

# 使用 go语言 golang 编译 dll 给易语言使用支持多线程

之前发过一份，但是大家提到多线程问题，因此重新修改了一份源码，好支持多线程。

## 一、必要准备

- 必须导入 `import "C"` 并且要单独一行，在其上方的注释中必须写上 `#include <stdlib.h>`;
- 这是在导入 `C` 的标准动态库头文件，若不导入将有几率编译失败。

## 二、形参与返回值

- 在go中，除了 `string` 特殊外，其他 `int`、`bool`，等基本类型原本怎样还是怎样；
- 传 `string` 与返回值 `string` 都改成 `*C.char` 类型，其他基本类型不用改；
- 有三个方法比较重要，`C.CString` 转成c字符串，`C.GoString` 转成go字符串，`C.free` 释放内存；
- 只要用到 `C.CString` 此方法，就必须记得释放内存。

## 三、内存泄漏

- 如果使用了 `C.CString` 却不使用 `C.free`，内存暂用只会越来越大，最后崩溃；
- 释放内存时，请不要重复取地址，例如 `unsafe.Pointer(&xx变量)`，这样等于没释放；
- 也可能是 `vc6` 的原因，使用 `defer` 在即将出栈时释放，会造成易语言得不到返回值；
- 解决方法，返回c字符串的同时，也将需要释放的指针传出去，再定义一个释放函数例如 `Free()` 用于释放！

## 四、如何编译

- 安装 `tdm-gcc` 编译器，可选择64位，依然可以编译出32位，下载地址：  
<https://jmeubank.github.io/tdm-gcc/download/>；
- 若想彻底解决 `gcc` 兼容性问题建议下载 `msys2` 后再安装 `gcc`，此条为建议并不一定需要；
- 易语言只支持32位dll，使用64位会出错，例如找不到此函数，因此下面两项设置尤为重要；
- 用 `cmd` 执行两项设置：`set GOARCH=386`，`set CGO_ENABLED=1`，每次打开新的 `cmd` 都要重新设置；
- 编译命令：`go build -ldflags "-s -w" -buildmode=c-shared -o dlldemo.dll dlldemo.go`。

## 五、如何调用

- 在填写dll命令时，请在填写，在库中对应命令名时，前面加个 `@`，不然会出现栈错误；
- 每次调用返回值是文本型dll命令时，请都使用前面准备的 `Free()` 释放内存！