Горячая перезагрузка кода в Go

Юрий Насретдинов

nasretdinov@gmail.com
nasretdinov@google.com

Disclaimer

Я работаю SRE в Google, но этот доклад про моё хобби и не связан с моей работой в компании.

Все мнения — мои собственные и не являются мнениями компании Google.

План

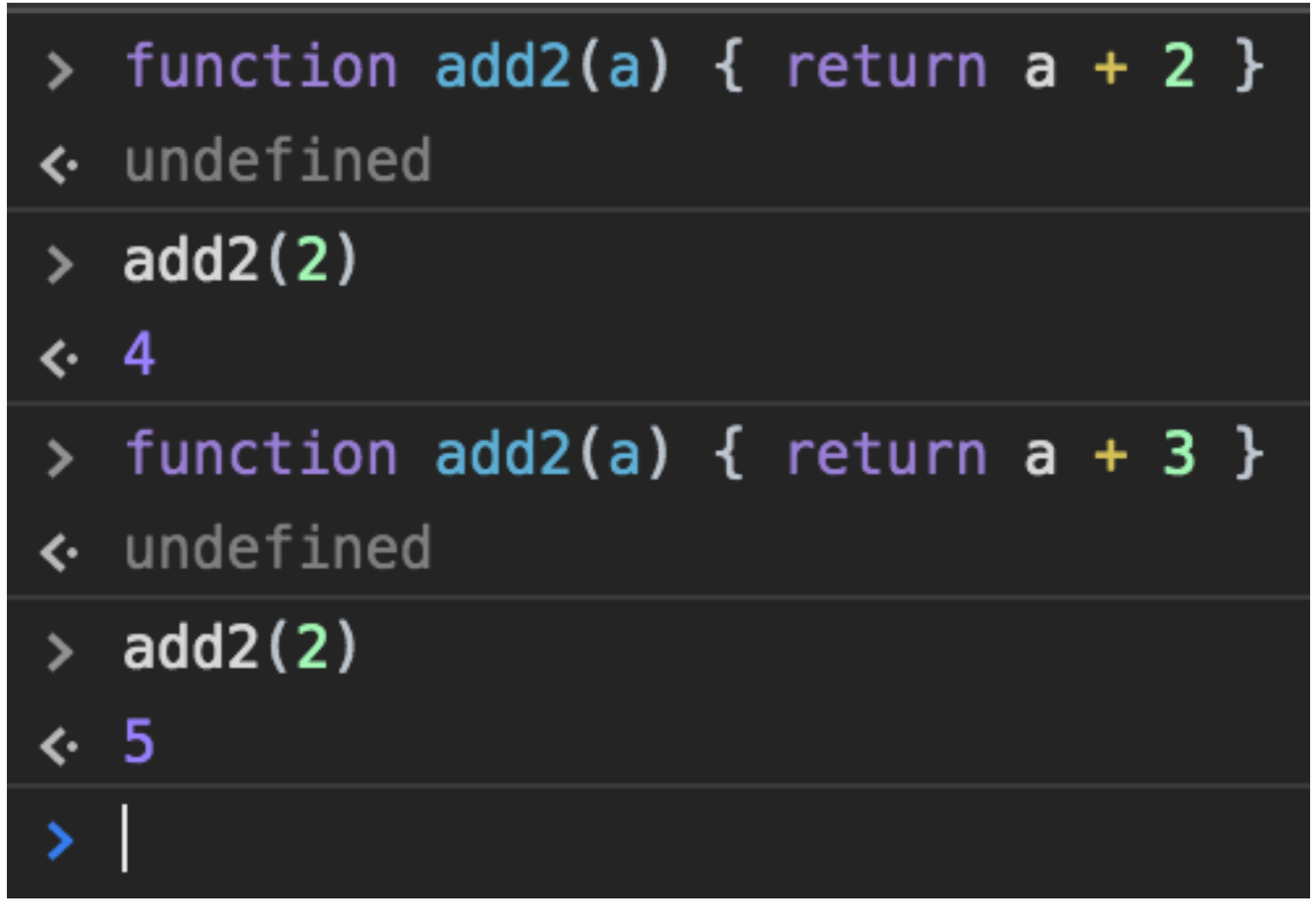
- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Горячая перезагрузка / Live Reload

Горячая перезагрузка / Live Reload



Горячая перезагрузка / Live Reload

Горячая перезагрузка кода — обновление кода программы в процессе её исполнения, без потери данных в памяти

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Зачем?

- Ускорение разработки (stateful приложений)
- ~~Применение патчей в production~~

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Плагины

```
go build -buildmode=plugin
```

Index ▾

type Plugin
  func Open(path string) (*Plugin, error)
  func (p *Plugin) Lookup(symName string) (Symbol, error)
type Symbol

https://golang.org/pkg/plugin/

Плагины

For example, a plugin defined as

```
package main

import "fmt"

var V int

func F() { fmt.Printf("Hello, number %d\n", V) }
```

https://golang.org/pkg/plugin/

Плагины

```
p, err := plugin.Open("plugin_name.so")
if err != nil {
        panic(err)
}
v, err := p.Lookup("V")
if err != nil {
        panic(err)
}
f, err := p.Lookup("F")
if err != nil {
        panic(err)
}
*v.(*int) = 7
f.(func())() // prints "Hello, number 7"
```

https://golang.org/pkg/plugin/

Плагины

- Всегда пакет main
- Нельзя выгрузить из памяти
- Код пакетов должен в точности совпадать

Плагины

- Всегда пакет main
- Нельзя выгрузить из памяти
- **Код пакетов должен в точности совпадать**

Плагины

Живая перезагрузка кода с помощью плагинов невозможна*, расходимся

*** в том плане, что нельзя собрать тот же пакет с другим кодом и потом загрузить оттуда новую функцию**

Плагины

Но можно же и так:

- Скопировать код функции в пакет main.
- Собрать плагин.
- Остается только каким-то образом подменить реализацию существующей функции...

Плагины

```go
package mypkg

import "log"

type T struct {}

func (t *T) Something() {
    log.Printf("Something")
}
```

=>

```go
package main

import (
    "log"
    "github/.../mypkg"
    "github/.../hotreload"
)

func Something(t *mypkg.T) {
    log.Printf("Reloaded!")
}

func Mock() {
    hot.Mock(".../Something", Something)
}
```

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Soft Mocks для Go

- Подмена функций и методов «на лету»
- Путем переписывания $GOPATH (и $GOROOT)

Soft Mocks для Go

```
3   func SomeFunc() {
4       if soft.IsMocked(SomeFunc) {
5           mock := soft.GetMock(SomeFunc).(func())
6           mock()
7           return
8       }
9
10      // usual SomeFunc Code
11  }
```

Soft Mocks для Go

- В тело каждой функции вставляется код в начале, который проверяет, не нужно ли выполнить другой код вместо текущего
- Для этого создается копия $GOPATH с новым содержимым

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Ограничения реализации

```go
package mypkg

import "log"

type T struct {}

func (t *T) Something() {
    log.Printf("Something")
}
```

=>

```go
package main

import (
    "log"
    "github/.../mypkg"
    "github/.../hotreload"
)

func Something(t *mypkg.T) {
    log.Printf("Reloaded!")
}

func Mock() {
    hot.Mock(".../Something", Something)
}
```

Ограничения реализации

- Нельзя добавлять новые функции, константы, переменные
- Можно только менять код существующих функций/методов
- Код обязан использовать публичные типы
- *(решаемо)* Нельзя ссылаться на глобальные типы / переменные из того же пакета
- *(решаемо)* Код должен жить в $GOPATH

План

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Демо

Выводы

- Понятие «горячей перезагрузки» кода
- Зачем?
- Плагины в Go
- "Soft Mocks" для Go
- Ограничения реализации
- Демо
- Выводы

Выводы

- «Живая перезагрузка» кода в Go возможна, но с оговорками
  - *Не переизобретайте Erlang :)*
- Open-Source: https://github.com/YuriyNasretdinov/hotreload
- Экономия по времени сборки будет не всегда
- **Не применяйте в production!**
- Будете применять для разработки?