

既然是做服务器，以及 go 对 windows 的支持不好，我只在 linux 上安装环境

这个博客是我自己从头开始搭建，在设置语法高亮的时候碰到了很多问题，主要原因嘛，就是网上搜索很多东西，大多都是同样的内容，但是根据这个内容根本搭建不出 Go 语言的开发环境，为了给大家减少不必要的麻烦，现在我把搭建的过程给大家讲一下。

我使用的环境是 CentOS 6.5 版，使用的是虚拟机版本，因为可能会经常搭建新的环境，我这里有虚拟机 VM10 的纯净版系统，下载地址如下：[纯净版虚拟机系统](#)

请选择 CentOS_64.rar 文件

文件名	大小	修改时间
CentOS_64公司环境.rar	2.57GB	2016-05-26 09:50
CentOS_64.rar	2.42GB	2016-05-26 09:48
Windows XP CleanVersion.rar	3.01GB	2015-07-23 17:11

我使用的 Go 编译器是 1.5 版本，也是目前最新的版本，下载地址是：[下载地址](#)

请选择 Go 语言开发文件夹，并将里面内容全部下载下来，我的是 64 位操作系统，如果您是 32 位操作系统，需要自己下载 32 位，搭建方法跟 64 位一模一样

文件名	大小	修改时间
go1.5.linux-amd64.tar.gz	74.25MB	2016-06-12 17:22
go.vim	3KB	2016-06-12 17:22
gocode-master.zip	139KB	2016-06-12 17:22

接下来我们开始搭建环境

1. 安装 Go 语言编译环境

1.1 执行 `tar -zxvf go1.5.linux-amd64.tar.gz` 命令解压文件

```
root@localhost:~/Desktop/go
File Edit View Search Terminal Help
[root@localhost go]# ls
go1.5.linux-amd64.tar.gz gocode-master.zip go.vim
[root@localhost go]# tar -zxvf go1.5.linux-amd64.tar.gz |
```

1.2 将解压生成的文件夹剪切到 /usr/local 目录下，这里也可以换成其他的文件夹，但是建议放在 /usr/local 下

```
[root@localhost go]# mv go /usr/local/ -f
```

1.3 新建 go 语言下载目录

执行命令 `mkdir /go_src`，该目录作为go语言下载组件的默认路径，当然，如果不想使用这个路径的话，完全可以自定义，在下面的步骤中记得改成你自己的路径即可。

1.3 设置环境变量

执行`vi /etc/profile`命令修改系统配置信息，添加以下内容

```
76
77 unset i
78 unset -f pathmunge
79 export PATH=$PATH:/usr/local/go/bin
80 export GOPATH=/go_src/
```

其中`export PATH=$PATH:/usr/local/go/bin`是将go语言编译器的路径添加到系统路径中，下面那一句是将我们上一步新建的目录添加到环境变量中，这个路径就会成为go语言下载组件的默认路径

1.4 使配置生效，执行`source /etc/profile`命令

```
File Edit View Search Terminal Help
[root@localhost go]# vi /etc/profile
[root@localhost go]# mkdir /go_src
[root@localhost go]# source /etc/profile
[root@localhost go]# |
```

1.5 好了现在go语言开发环境就基本搭建完成了，我们试一下下面几个命令

go

```
list      list packages
run       compile and run Go program
test      test packages
tool      run specified go tool
version   print Go version
vet       run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

c          calling between Go and C
buildmode  description of build modes
filetype   file types
gopath     GOPATH environment variable
environment environment variables
importpath import path syntax
packages   description of package lists
testflag   description of testing flags
testfunc   description of testing functions

Use "go help [topic]" for more information about that topic.
```

go version

```
[root@localhost go]# go version
go version go1.5 linux/amd64
[root@localhost go]#
```

1.6 我们在写一个测试程序

执行vi test.go输入如下代码

[plain] [view plain copy](#)

```
1. package main
2.
3. import "fmt"
4.
5. func main() {
6.     fmt.Printf("hello world...\n");
7. }
```

执行go build test.go命令开始编译，并生成test可执行文件，执行该文件即可

```
[root@localhost Desktop]# go build testgo.go
[root@localhost Desktop]# ls
go QtProject-qtcreator.desktop testgo testgo.go
[root@localhost Desktop]# ./testgo
hello world...
[root@localhost Desktop]#
```

2. 好了，当执行到这一步的时候，go语言的开发环境就搭建好了，但是我们在写代码的时候会发现go的语法不高亮，总是叫人觉得不习惯

```
File Edit View Search Terminal Help
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Printf("hello world...\n");
7 }
```

就像这样，那好，接下来我们设置一下语法高亮，刚才我们总共下载了三个文件，另外两个文件就是我们用来设置语法高亮的

2.1 解压gocode，执行unzip gocode-master.zip

```
[root@localhost Desktop]# cd go
[root@localhost go]# ls
go1.5.linux-amd64.tar.gz gocode-master.zip go.vim
[root@localhost go]# unzip gocode-master.zip
Archive:  gocode-master.zip
```

2.2 进入gocode-master/vim目录下，执行update.sh

```

[root@localhost go]# ls
go1.5.linux-amd64.tar.gz  gocode-master  gocode-master.zip  go.vim
[root@localhost go]# cd gocode-master
[root@localhost gocode-master]# ls
autocompletecontext.go  docs          nvim          rpc.go
autocompletefile.go    emacs         os_posix.go   scope.go
client.go              emacs-company os_windows.go server.go
config.go              formatters.go package_bin.go subl3
cursorcontext.go       _gccgo       package.go    _testing
debian                 gocode.go    package_text.go  utils.go
declcache.go           _goremove    README.md       vim
decl.go               LICENSE      ripper.go
[root@localhost gocode-master]# cd vim/
[root@localhost vim]# ls
autoload  ftplugin  pathogen_update.sh  symlink.sh  update.sh
[root@localhost vim]# ./update.sh
[root@localhost vim]# |

```

2.3 接下来我们将进入目录下的.vim文件夹下，查看一下文件内容(直接打cd ; cd .vim即可，该文件夹为隐身文件夹，ls不显示，但是ls -a可以)

```

root@localh
File Edit View Search Terminal Help
[root@localhost Desktop]# cd
[root@localhost ~]# cd .vim
[root@localhost .vim]# ls
autoload  ftplugin
[root@localhost .vim]# |

```

2.4 新建一个syntax文件夹，并将我们考北京来的go.vim文件放进来

```

root@localhost: ~/.vim
File Edit View Search Terminal Help
[root@localhost Desktop]# cd
[root@localhost ~]# cd .vim
[root@localhost .vim]# ls
autoload  ftplugin
[root@localhost .vim]# mkdir syntax
[root@localhost .vim]# cp ~/Desktop/go/go.vim ./syntax/
[root@localhost .vim]# |

```

2.5 回到用户根目录下(直接敲cd命令即可)，打开.vimrc文件，键入以下几句配置

[plain] [view plain copy](#)

1. filetype plugin on
2. syntax on
3. au BufRead,BufNewFile *.go set filetype=go

```
File Edit View Search Terminal Help
1 set nu
2 set mouse=a
3 set tabstop=4
4 set softtabstop=4
5 set shiftwidth=4
6 set autoindent
7 set cindent
8 set cinoptions={0,1s,t0,n-2,p2s(03s,=.5s,>1s,=1s,:1s
9 if &term=="xterm"
10     set t_Co=8 http://blog.csdn.net/
11     set t_Sb=^[4%dm
12     set t_Sf=^[3%dm
13 endif
14 filetype plugin on
15 syntax on
16 au BufRead,BufNewFile *.go set filetype=go
17
```

2.6 好了，重启终端，再次打开test.go文件，奇迹发生了

```
root@localhost:~/Desktop
File Edit View Search Terminal Help
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Printf("hello world...\n");
7 }
```