

# 什么是环境变量 (environment variables)

---

系统中的环境变量与编程语言中得环境变量，在性质上相同，但实际功能不同。

## 作用

---

操作系统中用来指定操作系统运行环境的一些参数，如：临时文件夹位置和系统文件夹位置等,可以理解  
为系统的视线范围（开机就初始化在内存中）。配置了环境变量的可执行程序，就等于是进入了系统的  
视线范围（提前存在内存中，使用时直接找到）。

## Linux 中的环境变量

---

Linux 中的环境变量与 windows 不同：

- Windows：系统环境变量（配置后所有受用所有用户），用户环境变量（配置后受用特定用户）。
- Linux：全局环境变量，局部环境变量（仅支持当前 shell）。

**全局环境变量：**

- 系统层面：支持，子 shell,子程序 shell，打开多个 shell,都可用，关闭 shell 重开依旧好用。
- 用户层面：支持，子 shell,子程序 shell。

---

## 设定变量以及对应用到命令

---

- 设定变量要用下划线命名法（不懂自己去查）。
- 前面有 \$ 符号的就是变量
- 定义用户级别全局变量用小写
- 定义用户级别局部变量用小写
- 定义系统系统级别全局变量用大写

**printenv 命令：**显示指定的系统层面全局环境变量的值

---

如果没有指定变量，则打印出所有变量的名称和值。

---

**set 显示或设置用户层面的环境变量（全局+局部）和系统层面的局部变量**

---

set 显示或设置 shell 特性及 shell 变量

shell 特性 = set 选项提供

shell 变量 = 用户层面的环境变量（全局+局部）和系统层面的局部变量

查看显示命令不需要选项，选项太深奥，涉及到 shell 编程,这里不够格。

---

## 新建用户层面局部变量

```
变量名="变量内容"
```

---

## 新建用户层面全局变量

export 为shell变量或函数设置导出属性（自建全局变量）

导出属性=用户全局变量属性

```
export 变量名="变量内容"
```

## 选项

- f: 指向函数。
- n: 删除变量的导出属性。
- p: 显示全部拥有导出属性的变量。
- pf: 显示全部拥有导出属性的函数。
- nf: 删除函数的导出属性。
- : 在它之后的选项无效。

## 用户层临时添加系统的全局变量

例：给系统 PATH 路径添加，我们需要的路径。

```
PATH=$PATH:/home/prophet/Desktop
```

---

## 删除用户层面全局和局部变量

`unset` 删除指定的shell变量或函数。

### 语法

```
unset [-f] [-v] [-n] [name ...]
```

### 选项

- `-f`: 仅删除函数。
- `-v`: 仅删除变量（不包括只读变量）。
- `-n`: 删除具有引用属性的变量名（如果该选项存在）。

---

## 设置系统层面的全局变量（永久的全局变量）

### 工作原理

- 系统层面的环境变量有多个启动文件负责储存。
- 开机后默认去执行这些文件，去完成加载环境变量。
- 利用登录 `shell` 的方式去划分，该使用那些启动文件：
  - 服务器（server）版本，输入账户密码。
  - 界面版本，打开 `bash shell`
  - 运行脚本非交互 `shell`
- `/etc/profile` #最重要的启动文件之一

总结：因此我们要想修改永久的全局变量，需要找到根目录的下，对应的配置文件，这些文件大多都是隐藏的。

### 修改永久的全局变量

在 `Linux` 中不同的发行版本的，给用户配置环境变量的文件是不同的。

Ubuntu

```
~/ .bashrc
```

还有其他三个常见的列举一下，根据实际情况去查。

```
~/ .bash_profile
```

```
~/ .profile
```

```
~/ .bash_login
```

最后修改这些文件需要用到 `vim`，这个后面说。