

原来 Linux history 命令这么强大!

Linux爱好者 前天

以下文章来源于编程修养，作者肖邦



编程修养

作者从事 Linux 系统开发多年，公众号主要分享 Linux 实用技巧、算法、网络、高质量编程、高性能编程等方面的文章，欢迎关注！

当我们频繁使用 Linux 命令行时，有效地使用历史记录，可以大大提高工作效率。

在平时 Linux 操作过程中，很多命令是重复的，你一定不希望大量输入重复的命令。如果你是系统管理员，你可能需要对用户操作进行审计，管理好 Linux 命令历史记录显得非常重要。

今天我们来介绍一下，在 Linux 使用 history 来减少重复命令的几个实用技巧。

1 基本原理



Linux 命令的历史记录，会持久化存储，默认位置是当前用户家目录的 `.bash_history` 文件。

当 Linux 系统启动一个 Shell 时，Shell 会从 `.bash_history` 文件中，读取历史记录，存储在相应内存的缓冲区中。

我们平时所操作的 Linux 命令，都会记录在缓冲区中。包括 `history` 命令所执行的历史命令管理，都是在操作缓冲区，而不是直接操作 `.bash_history` 文件。

当我们退出 Shell，比如按下 `Ctrl+D` 时，Shell 进程会把历史记录缓冲区的内容，写回到 `.bash_history` 文件中去。

2 使用详解

清楚了 `history` 的基本原理，我们来具体学习一下如何使用它。

（一）基础用法

直接输入 `history` 命令，可以看到最近操作的所有命令都显示出来了

```
$ history
1 bash
2 ls
3 vim .bash_history
4 cat .bash_history
5 history
6 bash
```

有时候我不需要显示所有的历史命令，只显示最后的 10 条历史记录，可以在命令后加数字 `N` 即可

```
$ history 10
```

正常情况下，只有在 Shell 正常退出时，才会将缓冲区内容保存到文件。如果你想主动保存缓冲区的历史记录，执行 `-w` 选项即可

```
$ history -w
```

当然，如果你执行了一些敏感的命令操作，可以执行 `-c` 将缓冲区内容直接删除

```
$ history -c
```

（二）重复执行命令

如果要重复执行一些命令，可以使用 `!` 来快速执行重复的命令。

举个例子，重复执行第 1024 历史命令，可以执行如下命令

```
$ !1024
```

`1024` 这个编号可以通过 `history` 查看哦

重复执行上一条命令

```
$ !!
```

重复执行倒数第 6 条历史命令，可以通过 **负数** 表示，**-6** 表示倒数第 6 条记录

```
$ !-6
```

(三) 搜索历史命令

有时候，需要重复执行某字符串开头的最后一个命令，同样可以通过 **!** 来操作，然后按 Enter 执行即可

比如，刚才执行了一个很长命令，只记录命令开头是 **curl**，这时就可以通过 **!curl** 快速执行该命令

```
$ !curl
```

这个用法很高效，但存在不安全因素，因为有可能执行的命令不是你想要执行的，那就坏了。可以通过 **:p** 来安全地执行。

```
$ !curl:p  
curl www.sina.com.cn
```

加上 **:p** 后，只是打印出了搜索到的命令，如果要执行，请按 **Up** 键，然后回车即可。

如果你只知道某条命令包含了 **x** 信息，不是以 **x** 开头，同样可以通过 **?** 来执行包含字符串的命令

```
$ ?x
```

```
➤ !:sina
```

(四) 交互式搜索历史命令

在 Linux 搜索历史命令，还可以通过交互式的搜索方式，简直高效直接。在命令行输入 **Ctrl+R** 后，进入交互界面，键入需要搜索的关键字，如果匹配到多条命令，可以多次键入 **Ctrl+R** 来切换上一条匹配的命令。

```
(reverse-i-search)`sina':echo sina
```

可以看到，我输入了 sina 后，就自动匹配到最近一次和 sina 匹配的命令，这时按下回车就可以执行该命令。

(五) 重复执行上条命令

在这里总结下多种重复执行上条命令的方式，你可以选择一种自己喜欢的就可以啦

- **!!**
- **!-1**
- **Ctrl+p**
- **Up**
- **Ctrl+R**

(六) 显示时间戳

有时候需要对 Linux 系统做审计，那为历史记录添加时间戳，显示非常有用。

```
$ export HISTTIMEFORMAT='%F %T '
```

```
$ history 3
46 2021-04-18 15:21:33 curl baidu.com
47 2021-04-18 15:21:35 pwd
48 2021-04-18 15:21:39 history 3
```

可以看到，历史记录已经显示了时间戳。其实这些对于审计需求，还不够，可以加上更详细的信息：

```
$ export HISTTIMEFORMAT="%F %T `who -u am i 2>/dev/null|awk '{print $NF}'|sed \-e 's/[()]/g'` `whoami` "
6 2021-04-18 16:07:48 113.200.44.237 root ls
7 2021-04-18 16:07:59 113.200.44.237 root pwd
8 2021-04-18 16:08:14 113.200.44.237 root history
```

（七）控制历史记录总数

默认情况下，Linux 系统最多存储 1000 条历史记录，可以通过 **HISTSIZE** 环境变量查看

```
$ echo $HISTSIZE
1000
```

对于需要做审计的场景，1000 条历史记录可能会太少了，我们可以修改为合适的值

```
$ export HISTSIZE=10000
```

注意，`HISTSIZE` 变量只能控制缓冲区中的历史记录数量，如果需要控制 `.bash_history` 文件存储的最大记录数，可以通过 `HISTFILESIZE` 进行控制

上述命令行修改只在当前 Shell 环境生效，如果需要永久生效，需要写入配置文件

```
$ echo "export HISTSIZE=10000" >> ~/.bash_profile
$ echo "export HISTFILESIZE=200000" >> ~/.bash_profile
$ source ~/.bash_profile
```

（八）更改历史记录文件名

有时，为了方便管理和备份，需要更改历史记录文件的路径和名称。简单，同样可以通过环境变量 `HISTFILE` 更改它的文件名称

```
$ echo "export HISTFILE=/data/backup/chopin.bash_history" >> ~/.bash_profile
$ source ~/.bash_profile
```

（九）禁用历史记录

处于某种特殊环境，我们需要禁用历史记录

```
$ echo "export HISTSIZE=0" >> ~/.bash_profile
$ echo "export HISTFILESIZE=0" >> ~/.bash_profile
$ source ~/.bash_profile
```

哈哈，直接把上述两个变量的值设置为 `0`，就实现了禁用历史记录的功能

(十) 黑客必知的一个小技巧

最后分享一个不为人知的，黑客必知的小技巧。

在命令前额外多加一个空格，这样的命令是会被记录到历史记录的，感觉是不是很酷

这个技巧如果在你的系统不管用，请查看下环境变量 `HISTCONTROL` 是否包含 `ignorespace`，貌似 centos 系统默认没有设置这个值。

3 总结时间

在 Linux 系统，`history` 命令可以非常方便，帮助我们管理历史命令，平时我们命令都会先记录在缓存区，在 Shell 退出时才会记录到文件中。

`history` 命令提供了很方便的管理功能，合理去配置和管理历史记录，可以让你的 Linux 系统更加健壮和安全。

好了，老规矩，贴心的肖哥还是来总结一下 `history` 命令常用方法

- `history n`：只显示最近的 `n` 条历史记录
- `history -c`：清除缓存区中的历史记录
- `history -w`：将缓存区的历史记录保存到文件
- `history -d N`：删除第 `N` 条历史记录

几种重复执行命令的方法：`!!`、`!-1`、`!N`、`!string` 等

交互式历史命令搜索，请使用 `Ctrl+R` 快捷键

合适使用几个相关的环境变量，让你的 Linux 系统更安全：

- **HISTSIZE**：控制缓冲区历史记录的最大个数
- **HISTFILESIZE**：控制历史记录文件中的最大个数
- **HISTIGNORE**：设置哪些命令不记录到历史记录
- **HISTTIMEFORMAT**：设置历史命令显示的时间格式
- **HISTCONTROL**：扩展的控制选项

如果在生产环境，这些环境变量需要持久化到配置文件 `~/.bash_profile`

```
export HISTCONTROL=ignoreboth
# ignorespace: 忽略空格开头的命令
# ignoredups: 忽略连续重复命令
# ignoreboth: 表示上述两个参数都设置

# 设置追加而不是覆盖
shopt -s histappend

export HISTSIZE=1000
export HISTFILESIZE=200000
export HISTTIMEFORMAT="%F %T "
export HISTIGNORE="ls:history"
```

- EOF -

推荐阅读 — 点击标题可跳转

[1、Linux 系统安全强化指南](#)

[2、吐血整理的万字 Linux 内核源码规范](#)

[3、Linux 内核_jump label 与 static key 的原理与示例](#)

看完本文有收获？请分享给更多人

推荐关注「Linux 爱好者」，提升Linux技能



Linux爱好者

点击获取《每天一个Linux命令》系列和精选Linux技术资源。「Linux爱好者」日常分享 Linux/Unix 相关内容，包括：…
89篇原创内容

公众号

点赞和在看就是最大的支持♥

喜欢此内容的人还喜欢

Linux 系统操作之 vim 命令总结

软件测试面试汇总