TURING 图灵原创

# Go 并发编程实战

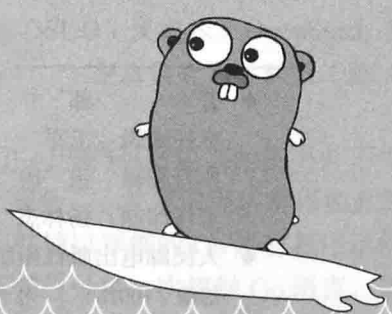
郝林◎著

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

# Go 并发编程实战

郝林◎著



人民邮电出版社
北京

**图书在版编目（CIP）数据**

Go并发编程实战 / 郝林著. -- 北京 : 人民邮电出版社, 2015.1
（图灵原创）
ISBN 978-7-115-37398-4

Ⅰ. ①G… Ⅱ. ①郝… Ⅲ. ①程序语言－程序设计
Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2014)第246517号

## 内 容 提 要

本书全面介绍了Go语言的特点、安装部署环境、工程规范、工具链、语言语法、并发编程模型以及在多个编程实战中的应用，重点阐述了Go语言并发编程模型和机制。本书共分为四个部分，介绍了Go语言编程环境搭建、Go语言基础编程、Go语言并发编程方法及其原理，以及使用Go语言开发的应用系统的案例讲解。

本书适用于有一定计算机编程基础的从业者以及对Go语言编程感兴趣的爱好者，非常适合作为Go语言编程进阶教程。

◆ 著　　郝　林
责任编辑　王军花
执行编辑　张　霞
责任印制　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：800×1000　1/16
印张：35.75
字数：845千字　　2015年1月第1版
印数：1－3 000册　　2015年1月河北第1次印刷

定价：89.00元

**读者服务热线：(010)51095186转600　印装质量热线：(010)81055316**
**反盗版热线：(010)81055315**
**广告经营许可证：京崇工商广字第0021号**

# 专家推荐

“并发编程的支持无疑是 Go 语言最大的亮点。但是，尽管 Go 语言大幅降低了并发编程的门槛，但至今大部分开发者对如何运用该语言编写高并发程序的认知仍然有限。我很高兴能有一本专门探讨 Go 语言并发编程的书。《Go 并发编程实战》这本书对 Go 语言并发编程的探讨之深入、讲解之细腻是它的一大亮点。同时，这本书也非常适合作为 Go 语言的入门教材，即便是对 Go 语言了解不深甚至从未接触的人也能从中获益。另外，书中的示例也非常有价值，它们贴切地展现了用 Go 语言进行编程的方法和技巧。总之，《Go 并发编程实战》是一份难得的 Go 语言学习资料。”

——许式伟，七牛云存储 CEO

“Go 语言作为优秀的开源编程语言，已逐渐成为云计算时代的必学语言之一。《Go 并发编程实战》不但对基本的 Go 语言编程方法和技巧进行了深入的阐释，还独树一帜地对 Go 语言的内部机制和原理进行了清晰的描述。这些都是学好和用好 Go 语言的极佳资料。推荐 Go 语言爱好者以及对 Go 语言感兴趣的技术人员都阅读这本书。”

——杜玉杰，中国 OpenStack 社区（COSUG）发起人，OpenStack 基金会董事，企业级云计算联盟（ECA）副秘书长

“Go 语言是服务端编程领域非常热门的语言，市面上关于 Go 语言的图书并不少见，但都没有像《Go 并发编程实战》这样，把 Go 语言最精髓的部分——并发编程讲解得如此深入浅出，明白透彻。本书系统地梳理了并发编程的概念和原理，并辅以详细的 Go 语言程序示例，非常容易让读者对并发、线程、信号等概念有清晰的理解。不管你是第一次接触 Go 语言，还是已经非常熟悉它了，如果想了解 Go 语言更多的技术内幕，这本书都值得仔细研读，相信读者能够从中受益匪浅。”

——郭理靖，京东云平台开放云事业部总监

“Go 语言从诞生之日起就充满了争议，但是社区对它的热情却越来越高，今年 InfoWorld 最佳的开源项目 Docker 就是用 Go 写的，可见 Go 闪烁着越来越耀眼的光芒。郝林的这本书非常注重对 Go 编程细节的清晰阐释，这在国内原创技术书中是不多见的。书中示例选取精心得当，深入浅出，完全没有那种看完仍需要参悟的感觉。这是一本 Go 学习者真正需要的图书。”

——程显峰，蓝海讯通 COO，《MongoDB 权威指南》译者

“Go 语言之所以被称为 21 世纪的 C 语言，不仅在于它精简的语法和高效的开发，更在于它具有原生支持和易于使用的高并发的特性。而越是简单的技术就越能够生成千变万化的组合，想要用到极致，需要对它有深刻的见解。在《Go 并发编程实战》一书中，作者由浅入深地对 Go 并发技术进行了剖析，并辅以翔实的案例，让读者真正了解和掌握 Go 并发编程，成为多核时代和云计算时代的开发尖兵。”

——陈佳桦，Go 语言签约讲师，Gogs 项目创始人

# 推 荐 序

我很幸运，在三年前就开始接触 Go 语言。由于那时候资料匮乏，我基本上是通过读取官方的源码包学习过来的。那个时候官方有一个三天入门系列，基本上花几个小时就可以完成 Go 语言的入门，后面就是靠自己在 Go 源码包中不断地深入学习。由于我之前是 Web 开发者，所以我就从本职工作出发写了《Go Web 编程》一书。它主要介绍了 Go 语言如何与 Web 开发结合起来，只花了很小的篇幅去介绍 Go 语言本身。至今已经过去了两年多的时间。在之后的这段时间里，我自己也在不断思考，是不是需要再写一本实战类的图书介绍 Go 语言。让我很惊讶和兴奋的是，郝林赶在我之前写出了这本《Go 并发编程实战》。

郝林的《Go 并发编程实战》一书不仅清楚地解释了 Go 语言的各个知识点，而且包含了很多案例和高层次的解读，还阐述了很多软件工程方面的设计和开发技巧。这本书最大的看点在于并发编程，这也是 Go 语言最大的特色。作者花了大量篇幅详细介绍了 Go 并发编程的核心要素——Goroutine 和 Channel 的概念、原理、基本用法和高级技巧，以及编写并发程序的过程中对各种同步工具的运用等问题。在讲述这些知识的过程中，作者还展示和细致讲解了各种各样的代码实例，尤其是包含了像载荷发生器和网络爬虫框架这样的实用程序。这对读者真正理解上述知识点是非常有帮助的。从基础知识到高级应用，再到实战，作者如此缜密的构思真是完美啊。这本书不仅让你知其然，而且还让你知其所以然。在我通读全书之后，不禁感慨作者的阐述是如此地全面和夯实。因此，这本书不仅适合新手入门 Go 语言，而且让我们这样有几年 Go 语言编程经验的读者也受益匪浅。

这本书围绕着并发编程做了大量的介绍，但前几章的编程基础也介绍得相当详细。我们知道，Go 语言的语法很简洁，关键字只有 25 个，但是表达能力超强。作者通过 5 个章节把 Go 语言的语言细节介绍得非常清楚，而且还介绍了很多底层的实现细节，贯穿于语言层面和源码层面，从而让读者对 Go 语言的实现有更加深刻的理解。我们学习的过程一般都是先进行基础知识的学习，接着开始动手写代码。这本书就是按这样的顺序编写的，从知识学习到实战应用的构思非常好，也让我学习到了很多知识，非常感谢作者能写出这样的一本书来。如果在三年前就有这样一本书的话，我相信在它的帮助下我可以更加深入地理解 Go 语言，而不需要天天深埋在源码中，研究各种设计思路和语言细节，这样也许我就有更多时间去写出更多的开源项目。作为一名 Go 语言程序员，我们需要不断地深入学习 Go 的各种细节，这样才能使用它编写出正确和高效的程序。而此书对于 Go 语言细节的讲解非常透彻，还通过各种实例演示了其使用方法及开发技巧。这是一本深度和广度俱佳的 Go 语言的实战图书。我在此郑重推荐给每一位学习 Go 语言的读者。

建议读者不仅要细读这本书，而且还要深入理解作者给出的一些实例，这样才能掌握 Go 语言的设计思想。

最后，感谢郝林邀请我写作这篇推荐序，能够为这样一本书写序，是我莫大的荣幸。

谢孟军

2014 年 10 月 20 日

# 前　言

很高兴你能选择这本书。希望这本书能够让你成为 Go 粉。很多 Go 语言爱好者都喜欢称自己是 Gopher。

Go 编程语言（或称 Golang，以下简称 Go 语言）是云计算时代的 C 语言。它的诞生是为了让程序员有更高的生产效率，并让程序在有并行计算支持的环境下更快速地运行。2009 年 11 月 10 日，Go 语言正式成为开源编程语言大家庭的一员。在本书截稿之时，它的最新版本是 1.3。因此，本书的内容和相关代码将会围绕 Go 语言的 1.3 版本展开。

这里所说的开源是指开放源代码。开源不仅仅意味着分享和免费。对于软件开发人员来讲，它更多的是代表着一种协同工作的方式。它可以使不同公司、不同城市甚至不同国家的技术爱好者们聚集起来，以共同分享和使用源代码的方式协同完成一个任务或目标。开源的意义并不在于目标大小，而在于对某个技术领域、某个应用行业甚至整个产业的启示和推动力。开源的编程语言会更多、更好地凝聚众人的力量，并使它更快地奔向成功。

Go 语言每半年就会发布一个大版本升级（比如从 1.1 升级到 1.2、从 1.2 升级到 1.3，等等）。这样的更新速度是很多其他编程语言望尘莫及的。其中，开源无疑起到了很大的推动作用。

Go 语言虽然在 2009 年才真正成为开源编程语言，在 2012 年才发布第一个正式版——1.0 版本，但是它很早就进入了中国软件开发者的视野。这不单单是因为它是由几位教父级软件开发先驱开发、由 Google 公司发布的开源编程语言，更是因为它是一门拥有诸多先进特性、拥有高性能和生产效率、为云计算时代和软件工程而生的多范式编程语言。Go 语言虽然年轻，但已经足以用于软件生产。

目前来看，Go 语言在中国发展的最大不足就是其开源社区非常松散，爱好者不论是在地理位置上还是在互联网络中都比较分散。当然，这也是由于 Go 语言的年轻所导致的。目前，在中国没有一个可以主导语言爱好者学习和交流的 Go 语言社区，语言爱好者的数量也相对较少。它不像 Python 那样，在中国有一个统一的、分支众多的语言爱好者联盟，Python 语言爱好者可以很轻易地找到志同道合的组织和同行，并积极地进行分享和交流。Go 语言在这方面有所缺失的很大一部分原因是在中国使用 Go 语言的软件开发者太少。因此，我非常希望能通过本书使更多的同行了解和使用 Go 语言，并用它来创造更多的价值。我也希望大家能够协力共建中国的 Go 语言技术社区。

通过本书，我会带领大家进入 Go 语言的世界，领略 Go 语言的魅力，为大家打开一扇门，也力求使大家对使用 Go 语言开发软件产生更多的兴趣。

## 本书结构

本书共分为四个部分。

第一部分，将会带领你进入 Go 语言的世界，让你对 Go 语言有个基本的了解，这一部分包含两章。

第 1 章，会简要介绍 Go 语言为大家带来的优秀特性和新鲜活力，以及与其他主流编程语言相比的优缺点。

第 2 章，将对编写和运行 Go 语言程序所需的所有前期工作进行说明。这包括安装和配置方法、工程结构以及标准命令。通过对这些知识的学习和实践，相信读者就可以为编写 Go 代码和建立 Go 项目做好准备了。

第二部分，会详细阐述 Go 语言的基础编程知识，这一部分包含了 3 章。

第 3 章，首先会介绍 Go 语言中的关键字、运算符、类型、表达式等最基本的概念。掌握这些知识，读者可以了解到 Go 语言在代码构建方面的言简意赅。然后，展示 Go 语言基本数据结构及其主要操作方法和技巧。Go 语言的基本数据结构的操作方式是非常脚本化的，使用起来也非常简单和方便。最后，我将加入一些相对高级一点的主题，比如基本数据结构的初始化方式、可比性与有序性，以及类型转换方式等，从中我们会看到 Go 语言的设计者们在灵活性和简洁性之间做出的取舍。

第 4 章，我们一起来学习 Go 语言的流程控制方法。这些方法已经足以体现出 Go 语言的先进和强大（当然，我们在后面还会看到更棒的特性）。这一章不但会介绍 Go 语言中基本的逻辑控制方法，还会深入阐述 Go 语言特有的程序编写方式，以及这些特有方式所体现出的 Go 语言程序设计哲学。

第 5 章，将详细介绍 Go 语言程序的测试和文档的编写。

第三部分，将集中笔墨讲解 Go 语言中最强悍的部分——并发编程的概念和知识，这一部分同样包含了 3 章的内容。

第 6 章，先简要介绍与多进程编程和多线程编程有关的知识。这些知识会作为理解 Go 语言并发编程模型的先导内容展现给大家。看过以后，大家应该可以对并发编程有一个比较清晰的理解。同时，还会指出以内存共享为基础的并发编程方式所带来的一些问题。最后，通过剖析多核时代（基于多 CPU 核心的计算时代）的并发编程需求，引出 Go 语言先进的并发编程模型。大家会看到 Go 语言为之做出的努力。

第 7 章，会向大家展示 Go 特有的编程要素——Goroutine（也可称为 Go 程）的魅力。这一章不但会介绍怎样利用 Goroutine 编写可并发运行的程序，还会深入阐述 Goroutine 背后的运作机理和执行过程，使读者知其然更知其所以然。此外，还会介绍标准库的 `runtime` 代码包中与 Goroutine 相关的一些 API。这一章的另一个重点是对 Go 语言并发编程中不可或缺的部分——Channel 的介绍。Channel 与 Goroutine 堪称绝配，它的用法相当灵活。这一章在介绍了 Channel 的基本概念和使用规则之后，还会介绍多种 Channel 的实际用法，并通过丰富的示例来说明每种使用方法和技巧。最后，还描述了标准库的 `time` 代码包中的 API 是如何通过 Channel 来实现其

强大功能的。

第 8 章，将是对 Go 语言提供的传统同步和互斥方法的介绍。这包括锁、条件变量、原子操作、WaitGroup、临时对象池，等等。这些同步方法的提供者大都是标准库的 sync 代码包。虽然 Go 语言官方并不推荐使用这些方式来控制和协调 Go 语言编写的并发程序，但不容忽视的是，它们确实在一些应用场景中是简单有效的。

第四部分仅包含一章，即第 9 章。

第 9 章，包括了一个基本囊括本书前三个部分所涉及的全部概念和知识的完整示例。我会带领大家一步一步地编写这个示例。在这个过程中，我会进一步阐述 Go 语言的哲学和理念，以及我在多年编程生涯中的一些见解和感悟。大家可以通过对这个示例的学习来巩固我们之前讲到的 Go 语言知识，并加深对 Go 语言并发编程的理解。

在本书的附录中，将向大家简单介绍目前在国内外比较活跃的一部分 Go 语言开源项目和 Go 语言社区。这会使大家学习 Go 语言的道路变得更加顺畅，也有利于大家找到志同道合的朋友。

## 目标读者

原则上来讲，任何对计算机编程和 Go 语言感兴趣的人都可以阅读本书。但是，当你学习一门编程语言的时候，往往还是需要有些许基础的。比如，怎样使用文本编辑器，怎样在相应的操作系统中安装软件，等等。并且，要想成为一名高级的 Go 软件工程师，你需要了解的周边知识可能在数量上会比 Go 语言编程本身多出几倍甚至几十倍。这就像摘苹果一样，如果要摘到苹果，就需要徒手爬上果树，或者找到工具帮助你；如果想摘到果树顶端最甜的那个苹果，就需要花费更多的时间和精力，爬过更多的枝叶。希望本书能成为帮助你摘苹果的工具。但是，你还是需要先活动活动手脚的，毕竟最后享用苹果的是你而不是梯子或者其他什么东西。在想有所收获之前，请先潜心地学习和积累。

我在这里列出了一些你可能需要爬过的枝叶。

- 知道计算机系统是什么，以及它的核心是由哪些部件组成的。比如，CPU 是什么？它是做什么用的？
- 知道软件是什么，以及它们是怎么设计和开发出来的。比如，操作系统和网络浏览器都是软件吗？
- 了解一些基本的 Linux 操作系统知识。比如，在命令行（也被称为终端）环境中怎样进入一个目录？怎样查看一个文本文件？
- 使用过一两门编程语言，真正编写过程序或软件。当然，没有也没关系，因为 Go 语言很适合作为计算机编程的入门语言。

当然，以上这些仅供参考。你在阅读本书的过程中边看边学也完全没问题，甚至可以看完本书再去学习相关知识。采用哪种学习方式完全取决于你自己。

## 关于示例代码

我会把本书涉及的所有 Go 示例代码（也许不只，可能会有小惊喜）都放到一个名为 goc2p 的项目中。读者可以访问 https://github.com/hyper-carrot/goc2p 下载它。该项目的根目录的路径应该被包含在环境变量 GOPATH 中。如果你不了解 Git（一款代码版本控制工具），请在互联网中搜索“git”并获取相关信息。

## 关于勘误

由于时间水平都比较有限，所以书中难免会出现一些纰漏和错误。如果大家发现了一些问题，请及时指正。我也会尽快在本书后续的版本中加以改正。我专为本书设立的电子邮箱是：hypermind.cn@gmail.com。我的微博是：特价萝卜。我欢迎也希望和大家一起学习和讨论 Go 语言，并共同推动 Go 语言中国社区的发展。

## 致谢

撰写图书是一项需要大量精力和一定毅力的工作，尤其是编写技术图书，更需要作者对相关知识进行深入的梳理和系统的整合，还需要制作各种图表，编写各种示例。对于个人而言，工作量确实是不小的。但是，这个写作过程也是有趣的，通过写作我也收获了很多。当然，很多收获是来自他人的传授。其中，图灵公司的编辑王军花、张霞以及傅志红老师都给予了我很大的帮助，尤其是在写作技巧和图书结构方面。此外，还有目前中国业内公认的 Go 语言专家许式伟、谢孟军等，我在编写本书的时候经常向他们讨教。在此，我对这些帮助过我的专家和同行表示由衷的感谢。同时，我也要感谢我的家人。没有他们的支持和理解，我不可能在有限的业余时间里完成这本书。

# 目　录

## 第一部分　Go 语言的世界



## 第二部分　编程基础

## 第三部分 并发编程



## 第四部分 编程实战

# Part 1 第一部分

# Go 语言的世界

作为本书的第一部分，我会先带领你从宏观上粗略地了解 Go 语言的一些特点，其中包括它与其他编程语言相比的优势。当然，我的描述会很客观。不过，如果你有不同意见尽可以和我探讨。

在经过一些概述之后，我们会一起来为编写 Go 语言程序做准备。这包括在不同的操作系统上安装和配置 Go 语言（别担心，这很简单），了解 Go 语言的工程结构和标准命令。

与 C++、Java、C# 等一些主流的编译型通用编程语言相比，学习 Go 语言的门槛并不高。不过，Go 语言确实也有它自己的一些规则。只要我们遵循了这些规则，就可以非常畅快地编写和运行 Go 语言程序了。稍后你就会了解到这一点。下面，就让我们一起快步进入 Go 语言的世界。

# 第1章 初识Go语言

在真正地讲解Go语言之前，先让我们对它有个宏观的认识。本章将介绍Go语言是什么、有哪些特性、有哪些优势和不足，以及应该怎样学习它。现在就让我们开始吧。

## 1.1 Go语言特性一瞥

我们先来看看Go语言的主要特性。

- 开放源代码的通用计算机编程语言。开放源代码的软件（以下简称开源软件）更容易被修正和改进。因为，几乎所有的互联网用户都可以看到这些源代码，并可以提供自己的意见和建议，甚至还可以参与到实际的开发工作中去。与一些不公开源代码的软件（也可称为闭源软件）相比，开源软件的开发迭代周期要短很多，并且迭代速度要快很多。这正印证了“众人拾柴火焰高”这句俗语。
- 虽为静态类型、编译型的语言，但Go语言的语法却趋于脚本化，非常简洁。这方面的内容，我们会在第3章集中论述。我们可以直接使用各种字面量来表示各种数据类型以及它们的值，并且还可以使用非常简约的方式操作它们。
- 卓越的跨平台支持，无需移植代码。这里的跨平台主要是指跨计算架构和操作系统。Go语言目前已经支持绝大部分主流的计算架构和操作系统了，并且这个范围还在不断地扩大。我们只要下载与之对应的Go语言安装包，并且经过基本一致的安装和配置步骤，就可以使Go语言就绪了。除此之外，在编写Go语言程序的过程中，我们几乎感觉不到不同平台的差异。
- 全自动的垃圾回收机制，无需开发者干预。Go语言程序在运行过程中的垃圾回收工作由Go语言运行时系统负责。不过，Go语言也允许我们进行人工干预。在第三部分，我们会适时地介绍这方面的内容。
- 原生的先进并发编程模型和机制。Go语言拥有自己独特的并发编程模型，其组成部分有Goroutine（也可称为Go程）和Channel（也可称为通道）等。实际上，这部分内容也是本书最重要的主题。在第三部分，我们会一起领略它们的风采。
- 拥有函数式编程范式的特性，函数为一等代码块。Go语言支持多种编程风格，包括面向对象编程和函数式编程。而对函数式编程的最有力的支撑就是Go语言将函数类型视为了

第一等类型。我们会在第3章中说明这一点。

- 无继承层次的轻量级面向对象编程范式。Go语言中的接口与实现之间完全是非侵入式的。这种接口实现方式总是被人们津津乐道。不但如此，在Go语言中只有类型嵌入而没有类型继承。这规避了很多与继承有关的复杂问题，也使类型层次更加简单化了。这部分内容同样会在第3章中展示。
- 内含完善、全面的软件工程工具。Go语言自带的命令和工具相当地强大。通过它们，我们可以很轻松地完成Go语言程序的获取、编译、测试、安装、运行、运行分析等一系列工作。可以看到，这几乎涉及了开发和维护一个软件的所有环节。这可以算是一站式的编程体验了。在本章，我们会简要地浏览一下这些工具。
- 代码风格强制统一。Go语言的安装包中有自己的代码格式化工具。我们可以利用它来统一程序的编码风格。
- 程序编译和运行速度都非常快。由于Go语言的简洁语法，我们可以快速地编写出相应的程序。加之Go语言强大的运行时系统和先进的并发编程模型，Go语言程序可以充分地利用计算环境并运行飞快。在这本书的内容中，我们会通过对Go语言的各个方面的深入介绍使你真正地了解到Go语言如此优秀的原因。
- 标准库丰富，极适合开发服务端程序和Web程序。Go语言是通用的编程语言，但是它也有尤为擅长的几个方面，例如系统级程序和Web程序等，这主要得益于它丰富的标准库。我们会在本书中介绍很多Go语言标准库及其使用方法。但是由于篇幅原因，这些只会是冰山一角。

不知道在看到Go语言如此多的先进特性之后，你是否已经心动了。反正我已经为此折服并感到激动了。这也是我迫不及待地深入研究它并通过编写一本专著把我知道的所有细节与大家分享的原因。

## 1.2 Go语言的优劣

在软件行业做过一段时间的人都知道，没有万能的编程语言，没有万能开发框架，也没有万能的解决方案。任何新技术的产生都应该归功于一部分人对老旧技术的强烈不满。Go语言也不例外。比如，C的依赖管理、C++的垃圾回收、Java笨重的类型系统和厚重的Java EE规范，以及脚本语言（如PHP、Python和Ruby）的性能，这些都是很多开发者社区经常争论和抱怨的问题。

Go语言是集多编程范式之大成者，体现了优秀的软件工程思想和原则，其特性可以使开发者快速地开发、测试和部署程序，大大提高了生产效率。下面我们来看看与其他主流语言相比，Go语言具有的优势。

- 相对于C/C++来讲，Go语言拥有清晰的依赖管理和全自动的垃圾回收机制，因此其代码量大大降低，开发效率大大提高。
- 相对于Java来讲，Go语言拥有简明的类型系统、函数式编程范式和先进的并发编程模型。因此其代码块更小更简洁、可重用性更高，并可在多核计算环境下更快地运行。