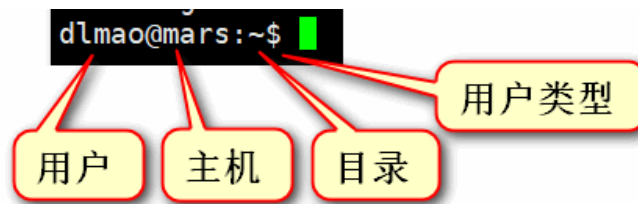


命令帮助和快捷使用

# 大纲

- 命令格式和选项
- man和info
- 命令历史
- 自动完成
- 快捷键
- 命令别名

# 命令格式



- Shell为用户提供命令行的接口
  - 输出提示符(PS1),缺省提示符给出了主机名/用户/当前工作目录/用户类型(\$表示普通用户, #表示root)
  - 等待用户输入一行命令
  - 执行该命令, 并输出结果

command [options] arguments    比如 `rm -rf test`

- command为要执行的命令: 可以是别名、shell内置命令、外部命令、shell函数等
- 选项给出了命令执行时要使用的可选的属性或者功能
- 参数: 是命令作用的对象, 0个或者多个
  - 在无法区分选项和参数部分时, 通常采用`--` 表示**选项部分的结束**, 后面为命令的参数, 比如`rm -- -rf test`

# 命令格式： 选项

command [options] arguments

选项给出了命令执行时要使用的可选的属性或者功能

```
ls -a -F ==> ls -aF
```

```
cp -f -t backup *.txt # backup为选项t的参数
```

```
cp -f -tbackup *.txt # 与上条命令一致
```

```
cp -ft backup *.txt # 与上条命令一致
```

```
ls --help
```

```
ls --hide=Desktop
```

- **短选项**以连字符-开始，紧跟单个字符，最后可能跟着用于该选项的参数。
  - 那些没有参数的多个短选项可以合并，这时连字符后面跟着多个对应着某短选项的字符
  - 短选项有参数时，一般在参数前有空格（有的时候也可无空格）
  - 两个带有参数的短选项显然不能合并一起，但一个有参数的短选项可以和其他无参数短选项合并，合并时，前面为无参数选项，最后一个为有参数的选项
- **长选项**以两个连字符--开始，后面跟着表示哪个选项的单词，后面可能还包括参数
  - 如果后面有参数，则一般用option=value的形式描述，**注意=前后没有空格**
- 几乎所有命令都支持**--help**和--version，用于显示命令的简略帮助信息和版本号

# 外部命令帮助： 软件文档和联机手册

- 软件文档一般存放在/usr/share/doc中，有些是文本、(gzip)压缩过的文本文件(.gz) 或者html文件，可通过less、**zless**和web浏览器查看
- 联机手册(manual)
  - man查看命令的联机手册，采用less分页显示
  - 每个命令包含的部分可能各不一致，但几乎所有的命令包括了NAME、SYNOPSIS和DESCRIPTION
  - man -f commands或者whatis commands查看命令的NAME部分的内容
  - man -k keyword或者apropos -r keyword搜索(单行描述)的NAME部分，列出匹配的命令

```
man [options] [section] command...  
-f, --whatis 查看命令的NAME部分内容  
-k, --apropos 搜索NAME部分的单行描述中包含相应关键字的命令  
-s, --sections=LIST 在指定的节搜索，节之间以冒号或逗号隔开  
-a, --all 显示所有节里匹配的命令，而不是第一个匹配的命令
```

头部名	含义
<b>NAME</b>	命令的名称和作用
<b>SYNOPSIS</b>	命令的语法，大纲
<b>DESCRIPTION</b>	命令的完整描述，最主要的部分
EXAMPLES	命令使用的例子
ENVIRONMENT	命令相关的环境变量
FILES	和该命令相关的文件
AUTHOR	作者
<b>SEE ALSO</b>	查看相关信息

# Manual Section

- 联机手册一般保存在目录/usr/share/man
- 根据命令的类型，分类存放在不同的目录(如man1/man2...)，对应着节（section）

节	含义
1	执行程序或者shell命令
2	系统调用
3	库函数
4	特殊文件（主要是设备文件）相关，还包括设备驱动程序
5	供程序员和系统管理员参考，主要给出了配置文件的内容和格式
6	与游戏相关
7	杂项信息，包括man
8	系统管理员使用的系统命令

- ✓ -s, --sections=LIST 在**指定的节里面搜索**，节之间以冒号或逗号隔开，如man -s 1,2 kill
- ✓ -a, --all 查看**所有相关节**(缺省所有节, 可通过-s选项指定搜索范围)里面指定命令的联机手册, 如man -a kill
- ✓ man SECTION COMMAND或者man COMMAND.SECTION(如man 2 kill或man kill.2) 查看**指定节**中命令为COMMAND的手册

# Man命令示例

```
/usr/share/man$ man kill
```

KILL(1)

User Commands

KILL(1)

NAME

kill - send a signal to a process

SYNOPSIS

kill [options] <pid> [...]

DESCRIPTION

...

SEE ALSO

kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), skill(1)

...

Manual page kill(1) line 1 (press h for help or q to quit)

demo@mars:~\$ man -a kill # 查看所有节里面kill的联机手册页, kill命令和kill函数

...

demo@mars:~\$ man 2 kill # 查看节2里面的kill联机手册

KILL(2)

Linux Programmer's Manual

KILL(2)

NAME

kill - send signal to a process

SYNOPSIS

```
#include <sys/types.h>
```

```
#include <signal.h>
```

# whatis: 根据命令名查找，显示命令的NAME部分的内容

man -f name...

精确匹配(完全匹配)、通配符匹配(-w, 完全匹配)、正则表达式匹配 (部分匹配 -r)

whatis [options] name ...

搜索联机手册的索引库，找到命令名与name匹配的命令并显示其NAME部分内容

- -e, --exact: 采用精确匹配, 缺省为精确匹配
- -r, --regex 正则表达式匹配, 命令名部分匹配, 比如 **whatis -r link**表示查找命令名包含link的所有命令
- -w, --wildcard shell通配符匹配, 命令名全部匹配, 比如**whatis -w 'r\*link'** 表示查找所有命令名以r开头link结尾的所有命令
- -s list, --sections list, --section list, 指定在哪些节中查找

```
demo@mars:~$ whatis id man # 等价于man -f id man
id (1)                - print real and effective user and group IDs
man (7)                - macros to format man pages
man (1)                - an interface to the on-line reference manuals
demo@mars:~$ whatis -s 1,8 -r kill # 在节1和8中查找包含link的命令
# whatis -s 1,2 -w '*kill*'      注意最好加上引号
kill (1)               - send a signal to a process
killall (1)            - kill processes by name
pkill (1)              - look up or signal processes based on name and other at...
skill (1)              - send a signal or report process status
xkill (1)              - kill a client by its X resource
```



# apropos: 根据关键字查找命令

- `apropos [OPTION...] KEYWORDS`: 在**命令名或NAME**部分的**单行描述中**，寻找包含有指定关键字（不区分大小写）的命令，显示NAME部分内容
  - `-r, --regex`: **缺省，采用正则表达式**
  - `-e, --exact`: 采用精确匹配
  - `-w, --wildcard`: 采用通配符匹配
  - **`-a, --and`**: 匹配所有的关键字，**缺省匹配任意一个**
  - `-s list, --sections list, --section list` 指定在哪些节中查找
- `man -k KEYWORD`与`apropos`命令在功能上基本一致，但选项部分不大支持

```
demo@mars:~$ man -k who
```

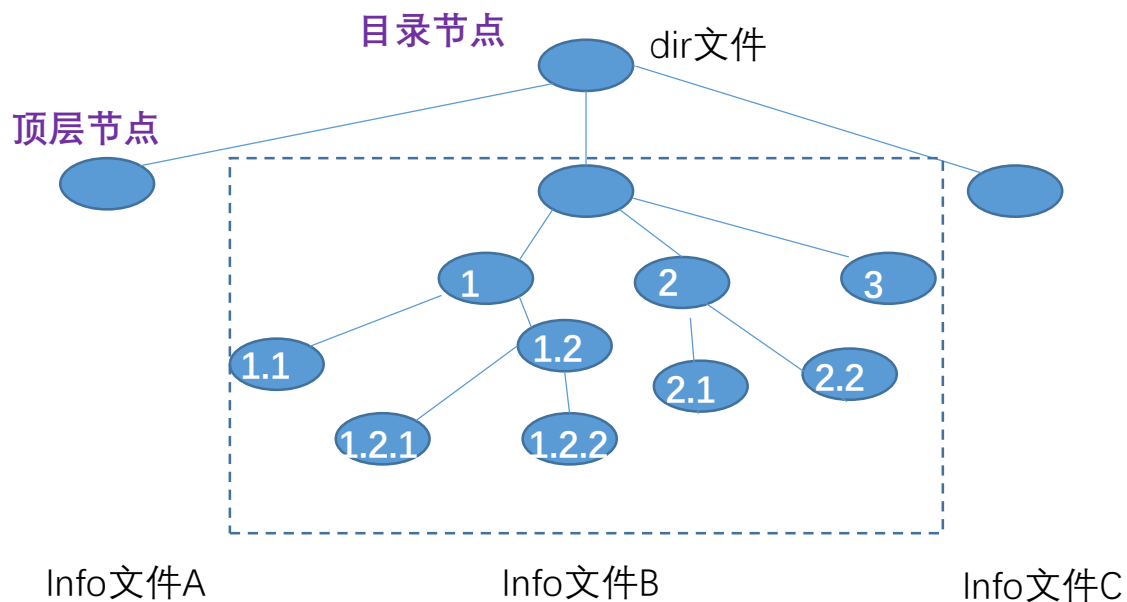
```
bsd-from (1)      - print names of those who have sent mail
from (1)          - print names of those who have sent mail
w (1)             - Show who is logged on and what they are doing.
w.procps (1)      - Show who is logged on and what they are doing.
who (1)           - show who is logged on
whoami (1)        - print effective userid
```

```
demo@mars:~$ apropos -a list directory
```

```
chacl (1)         - change the access control list of a file or directory
dir (1)           - list directory contents
File::Listing (3pm) - parse directory listing
ls (1)            - list directory contents
ntfsls (8)        - list directory contents on an NTFS filesystem
vdir (1)          - list directory contents
```

# info: 阅读Info文档

- GNU提供了联机帮助系统info, info文件(文本文件) 一般保存在目录/usr/share/info中
- info文档以树的形式来组织, 如同我们平时通过章节和各级标题组织一样
  - 所有info文档的根为**目录节点**, 对应文件/usr/share/info/dir
  - 根的子节点为**顶层节点**, 对应着不同的info文件
  - 顶层节点下面进一步有子节点, 如此继续, 每个节点对应着该info文件的**某个主题(菜单项)**



Info文件B的线性方式的节点顺序, 平时阅读顺序  
1→1.1→1.2→1.2.1→1.2.2→2→2.1→2.2→3

命令info [OPTOIN] ... [MENU-ITEM...]

**H** 可查看可用的info命令, 在帮助系统中通过**l(ast)**退出帮助  
**q** 退出info系统

- info树以菜单的形式组织下层的节点, 菜单项以\*开头, 通过<tab>或者<shift-tab>移动到下一个或上一个菜单项, Enter进入该节点
- d(ir)、t(op) 进入目录节点、顶层节点
- u(pper): 进入父节点
- n(ext)、p(revious): 进入同级下一节点、前一节点
- 注: top节点的next节点为其下一级的第一个节点
- l(ast) 上一个查看的节点

**右中括号]**和**左中括号[**切换到线性的下一个和上一个节点

# info: 阅读

- 通过箭头、<PageUp><PageDown>等移动阅读位置
- 空格键<Space>与<PageDown>类似，每次显示下一屏，到达当前节点的底部时会跳转(相当于命令j)到线性方式的下一个节点
  - PageUp和PageDown在有的系统中同样有到顶部或底部跳转的功能
- <BackSpace>或者<Delete>键与<PageUp>类似，显示上一屏，到达当前节点的顶部时跳转到(相当于命令k)线性方式的上一个节点
- b(begin)或<Home>， e(end)或<End>可以移动位置到当前节点的顶部和尾部
  - <Home>/<End>键在有的系统可能不支持
- s(earch)或/来在当前文档中向前(正向)搜寻匹配的文本， 匹配的模式采用正则表达式， 而要向后(反向)搜寻， 则采用?命令

# 命令历史

- 交互式shell可以使用命令历史，历史列表选项缺省打开 (set -o history)
- 用户输入的命令(在进行历史扩展之后)会被保存到历史列表
- 支持跨越多个shell会话，用户退出交互式shell会话时会将历史命令保存在 \$HISTFILE(~/.bash\_history); 下次开启shell会话时从该文件中加载

HISTSIZE	保存在历史列表中的命令个数的最大值
HISTFILE	用于保存历史纪录的文件名
HISTFILESIZE	历史纪录文件中的命令个数的最大值
HISTCONTROL	控制哪些命令保存到历史中 <ul style="list-style-type: none"><li>• ignoredups: 忽略重复的命令;</li><li>• ignorespace: <b>忽略以空白字符开头的命令</b>;</li><li>• ignoreboth: 以上两者同时生效，缺省</li></ul>

```
demo@mars:~$ set |grep HIST
```

```
HISTCONTROL=ignoreboth
```

```
HISTFILE=/home/demo/.bash_history
```

```
HISTFILESIZE=2000
```

```
HISTSIZE=1000
```

# 命令历史

内置命令fc可以列出、编辑和重新执行历史列表中的某些命令

- 历史列表中的每条命令称为一个事件(event)，而每个事件都有一个内部编号，称为事件编号
- 查看历史列表:
  - history [n] 如果指定n, 则仅给出最近的n条命令
  - fc -l [first] [last] 列出历史列表, 通过first/last指定范围, first可以为-N, 表示倒数第N条, 缺省为-15, 即列出最近15条命令。比如fc -l -5
- 浏览和搜索历史纪录
  - Ctrl-P(Previous)或**向上箭头键**: 移动到前一条历史纪录
  - Ctrl-N(Next)或者**向下箭头**: 移动到下一条历史纪录
  - Ctrl-R键开始以增量方式反向搜索历史纪录; ~~Ctrl-S正向搜索历史纪录~~
    - 如果要退出搜索, 按<Ctrl-C>即可
    - 每次一个新的字符输入可能会改变匹配的历史命令
    - 再次按<Ctrl-R>搜索下一个匹配的命令
    - 找到要使用的命令时, 按<Enter>键运行, <Tab>复制到命令行

```
1027 du > usags
1028 ls -ald .gvfs
1029 date
```

# 历史记录扩展

```
1027 du > usags
1028 ls -ald .gvfs
1029 date
```

!word, 表示历史记录扩展, 在扩展时也可以修改展开的内容

- 注意!后面有空格时, 不是历史记录扩展, 而是以后介绍的非运算符
- !后紧跟事件编号表示该事件编号对应的历史命令, 编号为负数, 表示最近倒数第几条命令, 比如!1027以及!-3

- !!表示上一条命令, 等价于!-1

- !string表示最近的以string开始的历史命令

- !?string表示最近的包含string的历史命令

- 在重新执行历史命令之前可对命令进行小的修改,

```
dlmao@mars:~$ more /etc/shadow
more: cannot open /etc/shadow: Permission denied
dlmao@mars:~$ less !!:
less /etc/shadow
/etc/shadow: Permission denied
dlmao@mars:~$ sudo !!
sudo less /etc/shadow
```

- :s/pattern/replacement/, 表示该历史命令中第一个出现的pattern替换为replacement
- :p 仅仅输出扩展后的命令但不执行, 但进入历史记录中
- :<n> 历史命令的第n个单词, 单词编号从0开始, \$表示最后一个单词
- :<x>-<y> 历史命令的第x到y个单词, x省略时为0, y省略时为\$
- **:\* 等价于:1-\$**
- ^pattern^replacement表示对于最近一个命令进行修改, 相当于 !!:s/pattern/replacement

<b>demo@mars:~\$ history 4</b> 1027 du > usags 1028 ls -ald .gvfs 1029 date 1030 history 4 <b>demo@mars:~\$ !1029</b> date Mon Aug 29 01:12:15 CST 2016 <b>demo@mars:~\$ !!</b> date Mon Aug 29 01:12:43 CST 2016 <b>demo@mars:~\$ !ls</b> ls -ald .gvfs	drwx----- 2 root root 4096 Aug 3 14:28 .gvfs <b>demo@mars:~\$ !?FILE</b> echo \$HISTFILE /home/demo/.bash_history <b>demo@mars:~\$ !du:s/usags/usages</b> du > usages du: cannot read directory './.cache/dconf': Permission denied du: cannot read directory './.gvfs': Permission denied <b>demo@mars:~\$ less usags</b> ... <b>demo@mars:~\$ ^usags^usages</b> less usages ...
--	---

命令或组合键	作用
!number	事件编号为number的历史命令
!!	最近一条历史命令
!string	最近的以string开始的历史命令
!?string	最近的包含string的历史命令
!number: s/old/new/	指定历史命令中第一个出现的old替换为new
!!:*	上一条命令中除了命令名外的所有单词
^pattern^replacement	上一条命令修改，!!:s/pattern/replacement
<Ctrl-P>或向上箭头	移动到前一条历史记录
<Ctrl-N>或向下箭头	移动到下一条历史记录
<Ctrl-R>或<Ctrl-S>	以递增方式反向(或正向)搜索历史记录
<Ctrl-C>	退出搜索
<Tab>	找到的命令复制到当前命令行

内置命令fc列出/编辑/重新执行历史命令
 

- fc -l [first] [last] 列出历史列表
- fc [first] [last] 通过编辑器修改一条或多条命令, 在保存后顺序执行这些命令
- fc -s [pat=rep ...] [cmd]** cmd可为事件编号或字符串, 表示重新执行某个命令或历史记录中最后以cmd开始的命令, pat=rep给出了执行前可以进行修改

 fc -s 相当于 !!  
 fc -s cmd 相当于 !cmd



# 自动完成

输入命令的过程中，按<Tab>键或<Ctrl-I>触发自动完成功能

- 如果只有一个可选项，则自动补齐后面的内容
- 如果有多个可选项或者没有可选项时会发出提示声表示有歧义
  - 可以选择输入更多的字符
  - 或再按一次<Tab>或<Ctrl-I>键列出所有的可选项

自动完成支持

- 命令名（如果为第一个单词）
- 变量（如果单词以\$开头）
- 命令参数：路径和文件名的自动完成
- 用户名（如果单词以波浪号(~)开头）
- 主机名（如果单词以@开头）

```
demo@mars:~$ ls .bash<Tab>
.bash_aliases  .bash_history  .bash_logout  .bashrc
demo@mars:~$ ls<Tab>
ls              lsblk          lsdiff         lsinitramfs   lsmod          lspcmcia
lsattr          lscpu          lslocks        lslogins      lsof           lsusb
lsb_release    lshw           lsppot
demo@mars:~$ echo $HIST<Tab>
$HISTCMD      $HISTCONTROL  $HISTFILE     $HISTFILESIZE $HISTSIZE
```



# 命令别名

- 交互式shell中支持命令别名，脚本中不支持
- 建议将其添加到bash配置文件：用户HOME目录下的 .bashrc中
- 用户输入的命令的第一个单词会在别名列表中查找，如果已设置，则该单词会被替换为所设置的别名的值
- alias [name[=value] ... ]: 查看或者设置别名
  - VALUE里面的命令也可以是别名，此时继续进行别名扩展；但是如果该命令的名字与别名NAME一致，则不会再次进行别名扩展（否则造成递归循环）
  - 注意等号前后不允许有空格
  - 允许在一行设置多个别名

```
alias name1=value1 name2=value2
```
- unalias [-a] name [name ...]: 取消别名
  - unalias -a 取消所有别名
- 命令名前加一个反斜杠\表示不进行别名扩展

## **\$ alias**

```
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -aF'
alias ls='ls --color=auto'
$ alias ..='cd ..'
$ alias h='history'
$ alias cls='clear' r='fc -s'
$ \ls
...
```