

Linux 環境變量配置的6 種方法，建議收藏！

程序員大白 2021-12-15 22:37

點擊上方“**程序員大白**”，選擇“星標”公眾號
重磅乾貨，第一時間送達

程序員大白



作者：悠悠 來源：<http://uusama.com/927.html>

Linux環境變量配置

在自定義安裝軟件的時候，經常需要配置環境變量，下面列舉出各種對環境變量的配置方法。

下面所有例子的環境說明如下：

- 系統：Ubuntu 14.0
- 用戶名：uusama
- 需要配置MySQL環境變量路徑：/home/uusama/mysql/bin

Linux讀取環境變量

讀取環境變量的方法：

- export命令顯示當前系統定義的所有環境變量
- echo \$PATH命令輸出當前的PATH環境變量的值

這兩個命令執行的效果如下

```
uusama@ubuntu:~export
declare -x HOME="/home/uusama"
declare -x LANG="en_US.UTF-8"
declare -x LANGUAGE="en_US:"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="uusama"
declare -x MAIL="/var/mail/uusama"
declare -x PATH="/home/uusama/bin:/home/uusama/.local/bin:/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
declare -x SSH_TTY="/dev/pts/0"
declare -x TERM="xterm"
declare -x USER="uusama"

uusama@ubuntu:~ echo $PATH
/home/uusama/bin:/home/uusama/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

其中PATH變量定義了運行命令的查找路徑，以冒號:分割不同的路徑，使用export定義的時候可加雙引號也可不加。

Linux環境變量配置方法一：export PATH

使用export命令直接修改PATH的值，配置MySQL進入環境變量的方法：

```
export PATH=/home/uusama/mysql/bin:PATH
```

或者把PATH放在前面

```
export PATH=PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：立即生效
- 生效期限：當前終端有效，窗口關閉後無效
- 生效範圍：僅對當前用戶有效
- 配置的環境變量中不要忘了加上原來的配置，即\$PATH部分，避免覆蓋原來配置

Linux環境變量配置方法二：vim ~/.bashrc

通過修改用戶目錄下的~/.bashrc文件進行配置：

```
vim ~/.bashrc
```

在最后一行加上

```
export PATH=$PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：使用相同的用戶打開新的終端時生效，或者手動source ~/.bashrc生效
- 生效期限：永久有效
- 生效範圍：僅對當前用戶有效
- 如果有後續的環境變量加載文件覆蓋了PATH定義，則可能不生效

Linux環境變量配置方法三：vim ~/.bash_profile

和修改~/.bashrc文件類似，也是要在文件最後加上新的路徑即可：

```
vim ~/.bash_profile
```

在最后一行加上

```
export PATH=$PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：使用相同的用戶打開新的終端時生效，或者手動source ~/.bash_profile生效
- 生效期限：永久有效
- 生效範圍：僅對當前用戶有效
- 如果沒有~/.bash_profile文件，則可以編輯~/.profile文件或者新建一個

Linux環境變量配置方法四：vim /etc/bashrc

該方法是修改系統配置，需要管理員權限（如root）或者對該文件的寫入權限：

如果/etc/bashrc文件不可編輯，需要修改為可編輯

```
chmod -v u+w /etc/bashrc
```

```
vim /etc/bashrc
```

在最后一行加上

```
export PATH=$PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：新開終端生效，或者手動source /etc/bashrc生效
- 生效期限：永久有效
- 生效範圍：對所有用戶有效

Linux環境變量配置方法五：vim /etc/profile

該方法修改系統配置，需要管理員權限或者對該文件的寫入權限，和vim /etc/bashrc類似：

```
# 如果/etc/profile文件不可编辑，需要修改为可编辑
chmod -v u+w /etc/profile
```

```
vim /etc/profile
```

```
# 在最后一行加上
export PATH=$PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：新開終端生效，或者手動source /etc/profile生效
- 生效期限：永久有效
- 生效範圍：對所有用戶有效

Linux環境變量配置方法六：vim /etc/environment

該方法是修改系統環境配置文件，需要管理員權限或者對該文件的寫入權限：

```
# 如果/etc/bashrc文件不可编辑，需要修改为可编辑
chmod -v u+w /etc/environment
```

```
vim /etc/profile
```

```
# 在最后一行加上
export PATH=$PATH:/home/uusama/mysql/bin
```

注意事項：

- 生效時間：新開終端生效，或者手動source /etc/environment生效

- 生效期限：永久有效
- 生效範圍：對所有用戶有效

Linux環境變量加載原理解析

上面列出了環境變量的各種配置方法，那麼Linux是如何加載這些配置的呢？是以什麼樣的順序加載的呢？

特定的加載順序會導致相同名稱的環境變量定義被覆蓋或者不生效。

環境變量的分類

環境變量可以簡單的分成用戶自定義的環境變量以及系統級別的環境變量。

- 用戶級別環境變量定義文件：`~/.bashrc`、`~/.profile`（部分系統為：`~/.bash_profile`）
- 系統級別環境變量定義文件：`/etc/bashrc`、`/etc/profile`（部分系統為：`/etc/bash_profile`）、`/etc/environment`

另外在用戶環境變量中，系統會首先讀取`~/.bash_profile`（或者`~/.profile`）文件，如果沒有該文件則讀取`~/.bash_login`，根據這些文件中內容再去讀取`~/.bashrc`。

測試Linux環境變量加載順序的方法

為了測試各個不同文件的環境變量加載順序，我們在每個環境變量定義文件中的第一行都定義相同的環境變量`UU_ORDER`，該變量的值為本身的值連接上當前文件名稱。

需要修改的文件如下：

- `/etc/environment`
- `/etc/profile`
- `/etc/profile.d/test.sh`，新建文件，沒有文件夾可略過
- `/etc/bashrc`，或者`/etc/bash.bashrc`
- `~/.bash_profile`，或者`~/.profile`
- `~/.bashrc`

在每個文件中的第一行都加上下面這句代碼，並相應的把冒號後的內容修改為當前文件的絕對文件名。

```
export UU_ORDER="$UU_ORDER:~/.bash_profile"
```

修改完之後保存，新開一個窗口，然後echo \$UU_ORDER觀察變量的值：

```
uusama@ubuntu:~echoUU_ORDER
$UU_ORDER:/etc/environment:/etc/profile:/etc/bash.bashrc:/etc/profile
```

可以推測出Linux加載環境變量的順序如下：

1. /etc/environment
2. /etc/profile
3. /etc/bash.bashrc
4. /etc/profile.d/test.sh
5. ~/.profile
6. ~/.bashrc

Linux環境變量文件加載詳解

由上面的測試可容易得出Linux加載環境變量的順序如下，：

系統環境變量-> 用戶自定義環境變量/etc/environment -> /etc/profile -> ~/.profile

打開/etc/profile文件你會發現，該文件的代碼中會加載/etc/bash.bashrc文件，然後檢查/etc/profile.d/目錄下的.sh文件並加載。

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
```

```
if [ "PS1" ]; then
  if [ "BASH" ] && [ "BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1=' '
    fi
  fi
fi
```

```
fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
```

其次再打開~/.profile文件，會發現該文件中加載了~/.bashrc文件。

```
# if running bash
if [ -n "$BASH_VERSION" ]; then
  # include .bashrc if it exists
  if [ -f "$HOME/.bashrc" ]; then
    . "$HOME/.bashrc"
  fi
fi

# set PATH so it includes user's private bin directories
PATH="$HOME/bin:$HOME/.local/bin:$PATH"
```

從~/.profile文件中代碼不難發現，/.profile文件**只在用戶登錄的時候讀取一次**，而/.bashrc會在每次運行Shell腳本的時候讀取一次。

一些小技巧

可以自定義一個環境變量文件，比如在某個項目下定義uusama.profile，在這個文件中使用export定義一系列變量，然後在~/.profile文件後面加上：source uusama.profile，這樣你每次登陸都可以在Shell腳本中使用自己定義的一系列變量。

也可以使用alias命令定義一些命令的別名，比如alias rm="rm -i"（雙引號必須），並把這個代碼加入到~/.profile中，這樣你每次使用rm命令的時候，都相當於使用rm -i命令，非常方便。

推薦閱讀

13個你一定要知道的PyTorch特性

解讀：為什麼要做特徵歸一化/標準化？

一文搞懂PyTorch 內部機制

張一鳴：每個逆襲的年輕人，都具備的底層能力

關於程序員大白

程序員大白是一群哈工大，東北大學，西湖大學和上海交通大學的碩士博士運營維護的號，大家樂於分享高質量文章，喜歡總結知識，歡迎關注[程序員大白]，大家一起學習進步！



程序員大白訂閱號



長按識別二維碼

關注大白

快樂coding 快樂成長

閱讀原文

喜歡此內容的人還喜歡

優秀！博士畢業兩年後任副教授，34歲就成為中國最年輕女博導之一！

程序員大白



防範“浮躁症”，因勢利導讓“國潮”賦能中國品牌 | 電訊調查

新華每日電訊



一個“正能量”專家的不幸

平民讀詩

