package 简介

作者：少林之巅

目录

Go的源码组织方式

1. Go通过package的方式来组织源码

`package 包名`

1 **注意：任何一个源码都属于一个包**

2 **作用：代码复用和可读性**

main函数和main包

2. 可执行程序的包名必须为main，并且包含一个main函数

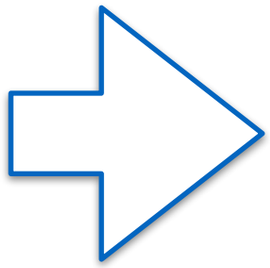
```
package main

import (
    "fmt"
)

func main() {
   fmt.Println(“hello world”)
}
```

main函数和main包

3. package组织案例

```
└── landlords
    ├── dal
    |   ├── db
    |   └── redis
    ├── logic
    |   ├── api.go
    |   ├── client.go
    |   ├── closer.go
    |   ├── cmd.go
    |   ├── common.go
    |   ├── desk.go
    |   ├── errors.go
    |   ├── player.go
    |   ├── player_mgr.go
    |   ├── poker.go
    |   ├── room_conf.go
    |   ├── room.go
    |   ├── room_mgr.go
    |   └── seat.go
    ├── main
    |   ├── home.html
    |   ├── main
    |   └── main.go
    ├── proto
    |   └── proto.go
    ├── README.md
```

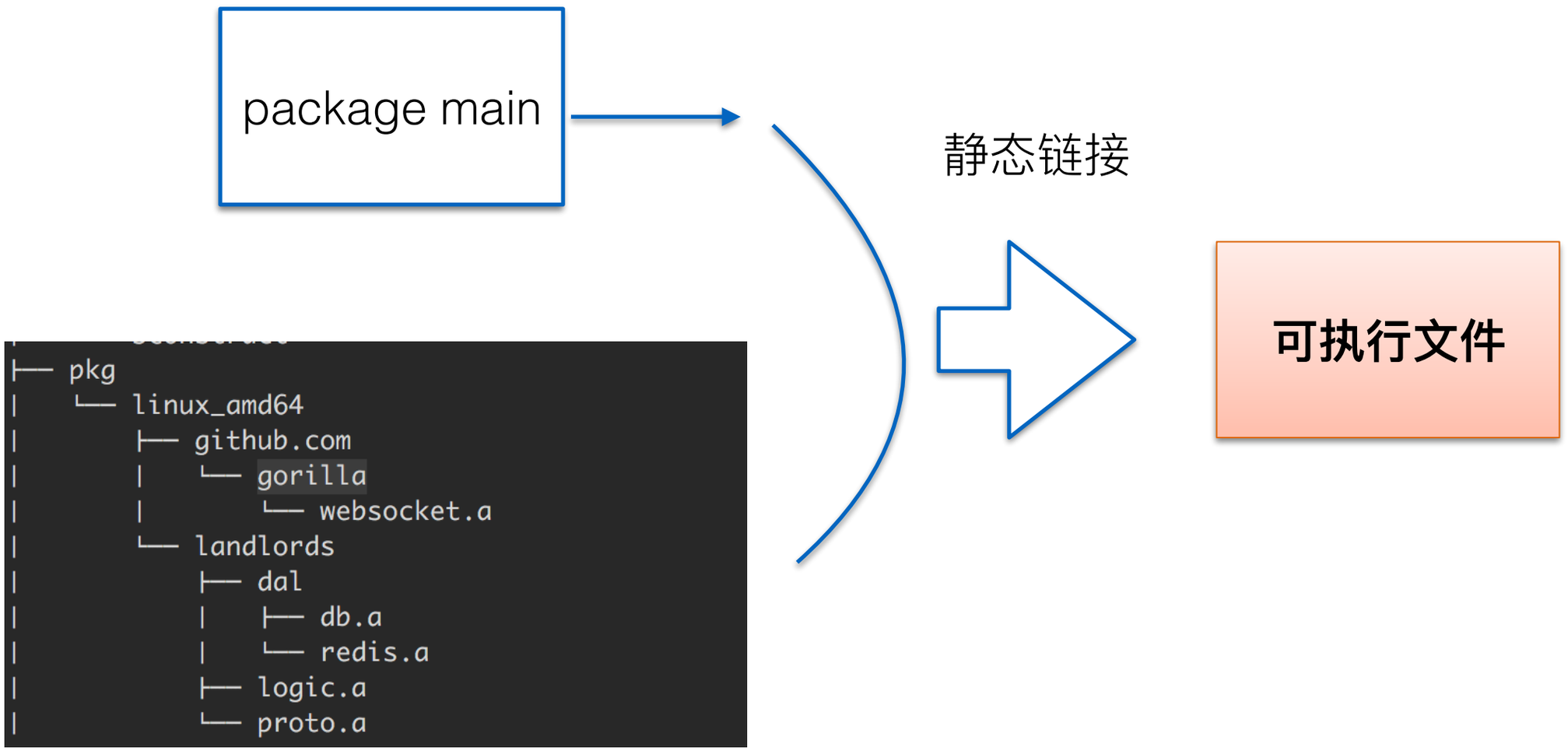
编译



```
├── pkg
|   └── linux_amd64
|       ├── github.com
|       |   └── gorilla
|       |       └── websocket.a
|       └── landlords
|           ├── dal
|           |   ├── db.a
|           |   └── redis.a
|           ├── logic.a
|           └── proto.a
```

main函数和main包

4. 链接过程



自定义包

5. 除了可执行程序之外，用户可以写自定义包，自定义包编译成静态库

```
package calc

import (
    "fmt"
)

func Add(a, b int) int{
   return a+b
}
```

自定义包

6. 导出变量或函数。首字母大写表示可导出，小写表示私有。不能被外部的包访问

```
package calc

import (
    "fmt"
)

func Add(a, b int) int{
   return a+b
}
```

编译命令

7. go run运行go代码， 如果有多个文件，需要把所有文件都写到go run后面

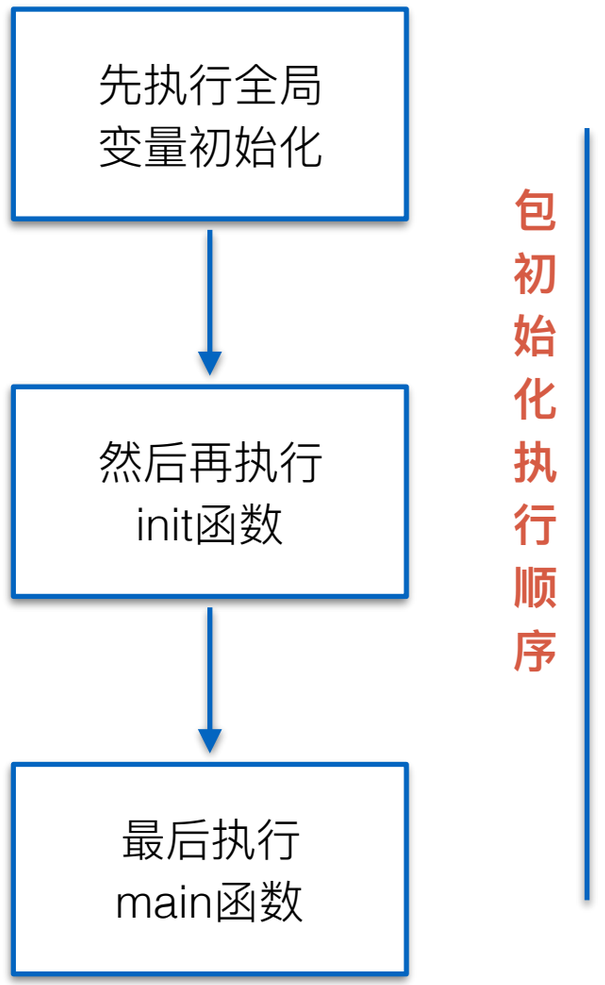
8. go build 编译go代码，如果是可执行程序，默认会在当前目录生成可执行程序，可以使用-o指定可执行程序生成的目录。

9. go install编译go代码，并且把可执行程序拷贝到GOPATH的bin目录，自定义或第三方包会拷贝到GOPATH的pkg目录

init函数

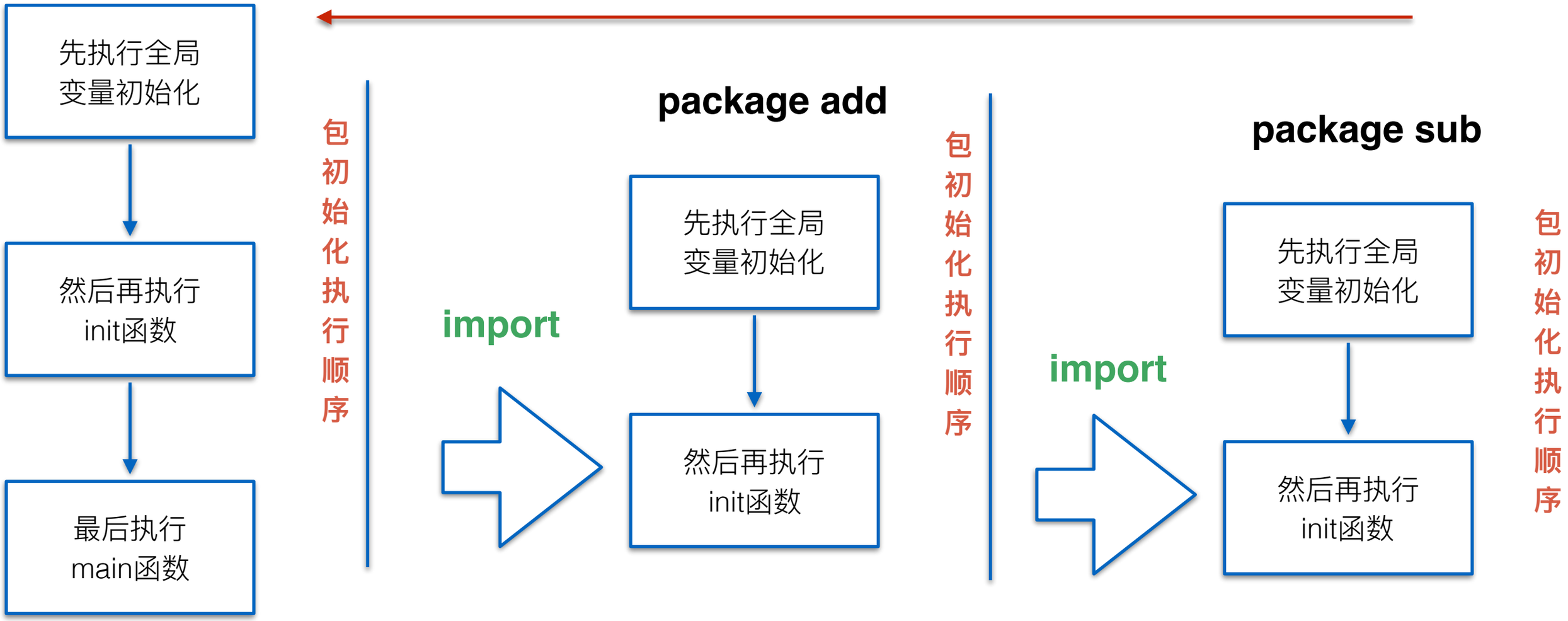
10. 一个包里可以有0个或多个init函数，在程序启动时会被自动调用

```go
package calc

import (
    "fmt"
)

var a int=10

func init() {

}

func Add(a, b int) int{
   return a+b
}
```



init函数

11. 如果一个包import另外一个包



_标识符

12. _标识符的另外一种用法

```
package calc

import (
    “fmt"
    _ “sub”
)

func Add(a, b int) int{
   return a+b
}
```

课后练习

1. 你有50枚金币，需要分配给以下几个人：Matthew, Sarah, Augustus, Heidi, Emilie, Peter, Giana, Adriano, Aaron, Elizabeth。分配规则如下所示：

   a. 名字中包含’a’或’A’：1枚金币
   b. 名字中包含’e’或’E’：1枚金币
   c. 名字中包含 ‘i’或’I’： 2枚金币
   d. 名字中包含’o’或’O’：3枚金币
   e. 名字中包含’u’或’U’：5枚金币

   写一个程序，计算每个用户分到了多少金币，以及最后剩余多少金币?

```
package main
import "fmt"
var (
    coins = 50
    users = []string{
        "Matthew", "Sarah", "Augustus", "Heidi", "Emilie",
        "Peter", "Giana", "Adriano", "Aaron", "Elizabeth",
    }
    distribution = make(map[string]int, len(users))
)
func main() {
}
```