

-gmutex

gmutex.Mutexsync.RWMutex

<https://github.com/gogf/gf/blob/master/os/gmutex/gmutex.go>

```
import "github.com/gogf/gf/v2/os/gmutex"
```

Content Menu

-
- [1](#)
- [2*Func](#)

<https://pkg.go.dev/github.com/gogf/gf/v2/os/gmutex>

```
type Mutex
    func (m *Mutex) LockFunc(f func())
    func (m *Mutex) TryLockFunc(f func()) (result bool)
type RWMutex
    func New() *RWMutex
    func (m *RWMutex) LockFunc(f func())
    func (m *RWMutex) RLockFunc(f func())
    func (m *RWMutex) TryLockFunc(f func()) (result bool)
    func (m *RWMutex) TryRLockFunc(f func()) (result bool)
```

1. Try**Func
2. Try*truefalse
3. *Funcdefer

gmutex.Mutexsync.Mutexsync.RWMutex [gmutex_bench_test.go](#)

```
goos: linux
goarch: amd64
pkg: github.com/gogf/gf/v2/os/gmutex
Benchmark_Mutex_LockUnlock-4      500000000      31.5 ns/op
Benchmark_RWMutex_LockUnlock-4    300000000      54.1 ns/op
Benchmark_RWMutex_RLockRUnlock-4 500000000      27.9 ns/op
Benchmark_GMutex_LockUnlock-4     500000000      27.2 ns/op
Benchmark_GMutex_TryLock-4        100000000      16.7 ns/op
Benchmark_GMutex_RLockRUnlock-4   500000000      38.0 ns/op
Benchmark_GMutex_TryRLock-4       100000000      16.8 ns/op
```

```

package main

import (
    "time"

    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gmutex"
)

func main() {
    ctx := gctx.New()
    mu := gmutex.New()
    for i := 0; i < 10; i++ {
        go func(n int) {
            mu.Lock()
            defer mu.Unlock()
            glog.Print(ctx, "Lock:", n)
            time.Sleep(time.Second)
        }(i)
    }
    for i := 0; i < 10; i++ {
        go func(n int) {
            mu.RLock()
            defer mu.RUnlock()
            glog.Print(ctx, "RLock:", n)
            time.Sleep(time.Second)
        }(i)
    }
    time.Sleep(15 * time.Second)
}

```

```

2019-07-13 16:19:55.417 Lock: 0
2019-07-13 16:19:56.421 Lock: 1
2019-07-13 16:19:57.424 RLock: 0
2019-07-13 16:19:57.424 RLock: 4
2019-07-13 16:19:57.425 RLock: 8
2019-07-13 16:19:57.425 RLock: 2
2019-07-13 16:19:57.425 RLock: 7
2019-07-13 16:19:57.425 RLock: 5
2019-07-13 16:19:57.425 RLock: 9
2019-07-13 16:19:57.425 RLock: 1
2019-07-13 16:19:57.425 RLock: 6
2019-07-13 16:19:57.425 RLock: 3
2019-07-13 16:19:58.429 Lock: 3
2019-07-13 16:19:59.433 Lock: 4
2019-07-13 16:20:00.438 Lock: 5
2019-07-13 16:20:01.443 Lock: 6
2019-07-13 16:20:02.448 Lock: 7
2019-07-13 16:20:03.452 Lock: 8
2019-07-13 16:20:04.456 Lock: 9
2019-07-13 16:20:05.461 Lock: 2

```

glog3gmutex.Mutex

2*Func

```
package main

import (
    "time"

    "github.com/gogf/gf/v2/os/glog"

    "github.com/gogf/gf/v2/os/gmutex"
)

func main() {
    mu := gmutex.New()
    go mu.LockFunc(func() {
        glog.Println("lock func1")
        time.Sleep(1 * time.Second)
    })
    time.Sleep(time.Millisecond)
    go mu.LockFunc(func() {
        glog.Println("lock func2")
    })
    time.Sleep(2 * time.Second)
}
```

```
2019-07-13 16:28:10.381 lock func1
2019-07-13 16:28:11.385 lock func2
```

*Func