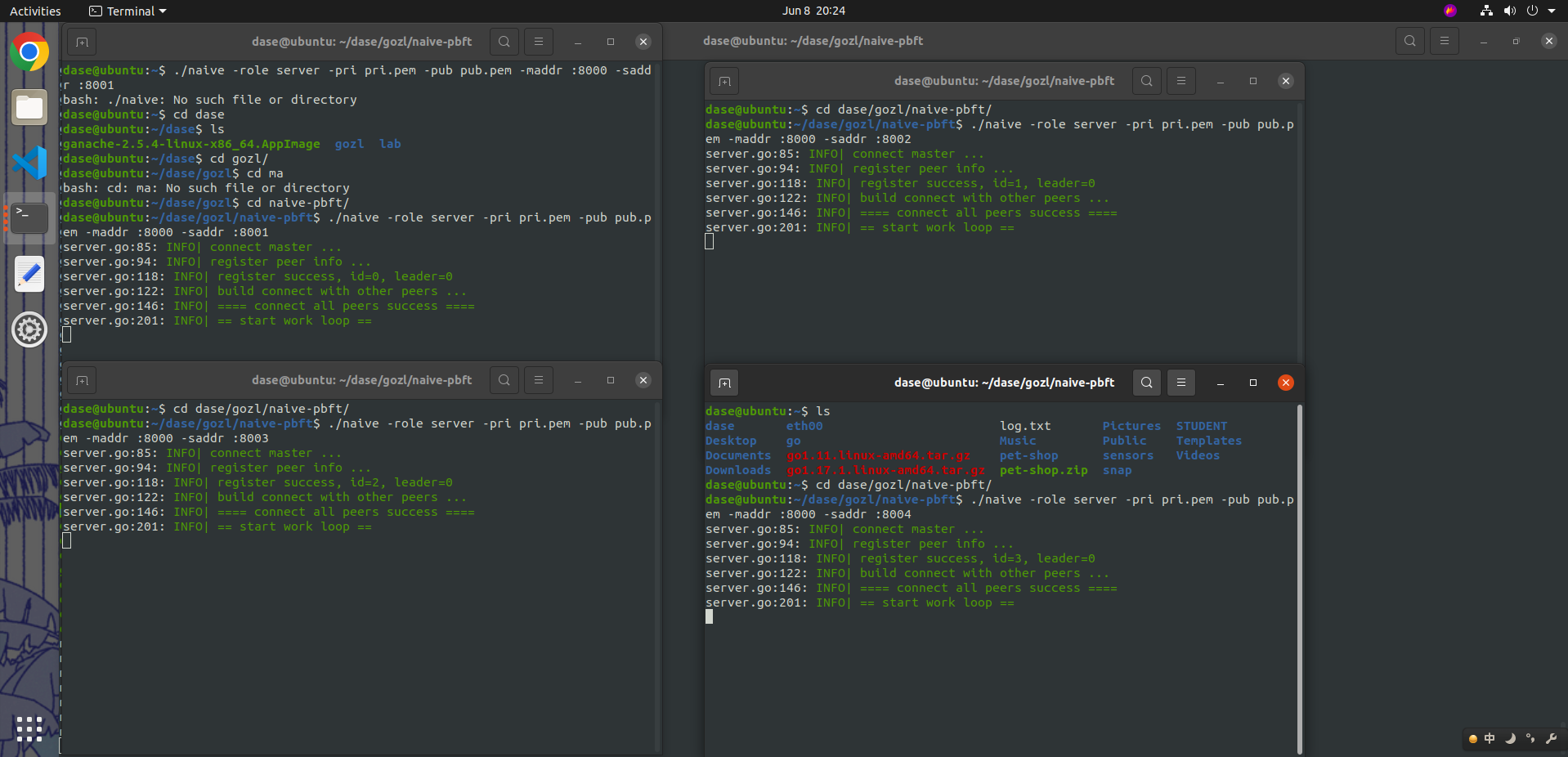
分别执行

./naive -role server -pri pri.pem -pub pub.pem -maddr :8000 -saddr :8001

./naive -role server -pri pri.pem -pub pub.pem -maddr :8000 -saddr :8002

./naive -role server -pri pri.pem -pub pub.pem -maddr :8000 -saddr :8003

./naive -role server -pri pri.pem -pub pub.pem -maddr :8000 -saddr :8004

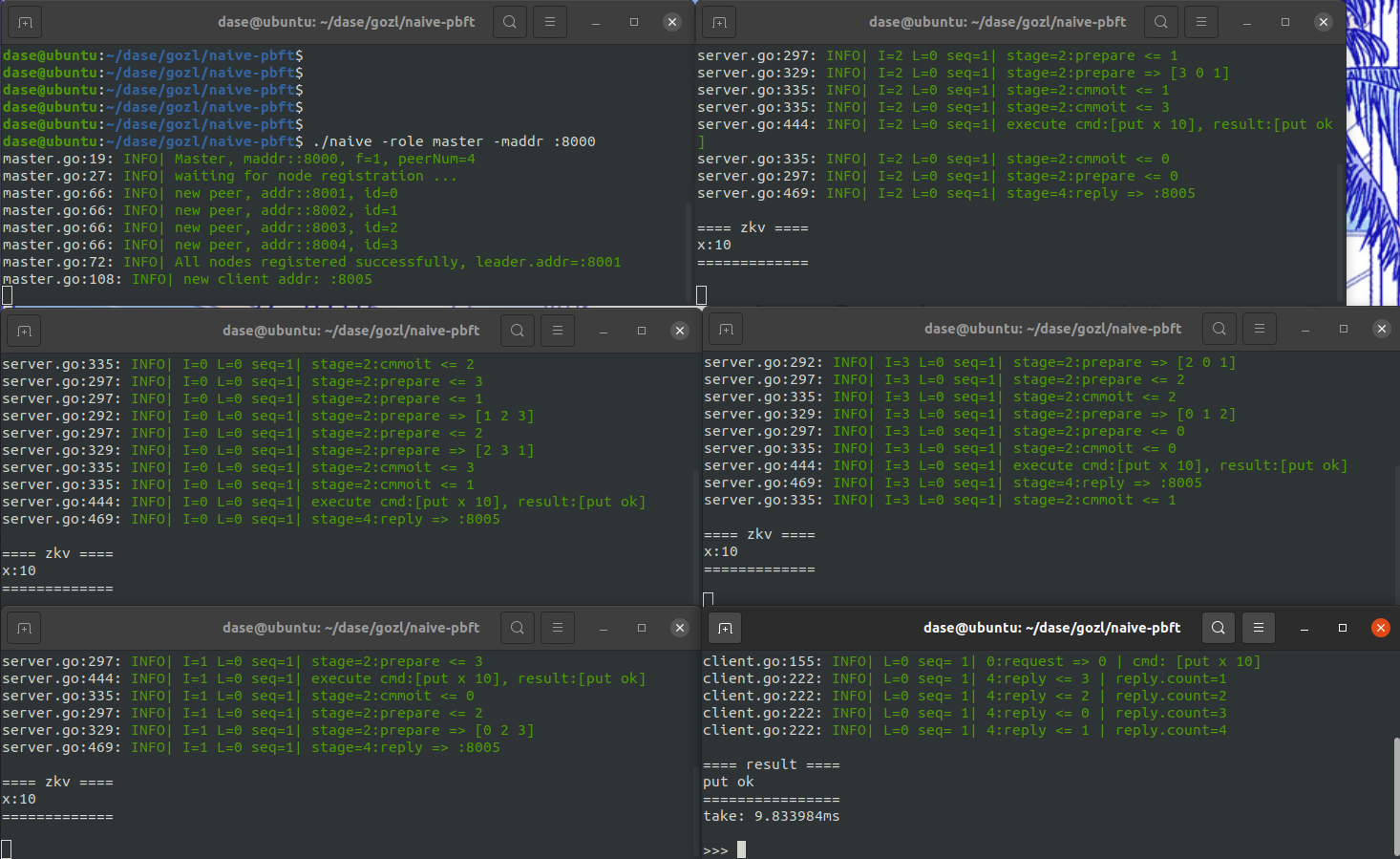


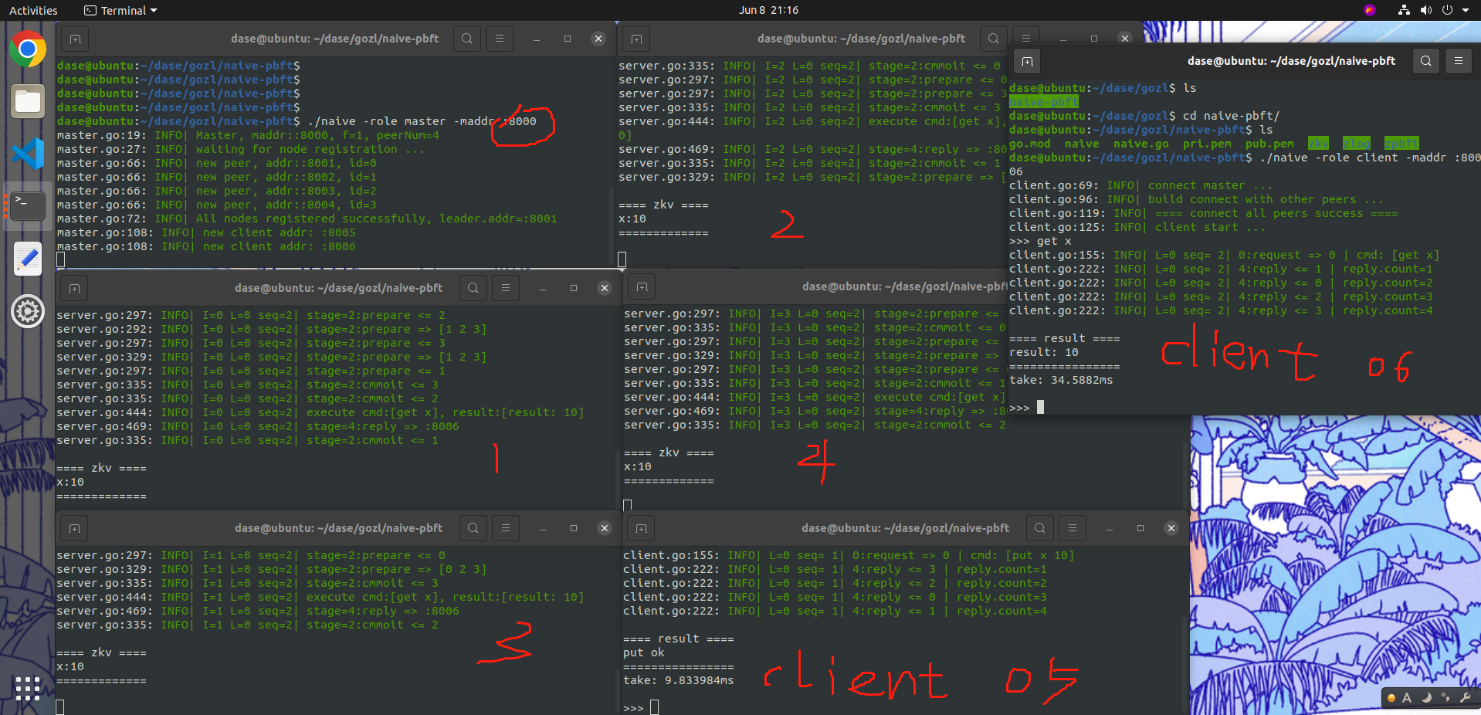
再另起一个窗口执行./naive -role client -maddr :8000 -caddr :8005启动client



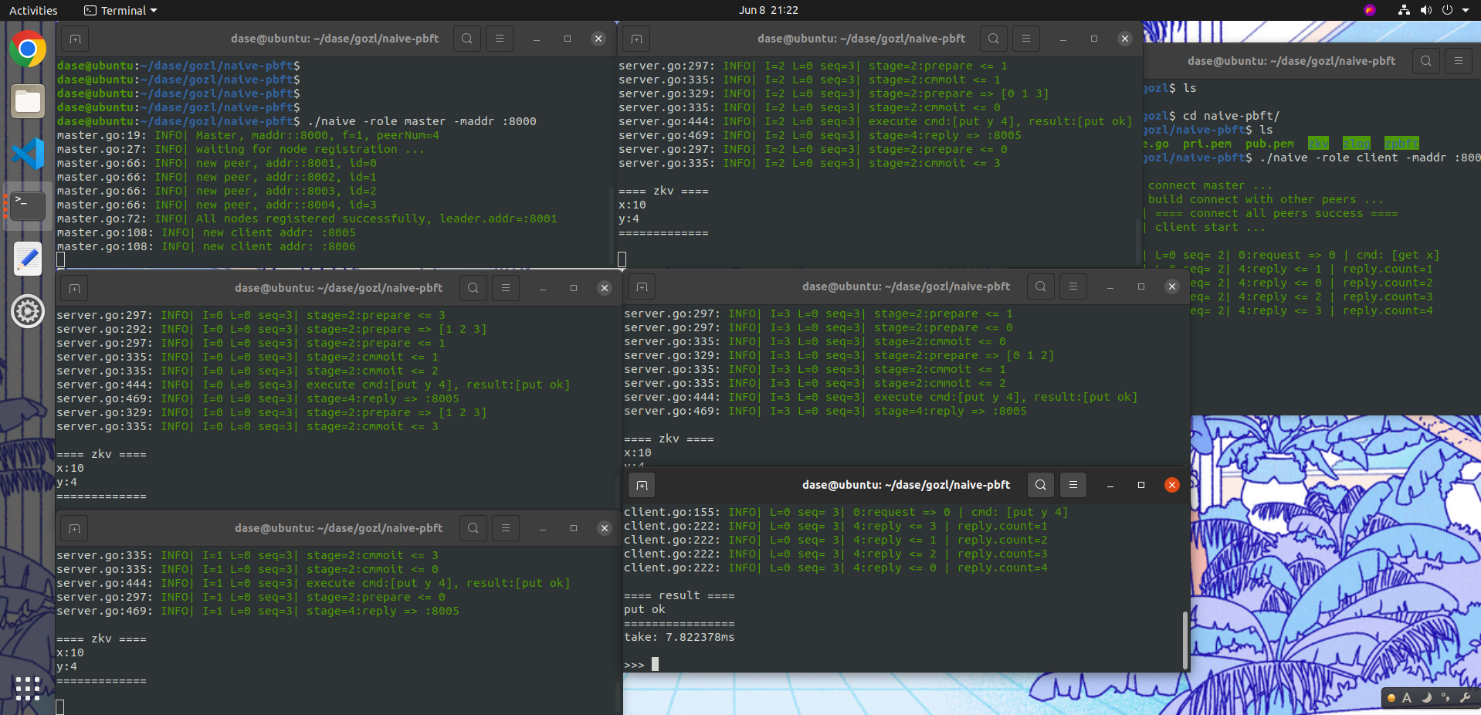
在8005中执行put x 10

同时另起client8006 执行get x查看

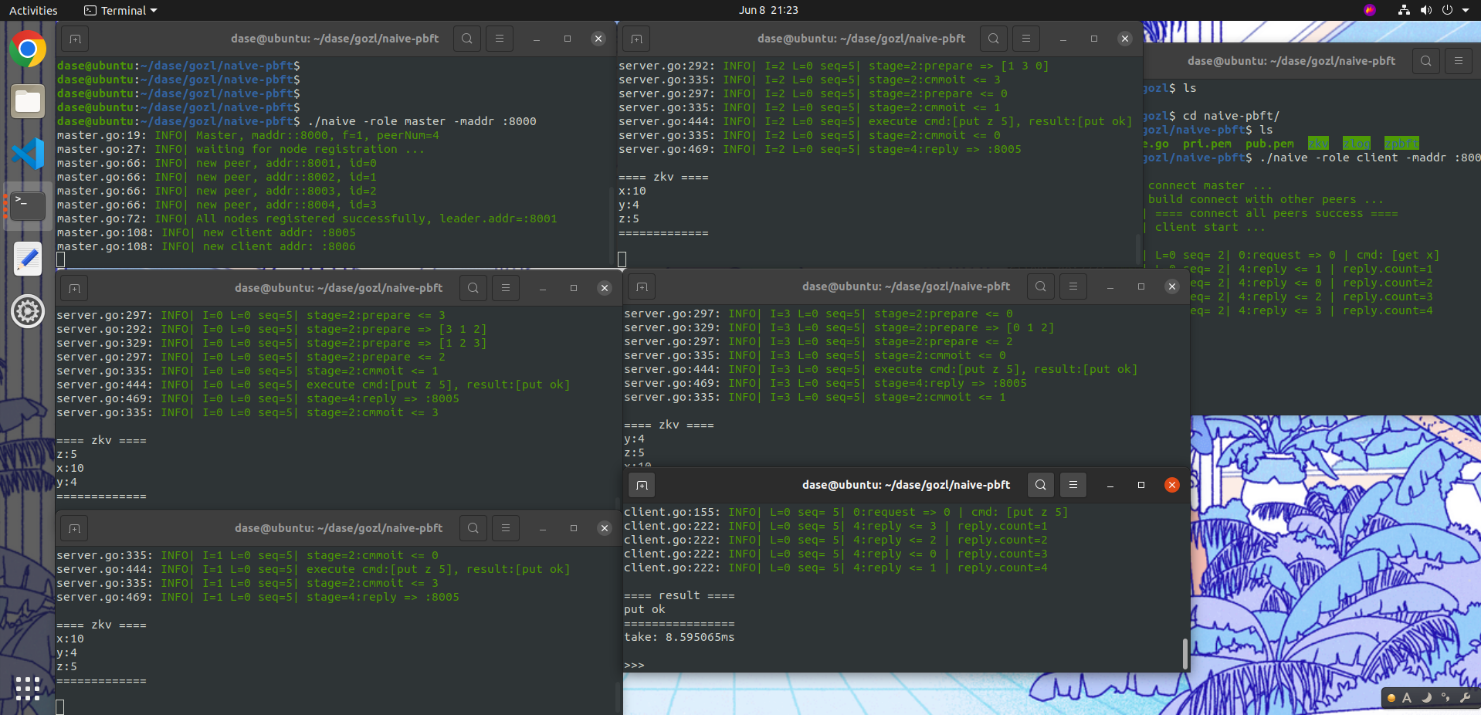




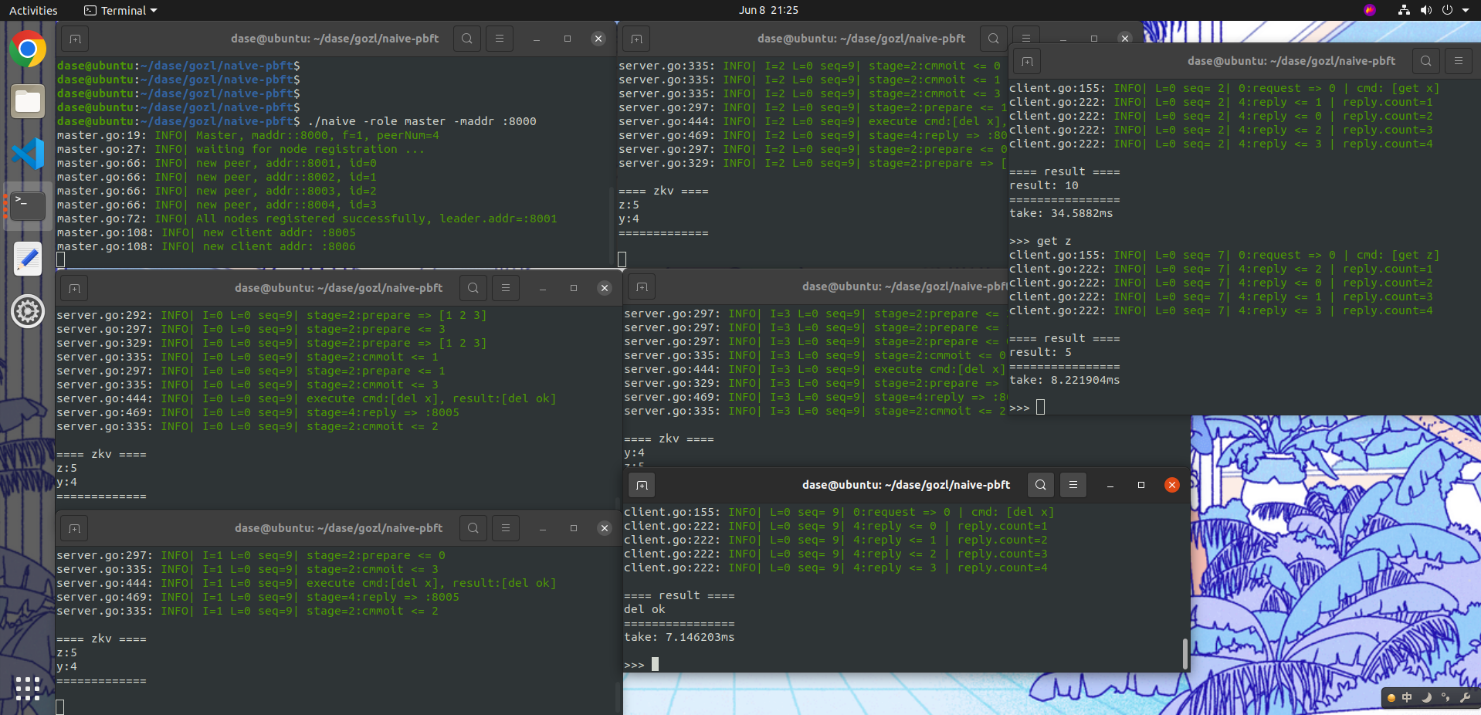
在05中执行 put y 4



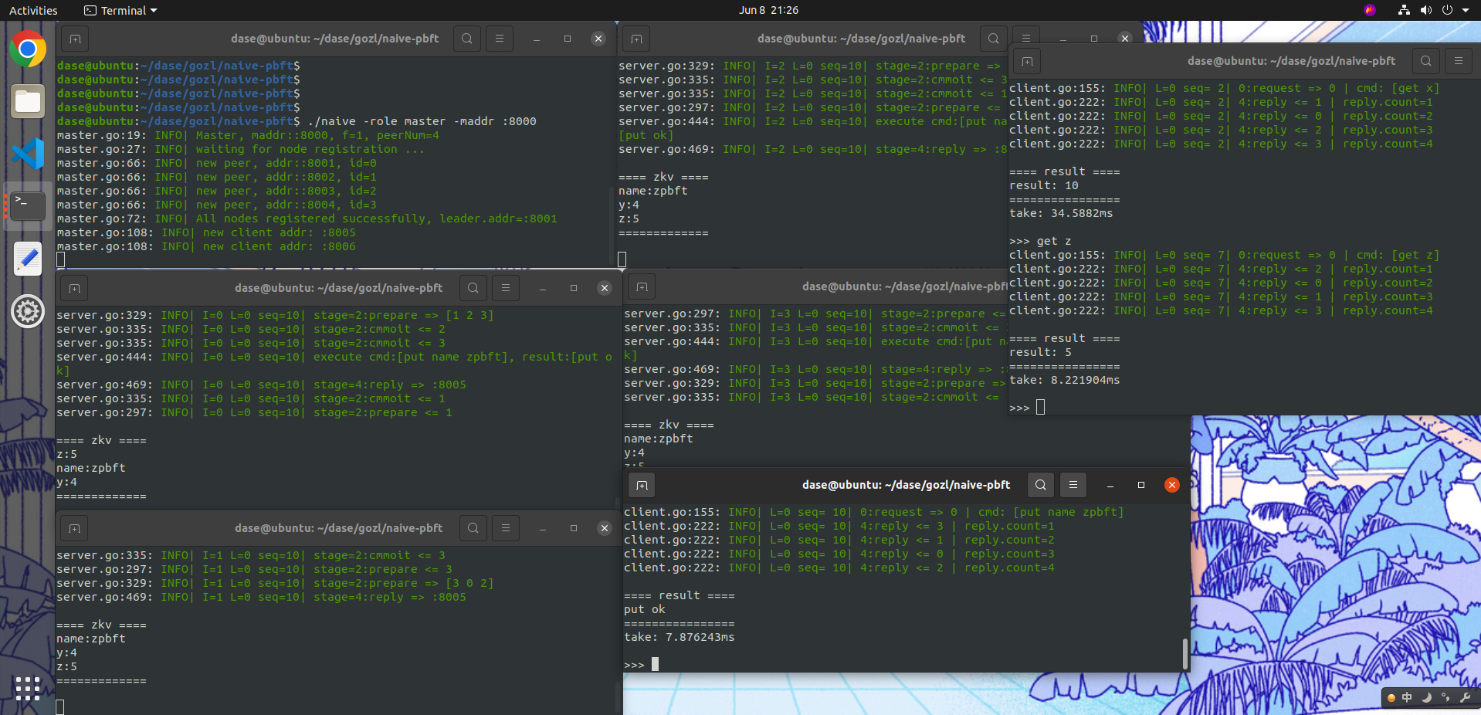
在05中执行 put z 5



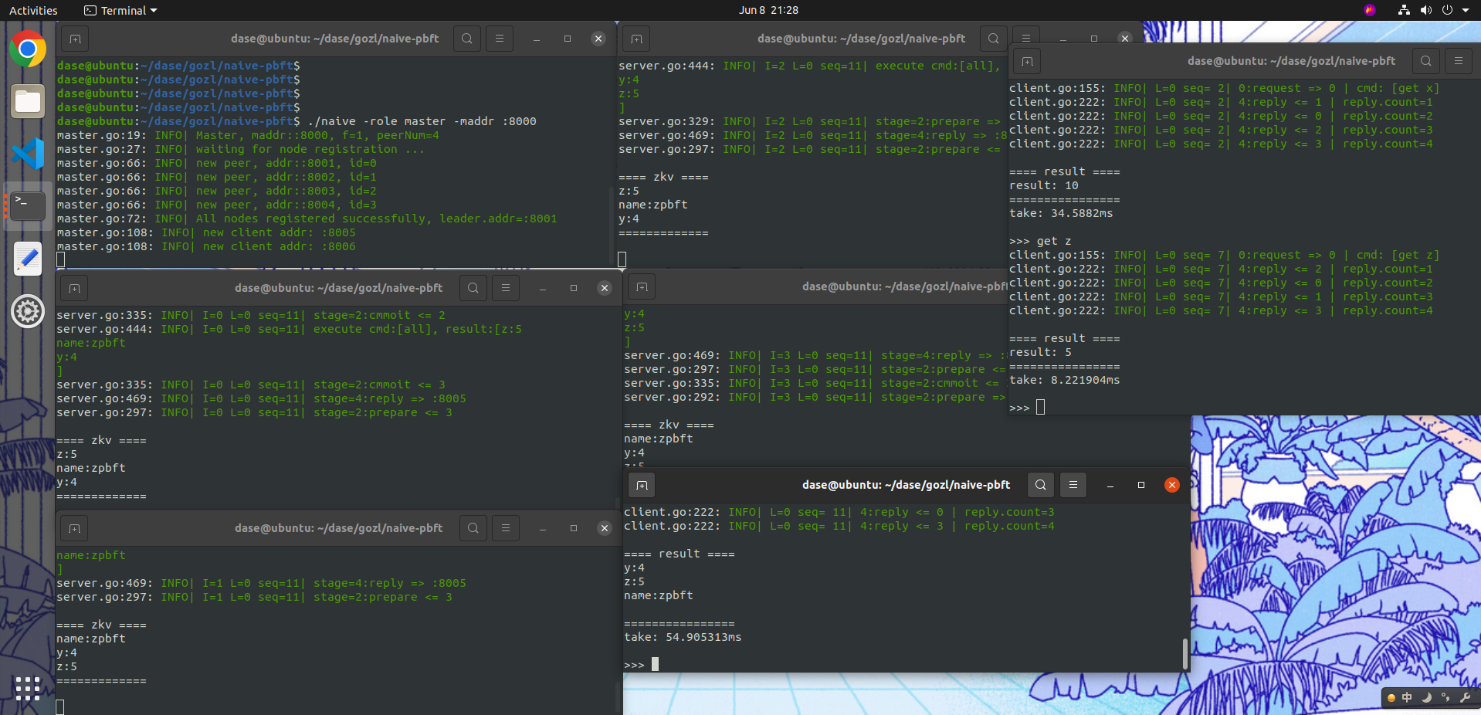
执行 del x



执行 put name zpbft

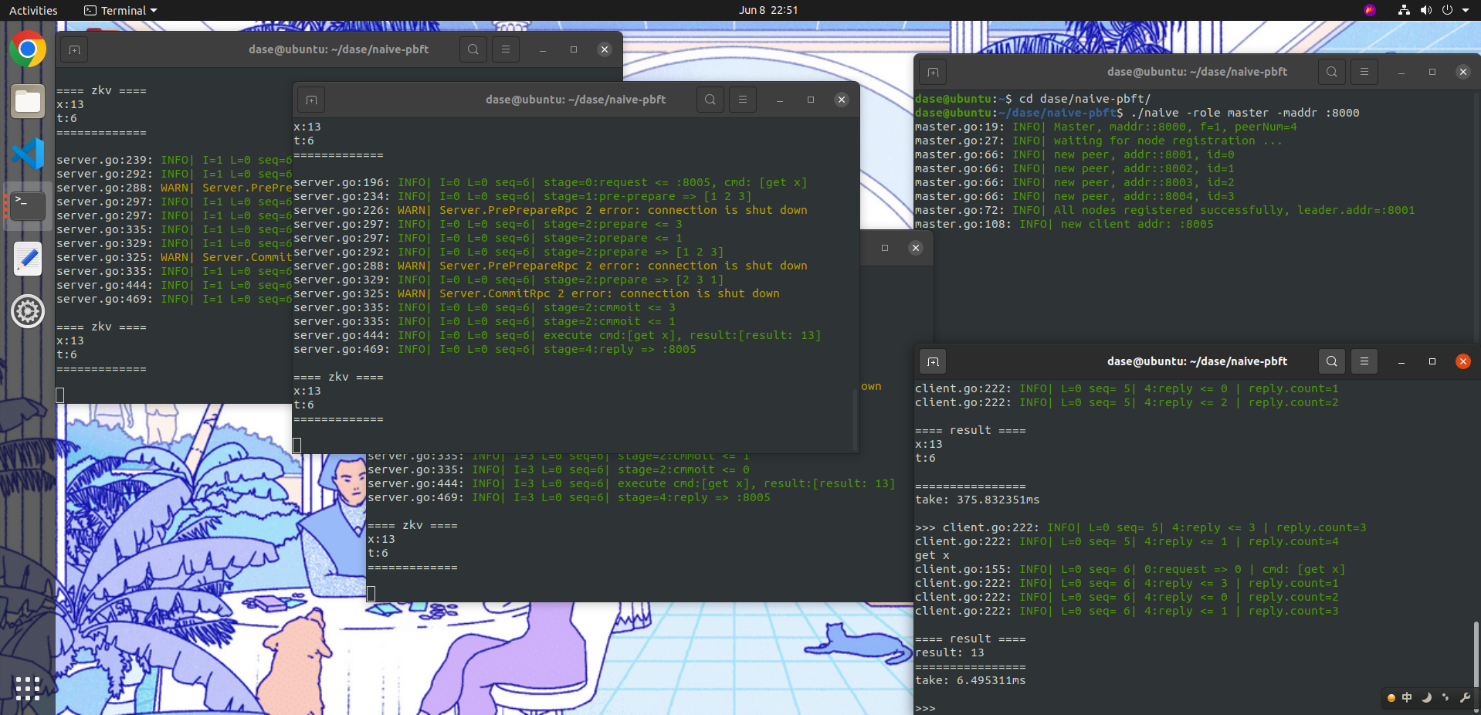


执行all

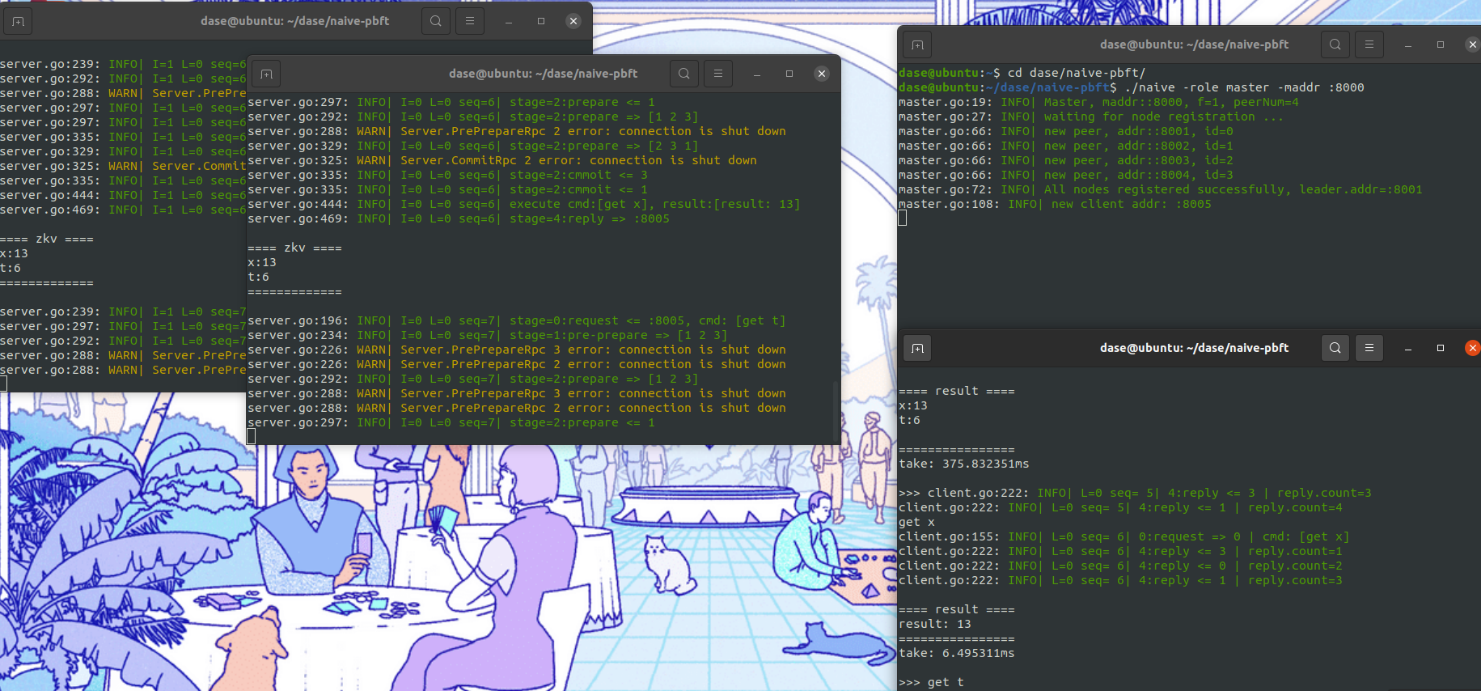


停掉8003

依旧能get x



停掉8003和8002；无法共识；原因是收到的通讯数少于f+1；造成线程一直在等待



1. **总结**

1.请问在节点数量很大时PBFT算法的性能如何？

至少需要O(N^2)的消息复杂度才能让拜占庭节点存在的网络里可靠地广播一条消息，节点数量大时，爆炸增的需求使得性能不好

2.PBFT算法解决了什么问题？请描述其五个阶段

解决BFT问题，（拜占庭容错）；

Request

Preprepare

Prepare

Commit

Reply

3. 以太坊、Fabric等区块链平台均开始使用Golang，经过使用你觉得Golang的优势在哪里？

部署简单、性能优秀、并行执行性能好