用LiteIDE编译Go代码 -Windows版



独华

64174234@qq.com

内容

- 1.1 下载LiteIDE
- 1.2 安装LiteIDE
- 1.3 配置LiteIDE
- 1.4 使用LiteIDE编译Go代码
- 1.5 使用LiteIDE调试Go代码

1.1 下载LiteIDE

□ 下载URL: http://golangtc.com/download/liteide

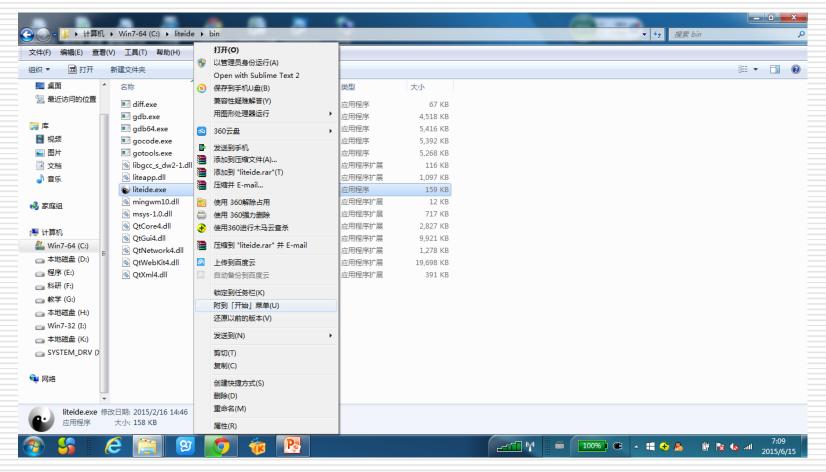


□ 这里选择下载

liteidex27.2.1.windows-qt4.zip

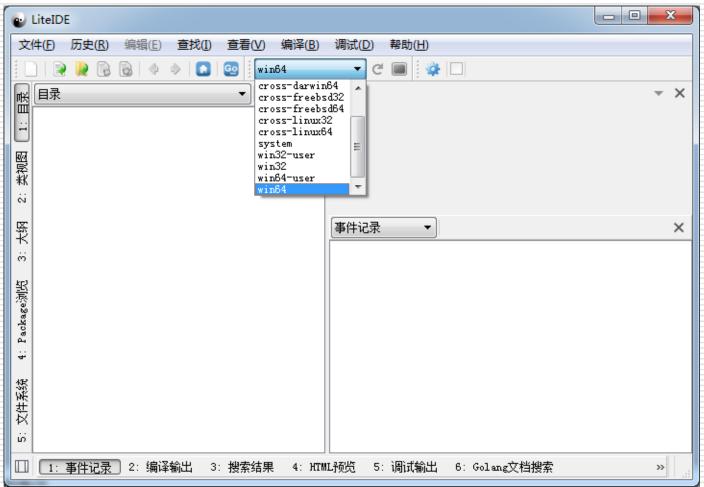
1.2 安装LiteIDE

- □ 解压liteidex27.2.1.windows-qt4.zip到C:\即
- 山 进入到C:\liteide\bin,在liteide.exe上右键选择
 附到开始菜单,方便从开始菜单中打开LiteIDE



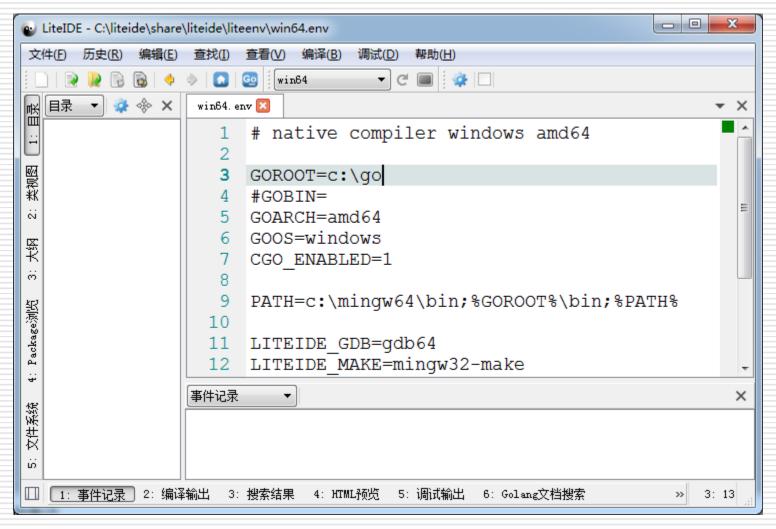
1.3 配置LiteIDE

- □ 打开LiteIDE
- 口 在工具栏上选择平台环境,这里是64位Win7, 所以选择win64



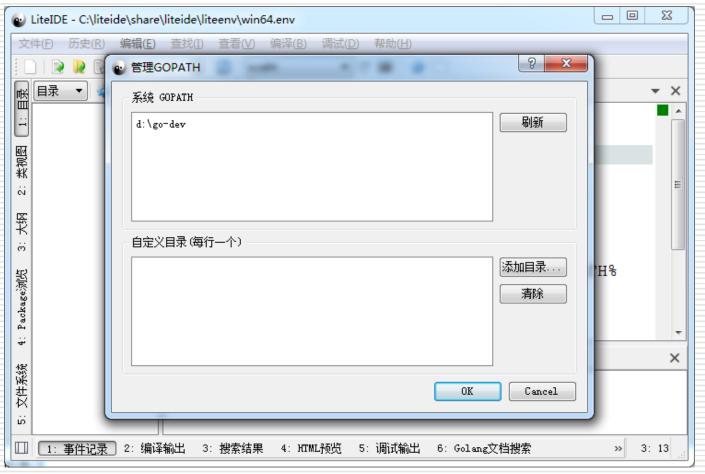
1.3 配置LiteIDE(续)

□ 选择查看\编辑当前环境,确认GOROOT与Go 编译环境安装路径一致,不一致则修改正确



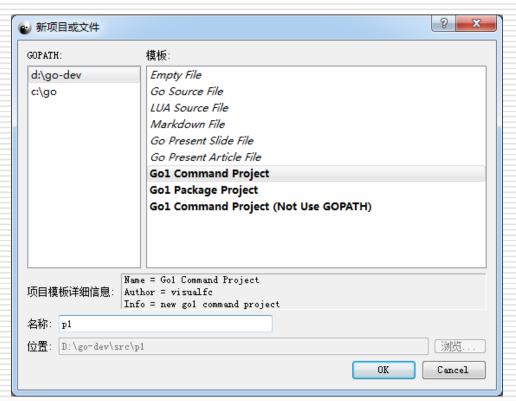
1.3 配置LiteIDE(续)

- □ 选择查看\管理GOPATH,图中,自动读出取已 经设置好的开发路径为d:\go-dev
- 口 也可以点击添加目录,自定义开发目录。



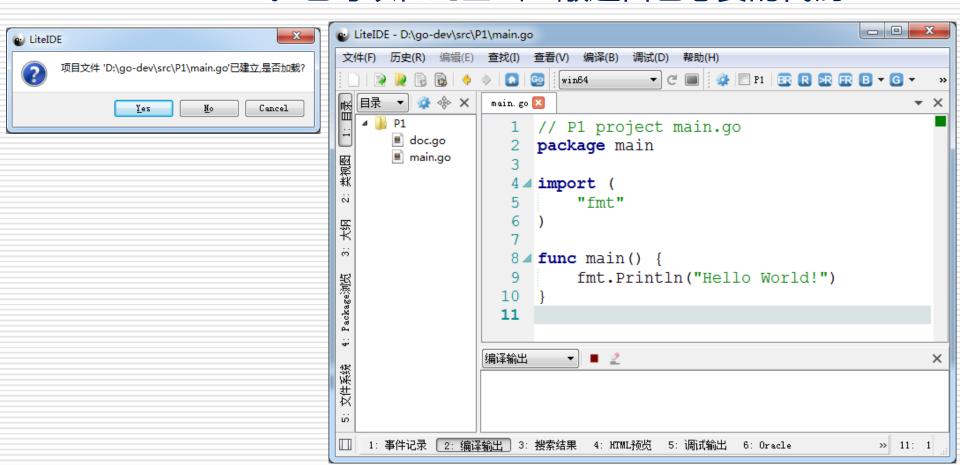
1.4 使用LiteIDE编译Go代码

- □ 建立项目
 - 选择文件\新建,弹出新建对话框
 - GOPATH目录选择d:\go-dev
 - 模板选择 Go1 Command Project
 - 名称这里定为P1,然后点击OK按钮。



1.4 使用LiteIDE编译Go代码(续)

- □ 建立Go源文件
 - 上一步点击OK后,会提示如下对话框,选择Yes,自动建立一个main.go的项目文件,功能是输出 Hello World!。也可以在此基础上敲进自己想要的代码



1.4 使用LiteIDE编译Go代码(续)

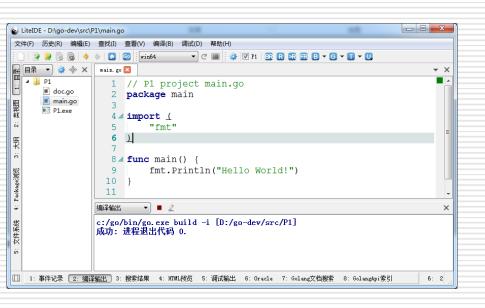
- □ 编译并输出,常用方法有如下三种
 - 选择编译菜单下的 BuildAndRun
 - 按ctrl+R 或ctrl+F7

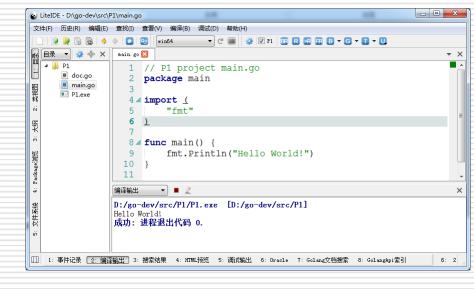
■ 点击工具栏上的BR按钮

```
LiteIDE - D:\go-dev\src\P1\main.go
文件(F) 历史(R) 编辑(E) 查找(I) 查看(V) 编译(B) 调试(D) 帮助(H)
                                        🔻 C 🔳 🕍 📝 P1 | 🔛 R 🖫 🖫 B 🔻 G 🕶 🚺 🕶 🕕
                            win64
監
  目录
                   main. go 🔣
                                                                                              ▼ X
      P1
                         // P1 project main.go
       doc.go
                         package main
       main.go
米洛图
      P1.exe
                         import (
ä
                              "fmt."
聚
ö
                      8 d func main() {
                              fmt.Println("Hello World!")
Package巡迟
                    10
                    11
                  编译输出
                             ▼ ■ 2
                                                                                                 ×
文件系统
                  c:/go/bin/go.exe build -i [D:/go-dev/src/P1]
                  成功: 进程退出代码 0.
                  D:/go-dev/src/P1/P1.exe [D:/go-dev/src/P1]
                  Hello World!
ö
                  成功: 进程退出代码 0.
   - 1: 事件记录 【2: 编译输出 】 3: 搜索结果 - 4: HTML预览 - 5: 调试输出 - 6: Oracle - 7: Golang文档搜索 - 8: GolangApi索引
                                                                                            11: 1
```

1.4 使用LiteIDE编译Go代码(续)

- □ 先编译:选择编译菜单下的Build、按ctrl+B、点击工具栏上的B按钮都可以完成编译工作
- □ 后执行:选择编译菜单下的Run、按ctrl+Alt+R、 点击工具栏上的R按钮 都可以完成执行工作





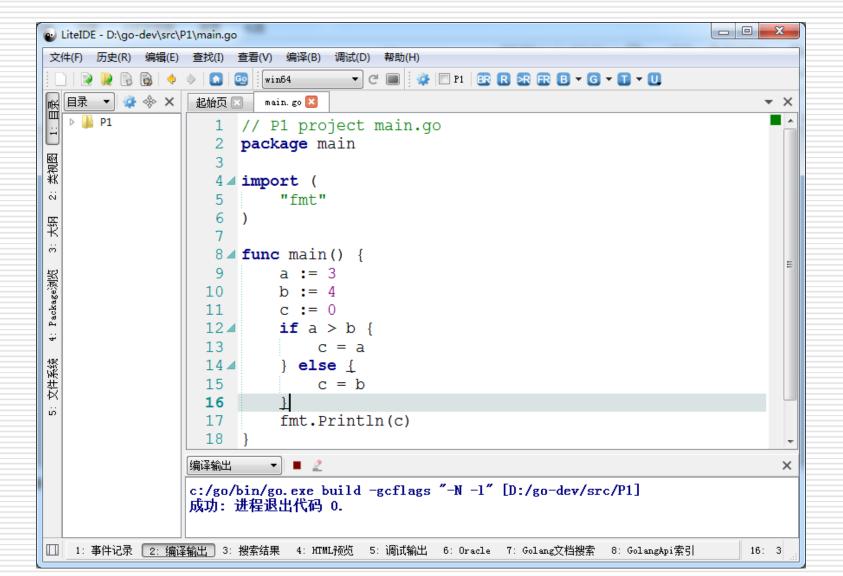
编译生成可执行文件

执行可执行文件

- □ 设置编译选项
 - 选择编译\编译配置,在弹出的对话框中,切换到自定义面板,设置BUILDARGS的值为-gcflags "-N -I",目的是去掉编译优化,方便调试。最后点击OK

编译配置 编译 ID [go tool 编译目录 [D:/go-dev/	/src/P1	X
LiteIDE 编译	自定义	
名称	值	^
TARGETARGS		
BUILDARGS	-gcflags "-N -1"	=
FORCEBUILDARGS	-i -a -v	
INSTALLARGS		
TESTARGS	-v	
BENCHMARKARGS		+

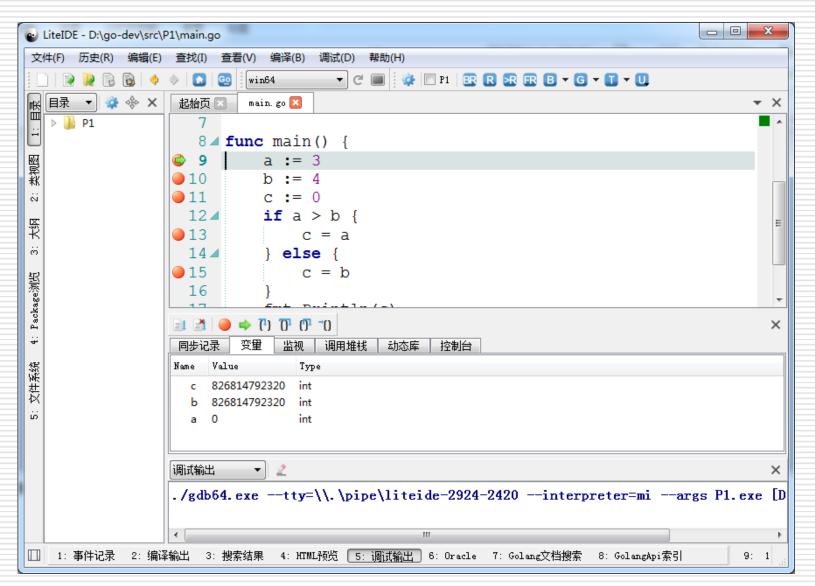
□ 按ctrl+B完成编译



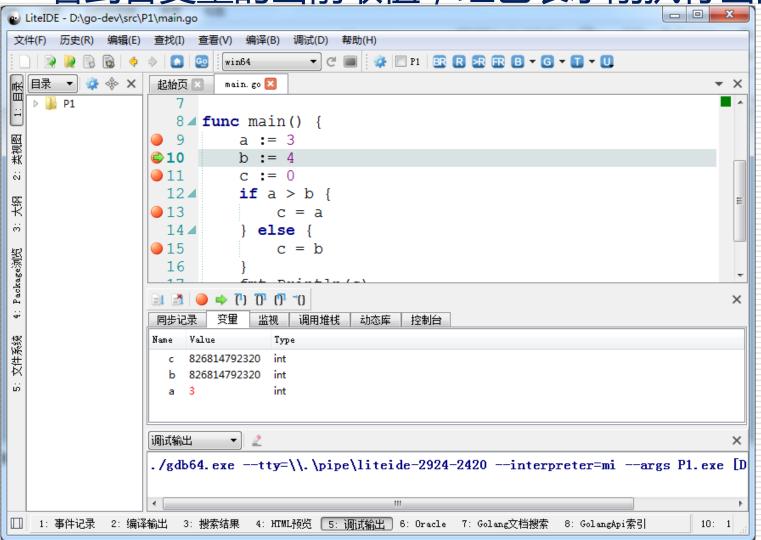
□ 在想调试的代码行上,按F9设置断点

```
- - X
LiteIDE - D:\go-dev\src\P1\main.go
文件(F) 历史(R) 编辑(E) 查找(I) 查看(V) 编译(B)
                                   调试(D)
                                        帮助(H)
                                     🔻 C 🔳 🔯 🔚 P1 | BR Ŗ 🖫 B 🕶 🕞 🔻 🚺 🕶 🕕
  ] 📦 📭 🕞 📵
                          win64
         🌞 🚸 🗙
  目录
                  起始页 🖂
                                                                                       ▼ X
                          main. go 🔣
   ▶ № P1
                      // P1 project main.go
                       package main
                    4 ⊿ import (
ö
                            "fmt"
                    6
ö
                    8 ⊿ func main() {
                            a := 3
                            b := 4
                 10
                 11
                            c := 0
                   12 ◢
                            if a > b {
                 13
                                c = a
女 在 聚%
                            } else {
                   14 ◢
                 15
                                c = b
                   16
                            fmt.Println(c)
                   17
                   18
                 编译输出
                                                                                         ×
                 c:/go/bin/go.exe build -gcflags "-N -1" [D:/go-dev/src/P1]
                 成功: 进程退出代码 0.
  1:事件记录 2:编译输出 3:搜索结果 4:HTML预览 5:调试输出 6:Oracle 7:Golang文档搜索 8:GolangApi索引
                                                                                      9: 1
```

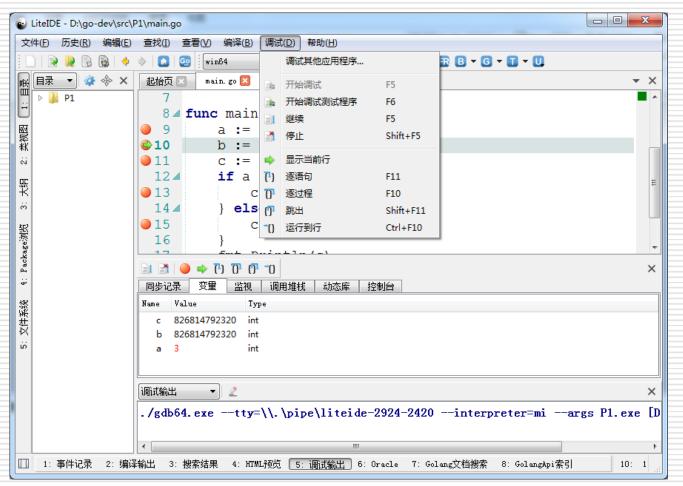
口 按F5执行到第一个断点处



继续按F5执行到下一个断点处,可以在变量视窗中 看到各变量的当前取值,红色表示刚执行出的结果



- □ 还可以按F10单步调试,可以不受断点控制,从当前即将执行的代码行开始,一次一行的顺序执行
- □ 各种调试命令,点击调试菜单即可看到并具体使用



Thank you very much

Any comments and suggestions are beyond welcome