درس ۱۰

Go

select · mutex · ctx

main.go

```
select {
case v := <-ch:
case <-time.After(time.Second):
}
```

همه را یاد بگیرید @MyWebUniversity-Dari

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Go

درس ۱۰

select، mutex ها و context

select با timeout، sync.Mutex، race detector و لغو با context



select

select روی چند عملیات channel یکجا منتظر می‌ماند و هر کدام اول آماده شود اجرا می‌کند. مثل ...

یک case با time.After اضافه کنید تا یک timeout تعیین شود، تا پروگرام شما هرگز برای همیشه ...

select.go

```
select {
case msg := <-messages:
    fmt.Println("got", msg)
case err := <-errs:
    fmt.Println("error", err)
case <-time.After(2 * time.Second):
    fmt.Println("timeout")
}
```

01

data race ها

وقتی دو goroutine همزمان یک متغیر را تغییر دهند، یک data race و نتیجه‌های غلط می‌گیرید، مثل ...

پروگرام یا آزمایش‌های خود را با بیرق -race اجرا کنید و Go race ها را پیدا می‌کند و دقیقاً ...

terminal

```
$ go run -race counter.go
==================
WARNING: DATA RACE
Write at 0x00c000012345 by goroutine 8:
  main.main.func1()  counter.go:12
Previous write by goroutine 7:
```

02

sync.Mutex

یک mutex داده‌های شریک را محافظت می‌کند:
پیش از دست زدن Lock کنید و بعد .Unlock فقط یک ...

mutex را کنار داده‌ای که محافظت می‌کند بگذارید، و از defer Unlock استفاده کنید تا هرگز ...

counter.go

```go
type Counter struct {
    mu sync.Mutex
    n  map[string]int
}
func (c *Counter) Inc(key string) {
    c.mu.Lock()
    defer c.mu.Unlock()
    c.n[key]++
}
```

03

context: لغو و مهلت

یک context یک مهلت یا سیگنال لغو را در پروگرام شما می‌برد. آن را به عنوان پارامتر اول، به ...

وقتی کاربر لغو می‌کند یا وقت تمام می‌شود، ctx.Done() بسته می‌شود، و هر goroutine که گوش ...

context.go

```go
ctx, cancel := context.WithTimeout(context.Background(), 3*time.Second)
defer cancel()

req, _ := http.NewRequestWithContext(ctx, "GET", url, nil)
resp, err := http.DefaultClient.Do(req)
if errors.Is(err, context.DeadlineExceeded) {
    fmt.Println("too slow, gave up")
}
```

04

قاعده‌های سادهٔ همزمانی

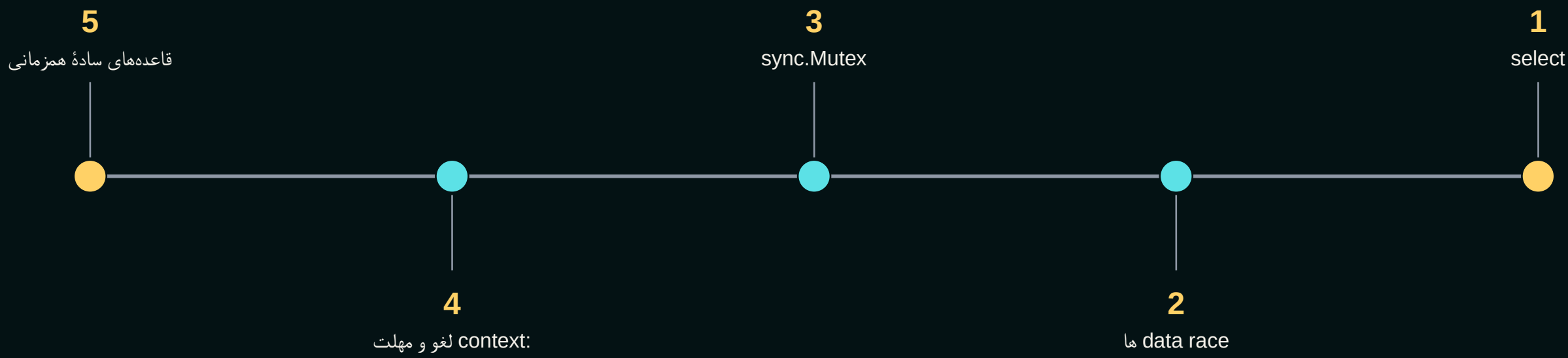
بدانید هر goroutine چطور متوقف می‌شود، وگرنه نشت می‌کند. از channel ها برای سپردن مالکیت ...

همیشه کود همزمان را با race detector آزمایش کنید، و به هر چیزی که کار شبکه یا دیسک انجام ...

قاعده	چرا
هر goroutine توقف کند	بدون نشت
channel داده می‌فرستد	یک مالک در یک وقت
mutex وضعیت را محافظت می‌کند	بدون data race
`go test -race`	race ها را زود بگیرید
ctx پارامتر اول	لغو همه جا

05

نقشهٔ درس


5
قاعده‌های سادهٔ همزمانی
3
sync.Mutex
1
select
4
context: لغو و مهلت
2
data race ها

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity