

GoLand

GoLand 是 JetBrains 公司推出的 Go 语言集成开发环境，基于 IntelliJ 平台开发，支持 JetBrains 的插件体系，可以在龙芯OpenJDK上比较好地运行。由于 GoLand 并没有发布社区版本，使用它需要购买相应的许可证。

运行所需配置和 IntelliJ IDEA 基本一致：如果运行最新的 GoLand 需要使用龙芯OpenJDK13并自行编译 fsNotifier □如果使用较旧的版本只需使用龙芯OpenJDK8并自行编译 fsNotifier 和 pty4j-native □具体可以参考 [IntelliJ IDEA](#) 页面进行安装和配置。

经测试，龙芯OpenJDK8可以正常运行的GoLand版本为 2019.3.4。

Go编译器

由于 Go 已经实现了自举，在这里新的 Go 编译器将使用旧的 Go 编译器编译。

从软件源中安装

Debian系

```
sudo apt-get update
sudo apt-get install go
```

RadHat系

```
sudo dnf update
sudo dnf install go
```

Arch系

```
sudo pacman -Syy
sudo pacman -S go
```

在 Fedora28 中最新的 go 版本为 1.12.7。

从源码构建

首先从 Go 的 [GitHub release](#) 页面下载所需合适的版本并解包：

```
tar -xf go.tar.gz
unzip go.zip
```

阅读 doc/install-source.html 并了解编译过程。编译过程非常简单，直接运行 src/all.bash 即可：

```
cd src
./all.bash
```

当终端打印 ALL TESTS PASSED 则预示着编译已经成功，二进制在 bin 目录中，之后可以配置环境变量。

如果需要将 Go 装在其他目录，请将整个目录迁移，而不是只迁移 bin 目录。如果只需要在 GoLand 中使用新编译获得的 Go 编译器，也可以不设置环境变量。

设置alternatives

设置前请参照已有的go配置，用下面的命令查看已有的配置：



```
$ alternatives --display go
```

或

```
$ update-alternatives --display go
```

这里以 Fedora28 为例，使用下面的命令安装，需要 sudo 提权：

```
# alternatives --install /usr/bin/go go /path/to/new/go 100 \
    --slave /usr/bin/gofmt gofmt /path/to/new/gofmt
```

用下面的命令重新选择 go 版本，需要 sudo 提权：

```
# alternatives --config go
```

Go Proxy

当 Go 下载第三方包时，由于网络环境的不同，可能会出现失败的情况。可以用设置环境变量的方式临时设置 Go Proxy

```
export GOPROXY="https://goproxy.io"
```

在 GoLand 中，可以在“File→Setting→Go→Go Modules(vgo)”中设置 proxy

GoLand

安装

从 JetBrains [官网页面](#) 下载最新的 GoLand 或者从 “[其他版本](#)” 下载较旧版本的 GoLand 经测试，龙芯 OpenJDK8 可以正常运行的 GoLand 版本为 2019.3.4。

将软件包解压并放置在你喜欢的地方，运行之并按提示进行默认配置：

```
cd bin
./goland.sh
```

配置编译器

在 “File→Setting→Go→GOROOT” 或 “Configure→Setting→Go→GOROOT” 选择编译器目录，注意这并不是选择二进制，也不是选择其 bin 目录，而是 bin 的上级目录。以 Fedora28 从软件源中安装的 go 为例，其目录为：

```
/usr/lib/golang
```

如果使用自行编译的 go 选择的应该是整个源码包目录。

运行hello, world

在 GoLand 欢迎界面选择 New Project 默认选择了 Go 工程，选择合适的 location 和 GOROOT 后单击 Create

工程创建后在 Project 窗口右键工程名或单击菜单栏的 File 选项，选择 New→Go File 键入合适的文件名。

新建 go 源码文件后，输入下面的 go 代码并保存：

```
package main

import "fmt"

func main() {
    fmt.Println("hello, world")
}
```

在主界面右上角，单击 Add Configuration... 添加构建配置，在 Run/Debug Configurations 界面点击左上角小加号→Go Build 先清空 Files 框后再选择前面创建的源码文件，单击 Apply 或 OK 此时可以看到新的构建配置已经添加成功，单击右边绿色的小箭头即可构建工程。单击构建，在 Run 窗口可以看到我们成功打印了 hello, world

From:

<https://wiki.chuang.ac.cn/> - 创学院百科

Permanent link:

https://wiki.chuang.ac.cn/loongson:programming_software_ide:goland

Last update: **2022/09/23 21:13**



