<https://blog.csdn.net/zqixiao_09/article/details/50340153>

环境变量一般是指在操作系统中用来指定操作系统运行环境的一些参数，比如临时文件夹位置和系统文件夹位置等等。

**一、Linux的变量种类**

      按变量的生存周期来划分，Linux变量可分为两类：

      1、永久的：需要修改配置文件，变量永久生效。

      2、临时的：使用export命令声明即可，变量在关闭shell时失效。

**二、设置变量的三种方法**

1、在/etc/profile文件中添加变量【对所有用户生效（永久的）】

      用VI在文件/etc/profile文件中增加变量，该变量将会对Linux下所有用户有效，并且是“永久的”。

      例如：编辑/etc/profile文件，添加PATH变量

      # vi /etc/profile

       export  PATH=/home/fs : $PATH

  注：修改文件后要想马上生效还要运行# source /etc/profile不然只能在下次重进此用户时生效。

2、在用户目录下的.bash\_profile文件中增加变量【对单一用户生效（永久的）】

      用VI在用户目录下的.bash\_profile文件中增加变量，改变量仅会对当前用户有效，并且是“永久的”。

      例如：编辑guok用户目录（/home/guok）下的.bash\_profile

      $ vi /home/guok/.bash.profile

      添加如下内容：

      export CLASSPATH=./JAVA\_HOME/lib;$JAVA\_HOME/jre/lib

      注：修改文件后要想马上生效还要运行$ source /home/guok/.bash\_profile不然只能在下次重进此用户时生效。

3、直接运行export命令定义变量【只对当前shell（BASH）有效（临时的）】

      在shell的命令行下直接使用[export 变量名=变量值]

      定义变量，该变量只在当前的shell（BASH）或其子shell（BASH）下是有效的，shell关闭了，变量也就失效了，再打开新shell时就没有这个变量，需要使用的话还需要重新定义。

**三、PATH声明，其格式为：**

      PATH=$PATH:<PATH 1>:<PATH 2>:<PATH 3>:------:<PATH N>

      你可以自己加上指定的路径，中间用冒号隔开。环境变量更改后，在用户下次登陆时生效。

      如果想立刻生效，则可执行下面的语句：$source .bash\_profile

      需要注意的是，最好不要把当前路径”./”放到PATH里，这样可能会受到意想不到的攻击。

      完成后，可以通过$ echo $PATH查看当前的搜索路径。这样定制后，就可以避免频繁的启动位于shell搜索的路径之外的程序了。

**四、常用的环境变量**

　　PATH      决定了shell将到哪些目录中寻找命令或程序

　　HOME      当前用户主目录

　　HISTSIZE　历史记录数

　　LOGNAME   当前用户的登录名

　　HOSTNAME　指主机的名称

　　SHELL 　　当前用户Shell类型

　　LANGUGE 　语言相关的环境变量，多语言可以修改此环境变量

　　MAIL　　　当前用户的邮件存放目录

　　PS1　　　基本提示符，对于root用户是#，对于普通用户是$

**五、常用的环境变量相关命令**

1. 显示环境变量HOME

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ echo $HOME
2. /home/fs
3. fs@ubuntu:~$

2. 设置一个新的环境变量hello

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ export HELLO="Hello"
2. fs@ubuntu:~$ echo $HELLO
3. Hello
4. fs@ubuntu:~$

3. 使用env命令显示所有的环境变量

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ env
2. SSH\_AGENT\_PID=2427
3. GPG\_AGENT\_INFO=/tmp/keyring-Sqfg93/gpg:0:1
4. TERM=xterm
5. SHELL=/bin/bash
6. XDG\_SESSION\_COOKIE=689f5a37acfced492491d99f00000008-1450313888.771442-154751925
7. HELLO=Hello
8. WINDOWID=62914565
9. OLDPWD=/home/fs/qiang/shell
10. GNOME\_KEYRING\_CONTROL=/tmp/keyring-Sqfg93
11. USER=fs
12. ....

4. 使用set命令显示所有本地定义的Shell变量

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ set
2. BASH=/bin/bash
3. BASH\_VERSINFO=([0]="2"[1]="05b"[2]="0"[3]="1"[4]="release"[5]="i386-redhat-linux-gnu")
4. BASH\_VERSION='2.05b.0(1)-release'
5. COLORS=/etc/DIR\_COLORS.xterm
6. COLUMNS=80
7. DIRSTACK=()
8. DISPLAY=:0.0
10. ...

5. 使用unset命令来清除环境变量

set可以设置某个环境变量的值。清除环境变量的值用unset命令。如果未指定值，则该变量值将被设为NULL。示例如下：

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ export TEST="Test" \\增加一个环境变量TEST
2. fs@ubuntu:~$ env | grep TEST \\此命令有输出，证明环境变量TEST已存在
3. TEST=Test
4. fs@ubuntu:~$ unset $TEST \\删除环境变量TEST
5. fs@ubuntu:~$ env | grep TEST \\此命令没输出，证明环境变量TEST已经存在了

6. 使用readonly命令设置只读变量

如果使用了readonly命令的话，变量就不可以被修改或清除了。示例如下：

**[cpp]** [view plain](https://blog.csdn.net/zqixiao_09/article/details/50340153) [copy](https://blog.csdn.net/zqixiao_09/article/details/50340153)

1. fs@ubuntu:~$ export TEST="Test" \\增加一个环境变量TEST
2. fs@ubuntu:~$ readonly TEST \\将环境变量TEST设为只读
3. fs@ubuntu:~$ unset TEST \\此变量无法删除
4. bash: unset: TEST: cannot unset: readonly variable
5. fs@ubuntu:~$ TEST="NEW" \\此变量不可更改
6. bash: TEST: readonly variable
7. fs@ubuntu:~$

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/zqixiao\_09/article/details/50340153