Go语言入门到精通

江洲老师云课堂

主讲：江洲老师

Go并发1

目 录

Go并发

1.什么叫Go并发

Go并发

Go 在语言级别支持协程，叫goroutine。

Go 语言标准库提供的所有系统调用操作（包括所有同步IO操作），都会出让CPU给其他goroutine。这让轻量级线程的切换管理不依赖于系统的线程和进程，也不需要依赖于CPU的核心数量。

有人把Go比作21世纪的C语言。
第一是因为Go语言设计简单，第二，21世纪最重要的就是并行程序设计，而Go从语言层面就支持并行。同时，并发程序的内存管理有时候是非常复杂的，而Go语言提供了自动垃圾回收机制。

Go语言为并发编程而内置的上层API基于顺序通信进程模型CSP(communicating sequential processes)。
这就意味着显式锁都是可以避免的，因为Go通过相对安全的通道发送和接受数据以实现同步，这大大地简化了并发程序的编写。

Go语言中的并发程序主要使用两种手段来实现：goroutine和channel。

Goroutine

1.什么是Goroutine

什么是Goroutine?

Goroutine是Go语言并行设计的核心，有人称之为go程。

Goroutine说到底其实就是协程，它比线程更小，十几个Goroutine可能体现在底层就是五六个线程，Go语言内部帮忙实现了这些Goroutine之间的内存共享，执行Goroutine只需极少的栈内存(大概是4~5KB)，当然会根据相应的数据伸缩。

也正因为如此，可同时运行成千上万个并发任务。

Goroutine比thread更易用、更高效、更轻便。

一般情况下，一个普通计算机跑几十个线程就有点负载过大了，但是同样的机器却可以轻松地让成百上千个Goroutine进行资源竞争。

Goroutine的创建

1.Goroutine的创建

Goroutine的创建

只需在函数调用语句前添加 go 关键字，就可创建并发执行单元。

开发人员无需了解任何执行细节，调度器会自动将其安排到合适的系统线程上执行。

在并发编程中，我们通常想将一个过程切分成几块，然后让每个Goroutine各自负责一块工作，当一个程序启动时，主函数在一个单独的Goroutine中运行，我们叫它main goroutine。

新的Goroutine会用go语句来创建，而go语言的并发设计，让我们很轻松就可以达成这一目的。

Goroutine特性

1.Goroutine特性

Goroutine特性

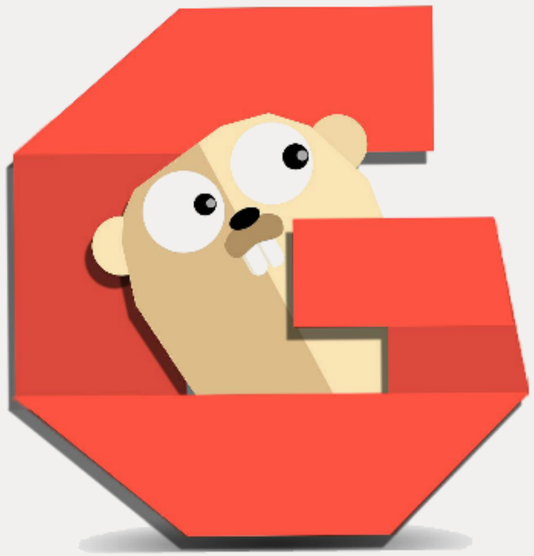
主goroutine退出后，其它的工作goroutine也会自动退出：

```
package main
import (
"fmt"
"time"
)
func newTask() {
   i := 0
   for {
      i++
      fmt.Printf("new goroutine: i = %d\n", i)
      time.Sleep(1 * time.Second) //延时1s
   }
}
func main() {
   //创建一个 goroutine，启动另外一个任务
   go newTask()
   fmt.Println("main goroutine exit")
}
```

程序运行结果：

```
main goroutine exit
成功: 进程退出代码 0.
```

实例演示



Go语言入门到精通

江洲老师云课堂

主讲：江洲老师

感谢您的聆听和观看