

## 实验配置

语言: golang 1.10

## 实验思路:

1. 两个 process 分别监听 4000,4001 端口。Process 之间通过 tcp 建立 connection , 并轮转消息 msg。
2. 当控制台输入字符 m 时, process1 开始 sendmarker,同时记录当前状态
3. Preocess 不断监听 connection 传递的消息, 如果是 msg 则继续轮转 msg。如果是 marker, 那么记录当前状态, 再将 channel 标记已遍历, 继续传递 marker。

多次执行 snapshot 效果如下:

```
[cui:AlgoritmoChandyLamport]$>go run *.go
a 101 :4000 false :4001
b 101 :4001 false :4000
输入字符m==start snapshot 输入字符c==clear record
m
2018/07/03 10:02:32 store snapshot in :4000 amount=102
2018/07/03 10:02:32 store snapshot in :4001 amount=102
c
m
2018/07/03 10:02:36 store snapshot in :4000 amount=103
2018/07/03 10:02:36 store snapshot in :4001 amount=103
c
m
2018/07/03 10:02:38 store snapshot in :4000 amount=103
2018/07/03 10:02:40 store snapshot in :4001 amount=104
c
```

注:

- 1.a b 进程其实状态均为 101. 由于轮转消息时间短, 后续两者状态相等情况居多
- 2.每次执行 snapshot 后需要重新将 channel 标记为未遍历状态。每次输入 m 后输入 c