

语言规范

- [Close Channels Gracefully in Golang](#): 如何优雅地关闭 channel?
- [Compile-time assertions in Go](#): 编译时断言
- [Why are slices sometimes altered when passed by value in Go?](#): Go 不是按值传递么，怎么 slice 传入后被更改了呢？其实 map 也一样
- [Bit Hacking with Go](#): Go 的位操作
- [Go Range Loop Internals](#): range 内幕
- [Blocks and Scopes in Golang](#): 代码块和作用域
- [Address Alignments in Go](#): 地址对齐
- [Understand Go pointers in less than 800 words](#): 回顾一下指针的概念
- [Another Introduction to Pointers in Go](#): 另一篇关于指针的介绍
- [There is no pass-by-reference in Go](#): Go 中没有引用传值
- [Why golang garbage-collector not implement Generational and Compact gc?](#): 为什么不使用分代或者压缩的垃圾回收器？
- [Modern garbage collection](#): Go 的垃圾回收器
- [Language Mechanics On Stacks And Pointers](#): William Kennedy 写的系列，一共四篇，深入底层
- [Structures: Go struct 介绍](#)
- [Go 调度详解: 调度概览](#)
- [也谈 goroutine 调度器: 也谈谈 Go 的调度器](#)
- [了解 Go 1.9 的类型别名: Go 1.9 中的类型别名](#)
- [Golang Internals Part 2: Nice benefits of named return values](#): 返回值采用命名方式的好处
- [Go Interface 源码剖析: interface 源代码](#)
- [Survey of Rounding Implementations in Go](#): 四舍五入不简单
- [Golang 热更新研究笔记: 热更新笔记](#)
- [go empty interface](#): 空接口
- [理解 go interface 的 5 个关键点: 接口关键点](#)
- [Go 汇编实战的坑: 赞，国人给 golang 提 patch](#)
- [golang 语言编译的二进制可执行文件为什么比 C 语言大](#): 深入分析
- [init functions in Go](#): init 函数不简单
- [Efficient Bit Manipulation in Go 1.9](#): 有效的位操作
- [Learn Go constants—A visual guide](#): 可视化系列
- [Go cheatsheet](#): 小抄
- [defer in go](#): 还有第二部分 part2
- [Using named return variables to capture panics in Go](#): 返回 panic 的错误
- [Go Reflection: Creating Objects from Types](#): 基本类型， part2

其它类型 测试

- [Go advanced testing tips & tricks](#): Go 测试提示和技巧
- [5 simple tips and tricks for writing unit tests in #golang](#): 写测试的 5 个技巧
- [5 Advanced Testing Techniques in Go](#): 5 个测试的高级技术

调试和性能调优

- [Profiling and optimizing Go web applications](#): 优化 web 程序, 使用 profiler
- [go tool trace](#): trace 工具介绍的很少, 这是其中一篇
- [Debugging Go core dumps](#): 使用 core dump 调试
- [Benchmarking Go programs](#): 测试和 pprof, 第二部分
- [GO APP MONITORING: EXPVAR, PROMETHEUS AND STATSD](#): 监控实践
- [一个内部 API 系统的性能优化](#): 国内用户的优化实践
- [Profiler labels in Go](#): 不常用但是有用的技巧
- [diagnostics](#): 官方的介绍更准确, 诊断。
- [Go's work-stealing scheduler](#): rakyll 的介绍
- [3 倍性能的 go 程序优化实践](#): 国内的优化实践
- [Seven steps to 100x faster](#): 100 倍的性能提升
- [A Million WebSockets and Go](#): 百万 websocket 连接和性能提升经验
- [Golang 与系统调用](#): 根据演讲整理
- [Performance comparison of regular expression engines](#): 正则表达式库的性能比较
- [Allocation Efficiency in High-Performance Go Services](#): 有效的内存分配
- [使用 bcc/BPF 分析 go 程序](#): Brendan Gregg 的文章
- [Your pprof is showing](#): pprof, 不想多说了
- [Using the Go tracer to speed up fractal making](#): go tracer
- [Profiling Go](#): 很好 profile 总结
- [The new pprof user interface](#): go 1.10 中已包含新界面
- [High Performance Go](#): Dave Cheney 在 QComSF 2017 上的演讲
- [Go Profiler Internals](#): 所谓的 profile 内幕
- [Optimizing GoLang for High Performance with ARM64](#)

[Assembly](#): 针对 ARM64 的优化

标准库

- [Plugins in Go 1.8](#): 对 Go 1.8 增加的 plugin 包的速览
- [TCP/IP Networking](#): 网络编程入门
- [Linux, Netlink, and Go—Part 1: netlink](#): 访问 linux netlink, 还有第二篇和第三篇
- [Port Forwarding with Go](#): 端口转发
- [Writing Modular Go Programs with Plugins](#): 插件
- [Advanced command execution in Go with os/exec](#): 学习 os/

exec

- [expvar in action](#):知道这个标准包吗，用过吗？
- [Golang 互斥锁内部实现](#): Go的锁实现
- [Go unsafe 包之内存布局](#)

&version=12020810&nettype=WIFI&fontScale=100&pass_ticket=Dip3QhAoFuCXmvNUJBCqwg2%2FTTrorNO8SzZpbc%2FtgnD4%3D):

unsafe 的一些函数

- [Go 1.9 sync.Map 揭秘](#): sync.Map的实现揭秘
- [Basic AST Traversal in Go](#): ast,做工具和代码分析的都需要
- [A Go Programmer's Guide to Syscalls](#):Liz Rice的演讲
- [Pitfalls of context values and how to avoid or mitigate them in Go](#): Context的陷阱以及如何避免它

- [The 30 Most Popular Go String Functions](#): 最常用的 strings 的 30 个方法

- [Serving HTTPS](#): 应用 https
- [How To Trust Extra CA Certs In Your Go App](#): 使用外部的 CA
- [GOLANG 中 time.After 释放的问题](#): time.After使用的时候注意

事项

- [The new kid in town—Go's sync.Map](#):sync.Map的性能
- [The ultimate guide to writing a Go tool](#): Go 工具开发终极手册, gomodifytags 的作者

- [Streaming IO in Go](#): io
- [Things to know about HTTP in Go](#): Go HTTP
- [Important interfaces that every Go developer should know](#):

重要接口

- [Go Walkthrough: fmt](#): fmt 详解
- [Context API explained](#):context 详解
- [Basic AST Manipulation in Go](#):还是 ast
- [In-depth introduction to bufio.Scanner in Golang](#):深入介绍

bufio.Scanner

- [Go Slice vs Maps](#):性能比较
- [Introduction to bufio package in Golang](#): bufio 介绍
- [TCP Socket Implementation On Golang](#): 从外部推断内部实现
- [Understanding Go programs with go/parser](#): 带你学习 go/

parser

- [unsafe.Pointer and system calls](#):理解 unsafe.Pointer 和系统调

用

- [Using Go Templates](#):使用模板

编程模式

- [Functional Options Pattern in Go](#): 介绍了参数以函数子的方式实现, 更出名的是 Rob Pike 提出的这种模式, 以及 Dave Cheney 的进一步介绍。
- [A Story of a Fat Go Binary](#): 使用 gofat 给 Go 程序瘦身
- [a size profiler for binaries](#):瘦身瘦身, google 出品

- [How to ship production-grade Go](#):开发产品级的程序
- [Go programming language secure coding practices guide](#):安全代码指南
- [Error handling patterns in Go](#): Go的错误处理总是让人困惑
- [Alternative Patterns for Method Overloading in Go](#): Go的方法重载的可选实现
- [Go, without package scoped variables](#): 不推荐包级别的变量, 但是怎么做呢
- [Don't defer Close\(\) on writable files](#): defer 会忽略错误
- [Go Patterns](#): Go 设计模式
- [Golang Concurrency Tricks](#):channel 和 goroutine 技巧
- [go make 和 new 的区别](#): 区别
- [Context isn't for cancellation](#): Context 目的不是为了 cancel
- [Inject build-time variables with Golang](#):
- [go advices](#): Go 建议清单
- [Error handling in Upspin](#): Rob Pike 的实践
- [Dynamic JSON in Go](#): 使用 *json.RawMessage 和 interface{}

开发技巧

- [Adding custom data to Go binaries at compile time](#): 编译时给二进制文件加点料
- [Go Quickstart Tips for Experienced Programmers](#): 给经验丰富的程序员的一些 Go 的小贴士
- [Gracefully terminate a program in Go](#):优雅滴关闭程序
- [Golang 中使用 JSON 的小技巧](#): JSON 使用的小技巧
- [How to find out which Go version built your binary](#): debug 发现 Go 版本
- [Generating good unique ids in Go](#):产生唯一的 ID
- [Fencing off Go Applied - A Practical Look at a Go Research Paper](#): 使用 dingo-hunter 和 Gong 编译时检查死锁和 channel
- [Go Tooling in Action](#): Go 开发时的一些工具介绍
- [How to use conditional compilation with the go build tool](#):条件编译
- [Strace in 60 lines of Go](#): 实现 strace
- [3 ways to iterate in Go](#):三种方式遍历
- [RustGo: Calling Rust From Go with Little Overhead](#):高效的从 Go 调用 Rust
- [Version Constraints and Go](#):指定的 Go 版本才能编译
- [6 Go Tips You Should \(probably not\) Use](#): 6 个 Go 小技巧, 但是你不应该在代码中使用, :)
- [15 Lessons in Golang](#):15 个经验教训
- [Using functional options instead of method chaining in Go](#): 使用函数式参数, 而不是链式操作
- [合理配置 GOMAXPROCS 提升一倍以上的性能](#): 对于 I/O 密集型的程序, 建议设置 GOMAXPROCS 大一些
- [Lint your #golang code like a mad man!](#): 像牛人一样编写代码

- [Rate Limiting Service Calls in Go](#):限流
- [Sending HTML email using Go](#):发送html邮件
- [Go Assembly 示例](#): Go使用汇编的例子
- [go web examples](#):web开发例子
- [Handling CORS In A Golang Web Application](#):CORS处理
- [Understanding Go panic output](#): 理解 Go 的 panic 输出
- [Go 语言中实现基于 event-loop 网络处理](#):不同的思路
- [Generating Go structs from JSON](#):5个json to gostruct的工具
- [go 语言死循环分析](#):当然不是go的bug,看看go的[讨论](#)
- [Access unexported fields in golang/reflect](#): 访问unexported的字段
- [Why is port a string and not an integer](#): URL.Port为什么是字符串而不是整数?
- [golang: How the select worked when multiple channel involved](#):使用伪随机
- [How to reuse request body of *http.Request](#): request.Body 读完就不能再多了, 如何重用?

面试题

- [golang 面试题](#): 诺唯总结的11个面试题, [解答](#)
- [Golang 精编 100 题](#):选择题和填空题, 试试看

第三方库

- [Infinite Sets in Go](#): 把 set 实现玩出花
- [awesome-go-storage](#): 100 多个 Go 语言实现的存储引擎, 强不强大?
- [Learn Web Programming in Go by Examples](#): 类似 gobyexamples,这次奉上的是关于web开发的
- [Visualizing Concurrency in Go](#):可视化 goroutine
- [Network Protocol Breakdown: Ethernet and Go](#): 学习底层网络协议处理
- [再谈谈获取 goroutine id 的方法](#): goroutine ID 的获得, hacker 方式
- [Listen gRPC and HTTP requests on the same port](#): grpc 和 http 共用端口
- [gRPC-Go performance Improvements](#): gRPC 近期的性能优化
- [Building Blockchain in Go](#):使用 Go 编写区块链程序系列
- [Minimal Perfect Hash Functions](#):完美的 hash 函数, 极小

应用

- [今日头条 Go 建千亿级微服务的实践](#): 今日头条的微服务

版本新特性

- [5 things to watch in Go programming in 2017](#): GO 1.8 新的特性介绍
- [Go 1.8](#): 2016 最后一天写的, 也算做 2017 年的文章吧
- [The State of Go - February 2017](#): 2017 对 Go 的综评, 来自 Go 团队
- [What is new in Go 1.8](#): what's new?

- [Logging, interfaces, and allocation](#): Go 1.9 中的一点优化
- [HTTP/2 Server Push](#): http2 push 功能
- [The State of Go - May 2017](#): The State of Go 系列
- [What's new in Google's Go 1.9](#): what's new
- [Toward Go 2](#): 目标 Go2
- [Go 1.9 中值得关注的几个变化](#): what's new in go 1.9
- [improve timers scalability on multi-CPU systems](#): go 1.10 提升 timer 性能
- [Go 1.10 Release Notes](#): go 1.10 先睹为快
- [Golang 1.10 前瞻](#): 新特性