

• 目录

3、常见子目录的作用

根目录	作用
/root	系统管理员 root 的宿主目录
/home	普通用户的宿主目录
/boot	系统内核启动文件
/dev	Device, 设备文件
/etc	配置文件
/bin	Binary (二进制), 所有用户可执行的命令
/sbin	System binary, 管理员可执行的命令
/usr	Unix software resource, 应用程序
/var	Variability (可变的), 日志文件等

/proc

存放虚拟文件系统目录。保存进程在内存中的信息。

/mnt

临时挂载点，用于挂载其它文件系统。

/tmp

临时文件，所有用户可访问。

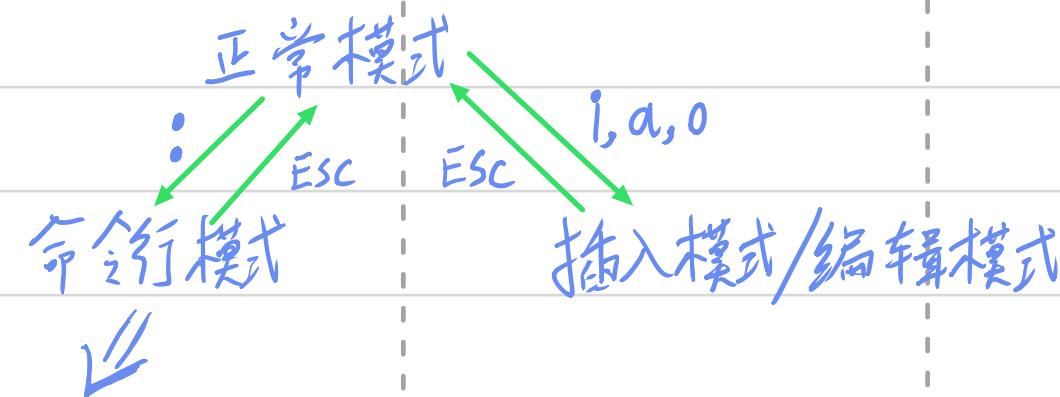
/media

可移除媒体如(CD)的挂载点

VIM 编辑器：

• 三种模式： 正常模式，编辑模式，命令行模式。

• 模式切换： 默认进入正常模式



• 末行模式命令：

命令行模式命令

:w	保存但是没有退出
:wq	保存退出
:x	保存退出，等同于:wq
:q	退出(不保存)
:q!	强制退出(不保存)

:set nu	设置行号
:set nonu	取消行号

• 正常模式命令：

操作类型	操作键	功能
删除	x或Del	删除光标处的单个字符
	dw	删除至一个单词的末尾
	dd	删除当前光标所在行
	#dd	删除从光标所在行开始的#行内容
	d^	删除当前光标之前到行首的所有字符(不含光标)
	d\$	从当前光标删除到行尾(包含光标)
复制	yy	复制当前行整行的内容到剪贴板
	#yy	复制从光标所在行开始的#行内容
粘贴	p	将最后删除的内容置入光标之后，粘贴到光标位置处之前
	ce+字符	输入r+字符，替换所在位置字符
替换	ce+字符	从光标处更改直到单词末尾
	c\$+字符	从光标处更改直到行尾
	R+字符	连续替换多个字符

先按 dd 再按 p

剪切光标所在行

dG 删除当前行之后的内容

• 光标移动：

1.1、光标移动

操作类型	操作键	功能
方向移动	H J K L或上下左右键	上下左右
翻页	Page Down或Ctrl+F	下翻页
	Page up或Ctrl+B	上翻页
行内快速跳转	HOME键或^、数字0	跳至行首
	END键或\$	跳至行尾
行间快速跳转	1G或gg	跳转文件的首行
	G	跳转文件的尾行

数字 + G

跳转到指定行

• 内容查找：

1.3、文件内容查找 (区别大小写)

操作键	功能
/word	从上而下在文件中查找字符串“word”
?word	(逆向) 从下而上在文件中查找字符串“word”
n	向下查找匹配字符串
N	向上查找匹配字符串

下一个
上一个

1.4、撤销编辑及保存退出

操作键	功能
u	撤销最后执行的命令,多次输入,多次撤销
U	撤销对整行的命令
ZZ	保存当前的文件内容并退出vi编辑器

文件目录类：

pwd	显示当前工作目录的绝对路径。
ls [选项] 目录或文件	显示当前目录下的文件和目录。
-a	显示所有文件，包括隐藏文件。
-l	以列表的方式显示详细信息。
-lh	将文件大小以更合适的单位显示。
-d	查看目录。
cd [选项] 目录路径	切换到指定目录。
cd ~ 或 cd	回到家目录。
cd ..	切换到上级目录。
cd ../../..	切换到上两级目录。
cd /	切换到根目录。
cd -	回到上个工作目录。

mkdir [选项] 要创建的目录 创建一个空目录。

-p 创建多级目录。

 mkdir -p 123/456 在当前路径下
创建123目录，并在123目录内创建456目录。

rmdir 要删除的空目录 删除空目录。

rm -rf 要删除的目录 删除非空目录。

touch 文件名称 创建一个新文件。

• 可一次创建多个文件 例 touch 01.txt 02.txt 03.txt

wc 文件名 统计文件的字数、行数等信息。

cp [选项] 原文件路径 目标文件路径 将文件复制到目标路径。

-r 递归复制，可用于复制目录。

-f 覆盖文件而不提示。

bin/cp 事实上 cp命令是一个别名，默认带选项 -i

bin/cp 是命令本身，而不是别名。

输出重定向

>

输出重定向，会覆盖原有内容。

>>

追加输出重定向，向文件内追加内容，不覆盖原有内容。

2>

错误输出重定向，当命令报错时，将错误信息，重定向。

2>>

错误输出追加重定向。

输入重定向

<< END

输入重定向，将键盘输入的字符 重定向到文件。

END为结束标志，要结束输入时，输入END。将结束重定向
END可替换为其它字符。

查看文件类

• cat	cat [选项] 要查看的文件	查看文件但不能修改
	-n	显示行号
	-b	与-n类似，只不过空自行不编号。
	-s	有两行以上空行就替换为一行
• more	more [选项] 要查看的文件	以页的方式显示。 空格：下一页。 b：上一页。 q：退出 :f 输出当前文件名和行号 √：调用Vi编辑器
	-num	一次显示 num 行
	-f	计算行数时，以实际的行数，而非自动换行过的行数。
	-P	不以滚动的方式显示，而是先清除再显示。
	-S	有两行以上空行就替换为一行
	+num	从第 num 行开始显示
	+/Pattern	搜索该字符串(Pattern)，并从该行开始显示。
• less	less [选项] 要查看的文件	与 more 类似，less 并不是一次将文件全部加载，因此对大文件有较高的效率。
	-b <缓冲区大小>	设置缓冲区的大小
	-m	显示百分比
	-N	显示行号
	-o <文件名>	将 less 输出的内容在指定文件中保存起来。
	/字符串	向下搜索“字符串”
	? 字符串	向上搜索“字符串”
	n	重复前一个搜索
	N	反向重复前一个搜索
• head	head [选项] 文件名	显示文件的开头内容，默认显示前10行
	-n <num>	显示文件的前 num 行
• tail	tail [选项] 文件名	显示文件的尾部内容，默认后10行
	-n <num>	显示文件的后 num 行
	-f	循环读取，只要文件更新就显示最新的内容。

• tac

(cat) 的反序，以行的形势倒序输出。

• nl

显示文件内容，并添加行号。可自定义行号和显示效果，
包括位数和自动补全 0

find

find [查找范围] [查找范围表达式]

-name	按名称查找	
-size	按大小查找	+IM [k,M,G]
-user	按属主查找	d f b c
-type	按类型查找	
-group	按用户组查找	
-atime	访问时间(天)	+30 前30天之前的 30 前30天当天的 -30 前30天到现在
-mtime	更改时间(天)	
-ctime	改动时间(天)	
-m min	更改时间(分钟)	-amin -cmin
-empty	空的文件	
-maxdepth	最大深度 即为只搜索当前目录,不搜索下级目录	
-regex	按正则匹配	
与或	-a and	-o or

文件属性

f 一般文件

b 块设备文件

l 软链接文件

d 目录

c 字符设备文件

s socket 通信套接文件.

取反： find ! -name "ol.txt"

find -! -name "ol.txt"

find -not -name "ol.txt"

grep

过滤,

• grep [选项] [模式] [文件]

-v	对结果取反	grep -i "^\i" 01.txt
-i	忽略大小写	单引号：仅作为字符串，不解析
-B<n>	显示匹配的那一行，和之前的前n行。	双引号：会解析引号内的特殊字符。
-A<n>	显示匹配的那一行，和之后的n行。	不加引号：等价于双引号
-b	显示匹配的那一行，和这一行的首个匹配到的字符，是文章中的第几个字符	
-c	显示匹配到的有几个行。	
-C	显示匹配的那一行，和之前之后各n行。	
-d<>	查询目录用。	
-e<范本样式>	grep -e "ABCD" -e "1234" 1.txt 可多次结合	
-E	将样式为延伸的正则表达式来使用。	egrep \Leftrightarrow grep -E
-f<规则文件>	规则文件中，含有一个或多个规则样式，一行一个。	默认支持扩展正则
-n	显示匹配到的行号	
-o	只显示匹配到的内容	
-q	不显示任何信息	
-m<n>	指定显示最大行数	
-w	精确匹配内容	

4. 规则表达式:

grep的规则表达式:

- ^ #锚定行的开始 如: '^grep' 匹配所有以grep开头的行。
- \$ #锚定行的结束 如: 'grep\$' 匹配所有以grep结尾的行。
- . #匹配一个非换行符的字符 如: 'gr.p' 匹配gr后接一个任意字符，然后是p。
- * #匹配零个或多个先前字符 如: '*grep' 匹配所有一个或多个空格后紧跟grep的行。
- .* #一起用代表任意字符。
- [] #匹配一个指定范围内的字符，如'[Gg]rep' 匹配Grep和grep。
- [^] #匹配一个不在指定范围内的字符，如: '[^A-FH-Z]rep' 匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。
- (..) #标记匹配字符，如'(love)'，love被标记为1。
- < #锚定单词的开始，如:<grep' 匹配包含以grep开头的单词的行。
- > #锚定单词的结束，如'grep>' 匹配包含以grep结尾的单词的行。
- x{m} #重复字符x，m次，如: '0{5}' 匹配包含5个o的行。
- x{m,} #重复字符x,至少m次，如: 'o{5,}' 匹配至少有5个o的行。
- x{m,n} #重复字符x，至少m次，不多于n次，如: 'o{5,10}' 匹配5--10个o的行。
- w #匹配文字和数字字符，也就是[A-Za-z0-9]，如: 'Gw*p' 匹配以G后跟零个或多个文字或数字字符，然后是p。
- W #w的反置形式，匹配一个或多个非单词字符，如点号句号等。
- #单词锁定符，如: '□grep□' 只匹配grep。

Sed

文本流编辑器 (行处理编辑器)

主要用来自动编辑文件

命令 | sed [选项] ‘匹配条件和操作指令’

sed [选项] ‘匹配条件和操作指令’ 输入文件



-n 仅显示处理后的结果.

★ 选项

-r 支持扩展正则



-e 可在同一行中执行多条命令, 以分号隔开 `sed -e 's/wang/w/g;s/xu/x/g' user.txt`

-f <script文件>

指定需要执行的 Sed 文件. 文件中需指前写入命令



-i [备份后缀名] 修改源文件, 若填写的备份后缀名, 则会先备份, 后修改



P 打印当前匹配的数据行

L 打印当前匹配的数据行(显示控制字符, 如回车符等)



= 打印当前读取的数据所在行数.



a text 在匹配的数据行后追加文本 text



i text 在匹配的数据前插入文本 text



d 删 除 匹 配 的 数据 行 (整行)



c text 将匹配的数据行整行 替换为 text



s /正则/ /text/ 将正则匹配到的内容用 text 替换

`s/正则/ /text/ g`

g 代表贪婪模式

r 文件名 将文件中的内容 追加到匹配行后面

w 文件名 将匹配到的内容 写入文件中.



number 直接匹配第 number 行

first ~ step 从 first 行开始, 步长为 step

数据定位

\$ 匹配最后一行



/正则/ 使用正则表达式 匹配行



n₁, n₂ 从 n₁ 行号到 n₂ 行号的所有行



n₁, +n 从 n₁ 行开始及后面的 n 行

awk

逐行读取文本，默认以“空格”或“tab”为分隔符，进行分割
是一门语言，类似于C语言。

单引号

awk 选项 ‘模式或条件 {操作}’ 文件1 文件2 ...

awk -f 脚本文件 文件1 文件2

awk 内建变量

FS 列分隔符。指定每行文本的字段分隔符，默认为空格或制表符，与“-F”作用相同

NF 当前处理的行的字段个数。

NR 当前处理的行的行号。

\$0 当前处理的行的整行内容。

\$n 当前处理行的第n个字段(第n列)

FILENAME 被处理的文件名。

RS 行分隔符

OFS 显示每一列时，列与列之间的分隔符(默认为空格)

-F <fs> 指定分隔符，fs可以是一个字符串或一个正则

-V var=Value 赋值一个用户自定义变量。

~

代表包含

常用

/哪里开始/, /哪里结束/ 中间为正则

NR=1, NR=5 从第1行开始，到第5行结束。类似于 sed 中的 sed -n '1,5p'

BEGIN { }

在读取文件之前要作的操作

END { }

读取文件结束后的操作。

打包解压类

tar [选项] 打包后的文件名 要打包的文件 打包指令，打包后的格式为 .tar.gz 文件

tar命令选项:

选项	说明
-c	创建打包文件
-v	显示打包或者解包的详细信息
-f	指定文件名称, 必须放到所有选项后面
-z	压缩或解压缩(.gz)
-j	压缩或解压缩(.bz2)
-x	解包
-C	解压缩到指定目录

-ZCVf 打包并压缩

-ZXvf 解压

-Xvf 解包

-cvf 打包(不压缩)

-tf 不解压查看含有什么文件

tar -ZXvf xxx.tar.gz -C 目录 将文件解压到指定目录。

tar -Zcf 打包后的文件名及路径 -C /test/ hahaha 打包 hahaha 文件夹及内容, 不打包 test 目录

/test/hahaha/. 打包 hahaha 文件夹下的内容, 不打包 hahaha 目录及

空格前的层级关系将不被

打包, 仅做路径定位用.



ZIP 固定文件 源文件

-r 固定文件名 目录 对目录进行压缩

unzip xxx.zip 解压到当前路径

unzip xxx.zip -d 目录 将文件解压到指定目录

-l 显示压缩文件内所包含的文件.

gzip / gunzip 压缩 / 解压

此命令会将原文件覆盖。

gzip -d 解压

扩展名 .gz

注意：

- gzip 命令只能用来压缩文件，不能压缩目录，即便指定了目录，也只能压缩目录内的所有文件。
- 压缩后会删除源文件。除 -C
- 压缩后文件的后缀格式是 .gz 。
- 该命令还支持解压缩。
- 该命令不需要指定压缩包名。

-n n 为 1-9 的数字，指定压缩级别 9 为最大。默认为 6

gzip -C 1.txt > temp.txt.gz 将 1.txt 的内容追加到 temp.txt.gz 中。1.txt 文件保留

gunzip -C temp.txt.gz > 路径 将文件解压到指定路径，同时可重命名原文件
-r 递归处理，用于目录内的所有文件。保留

zcat 文件名 在不解压文件的情况下，查看文件 仅用于 .zip 文件

-l 显示文件的压缩信息，当前大小 压缩前大小 压缩比率 文件名

bzip2 与 gzip 用法相同。

默认压缩级别为 9 (最大)

不解压查看文件 bzcat

bzip2 -d 1.txt 解压

xz 与 gzip 用法相同。 后缀为 .xz

不解压查看文件 xzcat

软件管理

• rpm

-q : 查询

-a : 全部

-e : 卸载

rpm -e 软件包名 --nodeps 不考虑依赖关系, 强制卸载

-ql 软件的安装路径, 文件列表

-qf 文件或目录 查询文件属于哪个软件包

• 安装

安装之前率先获取安装包

• 官网下载 • 从光盘(镜像)读取

rpm -ivh

软件包完整名称

-- force

强制安装.

-i

安装

-v

显示进度条

-h

表示以“#”形式显示进度条

扩展: 挂载与解挂

解挂

umount 当前挂载路径.

挂载

mount 设备原始地址 要挂载的路径

→ 为 /dev/设备名 设备名可用 lsblk 查看

lsblk

• yum

yum [选项] [子命令] [软件包名]

常用选项

-y

表示直接执行, 无须确认

子命令

install

安装软件

update

更新软件

remove

卸载软件

list <软件包名>

查询 列出可用软件包 (云端)

info <软件包名>

显示当前可用软件包的属性

search <关键字>

模糊搜索 可用软件

配置文件

/etc/yum.repos.d 因录下以 repo 结尾的文件.

更换配置文件后

yum clean all

yum repolist

用来从指定的URL下载文件

• wget

• 本地 yum 源配置

1. 备份原 yum 源

1.2 备份原 yum 源

```
cd /etc/yum.repos.d/  
mkdir bak  
mv *.repo bak
```

2. 创建本地 yum 源

```
touch mylocal.repo
```

3. 配置本地 yum 源

```
cat >>mylocal.repo <<END  
[local]  
name=local-yum-centos7  
enable=1  
gpgcheck=0  
baseurl=file:///media  
END
```

内容不限
使能
gpg校验使能
路径
协议格式

4. 清除 yum 缓存

```
yum -y clean all
```

5. 创建 yum 缓存

```
yum makecache
```

• 编译安装

要有依赖程序及算法环境 gcc , gcc-c++ make

常用的系统指令用 yum 安装

服务器中的节点应用程序用编译安装

流程

1. 解决基础依赖环境

指定安装目录

2. 准备源码包，进行校验，校验无误后解压缩 /usr/src



3. 进入解压后的目录，找到预配置文件进行预配置

./configure --prefix=/usr/local/apache

→ 定义安装路径，及安装模块

还可以增加第三方模块，禁用不需要的模块

4. 进行源码编译，使用 make 命令。

5. 进行安装，使用 make install

• 补丁文件

生成补丁文件 diff -Naur /root/test/old.txt /root/test/new.txt > txt.patch

比较两个文件的不同，并生成补丁文件

打入补丁

Patch -p n < 补丁文件

绝对路径

n为数字 修订路径用

进程管理类

• 查看进程

ps

只显示当前用户和与控制台相关的进程。

a	列出带有终端的所有用户的进程
x	列出当前用户的所有进程，包括没有终端的进程
u	面向用户友好的显示风格
-e	列出所有进程
-u	列出某个用户关联的所有进程
-f	显示完整格式的进程列表

常用： ps aux

查看系统中所有进程

-aux 也可用，但不推荐

ps -ef

可以查看父子进程之间的关系

aux 显示的内容

USER: 该进程是由哪个用户产生的

PID: 进程的 ID 号

%CPU: 该进程占用 CPU 资源的百分比，占用越高，进程越耗费资源；

%MEM: 该进程占用物理内存的百分比，占用越高，进程越耗费资源；

VSZ: 该进程占用虚拟内存的大小，单位 KB；

RSS: 该进程占用实际物理内存的大小，单位 KB；

TTY: 该进程是在哪个终端中运行的。对于 CentOS 来说，tty1 是图形化终端，

tty2-tty6 是本地的字符界面终端。pts/0-255 代表虚拟终端。?表示无终端

STAT: 进程状态。常见的状态有：R: 运行状态、S: 睡眠状态、T: 暂停状态、

Z: 僵尸状态、s: 包含子进程、l: 多线程、+: 前台显示

START: 该进程的启动时间

TIME: 该进程占用 CPU 的运算时间，注意不是系统时间

COMMAND: 产生此进程的命令名

UID: 用户 ID

PID: 进程 ID

PPID: 父进程 ID

C: CPU 用于计算执行优先级的因子。数值越大，表明进程是 CPU 密集型运算，

执行优先级会降低；数值越小，表明进程是 I/O 密集型运算，执行优先级会提高

STIME: 进程启动的时间

TTY: 完整的终端名称

TIME: CPU 时间

CMD: 启动进程所用的命令和参数

killall 进程名称

通过名称杀死进程，所有同名进程

• kill [选项] 进程号 杀死进程

-9

强迫进程立即停止。

-l

列出所有信息名称

常用信号

1 重新加载进程

9 杀死进程

15 正常停止一个进程

• pstrace

查看进程树。

-P

显示进程的 PID

-U

显示进程的所属用户

• top

实时监控显示系统进程状态。

-d <秒数> 指定 Top 命令每隔几秒刷新，默认3秒

-i 不显示闲置或僵死进程

-P PID 指定监控进程 ID 监控某个进程

• netstat

显示网络状态

- a 显示所有正在监听(listen)和未监听的套接字(socket)。即端口
- n 抑制显示别名，以数字显示
- l 仅显示监听状态的
- p 显示哪个进程在调用

常用 <

- anp
- nlp
- r 查看路由表
- t 显示TCP的连接
- u 显示UDP的连接

• 域名映射

/etc/hosts 文件

- 格式

192.168.100.129 abcd

Ping abcd <=> Ping 192.168.100.129

• 不重启识别新加盘:

触发系统重新扫描文件:
echo "----" > /sys/class/scsi_host/host0/scan
echo "----" > /sys/class/scsi_host/host1/scan
echo "----" > /sys/class/scsi_host/host2/scan

• 查看网卡规格

ethtool 网卡

• 查看路由表

route

-n

查看路由表

• 跟踪数据包

traceroute -n 网址

• 测试DNS域名解析

nslookup 网址

服务管理：

一、命令格式： `Systemctl [command] [unit]`

- `Command`：
command主要有

- `start`: 立刻启动后面接的unit
- `stop`: 立刻关闭后面接的unit
- `restart`: 立刻关闭后启动后面接的unit，亦即执行stop再start的意思
- `reload`: 不关闭后面接的unit的情况下，重载配置文件，让设定生效
- `enable`: 设定下次开机时，后面接的unit会被启动
- `disable`: 设定下次开机时，后面接的unit 不会被启动
- `status`: 目前后面接的这个unit 的状态，会列出是否正在执行、是否开机启动等信息。
- `is-active`: 目前有没有正在运行中

- `is-enable`: 开机时有没有预设要启用这个unit
- `list-units`: 依据unit列出目前有启动的unit。若加上--all才会列出没启动的。（等价于无参数）
- `list-unit-files`: 列出所有以安装unit以及他们的开机启动状态 (enabled、disabled、static、mask)
- `--type=TYPE`: 就是unit type，主要有service, socket, target等
- `get-default`: 取得目前的 target
- `set-default`: 设定后面接的 target 成为默认的操作模式
- `isolate`: 切换到后面接的模式
- `list-dependencies`: 列出unit之间依赖性
- `list-sockets`: 查看监听socket的unit

- `Unit`：主要是服务的名称

- `ntsysv`

DOS 界面

用户管理

• 添加用户

useradd

- c <备注> 添加备注文字
- d <目录> 指定家目录
- e <有效期限> 指定账号的有效期限
- g <群组> 指定用户所属的群组
- G <群组> 指定用户所属的附加群组
- M 不自动建立用户的家目录
- n 不建立以用户名为名的群组
- s <shell> 指定用户登录后所使用的shell
- u <uid> 指定用户ID
- r 建立系统帐号

帐户信息存放在 /etc/passwd 中

登录的方式 < 西行 /bin/bash 可登录并后台运行
/sbin/nologin 不可登录.

登录方式

用户名 密码哈希值

0 : X : 0 : 0 : 0 : 0
uid gid 备注 家目录路径

• 创建组

groupadd [选项] 组名

- d 指定新建组ID

• 删除用户

userdel

- r 同时删除用户的家目录

若组中有用户，则先删除用户。

• 修改用户信息

usermod

- c <备注> 修改用户备注
- d <目录> 修改用户家目录
- g <群组> 修改用户所属的群组
- G <群组> 修改用户的附加群组
- l <帐号名称> 修改用户的帐号名称
- L 锁定用户密码，使密码无效。
- U 解除密码锁定。
- u <uid> 修改用户ID

密码文件存放在 /etc/shadow 中，

只有 root 有权限查看。内容

用户名：加密密码：最后一次修改时间：最小修改时间间隔：

密码有效期：密码需要变更前的警告天数：密码过期后的宽限

时间：账号失效时间：保留字段

• 修改密码

passwd

- d 删除密码
- f 强迫用户下次登录时必须修改密码
- S 显示密码信息。
- L 停止帐号使用
- u 启用被停止的帐号

- w 指令要到期, 提前警告的天数。
- x 指定指令最长存活期
- i 密码过期后的宽限天数

● 修改组成员

gpasswd [选项] 组名

- a 添加用户到组
- d 从组删除用户
- A 指定管理员
- M 指定组成员
- r 删除密码

组配置存放在 /etc/group 文件

内容为： 组名:密码:gid : 该组用户列表

● 查询当前登录情况 id, groups, finger, w, whoami, who

权限管理

· 变更权限

`chmod`

`-R`

以递归的方式处理

表示字符 ^①	权限名称 ^②	对文件的含义 ^③	对目录的含义 ^④
r (4)	读 ^⑤	读取文件内容 ^⑥	检查目录内容 ^⑦
w (2)	写 ^⑧	修改文件内容 ^⑨	改变目录内容 ^⑩
x (1)	执行 ^⑪	执行文件 ^⑫	进入目录 ^⑬

数字方式

`chmod 777 文件名`

三位分别代表 属性权限 组权限 其它权限

$$r=4, w=2, x=1$$

符号模式

`chmod ug+w, o-w 文件名`

多个群组间用 , 隔开

who	用户类型	说明	
u	user	文件所有者	+ 为指定的用户类型增加权限
g	group	文件所有者所在组	- 去除指定用户类型的权限
o	others	所有其他用户	= 设置指定用户权限的设置, 即将用户类型的所有权限重新设置
a	all	所有用户, 相当于 ugo	

· 权限掩码

`umask`

umask 为新建文件或目录的权限掩码

$$\text{真实权限} = 777 - \text{权限掩码}$$

umask 保存在 /etc/profile

与 /etc/bashrc 中

`umask 222`

临时设置权限掩码。

文件的 x 权限只能用户手动赋值

不能用 umask 赋值。

即, umask 对目录的最大权限为 777

对文件的最大权限为 666

· 文件归属

只有 root 可用

`chown 用户名:组名 文件名`

将文件归属于某个用户, 某个组

组内成员可用 →

`chgrp [选项] 群组 文件名`

将文件归属于某个组

`-R` 递归处理

磁盘管理

· 接口

{ IDE
SATA
SCSI

并口，已淘汰，目前主要用于光驱
纠错能力强
支持热插拔

hd
sd
sd

文件系统 { ext4
xfs

Centos 6 的默认文件系统
日志型文件系统。centos7
将 xfs+LVM 作为默认文件系统

swap	swap 是 Linux 中用于交换分区的文件系统（类似于 Windows 中的虚拟内存），当内存不够用时，使用交换分区暂时替代内存。一般大小为内存的 2 倍，但是不要超过 2GB。它是 Linux 的必需分区
NFS	NFS 是网络文件系统（Network File System）的缩写，是用来实现不同主机之间文件共享的一种网络服务，本地主机可以通过挂载的方式使用远程共享的资源
iso9660	光盘的标准文件系统。Linux 要想使用光盘，必须支持 iso9660 文件系统
fat	就是 Windows 下的 fat16 文件系统，在 Linux 中识别为 fat
vfat	就是 Windows 下的 fat32 文件系统，在 Linux 中识别为 vfat。支持最大 32GB 的分区和最大 4GB 的文件
NTFS	就是 Windows 下的 NTFS 文件系统，不过 Linux 默认是不能识别 NTFS 文件系统的，如果需要识别，则需要重新编译内核才能支持。它比 fat32 文件系统更加安全，速度更快，支持最大 2TB 的分区和最大 64GB 的文件
ufs	Sun 公司的操作系统 Solaris 和 SunOS 所采用的文件系统
proc	Linux 中基于内存的虚拟文件系统，用来管理内存存储目录/proc
sysfs	和 proc 一样，也是基于内存的虚拟文件系统，用来管理内存存储目录/sysfs
tmpfs	也是一种基于内存的虚拟文件系统，不过也可以使用 swap 交换分区

· df

[选项] [文件]

文件系统磁盘使用情况

-a

显示特殊文件系统（显示所有文件系统）

-h

换算为合适单位

-T

显示文件系统类型列

· du

[选项] [目录或文件名]

-a

显示每个文件的磁盘占用。默认只显示子目录的磁盘占用量。

-h

换算为合适单位

-s

只显示总占用

不重启识别新加盘

du 与 df 的区别：du 是用于统计文件大小的，统计的文件大小是准确的；df 是用于统计空间大小的，统计的剩余空闲是准确的

du 与 df 不同时，可重启解决。原因：df 统计的大小包括了垃圾文件和未释放的文件。
du 统计的只有文件。

· 分区工具 fdisk

a	设置可引导标记
b	编辑 bsd 磁盘标签
c	设置 DOS 操作系统兼容标记
d	删除一个分区
l	显示已知的文件系统类型。82 为 Linux swap 分区，83 为 Linux 分区
m	显示帮助菜单
n	新建分区
o	建立空白 DOS 分区表
p	显示分区列表
q	不保存退出
s	新建空白 SUN 磁盘标签
t	改变一个分区的系统 ID
u	改变显示记录单位
v	验证分区表
w	保存退出
x	附加功能（仅专家）

格式化分区

mkfs.xfs /dev/sdb1

● 分区工具 Parted

- 2、格式: `parted [选项] [分区设备]`
- 3、交互式状态下的常用命令:
 - ?/help/--help: 帮助
 - quit/q: 保存退出
 - print/p: 打印
 - mklabel: 改变硬盘的文件类型
 - mkpart: 分区
 - rm: 删除某分区, 后面跟分区号, 例如: `rm 3`

rescue 分区恢复

● mkfs

`mkfs [选项] 设备号`

用于在特定的分区上建立 Linux 文件系统。(格式化)

`-t ext4`

指定要格式化为什么类型的文件系统

`-N 100000`

指定此分区的 inode 个数。查看 inode 数可用 `inode -i`

`-i 128`

每多少个字节分配一个 inode

XFS 文件系统不能用 -N 与 -i

可用

`mkfs.xfs`

的形式指定文件系统类型

指定 inode 个数。

`-b 区块容量`

设置区块的容量大小。Linux 最大为 4K

`-f`

强制格式化。

● mkswap

`mkswap 设备号`

将分区格式化为交换分区。格式化后需用 `swapon` 开启使用

`swapoff` 关闭

`swapon -s` 查看交换分区

● mount

`mount 设备号 挂载点`

挂载设备

`-a`

将 `/etc/fstab` 中的记录全部挂载

`-o ro`

以只读模式挂载

`-o rw`

以可读写模式挂载

`-o loop`

将一个文件作为硬盘挂载(例如镜像)

● blkid

查看设备 UUID 与 类型

LVM 逻辑卷管理

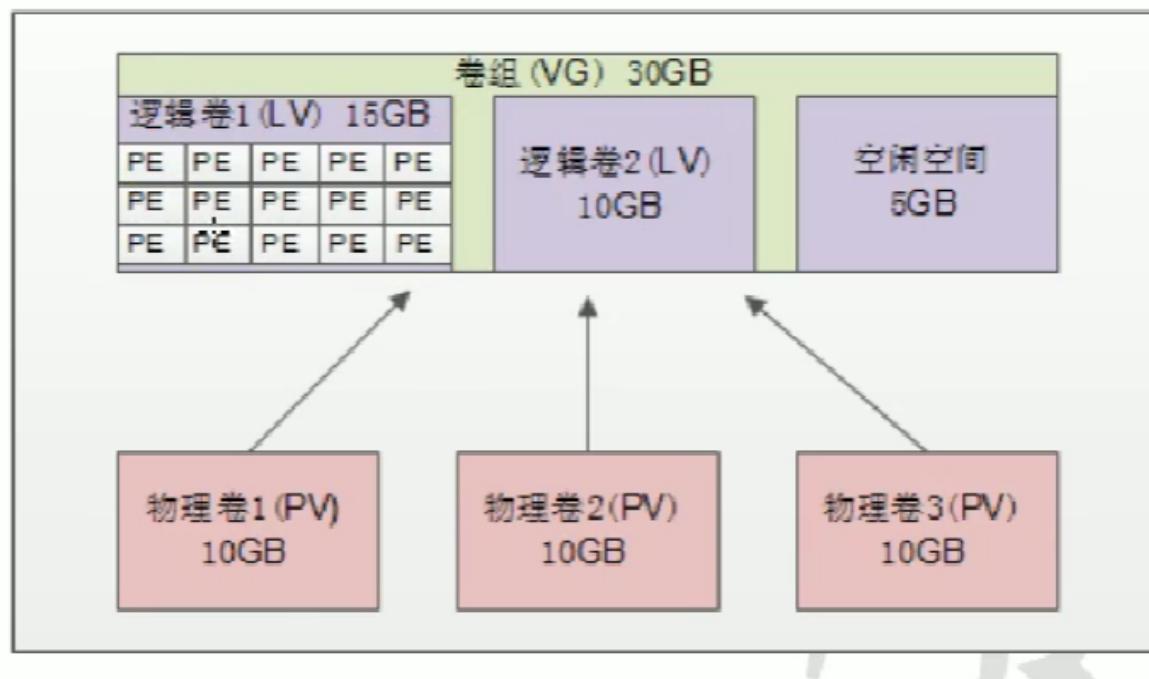


图 10-1 LVM 示意图

1. LVM 解决了 Linux 下分区大小不易调整的问题。
2. 卷组的空间可通过增加物理卷增大。
3. 逻辑卷的大小可在卷组内任意增大。
4. LVM 类似于虚拟机中的资源池化。

LVM 概念

1. 真正的物理硬盘或分区就是物理卷(PV)
2. 卷组(VG): 将多个物理卷合起来就组成了卷组，组成同一卷组的物理卷可以是一个硬盘的不同分区，也可以是不同硬盘上的不同分区。
3. 逻辑卷(LV): 卷组是一个逻辑硬盘，硬盘必须分区后才能使用。这个分区称为逻辑卷。逻辑卷可以格式化和写入数据可以把逻辑卷想象成分区。
4. 物理扩展(PE): PE是保存数据的最小单元，PE的大小可以配置默认是4MB。PE类似于分区时 直柱的概念，是划分分区大小的最小单元。

LVM 步骤

2. 建立 LVM 的步骤:
 - ◆ 首先需要把物理硬盘分成分区，当然也可以是整块物理硬盘。
 - ◆ 然后把物理分区建立成为物理卷 (PV)，也可以直接把整块硬盘都建立为物理卷。
 - ◆ 接下来把物理卷整合成为卷组 (VG)。卷组就已经可以动态的调整大小了，可以把物理分区加入卷组，也可以把物理分区从卷组中删除。
 - ◆ 最后就是把卷组再划分成为逻辑卷 (LV)，当然逻辑卷也是可以直接调整大小的。我们说逻辑卷可以想象成为分区，所以也需要格式化和挂载。

/boot 分区必须是普通分区

LVM 命令

常见的 LVM 管理命令

功能	物理卷管理	卷组管理	逻辑卷管理
扫描 scan	pvscan	vgscan	lvscan
建立 create	pvcreate <code>1</code>	vgcreate <code>2</code>	lvcreate <code>3</code>
显示 display	pvdisplay <code>PVs</code>	vgdisplay <code>VGs</code>	lvdisplay
删除 remove	pvremove	vgremove	lvremove
扩展 extend		vgextend	lvextend
减少 reduce		vgreduce	lvreduce

`xfs` 不支持 LVM 缩容

4. 格式化逻辑卷 5. 挂载使用

建立物理卷

`pvcreate [选项] 分区 分区2` 将分区或磁盘转化为物理卷。

可用 PVs 查看

- f 强制创建物理卷
- u 指定设备的UUID
- y 所有回答为 yes

建立卷组	<code>vgcreate [选项] 卷组名 要添加的物理卷</code>	可用 <code>vgs</code> 查看	
	-s < > 指定PE大小		
	-L < > 卷组上最多允许创建的逻辑卷数		
	-P < > 卷组中最多允许添加的物理卷数		
创建逻辑卷	<code>lvcreate -n 逻辑卷名 -L 逻辑卷大小 所属卷组名</code>		
删除 LVM	① 删除 LV <code>lvremove</code>	② 删除 VG <code>vgremove</code>	③ 删除物理卷 <code>PVremove</code>
			按顺序删除
扩容逻辑卷	<code>lv extend -L +2G 逻辑卷路径</code>		
	刷新文件系统 <code>resize2fs 逻辑卷路径</code>		<code>xfs_growfs 逻辑卷路径</code>
扩容卷组	① 添加物理卷 <code>PVcreate 分区路径</code>		
	② 卷组扩容 <code>vgextend 要扩容的卷组名 物理卷路径</code>		

RAID 阵列

模式 0 1 4 5 6

RAID 0：条带模式 IO 性能最强 数据无冗余，最不安全

RAID 1：镜像模式 数据冗余完整 安全系数最高 IO 压力最大，效率最低，使用成本最高 硬盘数量必须为 2 的倍数。

RAID 4：一个硬盘上存放校验信息 至少 3 块硬盘

RAID 5：分布存放校验和 至少 3 块硬盘 可经受一块硬盘故障

RAID 6：对校验和进行镜像备份 至少 4 块硬盘 可经受两块硬盘故障

RAID 1+0 至少 4 块硬盘 性能高，可靠性高，成本高

RAID 5+0 至少 6 块 比 1+0 经济一点

软 RAID 流程：

- ① 创建硬盘分区，并设置分区类型为 fd (Linux raid autodetect)

② 使用 mdadm 命令创建 RAID

mdadm -C /dev/md0 -n 4 -l 10 -x 1 /dev/sd[b-e]

③ 重建文件系统 mkfs.....

④ 挂载设备

-D	显示 RAID 设备的详细信息
-A	加入一个以前定义的 RAID
-B	创建一个没有超级块的 RAID 设备
-F	选项监控模式
-G	更改 RAID 设备的大小或形态
-I	添加设备到 RAID 中，或从 RAID 中删除设备
-Z	组建 RAID1、RAID4、RAID5、RAID6 后从每个 RAID 成员获取的空间容量
-S	扫描配置文件或 /proc/mdstat 以搜寻丢失的信息
-C /dev/md5	把 RAID 信息写入每个成员超级块中
-V	显示 RAID 创建过程中的详细信息
-B	不把 RAID 信息写入每个成员的超级块中
-I	指定 RAID 的级别
-N	指定 RAID 中活动设备的数目
-f	把 RAID 成员列为有问题，以便移除该成员
-r	把 RAID 成员移出 RAID 设备
-a	向 RAID 设备中添加一个成员
--re-add	把最近移除的 RAID 成员重新添加到 RAID 设备中
-c	创建一个 RAID 设备时默认为 512kb
-R	开始部分组装 RAID
-S	停用 RAID 设备，释放所有资源
-X	指定初始 RAID 设备的备用成员的数量
--zero-superblock	如果 RAID 设备包含一个有效的超级块，该块使用零覆盖



• crontab

-

一 选项

定时任务

后台守护进程 **crond**

-e 编辑 crontab 定时任务

-l 查询 crontab 定时任务

-r 删除当前用户所有的 crontab 任务

一 定时任务格式

* * * * * 要执行的任务
分 时 天 月

第一个“*”	一小时当中的第几分钟	0~59
第二个“*”	一天当中第几小时	0~23
第三个“*”	一月当中第几天	1~31
第四个“*”	一年当中第几月	1~12
第五个“*”	一周当中星期几	0~7 (0和7都表示周日)

特殊符号	含义
*	代表任何时间。比如第一个“*”就代表一小时中每分钟都执行一次的意思。
,	代表不连续的时间。比如“0 8,12,16 * * * 命令”，就代表在每天的 8 点 0 分，12 点 0 分，16 点 0 分都执行一次命令
-	代表连续的时间范围。比如“0 5 * * 1-6 命令”，代表在周一到周六的凌晨 5 点 0 分执行命令
/n	代表每隔多久执行一次。比如“*/10 * * * * 命令”，代表每隔 10 分钟就执行一遍命令

• xargs

xargs (英文全拼: eXtended ARGuments) 是给命令传递参数的一个过滤器, 也是组合多个命令的一个工具。
xargs 可以将管道或标准输入 (stdin) 数据转换成命令行参数, 也能够从文件的输出中读取数据。
xargs 也可以将单行或多行文本输入转换为其他格式, 例如多行变单行, 单行变多行。
xargs 默认的命令是 echo, 这意味着通过管道传递给 xargs 的输入将会包含换行和空白, 不过通过 xargs 的处理, 换行和空白将被空格取代。
xargs 是一个强有力的命令, 它能够捕获一个命令的输出, 然后传递给另外一个命令。
之所以能用到这个命令, 关键是由于很多命令不支持|管道来传递参数, 而日常工作中有这个必要, 所以就有了 xargs 命令, 例如:

```
find /sbin -perm +700 | ls -l      #这个命令是错误的  
find /sbin -perm +700 | xargs ls -l  #这样才是正确的
```

xargs 一般是和管道一起使用。

• column

将文本对齐显示,

• wc

统计文件内容

-l 行数

-c 字节数

-w 单词数

• du

统计文件占用磁盘情况

• lsblk

查看块状设备信息 (如, 光盘, 硬盘, U 盘等)

• 上传下载

安装 lrzsz

yum -y install lrzsz

rz 上传文件

sz 文件名 下载文件. 注: 不能下载文件夹, 需先打包。

• 链接

ln [参数] 源文件或目录 目标文件或目录.

创建一个硬链接.

-s

创建一个软链接

-b

删除覆盖以前的链接

-f

强制执行

-i

交互模式, 链接存在则提示用户是否覆盖.

• 因果只能创建软链接, 必须使用绝对路径

• 软链接与硬连接的区别

【硬连接】

硬连接指通过索引节点来进行连接。在Linux的文件系统中, 保存在磁盘分区中的文件不管是什么类型都给它分配一个编号, 称为索引节点号(inode index)。在Linux中, 多个文件名指向同一索引节点是存在的。一般这种连接就是硬连接。硬连接的作用是允许一个文件拥有多个有效路径名, 这样用户就可以建立硬连接到重要文件, 以防止“误删”的功能。其原因如上所述, 因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接, 只有当最后一个连接被删除后, 文件的数据块及目录的连接才会被释放。也就是说, 文件真正删除的条件是与之相关的所有硬连接文件均被删除。

【软连接】

另外一种连接称之为符号连接 (Symbolic Link), 也叫软连接。软链接文件有类似于Windows的快捷方式。它实际上是一个特殊的文件。在符号连接中, 文件实际上是一个文本文件, 其中包含的有另一文件的位置信息。

因果不允许创建硬连接

• file

file 文件名

判断文件类型

• 别名

alias 别名=原命令+参数

• dd

用于读取转化并输出数据.

if = 文件名

输入文件

of = 文件名

输出文件

bs = bytes

同时设置输入输出的块的大小为 bytes 个字节

ibs = bytes

输入块的大小

obs = bytes

输出块的大小

Count = n

输出几个块