



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

读写锁 (sync.RWMutex)

目录



PART 01
读写锁
(sync.RWMutex)



读写锁 (sync.RWMutex)

1.读写锁 (sync.RWMutex)



什么是读写锁?

读写锁可以让多个读操作并发，同时读取，但是对于写操作是完全互斥的。

也就是说，**当一个goroutine进行写操作的时候，其他goroutine既不能进行读操作，也不能进行写操作。**

GO中的读写锁由结构体类型`sync.RWMutex`表示，此类型的方法集合中包含两对方法：

1、对写操作的锁定和解锁，简称“写锁定”和“写解锁”：

```
func (*RWMutex)Lock()
```

```
func (*RWMutex)Unlock()
```

2、表示对读操作的锁定和解锁，简称为“读锁定”与“读解锁”：

```
func (*RWMutex)RLock()
```

```
func (*RWMutex)RUnlock()
```



读写锁 (sync.RWMutex)

处于读锁定状态，那么针对它的写锁定操作将永远不会成功，且相应的Goroutine也会被一直阻塞，因为它们互斥。

总结：读写锁控制下的多个写操作之间都是互斥的，并且写操作与读操作之间也都是互斥的，但是多个读操作之间不存在互斥关系。

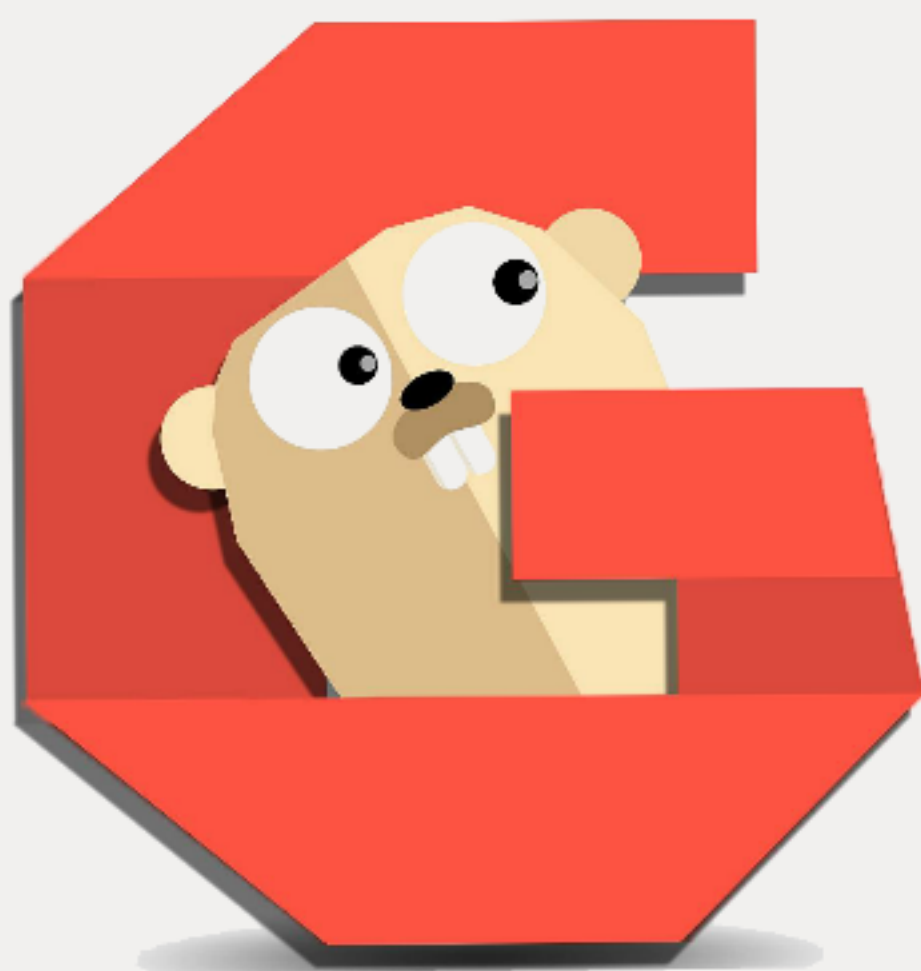
从互斥锁和读写锁的源码可以看出，它们是同源的。
读写锁的内部用互斥锁来实现写锁定操作之间的互斥。

可以把读写锁看作是互斥锁的一种扩展。



实例演示





江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看