



# MUTEX

# EXAMPLE

Incorrect increment



```
1 value++
```



```
1 oldValue := value  
2 newValue := oldValue + 1  
3 value = newValue
```

value = 100

G1

**oldValue := value // oldValue = 100**

G2

**oldValue := value // oldValue = 100**

G1

**newValue := oldValue + 1 // newValue = 101**  
**value = newValue // value = 101**

G2

**newValue := oldValue + 1 // newValue = 101**  
**value = newValue // value = 101**

value = 101

Terminal: Concurrency × + ∨



# МЬЮТЕКСЫ

Mutual Exclusion – взаимное исключение



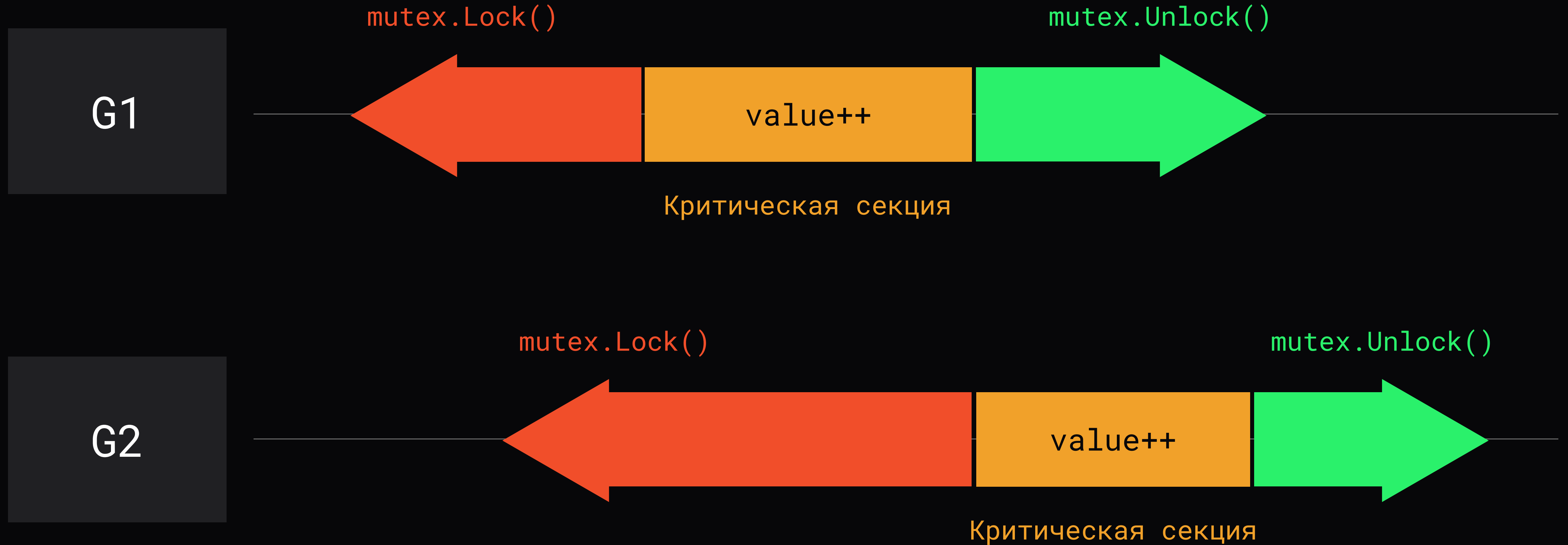
# API



```
1 type Mutex struct { ... }  
2  
3 func (m *Mutex) Lock()           // захватить мьютекс  
4 func (m *Mutex) Unlock()         // освободить мьютекс  
5 func (m *Mutex) TryLock() bool  // попробовать захватить мьютекс
```

# EXAMPLE

Correct increment



Горутина захватывает мьютекс, затем входит  
в критическую секцию, а в конце освобождает мьютекс



`mutex.Lock()`  
`value++`  
`mutex.Unlock()`

G1  
(running)

`mutex.Lock()`  
`value++`

G1  
(ready)

`mutex.Unlock()`

G1  
(running)

G1  
(ready)

`mutex.Lock()`  
`value++`  
`mutex.Unlock()`

G2  
(ready)

`mutex.Lock()`

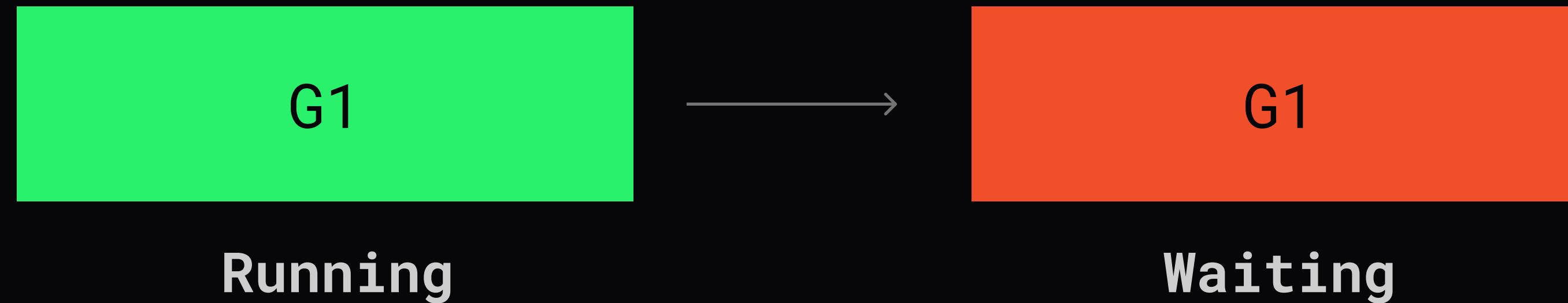
G2  
(running)

G2  
(waiting)

`value++`  
`mutex.Unlock()`

G2  
(running)

Когда горутина блокируется на мьютексе,  
она переходит из состояния **Running** в **Waiting**



# EXAMPLE

Mutex different operations



# EXAMPLE

Mutex with local values



# EXAMPLE

Mutex for order



# EXAMPLE

Print foo bar alternately