Go 编程要点

Go 1.8

函数是值类型

编译并运行

go run hello.go

只编译

go build hello.go

Go编程实践

# 参考资料：

1 Effective Go

# 工程calc

set GOPATH=f:\go\calc

cd f:\go\calc

mkdir bin

cd bin

go build calc

单元测试

go test simplemath

# 工程 sorter

set GOPATH=f:\go\sorter

go build sorter

go test algorithm/qsort

# 基本语法

:= 表示变量声明和初始化同时进行

多重赋值 **i,j = j,i**

1

## 常量

字面常量

预定义常量

itoa

无类型常量

## 字符类型

byte

rune 代表单个Unicode字符

## 数组和数组切片

**数组是一个值类型，函数参数传递会发生复制**

range 关键字 两个返回值，元素下标和元素值

for i,v := range array{

fmt.Println(i,v)

}

流程控制

条件语句

不允许将“最终的”return语句包含在if else 结构中，否则会编译失败

func example(x int)int{

if x == 0{

return 5

}else {

return x

}

}

## 错误处理

error接口

defer关键字

panic()和recover

## 函数

不定参数

func example(args …int){

for \_,arg := range args{

fmt.Println(arg)

}

}

任意类型的不定参数

func Println(format string,args …interface{}){

}

func MyPrint(args …interface{}){

for \_,arg :=range args{

switch arg.(type){ arg为interface对象，必须配合switch使用

case int:

case string:

default:

}

}

switch

}

匿名函数

func(a,b int,z float64)bool{

return a\*b < int(z)

}

闭包

C++虚基类

解决继承的二义性，菱形继承

# 面向对象编程

接口

类型查询

接口指向对象实例的类型

switch v:=v1.(type)

## 协程

不要通过共享内存来通信，而应该通过通信来共享内存

Go语言编译器不能智能地使用多核优势

协程缓冲机制

c := make(chan int,1024)

全局唯一性操作 sync.Once

使用Go语言开发游戏服务器

json struct

结构体的每一项必须是导出项，key对应的首字母必须为大写

type User struct{

Id int “id”

}

# 工程管理

命名：支持骆驼命名法

## 目录结构

ed

readmacho

calc

|-README

|-AUTHORS

|-LICENSE

|-bin 自动创建

|-pkg 自动创建

|-src 必须包含所有源代码

|-calc

|-calc.go

|-simplemath

|-add.go

|-add\_test.go

spider

## 文档构建

## 单元测试

功能测试

性能测试

# 源码阅读

# 编程练习

1 实现一个分析Mach-O文件符号解析，查看是否包含调试信息

2 实现一个ELF合并工具 Capstone

3 实现一个爬虫

4 实现SOCK5 协议

5 实现HTTP Proxy服务器