Acele işe ~~Şeytan~~ GitHub karışır

Fatih Arslan
Software Engineer @PlanetScale

Kodla 22'
Trabzon

Hakkımda

- Software Engineer **@PlanetScale**
  Öncesinde **GitHub, DigitalOcean, Koding, Pardus (Tübitak) vs …**

- Creator of **vim-go**

- **Go contributor.** Go paketleri (i.e: **color**, **structs**, **hcl**, **gomodifytags**, etc..)

- **Kahve** ve **Tasarım sevdalısı**

Benim bir

sorunum var(*dı*)

1.Bash Script

(ne yanlış gidebilir ?)

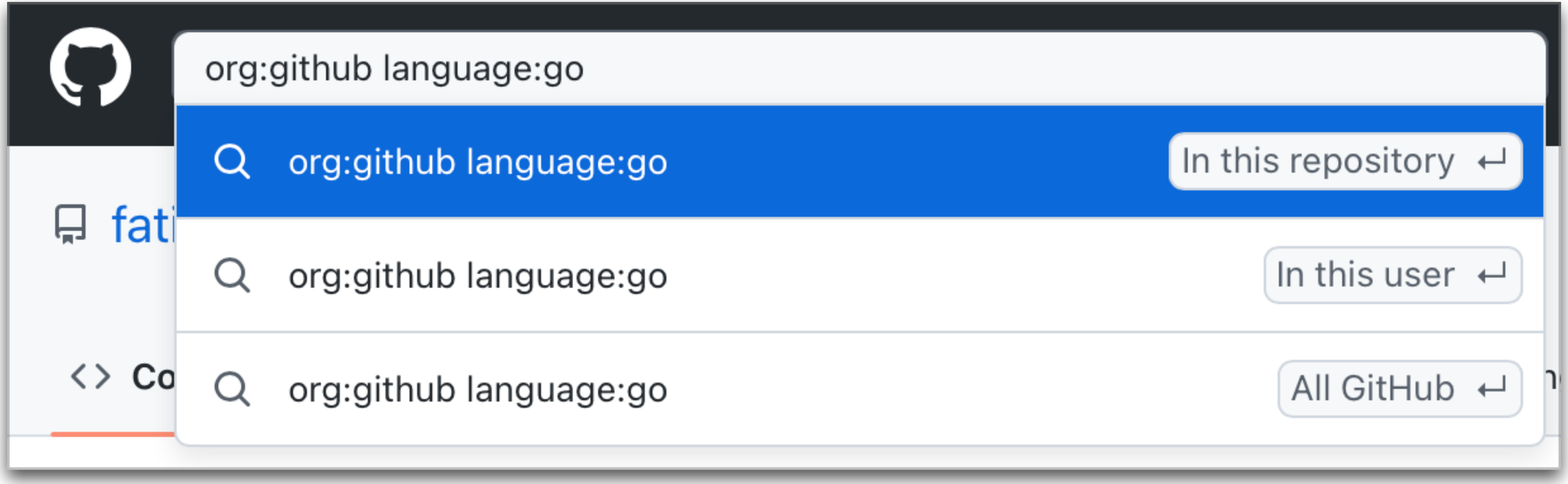
```bash
#!/usr/bin/env bash

repos=("structtag" "vim-go" "gomodifytags")

for repo in "${repos[@]}" ; do
  url="https://github.com/fatih/$repo"
  echo "cloning $url"
  git clone $url
done
```

2. GitHub API

GitHub Search API



org:github language:go

org:github language:go In this repository ↵

org:github language:go In this user ↵

org:github language:go All GitHub ↵

- **org:**github
- **org:**github **language:**ruby
- **user:**fatih **repo:**vim-go
- **user:**fatih **language:**go

Go kullanalım

3. Concurrent

(eşzamanlı)

```
panic: GET https://api.github.com/search/
repositories?
per_page=50&q=org%3Agithub+language%3Ago:
403 API rate limit exceeded for 77.185.235.20.

(But here's the good news: Authenticated requests
get a higher rate limit. Check out the
documentation for more details.) [rate reset in
24s]

goroutine 1 [running]:
main.main()
        /Users/fatih/Code/kodla-talk/4-concurrent-
wait/main.go:23 +0x1b5
exit status 2
```

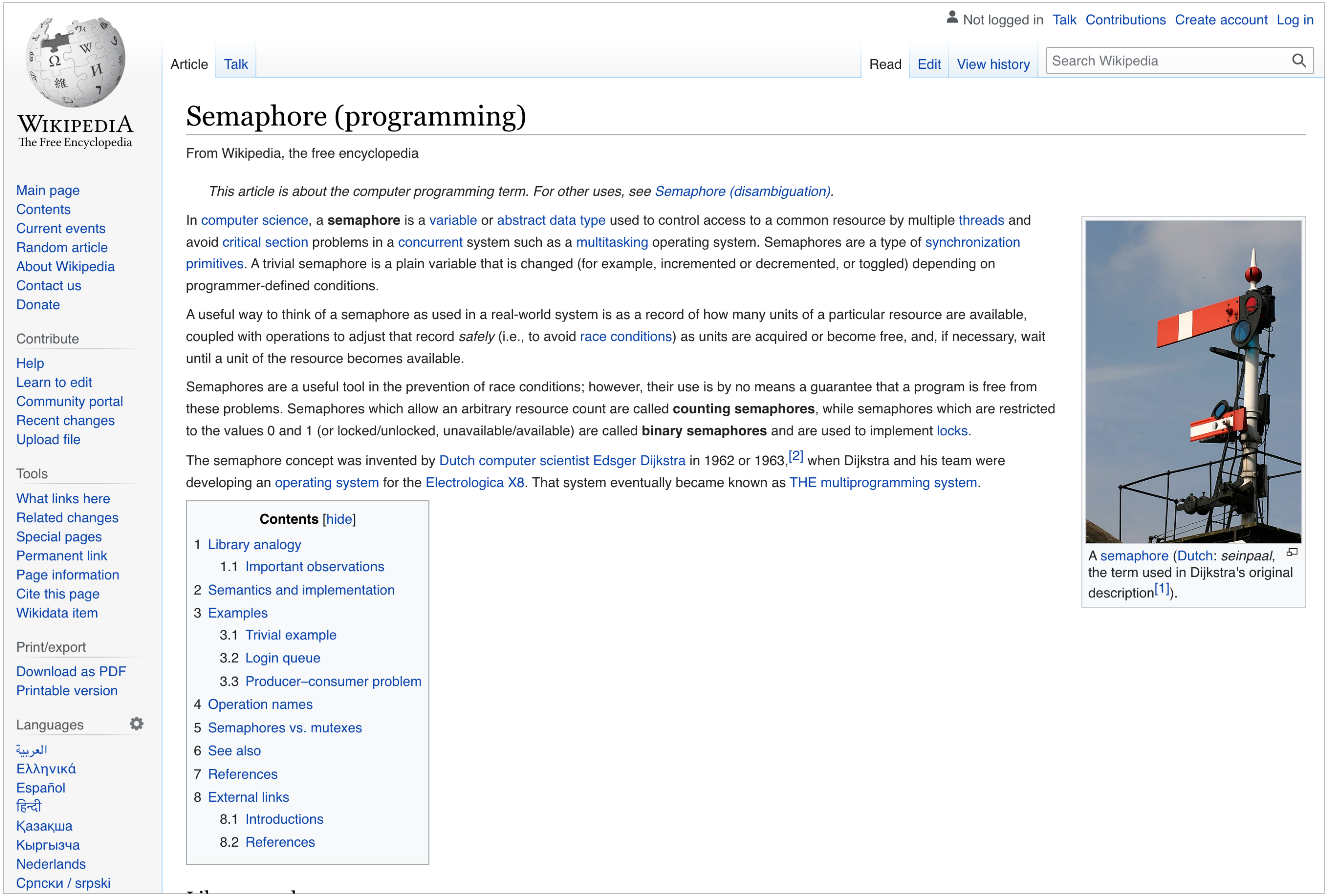
5. Authentication

6. Concurrent

(hâlâ çalışmıyor 😔)

```
err: too many open files
```

7. Semaphore



Semaphore (programming)

From Wikipedia, the free encyclopedia

This article is about the computer programming term. For other uses, see Semaphore (disambiguation).

In computer science, a **semaphore** is a variable or abstract data type used to control access to a common resource by multiple threads and avoid critical section problems in a concurrent system such as a multitasking operating system. Semaphores are a type of synchronization primitives. A trivial semaphore is a plain variable that is changed (for example, incremented or decremented, or toggled) depending on programmer-defined conditions.

A useful way to think of a semaphore as used in a real-world system is as a record of how many units of a particular resource are available, coupled with operations to adjust that record *safely* (i.e., to avoid race conditions) as units are acquired or become free, and, if necessary, wait until a unit of the resource becomes available.

Semaphores are a useful tool in the prevention of race conditions; however, their use is by no means a guarantee that a program is free from these problems. Semaphores which allow an arbitrary resource count are called **counting semaphores**, while semaphores which are restricted to the values 0 and 1 (or locked/unlocked, unavailable/available) are called **binary semaphores** and are used to implement locks.

The semaphore concept was invented by Dutch computer scientist Edsger Dijkstra in 1962 or 1963,[2] when Dijkstra and his team were developing an operating system for the Electrologica X8. That system eventually became known as THE multiprogramming system.

A semaphore (Dutch: *seinpaal*, the term used in Dijkstra's original description[1]).

Contents [hide]

In computer science, a **semaphore** is a variable or abstract data type used to control access to a common resource by multiple threads and avoid critical section problems in a concurrent system such as a multitasking operating system.

Semaphores are a type of synchronization primitives.

A trivial semaphore is a plain variable that is changed (for example, incremented or decremented, or toggled) depending on programmer-defined conditions.

Semaphore ("apparatus for signalling," from Greek *sema* "sign, signal" and *phoros* "bearer") [1] is the use of an apparatus to create a visual signal transmitted over distance.[2][3]

A semaphore can be performed with devices including: fire, lights, flags, sunlight and moving arms.

Semafor nedir?

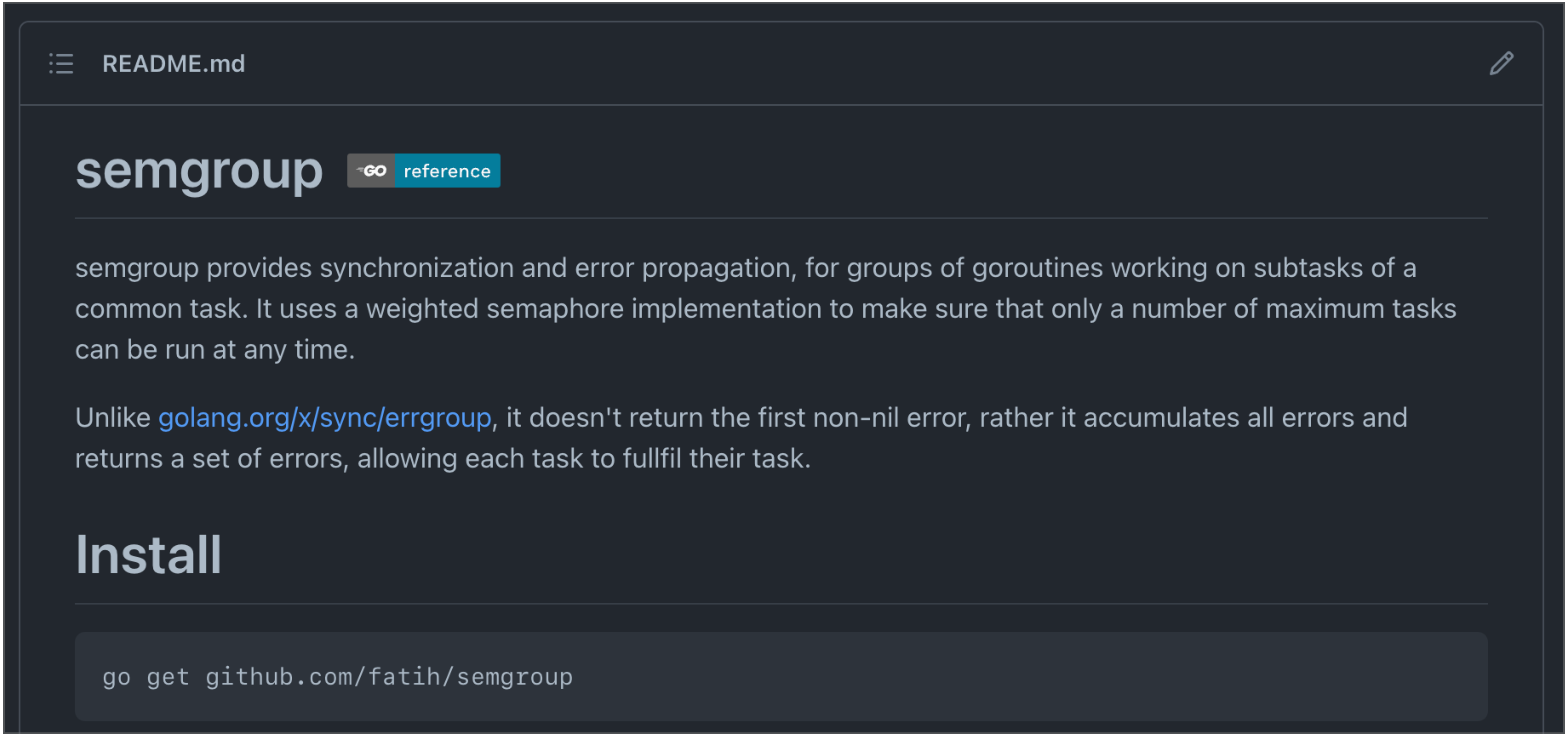
- **Demiryollarında**, gündüzleri mekanik olarak kırmızı bir kolla, gece vakti kırmızı ışıkla işaret veren aygıt.
- İki **gemi** arasında ya da bir gemiyle kıyı istasyonu arasında haberleşmede kullanılan, üç kollu işaret sütunu.

8. Semaphore

(peki hataları ne yapacağız?)

9. Semgroup

*(github.com/fatih/**semgroup**)*



semgroup

semgroup provides synchronization and error propagation, for groups of goroutines working on subtasks of a common task. It uses a weighted semaphore implementation to make sure that only a number of maximum tasks can be run at any time.

Unlike golang.org/x/sync/errgroup, it doesn't return the first non-nil error, rather it accumulates all errors and returns a set of errors, allowing each task to fullfil their task.

Install

```
go get github.com/fatih/semgroup
```

10. İleri Seviye

(daha ne yapabiliriz ?)

Sonuç

Teşekkürler

Fatih Arslan

@fatih

@fatih