

profile、bashrc 等文件的 区别和联系

author: Uncle Dragon

date: 2021-07-23

profile、bashrc 等文件的区别和联系

区别

/etc/profile

/etc/bash.bashrc

~/.bash_profile、~/.bash_login、~/.profile

~/.bash_profile

~/.bash_login

~/.profile

~/.bashrc

执行顺序

开机执行的顺序

source 文件

su 切换到该用户

不带参数

带 -l 参数

执行shell 脚本

Uncle e Dragon

profile、bashrc 等文件的区别和联系

一般配置环境变量时会用到这几个文件，那么这几个文件究竟有什么区别和联系呢？

下面从两个方面简单说下：

区别

几个文件的作用域不一样，`etc` 下面的是对所有用户的，`home` 目录下的只针对 shell 登录用户

`/etc/profile`

此文件为系统的每个用户设置环境信息,当用户第一次登录时,该文件被执行.并从 `/etc/profile.d` 目录的配置文件中搜集shell的设置.

所以如果你有对`/etc/profile`有修改的话必须得重启你的修改才会生效，此修改对每个用户都生效。

`/etc/bash.bashrc`

为每一个运行 `bash shell` 的用户执行此文件. 当 `bash shell` 被打开时,该文件被读取.每次用户打开一个终端时，执行此文件

如果你想对所有的使用`bash`的用户修改某个配置并在以后打开的`bash`都生效的话可以修改这个文件，修改这个文件也需要重启。

`~/.bash_profile`、`~/.bash_login`、`~/.profile`

三个文件中往往系统中往往只存在一个，在不同的发行版中不同，如CentOS和RedHat中是`~/.bash_profile`，而Debian和Ubuntu等系列中往往是`~/.profile`

但是 三个文件还是有个优先级，`~/.bash_profile` > `~/.bash_login` > `~/.profile`

所以自己不要随便增加文件，以避免出现其他问题

~/ .bash_profile

每个用户都可使用该文件输入专用于自己使用的shell信息,当用户登录时,该文件仅仅执行一次!默认情况下,他设置一些环境变量,执行用户的.bashrc文件

此文件类似于/etc/profile，也是需要需要重启才会生效，/etc/profile对所有用户生效，~/ .bash_profile只对当前用户生效

~/ .bash_login

~/ .profile

在Debian中使用 .profile 文件代替 .bash_profile 文件

.profile (由Bourne Shell和Korn Shell使用)和 .login (由C Shell使用)两个文件是 .bash_profile 的同义词，目的是为了兼容其它Shell。在Debian中使用 .profile 文件代替 .bash_profile 文件。

~/ .bashrc

该文件包含专用于你的bash shell的bash信息,当登录时以及每次打开新的shell时,该文件被读取.

执行顺序

我们可以做个实验：

以下实验都是在 20.04.2 LTS 中进行的，所有文件都是默认配置

在每个文件里追加 添加

```
1 echo "{filename} ... .."
2
3 文件开始添加:
4 echo -e "\033[32m----- {filename} ... .. ----- \033[0m"
5 文件结尾添加:
6 echo -e "\033[32m----- {filename} . end ----- \033[0m"
```

开机执行的顺序

```
Last login: Wed Jul 28 15:27:56 2021 from 127.0.0.1
----- /etc/profile ... .. -----
----- /etc/bash.bashrc ... .. -----
----- /etc/bash.bashrc . end -----
----- /etc/profile . end -----
----- ~/.profile ... .. -----
----- ~/.bashrc ... .. -----
----- ~/.bashrc ... .. -----
----- ~/.profile . end -----
dragon@WL-PC:~$
```

开机登录该用户读取顺序:

`/etc/profile` -> `/etc/bash.bashrc` -> `~/.profile` -> `~/.bashrc`

source 文件

```
1  $ source /etc/profile
2  ----- /etc/profile ... .. -----
3  ----- /etc/bash.bashrc ... .. -----
4  ----- /etc/bash.bashrc . end -----
5  ----- /etc/profile . end -----
6
7  $ source ~/.profile
8  ----- ~/.profile ... .. -----
9  ----- ~/.bashrc ... .. -----
10 ----- ~/.bashrc ... .. -----
11 ----- ~/.profile . end -----
12
13 $ source /etc/bash.bashrc
14 ----- /etc/bash.bashrc ... .. -----
15 ----- /etc/bash.bashrc . end -----
16
17 $ source ~/.bashrc
18 ----- ~/.bashrc ... .. -----
19 ----- ~/.bashrc ... .. -----
```

由于 `/etc/profile` 文件内执行了 `source /etc/bash.bashrc` 所以在执行 `source /etc/profile` 时也会读取 `/etc/bash.bashrc` 文件

同理 `~/.profile` 文件内执行了 `source ~/.bashrc` 所以在执行 `source ~/.profile` 时也会读取 `~/.bashrc` 文件

su 切换到该用户

不带参数

```
root@LWL-PC:~/home/dragon# su dragon
/etc/bash.bashrc ... ..
./.bashrc ... ..
dragon@LWL-PC:~$
```

`/etc/bash.bashrc -> ~/.bashrc`

带-l 参数

```
root@LWL-PC:~/home/dragon# su -l dragon
/etc/profile ... ..
/etc/bash.bashrc ... ..
./profile ... ..
./bashrc ... ..
```

`/etc/profile -> /etc/bash.bashrc -> ~/.profile -> ~/.bashrc`

执行shell 脚本

```
dragon@LWL-PC:~$ ./test.sh
this is a test bash file.
dragon@LWL-PC:~$
dragon@LWL-PC:~$
dragon@LWL-PC:~$ bash test.sh
this is a test bash file.
dragon@LWL-PC:~$
dragon@LWL-PC:~$ bash -l test.sh
----- /etc/profile ... .. -----
----- /etc/profile . end -----
----- ~/.profile ... .. -----
----- ~/.bashrc ... .. -----
----- ~/.profile . end -----
```

`./test.sh` 和 `bash test.sh` 不会读取上述文件

`bash -l test.sh` 执行顺序：

`/etc/profile -> ~/.profile -> ~/.bashrc`