

## 01 | 工作区和GOPATH

2018-08-10 郝林



### 01 | 工作区和G...

朗读人：黄洲君  
11'10" | 6.41M

0:00 / 11:10

【Go 语言代码较多，建议配合文章收听音频。】

你好，我是郝林。从今天开始，我将和你一起梳理 Go 语言的整个知识体系。

在过去的几年里，我与广大爱好者一起见证了 Go 语言的崛起。从 Go 1.5 版本的自举（即用 Go 语言编写程序来实现 Go 语言自身），到 Go 1.7 版本的极速 GC（也称垃圾回收器），再到 2018 年 2 月发布的 Go 1.10 版本对其自带工具的全面升级，以及可预见的后续版本的关键特性，比如用来做程序依赖管理的 `go mod` 命令，这些都无不令我们欢欣。Go 语言在一步步走向辉煌的同时，显然已经成为软件工程师们最喜爱的编程语言之一。

我开办这个专栏的主要目的是要与你一起探索 Go 语言的奥秘，并帮助你在学习和实践的过程中获取更多。我假设本专栏的读者已经具备了一定的计算机基础，比如，你要知道操作系统是什么、环境变量怎么设置、怎样正确使用命令行，等等。

当然了，如果你已经有了编程经验（尤其是一点点 Go 语言编程经验）那就更好了，因为我在这里教你的都是 Go 语言中非常核心的技术。如果你对 Go 语言中最基本的概念和语法还不够了解，那么可能需要在学习本专栏的过程中去查阅[Go 语言规范文档](#)

我们学习 Go 语言时要做的第一件事往往是，根据自己电脑的计算架构（比如，是 32 位的计算机还是 64 位的计算机）以及操作系统（比如，是 Windows 还是 Linux），从[Go 语言官网](#)下载对应的二进制包，也就是可以拿来即用的安装包。然后，解压缩安装包、放置到某个目录、配置环境变量，并通过在命令行中输入 `go version` 来验证是否安装成功。

一般情况下，我们应该配置 3 个环境变量，即 `GOROOT`、`GOPATH` 和 `GOBIN`。其中，`GOPATH` 背后的概念是最多的，也是最重要的。那么，今天我们的面试问题就是：设置 `GOPATH` 有什么意义？

关于这个问题，它的参考回答是这样的：环境变量 `GOPATH` 的值可以是一个目录的路径，也可以包含多个目录路径，每个目录都代表 Go 语言的一个工作区（workspace）。这些工作区用于放置 Go 语言的源码文件（source file），以及安装（install）后的归档文件（archive file，也就是以“.a”为扩展名的文件）和可执行文件（executable file）。

如果你不会这个问题，不要急，我只是通过这道面试题作为一个引子，提到它的目的是为了引出它背后重要的知识点。

那么，它背后的知识点到底是什么呢，接下来我一起来看一看。

Go 语言项目在其生命周期内的所有操作（编码、依赖管理、构建、测试、安装等）基本上都是围绕着 `GOPATH` 和工作区进行的。

所以，这个问题其实主要考察了 4 个知识点，分别是：

1. 你是否清楚 Go 语言源码的组织方式；
2. 你是否了解源码安装后的结果（只有在安装后，Go 语言源码才能被我们或其他代码使用）；
3. 你是否理解构建和安装 Go 程序的过程。这在开发程序以及查找程序问题的时候都很有用，否则你很可能会走弯路；
4. 你是否清楚 `go build` 命令的一些可选项的用途和用法，虽然有的集成开发环境已经为你选择好了。

下面我就重点来聊一聊这些内容。

## 知识扩展

### 1. Go 语言源码的组织方式


pdf 由 我爱学 it ([www.52studyit.com](http://www.52studyit.com)) 收集并免费发布

与许多编程语言一样，Go 语言的源码是以代码包为基本组织单位的。在文件系统中，这些代码包其实是与目录——对应的。目录可以有子目录，所以代码包也可以有子包。

一个代码包中可以包含任意个以.go 为扩展名的源码文件，这些源码文件都需要被声明为属于同一个代码包。代码包的名称一般会与这些源码文件所在的目录同名。如果不同名，那么在构建、安装的过程中会以代码包名称为准。

每个代码包都会有导入路径。代码包的导入路径是其他代码在使用该包中的程序实体时需要引入的路径。在实际使用程序实体之前，我们必须先导入其所在的代码包。具体的方式就是import 该代码包的导入路径。就像这样：

```
1 import "github.com/labstack/echo"
```

 复制代码

在工作区中，一个代码包的导入路径实际上就是从 src 子目录，到该包的实际存储位置的相对路径。

所以说，Go 语言源码的组织方式就是以环境变量 GOPATH、工作区、src 目录和代码包为主线的。一般情况下，Go 语言的源码文件都需要被存放在环境变量 GOPATH 包含的某个工作区（目录）中的 src 目录下的某个代码包（目录）中。

## 2. 了解源码安装后的结果

我们很有必要知道 Go 语言源码在安装后会产生怎样的结果。

源码文件以及安装后的结果文件都会放到哪里呢？我们都知道，源码文件通常会被放在某个工作区的 src 子目录下。


那么在安装后如果产生了归档文件，就会放进该工作区的 pkg 子目录；如果产生了可执行文件，就可能会放进该工作区的 bin 子目录。

我再讲一下归档文件存放的具体位置和规则。

源码文件会以代码包的形式组织起来，一个代码包其实就对应一个目录。安装某个代码包而产生的归档文件是与这个代码包同名的。

放置它的相对目录就是该代码包的导入路径的直接父级。比如，一个已存在的代码包的导入路径是

```
1 github.com/labstack/echo,
```

 复制代码

那么执行命令

生成的归档文件的相对目录就是 [github.com/labstack](https://github.com/labstack/echo) , 文件名为 echo.a。

顺便说一下, 上面这个代码包导入路径还有另外一层含义, 那就是: 该代码包的源码文件存在于 GitHub 网站的 labstack 组的代码仓库 echo 中。

再说回来, 归档文件的相对目录与 pkg 目录之间还有一级目录, 叫做平台相关目录。平台相关目录的名称是由 build ( 也称 “构建” ) 的目标操作系统、下划线和目标计算架构的代号组成的。

比如, 构建某个代码包时的目标操作系统是 Linux, 目标计算架构是 64 位的, 那么对应的平台相关目录就是 linux\_amd64。

因此, 上述代码包的归档文件就会被放置在当前工作区的子目录 pkg/linux\_amd64/github.com/labstack 中。

总之, 你需要记住的是, 某个工作区的 src 子目录下的源码文件在安装后一般会被放置到当前工作区的 pkg 子目录下对应的目录中, 或者被直接放置到该工作区的 bin 子目录中。

### 3. 理解构建和安装 Go 程序的过程

我们再来说说构建和安装 Go 程序的过程都是怎样的, 以及它们的异同点。

构建使用命令 `go build`, 安装使用命令 `go install`。构建和安装代码包的时候都会执行编译、打包等操作, 并且这些操作生成的任何文件都会先被保存到某个临时的目录中。

如果构建的是库源码文件, 那么操作的结果文件只会存在于临时目录中。这里的构建的主要意义在于检查和验证。

如果构建的是命令源码文件, 那么操作的结果文件会被搬运到那个源码文件所在的目录中。

安装操作会先执行构建, 然后还会进行链接操作, 并且把结果文件搬运到指定目录。进一步说, 如果安装的是库源码文件, 那么结果文件会被搬运到它所在工作区的 pkg 目录下的某个子目录中。

如果安装的是命令源码文件, 那么结果文件会被搬运到它所在工作区的 bin 目录中, 或者环境变量 `GOBIN` 指向的目录中。

这里需要记住的是, 构建和安装的不同之处, 以及执行相应命令后得到的结果文件都会出现在哪里。

### 4. go build 命令一些可选项的用途和用法

在运行`go build`命令的时候，默认不会编译目标代码包所依赖的那些代码包。当然，如果被依赖的代码包的归档文件不存在，或者源码文件有了变化，那它还是会被编译。

如果要强制编译它们，可以在执行命令的时候加入标记`-a`。此时，不但目标代码包总是会被编译，它依赖的代码包也总会被编译，即使依赖的是标准库中的代码包也是如此。

另外，如果不但要编译依赖的代码包，还要安装它们的归档文件，那么可以加入标记`-i`。

那么我们怎么确定哪些代码包被编译了呢？有两种方法。

1. 运行`go build`命令时加入标记`-x`，这样可以看到`go build`命令具体都执行了哪些操作。另外也可以加入标记`-n`，这样可以只查看具体操作而不执行它们。
2. 运行`go build`命令时加入标记`-v`，这样可以看到`go build`命令编译的代码包的名称。它在与`-a`标记搭配使用时很有用。

下面再说一说与 Go 源码的安装联系很紧密的一个命令：`go get`。

命令`go get`会自动从一些主流公用代码仓库（比如 GitHub）下载目标代码包，并把它们安装到环境变量`GOPATH`包含的第 1 工作区的相应目录中。如果存在环境变量`GOBIN`，那么仅包含命令源码文件的代码包会被安装到`GOBIN`指向的那个目录。

最常用的几个标记有下面几种。


- `-u`：下载并安装代码包，不论工作区中是否已存在它们。
- `-d`：只下载代码包，不安装代码包。
- `-fix`：在下载代码包后先运行一个用于根据当前 Go 语言版本修正代码的工具，然后再安装代码包。
- `-t`：同时下载测试所需的代码包。
- `-insecure`：允许通过非安全的网络协议下载和安装代码包。HTTP 就是这样的协议。

Go 语言官方提供的`go get`命令是比较基础的，其中并没有提供依赖管理的功能。目前 GitHub 上有很多提供这类功能的第三方工具，比如`glide`、`gb`以及官方出品的`dep`、`vgo`等等，它们在内部大都会直接使用`go get`。

有时候，我们可能会出于某种目的变更存储源码的代码仓库或者代码包的相对路径。这时，为了让代码包的远程导入路径不受此类变更的影响，我们会使用自定义的代码包导入路径。

对代码包的远程导入路径进行自定义的方法是：在该代码包中的库源码文件的包声明语句的右边加入导入注释，像这样：


```
1 package semaphore // import "golang.org/x/sync/semaphore"
```

 复制代码



这个代码包原本的完整导入路径是`github.com/golang/sync/semaphore`。这与实际存储它的网络地址对应的。该代码包的源码实际存在 GitHub 网站的 `golang` 组的 `sync` 代码仓库的 `semaphore` 目录下。而加入导入注释之后，用以下命令即可下载并安装该代码包了：

```
1 go get golang.org/x/sync/semaphore
```

 复制代码

而 Go 语言官网 `golang.org` 下的路径 `/x/sync/semaphore` 并不是存放 `semaphore` 包的真实地址。我们称之为代码包的自定义导入路径。

不过，这还需要在 `golang.org` 这个域名背后的服务端程序上添加一些支持才能使这条命令成功。

关于自定义代码包导入路径的完整说明可以参看[这里](#)。

好了，对于 `go build` 命令和 `go get` 命令的简短介绍就到这里。如果你想查阅更详细的文档，那么可以访问 Go 语言官方的[命令文档页面](#)，或者在命令行下输入诸如 `go help build` 这类的命令。

## 总结

工作区和 `GOPATH` 的概念和含义是每个 Go 工程师都需要了解的。虽然它们都比较简单，但是说它们是 Go 程序开发的核心知识并不为过。

然而，我在招聘面试的过程中仍然发现有人忽略掉了它们。Go 语言提供的很多工具都是在 `GOPATH` 和工作区的基础上运行的，比如上面提到的 `go build`、`go install` 和 `go get`，这三个命令也是我们最常用到的。

## 思考题

说到 Go 程序中的依赖管理，其实还有很多问题值得我们探索。我在这里留下两个问题供你进一步思考。

1. Go 语言在多个工作区中查找依赖包的时候是以怎样的顺序进行的？
2. 如果在多个工作区中都存在导入路径相同的代码包会产生冲突吗？

这两个问题之间其实是有一些关联的。答案并不复杂，你做几个试验几乎就可以找到它了。你也可以看一下 Go 语言标准库中 `go build` 包及其子包的源码。那里面的宝藏也很多，可以助你深刻理解 Go 程序的构建过程。

[戳此查看 Go 语言专栏文章配套详细代码。](#)

# GO语言核心36讲

3个月带你通关 GO 语言

郝林

《Go 并发编程实战》作者  
GoHackers 技术社群发起人  
前轻松筹大数据负责人



版权归极客邦科技所有，未经许可不得转载

写留言

## 精选留言



郝林

70

有很多读写归档文件是什么意思。归档文件在Linux下就是扩展名是.a的文件，也就是archive文件。写过C程序的朋友都知道，这是程序编译后生成的静态库文件。

2018-08-10



与狼共舞

19

纯技术类文章，建议多用图。

2018-08-12

作者回复

你好，考虑到用图的话，播音员会很难描述。

2018-08-15



小巨小巨

17

什么是归档文件，归档文件里都是什么？有什么作用。

2018-08-10

作者回复

请看我置顶的留言。

2018-08-10



千年孤独

3

如果在面试中让老师来回答“设置GOPATH有什么意义？”这个问题，除去典型回答

老师会如何简要明了回答这个问题？

2018-08-10

作者回复

可以说，为了集中组织代码，以及代码互相引用。当然了，这么说后面面试官可能还会让你具体解答。

2018-08-15



第五季节

👍 3

工作区是指：放go的源码目录。

gopath是指：指向工作区的路径。

归档文件：相当于java的jar包。下载到本地私服

不知道对不对。纯属个人观点。

2018-08-10

作者回复

前两个对，归档文件在Linux下就是.a文件，也就是archive文件，是程序编译成功后生成的静态库文件。这跟Java的jar包还不太一样，jar包相当于动态链接库了。

2018-08-10



Harry

👍 3

一脸懵逼，并不知道归档文件是啥😅

2018-08-10

作者回复

请看我置顶的留言。

2018-08-10



xyang

👍 2

go语言适合做什么业务，能概述性的罗列讲述下吗

2018-08-11

作者回复

你好，我会在后边另写文章介绍。

2018-08-15



第五季节

👍 2

gopath 的设计类似java。

具体的用途是执行命令要用 例如：go run、go install、go get等。

允许设置多个路径。

在idea里面的多个project或工具组建都并列放在gopath的src下面。

例如：go install myProject1

go install myProject2

所以老师说的这个归档文件。可以理解成工作空间吗？

至于老师说的两个问题。

1:按照代码的执行顺序 从上往下 每个引入的初始化。

2:引入一下试试就知道了。

2018-08-10

作者回复



归档文件的解释请看置顶的留言。

2018-08-10



江山如画

👍 1

对于初学者第一次看确实有些难懂，但是多看几遍后就会发现其实干货满满，我读了好几遍，接触golang也快一年了，但是很多知识点是第一次接触到，感谢郝林老师！

2018-08-14

作者回复

对你有帮助就好。

2018-08-15



许明

👍 1

ide 我觉得vscode就很好用了，我现在是vscode + glide

2018-08-10

作者回复

嗯，我也用这种组合。

2018-08-15



jians

👍 1

看完再结合测试后的疑问：

在不同项目src中有同名包时，go build, install只会执行GOPATH中最早发现包的工作区，哪如何编译后面其他工作区中的同名包呢？

2018-08-10

作者回复

这就需要自己去放置了，或者临时把前面的工作区从GOPATH中去掉。

2018-08-15



Nixus

👍 0

归档文件，可以理解为程序的动态链接库文件吗？

2018-08-12

作者回复

相当于静态链接文件。

2018-08-15



有风的林子

👍 0

目前还没用到GOPATH包含多个工作区，不知多个目录间的分隔符是什么？空格、冒号、还是分号？如果作者顺便说一下就好了，至少增加一个知识点。😄

2018-08-12

作者回复

你好，这根据具体的操作系统而定。

2018-08-15



白宇

👍 0

请教一下，如何解决下载第三方包失败情况

2018-08-11

作者回复

你好，这属于是网络问题。

2018-08-15



Diviner.

0

求解释什么叫归档文件

2018-08-10

作者回复

请置顶的留言。

2018-08-10



有风的林子

113

说实话，这开篇的介绍不是太清晰，起步就有些乱，好像写作者没太重视。个人猜测☺

package是一个重要的关键词，不应当直接翻译成代码包，即便它们是在正文里。可能不译更好，也更准确清晰。

个人觉得首先应当把工作区的3个基础目录列出来：bin，pkg，src。然后基于此就比较好讲了。

另外，逻辑上import的路径并不特定表达实际文件系统的路径，它只是一个标识，目前来说对应到src之下的本地文件系统。

pkg下的\*.a直接说成库文件就很清楚了，然后大家就好顺序思考下去，不会被概念疑问阻塞。

所谓的命令源码文件，如果补充说一下“即包含 main() 入口函数的源码文件”，可能更清楚。

不好意思，提了不少问题。主要是因为太喜欢go了。

对于初学者，或许《The Go Programming Language》是最经典的，它也是go语言的发明者参与写的，中文被译成了《Go语言圣经》。

期待作者后续对go的深度剖析和分享，谢谢！

2018-08-12



lovedebug

21

谢谢作者大大的更新，不过感觉这一篇有点琐碎，缺乏清晰的脉络。另外总结里面可以稍微梳理下文章的大纲。

2018-08-10



Bob.Chen

15

请问对于之前不太了解go的同学，是否有推荐的入门书可以和专栏配合着看？

2018-08-10



咖啡色的羊驼

12

对于作者的两个问题。三个纬度延伸总结回答下：

### 1.总执行顺序的角度

引入的包 -> 当前包的变量常量 -> init()[多个同一包则按照顺序执行] -> main函数

### 2.依赖包执行顺序

被依赖的总是优先执行初始化，一个包只会被初始化一次。

a引入b，b引入c，则执行顺序c -> b -> a

### 3.单个包执行顺序的角度

总的前提:按照包中源文件名的字典顺序来排序执行。

当前包排序后的变量常量 -> 排序后的init()

2018-08-10



乌慢

👍 8

对于新手来说看起来不是那么的通俗易懂，如果加上图解就更好了，希望以后更加好

2018-08-11



Jeremy King

👍 5

首先肯定文章很有内容，希望对一些基本概念比如归档文件做一些解释，希望与具体实例结合性更紧密，希望在编排上有改善，更好的呈现给读者。

2018-08-10



objcoding

👍 5

go get 虽然简单粗暴，但是缺点是不能进行版本控制啊，虽然有很多第三方包管理工具，但我还是挺期待官方的dep工具完善后加入Go工具链中。

2018-08-10



melon

👍 4

感觉一脸懵逼的，建议先看一下官网这篇深入浅出的文章:<https://golang.org/doc/code.html>，看完再回过头来看就了然了。

2018-08-29



无羨

👍 4

请问库源码文件和命令源码文件的区别是什么

2018-08-11



狩月

👍 3

可以多一些图，直观一些。

内容也偏薄，三个变量另外两个多少带一句啊

2018-08-10



abeng

👍 3

还有，个人认为pkg中的归档文件，相当于动态链接库，主要用于go build时加快编译的。

2018-08-10



李皮皮皮皮

👍 2

宇宙最强，功能最丰富ide-goland

小巧玲珑-liteide

文本编辑器+插件-vs code或sublime

整个听下来感觉稍稍有点不清晰，什么都点到了，就是没点透的感觉。前面有个同学建议很好，最后总结是梳理一下大纲。

import自定义那边没听懂.....  
2018-08-11



小雨

👍 2

1，包的导入有几种路径方式，文稿中给出了通俗的方式，但是还有根据相对路径导入的方式，所以在导入包时，应该是先从当前项目(不知道可否叫当前工作区)开始查找，在根据GOPATH从前往后查找  
2，不冲突，因为按顺序找到所需要的包就不往后找了。

看其他就要中有提到启动顺序的，从init到main，应该不是这两题的要点了。  
2018-08-11



abeng

👍 2

作为技术类的专题，语速还是偏快，因为一边听需要一边消化理解，不过问题不大，可以先预习下期内容或再听一遍。

最后交作业：

1、Go在多个工作区查找依赖包的顺序如何？

A：原先一直认为\$GOPATH只能设置一个目录，复杂问题自动忽略😄，那么设置了多个工作区，go build时也应该默认第一个路径作为编译的当前工作区，在遇到import时递归查找编译，不这样实现就是坑，😄

2、如果在多个工作区中存在相同的包会不会冲突？

A：不会，原因同上！

2018-08-10



阿尔特纳晨风

👍 1

老师您好，近日golang 1.11 正式发版了，有着两个重要的特性：modules 和 WebAssembly，请问modules是要取代GOPATH吗？专栏会涉及这些新特性吗？谢谢

2018-08-29

作者回复

你好，module什么时候会取代gopath现在还不好说，至少明年春节前gopath还会是主流。我也许会在专栏的最后说明一下module吧。

2018-08-29



zjhiphop

👍 1

大概能够理解，如果能够加一些结构图流程图就更清晰了

2018-08-12



游红宇🍵

👍 1

首先抱怨一下，web端竟然不能留言！

其次，建议把官网那个项目结构的图示拿过来讲，也就是说，用具体案例来讲，更容易明白。

2018-08-11



罗罗

👍 1

如果能有视频最就好了

2018-08-11



知夫

👍 1

如果说有go build,go run,go clean等命令来构建，运行和清理，那有没有go creat之类的命令来创建一个工程目录呢？我看似乎好像没有  
2018-08-10



x.p.

👍 1

同问归档文件  
2018-08-10



阿神

👍 1

go用什么ide比较多  
2018-08-10



何方

👍 0

静态库文件，要怎么使用呢？  
2018-10-15



马神

👍 0

我当然看到了置顶，我是问题是为什么要自创一些名词而不采用标准的人能听懂的名词？故作高深很牛逼么？请你转告作者，别自创一些乱七八糟的名词来误导读者，可不可以用通俗易懂的语言来讲课，讲课是给读者讲的，不是讲给他自己听的。  
2018-10-15



乘坐Tornado的线程魔法师

👍 0

文中第一节写到“如果不同名，则在安装、构建的过程中以代码包的名字为准”请作者帮忙再确认下，因为我查阅了中英文的资料，都显示go install和go build时都已实际文件目录的路径为准，而不是package name  
2018-10-07



sureingo

👍 0

老师您好，请问库源码文件和命令源码文件什么意思呢 有什么区别啊  
2018-09-22

作者回复

后边都会讲啊，往后看。

2018-09-23



乔巴

👍 0

代码示例或者图能表达的很清楚，文字差一点儿。  
2018-09-06



忘怀

👍 0

配点图吧，有些东西没图理解起来还是比较抽象。图片可以让解说略过。  
2018-09-05

作者回复

图我在后面再适当的补充一些吧，现在没那个精力。

2018-09-06



AboutABoy

👍 0



对于刚学go的我，可能因为是小白的缘故。你这章讲的一些库源码文件与命令源码文件在go build和install的过程看的很懵逼，因为我完全不知道何为库源码文件和命令源码文件的定义，然后去Google才知道，但是下两章就开始讲这两个源码文件，个人感觉讲的顺序有点乱。

2018-09-01

作者回复

专栏有个预习篇你可以先看看。

2018-09-02



My dream

0

老师，这个新增modules肯定有区别和作用，不然官方更新这个不就没有任何意义了么？新增的这个模块，要如何用，用在哪里！我们都是一脸懵

2018-08-28

作者回复

你可以先不关注module，成为主流最早也是明后年的事了。

2018-08-29



行者

0

这个地方还是挺让人迷惑的，我理解gopath与工作区是一对多的关系，通过gopath引入各种第三方库。

2018-08-26

作者回复

对，理解正确。

2018-08-26



My dream

0

老师，你好，我看今天go的1.11版本出来了，新增了modules等特性的支持，请问你要重点讲一下不？

2018-08-25

作者回复

你好，我看看是不是在最后的拾遗篇讲一下。不过如果没有什么让人迷惑之处的话也可能略过。

2018-08-26



平头哥

0

推荐一个比较好用的go包管理工具呗

2018-08-25

作者回复

glide、gb。

2018-08-26



Yangjing

0

终于感到全部看懂了👍

2018-08-23

作者回复

赞！

2018-08-24



Femtoyue

0

命令源码文件和库源码文件怎么区分

2018-08-23



夏姬八起

0

如果安装的是命令源码文件，那么结果文件会被搬运到它所在工作区的 bin 目录中，或者环境变量GOBIN指向的目录中。老师为什么我的main.go每次install都在gobin的目录下 而不是在工作区的bin路径下

2018-08-21

作者回复

只要设置了gobin，命令源码文件就会总安装到那里了。

2018-08-22



null

0

希望可以有更多的图和具体实例做说明，这样更容易让人理解

2018-08-20



lcf枫

0

我觉得有时候图示表达很清晰，事半功倍。既然大伙都来学习，只要语音说有图想必都会看原文的。

2018-08-19



来碗绿豆汤

0

老师，我用的是windows系统。我将我的一个工作目录配置到了GOPATH。然后在该目录的src目录下面创建了一个package。执行go install的时候报错，提示找不到这个包。然后错误信息显示，它只去go的安装目录下面找了，并没有去我新配置的GOPATH下面找。

2018-08-18

作者回复

我记得windows得用个方式让环境变量生效吧，不记得是哪个方式了。

2018-08-19



不瘦十斤不改名

0

构建的存在意义就是检查和校验吗，因为我看构建和安装都会生成一个可执行文件，只不过安装的可执行文件被搬运到了工作区bin下，然后库结果文件一个是临时目录，一个是pkg下

2018-08-18

作者回复

对于命令，你构建完了不是一般还要运行下看看嘛，所以得放在你目录里便于你运行啊。

2018-08-19



缘木求鱼

0

在工作区中，一个代码包的导入路径实际上就是从 src 子目录，到该包的存储位置的相对路径。老师你好这句话有点不明白

2018-08-17

作者回复

比如，这个代码包的源码文件的存放路径是<工作区目录>/src/a/b/c，那么它的代码包导入路径就是a/b/c。你对比一下这两个路径，体会一下相对路径的意思。

2018-08-17



balancer

0

期待：多一点深度剖析go内存 并发设计 gc ，工程中比较常遇到的问题

2018-08-16

作者回复

你好，这部分我会适当写一些，不过仍然以核心知识和面试题为主线。如果想深入了解可以看《Go并发编程实战》第二版。

2018-08-16



Yashin

0

说到gopath 是否可以补充说明一下当前项目下的vendor目录呢？已经常用的包管理工具基本原理

2018-08-14



Takky

0

打包编译好的归档文件，在没有源码的情况下，能够给别的项目使用吗？能够像Java的jar文件或者.net的dll文件那样使用吗？

2018-08-13



星潮

0

1.从编译的main文件的import倒叙开始寻找依赖层层递进到最后没有依赖的包开始编译，每一个包总是以import的倒叙进行，已经编译过的不再进行编译

2.不冲突

2018-08-13



孙要飞

0

什么情况下需要设置多个gopath。怎么设置多个gopath

2018-08-13



白宇

0

请教一下，您是如何解决不能下载官网的包

2018-08-11



熊出没

0

新手 还是有许多不理解的 文章能否提供操作性支持

2018-08-11



深藏Blue

0

请教老师：在实际工作中GOPATH通常设置几个路径？我刚接触go两天。在jetbrains提供的ide中看到可以设置全局的GOPATH目录（默认读取系统的环境变量的值），也可以设置project级别的GOPATH。这种方式不禁让我类比到了maven,全局GOPATH可以理解成maven里面的本地仓库，project级别的可以理解成具体项目的depends吗？如果可以这样理解的话 执行go install的时候，结果是放入全局GOPATH还是项目自己的GOPATH里面？纯粹go小白，如果有问错的，各位老鸟勿嘲笑

2018-08-11



原来你也在这里

0

像这种环境配置，最好配上一些图进行讲解，因为很多人未必理解文中的术语，有时候反而是一些图更让人一目了然。其实我也不是很懂，只知道是这样子配置，但是为什么是这样子配置

就知道了  
2018-08-11



Rover

👍 0

如果归档文件相当于静态库，那go有动态库的概念吗？  
2018-08-11



信天游

👍 0

我的理解是归档文件用来加速编译的，编译成二进制之后就用不上了。二进制的执行是不依赖归档文件的，  
2018-08-11



信天游

👍 0

归档文件内容是静态库的话，内容会包含在二进制文件里面吗？  
二进制文件执行是不依赖归档文件。  
2018-08-11



信天游

👍 0

工作区和maven里面的project是类似的概念。  
2018-08-11



董柏顺 kev<sup>n</sup>

👍 0

讲的很清楚，不清楚的朋友可以查查基本概念□  
2018-08-11



Jowin

👍 0

搜索顺序是GOROOT->GOPATH，先找到哪个用哪个  
2018-08-10



用心

👍 0

明白了gopath goroot gobin 作用  
赞一个  
2018-08-10



李小龙

👍 0

老师归档文件不是很理解，是指的workspace么  
2018-08-10



saiyn

👍 0

归档文件就是archive文件，也就是常见的.a文件，实际上就是将多个源码文件编译生成的目标文件(.o)打包归档成一份.a文件  
2018-08-10



(｡•̀ㅂ•́)面团

👍 0

什么是构建？什么是安装？归档文件有什么用？  
2018-08-10