

COND

Terminal: Concurrency × + ∨



УСЛОВНАЯ ПЕРЕМЕННАЯ

Примитив синхронизации, обеспечивающий блокирование одного или нескольких потоков до момента поступления сигнала от другого потока о выполнении некоторого условия

API



```
1 type Cond struct {  
2     L Locker  
3 }  
4  
5 func NewCond(l Locker) *Cond // создает Cond с использованием мьютекса  
6 func (c *Cond) Broadcast()   // оповещает все горютины  
7 func (c *Cond) Signal()      // оповещает одну горютину  
8 func (c *Cond) Wait()        // ожидает наступления сигнала
```

EXAMPLE

Cond

ЧТО ПРОИСХОДИТ ВНУТРИ WAIT()




```
1 c.checker.check()  
2 t := runtime_notifyListAdd(&c.notify)  
3 c.L.Unlock()  
4 runtime_notifyListWait(&c.notify, t)  
5 c.L.Lock()
```


EXAMPLE

Cond operations

THE CALLER SHOULD WAIT IN A LOOP



```
1 c.L.Lock()  
2 for !condition() {  
3     c.Wait()  
4 }  
5 ... make use of condition ...  
6 c.L.Unlock()
```

Terminal: Concurrency x + v



SPURIOUS WAKEUP

В вычислениях ложное пробуждение происходит, когда поток просыпается после ожидания условной переменной без удовлетворения этой переменной.

Его называют ложным, потому что поток, по-видимому, был разбужен без всякой причины

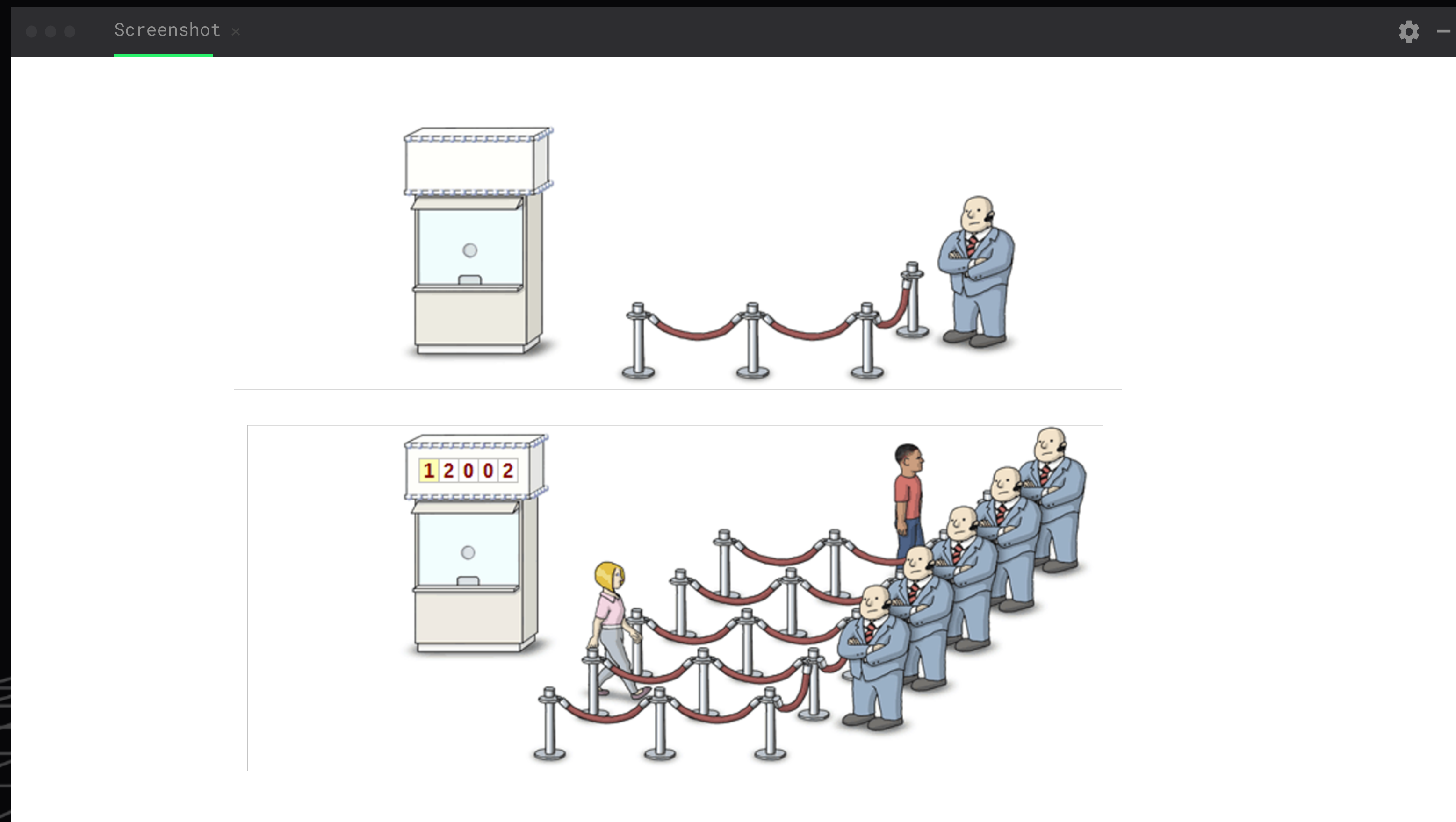
Terminal: Concurrency x + ∨



СЕМАФОР

Примитив синхронизации, который используется для управления доступом к общим ресурсам несколькими потоками (горутинами)

СЕМАФОРЫ



EXAMPLE

Semaphore