

TURING 图灵原创



云计算时代的新型编程语言 编程极客不可不学
示例丰富，引人入胜，一本就Go!

Go语言·云动力

 人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创



Go语言·云动力

[新加坡]樊虹剑◎著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Go语言·云动力 / (新加坡) 樊虹剑著. -- 北京 :
人民邮电出版社, 2012. 6
(图灵原创)
ISBN 978-7-115-28307-8

I. ①G… II. ①樊… III. ①程序语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2012)第104152号

内 容 提 要

Go语言是由谷歌的 Rob Pike、Ken Thompson 和 Robert Griesemer 共同设计开发的一种新型程序设计语言。2012年Go 1的推出,代表着Go语言的稳定成熟,也正式宣告Go走入了主流语言的行列。本书是Go语言程序设计入门书,介绍了Go语言的基础知识,包括静态类型、流程控制、函数、动态类型、面向对象、并发编程等内容,以及同其他C类语言相比,Go所具备的全新特性。同时,本书还介绍了Go语言在云计算中的应用。

本书适合Go语言初学者学习。

图灵原创

Go语言·云动力

- ◆ 著 [新加坡] 樊虹剑
责任编辑 王军花
执行编辑 李 静
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 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: 880×1230 1/32
印张: 9.625
字数: 250千字 2012年6月第1版
印数: 1-4 000册 2012年6月河北第1次印刷

著作权合同登记号 图字: 01-2012-2781号

ISBN 978-7-115-28307-8

定价: 39.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

Less is more.

(舍既是得。)

——Robert Browning

从 1978 年《C 程序设计语言》问世到现在，计算机世界已经发生了翻天覆地的变化。计算机器本身的性能已经得到几千倍的提高，计算机的互联互通更使很多不可思议的事情成为常态。身为计算机灵魂的工程师，程序员一直处在这场变革的风口浪尖。他们既有弄潮儿的自信与洒脱，又时刻面临被后浪吞噬的危险。更微妙的是，尽管计算机科学的先驱们很早就发现，编程是门文字和思维的艺术，而且顶尖程序员的生产力要远大于集体化大规模生产作业中的员工。但是，由于整个计算机产业的迅速发展，以及具备艺术细胞的程序员的稀缺，从业者只能重点投资工具和流程。自从 20 世纪 60 年代提出“软件危机”这一问题以来，各种理论和实践不断涌现，而通过变革编程语言来解决问题的努力，也一直在继续。但到目前为止，软件还在危机，同志仍需努力。

C.A.R Hoare 在 1980 年图灵奖获奖致辞中阐述得很精辟：

“设计软件有两种方式：第一种是尽量简化，使之明显地看不到弱点；另一种是尽量复杂，从而看不到明显的弱点。第一种方式非常困难。就像发现复杂的自然现象之中蕴含着简单的

物理定律一样，它需要投入、明察、甚至灵感。”

今天，我非常高兴地向大家介绍 Go 语言。40 年前，C 编程语言极大地解放了程序员的生产力。今天，新兴的 Go 语言也一定能够再次提升程序员的创造力和协作能力，从而让有才能的程序员得以在创作中充分发挥，让程序员重新成为令人尊敬的计算机灵魂的工程师。

目前，Go 语言才发布了短短几年时间，就已经赢得了广泛赞誉，并得到了广泛应用。它不但用于支持谷歌和 Heroku 的云计算平台，还在一些大公司的内部系统中担当关键角色，甚至普通程序员也用 Go 代替其他常见语言来处理手边的问题。这充分说明 Go 是一种成熟、实用、好用的语言。它的未来非常光明。

本书所介绍的内容正是使用 Go 语言时必不可少的基础知识。作者不满足于平铺直叙的方式，而希望能通过轻松的表达，让 Go 自然而然地接近读者，并以作者自身的经验和观点，启发读者独立地分析和试验，从而让读者能顺利地入门到精通，让 Go 成为最好的编程语言。

简介

Go 是种通用的编程语言。它类似 C，也参考了其他语言的一些设计。但从各方面来看，Go 都是种新语言。它的诞生源于 Go 的发明者对现有编程语言的不满，认为它们使编程变得太过困难，编译速度太慢，或者运行效率太低。所以，2007 年，谷歌的 Rob Pike、Ken Thompson 和 Robert Griesemer 决定设计一种全新的编程语言。两年后的 2009 年 11 月 10 日，他们发布了源代码全部开放的 Go 项目。而 2012 年 Go1 的推出，既代表着 Go 语言的稳定成熟，也正式宣告 Go 走入了主流语言的行列。

Go 的作者在介绍 Go 项目的目的时指出：

已经有十几年没有出现新的系统语言了，但此时计算环境发生了巨大变化，出现如下趋势。

- (1) 计算机的运行速度变得极快，而软件开发速度基本没变。
- (2) 依赖管理是今天软件开发的一大部分，但传统 C 语言的头文件是干净的依赖分析和快速编译的对立面。
- (3) 反抗 Java 和 C++ 等语言笨重的类型系统的起义已在壮大，推动着人们使用 Python 和 JavaScript 等动态类型系统语言。
- (4) 一些基本概念，例如垃圾回收和并行计算，并未在主流语言中得到充分支持。
- (5) 多核计算机的兴起带来了担忧和困扰。

所以，我们认为值得尝试开发一种新语言，一种并发的、支持垃圾回收的、快速编译型语言。

Go的产生源于对现有系统编程语言和环境的不满。编程已经变得很难，部分原因要归咎于选用的语言。人们要在高效编译、高效执行和轻松编程之间取舍；没有一种主流语言同时满足这三个方面的要求。相比安全性和效率，更注重轻松编程的程序员选择使用动态类型的语言，如Python和JavaScript，而不是C++，或者相对舒缓些的Java。

Go试图结合能轻松编程的解释型动态类型语言，以及能提高效率和安全性的静态类型编译语言。它面向的是现代的、支持网络功能的多核计算。最后，它还能保证速度：在一台计算机上编译一个大型执行文件最多就几秒钟。而要实现这些目标则需要解决几个语法问题：一个丰富但轻盈的类型系统；并发和支持垃圾回收；稳定的依赖规范等。这些靠库和工具都不能很好解决，新的语言在召唤。

这就是说，Go语言在设计时，就糅合了静态编译型语言的高效和安全，以及动态解释型语言容易编程的特点，又面向当代的网络和多核计算，强调速度，精简了类型，避免冗长的声明，保持每个概念的独立。这些，使Go既容易学习，也便于使用。无论是解决手边的小问题，还是依靠集体完成的大项目，Go都是合适的通用语言。

Go是对C语言最彻底的一次改进，不只是声明语法与C完全不同，而且舍弃了危险的指针运算，还调整了运算符的优先级，并在各个细微的地方做出必要的改变。这使Go语言更规整、更安全。所以尽管具备C类语言的基础可以帮助学习Go，但舍弃C的使用习惯，熟悉Go的语言风格，也是对有经验的程序员的一种挑战。

第1章通过具体的案例，解释Go程序的基本结构，让有一定功力的读者迅速上手，也让初学者对Go有个初步印象，便于展开后续章节。

Go提供了丰富的数据类型，包括不同字宽的整型、浮点型、复数型、数组、字符串、切片、结构、映射、程道和界面。它们声明的变量和常

量，像细胞一样成为一个程序最基本的单元。数据类型、变量和表达式构成了第 2 章的内容。

Go 的流程控制语句能搭建结构良好的程序：块用于语句分组和作用域控制，if-else 用于判断，switch 用于选择，for 用于循环。第 3 章将通过具体的例子讲述每个基本的控制语句。

Go 的函数声明不可以嵌套，但本身可以作为变量，可以返回多个值，可以递归调用，也可以使用并保存外部的变量。变量通常在离开作用域时失效。但 Go 自动跟踪变量的分配，确保仍被使用的变量不会回收，而不再使用的变量，也能及时地被 GC（Garbage Collection，垃圾收集器）收集，释放其所占内存。第 4 章专门讲解函数。

这样，前四章就全面介绍了 Go 这一静态编译语言的基础知识。

Go 的界面类型是动态编程的基础，在程序运行时才根据具体值的类型来确定调用的方法。第 5 章介绍映射、界面和动态编程。

第 6 章介绍 Go 面向对象的设计。Go 的各种数据类型，都可以作为函数的接受者。这样函数成为此类型的方法。这使 Go 具备面向对象的语法和数据封装特性，但 Go 不使用继承。Go 的内置结构和界面类型可以满足类似功能，也更加灵活。

Go 的招牌是并发机制。Go 的函数可以用 go 语句并发执行，称为“去程”（goroutine）。并发去程的同步与数据传递靠的是程道类型的变量。select 语句可以从多个程道中随机挑选一个不再阻塞的去程执行。第 7 章将介绍 Go 的并发原语的使用。

第 8 章介绍 GAP 的 Go 运行器的使用，分析云计算的特点，以及 Go 如何在语言层级为云计算提供动力。

第 9 章介绍 Go 语言的一些标准包，包括 fmt、bytes、http、time、template、regexp、gob 和 json。

附录提供 Go 工具的用法、完整的 EBNF 语法规范，以及中英术语对照表，方便大家参考。

写作风格

语言是我们分析和交流的工具。尽管计算机编程需要使用简单的英语，但我们对于问题的思考和解决方法的推导，仍然要依靠母语。所以，本书尽量不使用英文术语，而是试图和其他科学一样，使用对应的中文翻译。这样，程序就像数学公式一样，成了抽象推理的工具。例如，对于变量名，本书会尽量使用类似数学公式中简单的 `xyz` 那样的字母，而不使用有意义的英文。尽量不使用注释，而是依靠正文的讲解，帮助理解程序。

Go 有一些新的概念，例如 `goroutine` 和 `channel`，并没有现成的译法，作者就自作聪明地把它译成“去程”和“程道”，希望能音意皆通。而在 Go 中，还有一些受 Java 影响但用法不同的词，例如 `interface`，为了不受旧有概念的影响，也使用了新的译名“界面”。作者相信这些都能对读者使用母语学习 Go 语言有所帮助。

作者深信，学习和阅读是个需要集中精力的过程，任何花边注脚都是对正文学习的干扰。所以，本书版面非常简单，只有文字介绍和 Go 语言程序。作者也深信，学习编程的最好方法是亲自动手编写程序。只有这样才能理解程序的整体结构和细枝末节，只有自己动手发现和排除错误，才能从中学到东西。所以，尽管本书提供了 70 多个完整的示例程序，但不提供源代码下载，给大家一个边学边练的机会。

目 录

第 1 章 快速入门	1
1.1 编辑和编译	2
1.2 世界, 你好!	9
1.3 自我复制	11
1.4 猜数游戏	14
1.5 图灵机	19
1.6 排版工具	23
1.7 游乐场	30
1.8 位钱	36
1.9 小结	43
第 2 章 静态类型	44
2.1 数制	45
2.1.1 整数	45
2.1.2 二进制补码	46
2.1.3 整数比较	47
2.1.4 按位运算	48
2.1.5 浮点数	50
2.1.6 复数	51
2.1.7 优先级	52
2.1.8 表达式	53
2.2 字符和字符串	54

2.2.1	UTF-8 编码	54
2.2.2	Unicode 字符	55
2.2.3	转义字符	56
2.2.4	字符串	56
2.2.5	字符串转换	57
2.3	数组	60
2.3.1	声明	60
2.3.2	下标	61
2.3.3	赋值	61
2.4	切片	63
2.5	结构体	67
2.5.1	项	68
2.5.2	内置	69
2.6	指针	71
2.7	小结	73
第 3 章	流程控制	74
3.1	简单语句	75
3.2	判断语句 if	77
3.3	多分支语句 switch	79
3.4	循环语句 for	82
3.5	遍历	84
3.6	标号和跳转	85
3.7	作用域	86
3.8	小结	88
第 4 章	函数	89
4.1	签名	90
4.2	参数	91
4.3	返回语句	93
4.4	函数调用	95
4.5	闭包	97
4.6	压后	99
4.7	派错和恢复	100

4.8	方法	102
4.9	包	105
4.10	导入	107
4.11	程序初始化	108
4.12	小结	110
第 5 章	动态类型	111
5.1	映射	112
5.2	界面类型	116
5.3	界面值	119
5.4	error 界面	122
5.5	有界无类	124
5.6	排序	126
5.7	类型断言	130
5.8	类型分支	133
5.9	反射	135
5.10	小结	141
第 6 章	面向对象	142
6.1	背景	143
6.2	术语	146
6.3	与 C++ 对比	149
6.3.1	继承	149
6.3.2	抽象类	151
6.3.3	泛型	153
6.4	小结	154
第 7 章	并发编程	155
7.1	背景	156
7.2	同步通信	158
7.3	去程	162
7.4	程道	164
7.5	遍历与关闭	166
7.6	MapReduce	168

4 | 目 录

7.7	select 语句	170
7.8	程道值	173
7.9	互斥	175
7.10	小结	177
第 8 章	云计算	178
8.1	背景	179
8.2	GAE	181
8.3	Hello 世界!	183
8.4	画胡子	185
8.5	留言录	191
8.6	用户 API	195
8.7	数据库 API	196
8.7.1	术语	197
8.7.2	Go 数据库 API	197
8.7.3	实体键	199
8.7.4	查询和索引	200
8.7.5	实体组	201
8.7.6	限制	201
8.8	交易	203
8.9	散段	205
8.10	内存缓冲	208
8.11	大件库	210
8.12	URL 抓取	213
8.13	任务队列	214
8.13.1	任务	216
8.13.2	任务执行	217
8.13.3	队列	218
8.14	后端	220
8.15	能力 API	221
8.16	电子邮件 API	222
8.16.1	发送	222
8.16.2	接收	223
8.17	信道 API	224
8.18	小结	226

第 9 章 标准包	227
9.1 格式包	228
9.1.1 格式输出 Printf	229
9.1.2 动词表	230
9.1.3 宽度和精度	231
9.1.4 报错	232
9.1.5 额外标记	232
9.1.6 格式输入	233
9.1.7 字符串格式	234
9.2 字节包	236
9.3 模板包	239
9.4 正则表达式包	245
9.5 时间包	253
9.6 超链接包	258
9.6.1 http 服务器和客户机	258
9.6.2 https 加密通信	260
9.6.3 Get	263
9.6.4 Post	265
9.6.5 Cookie	268
9.7 编码包	271
9.7.1 gob	271
9.7.2 json	275
附录 A Go 的安装和使用	277
附录 B EBNF	284
附录 C 中英术语对照表	291

第 1 章

快速入门

Well begun is half done.

(万事开头难。)

——谚语

本章通过几个具体的案例，解释 Go 语言程序的基本结构，让有一定功力的读者迅速上手，也让初学者对 Go 有个初步印象，便于后续章节的展开。

1.1 编辑和编译

大家都知道，学习编程的最佳方式就是动手编程。但这里有一个巨大的障碍。动手之前你要熟悉相关的工具，知道怎样使用编辑器，写出源代码。怎样操作编译器、链接器，得到可以执行的程序。另外还要考虑怎样运行这个程序，去哪里查看运行输出。并且还要知道，每一个步骤如果出了错，该怎样应对。

本书不打算手把手地教你使用这些必要工具。最好的办法是请教身边的专家。也可以参考其他资料，尤其是 Go 的正式发布网站 www.golang.org，自己慢慢摸索，并在使用中不断尝试。

为了帮助初学者克服这些与语言无关的技术障碍，我们特意整合了 Windows 上的 Acme 程序编辑器和 Go 编译器，并针对 Go 的编写、编译、执行和除错做了一个简单演示。Go 也可以运行在 Linux、FreeBSD 和 Mac OS X 上。读者可以根据自己的使用环境和习惯相应地调整。这些平台上 Go 语言环境的安装和使用，请参考附录。

读者可以从本书的支持网站 www.goplace.org 下载 `acme.zip`。它完全不需安装，直接解压到 Windows 的 `c:\` 即可。Go 工具和运行环境要求安装在 `c:\Go` 下面。而 Acme 是在 `c:\acme.app` 下面，执行它里面的 `acme.bat`，就可以启动编辑器了。

我们看一下 `acme.bat`：

```
set GOROOT=c:\\Go
set GOPATH=c:\\acme-home\\
PATH=%PATH%;%GOROOT%\bin
c:
cd \acme.app\Contents\Resources
acme.exe
```


这里有几个重要的路径：

- GOROOT 是 Go 的正式软件所在路径；
- GOPATH 是我们自己的软件和第三方的 Go 软件所在路径；
- PATH 中必须包括 GOROOT 的 bin 才可以使 Go 工具；
- acme-home 是 Acme 存放文件的根目录。

如果读者希望安装到其他路径下，则需要对 acme.bat 做相应的修改。

Acme 很神秘？其实编辑器就像钢笔铅笔，纯属个人喜好。一开始用哪个都写不出好字。慢慢地习惯了一种，就觉得它用着顺手了。所以，还没有习惯使用具体某一款编辑器的读者，不妨多试试，或许你会发现 Acme 是最出色的程序编辑器。顺便一提，Acme 和 Go 一样，是 Rob Pike 的作品。C 语言和 Unix 之父 Dennis Ritchie 也使用 Acme。

启动 Acme 后的第一感觉就是“它很不一般”，而且几乎也可以说是“不知所措”。菜单在哪里？帮助在哪里？怎么打开文件？在直接给出这些答案之前，我们有必要了解它的设计哲学。只有明白了我们真正需要的是是什么，才晓得为何这样以及如何才能这样。

Acme 和 Unix 的设计哲学是一脉相承的，都是只提供少量基本工具以及组合它们的方法，而不是针对每一项需求都准备互不相干的几十个上百个选项。

其实，Acme 就是一个运行在类似 Unix 的虚拟机上的编辑器。这个虚拟机使用 Inferno 操作系统。这和 Emacs 编辑器运行在 Lisp 虚拟机上类似。只不过，Emacs 只能使用 Lisp 编程来配置编辑器的各项功能，而它的 Lisp 程序只为 Emacs 服务。Acme 则只是 Inferno 的一个程序。我们可以使用类似 Unix 的 sh 脚本，以及使用类似 Go 的 Limbo 编程语言，来编写 Inferno 的其他命令和程序，与 Acme 一起配合，完成所需的编辑工作。

这样的 Acme，也可以认为是 Inferno 操作系统的图形用户界面