链滴

Golang 获取用户 home 目录路径

作者：88250

原文链接：https://ld246.com/article/1424065592565

来源网站：链滴

os/user

一般情况下我们可以使用 os/user 包提供的 Current() 函数来获取用户信息：

```
<pre class="prettyprint">
user, err := user.Current()
if nil == err {
    return user.HomeDir, nil
}
</pre>
```

但这个方式交叉编译后不能完全跨平台，在 darwin 下需要 cgo 才能正常工作。

改进

为了解决这个问题，我们需要进行一点增强，在通过 os/user 获取失败时再通过环境变量、命令来获：

```
<pre class="prettyprint">
// Home returns the home directory for the executing user.
//
// This uses an OS-specific method for discovering the home directory.
// An error is returned if a home directory cannot be detected.
func Home() (string, error) {
    user, err := user.Current()
    if nil == err {
        return user.HomeDir, nil
    }

    // cross compile support

    if "windows" == runtime.GOOS {
        return homeWindows()
    }

    // Unix-like system, so just assume Unix
    return homeUnix()
}

func homeUnix() (string, error) {
    // First prefer the HOME environmental variable
    if home := os.Getenv("HOME"); home != "" {
        return home, nil
    }

    // If that fails, try the shell
    var stdout bytes.Buffer
    cmd := exec.Command("sh", "-c", "eval echo ~$USER")
    cmd.Stdout = &stdout
    if err := cmd.Run(); err != nil {
        return "", err
    }
```

```
    result := strings.TrimSpace(stdout.String())
    if result == "" {
        return "", errors.New("blank output when reading home directory")
    }

    return result, nil
}

func homeWindows() (string, error) {
    drive := os.Getenv("HOMEDRIVE")
    path := os.Getenv("HOMEPATH")
    home := drive + path
    if drive == "" || path == "" {
        home = os.Getenv("USERPROFILE")
    }
    if home == "" {
        return "", errors.New("HOMEDRIVE, HOMEPATH, and USERPROFILE are blank")
    }

    return home, nil
}
</pre>
```

参考

- Obtain user's home directory
- go-homedir
- 一些有用的工具函数