CentOS 丛书目录 — 系统管理 — 网络服务 — 应用部署

提高工作效率

内容提要

1. 学会使用命令行补全
2. 学会使用命令历史
3. 学会使用命令别名

命令行补全

命令行补全（Command-Line Completion）是 bash 的一个重要功能，它为用户输入命令行提供了方便。

当用户在输入命令行时，只需输入部分目录名、文件名、用户名、变量名等，然后按<Tab>键，bash便会自动补齐未输入的部分。

例如，假定当前的工作目录包含以下的文件和子目录：

```
$  ls
system/                 myprogram
```

如果要进入 system 子目录，因为 system 是当前目录里惟一以字母s开头的子目录，bash 在用户只输入字母s后就能判断出用户要做什么了：

```
$  cd s
```

在键入字母s后，惟一的可能就是system。想让 bash 帮助结束命令的话，按下<Tab>键：

```
$  cd s<Tab>
```

当按<Tab>键以后，bash 将帮助补齐命令并显示在屏幕上。但在按下回车键以前命令并没有被执行，bash会让用户检验补齐的命令是否是用户真正需要的。

同样，如下的操作即可运行程序 myprogram

```
$   ./m<Tab>
```

当用户输入命令时不论何时按下<Tab>键，bash都将尽其所能地试图补齐命令，不行的话会发出蜂鸣来提醒你需要更多的信息。用户需要键入更多的字符，并再次按下<Tab>键，重复这个过程直至你期望的命令出现。

若用户按下<Tab>键后，系统发出蜂鸣，若再按一次Tab键，此时bash将符合条件的目录或文件显示出来，以便用户继续输入更多的信息。例如：

```
$  a<tab><tab>
a2p             alias           ascii-xfr       audit2allow
ab              amuFormat.sh    aserver         authconfig
```

```
ac                apm               aspell            authconfig-tui
acpi_listen       apmsleep          aspell-import     autovbox
activation-client apropos           at                awk
addftinfo         ar                atq
addr2line         arch              atrm
afs5log           as                attr
```

若按两次<tab>键后，系统仍发出蜂鸣声，则无法进行补齐，通常是已经输入的部分有错误。

当在 $ 之后键入<tab>键可以补全变量名，例如：

```
$  echo $B<tab>
/bin/bash
$  abc=123
$  echo $a<tab>
123
```

当在 ~ 之后键入<tab>键还可以补全用户名，例如：

```
$  echo ~o<tab>
~operator/ ~osmond/
$  ~<tab><tab>
~adm        ~distcache/ ~lp/        ~nobody/    ~rpcuser/   ~sync/
~apache/    ~dovecot/   ~lrj/       ~nscd/      ~rpm/       ~uucp
~avahi/     ~ftp/       ~mail/      ~operator/  ~shutdown/  ~vcsa/
~bin/       ~games/     ~mailnull/  ~osmond/    ~smart/     ~webalizer/
~crq/       ~gopher     ~named/     ~pcap       ~smmsp/
~daemon/    ~haldaemon/ ~news/      ~root/      ~squid/
~dbus/      ~halt/      ~nfsnobody/ ~rpc/       ~sshd/
```

命令历史

bash 可以记录一定数目的以前在 Shell 中输入的命令。可以记录历史命令的数目由环境变量 HISTSIZE 的值所指定。记录历史命令的文本文件由环境变量 HISTFILE 来指定，默认的记录文件是.bash_history，这是一个隐含文件，位于用户的自家目录中。

仅将先前的命令存在历史文件里是没有用的，将历史命令记录后，用户如何使用它们呢？有如下的方式：

1. 最简单的方法是用上下方向键、<PgUp>和<PgDn>键来查看历史命令
2. 如果需要的话，可以使用键盘上的编辑功能键对显示在命令行上的命令进行编辑
3. 用 history 命令来显示和编辑历史命令
4. 用 !! 执行最近执行过的命令
5. 用！<命令事件号> 执行已经运行过的命令
6. 用！<已经使用过的命令前面的部分> 执行已经运行过的命令

在Linux环境下使用命令历史举例：

```
// 显示命令历史
$  history
（略）
1002  whereis passwd
1003  ll /usr/bin/passwd
```

```
1004  ll -d /tmp
1005  clear
1006  history
$
// 执行命令历史中最近一次以c开头的命令
$  !c
clear
// 执行命令历史中编号为1003的命令
$  !1003
ll /usr/bin/passwd
-rwsr-xr-x 1 root root 26972 2006-04-03 21:37 /usr/bin/passwd
// 执行最近执行过的命令
$  !!
ll /usr/bin/passwd
-rwsr-xr-x 1 root root 26972 2006-04-03 21:37 /usr/bin/passwd
$
```

命令别名

命令别名是bash提供的另一个使用户的工作变得轻松的方法。命令别名通常是其他命令的缩写，用来减少键盘输入。同时也允许用户为命令另外取一个自己习惯使用的名字。 可以使用 alias 命令来达到上述的目的。命令格式为：

```
alias [alias_name='original_command']
```

其中:

- alias_name是用户给命令取的别名
- original_command是原来的命令和参数

1. 在定义别名时，等号两边不允许有空格存在，否则bash将不能确定用户的意图。若命令中包含空格或其他的特殊字符串必须使用引号。
2. 如果用户需要别名的定义在每次登录时均有效，应该将其写入用户自家目录下的.bashrc文件中。

如果用户不使用任何参数来使用alias命令，将显示当前的别名和其对应的原始命令。例如，下面是 CentOS 中使用 alias 命令的输出结果：

```
$  alias
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

又例如：对于一名DOS用户并且习惯了DOS命令的使用者，可以用下面的别名定义使其Linux表现得像DOS一样。

```
$  alias dir='ls'
$  alias copy='cp'
$  alias ren='mv'
$  alias move='mv'
$  alias md='mkdir'
$  alias rd='rmdir'
$  alias md='mkdir'
$  alias type='cat'
```

定义了别名之后就可以直接使用别名来执行相应的命令。

若系统中有一个命令，同时又定义了一个与之同名的别名（例如，系统中有ls命令，且又定义了ls的别名），则别名将优先于系统中原有命令的执行。要想临时使用系统中的命令而非别名，应该在命令前添加"\"字符，例如：

```
$ \ls
```

以上命令将运行系统中原来的ls命令而不是ls的别名，它不区分文件的类型和颜色。

当用户要取消别名的定义时使用unalias命令，其命令格式为：

```
unalias alias_name
```

其中 alias_name 为先前用 alias 命令所定义的别名。例如：可以使用如下的命令取消 ls 的别名设置：

```
$ unalias ls
```

在字符环境下复制和粘贴也是一种提高工作效率的方法，但这不是shell的功能。 gpm 守护进程可以在字符界面下用鼠标来复制与粘贴。具体做法是按住鼠标左键拖动，使要复制的文字反白显示，这时反白的区域已经被复制，再按鼠标右键复制的内容就会被粘贴在光标所在位置了。

- 显示源文件
- 登录