

如何在 Linux & MacOS 上优雅的发布 Go 程序

Go语言中文网 2019-12-13

以下文章来源于一艘慢船，作者JayL



一艘慢船

落盘固化记忆，分享提升价值。

这两天又写了一个开源小工具，时间戳转换与对比工具:ts，主要是在公司运维时使用。程序写完了发现安装过程不够自动化，特别是对于个人使用 MacOS 系统，日常运维是 Linux 操作系统。虽然 Go 语言支持跨平台编译，但是拿着个二进制程序拷贝来拷贝去，总归不像样。本文记录一下如何优雅在 Linux & MacOS 上发布 Go 程序的过程。

🔗必备工具

在实操之前，请首先确认成功安装以下工具：

- brew
- goreleaser
- godownloader

🔗操作步骤

首先当然是完成项目的编码工作。程序能够正常的编译执行。以ts项目为例。

```
$: git clone https://github.com/liujianping/ts
$: cd
$: |— LICENSE
```

```
├── README.md
├── cmd
│   ├── main.go
│   ├── main_test.go
│   └── root.go
├── **dist**
│   ├── CHANGELOG.md
│   ├── checksums.txt
│   ├── config.yaml
│   ├── darwin_386
│   │   └── ts
│   ├── darwin_amd64
│   │   └── ts
│   ├── linux_386
│   │   └── ts
│   ├── linux_amd64
│   │   └── ts
│   ├── ts_0.0.6_Darwin_i386.tar.gz
│   ├── ts_0.0.6_Darwin_x86_64.tar.gz
│   ├── ts_0.0.6_Linux_i386.tar.gz
│   └── ts_0.0.6_Linux_x86_64.tar.gz
├── go.mod
├── go.sum
├── **install.sh**
├── main.go
...
```

🔗Shell 脚本安装

该项目非常简单，代码也很少。以上目录展示了两个关键目录或文件，分别是：

- dist 目录
- install.sh 安装脚本

这两处变更均是通过工具生成的。现在我们生成它：

```
$: rm -rf dist install.sh

# 生成安装脚本
$: godownloader --repo=liujianping/ts > ./install.sh

# 生成发布目录
$: goreleaser init
# 直接操作会自动发布到 github 对应的 repo 上，前提是需要 tag
$: goreleaser
```

通过 goreleaser 操作，会自动将发布目录发布到 <https://github.com/liujianping/ts/releases> 发布页中。当然前提是在系统环境变量中设置好 `GITLAB_API_PRIVATE_TOKEN` 环境变量，具体值请在 github 上进行申请。

其实以上过程完成后，就可以实现跨平台安装了。`install.sh` 脚本可以直接执行，通过判断本地系统版本相应的去下载安装包，自动进行安装。

```
$: curl -sL https://raw.githubusercontent.com/liujianping/ts/
```

🔗Brew 安装

以上讲的是通过 SHELL 脚本的方式安装。对于 MacOS 系统而言，程序还可以打包进入 brew 包管理服务中，通过 `brew install` 的方式进行安装。

首先，在项目的 <https://github.com/liujianping/ts/releases> 发布页中找到 MacOS 的安装包地址：<https://github.com/liujianping/ts/releases/download/v0.0.6/ts0.0.6-Darwinx8664.tar.gz>。

创建一个 brew formula:

```
$: brew create https://github.com/liujianping/ts/releases/dowr
```

通过这个命令，brew 会创建文

件: `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/ts.rb`。进行简单的修改，仅提供安装脚本:

```
# Documentation: https://docs.brew.sh/Formula-Cookbook
#               https://rubydoc.brew.sh/Formula
# PLEASE REMOVE ALL GENERATED COMMENTS BEFORE SUBMITTING YOUR
class Ts < Formula
  desc "timestamp convert & compare tool"
  homepage ""
  url "https://github.com/liujianping/ts/releases/download/v0.
  sha256 "cf28627f973b03c2c103032f9c2be35d1d2f556fdf79f5139cc6
  # depends_on "cmake" => :build

  def install
    bin.install "ts"
  end
end
```

保存修改。现在在个人电脑上就可以进行 `brew install ts` 安装了。但这还没完成发布。在个人 github 账号下创建一个 homebrew- 前缀的 repo。如 `liujianping/homebrew-tap`。将新创建文

件: `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/ts.rb` 发布到该 repo 下。现在就可以通过 brew 的方式安装应用了。

```
$: brew tap liujianping/tap && brew install ts
```

📌 小结

很多好的开源项目都有完善的安装指南，通过学习这些开源项目，将好的经验应用到自己的项目中，是一条捷径。

如果觉得有用，欢迎点赞我的开源项目：

- github.com/liujianping/job 将命令规划成计划Job
- github.com/liujianping/ts 时间戳转换与对比工具

推荐阅读

- [Go 创建对象时，如何优雅的传递初始化参数](#)
- [GCTT 出品 | 优雅关闭的 Go Web 服务器](#)

喜欢本文的朋友，欢迎关注“**Go语言中文网**”：



Go语言中文网启用微信学习交流群，欢迎加微信：**274768166**