

每个Linux 用户都应该知道的命令行技巧

2013/07/23 | 分类： IT技术 ^[1] | 2 条评论 | 标签： Linux ^[2], shell ^[3]

分享到： 117

本文由 伯乐在线 ^[4] - 高磊 ^[5] 翻译。未经许可，禁止转载！

英文出处：Joshua Levy ^[6]。欢迎加入翻译小组 ^[7]。

伯乐在线导读：有网友在问答网站Quora上提问：“有哪些省时小技巧，是每个Linux用户都应该知道的？”Joshua Levy 平常就在Linux平台工作，并且他积累了不少实用命令行技巧，他在回复中精选出一部分。对技术用户来说，这些技巧挺重要或实用，但知道的人并不多。下文略有点长，一般来说，用户也不需要对全部内容都了解，但为了达到省时方便的目的，Joshua Levy 仍不遗余力做了校对，以保证列出的每一条都值得一读，前提是你是一位Linux重度用户。

为了获取文中提到的一个命令的更多信息，先试下“man <命令名称>”，在一些情况下，为了让这条命令可以正常执行，你必须安装相应的包，可以用aptitude 或者 yum。如果失败了，求助Google。

- 学习基础的Bash。事实上，读整个的bash的帮助手册；很容易理解而且篇幅也不算长。其他一些可选的shell外观可能更漂亮，但是bash功能很强大而且总是能用（主要学习zsh或者tcsh在很多情况下你会受到限制）。
- 学习vim，对于Linux下的随机编辑，几乎没有工具能出其右（即使你大部分的时间里都在使用Emacs或者Eclipse）。
- 通过ssh-agent，ssh-add等命令，了解ssh，以及跳过每次登陆时密码验证的基础办法，。
- 熟悉bash下的工作管理：&，Ctrl-Z，Ctrl-C，jobs，fg，bg，kill，等等。
- 基础的文件管理：ls 以及 ls -l (特别的，学习”ls -l”中列出的每一列字段的含义)，less，head，tail，tail -f，ln，ln -s (学习软链接和硬链接的区别)，chown，chmod，du（快速了解磁盘总体占用情况），df，mount。
- 基础的网络管理命令：ip 或者 ifconfig，dig。
- 了解正则表达式，以及grep、egrep的不同命令选项，-0，-A，-B 都值得了解一下。
- 学习使用apt-get 或者 yum(取决于你的发行包)来找到并安装你需要的包。

日常使用篇

- 使用bash时，用Ctrl-R来搜索命令的历史记录。
- 使用bash时，用Ctrl-W来清除最后一个单词，使用Ctrl-U来清除整行。可以查看man readline来获取bash里面默认键的绑定设置。内容很多。比如Alt-.(注：点)遍历之前命令中使用过的参数，Alt-* 扩展了参数的匹配模式。
- 回到上次的工作目录：cd -。
- 如果你的命令敲到一半时改变了主意，可以用Alt-#来在命令前面增加一个#，使之成为一行注释（或者使用Ctrl-A回到命令开头，然后再键入#）。你可以之后再通过搜索历史记录回来。
- 使用xargs（或者parallel）。它非常强大。注意你能控制每一行(-L)执行多少项，也能控制如何并发（-P）。如果你不太确定它会如你所愿的工作，先使用xargs。再者，-l{} 很有用。例如：

```
find . -name \*.py | xargs grep some_function
```

```
cat hosts | xargs -l{} ssh root@{} hostname
```

- pstree -p 可以很方便的显示整个进程树。
- 使用pgrep 和pkill 来通过名字来发现进程或者给进程发信号（-f选项会有用）。
- 了解你能向进程发送的信号种类。比如，要挂起一个进程，使用kill -STOP [进程ID]。要了解整个列表，请参考man 7 signal。
- 如果你想让一个后台进程一直运行，使用nohup or disown。
- 通过netstat -lntp 来检测哪些进程在监听。同样可以用lsof。
- bash脚本中，使用set -x 来调试输出。使用set -e在有错误时终止时终止执行。要想严格输出错误，可以考虑使用set -o pipefail（虽然这个主题说起来有些复杂）。对于更复杂的脚本，也可以使用trap。
- bash脚本中，子shell（通过写在括号里）是一种组织命令的方便的方法。一个很常见的例子是暂时移动到另外一个工作目录，例如：

```
#在当前目录下做一些事情
```

```
（ cd /一些/另外的/目录；执行别的操作 ）
```

```
#继续在原来的目录下执行
```

- 要注意bash中有很多种变量表达式。检查一个变量是否存在：\${name:?错误信息}。例如：如果一个bash脚本需要一个单变量，只需要写input_file=\${1:?usage: \$0 input_file}。数值扩展：i=\$((i+1)%5)。序列：{1..10}。字符串的

整理: `${var%suffix}` 和 `${var#prefix}`。例如:

```
if var==foo.pdf, then echo ${var%.pdf}.txt    #会打印"foo.txt"。
```

- 通过 `<`(其他指令), 一条命令的输出可以被当作是一个文件的内容来对待。例如, 比较本地和远程的 `/etc/hosts` 文件, 可以用 `diff /etc/hosts <(ssh [远程主机] cat /etc/hosts)`。
- 了解bash中的“here documents”, 比如 `cat <<EOF ...`
- bash中, 通过 其他指令 `> 日志文件 2>&1` 把标准输出以及标准错误重定向。常见的情况是, 为了保证一条指令没有为标准输入留下一个打开的文件描述符, 从而输出至你当前所在的终端, 增加“`</dev/null`”也是好的习惯。
- 用 `man ascii`可以得到一个完整的ASCII表, 有对应的16进制和10进制的值。
- 通过ssh连接远程终端时, 使用screen或者dtach 来保持你的session, 防止被打断。在ssh中, 了解如何使用-L或者-D选项 (有时也会用到-R) 会很有用处, 比如, 如果通过从一个远程的服务器访问一个网页。
- 优化你的SSH选项也可能管用。比如, 下面的.ssh/config 内容在一些网络环境下可以防止连接掉线, 当连接到新主机时不需要再次确认, 跳转验证, 并且还使用了压缩 (对在一些低宽带的连接环境下使用scp时会有帮助)。

```
TCPKeepAlive=yes
```

```
ServerAliveInterval=15
```

```
ServerAliveCountMax=6
```

```
StrictHostKeyChecking=no
```

```
Compression=yes
```

```
ForwardAgent=yes
```

数据处理篇

- 把HTML转成文本: `lynx -dump` 标准输入
- 如果要处理XML, `xmlstarlet`会很棒。
- 对于Amazon S3, `s3cmd` 很方便 (虽然还不太成熟, 可能会有一些不太好的特性)。
- 了解sort 以及 uniq (包括uniq的 -u 以及 -d 选项)。
- 了解cut, paste, join 来操作文本文件。许多人使用cut但却忘了还有join。
- 当你要在文件之间做集合的加, 减, 以及差运算时, 用sort/uniq是非常方便的。假如a和b是两个已经去重的文本文件, 那么运算起来会很快, 而且可以在

任意大小的文件之间执行操作，甚至可以到GB字节大小。（sort不受内存限制，不过如果/tmp 在一个很小的root分区的话，你可能需要使用-T选项）

```
cat a b | sort | uniq > c    # c is a union b
```

```
cat a b | sort | uniq -d > c    # c is a intersect b
```

```
cat a b b | sort | uniq -u > c    # c is set difference a - b
```

- 了解本地化会影响到许多命令行的工作，包括排序的顺序和性能。多数的linux安装包会把LANG或者其他一些本地化的变量设置为类似美国英语的一个本地设置。这会让sort和其他一些命令运行起来慢很多。（注意即使你使用UTF-8编码的文本，你仍然可以放心的通过ASCII码的顺序来排序，这一点用处很多）为避免i18n拖慢日常的工作，使用传统的基于字节的排序顺序，使用export LC_ALL=C（实际上，考虑在你的.bashrc里加进去）。
- 了解基本的AWK和sed命令来做简单的数据处理。例如：对一个文本文件的第三列的数字求和：awk '{x += \$3} END {print x}'。这大概比同等的python速度要快三倍并且代码长度也会简短3倍。
- 就地替换一个字符串在所有文件里所有出现的地方。

```
perl -pi.bak -e 's/old-string/new-string/g' my-files-*.txt
```

- 使用shuf来随机打乱一个文件中的行或者选择一个随机的行。
- 了解sort的各个选项。知道键值是如何工作的。特别是，当你要使用 -k1时，要格外注意：1只对第一个字段排序，-k1则意味着根据整个行排序。
- 稳定排序（sort -s）可能会有用。例如，先根据第二个字段排序，再根据第一个字段排序时，你可以使用sort -k1,1 | sort -s -k2,2
- 如果你需要在bash里的命令行里写入一个tab键的字面值的话，按Ctrl+V, <tab>或者\$'\t'（后者更好，因为你可以复制、粘贴）。
- 对于二进制文件，使用hd来进行简单的导出16进制表示或者用bvi进行二进制的编辑。
- 对于二进制文件，strings（还有grep等等）可以让你发现文件的字节位（0101）。要对文件转编，可以试下iconv，或者如果要使用更高级的用法，试试uconv，它可以支持一些高级的Unicode方面的事情。比如，这条命令可以将重音都小写，并且去掉（通过扩展并且丢掉）：

```
uconv -f utf-8 -t utf-8 -x '::Any-Lower; ::Any-NFD; [:Nonspacing Mark:] >;  
::Any-NFC; ' < input.txt > output.txt
```

- 要将文件切片，可以试试split（根据大小切分）或者csplit（根据模式切分）。

系统调试篇

- 对于web调试来说，curl和curl -l会有用，以及和wget相同的那部分功能。
- 如果想了解磁盘/cpu/网络的状态，可以使用iostat，netstat，top(更好一些的话，用htop)，以及（尤其是）dstat，对于想快速了解系统当前正在发生的事情，非常的方便。如果了解内存当前的状态，可以使用free以及vmstat，还要了解各项输出的含义。特别值得一提的是，你要知道“cached”的数值是linux内核保留用来做文件缓存的空间的大小，所以真正可用的有效内存是“free”项的对应值。
- java的系统调试则完全是另外一回事，但在Sun以及其他的JVM上有一个简单的技巧，就是你可以运行kill -3 <pid>，得到一个完整的栈调用轨迹以及堆使用的总体情况（包括产生的垃圾回收细节，这里面包含有很多的信息），会被定向到标准错误或者日志。
- 使用mtr作为更好的网络追踪，识别网络存在的问题。
- 要查看一个磁盘是否是满的，ncdu要比一般用的“du -sk *”要快。
- 要查看哪些socket或者进程在占用带宽，试试iftop或者netlogs。
- ab 工具（随apache的安装包一起发布）对于检测网络服务器的性能很有帮助，对于更加复杂的压力测试，可以试下siege。对于更加严重的网络问题的调试，试试wireshark或者tshark。了解strace和ltrace。这在一个程序突然失败，挂掉，或者崩溃，而你却不知所措，或者是你想知道程序的整体性能的情况时，会很有帮助。可以注意下-c和-p选项。
- 了解用ldd来检查共享库函数等的一些问题。
- 了解如何用gdb连接到一个正在运行的程序，并且得到它的调用堆栈。
- 使用/proc. 对于现场调试问题会很有帮忙。例如：/proc/cpuinfo，/proc/xxx/cwd, /proc/xxx/exe, /proc/xxx/fd/, /proc/xxx/smaps。
- 当要调试过去一段时间内出现的问题时，sar 会有用，它可以显示过去一段时间内的CPU，内存，网络的统计信息。
- 对于更深层次的系统性能优化，可以关注下stap（systemtap）或者perf。
- 当出现了一些很诡异的问题时，可以试下dmesg（比如硬件或者驱动的问题）。

关于作者：高磊 [8]

关注技术，爱好技术，许不大成，心仍往之！ 新浪微博：@kaulie [9]

查看高磊的更多文章 » [10]

117

[11]



每次谈加薪
老板都很忙
然而我已经看穿了一切

立即访问JobDeer.com

国内第一的人才拍卖网站

[12]

[13]

1. <http://blog.jobbole.com/category/it-tech/>
2. <http://blog.jobbole.com/tag/linux/>
3. <http://blog.jobbole.com/tag/shell/>
4. <http://blog.jobbole.com/>
5. <http://blog.jobbole.com/author/kaulie/>
6. <http://www.quora.com/Linux/What-are-some-time-saving-tips-that-every-Linux-user-should-know>
7. http://www.jobbole.com/groups/6/?utm_source=jobboleblog-article

8. <http://blog.jobbole.com/author/kaulie/>
9. <http://weibo.com/p/1005051800460410>
10. <http://blog.jobbole.com/author/kaulie/>
11. <http://gh-tools.sinaapp.com/bole.html>
12. <http://gh-tools.sinaapp.com/bole.html>
13. <http://gh-tools.sinaapp.com/bole.html>