



# Semaphore – a more general lock

Creating a semaphore

```
Semaphore semaphore = new Semaphore(4);
```

**Permits**

(can be negative)

```
try {
```

```
    semaphore.acquire();
```

```
} catch (InterruptedException e) {
```

```
    e.printStackTrace();
```

```
}
```

Taking the semaphore

(Decrements number of **permits**)

(Blocks if there are no **permits**)

Releasing the semaphore

(Increments number of **permits**)



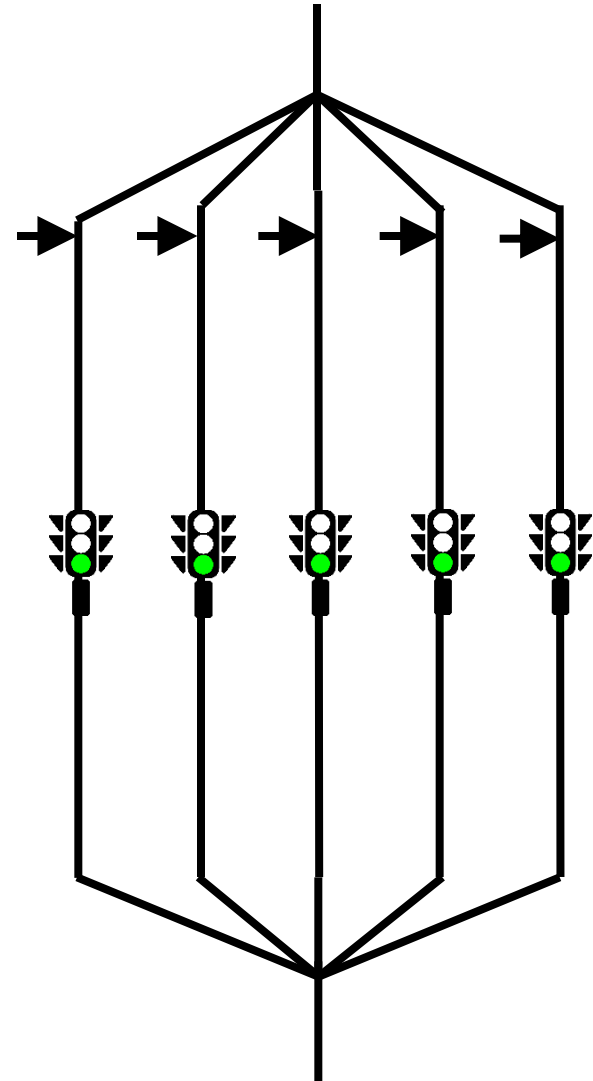
```
semaphore.release();
```

**A semaphore initialized with 1 can be used exactly as a lock**

# Semaphore

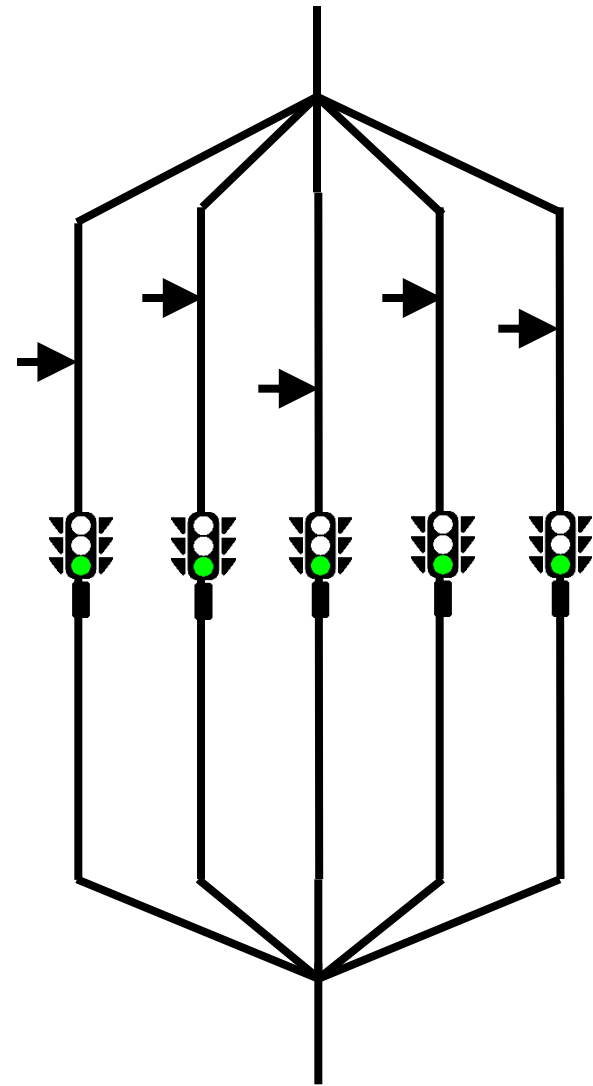
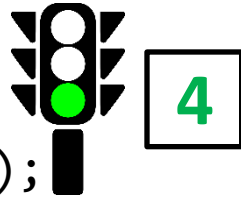
```
Semaphore semaphore = new Semaphore(4);
```

```
semaphore.acquire();
```

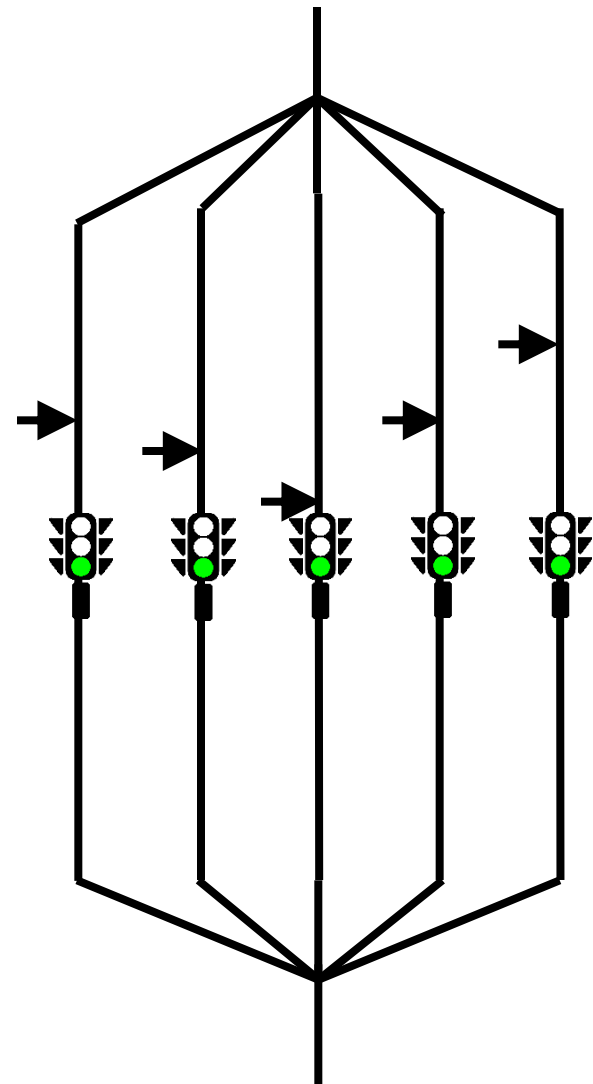
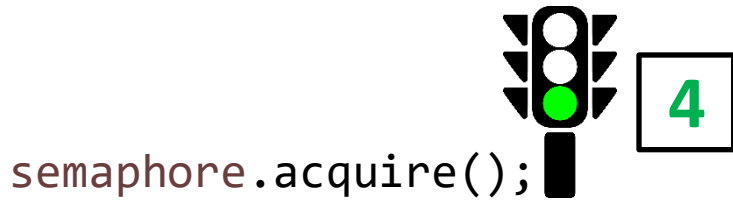
A stylized icon of a traffic light with three lenses, the bottom one of which is illuminated green.A square box containing the number '4' in green, representing the semaphore's initial count.

# Semaphore

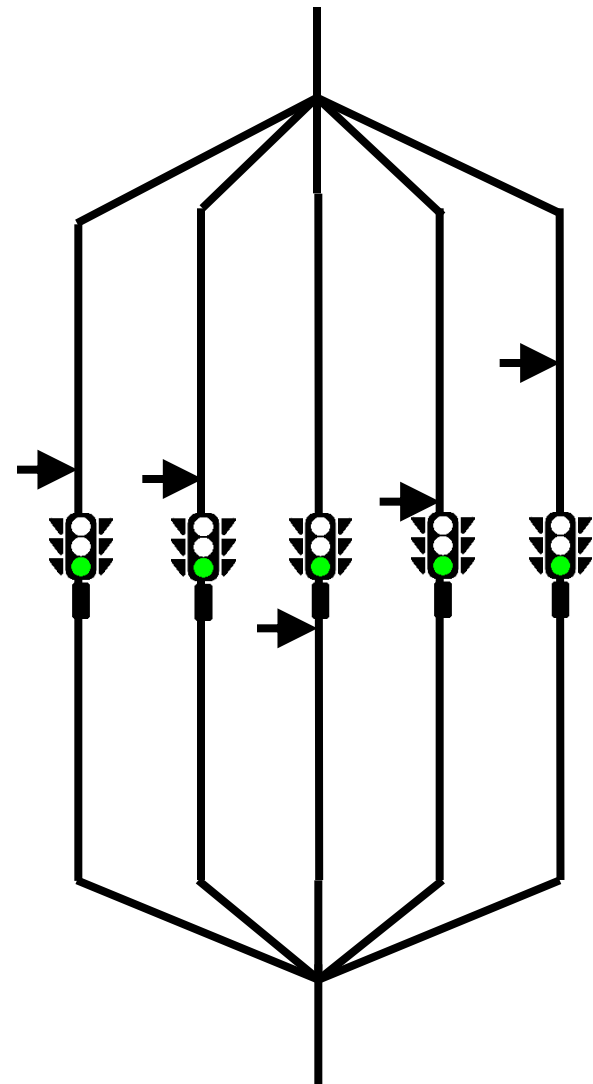
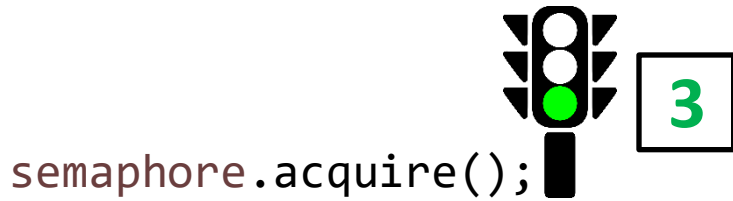
```
semaphore.acquire();
```



# Semaphore

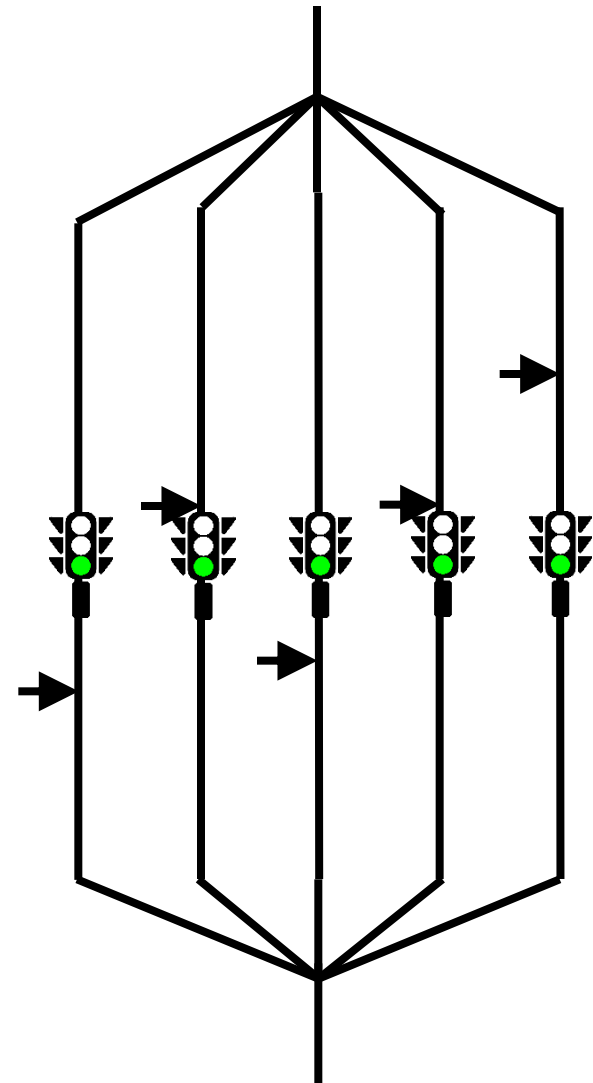
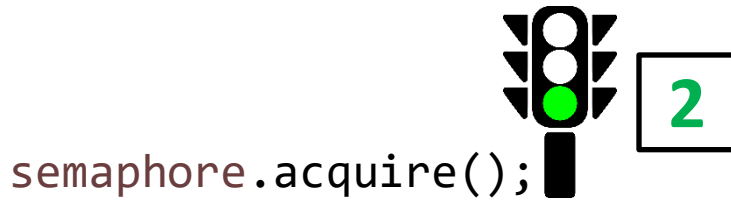


# Semaphore



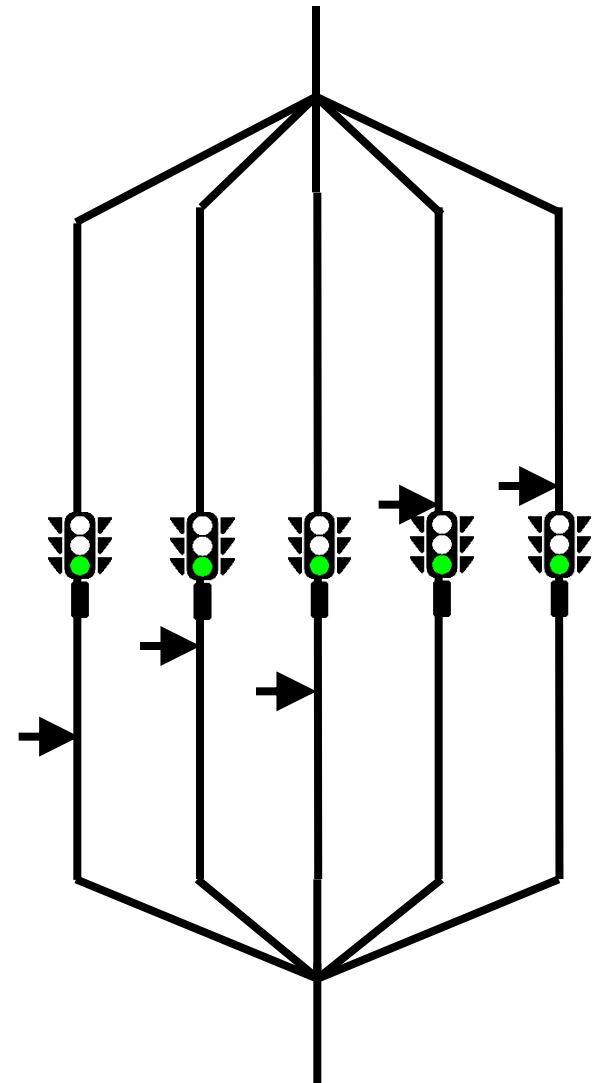
# Semaphore

It does not matter that 2 threads get to the semaphore at the same time. Only one can interact with it at a time.



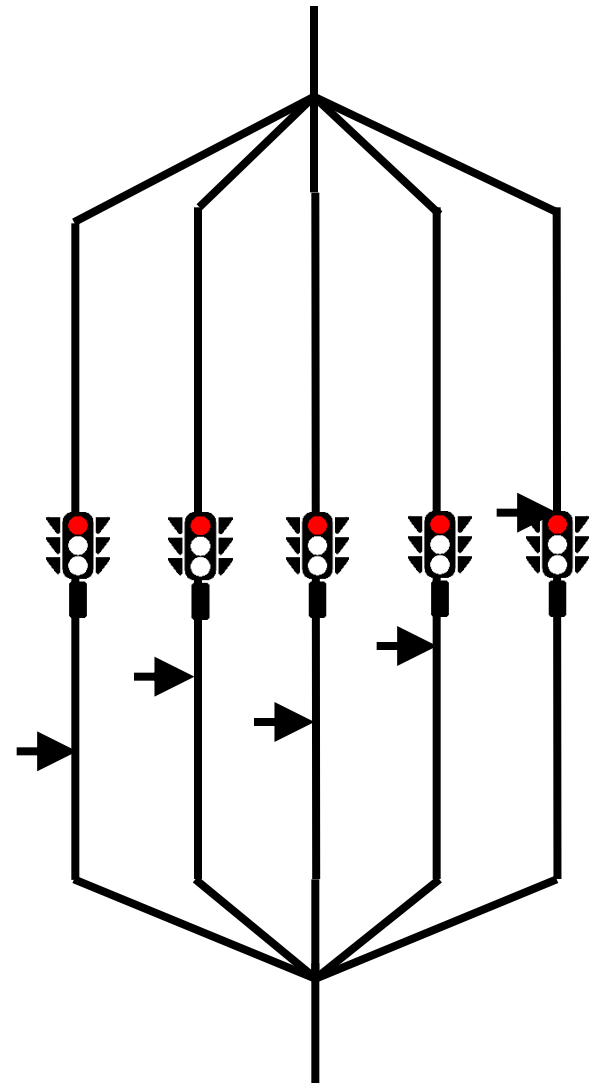
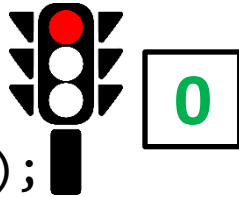
# Semaphore

It does not matter that 2 threads get to the semaphore at the same time. Only one can interact with it at a time.



# Semaphore

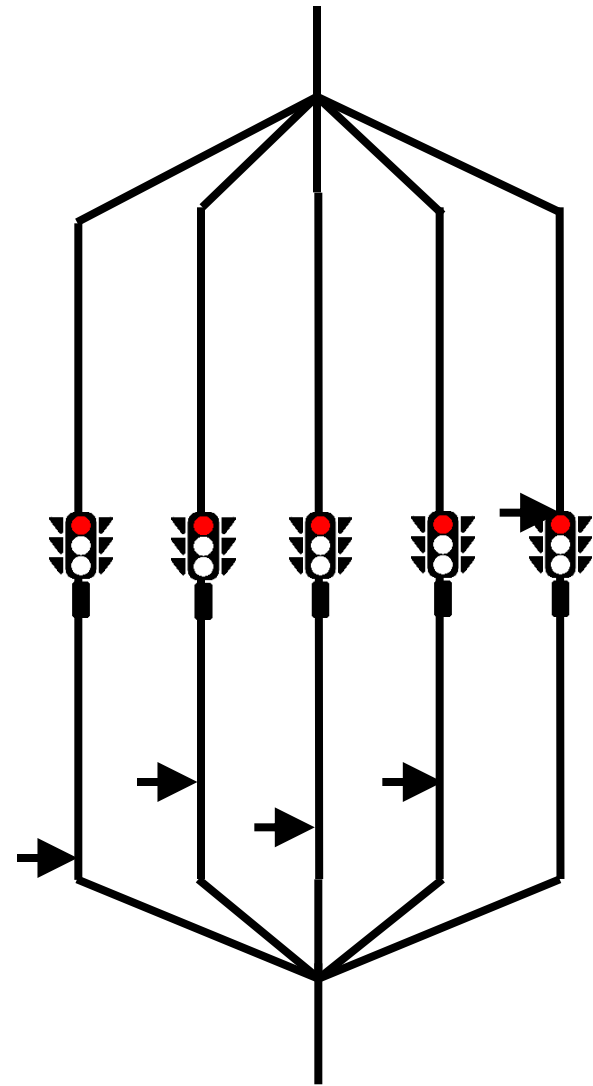
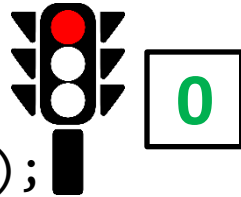
`semaphore.acquire();`





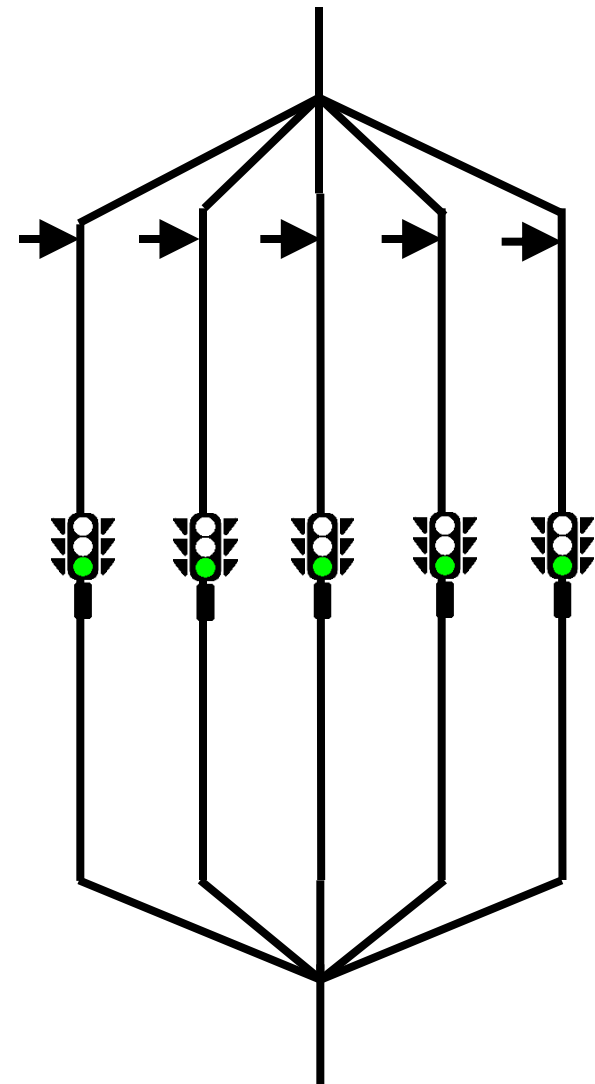
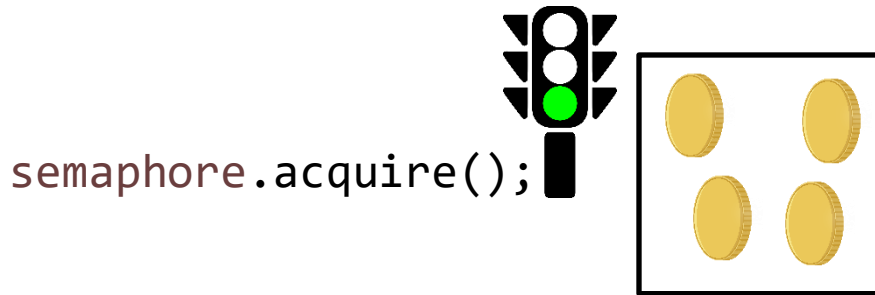
# Semaphore

`semaphore.acquire();`



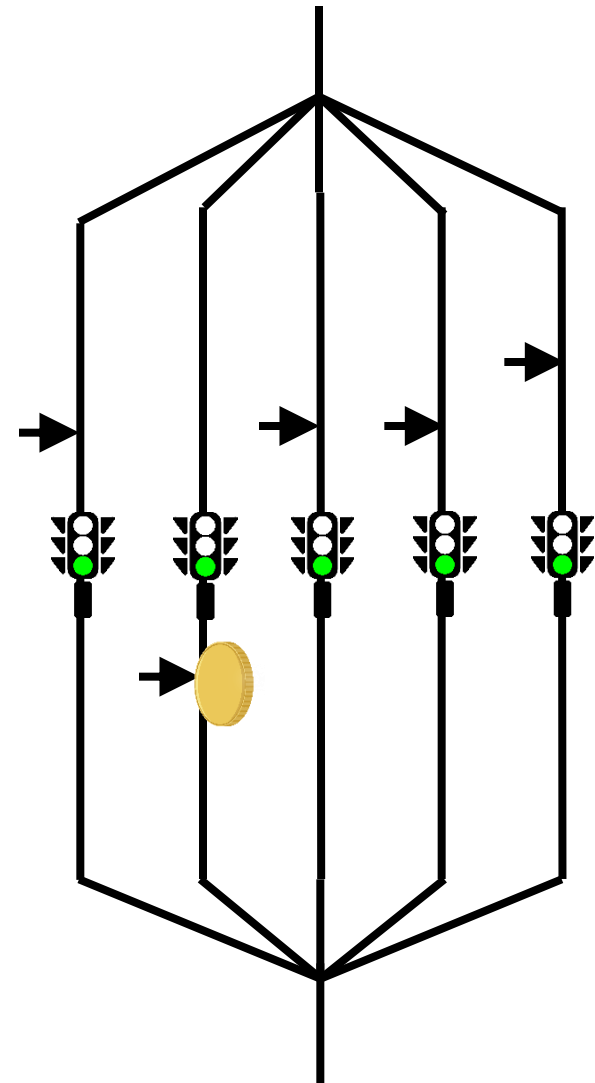
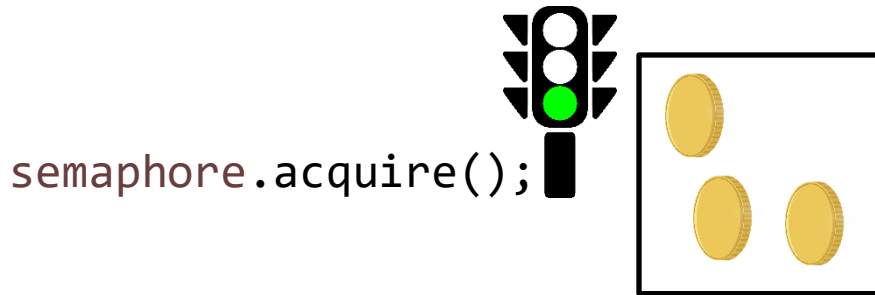
# Semaphore – A different way of thinking

A semaphore has a number of tokens.



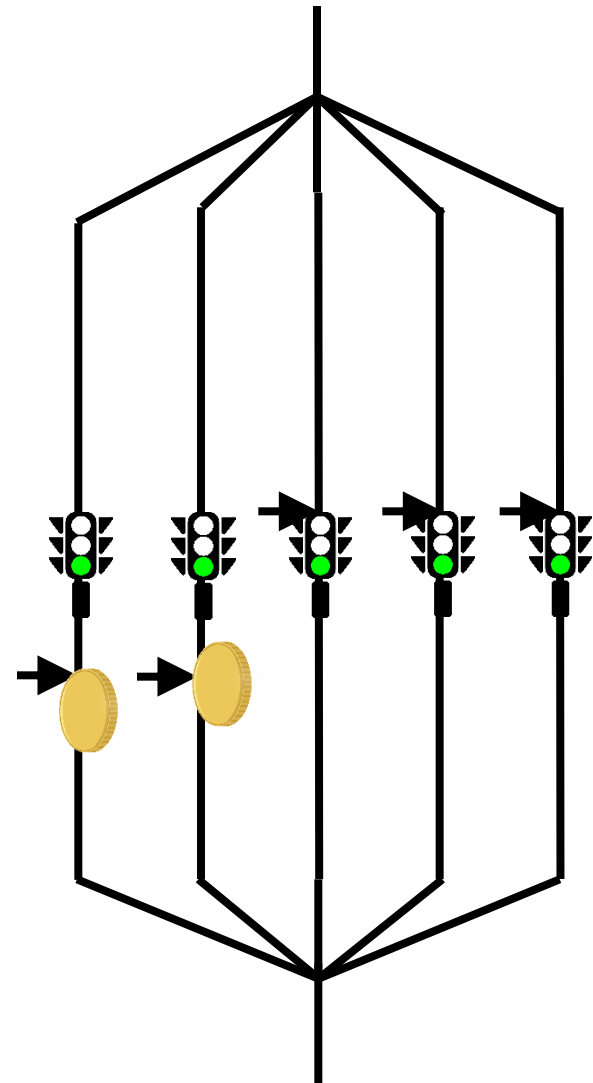
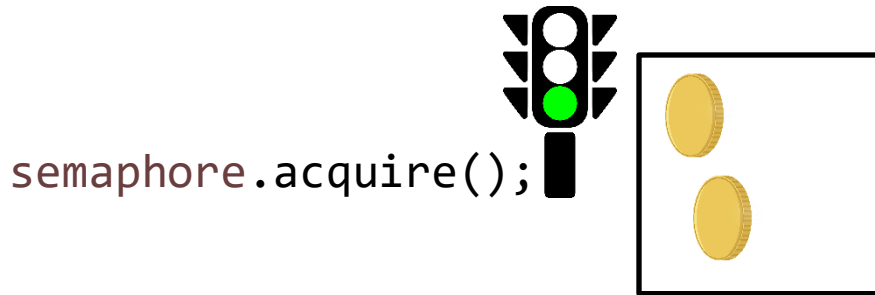
# Semaphore – A different way of thinking

A semaphore has a number of tokens.



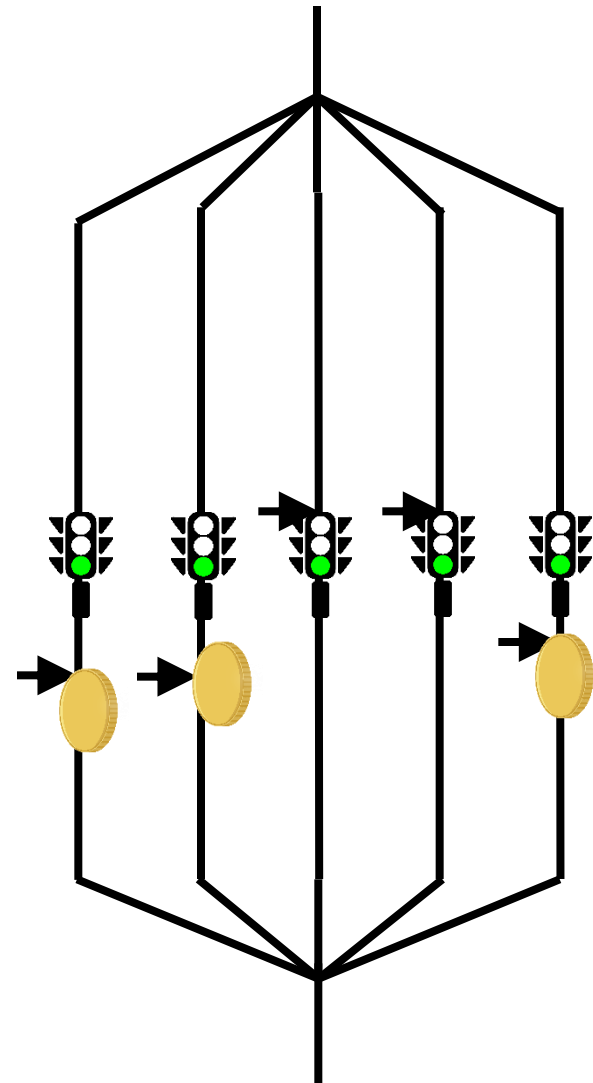
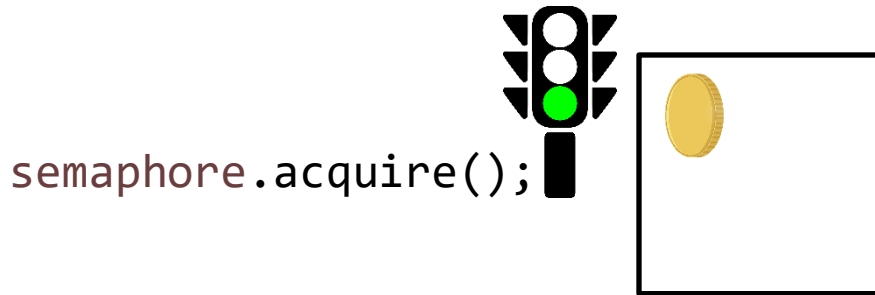
# Semaphore – A different way of thinking

A semaphore has a number of tokens.



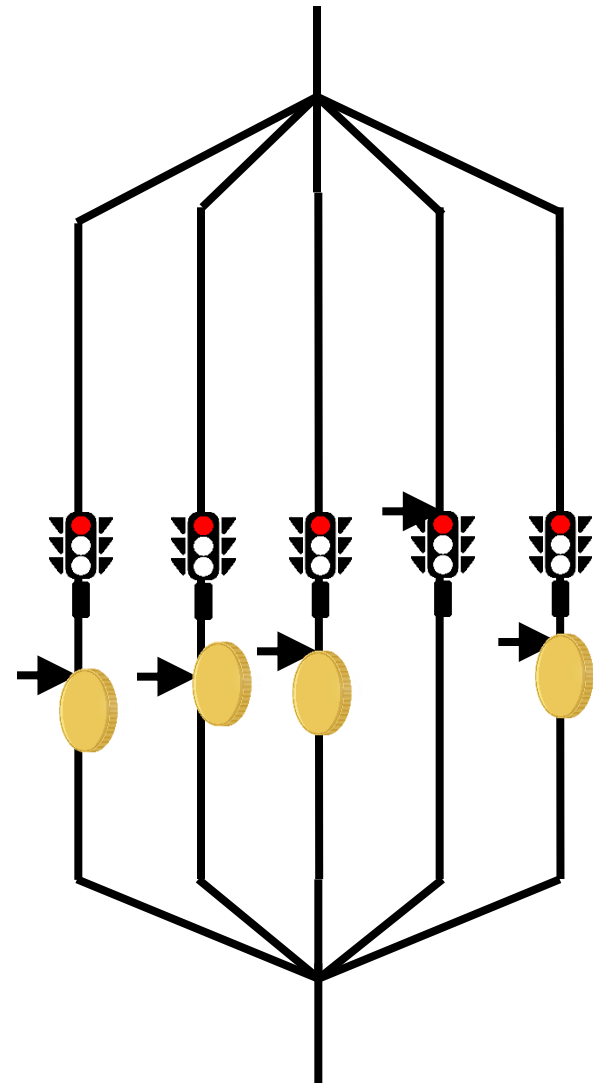
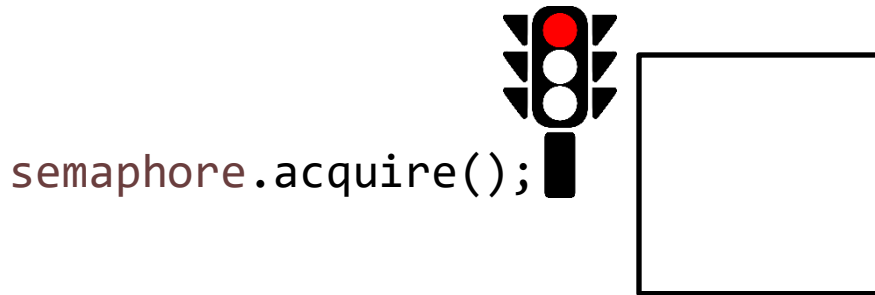
# Semaphore – A different way of thinking

A semaphore has a number of tokens.






# Semaphore – A different way of thinking

A semaphore has a number of tokens.

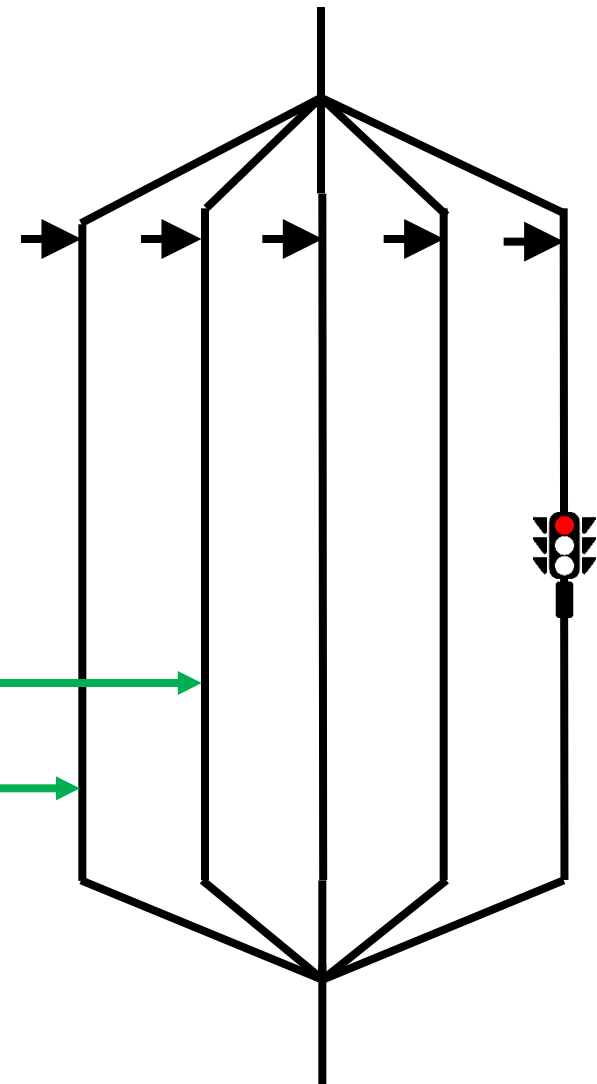


# Semaphore – Signaling

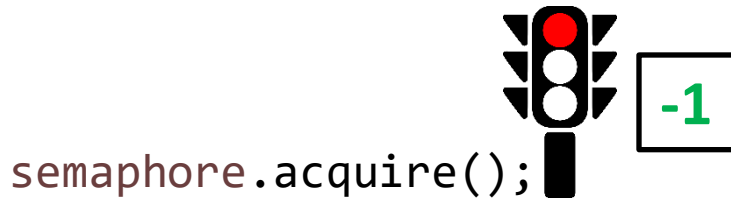
`semaphore.acquire();`  

`semaphore.release();` 

`semaphore.release();` 



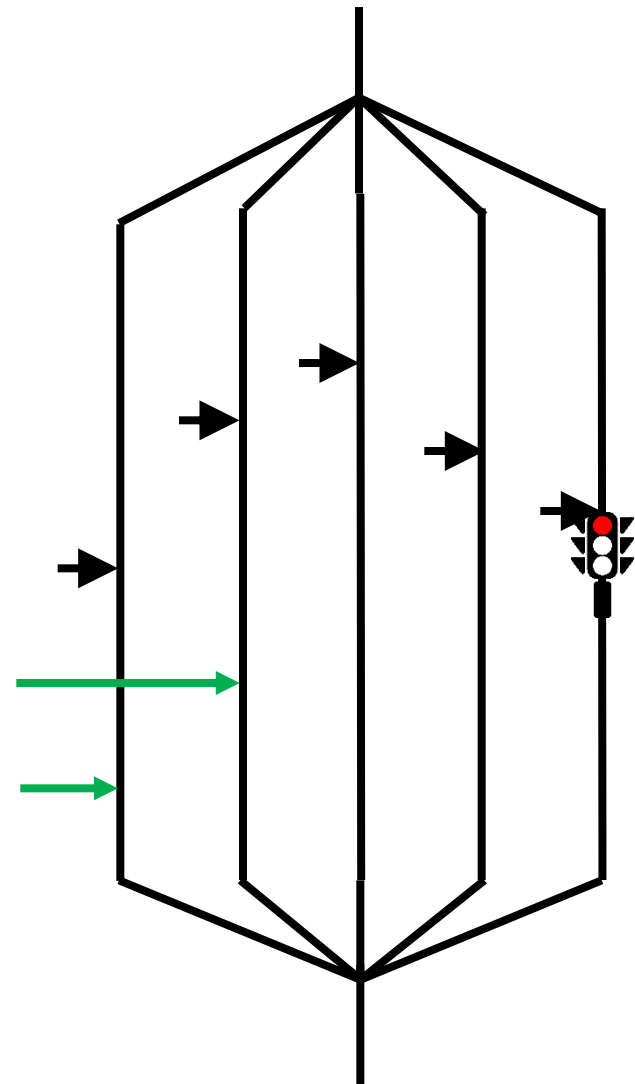
# Semaphore – Signaling



`semaphore.acquire();`

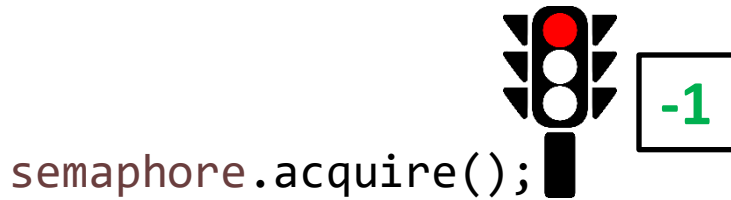
`semaphore.release();`

`semaphore.release();`





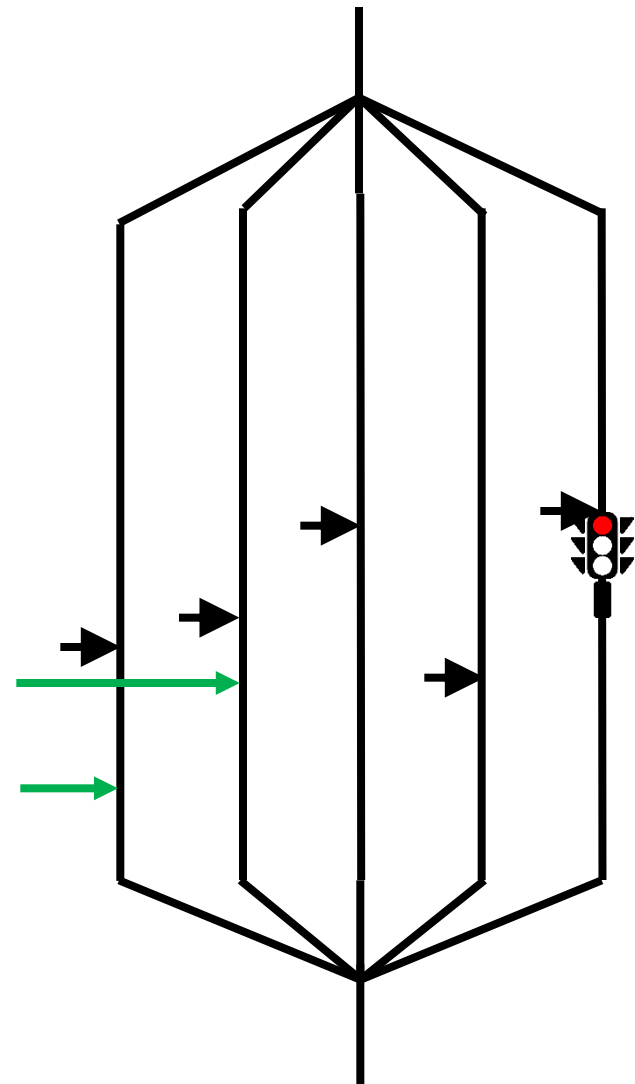
# Semaphore – Signaling



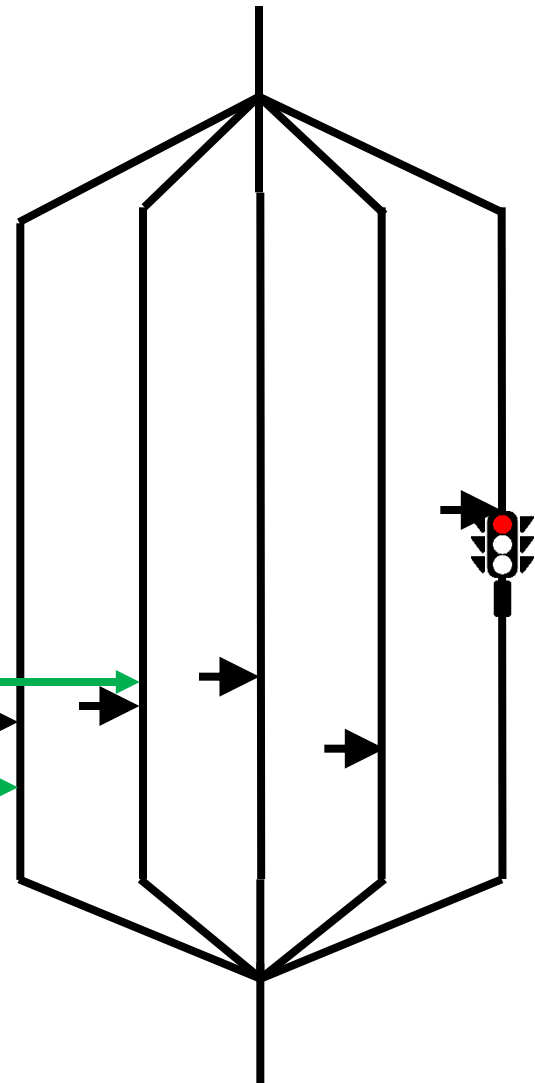
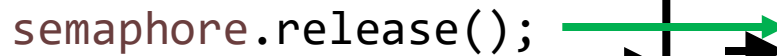
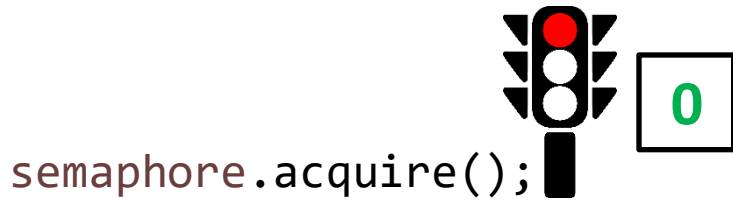
`semaphore.acquire();`

`semaphore.release();`

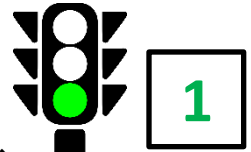
`semaphore.release();`



# Semaphore – Signaling



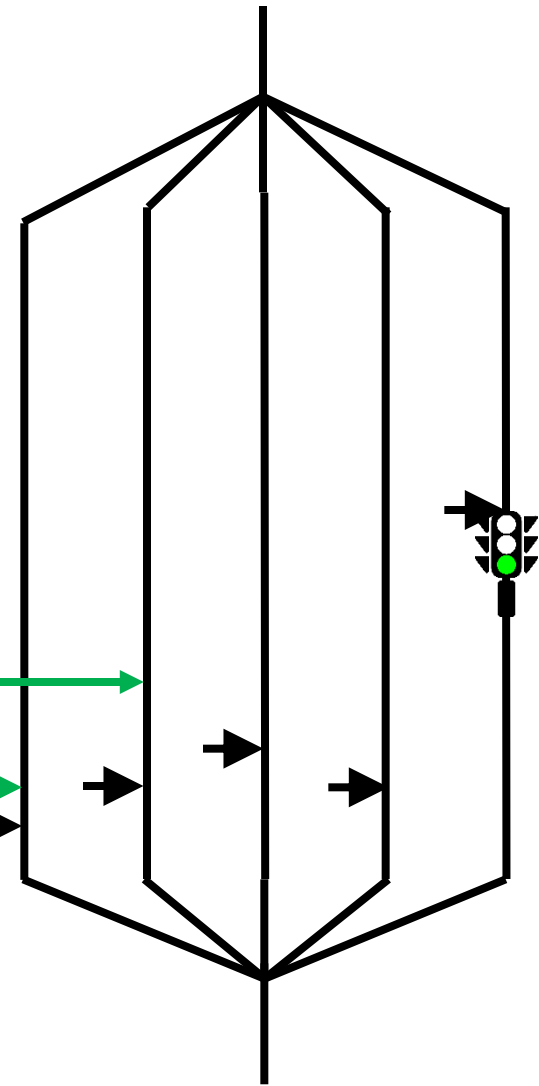
# Semaphore – Signaling



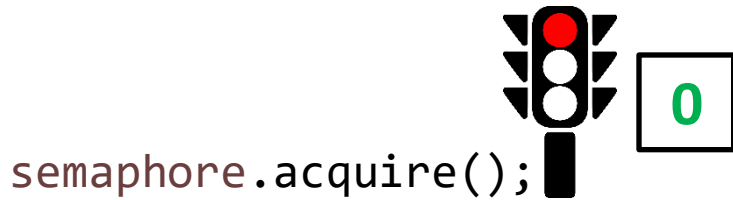
```
semaphore.acquire();
```

```
semaphore.release();
```

```
semaphore.release();
```



# Semaphore – Signaling



`semaphore.release();`

`semaphore.release();`

