



链滴

linux一些强大的命令

作者: [mainlove](#)

原文链接: <https://ld246.com/article/1361414888363>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在阅读以下部分前，强烈建议读者打开一个shell实验，这些都不是shell教科书里的大路货哦：)

!\$

!\$是一个特殊的环境变量，它代表了上一个命令的最后一个字符串。如：你可能会这样：

```
$mkdir mydir
```

```
$mv mydir yourdir
```

```
$cd yourdir
```

可以改成：

```
$mkdir mydir
```

```
mv ! yourdir
```

```
cd !
```

```
sudo !!
```

以root的身份执行上一条命令。

场景举例：比如Ubuntu里用apt-get安装软件包的时候是需要root身份的，我们经常会忘记在apt-get前加sudo。每次不得不加上sudo再重新键入这行命令，这时可以很方便的使用sudo !!完事。

(陈皓注：在shell下，有时候你会输入很长的命令，你可以使用!xxx来重复最近的一次命令，比如，以前输入过，vi /where/the/file/is, 下次你可以使用 !vi 重得上次最近一次的vi命令。)

```
cd -
```

回到上一次的目录。

场景举例：当前目录为/home/a，用cd ../b切换到/home/b。这时可以通过反复执行cd -命令在/home/a和/home/b之间来回方便的切换。

(陈皓注：cd ~ 是回到自己的Home目录，cd ~user, 是进入某个用户的Home目录)

```
'ALT+.' or '&lt;ESC&gt;.'
```

热键alt+. 或 esc+. 可以把上次命令行的参数给重复出来。

```
^old^new
```

替换前一条命令里的部分字符串。

场景：echo "wonderful", 其实是想输出echo "wonderful"。只需要^a^o就行了，对很长的命令的误拼写有很大的帮助。(陈皓注：也可以使用!!:gs/old/new)

```
du -s * | sort -n | tail
```

列出当前目录里最大的10个文件。

```
:w !sudo tee %
```

在vi中保存一个只有root可以写的文件

```
date -d@1234567890
```

时间截转时间

```
&gt; file.txt
```

创建一个空文件，比touch短。

```
mtr coolshell.cn
```

mtr命令比traceroute要好。

在命令行前加空格，该命令不会进入history里。

```
echo "ls -l" | at midnight
```

在某个时间运行某个命令。

```
curl -u user:pass -d status=" Tweeting from the shell" http://twitter.com/statuses/update.xml
```

命令行的方式更新twitter。

```
curl -u username -silent "https://mail.google.com/mail/feed/atom" | perl -ne 'print "\t"
f/&lt;name&gt;/; print "$2\n" if /&lt;
title|name)&gt;(.
)&lt;/\1&gt;/;'
```

检查你的gmail未读邮件

```
ps aux | sort -nk +4 | tail
```

列出头十个最耗内存的进程

```
man ascii
```

显示ascii码表。

场景：忘记ascii码表的时候还需要google么？尤其在天朝网络如此“顺畅”的情况下，就更麻烦在GWF应用一次规则了，直接用本地的man ascii吧。

```
ctrl-x e
```

快速启动你的默认编辑器（由变量\$EDITOR设置）。

```
netstat -tlnp
```

列出本机进程监听的端口号。（陈皓注：netstat -anop 可以显示侦听在这个端口号的进程）

```
tail -f /path/to/file.log | sed '/^Finished: SUCCESS$/ q'
```

当file.log里出现Finished: SUCCESS时候就退出tail，这个命令用于实时监控并过滤log是否出现了某条录。

```
ssh user@server bash & /path/to/local/script.sh
```

在远程机器上运行一段脚本。这条命令最大的好处就是不用把脚本拷到远程机器上。

```
ssh user@host cat /path/to/remotefile | diff /path/to/localfile -
```

比较一个远程文件和一个本地文件

```
net rpc shutdown -l ipAddressOfWindowsPC -U username%password
```

远程关闭一台Windows的机器

```
screen -d -m -S some_name ping my_router
```

后台运行一段不终止的程序，并可以随时查看它的状态。-d -m参数启动“分离”模式，-S指定了一个session的标识。可以通过-R命令来重新“挂载”一个标识的session。更多细节请参考screen用法 man screen。

```
wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com
```

下载整个www.example.com网站。（注：别太过分，大部分网站都有防爬功能了：））

```
curl ifconfig.me
```

当你的机器在内网的时候，可以通过这个命令查看外网的IP。

```
convert input.png -gravity NorthWest -background transparent -extent 720x200 output.png
```

改一下图片的大小尺寸

```
lsof -i
```

实时查看本机网络服务的活动状态。

```
vim scp://username@host//path/to/somefile
```

```
vim一个远程文件
```

```
python -m SimpleHTTPServer
```

一句话实现一个HTTP服务，把当前目录设为HTTP服务目录，可以通过http://localhost:8000访问 这许是这个星球上最简单的HTTP服务器的实现了。

```
history | awk '{CMD[$2]++;count++;} END { for (a in CMD )print CMD[a] " " CMD[a]/count100 "% " } | grep -v "./" | column -c3 -s " " -t | sort -nr | nl | head -n10
```

(陈皓注: 有点复杂了, history|awk '{print \$2}' |awk 'BEGIN {FS="|" } {print \$1}' |sort|uniquc|sort -rn|head -10)

这行脚本能输出你最常用的十条命令，由此甚至可以洞察你是一个什么类型的程序员。

```
tr -c "[:digit:]" " " &lt; /dev/urandom | dd cbs=$COLUMNS conv=u  
block | GREP_COLOR=" 1;32" grep -color "[^ ] "
```

想看看Marix的屏幕效果吗？（不是很像，但也很Cool!）

转载<http://coolshell.cn/articles/8619.html>