

@[TOC](

Linux基础入门

)

一.Linux用户类型

1.root 用户

在linux中root用户属于一个特殊的管理帐户，也被称为超级用户。root拥有接近百分之百的系统控制权限，对系统损害几乎有无限的能力。除非必要,一般不会使用root登录系统。

2.普通（非特权）用户

普通用户权限有限，造成损害的能力比较有限。普通用户也可以在root或者管理员用户手偶全下成为管理员用户。

二.Linux下的终端种类介绍

设备终端：键盘、鼠标、显示器

控制台终端： /dev/console

串行终端： /dev/ttyS#

虚拟终端：tty: teletypewriters, /dev/tty#, tty 可有n个, Ctrl+Alt+F#

图形终端：startx, xwindows

CentOS 6: Ctrl + Alt + F7

CentOS 7: 在哪个终端启动，即位于哪个虚拟终端

伪终端：pty: pseudo-tty , /dev/pts/# 如：SSH远程连接

查看当前的终端设备：#tty

交互式接口

交互式接口：启动终端后，在终端设备附加的一个交互式应用程序

GUI: Graphic User Interface

X protocol, window manager, desktop

Desktop:

GNOME (C, 图形库gtk) # linux 桌面版有名的桌面环境之一

KDE (C++,图形库qt) # 同上

XFCE (轻量级桌面) # 同上

CLI: Command Line Interface # 命令行接口

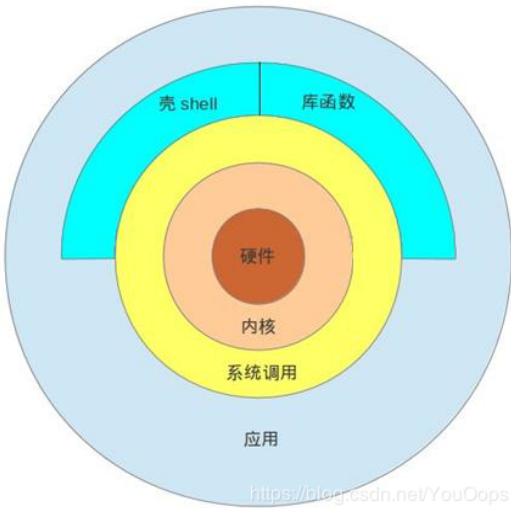
如：shell程序 就是一个命令行接口

三.Shell介绍

1.什么是shell

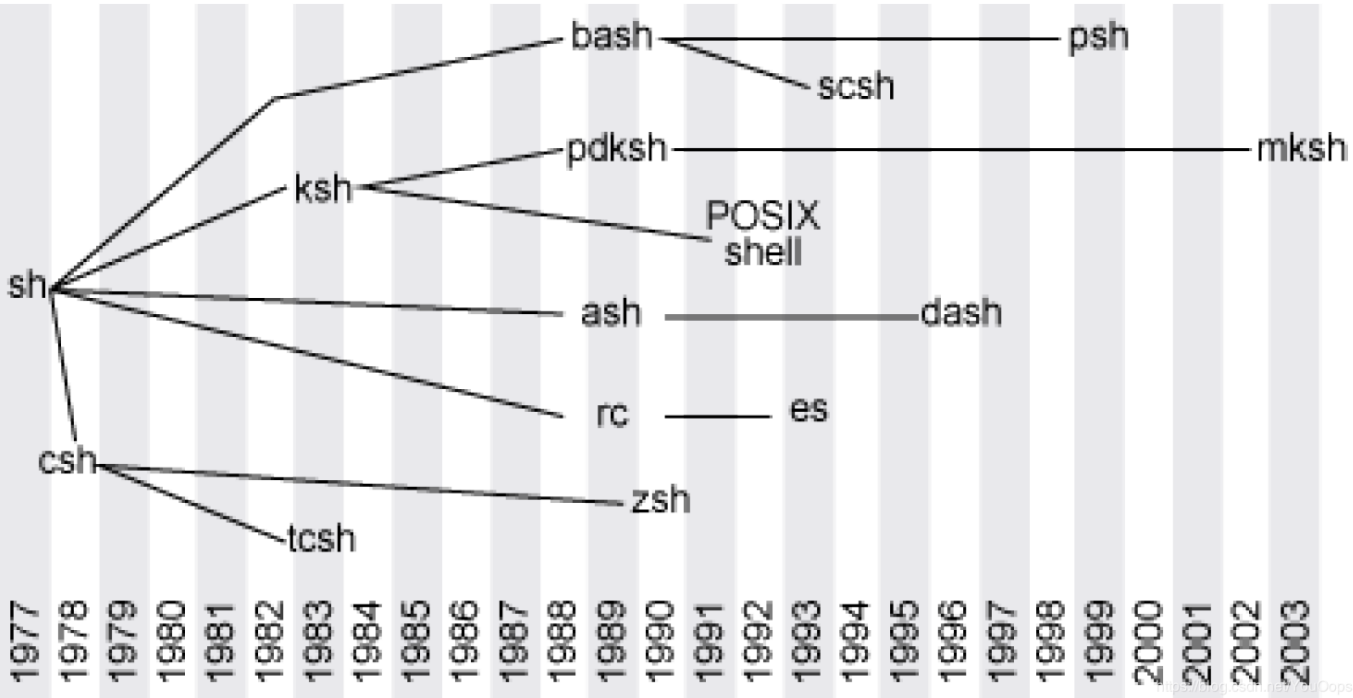
Shell 是Linux系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行

shell也被称为LINUX的命令解释器（command interpreter）,是一种高级程序设计语言。



shell处于内核和用户之间

2.各种shell及其分支



从1977年到2003年各种衍生于shell的不同shell版本示意

3.bash shell简介

bash shell是目前linux系统的标配shell，其起源于GNU Bourne-Again Shell(bash)；是GNU计划中重要的工具软件之一，与sh兼容。

CentOS默认使用 bash shell

```

在bash shell中显示当前使用的shell
echo ${SHELL}
显示当前系统使用的所有shell
cat /etc/shells
主机名示例
bj-yz-k8s-node1-100-10.magedu.com

```

- 注意：主机名不建议用下划线

4.bash shell的命令提示符认识认识

```

命令提示符: prompt
[root@localhost ~]#
      # 管理员
      $ 普通用户
显示提示符格式
[root@localhost ~]#echo $PS1
修改提示符格式
PS1="\[\e[1;5;41;33m\][\u@\h \W]\\$\\[\e[0m\]"
PS1="\[\e[1;32m\][\[\e[0m\]\t \[\e[1;33m\]\u\[\e[36m\]@\h\[\e[1;31m\] \W\
[\e[1;32m\]]\[\e[0m\]\\$"
\e 控制符\033      \u 当前用户
\h 主机名简称      \H 主机名
\w 当前工作目录    \W 当前工作目录基名
\t 24小时时间格式 \T 12小时时间格式
\! 命令历史数      \# 开机后命令历史数

```

四.shell执行命令

1.在shell中执行命令

- > 输入命令后回车
 - shell程序找到键入命令所对应的可执行程序或代码，并由其分析后提交给内核分配资源将其运行起来
- > 在shell中可执行的命令有两类
 - 内部命令：由shell自带的，而且通过某命令形式提供
 - help 内部命令列表 # 查询内部命令的帮助文档
 - enable cmd 启用内部命令
 - enable -n cmd 禁用内部命令
 - enable -n 查看所有禁用的内部命令
 - 外部命令：在文件系统路径下有对应的可执行程序文件
 - 查看路径: which -a |--skip-alias ; whereis
- > 区别指定的命令是内部或外部命令
 - type COMMAND

2. 执行外部命令

- Hash缓存表

系统刚启动时hash表为空，当外部命令执行时，bash默认会从PATH路径下寻找该命令，找到后会将这条命令的路径记录到hash表中。当再次使用该命令时，shell解释器首先会查看hash表，存在将执行之，如果不存在，将会去PATH路径下寻找，利用hash缓存表可大大提高命令的调用速率。

hash常见用法

```
hash 显示hash缓存
hash -l 显示hash缓存，可作为输入使用
hash -p path name 将命令全路径path起别名为name
hash -t name 打印缓存中name的路径
hash -d name 清除name缓存
hash -r 清除缓存
```

3. 命令别名

显示当前shell进程所有可用的命令别名

```
alias
```

定义别名NAME，其相当于执行命令VALUE

```
alias NAME='VALUE'
```

在命令行中定义的别名，仅对当前shell进程有效

如果想永久有效，要定义在配置文件中

仅对当前用户：~/.bashrc

对所有用户有效：/etc/bashrc

编辑配置文件给出的新配置不会立即生效

如果要不重启电脑的情况下是配置文件生效，则可以让bash进程重新读取配置文件：

```
source /path/to/config_file
```

```
./path/to/config_file
```

此处source和.都是shell built-in类型

撤消别名：unalias

```
unalias [-a] name [name ...]
```

-a 取消所有别名

别名同原命令同名的情况下，如果要执行原命令，可使用

```
\ALIASNAME
```

```
"ALIASNAME"
```

```
'ALIASNAME'
```

```
command ALIASNAME
```

```
/path/command
```

4. 命令格式

COMMAND [OPTIONS...] [ARGUMENTS...]

选项：用于启用或关闭命令的某个或某些功能

短选项: UNIX 风格选项, -c 例如: -l, -h

长选项: GNU风格选项, --word 例如: --all, --human

BSD风格选项: 一个字母, 例如: a

参数: 命令的作用对象, 比如文件名, 用户名等

注意:

多个选项以及多参数和命令之间使用空白字符分隔

取消和结束命令执行: Ctrl+c, Ctrl+d

多个命令可以用;符号分开

一个命令可以用\分成多行

```
[root@centos7 ~]$echo hello; echo hi
hello
hi
[root@centos7 ~]$host\
> na\
> me
centos7.steve
```

五.获取命令的帮助文档

type 判断**command**是内部还是外部命令

内部命令 `help command`

外部命令 `whatis --> command --help --> man -f -k ...`

- 各个帮助的类型

命令帮助

内部命令: `help COMMAND` 或 `man bash`

外部命令: `COMMAND --help` 或 `COMMAND -h`

(2) 使用手册(manual)

`man COMMAND`

(3) 信息页

`info COMMAND`

(4) 程序自身的帮助文档

`README`

`INSTALL`

`ChangeLog`

(5) 程序官方文档

官方站点: `Documentation`

(6) 发行版的官方文档

(7) Google

六.练习

1、显示当前时间，格式：2016-06-18 10:20:30

```
date +%F\ %T
```

2、显示前天是星期几

```
date -d '-2 day' +%A
```

3、设置当前日期为2019-08-07 06:05:10

```
date 080706052019.30
```

七.man

1.man命令

man命令提供完整的命令帮助信息，手册页存放在/usr/share/man；几乎每个命令都有man的“页面”，man页面分组为不同的“章节”，统称为Linux手册。

man命令的配置文件：/etc/man.config | man_db.conf

MANPATH /PATH/TO/SOMEWHERE：指明man文件搜索位置

man -M /PATH/TO/SOMEWHERE COMMAND：到指定位置下搜索COMMAND命令的手册页并显示

2.man 章节

- man帮助分为9个章节，带(p)标识的章节属于开发人员帮助手册

```
man 1: 用户命令
man 2: 系统调用
man 3: C库调用
man 4: 设备文件及特殊文件
man 5: 配置文件格式
man 6: 游戏
man 7: 杂项
man 8: 管理类的命令
man 9: Linux 内核API
```

3.man帮助段落说明

帮助手册中的段落说明：

NAME 名称及简要说明

SYNOPSIS 用法格式说明

•[] 可选内容

- <> 必选内容
- a|b 二选一
- { } 分组
- ... 同一内容可出现多次

DESCRIPTION 详细说明

OPTIONS 选项说明

EXAMPLES 示例

FILES 相关文件

AUTHOR 作者

COPYRIGHT 版本信息

REPORTING BUGS bug信息

SEE ALSO 其它帮助参考

4.man帮助命令使用

查看man手册页

```
man [章节] keyword
```

列出所有帮助

```
man -a keyword
```

搜索man手册

```
man -k keyword 列出所有匹配的页面
```

使用 whatis 数据库;相当于whatis

```
man -f keyword
```

打印man帮助文件的路径

```
man -w [章节] keyword
```

5.通过在线文档获取帮助

- 第三方应用官方文档

[apache](#)

[nginx](#)

[Mariadb](#)

[mysql](#)

[tomcat](#)

[python](#)

通过发行版官方的文档光盘或网站可以获得 安装指南、部署指南、虚拟化指南等.

- 红帽知识库和官方在线文档

<http://kbase.redhat.com>

<http://www.redhat.com/docs>

<http://access.redhat.com>

<https://help.ubuntu.com/lts/serverguide/index.html>

八.练习

1、在本机字符终端登录时，除显示原有信息外，再显示当前登录终端号，主机名和当前时间

```
man -k issue --> man 8 pam_issue
\l --tty
\n --hostname
\t --time
vim /etc/issue
```

2、今天18:30自动关机，并提示用户

```
shutdown 18:30 wall "System will shutdown at 18:30!!"
```

九.简单的小命令

1.关机重启

关机: `halt`, `poweroff`

重启: `reboot`

-f: 强制, 不调用`shutdown`

-p: 切断电源

关机或重启: `shutdown`

`shutdown [OPTION]... [TIME] [MESSAGE]`

-r: `reboot`

-h: `halt`

-c: `cancel`

TIME: 无指定, 默认相当于+1 (CentOS7)

now: 立刻, 相当于+0

+m: 相对时间表示法, 几分钟之后; 例如 +3

hh:mm: 绝对时间表示, 指明具体时间

2.查看登录信息

用户登录信息查看命令:

`whoami`: 显示当前登录有效用户

`who`: 系统当前所有的登录会话

`w`: 系统当前所有的登录会话及所做的操作1

3.screen命令

- `screen`命令可以用来在同一台服务器上互相协助处理问题


```
创建新screen会话
    screen -S [SESSION]
加入screen会话
    screen -x [SESSION]
退出并关闭screen会话
    exit
剥离当前screen会话
    Ctrl+a,d
显示所有已经打开的screen会话
    screen -ls
恢复某screen会话
    screen -r [SESSION]
```

4.echo命令

功能：显示字符

语法：echo [-neE][字符串]

说明：echo会将输入的字符串送往标准输出。输出的字符串间以空白字符隔开，并在最后加上换行号

选项：

- E （默认）不支持 \ 解释功能
- n 不自动换行
- e 启用 \ 字符的解释功能

显示变量

echo "\$VAR_NAME" 变量会替换，弱引用

decho '\$VAR_NAME' 变量不会替换，强引用

启用命令选项-e，若字符串中出现以下字符，则特别加以处理，而不会将它当成一般文字输出：

- \a 发出警告声
 - \b 退格键
 - \c 最后不加上换行符号
 - \e escape，相当于\033
 - \n 换行且光标移至行首
 - \r 回车，即光标移至行首，但不换行
 - \t 插入tab
 - \\ 插入\字符
 - \0nnn 插入nnn（八进制）所代表的ASCII字符
- ```
echo -e '\033[43;31;5mmagedu\e[0m'
```
- \xHH插入HH（十六进制）所代表的ASCII数字（man 7 ascii）

## 5.ASCII码表

- ASCII: American Standard Code for Information Interchange

ASCII 字符代码表 一

| 高四位<br><br>低四位 |   | ASCII非打印控制字符 |            |        |       |      |      |    |      |        | ASCII 打印字符 |      |      |      |      |      |    |     |     |     |     |      |  |  |  |  |
|----------------|---|--------------|------------|--------|-------|------|------|----|------|--------|------------|------|------|------|------|------|----|-----|-----|-----|-----|------|--|--|--|--|
|                |   | 0000         |            |        |       |      | 0001 |    |      |        | 0010       | 0011 | 0100 | 0101 | 0110 | 0111 |    |     |     |     |     |      |  |  |  |  |
|                |   | 0            |            |        |       |      | 1    |    |      |        | 2          | 3    | 4    | 5    | 6    | 7    |    |     |     |     |     |      |  |  |  |  |
|                |   | +进制          | 字符         | ctrl   | 代码    | 字符解释 | +进制  | 字符 | ctrl | 代码     | 字符解释       | +进制  | 字符   | +进制  | 字符   | +进制  | 字符 | +进制 | 字符  | +进制 | 字符  | ctrl |  |  |  |  |
| 0000           | 0 | 0            | BLANK NULL | ^@ NUL | 空     | 16   | ▶    | ^P | DLE  | 数据链路转意 | 32         |      | 48   | 0    | 64   | @    | 80 | P   | 96  | `   | 112 | p    |  |  |  |  |
| 0001           | 1 | 1            | ☺          | ^A SOH | 头标开始  | 17   | ◀    | ^Q | DC1  | 设备控制 1 | 33         | !    | 49   | 1    | 65   | A    | 81 | Q   | 97  | a   | 113 | q    |  |  |  |  |
| 0010           | 2 | 2            | ☹          | ^B STX | 正文开始  | 18   | ↕    | ^R | DC2  | 设备控制 2 | 34         | "    | 50   | 2    | 66   | B    | 82 | R   | 98  | b   | 114 | r    |  |  |  |  |
| 0011           | 3 | 3            | ♥          | ^C ETX | 正文结束  | 19   | !!   | ^S | DC3  | 设备控制 3 | 35         | #    | 51   | 3    | 67   | C    | 83 | S   | 99  | c   | 115 | s    |  |  |  |  |
| 0100           | 4 | 4            | ◆          | ^D EOT | 传输结束  | 20   | ¶    | ^T | DC4  | 设备控制 4 | 36         | \$   | 52   | 4    | 68   | D    | 84 | T   | 100 | d   | 116 | t    |  |  |  |  |
| 0101           | 5 | 5            | ♣          | ^E ENQ | 查询    | 21   | ¢    | ^U | NAK  | 反确认    | 37         | %    | 53   | 5    | 69   | E    | 85 | U   | 101 | e   | 117 | u    |  |  |  |  |
| 0110           | 6 | 6            | ♠          | ^F ACK | 确认    | 22   | ■    | ^V | SYN  | 同步空闲   | 38         | &    | 54   | 6    | 70   | F    | 86 | V   | 102 | f   | 118 | v    |  |  |  |  |
| 0111           | 7 | 7            | ●          | ^G BEL | 震铃    | 23   | ↑    | ^W | ETB  | 传输块结束  | 39         | '    | 55   | 7    | 71   | G    | 87 | w   | 103 | g   | 119 | w    |  |  |  |  |
| 1000           | 8 | 8            | ◼          | ^H BS  | 退格    | 24   | ↑    | ^X | CAN  | 取消     | 40         | (    | 56   | 8    | 72   | H    | 88 | X   | 104 | h   | 120 | x    |  |  |  |  |
| 1001           | 9 | 9            | ◯          | ^I TAB | 水平制表符 | 25   | ↓    | ^Y | EM   | 媒体结束   | 41         | )    | 57   | 9    | 73   | I    | 89 | Y   | 105 | i   | 121 | y    |  |  |  |  |
| 1010           | A | 10           | ◻          | ^J LF  | 换行/新行 | 26   | →    | ^Z | SUB  | 替换     | 42         | *    | 58   | :    | 74   | J    | 90 | Z   | 106 | j   | 122 | z    |  |  |  |  |
| 1011           | B | 11           | ♂          | ^K VT  | 垂直制表符 | 27   | ←    | ^[ | ESC  | 转意     | 43         | +    | 59   | ;    | 75   | K    | 91 | [   | 107 | k   | 123 | {    |  |  |  |  |
| 1100           | C | 12           | ♀          | ^L FF  | 换页/新页 | 28   | └    | ^_ | FS   | 文件分隔符  | 44         | ,    | 60   | <    | 76   | L    | 92 | \   | 108 | l   | 124 |      |  |  |  |  |
| 1101           | D | 13           | ♪          | ^M CR  | 回车    | 29   | ↔    | ^] | GS   | 组分隔符   | 45         | -    | 61   | =    | 77   | M    | 93 | ]   | 109 | m   | 125 | }    |  |  |  |  |
| 1110           | E | 14           | 🎵          | ^N SO  | 移出    | 30   | ▲    | ^_ | RS   | 记录分隔符  | 46         | .    | 62   | >    | 78   | N    | 94 | ^   | 110 | n   | 126 | ~    |  |  |  |  |
| 1111           | F | 15           | ◻          | ^O SI  | 移入    | 31   | ▼    | ^- | US   | 单元分隔符  | 47         | /    | 63   | ?    | 79   | O    | 95 | _   | 111 | o   | 127 | Δ    |  |  |  |  |

注：表中的ASCII字符可以用:ALT + “小键盘上的数字键” 输入

6.字符集和编码

- UTF-8 在计算机上存储时可以占用1、2、3或者4个字节
  - UTF-16 在计算机上存储时占用2或4个字节
  - UTF-32 全部存储为4个字节
  - 一般在网络中传输数据时使用utf-8格式传输
  - 读取数据在内存中处理时以unicode格式存储
  - 编码转换和查询
- <http://www.chi2ko.com/tool/CJK.htm>
- <https://javawind.net/tools/native2ascii.jsp?action=transform>
- <http://tool.oschina.net/encode>

十.命令历史简介

1.命令历史是shell的特性之一

- 保存你输入的命令历史。可以用它来重复执行命令
- 登录shell时，会读取命令历史文件中记录下的命令 ~/.bash\_history
- 登录进shell后新执行的命令只会记录在缓存中；这些命令会用户退出时“追加”至命令历史文件中

2.命令历史的使用

重复前一个命令，有4种方法

重复前一个命令使用上方向键，并回车执行

按 **!!** 并回车执行

输入 **!-1** 并回车执行

按 **Ctrl+p** 并回车执行

**!:0** 执行前一条命令（去除参数）

**!n** 执行**history**命令输出对应序号**n**的命令

**!-n** 执行**history**历史中倒数第**n**个命令

**!string** 重复前一个以“**string**”开头的命令

**!?string** 重复前一个包含**string**的命令

**!string:p** 仅打印命令历史，而不执行

**!\$:p** 打印输出 **!\$**（上一条命令的最后一个参数）的内容

**!\*:p** 打印输出 **!\***（上一条命令的所有参数）的内容

**^string** 删除上一条命令中的第一个**string**

**^string1^string2** 将上一条命令中的第一个**string1**替换为**string2**

**!:gs/string1/string2** 将上一条命令中所有的**string1**都替换为 **string2**

使用**up**（向上）和**down**（向下）键来上下浏览从前输入的命令

**ctrl-r**来在命令历史中搜索命令

（**reverse-i-search**）`**':**

**Ctrl+g**: 从历史搜索模式退出

要重新调用前一个命令中最后一个参数

**!\$** 表示

**Esc**, **.**（点击**Esc**键后松开，然后点击 **.** 键）

**Alt+.**（按住**Alt**键的同时点击 **.** 键）

### 3.调用历史参数

**command !^** 利用上一个命令的第一个参数做**cmd**的参数

**command !\$** 利用上一个命令的最后一个参数做**cmd**的参数

**command !\*** 利用上一个命令的全部参数做**cmd**的参数

**command !:n** 利用上一个命令的第**n**个参数做**cmd**的参数

**command !n:^** 调用第**n**条命令的第一个参数

**command !n:\$** 调用第**n**条命令的最后一个参数

**command !n:m** 调用第**n**条命令的第**m**个参数

**command !n:\*** 调用第**n**条命令的所有参数

**command !string:^** 从命令历史中搜索以 **string** 开头的命令，并获取它的第一个参数

**command !string:\$** 从命令历史中搜索以 **string** 开头的命令，并获取它的最后一个参数

**command !string:n** 从命令历史中搜索以 **string** 开头的命令，并获取它的第**n**个参数

**command !string:\*** 从命令历史中搜索以 **string** 开头的命令，并获取它的所有参数

调用历史参数

### 4.打印历史命令列表的命令history

**history [-c] [-d offset] [n]**

**history -anrw [filename]**

**history -ps arg [arg...]**

- c: 清空命令历史
- d offset: 删除历史中指定的第offset个命令
- n: 显示最近的n条历史
- a: 追加本次会话新执行的命令历史列表至历史文件
- r: 读历史文件附加到历史列表
- w: 保存历史列表到指定的历史文件
- n: 读历史文件中未读过的行到历史列表
- p: 展开历史参数成多行, 但不存在历史列表中
- s: 展开历史参数成一行, 附加在历史列表后`

## 5.与命令历史相关的环境变量

HISTSIZE: 命令历史记录条数  
 HISTFILE: 指定历史文件, 默认为~/.bash\_history  
 HISTFILESIZE: 命令历史文件记录历史的条数  
 HISTTIMEFORMAT="%F %T " 显示时间  
 HISTIGNORE="str1:str2\*:..." 忽略str1命令, str2开头的历史  
 控制命令历史的记录方式:  
 环境变量: HISTCONTROL 可选值:

- ignoredups 默认, 忽略重复的命令, 连续且相同为“重复”
- ignorespace 忽略所有以空白开头的命令
- ignoreboth 相当于ignoredups, ignorespace的组合
- erasedups 删除重复命令

**export** 变量名="值"

编辑后可以存放在 /etc/profile (全局有效) 或 ~/.bash\_profile (某个用户有效)

## 6.bash快捷键列表

| 快捷键      | 功能                |
|----------|-------------------|
| Ctrl + l | 清屏, 相当于clear命令    |
| Ctrl + o | 执行当前命令, 并重新显示本命令  |
| Ctrl + s | 阻止屏幕输出, 锁定        |
| Ctrl + q | 允许屏幕输出            |
| Ctrl + c | 终止命令              |
| Ctrl + z | 挂起命令              |
| Ctrl + a | 光标移到命令行首, 相当于Home |
| Ctrl + e | 光标移到命令行尾, 相当于End  |
| Ctrl + f | 光标向右移动一个字符        |
| Ctrl + b | 光标向左移动一个字符        |
| Alt + f  | 光标向右移动一个单词尾       |

| 快捷键       | 功能                  |
|-----------|---------------------|
| Alt + b   | 光标向左移动一个单词首         |
| Ctrl + xx | 光标在命令行首和光标之间移动      |
| Ctrl + u  | 从光标处删除至命令行首         |
| Ctrl + k  | 从光标处删除至命令行尾         |
| Alt + r   | 删除当前整行              |
| Ctrl + w  | 从光标处向左删除至单词首        |
| Alt + d   | 从光标处向右删除至单词尾        |
| Ctrl + d  | 删除光标处的一个字符          |
| Ctrl + h  | 删除光标前的一个字符          |
| Ctrl + y  | 将删除的字符粘贴至光标后        |
| Alt + c   | 从光标处开始向右更改为首字母大写的单词 |
| Alt + u   | 从光标处开始，将右边一个单词更改为大写 |
| Alt + l   | 从光标处开始，将右边一个单词更改为小写 |
| Ctrl + t  | 交换光标处和之前的字符位置       |
| Alt + t   | 交换光标处和之前的单词位置       |
| Alt + N   | 提示输入指定字符后，重复显示该字符N次 |
| 注意        | Alt组合快捷键经常和其它软件冲突   |