



链滴

# java 学习应用篇 | 环境变量的应用

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1600700959012>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 环境变量综述

一直以来，我们好像对环境变量的使用，一直停留在配置JDK环境变量，配置Maven环境变量，配置Gradle环境变量等等，但其实环境变量的作用，远远不止于此，在本文中，小刀简述一下几个小功能的路作为抛砖引玉，大家有好的想法和实践欢迎随时和小刀交流。

## 环境变量做工具的思路

The screenshot shows the Windows 'System Properties' dialog box, specifically the 'Environment Variables' tab. It displays two sections: 'User variables for lixiang' and 'System variables'. In the 'User variables' section, the 'TEST\_USER' variable is highlighted with a red circle and labeled '1', with its value 'lixiang'. In the 'System variables' section, the 'A\_TEST\_SYSTEM' variable is highlighted with a red circle and labeled '2', with its value 'system\_lixiang'. A third red circle labeled '3' is placed over the command prompt output showing the values of these variables.

```
Microsoft Windows [版本 10.0.20180.1000]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\Users\lixiang>echo %TEST_USER%
lixiang

C:\Users\lixiang>echo %A_TEST_SYSTEM%
system_lixiang

C:\Users\lixiang>
```



```
lixiang@Dandelion: ~/code/java/sunshine
lixiang@Dandelion:~/code/java/sunshine$ echo $GP
git pull
lixiang@Dandelion:~/code/java/sunshine$ $GP
Already up to date.
lixiang@Dandelion:~/code/java/sunshine$ $GP
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 668 bytes | 334.00 KiB/s, done.
From https://github.com/xiaodaojava/sunshine
   b454bdc..d16c5f2  master    -> origin/master
Updating b454bdc..d16c5f2
Fast-forward
 README.md | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
lixiang@Dandelion:~/code/java/sunshine$
```

我们对先用echo 对刚才的变量进行输出，然后再用\$GP来执行一下，就可以看到是等同于 git pull的结果

这个例子之后，小伙伴们就可以扩宽思路，就不仅仅是Git命令了，一些常用的命令都可以这样简化，再有甚者，可以编一套指令，如

G开头的代表GIT命令， GP(git pull),GPH(git push)

M开头的代表Maven命令,MP(mvn package)

GD开头的代表Gradle命令， GDJAR(gradle bootjar)

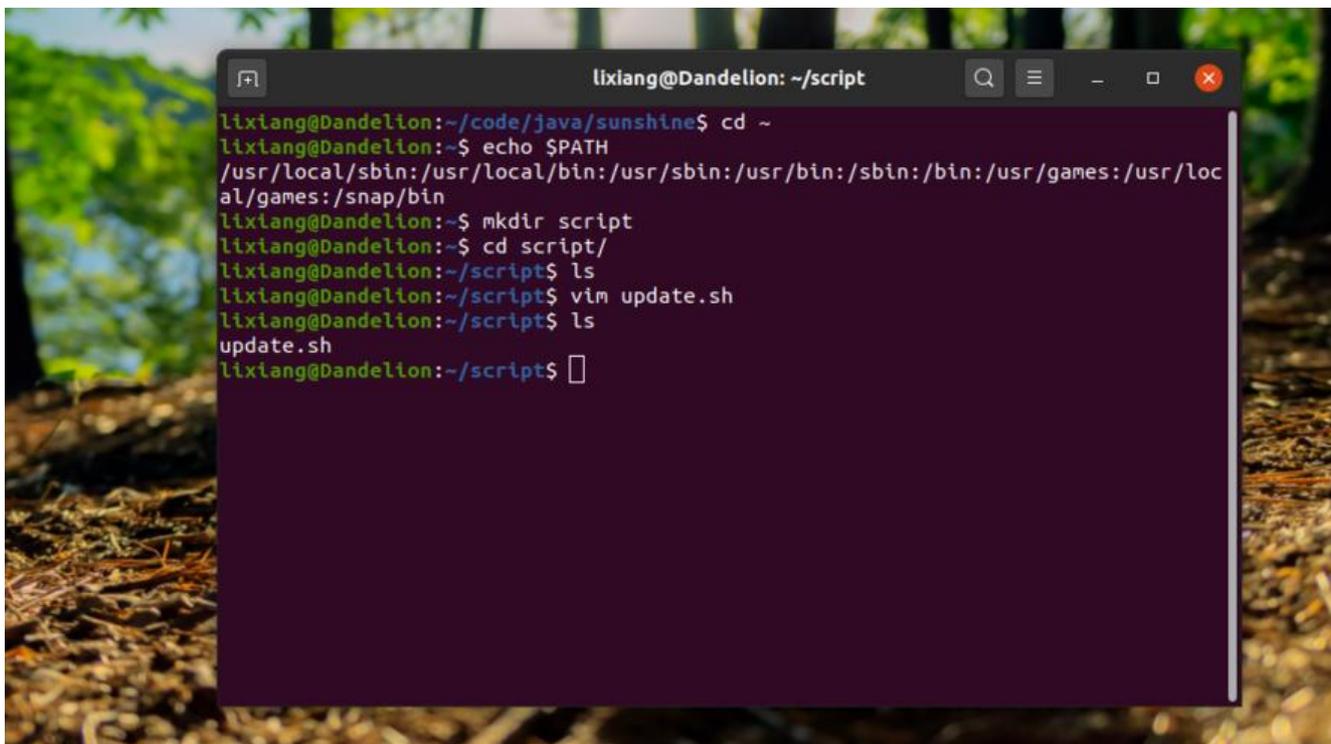
S开头的代表SSH命令， STEST1(ssh root@test1)

再者就要靠小伙伴自己发挥了

## 用环境变量执行脚本

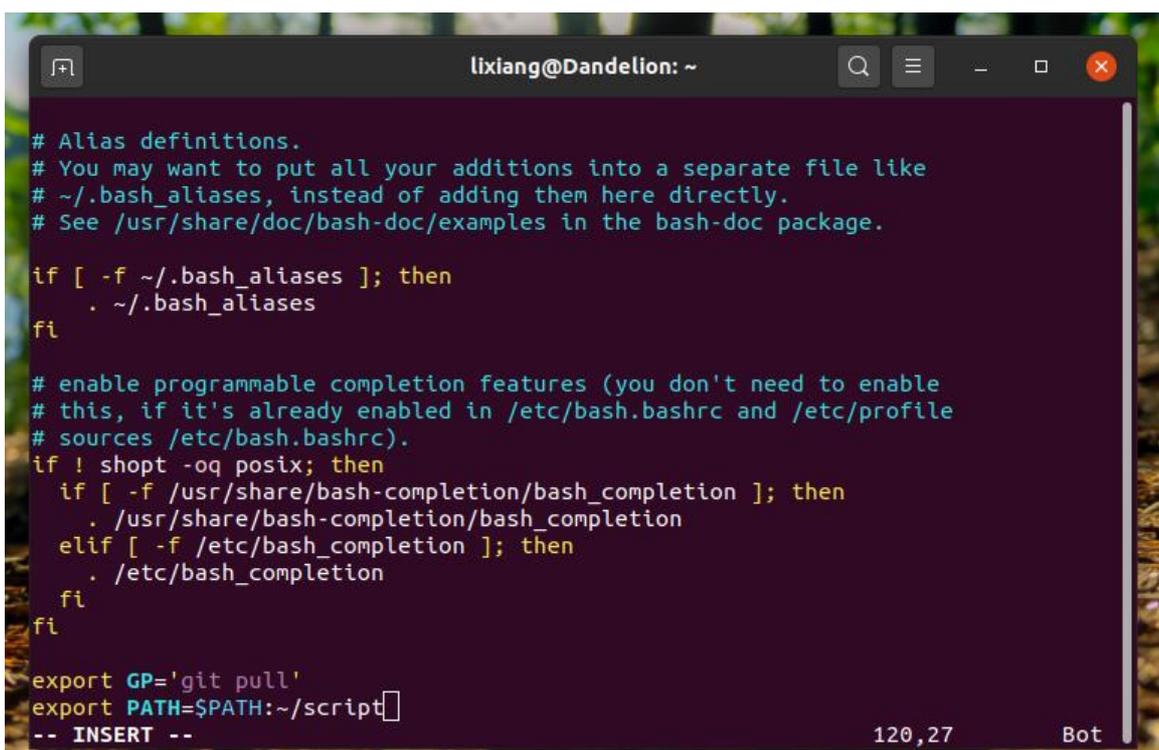
我们要引入一个新的环境变量值 PATH, 对其直接输出可以看到一系列的bin目录



A terminal window titled 'lixiang@Dandelion: ~/script' showing a series of commands and their outputs. The user navigates to the home directory, prints the current PATH, creates a 'script' directory, moves into it, and lists its contents, which now includes 'update.sh'.

```
lixiang@Dandelion:~/code/java/sunshine$ cd ~
lixiang@Dandelion:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
lixiang@Dandelion:~$ mkdir script
lixiang@Dandelion:~$ cd script/
lixiang@Dandelion:~/script$ ls
lixiang@Dandelion:~/script$ vim update.sh
lixiang@Dandelion:~/script$ ls
update.sh
lixiang@Dandelion:~/script$
```

现在的目录情况如上所示，然后我们把目录~/script也加入到PATH中,在~/.bashrc中编辑如下

A terminal window titled 'lixiang@Dandelion: ~' showing the contents of the ~/.bashrc file. The user has added a line to the PATH variable to include the ~/script directory. The terminal shows the file's content with the new line highlighted in blue.

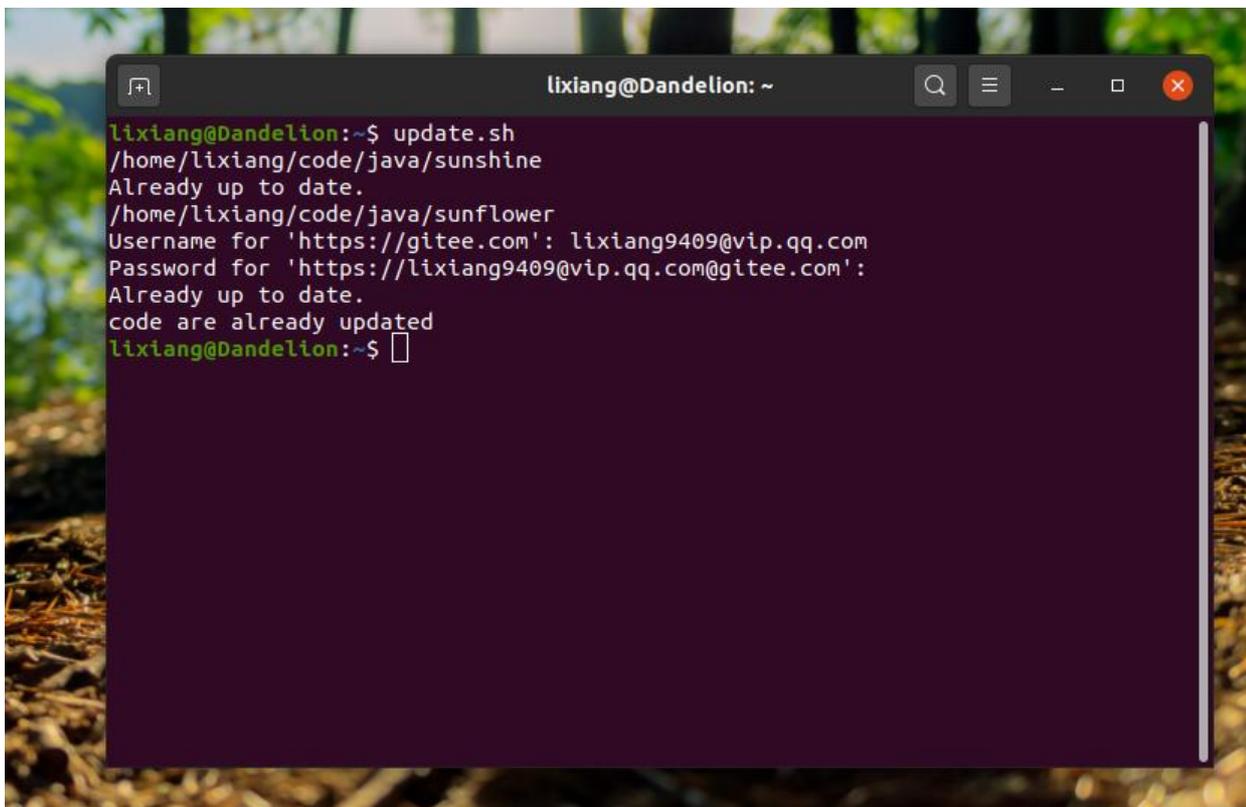
```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export GP='git pull'
export PATH=$PATH:~/script
-- INSERT --
```

保存之后，重新打开终端或者source ~/.bashrc，然后直接执行update.sh

A terminal window titled 'lixiang@Dandelion: ~' with a search icon, menu icon, and window control buttons. The terminal output shows the execution of 'update.sh' script. It checks for updates in two directories: '/home/lixiang/code/java/sunshine' and '/home/lixiang/code/java/sunflower'. For the first directory, it reports 'Already up to date.'. For the second directory, it prompts for a Gitee.com username and password, both of which are reported as 'Already up to date.'. The final output is 'code are already updated' and the prompt returns to 'lixiang@Dandelion:~\$' with a cursor.

```
lixiang@Dandelion:~$ update.sh
/home/lixiang/code/java/sunshine
Already up to date.
/home/lixiang/code/java/sunflower
Username for 'https://gitee.com': lixiang9409@vip.qq.com
Password for 'https://lixiang9409@vip.qq.com@gitee.com':
Already up to date.
code are already updated
lixiang@Dandelion:~$
```

可以看到，通过脚本，把我们电脑中的代码都更新了一遍，这个是很简单的脚本，我们可以再写的杂一点，比如，一键打开IDEA,SwitchHosts,DataGrip,网易云,等工具。做个自定义启动。

再比如，可以写一个备份资料，文件夹的脚本，直接输入一个back.sh就可以完成备份了。

## 扩展思考

- 1: 对应的windows环境变量工具应该怎么实现?
- 2: PATH变量有什么用?

## 推荐阅读

[java学习原理篇|多用户及环境变量](#)

[java学习应用篇|windows安装JDK及配置环境变量](#)

[和大家聊一聊学习\(二\)-如何去模仿学习](#)

[回到最开始，想和大家聊一聊怎么学习](#)

## 关于小刀

1. 自封的初级java劝退师
2. 自封的java交流群管理员
3. 自封的java学习微信客服小哥
4. 自封的微信号: best396975802