

六个优雅的 **Linux** 命令行技巧



一些非常有用的命令能让命令行的生活更满足

使用 Linux 命令工作可以获得许多乐趣，但是如果您使用一些命令，它们可以减少您的工作或以有趣的方式显示信息时，您将获得更多的乐趣。在今天的文章中，我们将介绍六个命令，它们可能会使您在命令行上的时间更加值当。

watch

`watch` 命令会重复运行您给出的任何命令，并显示输出。默认情况下，它每两秒运行一次命令。命令的每次运行都将覆盖上一次运行时显示的内容，因此您始终可以看到最新的数据。

您可能会在等待某人登录时使用它。在这种情况下，您可以使用 `watch who` 命令或者 `watch -n 15 who` 命令使每 15 秒运行一次，而不是两秒一次。另外终端窗口的右上角会显示日期和时间。

```
1. $ watch -n 5 who
2. Every 5.0s: who                                stinkbug: Wed Aug 23
   14:52:15 2017
3.
4. shs      pts/0      2017-08-23 14:45 (192.168.0.11)
5. zoe      pts/1      2017-08-23 08:15 (192.168.0.19)
```

您也可以使用它来查看日志文件。如果您显示的数据没有任何变化，则只有窗口角落里的日期和时间会发生变化。

```
1. $ watch tail /var/log/syslog
2. Every 2.0s: tail /var/log/syslog                stinkbug: Wed Aug 23
   15:16:37 2017
3.
4. Aug 23 14:45:01 stinkbug CRON[7214]: (root) CMD (command -v
   debian-sal > /dev/nu
5. ll && debian-sal 1 1)
6. Aug 23 14:45:17 stinkbug systemd[1]: Started Session 179 of user
   shs.
7. Aug 23 14:55:01 stinkbug CRON[7577]: (root) CMD (command -v
   debian-sal > /dev/nu
8. ll && debian-sal 1 1)
9. Aug 23 15:05:01 stinkbug CRON[7582]: (root) CMD (command -v
   debian-sal > /dev/nu
10. ll && debian-sal 1 1)
11. Aug 23 15:08:48 stinkbug systemd[1]: Starting Cleanup of Temporary
   Directories...
12. Aug 23 15:08:48 stinkbug systemd-tmpfiles[7584]: [/usr/lib
   /tmpfiles.d/var.conf:1
13. 4] Duplicate line for path "/var/log", ignoring.
14. Aug 23 15:08:48 stinkbug systemd[1]: Started Cleanup of Temporary
   Directories.
15. Aug 23 15:13:41 stinkbug systemd[1]: Started Session 182 of user
   shs.
16. Aug 23 15:14:29 stinkbug systemd[1]: Started Session 183 of user
   shs.
17. Aug 23 15:15:01 stinkbug CRON[7828]: (root) CMD (command -v
   debian-sal > /dev/nu
18. ll && debian-sal 1 1)
```

这里的输出和使用命令 `tail -f /var/log/syslog` 的输出相似。

look

这个命令的名字 `look` 可能会让我们以为它和 `watch` 做类似的事情，但其实是不同的。`look` 命令用于搜索以某个特定字符串开头的单词。

```
1. $ look ecl
2. eclectic
3. eclectic's
4. eclectically
5. eclecticism
6. eclecticism's
7. eclectics
8. eclipse
9. eclipse's
10. eclipsed
11. eclipses
12. eclipsing
13. ecliptic
14. ecliptic's
```

`look` 命令通常有助于单词的拼写，它使用 `/usr/share/dict/words` 文件，除非你使用如下的命令指定了文件名：

```
1. $ look esac .bashrc
2. esac
3. esac
4. esac
```

在这种情况下，它的作用就像跟在一个 `awk` 命令后面的 `grep`，只打印匹配行上的第一个单词。

man -k

`man -k` 命令列出包含指定单词的手册页。它的工作基本上和 `apropos` 命令一样。

```
1. $ man -k logrotate
2. dh_installlogrotate (1) - install logrotate config files
3. logrotate (8) - rotates, compresses, and mails system logs
4. logrotate.conf (5) - rotates, compresses, and mails system logs
```

help

当你完全绝望的时候，您可能会试图使用此命令，`help` 命令实际上是显示一个 shell 内置命令的列表。最令人惊讶的是它有相当多的参数变量。你可能会看到这样的东西，然后开始想知道这些内置功能可以为你做些什么：


```
1.  $ help
2.  GNU bash, version 4.4.7(1)-release (i686-pc-linux-gnu)
3.  These shell commands are defined internally. Type 'help' to see
    this list.
4.  Type 'help name' to find out more about the function 'name'.
5.  Use 'info bash' to find out more about the shell in general.
6.  Use 'man -k' or 'info' to find out more about commands not in this
    list.
7.
8.  A star (*) next to a name means that the command is disabled.
9.
10.  job_spec [&]                                history [-c] [-d offset]
    [n] or hist>
11.  (( expression ))                            if COMMANDS; then
    COMMANDS; [ elif C>
12.  . filename [arguments]                     jobs [-lnprs] [jobspec
    ...] or jobs >
13.  :                                            kill [-s sigspec | -n
    signum | -sigs>
14.  [ arg... ]                                  let arg [arg ...]
15.  [[ expression ]]                          local [option]
    name[=value] ...
16.  alias [-p] [name[=value] ... ]            logout [n]
17.  bg [job_spec ...]                         mapfile [-d delim] [-n
    count] [-O or>
18.  bind [-lpsvPSVX] [-m keymap] [-f file>    popd [-n] [+N | -N]
19.  break [n]                                printf [-v var] format
    [arguments]
20.  builtin [shell-builtin [arg ...]]         pushd [-n] [+N | -N | dir]
21.  caller [expr]                             pwd [-LP]
22.  case WORD in [PATTERN [| PATTERN]...)>    read [-ers] [-a array]
    [-d delim] [->
23.  cd [-L|[-P [-e]] [-@]] [dir]              readarray [-n count] [-O
    origin] [-s>
24.  command [-pVv] command [arg ...]         readonly [-aAf]
    [name[=value] ...] o>
```

```

25.  compgen [-abcdefgjkuv] [-o option] [> return [n]
26.  complete [-abcdefgjkuv] [-pr] [-DE] > select NAME [in WORDS ...
;] do COMM>
27.  compopt [-o|+o option] [-DE] [name ..> set [-abefhkmnptuvxBCHP]
[-o option->
28.  continue [n] shift [n]
29.  coproc [NAME] command [redirections] shopt [-pqsu] [-o]
[optname ...]
30.  declare [-aAfFgIlNrtux] [-p] [name[=v> source filename
[arguments]
31.  dirs [-clpv] [+N] [-N] suspend [-f]
32.  disown [-h] [-ar] [jobspec ... | pid > test [expr]
33.  echo [-neE] [arg ...] time [-p] pipeline
34.  enable [-a] [-dnps] [-f filename] [na> times
35.  eval [arg ...] trap [-lp] [[arg]
signal_spec ...]
36.  exec [-cl] [-a name] [command [argume> true
37.  exit [n] type [-afptP] name [name
...]
38.  export [-fn] [name[=value] ...] or ex> typeset [-aAfFgIlNrtux]
[-p] name[=v>
39.  false ulimit
[-SHabdefiklmnpqrstuvxPT] [l>
40.  fc [-e ename] [-lnr] [first] [last] o> umask [-p] [-S] [mode]
41.  fg [job_spec] unalias [-a] name [name
...]
42.  for NAME [in WORDS ... ] ; do COMMAND> unset [-f] [-v] [-n]
[name ...]
43.  for (( exp1; exp2; exp3 )); do COMMAN> until COMMANDS; do
COMMANDS; done
44.  function name { COMMANDS ; } or name > variables - Names and
meanings of so>
45.  getopts optstring name [arg] wait [-n] [id ...]
46.  hash [-lr] [-p pathname] [-dt] [name > while COMMANDS; do
COMMANDS; done
47.  help [-dms] [pattern ...] { COMMANDS ; }

```

stat -c

stat 命令用于显示文件的大小、所有者、用户组、索引节点号、权限、修改和访问时间等重要的统计信息。这是一个非常有用的命令，可以显示比 **ls -l** 更多的细节。

```
1. $ stat .bashrc
2.   File: .bashrc
3.   Size: 4048          Blocks: 8          IO Block: 4096
   regular file
4.  Device: 806h/2054d    Inode: 421481    Links: 1
5.  Access: (0644/-rw-r--r--)  Uid: ( 1000/   shs)   Gid: ( 1000/   shs)
6.  Access: 2017-08-23 15:13:41.781809933 -0400
7.  Modify: 2017-06-21 17:37:11.875157790 -0400
8.  Change: 2017-06-21 17:37:11.899157791 -0400
9.  Birth: -
```

使用 `-c` 选项，您可以指定要查看的字段。例如，如果您只想查看一个文件或一系列文件的文件名和访问权限，则可以这样做：

```
1. $ stat -c '%n %a' .bashrc
2. .bashrc 644
```

在此命令中，`%n` 表示每个文件的名称，而 `%a` 表示访问权限。`%u` 表示数字类型的 UID，而 `%U` 表示用户名。

```
1. $ stat -c '%n %a' bin/*
2. bin/loop 700
3. bin/move2nohup 700
4. bin/nohup.out 600
5. bin/show_release 700
6.
7. $ stat -c '%n %a %U' bin/*
8. bin/loop 700 shs
9. bin/move2nohup 700 shs
10. bin/nohup.out 600 root
11. bin/show_release 700 shs
```

TAB

如果你没有使用过 `tab` 键来补全文件名，你真的错过了一个非常有用的命令行技巧。`tab` 键提供文件名补全功能（包括使用 `cd` 时的目录）。它在出现歧义之前尽可能多的填充文件名（多个文件以相同的字母开头。如果您有一个名为 `bigplans` 的文件，另一个名为 `bigplans2017` 的文件会发生歧义，你将听到一个声音，然后需要决定是按下回车键还是输入 `2` 之后再按下 `tab` 键选择第二个文件。