本章作业

使用 Channel 开发一个锁

之前的课程已经提到，go 的 mutex 是没有非阻塞功能的。想要获取锁的协程，在获取不到时，会进入休眠。如果我们不想进入休眠，而是一段时间获取不到就返回失败，就需要一个带有非阻塞功能的锁。本章的作业就是跟着下面的教程，使用 Channel 开发一个带有非阻塞功能的 mutex。**（如果对 Channel 不熟悉，也可以学完下一个章节再回来看这个作业）**

使用 Channel 开发一个基本互斥锁

新建一个 mymutex.go 文件：

```
package main

//声明 MyMutex 结构体
type MyMutex chan struct{}

//构造方法：使用一个缓冲大小为 1 的 channel，载体为空结构体
func NewMyMutex() MyMutex {
 ch := make(chan struct{}, 1)
 return ch
}

//加锁时，向 channel 塞一个数据
//如果已经被加锁，后面的协程无法塞入数据，阻塞
func (m *MyMutex) Lock() {
 (*m) <- struct{}{}
}

//解锁时，从 channel 取一个数据
func (m *MyMutex) UnLock() {
 <-(*m)
}
```

一个基本互斥锁就开发好了。