

很多时候，我们需要运行多个命令来完成一件事情

可以有两个方案：

1. 写shell脚本
 1. build.sh
 2. 比较复杂，不通用
2. makefile
 1. 定义一些常规命令（build, clean等）
 2. 非常简单（跟编程语言无关，不受限制）
 3. 可以完成对项目，源码进行编译构建。

基本规则：

```
target: dependencies
      command
```

1. target: 目标
 1. 要生成的目标文件的名字：blockchain
 2. 只是一个动作，是一个标签：clean
2. dependencies: 依赖
 1. 是依赖，通常是文件，完成target所需要的输入
3. command: 命令
 1. shell命令，一条语句一行，前面有一个

安装make命令

linux: 自带

windows: make-3.81.exe

安装之后，将make.exe所在的目录添加到PATH中,例如：C:\Program Files (x86)\GnuWin32\bin

在工程目录下，创建文件：

makefile或者Makefile,

make命令在执行时会自动寻找这两个文件。

```
#编译命令
blockchain:
    go build -o blockchain *.go

clean:
    rm *.db blockchain
```

执行:

1. make blockchain ==> 正常执行
2. make ==》会自动执行第一个标签
3. make clean ==》删除文件

.PHONY

问题: 当执行命令时, 如果在当前目录下面存在了与标签 (命令) 同名的文件, 那么命令不会生效, 不会触发

解决办法: 使用.PHONY关键字, 将命令写makefile中

```
.PHONY:blockchain clean

#编译命令
blockchain:
    go build -o blockchain *.go

clean:
    rm *.db blockchain
```

```
duke@DUKEDU51C6 MINGW64 /c/goprojects/src/go5期/03-比特币/v2-pow
$ make clean
make: "clean"是最新的。
duke@DUKEDU51C6 MINGW64 /c/goprojects/src/go5期/03-比特币/v2-pow
$ make clean
rm *.db blockchain
rm: cannot remove '*.db': No such file or directory
make: *** [clean] 错误 1
```

不添加.PHONY clean时, 命令不执行, 因为当前目录有clean文件

别名

```
.PHONY:blockchain clean

app:blockchain    #<<<<=====

#编译命令
blockchain:
```

```

    go build -o blockchain *.go

clean:
    rm *.db blockchain

test:
    go test -v *_test.go

docker:
    #docker build xxx

```

make app 等同于make blockchain

bee框架的makefile参考:

```

VERSION = $(shell grep 'const version' cmd/commands/version/version.go | sed -E
's/.*"(.+)"$$/v\1/')

.PHONY: all test clean build install

GOFLAGS ?= $(GOFLAGS)

all: install test

build:
    go build $(GOFLAGS) ./...

install:
    go get $(GOFLAGS) ./...

test: install
    go test $(GOFLAGS) ./...

bench: install
    go test -run=NONE -bench=. $(GOFLAGS) ./...

clean:
    go clean $(GOFLAGS) -i ./...

publish:
    mkdir -p bin/$(VERSION)
    cd bin/$(VERSION)
    xgo -v -x --targets="windows/*,darwin/*,linux/386,linux/amd64,linux/arm-
5,linux/arm64" -out bee_$(VERSION) github.com/beego/bee
    cd ..
    ghr -u beego -r bee $(VERSION) $(VERSION)

```