golang syscall 例子

作者：kuuyee

原文链接：https://ld246.com/article/1446417162636

来源网站：链滴

获取Windows的系统默认目录

Windows系统目录函数请参考MSDN：https://msdn.microsoft.com/en-us/library/windows/desktop/bb762181%28v=vs.85%29.aspx

```
package main

import (
    "fmt"
    "syscall"
    "unsafe"
)

var (
    shell         = syscall.MustLoadDLL("Shell32.dll")
    getFolderPath = shell.MustFindProc("SHGetFolderPathW")
)

const (
    CSIDL_DESKTOP = 0    //用户桌面默认目录
    CSIDL_APPDATA = 26    //用户AppData目录
)

func main() {
    b := make([]uint16, syscall.MAX_PATH)

    // https://msdn.microsoft.com/en-us/library/windows/desktop/bb762181%28v=vs.85%29.apx
    // 这里第二个参数CSIDL_A必须定义为const，否则编译报错
    r, _, err := getFolderPath.Call(0, CSIDL_DESKTOP, 0, 0, uintptr(unsafe.Pointer(&b[0])))
    if uint32(r) != 0 {
        fmt.Sprintf("获取DIR错误：", err)
    }
    a_dir := syscall.UTF16ToString(b)

    r, _, err = getFolderPath.Call(0, CSIDL_APPDATA, 0, 0, uintptr(unsafe.Pointer(&b[0])))
    if uint32(r) != 0 {
        fmt.Sprintf("获取DIR错误：", err)
    }
    b_dir := syscall.UTF16ToString(b)

    fmt.Printf("目录ID：%d  目录地址：%s\n", CSIDL_DESKTOP, a_dir)
    fmt.Printf("目录ID：%d  目录地址：%s\n", CSIDL_APPDATA, b_dir)
}
```

speech_balloon **SHGetFolderPathW的第二个参数必须定义为const，否则编不过**

输出

```
目录ID：0  目录地址：C:\Users\kuuyee\Desktop
```

原文链接：golang syscall 例子

目录ID：26 目录地址：C:\Users\kuuyee\AppData\Roaming

原文链接：golang syscall 例子