下面两个函数主要用于将 Windows 路径分隔符转换为 Linux 路径分隔符，处理完后再转换回去，只在 Windows 中有用，在 Linux 中没必要：

**1. ToSlash**

函数：func ToSlash(path string) string

释义: 将 path 中平台相关的路径转换为 '/'

s = filepath.ToSlash(s)

### 2. FromSlash

函数：func FromSlash(path string) string

释义: 将 path 中的 '/' 转换为系统相关的路径分隔符

//取 路径 的 目录

Dir 返回路径中除去最后一个路径元素的部分，即该路径最后一个元素所在的目录。

//取 路径 的 文件名

Base 函数返回路径的最后一个元素。在提取元素前会去掉末尾的斜杠。如果路径是 ""，会返回 "."；如果路径是只有一个斜杆构成的，会返回 "/"。

### 5. Split

函数：func Split(path string) (dir, file string)

释义: 获取path中最后一个分隔符前后的两部分

例子：

1. // 5、Base 获取 path 中最后一个分隔符前后的两部分
2. // 之前包含分隔符，之后不包含分隔符
3. s = "/golang\_workspace/leetcode/aa.js"
4. d, s := filepath.Split(s)
5. fmt.Println("Split: ", d, s)
6. // Split: /golang\_workspace/leetcode/ aa.js

### 6. Ext

函数：func Ext(path string) string

释义: 获取路径字符串中的文件扩展名

例子：

1. // 6、Base 获取路径字符串中的文件扩展名
2. s = "/golang\_workspace/leetcode/aa.js"
3. s = filepath.Ext(s)
4. fmt.Println("Ext: ", s)
5. // Ext: .js

### 8. Join

函数：func Join(elem ...string) string

释义: 将elem中的多个元素合并成一个路径，忽略空元素，清理多余字符

例子：

1. // 8、Join 将 elem 中的多个元素合并为一个路径，忽略空元素，清理多余字符。
2. s = "golang\_workspace"
3. s2 = "leetcode/aa.js"
4. s = filepath.Join(s,s2)
5. fmt.Println("Join: ", s)
6. // Join: golang\_workspace\leetcode\aa.js

### 14. Match

函数：func Match(pattern, name string) (matched bool, err error)

释义: 根据pattern来判断name是否匹配，如果匹配则返回true

例子：

1. // 14、根据pattern来判断name是否匹配，如果匹配则返回true
2. r := "/home/catch/\*"
3. s = "/home/catch/foo"
4. f, \_ = filepath.Match(r, s)
5. fmt.Println("Match: ", f)
6. // Match: true

### 15. Glob

函数：func Glob(pattern string) (matches []string, err error)

释义: 列出与指定的模式 pattern 完全匹配的文件或目录（匹配原则同上）

例子：

1. // 15、列出与指定的模式 pattern 完全匹配的文件或目录（匹配原则同match）
2. r = "F:\\golang\_workspace\\[s]\*"
3. sList,\_ = filepath.Glob(r)
4. fmt.Println("Glob: ", sList)
5. // Glob: [F:\golang\_workspace\shenqi\_server F:\golang\_workspace\src]

### 16. Walk

函数：func Walk(root string, walkFn WalkFunc) error

释义: 遍历指定目录(包括子目录)，对遍历的项目用walkFn函数进行处理

例子：

1. // 16、遍历指定目录(包括子目录)，对遍历的项目用walkFn函数进行处理
2. pwd,\_ := os.Getwd()
3. filepath.Walk(pwd,func(fpath string, info os.FileInfo, err error) error {
4. if match,err := filepath.Match("???",filepath.Base(fpath)); match {
5. fmt.Println("Walk path:",fpath)
6. fmt.Println("Walk info:",info)
7. return err
8. }
9. return nil
10. })
11. // Walk path: F:\golang\_workspace\GoLearn\src