Assignment 1

MG1833074

王易木

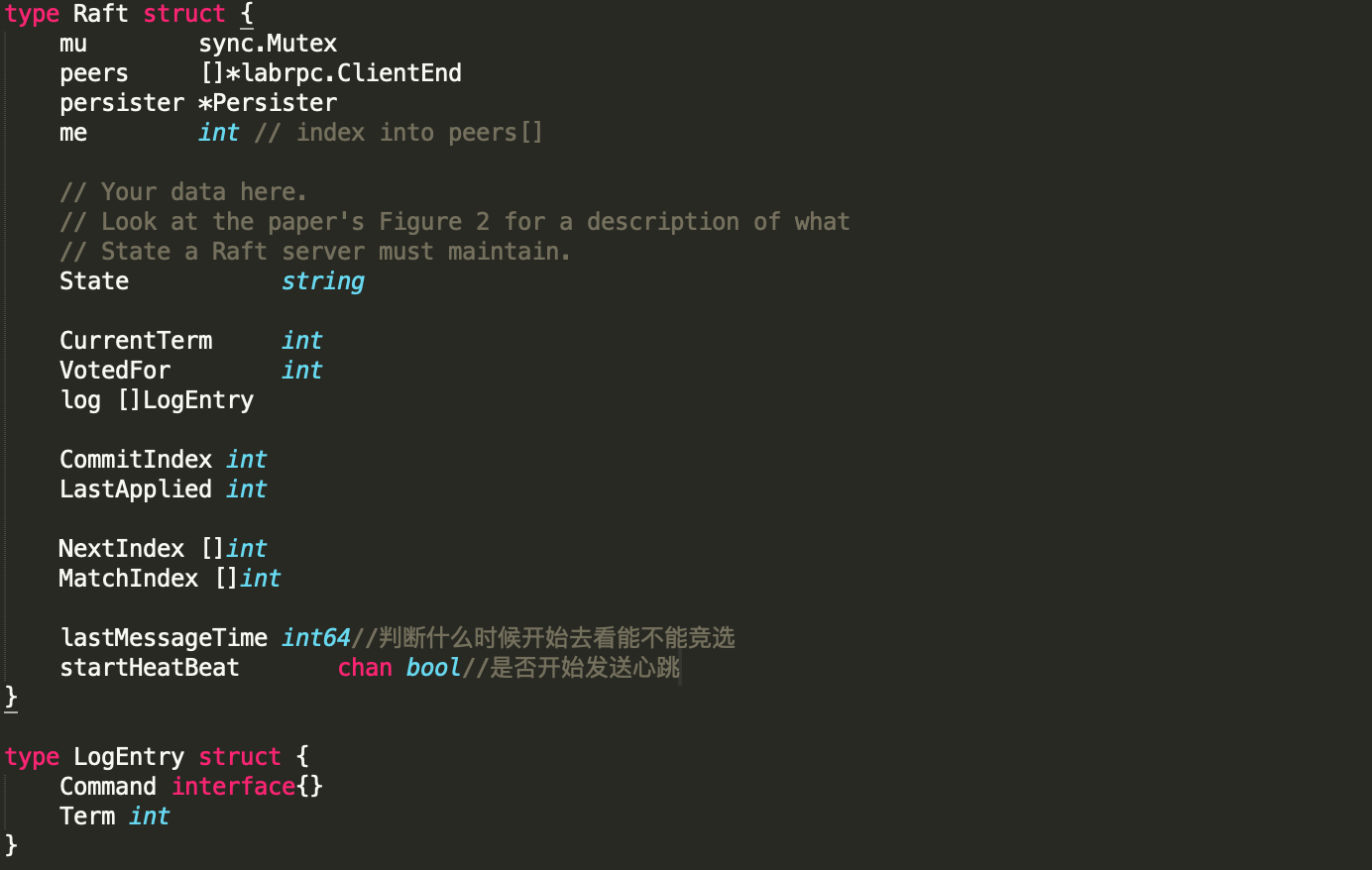
## 如何达到目标

Raft算法主要是Follower、Candidate和Leader之间的转换，以及其中的通信。

在具体实现中，利用go语言的并发性和管道特性，使得角色转换和心跳发送可以同时进行，完成了这一功能。

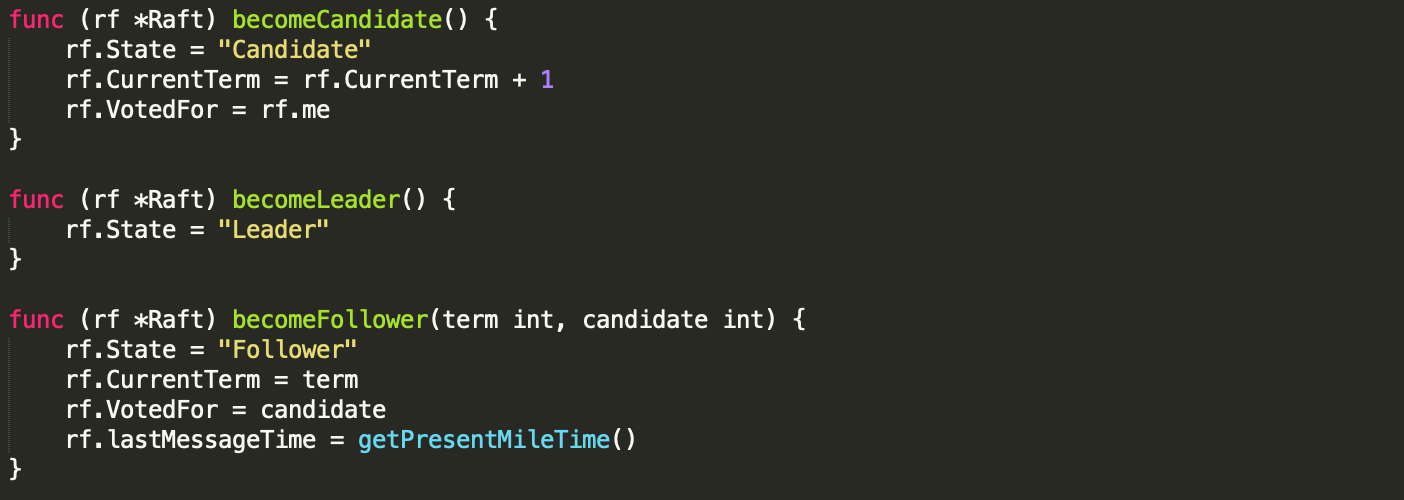
## 分析与设计

首先定义好raft。



其中lastMessageTIme和startHeatBeat是两个自己定义的变量。

以及三个类型的转换：

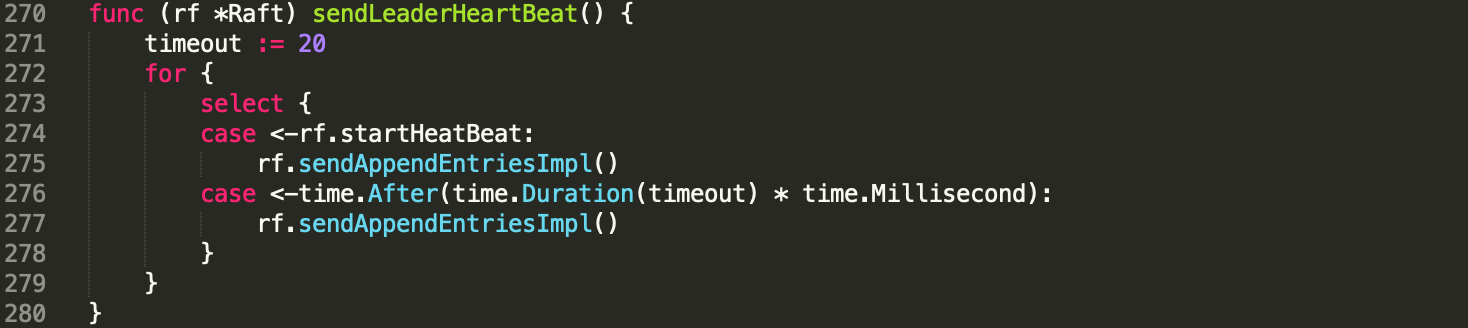


在make函数重定义好整体。

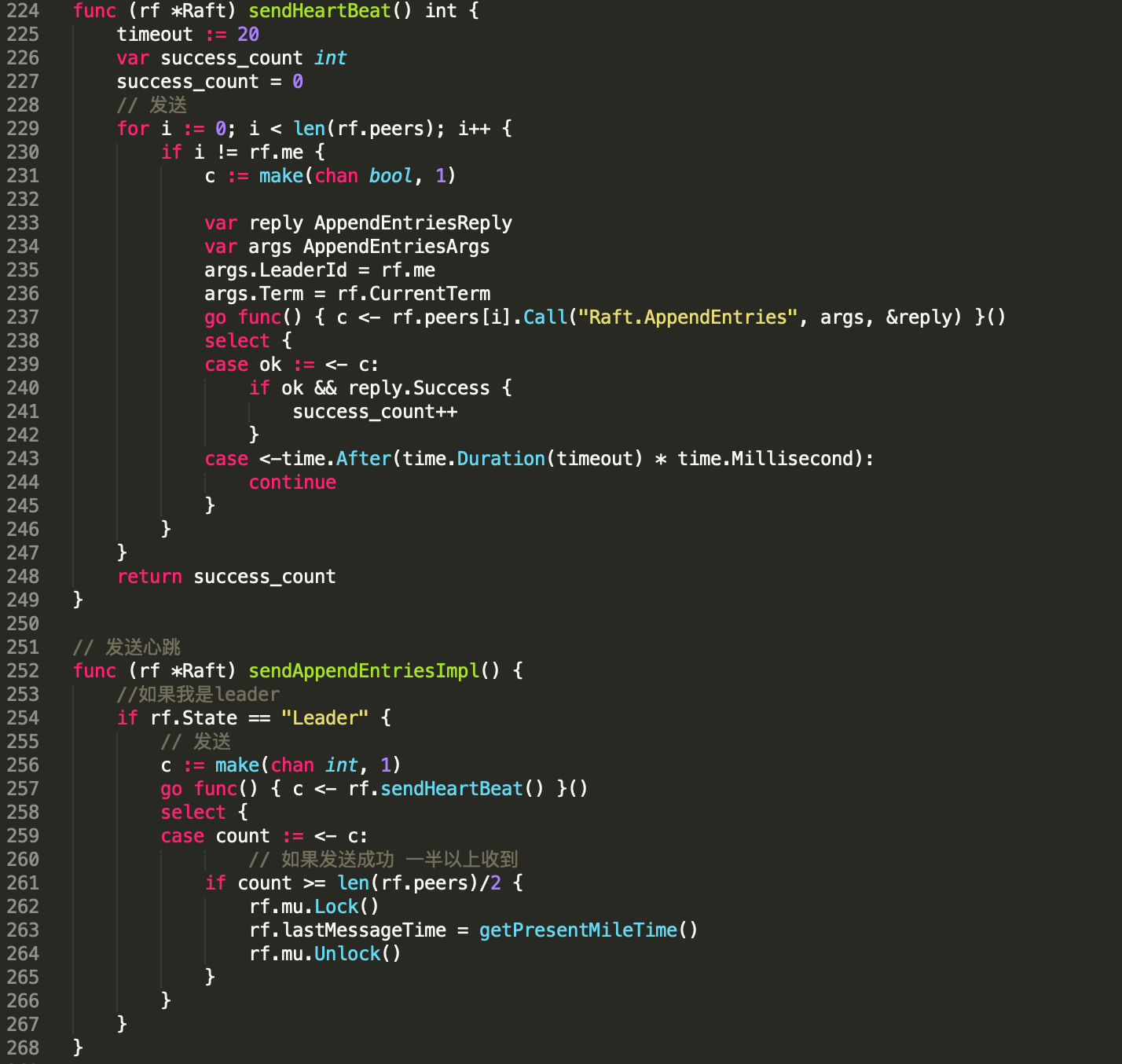


使用GO语法同时运行election和sendleaderheartbeat两个函数。分别进行选举和发送心跳。

首先是简单的发送心跳。当且仅当是leader的时候才可以发送心跳。于是判定条件为管道接收到这个信号。



具体的实现部分就比较简单，一个个发送，如果没有超市并且接收到回复为成功的话那么就计数，当数量超过一半的服务器接收到即视为发送成功。

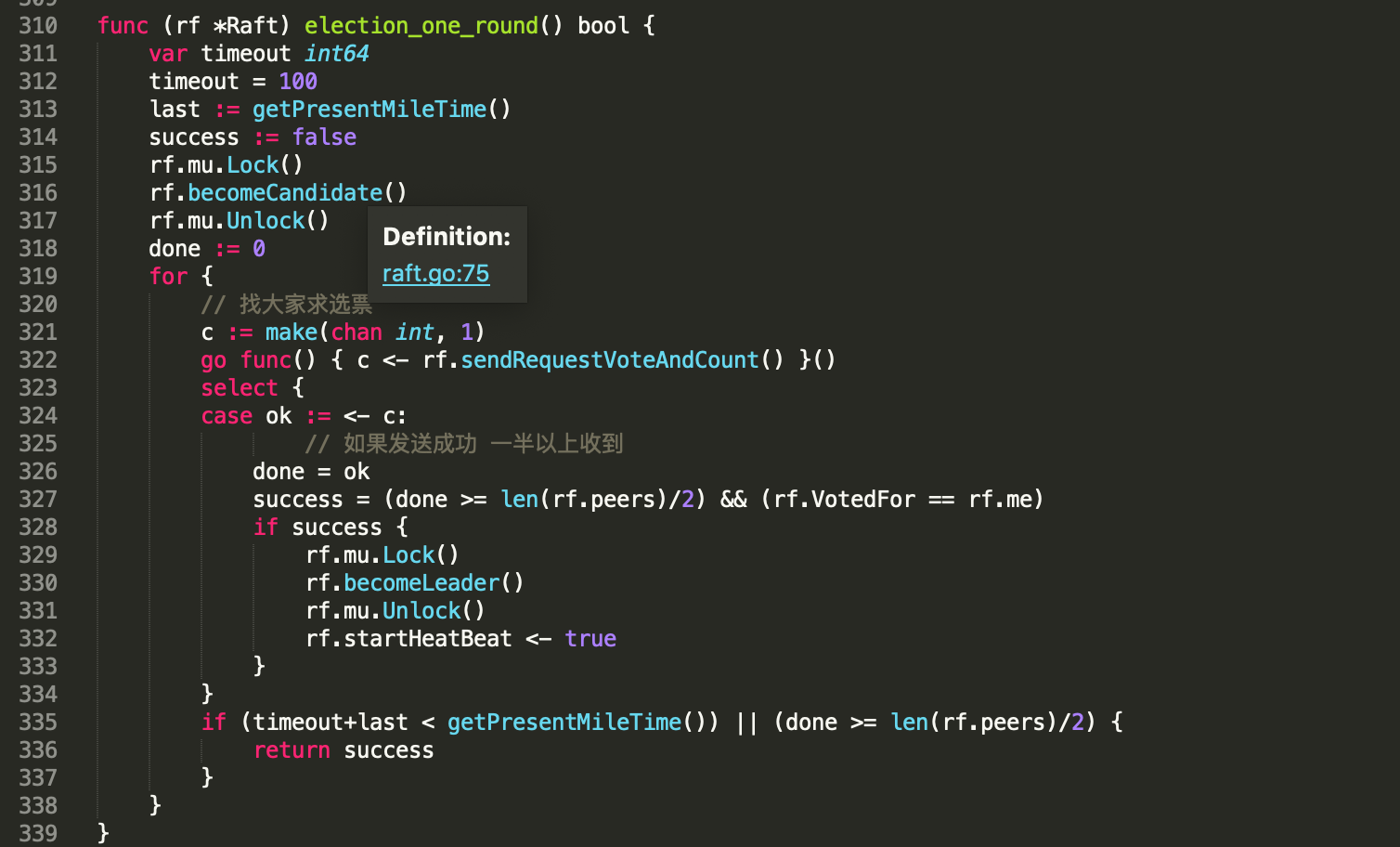


选举部分比较复杂。

首先判断自己是否可以去选举（超时并且没人去选举）。



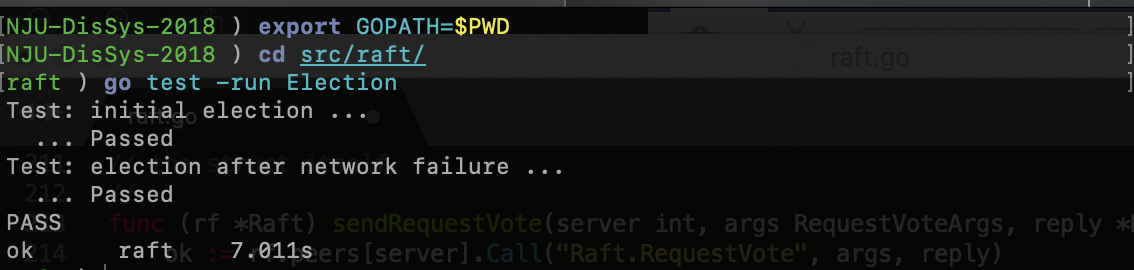
然后开始进行拉票。拉票的算法和发送心跳类似，当对方在规定时间内（没有超时）的情况下回复了给我投票的时候，计数。当计数超过一半成功竞选为leader。



主要的函数便是这样。

## 实现演示

进入主目录然后加入到gopath后便可直接运行。



## 总结

熟悉了Raft算法中三个角色的转换，以及GO语言中的一些语法运用。

Go语言的Channel像管道一样使用起来方便，结合select可以直接设置定时/响应函数。