

Linux操作系统

02 Linux常用命令

主讲：杨东平
中国矿大计算机学院

Linux常用命令

- Linux中命令超过3000个，常用的达200多个，掌握几十个就差不多了
- ❖ (1) 命令基本格式
 - ❖ (2) 文件处理命令
 - ❖ (3) 文件搜索命令
 - ❖ (4) 帮助命令
 - ❖ (5) 压缩与解压缩命令
 - ❖ (6) 关机与重启命令
 - ❖ (7) 其他常用命令

命令的基本格式

- 命令提示符
- ❖ 格式：`[root@localhost ~]#`
 - `root` 当前登录的用户名
 - `localhost` 主机名(安装时可修改)
 - `~` 用户当前所在的目录(家目录)
 - `#` 超级用户提示符
 - `$` 普通用户提示符
 - ❖ 家目录即用户登录后的初始登录位置
 - ◆ 管理员：`/root`
 - ◆ 普通用户：每个普通用户都会在`/home`目录下有一个自己的子目录，它是普通用户的初始登录位置
 - 普通用户子目录在创建用户时自动生成

命令的基本格式(续)

- 命令基本语法：命令 [选项] [参数]
- ❖ 命令、选项、参数之间用空格分隔
 - ❖ 选项：用来调整命令的功能
 - ❖ 长格式选项：英文完整单词，一般前面用两个减号
 - ◆ 如：`--all`
 - ❖ 短格式选项：英文缩写，一般用一个减号加一个字母
 - ◆ 如：`-a`
 - ❖ 通常，一个短格式选项会有对应的长格式选项，但个别命令不遵循此格式
 - ◆ 如：`-a`等于`--all`
 - ❖ 当有多个选项时，可以写在一起
 - ◆ 如：`-a`和`-l`可以写成`-al`
 - ❖ 参数：是命令的操作对象
 - ❖ 命令一般都需要参数，如果省略则一般都有默认参数

查询目录中内容命令 ls

- 原文：`list`
- 格式：`ls [-alrtAFR] [文件或目录]`
- a 显示所有文件及目录 (ls默认将“.”开头的文件名或目录名视为隐藏档，不会列出)
 - l 除文件名外，还列出文件类型、权限、拥有者、文件大小等信息
 - r 将文件以相反次序显示(默认按英文字母次序)
 - t 将文件依建立时间之先后次序列出
 - A 同 -a，但不列出“.”(当前目录)及“..”(父目录)
 - F 在文件名后加文件类别标志，如可执行档则加“*”，目录则加“/”
 - R 递归显示，列出子目录及其下的文件
- 功能：显示指定工作目录下的内容(所含之文件及子目录)，如果不指定文件或目录则显示当前目录的内容

查询目录中内容命令 ls(续)

- `ls -l`
- ```
drwxrwxr-x 2 lx gx 4096 Jun 27 2001 d1/
-rw-rw-r-- 1 lx gx 2236 Aug 16 10:38 a1
```
- ❖ 第1位表示文件类型
    - ❖ 常用：`-` 普通文件 `d` 目录文件 `l` 软链接文件
    - ❖ 不常用：`b` 块设备文件 `c` 字符设备文件 `s` 套接字文件 `p` 管道文件
  - ❖ 2-4位表示文件所有者权限，5-7位表示同组用户的权限，8-10位表示其他用户的权限
    - `r` 读 `w` 写 `x` 执行 `-` 无权限
  - ❖ 其后的数字表示文件的硬链接的引用计数(例中：1、2)
  - ❖ 接着是文件的所有者(lx)和文件的所属组名(gx)
  - ❖ 文件大小，单位字节
  - ❖ 文件的最后修改时间和文件名或目录名

查询目录中内容命令 ls(续)

```
[root@localhost ~]# ls -l
total 28
-rw-----. 1 root root 1099 Sep 6 16:23 anaconda-ks.cfg
-rw-r--r--. 1 root root 8815 Sep 6 16:23 install.log
-rw-r--r--. 1 root root 3384 Sep 6 16:22 install.log.syslog
```

```
[root@localhost ~]# ls -ld
dr-xr-x---. 2 root root 4096 Sep 6 17:33
```

```
[root@localhost ~]# ls -al
total 52
dr-xr-x---. 2 root root 4096 Sep 6 17:33 .
dr-xr-xr-x. 22 root root 4096 Sep 6 16:23 ..
-rw-----. 1 root root 1099 Sep 6 16:23 anaconda-ks.cfg
-rw-----. 1 root root 58 Sep 6 17:33 .bash_history
-rw-r--r--. 1 root root 188 May 28 2009 .bash_logout
-rw-r--r--. 1 root root 176 May 28 2009 .bash_profile
-rw-r--r--. 1 root root 176 Sep 23 2004 .bashrc
-rw-r--r--. 1 root root 100 Sep 23 2004 .cshrc
-rw-r--r--. 1 root root 8815 Sep 6 16:23 install.log
-rw-r--r--. 1 root root 3384 Sep 6 16:22 install.log.syslog
-rw-r--r--. 1 root root 129 Dec 4 2004 .tcshrc
```

文件处理命令

- 目录处理命令
- 文件处理命令
- 链接命令

文件处理命令(续)

➢ 切换目录: cd

- ❖ 原文: change directory
- ❖ 格式: **cd [目标目录]**
- ❖ 简化操作:

- cd ~** 进入当前用户的家目录
- cd** 进入当前用户的家目录
- cd -** 进入上次目录
- cd ..** 进入上一级(父)目录
- cd .** 进入当前目录
- cd /** 进入到根目录

```
[root@localhost ~]# cd
[root@localhost ~]# cd /dev
[root@localhost dev]# cd /
[root@localhost /]# cd -
/dev
[root@localhost dev]# cd ..
[root@localhost /]# pwd
/
[root@localhost /]# cd /dev
[root@localhost dev]# pwd
/dev
[root@localhost dev]# _
```

➢ 查看当前路径: pwd

- ❖ 原文: print working directory
- ❖ 格式: **pwd**

文件处理命令(续)

➢ 建立目录: mkdir

- ❖ 原文: make directories
- ❖ 格式: **mkdir [-p] [目录名]**  
-p/--parents 同时建立目录名的上层目录(未建时)

```
[root@localhost /]# ls
bin dev home lib64 media opt root selinux src usr
boot etc lib lost+found mnt proc sbin srv tmp var
[root@localhost /]# mkdir -p computer/yang
[root@localhost /]# ls
bin computer etc lib lost+found mnt proc sbin srv tmp var
boot dev home lib64 media opt root selinux src usr
```

| 颜色   | 含义                      |
|------|-------------------------|
| 浅蓝色  | 链接文件                    |
| 灰色   | 其他文件                    |
| 绿色   | 可执行文件                   |
| 红色   | 压缩文件                    |
| 蓝色   | 目录                      |
| 红色闪烁 | 链接的文件有问题了               |
| 黄色   | 设备文件, 包括block、char、fifo |

文件处理命令(续)

➢ 删除空目录: rmdir

- ❖ 原文: remove empty directories
- ❖ 格式: **rmdir [-p] [目录名]**  
-p 当子目录被删除后使它也成为空目录, 则一并删除
- 说明: 只能删除空目录, 较少使用

➢ 删除文件或目录: rm

- ❖ 原文: remove
- ❖ 格式: **rm -irf [文件或目录]**  
-r 递归删除, 删除目录及其下的文件和子目录  
-f 强制(force)删除文件或目录, 无需确认  
-i 删除前逐一询问确认
- 说明: 可以删除一个目录中的一个或多个文件或目录, 也可以将某个目录及其下属的所有文件及其子目录均删除掉。对于链接文件, 只是删除整个链接文件, 而原有文件保持不变
- 注意: **rm -rf /** 后果严重!!! 删除根目录下所有文件

文件处理命令(续)

➢ 复制命令: cp

- ❖ 原文: copy
- ❖ 格式: **cp [选项] [原文件名或目录] [目标文件名或目录]**  
-a 相当于-dpr组合  
-d 若源文件是链接文件, 则复制链接属性  
-f 覆盖已经存在的目标文件而不给出提示  
-i 提示并要求用户确认是否覆盖目标文件, 回答"y"时目标文件将被覆盖  
-p 除复制文件的内容外, 还把修改时间和访问权限也复制到新文件中  
-r 若给出的源文件是一个目录文件, 则将复制该目录下所有的子目录和文件  
-l 不复制文件, 只是生成链接文件
- 说明: 不加目标文件名是原名复制, 否则是改名复制

文件处理命令(续)

➤剪切或改名命令: mv

- ❖ 原文: move
- ❖ 格式: mv [参数] [原文件或目录] [新文件或目标目录]
  - i 若指定目录已有同名文件, 则询问是否覆盖旧文件
  - f 直接覆盖已有目标文件, 不给出任何提示

❖ 说明:

- ☞ 原文件和目标文件不在同一个目录下, 是**剪切(移动)**
- ☞ 原文件和目标文件在同一个目录下, 是**改名**

文件处理命令(续)

➤链接命令: ln

- ❖ 原文: link
- ❖ 格式: ln [参数] [原文件] [目标文件]
- ❖ 功能: 生成链接文件
  - b 删除, 覆盖以前建立的链接
  - d 允许超级用户制作目录的硬链接
  - f 强制执行
  - i 交互模式, 文件存在则提示用户是否覆盖
  - n 把符号链接视为一般目录
  - s 软链接(符号链接)
  - v 显示详细的处理过程

硬链接与软链接

➤Linux的文件与目录

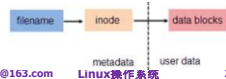
- ❖ Linux系统中除进程之外的一切皆文件, 包括输入输出设备及网络通信、目录等
- ❖ 目录是记录了其他文件名的文件, 目录使文件可被分类管理, 使Linux的文件系统形成一个层级结构的目录树。下图是普通Linux系统的顶层目录结构:



硬链接与软链接(续)

➤Linux文件的组成

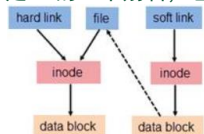
- ❖ 由用户数据(user data)与元数据(metadata)两部分组成
  - ☞ 用户数据, 即文件数据块(data block), 是记录文件真实内容的地方
  - ☞ 元数据是文件的附加属性, 如文件大小、创建时间、所有者等信息
    - ◆ 元数据中的inode号(索引节点号)是文件的唯一标识, 系统或程序通过inode号寻找正确的文件数据块
    - ◆ inode是元数据的一部分但并不包含文件名
    - ◆ 文件名仅是为了方便人们的记忆和使用
- ❖ 程序通过文件名获取文件内容的过程:



硬链接与软链接(续)

➤硬链接

- ❖ 为解决文件共享, Linux系统引入了两种链接: 硬链接(hard link)与软链接(又称符号链接, 即soft link或symbolic link)
- ❖ 链接还带来了隐藏文件路径、增加权限安全及节省存储等好处
- ❖ 若一个inode号对应多个文件名, 则称这些文件为硬链接。换言之, 硬链接就是同一个文件使用了多个别名
- ❖ 例: hard link是file的一个别名, 它们有共同的inode



硬链接与软链接(续)

➤硬链接的特性:

- ❖ 文件有相同的inode及data block
- ❖ 只能对已存在的文件进行创建
- ❖ 不能交叉文件系统创建硬链接
- ❖ 不能对目录进行创建, 只可对文件创建
- ❖ 删除一个硬链接文件并不影响其它有相同inode号的文件

➤需要注意的是

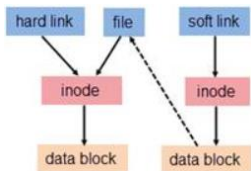
- ❖ Linux存在inode号被用完但磁盘空间还有剩余的情况
- ❖ Linux文件系统目录均隐藏了两个特殊的目录:
  - ☞ 当前目录: .
  - ☞ 父目录: ..

## 硬链接与软链接(续)



### ➤ 软链接

- ❖ 若文件用户数据块中存放的内容是另一个文件的路径名的指向，则该文件就是**软连接**
- ❖ **软链接就是一个普通文件**，只是数据块内容有点特殊
- ❖ 软链接有自己的inode号以及用户数据块，因此软链接的创建与使用没有类似硬链接的诸多限制



网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

19

## 硬链接与软链接(续)



### ➤ 软链接的特点：

- ❖ 软链接有自己的文件属性及权限等
- ❖ 可对不存在的文件或目录创建软链接
- ❖ 软链接可交叉文件系统
- ❖ 软链接可对文件或目录创建
- ❖ 创建软链接时，链接计数i\_nlink不会增加
- ❖ 删除软链接并不影响被指向的文件
  - 但若被指向的原文件被删除，则相关软链接被称为**死链接**(即dangling link，若被指向路径文件被重新创建，死链接可恢复为正常的软链接)

### ➤ 软链接的用户数据也可以是另一个软链接的路径，其解析过程是**递归**的

- ❖ 需注意：软链接创建时原文件的**路径指向**使用绝对路径较好。使用相对路径创建的软链接被移动后该软链接文件将成为一个死链接，因为链接数据块中记录的亦是相对路径指向

网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

20

## 文件搜索命令



- 文件搜索命令 locate
- 命令搜索命令 whereis 与 which
- 文件搜索命令 find
- 字符串搜索命令 grep
- find 命令与 grep 命令的区别

网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

21

## 文件搜索命令(续)



### ➤ 文件搜索命令：locate

- ❖ 格式：**locate [选项] 文件名**

**-d/--database=数据库** 指定搜索数据库，默认为/var/lib/mlocate.db

- ❖ 注意：locate只在数据库中搜索，速度快。但数据库不能实时更新，只有在更新数据库后才能搜索新建文件
- ❖ 缺点：只能按文件名搜索，功能较弱(find最强)

### ❖ 更新搜索数据库命令：

- 格式：**updatedb**

网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

22

## 文件搜索命令(续)



### ➤ 搜索命令的命令：**whereis**

- ❖ 格式：**whereis [选项] [命令名]**
- ❖ 功能：搜索**命令(二进制文件)**及帮助文档所在路径
  - d 只搜索二进制文件
  - m 只搜索 man 手册
  - s 只搜索源代码
  - f 终止 <目录> 参数列表
  - u 搜索不常见记录
  - l 输出有效查找路径

```
[root@localhost ~]# whereis -b whereis
whereis: /usr/bin/whereis
[root@localhost ~]# whereis -m whereis
whereis: /usr/share/man/man1/whereis.1.gz
[root@localhost ~]# whereis -s whereis
whereis:
[root@localhost ~]# whereis -bm whereis
whereis: /usr/bin/whereis /usr/share/man/man1/whereis.1.gz
```

网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

23

## 文件搜索命令(续)



### ➤ 搜索命令的命令：**which**

- ❖ 格式：**which 命令名**
- ❖ 功能：在环境变量\$PATH设置的目录里查找符合条件的文件

### ➤ which与whereis区别：

- ❖ (1) which可以看见命令的别名，看不到帮助文档，而whereis则相反

```
[root@localhost ~]# which ls
alias ls='ls --color=auto'
/bin/ls
```
- ❖ (2) 有命令文件的是**外部命令**，可以用whereis找到外部文件及其帮助文件，可以用man看外部文件的帮助

```
[root@localhost ~]# whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
```
- ❖ (3) 无文件的是**shell命令(内置命令)**，只能用help看帮助
  - 例：**help cd**

网络安全与网络工程系新东平 jsxhbc@163.com

Linux操作系统

2021年3月10日9时56分

24

文件搜索命令(续)



文件搜索命令：find

- ❖ 格式：find [搜索路径] [选项] [文件名]
- ❖ 功能：在指定目录下查找文件(从指定目录开始搜索)
  - ☞ 例：find / -iname install.log
- ❖ 说明：
  - ☞ 应避免大范围搜索(非常耗费系统资源)
  - ☞ find是在系统当中搜索符合条件的文件名(完全匹配)
  - ☞ 可以使用通配符进行模糊查询
  - ☞ 速度慢，避免搜索整个根目录

Linux命令中的通配符

- \* 匹配任意内容
- ? 匹配任意一个字符
- [] 匹配任意一个中括号内的字符

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 25

文件搜索命令(续)



find举例

```
find /root -iname install.log
☞ -iname: 搜索时忽略大小写

[root@localhost ~]# find /root -iname install.log
/root/install.log

find /root -user root
☞ -user: 按所有者搜索

[root@localhost ~]# find /root -user root
/root
/root/install.log.syslog
/root/.tcshrc
/root/.install.log
/root/.bash_history
/root/.bash_logout
/root/.cshrc
/root/.bash_profile
/root/anaconda-ks.cfg
/root/.bashrc

find /root -nouser
☞ -nouser: 查找没有所有者的文件

[root@localhost ~]# find /root -nouser
[root@localhost ~]# _
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 26

文件搜索命令(续)



Linux是严格区分大小写的

- 一般地，无所有者的文件都是垃圾文件，但是：
  - ❖ (1) 内核直接生成的文件无所有者，由内核直接调用，不需要用户，不能删
  - ❖ (2) 外来数据无所有者，比如U盘拷贝来的文件

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 27

文件搜索命令(续)



find举例(续)

- ```
find /var/log -mtime +10
❖ 查找10天前修改的文件
❖ 关于时间格式及选项：
-10 10天内修改的
10 当前日期前恰好第10天修改的
+10 10天前修改的
❖ 时间参数：
-atime 文件访问时间（单位：分钟）
-ctime 改变文件属性时间（单位：分钟）
-mtime 修改文件内容时间（单位：天）
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 28

文件搜索命令(续)



find举例(续)

- ```
find . -size 25k
❖ 查找当前目录下文件大小是25KB的文件
❖ 关于文件大小的格式：
-25k 小于 25KB 的文件
25k 等于 25KB 的文件
+25k 大于 25KB 的文件
```

- ```
find -inum 391692
❖ 查找当前目录下索引号是391692的文件
```

```
[root@localhost ~]# ls -li
total 20
391692 -rw----- 1 root root 1899 Sep  6 16:23 anaconda-ks.cfg
391683 -rw-r--r-- 1 root root 8815 Sep  6 16:23 install.log
391684 -rw-r--r-- 1 root root 3384 Sep  6 16:22 install.log.syslog
[root@localhost ~]# find -inum 391692
./anaconda-ks.cfg
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 29

文件搜索命令(续)



find 举例(续)

- ```
find /etc -size +20k -a -size -50k
❖ 查找 /etc/ 目录下，大于 20KB 且小于 50KB 的文件
❖ 逻辑选项：
-a 逻辑与(and)，两个条件都满足
-o 逻辑或(or)，两个条件满足一个即可
```

```
[root@localhost ~]# find /etc -size +20k -a -size -50k
/etc/sysconfig/network-scripts/network-functions-ipv6
/etc/makedev.d/01linux-2.6.x
/etc/postfix/main.cf
/etc/lvm/lvm.conf
/etc/selinux/targeted/modules/active/modules/nagios.pp
/etc/selinux/targeted/modules/active/modules/xquest.pp
/etc/selinux/targeted/modules/active/modules/cups.pp
/etc/selinux/targeted/modules/active/modules/unconfineduser.pp
/etc/selinux/targeted/modules/active/modules/postfix.pp
/etc/selinux/targeted/modules/active/modules/unprivuser.pp
/etc/selinux/targeted/modules/active/modules/virt.pp
/etc/selinux/targeted/modules/active/modules/rhcs.pp
/etc/selinux/targeted/modules/active/modules/samba.pp
/etc/selinux/targeted/modules/active/modules/staff.pp
/etc/selinux/targeted/modules/active/modules/apache.pp
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2021年3月10日9时56分 30

文件搜索命令(续)



➤ find 举例(续)

find /etc -size +20k -a -size -50k -exec ls -lh {} \;

❖ 查找 /etc/ 目录下, 大于 20KB 且小于 50KB 的文件, 并显示详细信息

❖ -exec 以 ; 结束, 防止歧义一般用 \;

❖ {} 代表 find 查找出来的文件名

❖ ls 的 -h 选项使结果人性化显示, 而不是默认的 byte 单位

```
drwxr-xr-x. 1 root root 40K Mar 6 2012 /etc/passwd/main.cf
-rw-r--r--. 1 root root 44K Mar 24 2012 /etc/passwd.conf
-rw-r--r--. 1 root root 21K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/agent.pp
-rw-r--r--. 1 root root 26K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/agent.pp
-rw-r--r--. 1 root root 21K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/certs.pp
-rw-r--r--. 1 root root 29K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/mduer.pp
-rw-r--r--. 1 root root 31K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 36K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 23K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 26K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 25K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 42K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
-rw-r--r--. 1 root root 26K Sep 6 16:22 /etc/passwd/targeted-modules/actions/m
d/lex/cond/pp
```

搜索字符串命令



➤ 搜索字符串命令: grep

❖ 原文: global search regular expression(RE) and print out the line

❖ 格式: grep [选项] 字符串 文件名

❖ 功能: 搜索文件中 含有指定字符串的行

❖ 选项:

-i 忽略大小写

-v 排除指定字符串

-l 列出包含匹配字符串的文件

❖ 说明: 被搜索的文件名可以是空格分隔的文件名列表或目录

```
[root@localhost ~]# grep "enable" ana*
auth --enableshadow --passalgo=sha512
firstboot --enable
[root@localhost ~]# grep -l "enable" ana*
anaconda-ks.cfg
```

文件搜索命令(续)



➤ find 命令与 grep 命令的区别

❖ find 命令:

❖ 在系统中搜索符合条件的文件名(完全匹配), 如果需要可以使用通配符匹配

❖ grep 命令:

❖ 在文件中搜索符合条件的字符串, 如果需要可使用正则表达式进行匹配, 正则表达式是包含匹配

帮助命令



➤ 帮助命令 man

❖ 原文: manul pages

❖ 格式: man 命令

❖ 功能: 获得某个命令的说明和使用方式的详细介绍

❖ 例:

man ls #查看 ls 的帮助

❖ 使用最小 CentOS 时, 很多命令的帮助需要用 yum (yellow dog updater, modified) 安装(yum install man)

```
[root@localhost ~]# whereis man
man: /usr/share/man
[root@localhost ~]# which man
/usr/bin/which: no man in (/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin)
```

帮助命令(续)



➤ 外部命令的选项帮助 --help

❖ 格式: 外部命令 --help

❖ 说明: 外部命令基本上都有一个参数 --help, 用它可以得到相应的帮助

❖ 注意: shell 内部命令不能用 man 找到帮助

❖ 例:

ls --help

❖ 可以用 | more 命令暂停屏幕显示, 屏幕翻页快捷键:

Shift-PageUp 向前翻看

Shift-PageDown 向后翻看

帮助命令(续)



➤ shell 内置命令帮助 help

❖ 格式: help shell 内置命令

❖ 功能: 显示命令的简要说明以及一些参数的使用以及说明

❖ 例:

whereis cd

❖ 功能: 确定是否是 shell 内置命令(内置命令找不到可执行文件)

help cd

❖ 功能: 获取内置命令帮助

❖ 说明:

❖ 外部命令有可执行文件, 能用 man 查看帮助(如 ls)

❖ shell 内置命令没有独立的可执行文件, 只能用 help 查看帮助(如 cd)

❖ man cd 打开的不是帮助, 同样 help ls 也会报错



帮助命令(续)

➤详细帮助命令 info

- ❖ 格式: info 命令
- ❖ 常用快捷键:
  - 回车 进入帮助页面(带有\* 号标记)
  - u 进入上层页面
  - n 进入下一个帮助小节
  - p 进入上一个帮助小节
  - q 退出 info

压缩和解压缩命令

➤常用压缩格式

- ❖ .zip
- ❖ .gz
- ❖ .bz2
- ❖ .tar.gz
- ❖ .tar.bz2

压缩和解压缩命令 (续)

➤.zip 格式压缩

- ❖ 格式1: zip 压缩文件名 源文件名
- ❖ 功能: 压缩文件, 不删除源文件, 一般用 .zip 作后缀
- ❖ 格式2: zip -r 压缩文件名 源目录
- ❖ 功能: 递归, 压缩目录

➤.zip 格式解压缩

- ❖ 格式: unzip 压缩文件
- ❖ 功能: 解压缩 zip 文件

➤说明: .zip 格式与 Windows 通用

压缩和解压缩命令 (续)

➤.gz 格式压缩

- ❖ 格式1: gzip 源文件
- ❖ 说明1: 压缩为 .gz 格式的压缩文件, 并删除源文件
- ❖ 格式2: gzip -c 源文件 > 压缩文件
- ❖ 说明2: 压缩为 .gz 格式并重命名, 源文件保留
- ❖ 格式3: gzip -r 目录
- ❖ 说明3: 递归, 将指定目录下的所有文件及子目录一并处理

➤.gz 格式解压缩

- ❖ 格式1: gunzip -d 压缩文件
- ❖ 说明1: 解压缩文件
- ❖ 格式2: gunzip 压缩文件
- ❖ 说明2: 解压缩文件
- ❖ 注意: 两种方法解压后默认都删除原文件

压缩和解压缩命令 (续)

➤.bz2 格式压缩

- ❖ 格式1: bzip2 源文件
- ❖ 说明: 压缩为 .bz2 格式, 不保留源文件
- ❖ 格式2: bzip2 -k 源文件
- ❖ 说明: 压缩之后保留源文件
- ❖ 注意: bzip2 命令不能压缩目录

➤.bz2 格式解压缩

- ❖ 格式1: bzip2 -d 压缩文件
- ❖ 说明1: 解压缩, 用 -k 保留压缩文件
- ❖ 格式2: bunzip2 压缩文件
- ❖ 说明2: 解压缩, 用 -k 保留压缩文件

压缩和解压缩命令 (续)

➤打包命令: tar

- ❖ 原文: tape archive(原意是将文件备份到磁带上)
- ❖ 打包格式: tar [选项] 包文件名 被打包的空格分隔文件列表
- ❖ 解包格式: tar [选项] 包文件名
- ❖ 说明:
  - ☞ 根据参数不同, 可以打包、解包和查看包中内容, 还可以是压缩的, 但 -f 选项是必须的(指定包文件名), 常用:
    - c 建立新的备份文件
    - x 从备份文件中还原文件
    - f <备份文件> 指定备份文件
    - v 显示指令执行过程
    - k 保留原有文件不覆盖
    - m 保留文件不被覆盖

关机与重启

关机命令 shutdown

- ❖ 格式: shutdown [选项] 时间
- ❖ 常用选项:
  - c 取消已经进行中的关机动作
  - h 关机后停机
  - r 重启
- ❖ 使用权限: 系统管理者
- ❖ 注意:
  - ☞ 服务器不要远程关机(如何开机?)
  - ☞ 与个人机不同, 服务器因可能有多人正在使用, 尽量不要重启, 强制重启会造成硬盘或者一些设备损坏

关机与重启 (续)

关机命令 shutdown (续)

- ❖ 例:

```
[root@localhost ~]# shutdown -r 05:30
```

  - ☞ 凌晨 5:30 重启, 但是执行后控制台就挂起了

```
[root@localhost ~]# shutdown -r 05:30 &
```

  - ☞ 后台重启

```
[root@localhost ~]# shutdown -c
```

  - ☞ 取消shutdown命令, 也可用 ctrl+c 取消

```
[root@localhost ~]# shutdown -h now
```

  - ☞ 立即关机

❖ shutdown 是比较安全的关机命令

关机与重启 (续)

其它关机命令

- ❖ 格式1: halt
- ❖ 格式2: poweroff
- ❖ 格式3: init 0
- ❖ 以上三条命令不太安全, 建议不要用

关机与重启 (续)

其它重启命令

- ❖ 格式1: reboot
- ❖ 格式2: init 6
- ❖ 说明:
  - ☞ reboot 相对安全一些
  - ☞ init 6 尽量不要用, 其后的数字表示系统运行级别
- 0 停机状态, 系统不能默认为0, 否则不能正常启动
- 1 单用户状态, root权限, 系统维护用, 禁止远程登录
- 2 多用户状态(没有NFS)
- 3 完全多用户状态(有NFS), 登录后进入控制台 命令行模式
- 4 未使用, 保留
- 5 X11控制台, 登录后进入 图形GUI模式
- 6 系统重启, 默认运行级别不能设为6, 否则不能正常启动

关机与重启 (续)

修改与查询系统运行级别

- ❖ 格式1: cat /etc/inittab
  - ☞ 功能: 查看系统启动时的默认级别, 并将类似以下的行中的数字修改

```
id:3:initdefault
```
- ❖ 格式2: runlevel
  - ☞ 功能: 查看系统当前运行级别

```
[root@localhost ~]# runlevel
N 3
```

  - ☞ 显示前一次的运行级别和现在的运行级别, 如果前次的运行级别为N, 那么说明前次没有运行级别(可能刚刚 power on)
  - ☞ 注意, 默认级别不能是0或者6, 否则永远开不了机

关机与重启 (续)

退出登录命令 logout

- ❖ 格式: logout
- ❖ 说明:
  - ☞ 执行该命令后, 系统将返回到登录状态



其它常用命令

挂载命令

登录用户查看和用户交互命令

其它常用命令（续）

查询与自动挂载 mount

- ❖ 格式1: mount
  - ☞ 功能: 查询系统中已经挂载的设备
- ❖ 格式2: mount -a
  - a 加载 /etc/fstab 文件中描述的所有文件系统
  - ☞ 功能: 将 /etc/fstab 文件中描述的所有文件系统自动挂载
  - ☞ 自动挂载文件 /etc/fstab 的格式:  
filesystem dir type options dump pass
    - ◆ options 挂载时使用的参数
    - ◆ dump 0/1: 忽略/备份, 用于dump工具进行备份
    - ◆ pass fsck检查顺序, 0:忽略, 1:最高优先权, 2:其他设备
  - ☞ 例: /dev/mapper/centos-swap swap swap defaults 0 0
  - ☞ 不建议自动挂载光驱、U盘等, 因为若忘记放入光盘, 则系统会起不来

其它常用命令（续）

挂载命令 mount

- ❖ 格式: mount [-t 文件系统] [-o 特殊选项] 设备文件名 挂载点
  - t 文件系统 文件系统类型用来指定挂载的类型, 可以是 ext3、ext4、iso9660 等文件系统
  - o 特殊选项 可以指定挂载的额外选项
- ❖ CentOS6 以上文件系统均是 ext4, 光盘文件系统是 ISO9660
- ❖ 挂载点即已经存在的空白目录

其它常用命令（续）

挂载的特殊选项(options选项)

| 参数            | 说明                                                   |
|---------------|------------------------------------------------------|
| atime/noatime | 更新访问时间/不更新访问时间。访问分区文件时, 是否更新文件的访问时间, 默认为更新           |
| async/sync    | 异步/同步, 默认为异步                                         |
| auto/noauto   | 自动/手动, mount -a命令执行时, 是否会自动安装/etc/fstab文件内容挂载, 默认为自动 |
| defaults      | 定义默认值, 相当于rw,suid,dev,exec,auto,nouser,async这七个选项    |
| exec/noexec   | 执行/不执行, 设定是否允许在文件系统中执行可执行文件, 默认是exec允许               |
| remount       | 重新挂载已经挂载的文件系统, 一般用于指定修改特殊权限                          |
| rw/ro         | 读写/只读, 文件系统挂载时, 是否具有读写权限, 默认是rw                      |
| suid/nosuid   | 具有/不具有SUID权限, 设定文件系统是否具有SUID和SGID的权限, 默认是具有          |
| user/nouser   | 允许/不允许普通用户挂载, 设定文件系统是否允许普通用户挂载, 默认是不允许, 只有root可以挂载分区 |
| usrquota      | 写入代表文件系统支持用户磁盘配额, 默认不支持                              |
| grpquota      | 写入代表文件系统支持组磁盘配额, 默认不支持                               |

其它常用命令（续）

挂载举例（一定记得马上改回来！）

```
[root@localhost ~]# vim /boot/hello.sh # 创建脚本文件
[root@localhost ~]# chmod 755 /boot/hello.sh # 赋予执行权限
[root@localhost ~]# /boot/hello.sh # 执行脚本文件
[root@localhost ~]# mount -o remount, noexec /boot
重新挂载 /boot 分区, 并使用 noexec (不可执行)权限
[root@localhost ~]# /boot/hello.sh # 再次执行, 提示权限不够
[root@localhost ~]# mount -o remount, exec /boot
把权限修改回来, 否则会影响系统启动
```

```
[root@localhost ~]# vim /boot/hello.sh
[root@localhost ~]# /boot/hello.sh
[root@localhost ~]# /boot/hello.sh
hello world!
[root@localhost ~]# mount -o remount, noexec /boot
[root@localhost ~]# /boot/hello.sh
-bash: /boot/hello.sh: Permission denied
[root@localhost ~]# mount -o remount, exec /boot
[root@localhost ~]# /boot/hello.sh
hello world!
```

hello.sh的内容:  
#!/bin/bash  
echo "hello world!"

其它常用命令（续）

挂载光盘

```
[root@localhost ~]# mkdir /mnt/cdrom
#建立挂载点

[root@localhost ~]# mount -t iso9660 /dev/sr0 /mnt/cdrom
#挂载光盘

#简写:
[root@localhost ~]# mount /dev/cdrom /mnt/cdrom
```

- 光盘挂载后会报错, 这说明挂载正常了, 因光盘不能写
- 进入 /mnt/cdrom 即可读取光盘文件

其它常用命令（续）



➤卸载命令 umount

❖ 格式：umount 设备文件名或挂载点

❖ 举例：

[root@localhost ~]# umount /mnt/cdrom

❖ 说明：

- ❖ 卸载时注意退出挂在目录，不要在光盘目录里卸载
- ❖ 卸载命令是必须的
- ❖ 虚拟机如果不卸载，直接强制换光盘，读的数据是错乱的

其它常用命令（续）



➤挂载 U 盘

[root@localhost ~]# fdisk -l

❖ 查看 U 盘设备文件名(查看所有分区)

[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/

❖ 注意：

- ❖ Linux 默认不支持 NTFS 文件系统
- ❖ 系统把 U 盘当硬盘看待，插入 U 盘后，看它的设备名，一般是 sdb1，vfat 就是 FAT32

其它常用命令（续）



➤查看登录用户信息 w

❖ 格式：w 用户名

❖ 命令输出信息含义：

- USER 登录的用户名
- TTY 登录终端
- FROM 登录的 IP 地址
- LOGIN@ 登录时间
- IDLE 用户闲置时间
- JCPU 与该终端连接的所有进程占用的时间。这个时间里并不包括过去的后台作业时间，但却包括当前正在运行的后台作业所占用的时间
- PCPU 当前进程所占用的时间
- WHAT 当前正在运行的命令

w 命令结果：

- 第1行：时间、到现在系统运行时间、当前系统中用户数、系统在一分钟、五分钟、十五分钟前的负载情况；
- 第2行是信息说明行；
- 第3行开始才是实际数据；tty1 代表本机登录，pts/0 代表远程登录；

```
[root@localhost ~]# w root
 23:28:57 up 7:05, 1 user, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root tty1 - 23:18 1.00s 0.02s 0.01s w root
```

其它常用命令（续）



➤查看登录用户信息 who

❖ 格式：who [选项] [用户名]

❖ 例：显示当前登入系统的用户

```
[root@localhost ~]# who
root tty1 2018-09-06 23:18
```

❖ 例：显示标题栏

```
[root@localhost ~]# who -H
NAME LINE TIME COMMENT
root tty1 2018-09-06 23:18
```

❖ 例：显示用户登录来源

```
[root@localhost ~]# who -lH
NAME LINE TIME IDLE PID COMMENT
LOGIN tty2 2018-09-06 16:23 1101 id=2
LOGIN tty3 2018-09-06 16:23 1105 id=3
LOGIN tty4 2018-09-06 16:23 1108 id=4
LOGIN tty5 2018-09-06 16:23 1111 id=5
LOGIN tty6 2018-09-06 16:23 1113 id=6
```