Atitit golang开发环境搭建

目录

[1. 编辑helo.go 1](#_Toc19735)

[2. Down sdk 100M,,,amd64 ver.... not i386 1](#_Toc18607)

[3. Ide选择liteide 1](#_Toc9916)

[3.1. 调试编译 2](#_Toc15813)

[3.2. 交叉编译linux下的程序 2](#_Toc14857)

[4. 问题解决 3](#_Toc16977)

[5. 附录 3](#_Toc11948)

[5.1. Go语言标准库常用的包及功能 3](#_Toc9394)

[5.2. golang调试工具delve 之前一直在烦心不知道怎么打印所有goroutine的stack，最近终于发现一个该工具。 5](#_Toc29002)

# 编辑helo.go

// gohelo

**package** main

**import** (

"fmt"

)

**func** main() {

fmt.Println("Hello World! attilaz")

}

# Down sdk 100M,,,amd64 ver.... not i386

# Ide选择liteide

Vscode但是不能delve调试放弃

Liteideok

Dbug>>> dbugr /devle >>f5 debug

## 调试编译

生产

**C:/BaiduYunDownload/liteide/x36.1/liteide/bin/dlv.exe --headless --api-version=2 --accept-multiclient exec C:/Users/aaa.ATTILAXPC188/aaa.ATTILAXPC188.debug.exe [C:/Users/aaa.ATTILAXPC188]**

time="2019-09-17T18:28:02+08:00" level=warning msg="CGO\_CFLAGS already set, Cgo code could be optimized." layer=dlv

**C:/BaiduYunDownload/liteide/x36.1/liteide/bin/dlv.exe connect 127.0.0.1:12351 [C:/Users/aaa.ATTILAXPC188]**

API server listening at: 127.0.0.1:12351

Hello World! attilaz

**Dlv server error! 进程崩溃或运行终止**

**Dlv server exited with code 62097**

**Dlv error! 进程崩溃或运行终止**

**Dlv exited with code 62097**

## 交叉编译linux下的程序

Tool>slect envi >cross linux 64bit

Tool>edit edit cur envi ,,then open cross-linux.env

# cross-compiler linux amd64

GOROOT=c:\go

#GOBIN=

GOARCH=amd64

GOOS=linux

CGO\_ENABLED=0

If def is system evni

System.env

# native compiler windows 386

#GOROOT=c:\go

#GOBIN=

#GOARCH=386

#GOOS=windows

#CGO\_ENABLED=1

此时admin dir hamyer zo yolei linux ver app

# 问题解决

1. 调试时unable to initialize decompress status for section .zdebug\_abbrev，工具栏 调试(D) -----> 选中 debugger/delve 将debugger 从 gdb 切换成 delve 就可以正常debug了

# 附录

|  |  |
| --- | --- |
| Go语言标准库常用的包及功能 | |
| **Go语言标准库包名** | **功  能** |
| bufio | 带缓冲的 I/O 操作 |
| bytes | 实现字节操作 |
| container | 封装堆、列表和环形列表等容器 |
| crypto | 加密算法 |
| database | 数据库驱动和接口 |
| debug | 各种调试文件格式访问及调试功能 |
| encoding | 常见算法如 JSON、XML、Base64 等 |
| flag | 命令行解析 |
| fmt | 格式化操作 |
| go | Go语言的词法、语法树、类型等。可通过这个包进行代码信息提取和修改 |
| html | HTML 转义及模板系统 |
| image | 常见图形格式的访问及生成 |
| io | 实现 I/O 原始访问接口及访问封装 |
| math | 数学库 |
| net | 网络库，支持 Socket、HTTP、邮件、RPC、SMTP 等 |
| os | 操作系统平台不依赖平台操作封装 |
| path | 兼容各操作系统的路径操作实用函数 |
| plugin | Go 1.7 加入的插件系统。支持将代码编译为插件，按需加载 |
| reflect | 语言反射支持。可以动态获得代码中的类型信息，获取和修改变量的值 |
| regexp | 正则表达式封装 |
| runtime | 运行时接口 |
| sort | 排序接口 |
| strings | 字符串转换、解析及实用函数 |
| time | 时间接口 |
| text | 文本模板及 Token 词法器 |

## golang调试工具delve 之前一直在烦心不知道怎么打印所有goroutine的stack，最近终于发现一个该工具。

1. 什么是delve

delve是golang推荐的专门go语言调试工具，用来替代gdb，因为：golang组织说delve能更好的理解go语言

Go语言和其它编程语言的对比.mhtml