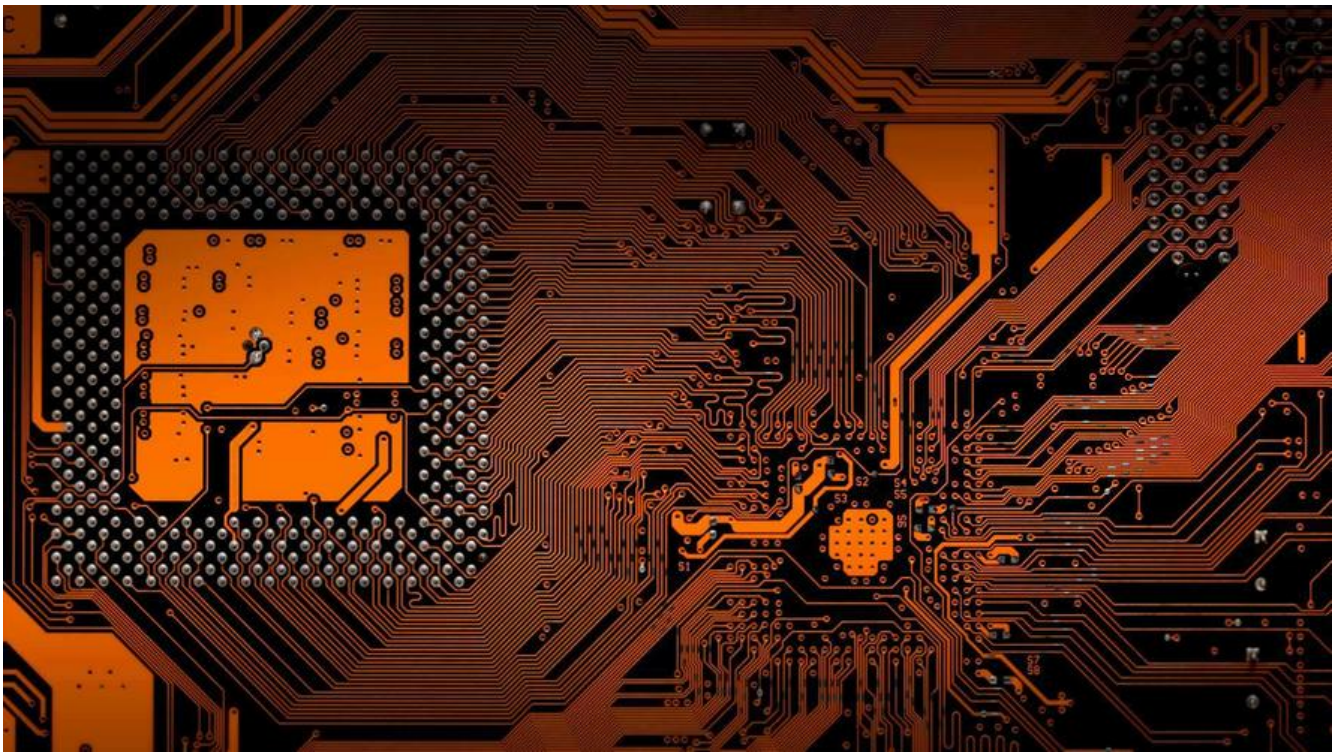
链滴

RHCSA 笔记

作者：Gao-Eason

原文链接：https://ld246.com/article/1649666586928

来源网站：链滴



Linux显示行号设置

第一步，打开vim

```
vi ~/.vimrc
1
```

第二步，在该文件中加入一行，命令如下：

```
set nu                  # 显示行号
set nonu                  # 不显示行号
```

命令终端字段含义介绍

- **[root@localhost ~]# **
- **解释：**
  - **root：当前登录系统用户名(root超级管理员)**
  - **localhost ： 当前主机名**
  - ~~**：当前用户所在目录（****~~** 为家目录） ，root超级管理员家目录：/root**
  - **#： 当前用户身份是超级管理员**
- **[student@localhost ~]$ **
  - **$：当前用户身份为普通用户，普通用户的家目录：/home/用户名同名**

Linux系统基本概念

- **多用户的系统：允许同时有很多个用户登录系统，使用系统里的资源**
- **多任务的系统：允许同时执行多个任务**
- **严格区分大小写：命令，选项，参数，文件名，目录名都严格区分大小写**
- **一切皆文件：硬件设备（内存、CPU、网卡、显示器、硬盘等等）都是以文件的形式存在的**
- **不管是文件还是目录都是以倒挂的树形结构，存在于系统的“/”根目录下，根目录是Linux系统的点**
- **对于Linux系统而言，目录/文件没有扩展名一说，扩展名如：.sh（脚本文件） .conf（配置文件） .og（日志文件） .rpm（软件包）.tar（压缩包）是易于用户方便识别**
- **没有提示就是最好的提示（成功了）**
- **Linux系统没有回收站**

命令行编辑技巧

键盘上下键调出历史命令

Ctrl + c：废弃当前命令行中的命令，取消当前执行的命令，例如ping

Ctrl + l,clear：清屏

tab建自动补齐：可补齐命令、参数、文件路径、软件名

esc + . ：将上一条命令参数变成当前命令的执行对象

Ctrl + a：将当前光标移动至行首

Ctrl + e：将当前光标移动至行尾

** Ctrl + u 清空至行首**

** Ctrl + w 删除一个单词**

exit，logout：退出系统

命令行一般命令格式

- **命令字 [-选项...] [参数...]**
  - **命令字：命令本身（功能）**
  - **选项：调整命令功能的**
    - **短选项：-l -a -d -h（单个字符），短选项可以合并使用：-lad -lh**
    - **长选项：--help（单词），长选项通常是不能合并使用的**
  - **参数：命令的执行对象，文件/目录/程序等**
  - **[]：可选的**
  - **...：可以同时有多个选项或参数**

学习方法

- **遇到问题：前期不要求你们有排错的能力**
- **思考自己能不能决绝：百度、Google、最后在问老师**
- **主动学习的爱好，不要被动学习**
- **不要死磕一个技术点，低头学习的时候不要忘了抬头看路**

Linux系统辨别目录与文件的方法

蓝色表示目录（windows系统里的文件夹）

白色表示文件

浅蓝色表示链接文件（类似于windows系统的快捷方式）

绿色表示可执行文件（如脚本，命令程序文件）

红色表示压缩文件

黄色表示设备文件（硬盘、键盘、鼠标、网卡、CPU硬件设备都是以文件的形式存在的）

红色闪动文件——>表示链接文件不可用

ls 查看目录/文件命令

- **ls命令（英文全拼：list）：用于查看目录下内容及目录和文件详细属性信息**
- **命令格式：ls [-选项...] [参数...]**
- **常用选项：**
  - **-a 显示目录下所有内容，包含隐藏的内容**
  - **-l 以长格式显示目录下的内容及详细属性**
  - **-h 人性化显示目录下内容大小（kB、MB、GB）**
  - **-d 仅显示目录本身而不显示目录下的内容**
  - **-i 查看inode号（系统任何的文件或目录都有一个唯一的编号）**
  - **-R：递归查看目录下所有内容（从头到尾）**
- **注意（附加）：递归是指将所有的目录从头到尾全部呈现出来。**

Linux 系统文件类型

-** 文件：**

d 目录：

l 链接文件

b 跨设备文件

c 字符设备文件

p 管道设备文件

s 套接字

Linux 系统下的归属关系

- **在Linux系统下，文件给用户分成了三类**
  - **所有者：文件或目录的拥有者，拥有者的权限通常是最大的**
  - **所属组：文件或目录属于哪一个组，所属组的权限略微比所有者小**
  - **其他人：既不是文件或目录的所有者，也不属于文件或目录组内的成员，其他人的权限通常最的权限**

- **ls命令示例：**

```
#显示当前所在目录下的所有内容
[root@localhost ~]# ls

#查看根目录下所有内容
[root@localhost ~]# ls  /
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr
ar

#查看/etc目录下所有内容
[root@localhost ~]# ls /etc

#查看/bin目录下所有内容
[root@localhost ~]# ls /bin

#查看/dev目录下所有内容
[root@localhost ~]# ls /dev

#查看目录下所有目录和文件，包括隐藏的内容
[root@localhost ~]# ls -a

#以长格式显示目录下所有内容，包括详细的属性信息
[root@localhost ~]# ls -l
-rw-r--r--. 1 root root 0 10月 24 15:16 hello

#解释
-：文件类型
1：代表文件的引用次数
root：文件的所有者
root：文件的所属组
0：文件的大小，默认以字节为单位显示大小
10月 24 15:16：文件最近一次的修改时间
hello：文件名

#以长格式显示目录所有内容，以人性化的方式显示详细的属性信息
[root@localhost ~]# ls -l -h

#短选项合并使用
[root@localhost ~]# ls -lh
```

#以长格式显示目录所有内容，以人性化的方式显示详细的属性信息，包括隐藏的内容
[root@localhost ~]# ls -lha

#以长格式显示根目录下所有内容，包括详细的属性信息
[root@localhost ~]# ls -l /
lrwxrwxrwx. 1 root root   7 3月 13 17:15 bin -> usr/bin

#创建hello.txt文件
[root@localhost ~]# touch hello.txt

#查看文件的元数据信息
[root@localhost ~]# stat hello.txt
  文件："hello.txt"
  大小：0   块：0   IO 块：4096   普通空文件
设备：fd00h/64768d   Inode：33575020   硬链接：1
权限：(0644/-rw-r--r--) Uid：(0/   root)   Gid：(0/   root)
环境：unconfined_u:object_r:admin_home_t:s0
最近访问：2021-03-14 16:38:14.349861770 +0800
最近更改：2021-03-14 16:38:14.349861770 +0800
最近改动：2021-03-14 16:38:14.349861770 +0800
创建时间：-

Linux 基本权限的类别

- **r 读取 w 写入 x 执行 - 没有权限**
- **权限顺序：rwx rwx rwx**

[root@localhost ~]# ls -l
-rw-r--r--. 1 root root 1831 3月 13 17:45 initial-setup-ks.cfg
#解释
-：文件类型
rw- r-- r--：所有者u、所属组g、其他人o的权限
u   g   o

r 读取权限，w写入权限，x执行权限，-没有任何权限

1：代表文件的引用次数，只针对与做了硬连接的文件才有效
root：文件的所有者
root：文件的所属组
1831：文件的大小，默认以字节为单位显示大小
3月 13 17:45：文件最近一次的修改时间
initial-setup-ks.cfg：文件名

#查看/root目录本身详细属性信息
[root@localhost ~]# ls -ld /root
dr-xr-x---. 14 root root 4096 3月 14 16:38 /root

#查看当前目录下所有内容的inode号
[root@localhost ~]# ls -i
33574979 anaconda-ks.cfg 33574984 initial-setup-ks.cfg 33575035 模板 33575036 图片 174
0701 下载   17470702 音乐

33575020 hello.txt 51909391 公共 51909392 视频 3204374 文档 33575017 新文件夹.zip 3204373 桌面

#查看hello.txt文件的inode号
[root@localhost ~]# ls -i hello.txt
33575020 hello.txt

#查看/etc/目录本身的inode号
[root@localhost ~]# ls -id /etc
16777281 /etc

课后练习

1.命令行以$作为结尾代表什么含义?

普通用户

2.请写出Linux系统一般的命令格式?

命令字 [-选项...] [参数...]

3.在Linux系统下，如何辨别目录与文件及其他的文件?

白色：文件

蓝色：目录

浅蓝色：链接文件

绿色：可执行文件

红色：压缩文件

红色带闪动的文件：链接文件不可用

黄色：设备文件（硬盘，网卡，CPU，鼠标，键盘）

4.如何查看一个文件的详细属性?

ls -l 文件名

5.如何查看一个目录本身的详细属性?

ls -dl 目录名字

6.查看文件详细属性，并以KB、MB、GB的方式显示文件的大小?

ls -lh 文件名

7.如何查看一个文件的inode号?

ls -i 文件名

8.请写出Linux下文件和目录的三个归属关系?

u 所有者

g 所属组

o 其他人

9.请写出Linux下基本权限的表示方式?

r：读取，w写入，x执行

10.命令行以#作为结尾代表什么含义?

超级管理员

mkdir 创建目录命令

- **mkdir（英文全拼：make directory）用于创建新目录**
- **命令格式：mkdir [-选项] 目录名**
- **常用选项：**
  - **-p 递归创建多个目录**
- **注意事项：**
  - **目录还是文件的名字，除了以“/”以外的任意名称，“/”根目录，路径分隔符**
  - **文件或目录的名字长度不能超过255个字符**
- **mkdir命令示例**

```
#在当前所在目录创建test目录
[root@localhost ~]# mkdir test
[root@localhost ~]# ls

#在当前所在目录同时创建多个目录
[root@localhost ~]# mkdir test1 test2 test3
[root@localhost ~]# ls

#指定在/tmp目录下创建abc目录
[root@localhost ~]# mkdir /tmp/abc
[root@localhost ~]# ls /tmp
abc

#在指定目录下同时创建多个目录
[root@localhost ~]# mkdir /tmp/abc1 /tmp/abc2 /tmp/abc3
[root@localhost ~]# ls /tmp

#在/opt目录下创建student，在当前目录创建student1..3
[root@localhost ~]# mkdir /opt/student student1  student2 student3
[root@localhost ~]# ls /opt
rh  student

#mkdir默认无法在一个不存在的目录下创建目录，需要通过-p选项
[root@localhost ~]# mkdir /opt/xx/oo
```

```
mkdir: 无法创建目录"/opt/xx/oo": 没有那个文件或目录

[root@localhost ~]# mkdir /opt/a/b/c/d
mkdir: 无法创建目录"/opt/a/b/c/d": 没有那个文件或目录

#在/opt目录下递归创建目录
[root@localhost ~]# mkdir -p /opt/xx/oo
[root@localhost ~]# ls /opt
rh  student  xx

[root@localhost ~]# mkdir -p /opt/a/b/c/d
[root@localhost ~]# ls /opt
a  rh  student  xx

#ls -R选项可以递归目录下所有内容
[root@localhost ~]# ls -R /opt/a
/opt/a:
b

/opt/a/b:
c

/opt/a/b/c:
d
```

cd 切换工作目录命令

- **cd（英文全拼：change directory）切换目录**

命令格式：cd [-选项] [目录名]

- **提示：目录名称可以是绝对路径或相对路径，如果不指定目录名称，则切换到当前用户的家目录~**

绝对路径与相对路径

- **绝对路径：以/（根）为起点，到达你想去的目标目录称为绝对路径**
- **相对路径：以当前路径为起点，到达你想去的目标目录**
- **常用快捷操作：**
  - **~ 表示为家目录**
  - **. 表示为当前目录**
  - **.. 表示上一级目录**
- **-可在两路径之间来回切换**

pwd 打印当前所在目录

- **pwd（英文全拼：print work directory）打印当前所在的工作目录，执行pwd命令后，可显示前所在的工作目录的绝对路径名称**
- **命令格式：pwd [-选项] **

```
[root@localhost ~]# cd /opt/a/b/c/d

#打印当前所在目录绝对路径
[root@localhost d]# pwd
/opt/a/b/c/d

#切换到用户家目录
[root@localhost d]# cd ~
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /opt/a/b/c/d
[root@localhost d]# pwd
/opt/a/b/c/d
[root@localhost d]# cd
[root@localhost ~]# pwd
/root

[root@localhost ~]# cd /bin
[root@localhost bin]# pwd
/bin

[root@localhost bin]# cd /boot
[root@localhost boot]# pwd
/boot
[root@localhost boot]# ls

[root@localhost boot]# cd /dev
[root@localhost dev]# pwd
/dev
[root@localhost dev]# ls

[root@localhost dev]# cd /etc
[root@localhost etc]# pwd
/etc
[root@localhost etc]# ls

[root@localhost etc]# ls /
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr
ar

# “.”表示当前所在目录，对于cd命令而言作用不大
[root@localhost etc]# cd .

[root@localhost etc]# cd /opt/a/b/c/d
[root@localhost d]# pwd
/opt/a/b/c/d

# “..”切换到当前目录的上一级目录
[root@localhost d]# cd ..
[root@localhost c]# pwd
/opt/a/b/c

[root@localhost c]# cd ..
[root@localhost b]# pwd
```

```
/opt/a/b

[root@localhost b]# cd ..
[root@localhost a]# cd ..
[root@localhost opt]# pwd
/opt

[root@localhost opt]# cd ..
[root@localhost /]# cd ..
[root@localhost /]# cd
[root@localhost ~]# ls

[root@localhost ~]# cd /opt/a/b/c/d
[root@localhost d]# pwd
/opt/a/b/c/d

#"-"可在两个路径之间来回切换
[root@localhost d]# cd /etc/yum
[root@localhost yum]# cd -
/opt/a/b/c/d

[root@localhost d]# pwd
/opt/a/b/c/d

[root@localhost d]# cd -
/etc/ym

[root@localhost yum]# cd -
/opt/a/b/c/d

[root@localhost d]# cd -
/etc/yum
```

rmdir 删除空目录命令

- **rmdir（英文全拼：remove directory）删除空目录**
- **命令格式：rmdir [-选项] 目录名**

```
#rmdir只能删除空目录，如果目录下存在数据无法删除
[root@localhost ~]# rmdir /opt/a
rmdir: 删除 "/opt/a" 失败: 目录非空
[root@localhost ~]# ls -R /opt/a
/opt/a:
b

/opt/a/b:
c

/opt/a/b/c:
d

/opt/a/b/c/d:
```

```
[root@localhost ~]# rmdir /opt/a/b/c/d
[root@localhost ~]# ls -R /opt/a
/opt/a:
b

/opt/a/b:
c

/opt/a/b/c:

[root@localhost ~]# rmdir /opt/a/b/c
[root@localhost ~]# ls -R /opt/a/b
/opt/a/b:

[root@localhost ~]# rmdir /opt/a/b
[root@localhost ~]# ls -R /opt/a
/opt/a:

[root@localhost ~]# rmdir /opt/a
[root@localhost ~]# ls /opt
rh  student  xx

[root@localhost ~]# rmdir /opt/
rmdir: 删除 "/opt/" 失败: 目录非空
```

touch 创建文件命令

- **touch 命令用于创建新的空白文件**
- **命令格式：touch [-选项] 文件名**

```
#在当前路径创建空文件
[root@localhost ~]# touch hello
[root@localhost ~]# ls

#在当前路径同时创建多个文件
[root@localhost ~]# touch t1 t2 t3 t4
[root@localhost ~]# ls

#在指定路径同时创建多个文件
[root@localhost ~]# touch /opt/test1 /opt/test2 /opt/test3
[root@localhost ~]# ls /opt
rh  student  test1  test2  test3  xx

#如果存在同名目录时，无法创建
[root@localhost ~]# mkdir test
mkdir: 无法创建目录"test": 文件已存在

#如果存在同名文件时，touch命令没有提示，但原有文件不会被覆盖
[root@localhost ~]# touch t1

#对于目录而言，只有单个目录的时候，  “/”  可有可无
[root@localhost ~]# ls /opt/
rh  student  test1  test2  test3  xx
```

```
[root@localhost ~]# ls /opt
rh  student  test1  test2  test3  xx

#对于目录而言，查看目录下的内容时，必须要有“/”
[root@localhost ~]# ls /opt/xx
oo

#对于文件而言，后边绝对不能有“/”
[root@localhost ~]# ls /opt/test1
/opt/test1
[root@localhost ~]# ls /opt/test1/
ls: 无法访问/opt/test1/: 不是目录
```

cp 复制命令

- **cp（英文全拼：copy file）用于复制文件或目录，cp命令在复制时也可修改目录或文件名字**
- **命令格式：cp [-选项] 源文件或目录 目标目录**
- **常用选项：**
  - **-p 保留源文件属性不变（如：修改时间、归属关系、权限）**
  - **-r 复制目录（包含该目录下所有的子目录和文件）**

```
#复制当前目录文件到/opt目录（相对路径方式复制）
[root@localhost ~]# cp t1 /opt/
[root@localhost ~]# ls /opt
rh  student  t1  test1  test2  test3  xx

#复制文件到/opt目录（绝对路径方式复制）
[root@localhost ~]# cp /root/t2 /opt
[root@localhost ~]# ls /opt
rh  student  t1  t2  test1  test2  test3  xx

#同时复制多个文件
[root@localhost ~]# cp t3 t4 /opt/
[root@localhost ~]# ls /opt

#创建目录
[root@localhost ~]# mkdir abc

#使用-r对目录执行复制
[root@localhost ~]# cp -r abc /opt
[root@localhost ~]# ls /opt

#同时复制多个目录
[root@localhost ~]# mkdir abc1 abc2 abc3
[root@localhost ~]# cp -r abc1 abc2 abc3 /opt
[root@localhost ~]# ls /opt

#复制hello文件到/opt并改名为hello.txt
[root@localhost ~]# cp hello /opt/hello.txt
[root@localhost ~]# ls /opt
```

```
#复制xxxx目录到/opt并改名xxoo
[root@localhost ~]# mkdir xxxx
[root@localhost ~]# cp -r xxxx /opt/xxoo
[root@localhost ~]# ls /opt

#使用“.”配合cp命令执行复制
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# pwd
/etc/sysconfig/network-scripts

[root@localhost network-scripts]# cp /root/t1 .
[root@localhost network-scripts]# ls

#操持属性不变复制文件
[root@localhost ~]# cp -p anaconda-ks.cfg /opt
cp：是否覆盖"/opt/anaconda-ks.cfg"？ y
[root@localhost ~]# ls -l /opt/anaconda-ks.cfg
-rw-------. 1 root root 1800 3月  13 17:34 /opt/anaconda-ks.cfg

#对比以上两个文件的详细属性信息（最后一次修改时间）
[root@localhost ~]# ls -l anaconda-ks.cfg
-rw-------. 1 root root 1800 3月  13 17:34 anaconda-ks.cfg

#这两个操作代表什么意思?
[root@localhost ~]# cp -r xxxx /mnt/oooo  #拷贝并改名
[root@localhost ~]# cp -r xxxx /mnt/oooo  #拷贝
```

mv 移动命令

- **mv（英文全拼：move file）用于移动文件或目录到其他位置，也可用于修改目录或文件名**
- **命令格式：mv [-选项] 源文件... 目标路径**

```
#移动当前路径hello文件到/mnt目录
[root@localhost ~]# mv hello /mnt
[root@localhost ~]# ls /mnt
hello  home  oooo  test

#同时移动多个文件
[root@localhost ~]# mv t1 t2 t3 t4 /mnt
[root@localhost ~]# ls /mnt
hello  home  oooo  student1  t1  t2  t3  t4  test

#移动/opt目录下文件到/mnt
root@localhost ~]# mv /opt/test1 /opt/test2 /opt/test3 /mnt/
[root@localhost ~]# ls /mnt
hello  home  oooo  student1  t1  t2  t3  t4  test  test1  test2  test3

#移动目录
[root@localhost ~]# mv student1 /mnt
[root@localhost ~]# ls /mnt
hello  home  oooo  student1  test

#移动文件并改名
```

```
[root@localhost ~]# mv hello.txt /media/hello
[root@localhost ~]# ls /media/
hello

#移动目录并改名
[root@localhost ~]# mv test /media/testxx
[root@localhost ~]# ls /media/
hello  testxx
```

cat 查看文件内容命令

- **cat （英文全拼：concatenate）命令用于查看文本文件内容**
- **命令格式：cat [选项] 文件名**
- **常用选项**
  - **-n #查看文件时以行号的形式显示文件内容**

```
#查看文件内容
[root@localhost ~]# cat anaconda-ks.cfg
[root@localhost ~]# cat initial-setup-ks.cfg
[root@localhost ~]# cat /etc/hosts

#查看网卡文件内容，网卡配置文件
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens32
...
NAME="ens32"  //网卡名
UUID="16085f4c-f690-4058-b29e-d55c73387026"
DEVICE="ens32"
ONBOOT="yes"
IPADDR="192.168.0.50"    //网卡IP地址
PREFIX="24"              //子网掩码
GATEWAY="192.168.0.254"  //网关
DNS1="114.114.114.114"   //DNS

#查看当前系统用户基本信息文件内容
[root@localhost ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

#查看当前系统主机名配置文件内容
[root@localhost ~]# cat /etc/hostname
localhost.localdomain

#查看当前系统版本信息文件内容
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 7.6.1810 (Core)

#查看当前系统开机自动挂载配置文件内容
[root@localhost ~]# cat /etc/fstab
```

```
#查看系统组基本信息文件内容
[root@localhost ~]# cat /etc/group

#使用“-n”以行号形式显示文件内容
[root@localhost ~]# cat -n /etc/passwd
[root@localhost ~]# cat -n /etc/hostname
[root@localhost ~]# cat -n /etc/fstab
[root@localhost ~]# cat -n /etc/group
[root@localhost ~]# cat -n /etc/services
```

less命令

- **less工具是对文件的输出进行分页显示的工具，常用于查看内容量较大的文件**
- **命令格式：less [-选项] 文件**
- **常用选项：**
  - **-N #以行号形式显示文件内容**
- **使用技巧：**
  - **键盘上下键逐行查看**
  - **pgdn：向下翻一页（Fn + 下键）**
  - **pgup：向上翻一页（Fn + 上键）**
- **/字符串：搜索指定字符串（n从上向下搜索，N从下向上搜索）**
  - **G：直接跳转到文件最后一行**
  - **gg：直接跳转到文件行首**
  - **：1000 #精准的定位到某一行**
  - **q：退出**

```
[root@localhost ~]# less -N /etc/services
```

head与tail命令

- **head命令：用来显示文件开头部分内容，默认显示文件开头10行内容**
- **命令格式：head [选项] 参数**
- **常用选项：**
  - **-n<行数> 指定显示的行数 **

```
[root@localhost ~]# head /etc/passwd
[root@localhost ~]# head /etc/fstab
[root@localhost ~]# head /etc/group
[root@localhost ~]# head /etc/hostname
[root@localhost ~]# head /etc/hosts
[root@localhost ~]# head /etc/sysconfig/network-scripts/ifcfg-ens32
```

```
#查看存放DNS配置文件信息
[root@localhost ~]# head /etc/resolv.conf

#使用-n指定显示文件前多少行内容
[root@localhost ~]# head -n 5 /etc/passwd
[root@localhost ~]# head -n 6 /etc/passwd
[root@localhost ~]# head -n 15 /etc/passwd
[root@localhost ~]# head -n 20 /etc/passwd
```

- **tail命令：用来显示文件末尾部分内容，默认显示文件末尾10行内容**
- **命令格式：tail [选项] 参数**
- **常用选项：-n<行数> 指定显示的行数 -f 动态显示**

```
[root@localhost ~]# tail /etc/passwd

#使用“-n”指定显示文件末尾多少行内容
[root@localhost ~]# tail -n 5 /etc/passwd
[root@localhost ~]# tail -n 5 /etc/sysconfig/network-scripts/ifcfg-ens32
IPADDR="192.168.0.50"
PREFIX="24"
GATEWAY="192.168.0.254"
DNS1="114.114.114.114"
IPV6_PRIVACY="no"

#动态查看文件内容
[root@localhost ~]# touch t1
root@localhost ~]# tail -f t1

#另开一个终端向文件写入内容
[root@localhost ~]# echo 123 > t1
```

rm删除命令

- **rm（英文全拼：remove）命令用于删除文件或者目录。**
- **命令格式：rm [-选项…] 目录或文件…**
- **常用选项**
  - **-f 强制删除**
  - **-r 删除目录**
  - **“*”特殊字符：系统常用符号，用来代表任意所有字符**

```
[root@localhost ~]# ls /opt
abc  abc1  abc2  abc3  anaconda-ks.cfg  hello.txt  home  rh  student  t1  t2  t3  t4  xx  xxoo

[root@localhost ~]# ls /mnt
hello  home  oooo  student1  t1  t2  t3  t4  test  test1  test2  test3

#删除指定目录下文件
[root@localhost ~]# rm /opt/anaconda-ks.cfg
rm：是否删除普通文件 "/opt/anaconda-ks.cfg"？ y  #默认需要确认（y|n）
```

```
#查看文件是否被成功删除
[root@localhost ~]# ls /opt
abc  abc1  abc2  abc3  hello.txt  home  rh  student  t1  t2  t3  t4  xx  xxoo

[root@localhost ~]# rm /opt/hello.txt
rm：是否删除普通空文件 "/opt/hello.txt"？ y

#同时删除目录下指定文件
[root@localhost ~]# rm /opt/t1 /opt/t2 /opt/t3 /opt/t4
rm：是否删除普通空文件 "/opt/t1"？ y
rm：是否删除普通空文件 "/opt/t2"？ y
rm：是否删除普通空文件 "/opt/t3"？ y
rm：是否删除普通空文件 "/opt/t4"？ y

#查看文件是否被成功删除
[root@localhost ~]# ls /opt
abc  abc1  abc2  abc3  home  rh  student  xx  xxoo

#使用“-f”强制删除文件（无需确认，直接删除）
[root@localhost ~]# rm -f /mnt/hello
[root@localhost ~]# ls /mnt
home  oooo  student1  t1  t2  t3  t4  test  test1  test2  test3

#同时强制删除多个文件
[root@localhost ~]# rm -f /mnt/t1 /mnt/t2 /mnt/t3 /mnt/t4
[root@localhost ~]# ls /mnt

#删除目录
[root@localhost ~]# rm  -r /opt/abc
rm：是否删除目录 "/opt/abc"？ y

[root@localhost ~]# ls /opt
abc1  abc2  abc3  home  rh  student  xx  xxoo

#同时删除多个目录
[root@localhost ~]# rm -r /opt/abc1 /opt/abc2 /opt/abc3
rm：是否删除目录 "/opt/abc1"？ y
rm：是否删除目录 "/opt/abc2"？ y
rm：是否删除目录 "/opt/abc3"？ y

[root@localhost ~]# ls /opt
home  rh  student  xx  xxoo

#同时强制删除多个目录
[root@localhost ~]# rm -rf /opt/home /opt/student /opt/xx /opt/xxoo
[root@localhost ~]# ls /opt
rh

#创建目录与文件
[root@localhost ~]# touch /opt/t1
[root@localhost ~]# mkdir /opt/test
[root@localhost ~]# ls /opt
rh  t1  test
```

```
#rm命令在删除目录时，包含改目录及目录下所有数据全部删除
[root@localhost ~]# rm -rf /opt/
[root@localhost ~]# ls /

[root@localhost ~]# ls /mnt
home  oooo  student1  test  test1  test2  test3

#使用“*”通配任意所有字符，删除/mnt目录下所有数据
[root@localhost ~]# rm -rf /mnt/*
[root@localhost ~]# ls /mnt
```

软连接与硬连接

- **Linux中的链接文件类似于windows中的快捷方式**
- **软连接特点：软连接可以跨分区，可以对目录进行链接，源文件删除后，链接文件不可用**
- **软连接命令格式：ln -s 源文件路径 目标路径**
- **注意：创建链接时一定要写目录或文件的绝对路径，哪怕是在当前路径下，也要写绝对路径•**

```
[root@localhost ~]# touch hello.soft
[root@localhost ~]# ls

#创建软连接（必须要绝对路径创建）
[root@localhost ~]# ln -s /root/hello.soft /opt
[root@localhost ~]# ls /opt

#查看连接文件详细属性
[root@localhost ~]# ls -l /opt/hello.soft
lrwxrwxrwx. 1 root root 16 3月  21 14:28 /opt/hello.soft -> /root/hello.soft
#提示：链接文件的权限最终取决于源文件的权限

#普通用户验证
[lisi@localhost ~]$ ls /opt
hello.soft
[lisi@localhost ~]$ ls -l /opt/hello.soft
lrwxrwxrwx. 1 root root 16 3月  21 14:28 /opt/hello.soft -> /root/hello.soft
[lisi@localhost ~]$ cat /opt/hello.soft
cat: /opt/hello.soft: 权限不够
#提示：由于源文件存放于/root目录下，而普通用户对/root目录没有任何权限，所以普通用户无法
看

#删除源文件
[root@localhost ~]# rm -f /root/hello.soft
[root@localhost ~]# ls

#删除源文件后，软链接文件不可用
[root@localhost ~]# ls -l /opt/hello.soft
lrwxrwxrwx. 1 root root 16 3月  21 14:28 /opt/hello.soft -> /root/hello.soft

#创建文件并创建软连接
[root@localhost ~]# touch hello.soft
[root@localhost ~]# ln -s /root/hello.soft /opt
```

```
[root@localhost ~]# ls -l /opt/hello.soft
lrwxrwxrwx. 1 root root 16 3月  21 14:39 /opt/hello.soft -> /root/hello.soft

#删除链接文件后，源文件仍然可用
[root@localhost ~]# rm -f /opt/hello.soft
[root@localhost ~]# ls
[root@localhost ~]# cat hello.soft

#对目录创建软连接
[root@localhost ~]# ln -s /root/test1 /opt/

[root@localhost ~]# ls -ld /opt/test1
lrwxrwxrwx. 1 root root 11 3月  21 14:44 /opt/test1 -> /root/test1

3创建链接时一定要写目录或文件的绝对路径，哪怕是在当前路径下，也要写绝对路径
[root@localhost ~]# ln -s hello.soft /opt
[root@localhost ~]# ls /opt
hello.soft  test1
[root@localhost ~]# ls -l /opt/hello.soft
lrwxrwxrwx. 1 root root 10 3月  21 14:47 /opt/hello.soft -> hello.soft
```

- **硬链接特点：硬连接不可以跨分区，不可以对目录进行链接，源文件删除后，链接文件仍然可用**
- **硬连接命令格式：ln 源文件路径 目标路径**

```
#创建文件，并创建硬连接
[root@localhost ~]# touch hello.hard
[root@localhost ~]# ln /root/hello.hard /opt/
[root@localhost ~]# ls /opt
hello.hard  hello.soft  test1

#向硬连接的源文件写入内容
root@localhost ~]# echo 123 > /root/hello.hard

#查看源文件内容
[root@localhost ~]# cat /root/hello.hard
123

#查看链接文件内容，以同步更新
[root@localhost ~]# cat /opt/hello.hard
123

#向链接文件写入内容，查看源文件以同步更新
[root@localhost ~]# echo xx >> /opt/hello.hard

#擦看源文件，以同步更新
[root@localhost ~]# cat /root/hello.hard
123
xx

#硬连接文件的特点可以保持文件属性不发生改变
[root@localhost ~]# ls -l /root/hello.hard
-rw-r--r--. 2 root root 7 3月  21 14:55 /root/hello.hard
[root@localhost ~]# ls -l /opt/hello.hard
```

```
-rw-r--r--. 2 root root 7 3月  21 14:55 /opt/hello.hard

#并且硬连接文件的i节点号相同
[root@localhost ~]# ls -i /root/hello.hard
33711090 /root/hello.hard
[root@localhost ~]# ls -i /opt/hello.hard
33711090 /opt/hello.hard

#硬连接不允许对目录进行连接
root@localhost ~]# ln /root/test1 /opt
ln: "/root/test1": 不允许将硬链接指向目录

#硬连接源文件删除后，链接文件仍然可用
[root@localhost ~]# rm -f /root/hello.hard
[root@localhost ~]# cat /opt/hello.hard
123
xx

#向硬连接文件写入内容
[root@localhost ~]# echo  abc >> /opt/hello.hard
[root@localhost ~]# cat /opt/hello.hard
123
xx
abc

#硬连接不允许跨分区
[root@localhost ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda               8:0    0   20G  0 disk
├─sda1            8:1    0    1G  0 part /boot
└─sda2            8:2    0   19G  0 part
  ├─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sr0              11:0    1  4.3G  0 rom
[root@localhost ~]# ln /root/hello.soft /boot
ln: 无法创建硬链接"/boot/hello.soft" => "/root/hello.soft": 无效的跨设备连接
```

Linux命令的分类

- **内部命令：bash程序自带的基本管理命令**
- **外部命令：有独立的外部可执行程序文件命令**
- **type 用于区别内部命令与外部命令**
- **which 用于查找可以执行程序文件位置**

```
[root@localhost opt]# type ls

[root@localhost opt]# type cat

[root@localhost opt]# type hash

[root@localhost ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

```
[root@localhost ~]# hash
命中    命令
   1    /usr/bin/cat
   1    /usr/bin/ls


[root@localhost opt]# hash -r
[root@localhost opt]#
[root@localhost opt]# hash
hash: 哈希表为空

[root@localhost opt]# ls
hello.hard  hello.soft  t1  test1  test.txt
[root@localhost opt]# hash
命中    命令
   1    /usr/sbin/ls
```

- **总结：**
  - **shell程序是用户和系统之间的接口，用于解释用户的命令**
  - **查找命令对应的程序文件所在位置：which 命令**
  - **shell程序大多数存放在/etc/shells文件中**
  - **系统默认使用的shell为/bin/bash**
  - **查看当前使用的shell：echo $SHELL**
  - **区别内部命令与外部命令的方式：typt 命令**
  - **shell程序查找可执行程序文件路径定义在$PATH环境变量中**
  - **shell查找的外部命令路径结果会记录到缓存的hash表中**

help 命令帮助手册

- **help命令用于查看shell内部命令的帮助信息，包括使用方法、选项等...**
- **命令格式：help [选项] 命令**

```
#获取内部命令帮助信息
[root@localhost etc]# help cd

#help无法获取外部命令的帮助信息
root@localhost etc]# help ls
bash: help: 没有与 `ls' 匹配的帮助主题。尝试 `help help' 或者 `man -k ls' 或者 `info ls'。

[root@localhost etc]# type help
help 是 shell 内嵌

#获取help命令本身的帮助信息
[root@localhost etc]# help help

[root@localhost etc]# type cat
cat 是 /usr/bin/cat
```

```
[root@localhost etc]# help cat
bash: help: 没有与 `cat' 匹配的帮助主题。尝试 `help help' 或者 `man -k cat' 或者 `info cat'。

#查看命令帮助手册（命令自带）
[root@localhost etc]# cat --help
[root@localhost etc]# ls --help
```

man 获取命令帮助手册

- **man 命令用于查看系统命令的帮助信息，包括使用方法、选项、使用例子等...，对比--help ，man 输出的信息更加详细**
- **命令格式：man [-选项] 命令**
- **常用快捷操作**
  - **向下键向下移一行**
  - **向上键向上移一行**
  - **[Page Down] 向下翻一页**
  - **[Page Up] 向上翻一页**
  - **/关键字 #搜索关键字，配合n（向下查询）、N（向上查询）**
  - **q 退出**

```
[root@localhost etc]# man ls
[root@localhost etc]# man cat
[root@localhost etc]# man touch
[root@localhost etc]# man mkdir

[root@localhost etc]# info ls
```

Linux系统的运行级别

Linux系统运行级别：linux系统有7个运行级别，不同的运行级别运行的程序和功能都不一样，而Linux系统默认是运行在一个标准的级别上，系统运行级别文件/etc/inittab文件

运行级别 0：所有进程被终止，机器将有序的停止，关机时系统处于这个运行级别（关机）

运行级别 1：单用户模式，（root用户进行系统维护），系统里运行的所有服务也都不会启动

运行级别 2：多用户模式（网络文件系统NFS服务没有被启动）

运行级别 3：完全多用户模式，（有NFS网络文件系统）标准的运行级别，命令行模式

运行级别 4：系统未使用

运行级别 5：登录后，进入带GUI的图形化界面，标准的运行级别

运行级别 6：系统正常关闭并重启

```
#查看当前系统运行级别
[root@localhost etc]# runlevel
```

```
N 5
#解释；当前系统处于的运行级别
#解释：N代表没有从任何级别跳转过来

#切换系统运行级别
[root@localhost ~]# init N

#查看运行级别文件内容
[root@localhost ~]# cat /etc/inittab
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main targets:
#
# multi-user.target: analogous to runlevel 3   #运行级别3
# graphical.target: analogous to runlevel 5    #运行级别5
#
# To view current default target, run:
# systemctl get-default                    #查看当前系统默认的运行级别
#
# To set a default target, run:
# systemctl set-default TARGET.target    #修改当前系统默认运行级别

#查看默认运行级别
[root@localhost ~]# systemctl get-default
graphical.target  #默认运行级别为5

 #修改默认运行级别为3
[root@localhost ~]# systemctl set-default multi-user.target
[root@localhost ~]# systemctl get-default
multi-user.target

#修改默认运行级别为5
[root@localhost ~]# systemctl set-default graphical.target
[root@localhost ~]# systemctl get-default
graphical.target
```

关机与重启

- **linux下常用的关机命令有：shutdown、halt、poweroff、init**
  - **init 0 关机**
  - **halt #立刻关机**
- **shutdown –h now   #立刻关机**
  - **shutdown -h 10   #10分钟后自动关机**
  - **poweroff   #立刻关机   （记这个）**

```
[root@localhost ~]# poweroff
```

- **重启命令：reboot shutdown**
  - **reboot #立刻重启 （记这个）**
  - **shutdown -r now #立刻重启**
  - **shutdown -r 10 #过十分钟后重启**

[root@localhost ~]# reboot

课后练习

1.请在/tmp目录下创建student目录，并在student目录下同时创建t1、t2、t3文件

mkdir /tmp/student

cd /tmp/student/

touch t1 t2 t3

touch /tmp/student/t1 /tmp/student/t2 /tmp/student/t3

2.请在/tmp目录下递归创建test1/test2/test3目录

mkdir -p /tmp/test1/test2/test3

3.切换到/tmp/test1/test2/test3目录下，并打印(查看)当前所在目录

cd /tmp/test1/test2/test3

pwd

4.请同时在/opt、/media目录下创建upload文件

touch /opt/upload /media/upload

5.请将/opt目录下的upload文件移动至/tmp/test1/test2/test3目录下，并改名为upload.bak

mv /opt/upload /tmp/test/1/test/2/test3/upload.bak

6.请将/etc/passwd文件拷贝至/opt目录下，改名为passwd.bak，并保持属性不变

cp -p /etc/passwd /opt/passwd.bak

7.请将/etc/fstab文件拷贝至/opt目录下，并改名为fstab.bak

cp -p /etc/fstab /opt/fstab.bak

8.请将/etc/sysconfig/network-scripts/ifcfg-ens32 文件拷贝至/opt目录下，并改名为ens32.bak

cp /etc/sysconfig/network-scripts/ifcfg-ens32 /opt/ens32.bak

9.请删除/etc/yum.repos.d/目录下所有内容

rm -rf /etc/yum.repos.d/*

10.请在/etc/yum.repos.d/目录下创建local.repo文件

touch /etc/yum.repos.d/local.repo

11.请查看/etc/sysconfig/network-scripts/ifcfg-ens32文件末尾5行内容

**tail -5 /etc/sysconfig/network-scripts/ifcfg-ens32 **

tail -n 5 /etc/sysconfig/network-scripts/ifcfg-ens32

12.请查看/etc/passwd文件第1行内容

head -n 1 /etc/passwd

head -1 /etc/passwd

13.请查看/etc/hostname文件内容

**cat /etc/hostname **

14.请查看/etc/hosts文件内容

cat /etc/hosts

15.请说出软连接与硬连接的特点

软连接：可以跨分区，可以对目录链接，源文件删除后链接文件不可用

硬连接：不可以跨分区，不可以对目录进行连接，源文件删除后，链接文件以然可用

16.请在/opt目录下创建hello.soft文件，并创建软连接到/tmp目录下

touch /opt/hello.soft

ln -s /opt/hello.soft /tmp

17.请在/opt目录下创建hello.hard文件，并创建硬连接到/tmp目录下，并查看连接文件详细属性

touch /opt/hello.hard

ln /opt/hello.hard /tmp

18.如何获取ls命令的帮助信息?

man ls

ls --help

19.请说出Linux系统的运行级别

0：关机

1：单用户模式

2：多用户模式（没有NFS）

3：完全多用户模式，标准运行级别

4：保留

5：带GUI图形化界面，标准的运行级别

6：系统关闭并重启

20.如何重启Linux系统?

reboot

init 6

计算机硬件组成部分

- **输入设备：键盘、鼠标、触控屏等**
- **主机设备：主板、中央处理器（CPU）、主存储器（内存）、网卡、声卡、显示卡等**
- **输出设备：屏幕、耳机、打印机、投影仪等**
- **外部存储设备：硬盘、软盘、光盘、U盘等、蓝光光驱**
- **CPU缓存**
- **CPU比较主流的厂商**
  - **AMD公司**
  - **Interl公司**
- **CPU架构**
  - **x86架构，8086架构，80286，80386，x86称号**
  - **8位、16位、32位、64位，CPU一次可以处理的数据量，**
  - **32位CPU一次可以从内存中读取大约3.25G左右的数据量**
  - **64位CPU一次可以从内存中读取大约128G左右的数据量**
- **CPU核心**
  - **单核心，一颗CPU只能有一个运算单元**
  - **多核心，一颗CPU里边有两个以上的运算单元**

Linux系统目录介绍

- **/（根）:系统所有数据都存放在根目录下**
- **/bin：存放用户和管理员必备的可执行的二进制程序文件**
- **/boot：存放Linux系统内核及引导系统程序所需要的文件目录 **
- **/dev：存放硬件设备的目录，如键盘、鼠标、硬盘、光盘等等 （记住）**
- **/etc：存放服务的配置文件，用户信息文件 （记住）**
- **/root：超级管理员的家目录**

- **/home：系统普通用户的家目录（记住）**
- **/lib：存放系统中的程序运行所需要的共享库及内核模块 **
- **/opt：额外安装的可选应用程序包所放置的位置**
- **/srv：服务启动之后需要访问的数据目录 **
- **/tmp：一般用户或正在执行的程序临时存放文件的目录,任何人都可以访问,重要数据不可放置在目录下 **
- **/var：存放系统执行过程中经常变化的文件，如随时都在变化的日志文件就存放/var/log/下 （记）**
- **/mnt、/media ： 光盘和镜像等预设的挂载点（记住）**
- **/proc：Linux伪文件系统，该目录下的数据存在于内存当中，不占用磁盘空间**
- **/lib64 ： 存放函式库**
- **/run ： 程序或服务启动后，存放PID的目录**
- **/sys：存放被建立在内存中的虚拟文件系统**
- **/usr：操作系统软件资源所放置的目录 **
  - **/usr/bin：与/bin目录相同，存放用户可以使用的命令程序**
  - **/usr/lib：与/lib目录相同，存放系统中的程序运行所需要的共享库及内核模块**
  - **/usr/etc：用于存放安装软件时使用的配置文件**
  - **/usr/games：与游戏比较相关的数据放置处**
  - **/usr/include：c/c++等程序语言的档头(header)与包含档(include)放置处 **
  - **/usr/lib64：与/lib64目录相同，存放函式库**
  - **/usr/libexec：不经常被使用的执行程序或脚本会放置在此目录中**
  - **/usr/local： 额外安装的软件存放目录（记住）**
  - **/usr/sbin：该目录与/sbin目录相同，存放用户可执行的二进制程序文件**
  - **/usr/share： 放置只读架构的杂项数据文件 **
  - **/usr/src：一般软件源代码建议存放该目录下**

查看内核信息

- **uname 命令用于显示系统内核信息**
- **命令格式：uname [-选项...]**
- **常用选项：**
  - **-s ： 显示内核名称**
  - **-r ： 显示内核版本**

```
[root@localhost ~]# uname
Linux

[root@localhost ~]# uname -rs
Linux 3.10.0-957.el7.x86_64
#解释：
```

```
Linux     #内核名称
3         #主版本
10        #次版本
0         #修改版本
957       #补丁次数
el7       #Enterprise Linux（企业版Linux）
x86_64    #CPU架构

#Linux内核官网
https://www.kernel.org/
```

查看CPU信息

- **/proc/cpuinfo文件用于存放系统CPU信息**
- **lscpu 用于显示CPU架构信息**
- **命令格式：lscpu [-选项]**

```
#查看/proc/cpuinfo文件内容
[root@localhost ~]# cat /proc/cpuinfo
processor    ：#系统中逻辑处理核的编号。对于单核处理器，则可认为是其CPU编号，对于多核处理则可以是物理核、或者使用超线程技术虚拟的逻辑核
vendor_id    ：    #CPU制造商
cpu family   ：   #CPU产品系列代号
model        ：    #CPU属于其系列中的哪一代的代号
model name:     #CPU属于的名字及其编号、标称主频
stepping     ：    #CPU属于制作更新版本
cpu MHz      ：    #CPU的实际使用主频
cache size   ：   #CPU二级缓存大小
physical id  ：  #单个CPU的标号
siblings     ：#单个CPU逻辑物理核数
core id      ：#当前物理核在其所处CPU中的编号，这个编号不一定连续
cpu cores    ：   #该逻辑核所处CPU的物理核数
apicid       ：#用来区分不同逻辑核的编号，系统中每个逻辑核的此编号必然不同，此编号不一定续
fpu          ：  #是否具有浮点运算单元（Floating Point Unit）
fpu_exception  ：   #是否支持浮点计算异常
cpuid level   ：    #执行cpuid指令前，eax寄存器中的值，根据不同的值cpuid指令会返回不同的内容
wp           ：   #表明当前CPU是否在内核态支持对用户空间的写保护（Write Protection）
flags        ：    #当前CPU支持的功能
bogomips   ：       #在系统内核启动时粗略测算的CPU速度（Million Instructions Per Second）
clflush size  ：     #每次刷新缓存的大小单位
cache_alignment：   #缓存地址对齐单位
address sizes    ：#可访问地址空间位数
power management：  #对能源管理的支持，有以下几个可选支持功能：

#使用lscpu查看cpu信息
[root@localhost ~]# lscpu
  Architecture: #架构
    CPU(s): #逻辑cpu颗数
    Thread(s) per core: #每个核心线程
    Core(s) per socket: #每个cpu插槽核数/每颗物理cpu核数
    CPU socket(s): #cpu插槽数
    Vendor ID: #cpu厂商ID
```

CPU family: #cpu系列
Model: #型号
Stepping: #步进
CPU MHz: #cpu主频
Virtualization: #cpu支持的虚拟化技术
L1d cache: #一级缓存（google了下，这具体表示表示cpu的L1数据缓存）
L1i cache: #一级缓存（具体为L1指令缓存）
L2 cache: #二级缓存

查看系统内存信息

- **/proc/meminfo文件用于存放系统内存信息**
- **free 用于查看内存使用情况**
- **命令格式：free [-选项]**
- **常用选项：-h　#以人类易读方式显示大小（KB，MB，GB）**

```
#查看/proc/meminfo文件内容
[root@localhost ~]# cat /proc/meminfo
MemTotal:         995896 kB    #所有可用的内存大小，物理内存减去预留位和内核使用。系统从加开始到引导完成，firmware/BIOS要预留一些内存，内核本身要占用一些内存，最后剩下可供内核支的内存就是MemTotal。这个值在系统运行期间一般是固定不变的，重启会改变。
MemFree:          244196 kB    #表示系统尚未使用的内存。
MemAvailable:     435080 kB    #真正的系统可用内存，系统中有些内存虽然已被使用但是可以回的，比如cache/buffer、slab都有一部分可以回收，所以这部分可回收的内存加上MemFree才是系可用的内存
Buffers:            2132 kB    #用来给块设备做缓存的内存，(文件系统的 metadata、pages)
Cached:           314632 kB    #分配给文件缓冲区的内存,例如vi一个文件，就会将未保存的内容写到缓冲区
SwapCached:            0 kB    #被高速缓冲存储用的交换空间（硬盘的swap）的大小
Active:           295908 kB    #经常使用的高速缓冲存储器页面文件大小
Inactive:         271552 kB    #不经常使用的高速缓冲存储器文件大小
Active(anon):     251528 kB    #活跃的匿名内存
Inactive(anon):    13044 kB    #不活跃的匿名内存
Active(file):      44380 kB    #活跃的文件使用内存
Inactive(file):   258508 kB    #不活跃的文件使用内存
Unevictable:           0 kB    #不能被释放的内存页
Mlocked:               0 kB    #系统调用 mlock 家族允许程序在物理内存上锁住它的部分或全部地址空。这将阻止Linux 将这个内存页调度到交换空间（swap space），即使该程序已有一段时间没有访问段空间
SwapTotal:             0 kB    #交换空间总内存
SwapFree:              0 kB    #交换空间空闲内存
Dirty:                 4 kB    #等待被写回到磁盘的
Writeback:             0 kB    #正在被写回的
AnonPages:         15100 kB    #未映射页的内存/映射到用户空间的非文件页表大小
Mapped:             7160 kB    #映射文件内存
Shmem:               100 kB    #已经被分配的共享内存
Slab:               9236 kB    #内核数据结构缓存
SReclaimable:       2316 kB    #可收回slab内存
SUnreclaim:         6920 kB    #不可收回slab内存
KernelStack:        2408 kB    #内核消耗的内存
PageTables:         1268 kB    #管理内存分页的索引表的大小
NFS_Unstable:          0 kB    #不稳定页表的大小
```

```
Bounce:                0 kB    #在低端内存中分配一个临时buffer作为跳转，把位于高端内存的缓存数据
制到此处消耗的内存
WritebackTmp:          0 kB    #FUSE用于临时写回缓冲区的内存
CommitLimit:       22980 kB    #系统实际可分配内存
Committed_AS:     536244 kB    #系统当前已分配的内存
VmallocTotal:     892928 kB    #预留的虚拟内存总量
VmallocUsed:       29064 kB    #已经被使用的虚拟内存
VmallocChunk:     860156 kB    #可分配的最大的逻辑连续的虚拟内存

#使用free命令查看内存使用情况
[root@localhost ~]# free -h
              total        used        free      shared  buff/cache   available
Mem:           972M        344M        238M         13M        389M        424M
Swap:          2.0G          0B        2.0G
#解释：Mem 物理内存统计信息
total:          #物理内存总量
used:           #以使用的内存总量
free:          #空闲内存总量
shared:          #共享内存总量
buff/cache:      #块设备与普通文件占用的缓存数量
available:      #还可以被应用程序使用的物理内存大小

#解释：Swap 内存交换空间，当物理内存不足时，可以使用硬盘空间充当内存使用
total:          #交换分区内存总量
used:           #正在使用的交换分区内存
free:          #空闲交换分区内存
```

查看网卡信息

- **网卡配置文件地址： /etc/sysconfig/network-scripts/网卡名**
- **ifconfig 用于显示和设置网卡的参数**
- **命令格式： ifconfig [网卡名]**

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens32
TYPE=“Ethernet“           #网卡类型=以太 ※
PROXY_METHOD=“none“       #代理方式=关闭
BROWSER_ONLY="no“         #只是浏览器=否
BOOTPROTO=“none“          #获取IP地址的方式=固定IP ※
DEFROUTE=“yes“            #是否设置默认路由=是
IPV4_FAILURE_FATAL=“no“   #是否开启ipv4致命检测=否（如果ipv4配置失败禁用设备）
NAME=“ens32“              #物理网卡设备名字 ※
UUID=“3ef0d258-f9a4-49e5-a9da-7b47bc98daa0     “#网卡UUID
DEVICE=“ens32“            #网卡名字 ※
ONBOOT=“yes“              #开机或重启时是否启动网卡 ※
IPADDR=“192.168.0.210“    #IP地址 ※
PREFIX=“24“               #子网掩码 ※
GATEWAY=“192.168.0.254“   #网关   ※
DNS1=“8.8.8.8“            #dns服务器IP地址 ※
DNS2=8.8.4.4              #备用dns服务器IP地址 ※

#使用ifconfig命令查看网卡信息
[root@localhost ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

```
        inet 192.168.0.29  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::8d50:c4d5:97b0:9d64  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b0:cf:c8  txqueuelen 1000  (Ethernet)
        RX packets 3948  bytes 1811465 (1.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2538  bytes 459113 (448.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
#解释：
ens32:        #网卡名称 ※
flags=4163：     #标志
UP：            #网卡处于活跃状态 ※
BROADCAST：     #支持广播
RUNNING：     #网线已接入
MULTICAST：     #支持组播
mtu 1500：     #最大传输单元（字节），表示此网卡一次能传输的最大数据包 ※
inet 192.168.0.29        #IPV4地址 ※
netmask 255.255.255.0    #子网掩码 ※
broadcast 192.168.0.255    #广播地址 ※
inet6 fe80::8d50:c4d5:97b0:9d64        #IPV6地址
prefixlen 64  scopeid 0x20<link>    #前缀 64 作用域 0x20
ether 00:0c:29:b0:cf:c8    #网卡MAC地址 ※
xqueuelen 1000           #网卡设置的传送队列长度
(Ethernet)              #网卡连接类型
RX packets 3948           #接收正确的数据包 ※
bytes 1811465 (1.7 MiB)    #接收的数据量与字节 ※
RX errors 0  dropped 0  overruns 0  frame 0 #接收到的错误包、丢弃的数据包数、由于速度过快丢失的数据包、发生frame错误而丢失的数据包数 ※
TX packets 100         #发送的正确的数据包数 ※
bytes 8116 (7.9 KiB)#发送的数据量、字节 ※
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0 #发送时产生错误的数据包数、丢弃的据包数、由于速度过快而丢失的数据包数、发生carrier错误而丢失的数据包数、冲突信息包的数目 ※

#只查看指定的网卡
[root@localhost ~]# ifconfig ens32


lo: 本地回环网卡，不是物理网卡，通过软件虚拟出来的一个网卡，127.0.0.1，用于测试本机的联通性
[root@localhost ~]# ping 127.0.0.1

virbr0: 虚拟化的网络接口，通过软件技术虚拟出来的一个网卡，192.168.122.1，KVM虚拟化技术的候
```

查看主机名及修改主机名

- **/etc/hostname文件用于存放主机名**
- **hostname 命令用于显示和设置主机名**
- **命令格式：hostname [新名称]**

```
#查看主机名
[root@localhost ~]# hostname
localhost.localdomain

#查看主机名配置文件
```

```
[root@localhost ~]# cat /etc/hostname
localhost.localdomain

#临时修改主机名（立刻生效，服务器重启以后失效）
[root@localhost ~]# hostname test
[root@localhost ~]# hostname
test

#exit/loguot登出系统
[root@localhost ~]# exit
[c:\~]$ ssh 192.168.0.50
[root@test ~]#

[root@test ~]# hostname fhsd.jhglshdjkghjkdfhgkjhgdsahgjklhdsfjghsdhgjlhsd
[root@test ~]# logout

[root@fhsd ~]# hostname sdhjghsdfjkhgkjdfshkgljhsdjfhgjksdhgjkhsdjgjkl
[root@fhsd ~]# exit

#命令行永久修改主机名（立刻生效，不需要重启系统）
[root@localhost ~]# hostnamectl set-hostname test
[root@localhost ~]# exit
```

vi/vim文本编辑器

- **Vim是从 vi 发展出来的一个文本编辑器，vim 具有程序编辑的能力，可以主动的以字体颜色辨别法的正确性**
- **vi/vim 共分为三种模式：命令模式、输入模式、底线命令模式（末行模式）**
  - **命令模式：刚刚启动 vi/vim，便进入了命令模式**
  - **输入模式：在命令模式下按 a/i/o 就进入了输入模式**
  - **ESC，退出输入模式，切换到命令模式**
  - **底线命令模式：在命令模式下按下:（英文冒号）就进入了底线命令模式**
- **命令格式：vim 文件名**
  - **若目标文件不存在，则新创建文件并编辑**
  - **若目标文件以存在，则打开文件并编辑**
- **命令模式：刚刚启动 vi/vim，便进入了命令模式**
  - **i　切换到输入模式，在当前光标所在字符前插入**
  - **a　切换到输入模式，在当前光标所在字符后插入**
  - **o　切换到输入模式，在当前光标所在行下插入新行**
  - **:　切换到底线命令模式，以在最底一行输入命令**
  - **x　在命令模式下删除当前光标所在的单字符**
  - **dd　删除一整行内容，配合数字可删除指定范围内的行**
  - **C　删除当前光标及光标后所有内容并进入输入模式**
  - **u　恢复上一次修改内容，一次恢复一个操作，可多次恢复，直到恢复本次操作初始状态为止**

- **$　将光标移动至行尾**
  - **0（零）　将光标移动至行首**
  - **gg　跳转至文件第一行**
  - **G　跳转至文件最后一行**
  - **yy　复制当前行，配合数字可以同时复制多行**
  - **p　粘贴当前光标所在行下**
  - **/关键字　搜索文件内关键字，n从上向下快速定位关键字，N从下向上快速定位关键字**
- **底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。**
  - **:w　保存**
  - **:q　退出**
  - **:wq　保存并退出**
  - **:x　保存并退出**
  - **ZZ　保存并退出**
  - **:q!　强制退出不保存**
  - **:wq!　强制保存并退出，适用于只读文件（没有写权限）**
  - **:set nu　以行号形式显示文件内容**
  - **:set nonu 取消行号显示**
  - **:行号　快速跳转到指定行**
  - **:%s　替换文件内容，g替换全文，默认只替换每一行匹配到的第一个关键字（数字s 指定换的行）**
  - **:nohl　取消高亮显示**

```
[root@test ~]# vim /etc/services
```

修改网卡IP地址

- **网卡配置文件地址：　/etc/sysconfig/network-scripts/网卡名**
- **ifconfig　****　****　　#用于显示和设置网卡的参数**
- **systemctl restart network　#重启网络**
- **ifup 网卡名　　#启动该网卡设备**
- **ifdown 网卡名　#禁用该网卡设备**

```
#修改IP地址
[root@test ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens32
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="none"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
```

```
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens32"
UUID="16085f4c-f690-4058-b29e-d55c73387026"
DEVICE="ens32"
ONBOOT="yes"
IPADDR="192.168.0.60"  #修改IP地址
PREFIX="24"
GATEWAY="192.168.0.254"
DNS1="114.114.114.114"
IPV6_PRIVACY="no"
~
#重启网络（IP地址发生改变，当前终端会断开）
[root@test ~]# systemctl restart network
[c:\~]$ ssh 192.168.0.60

#关闭网卡并激活网卡
[root@localhost ~]# ifdown ens32;ifup ens32

#查看所有网卡信息
[root@test ~]# ip a
```

- **使用命令修改网卡IP地址**

nmcli connection modify 网卡名 ipv4.method manual ipv4.addresses Ip地址/掩码 connection.autoconnect yes

解释：2

**nmcli connection modify（修改） **

网卡名 ipv4.method（配置ipv4地址方法）

manual （手动配置）

**ipv4.addresses（ipv4地址） **

Ip地址/掩码 connection.autoconnect yes（开机自动连接）

- **激活网卡：nmcli connection up 网卡名**
- **关闭网卡：nmcli connection down 网卡名**
- **重启网卡：nmcli connection reload 网卡名**

```
#使用命令修改网卡IPV地址
[root@test ~]# nmcli connection modify ens32 ipv4.method manual ipv4.addresses 192.168.0
50/24 connection.autoconnect yes

#激活网卡
[root@test ~]# nmcli connection up ens32
[c:\~]$ ssh 192.168.0.50
```

host命令

- **host用于将一个域名解析到一个IP地址**

```
[root@test ~]# host www.baidu.com
www.baidu.com has address 110.242.68.3
www.baidu.com has address 110.242.68.4
www.baidu.com is an alias for www.a.shifen.com.
www.baidu.com is an alias for www.a.shifen.com.
```

nslookup命令

- **nslookup用于查询域名解析是否正常，在网络故障时用来诊断网络问题 **

```
[root@test ~]# nslookup www.baidu.com
Server:         114.114.114.114
Address:    114.114.114.114#53

Non-authoritative answer:
Name:    www.baidu.com
Address: 110.242.68.4
Name:    www.baidu.com
Address: 110.242.68.3
```

alias别名管理

- **alias命令用于设置命令别名，用户可以使用alias自定义命令别名来简化命令的复杂度**
- **.bashrc 文件存放命令别名**
- **命令格式：aliasi [别名]=[命令]　　#注意事项：等号（=）前后不能有空格****
- **unalias 别名　　#取消别名**

```
#定义别名
[root@test ~]# alias lsnet='ls /etc/sysconfig/network-scripts/'
[root@test ~]# lsnet
[root@test ~]# alias myls='ls -ldh'
[root@test ~]# myls /opt

#查看当前系统可用命令别名
[root@test ~]# alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias lsnet='ls /etc/sysconfig/network-scripts/'
alias mv='mv -i'
alias myls='ls -ldh'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

```
#两条命令效果相同
[root@test ~]# ls -l hello
-rw-r--r--. 1 root root 426 3月  28 15:00 hello
[root@test ~]# ll hello
-rw-r--r--. 1 root root 426 3月  28 15:00 hello

[root@test ~]# which ls
alias ls='ls --color=auto'
    /usr/sbin/ls
[root@test ~]# /usr/sbin/ls
[root@test ~]# ls

#取消本次命令的别名功能“\”
[root@test ~]# \ls

#取消命令别名
[root@test ~]# unalias myls
[root@test ~]# myls
bash: myls: 未找到命令...

#定义别名不要跟系统命令发生冲突
[root@test ~]# alias ls=hostname
[root@test ~]# ls
test

#取消命令别名
[root@test ~]# unalias ls
[root@test ~]# alias

#重新定义别名
[root@test ~]# alias ls='ls --color=auto'
[root@test ~]# ls
```

history 管理历史

- **history命令用于显示历史记录和执行过的命令，登录shell时会读取~./bash_history历史文件中录下的命令，当退出或登出shell时，会自动保存到历史命令文件，该命令单独使用时，仅显示历史命令**
- **历史命令默认只能存储1000条，可以通过/etc/profile文件修改**
- **命令格式：history [-选项] [参数]**
- **常用选项：**
  - **-a 追加本次新执行的命令至历史命令文件中**
  - **-d 删除历史命令中指定的命令**
  - **-c 清空历史命令列表**
- **快捷操作：**
  - **!# 调用命令历史中第N条命令**
  - **!string 调用命令历史中以strind开头的命令**
  - **!! 重复执行上一条命令**

```
#获取命令帮助
[root@test ~]# help history

#查看历史命令
[root@test ~]# history

#查看记录历史命令文件
[root@test ~]# cat .bash_history

#将历史命令同步至历史命令配置文件中
[root@test ~]# history -a
[root@test ~]# cat .bash_history

#删除历史命令中655条命令历史
[root@test ~]# history -d 655
[root@test ~]# history -d 637

#清空缓存中所有历史命令
[root@test ~]# history -c
[root@test ~]# history
    1  history

#删除历史命令配置文件（该文件删除后系统会再次自动创建）
[root@test ~]# rm -rf .bash_history

#快速调用历史命令中第1条
[root@test ~]# !1
[root@test ~]# !3

#调用历史命令中以cat开头的命令（只调用最近使用的cat历史命令）
[root@test ~]# !cat

#重复执行上一条命令
[root@test ~]# !!

#历史命令默认只能记录1000条，可以通过/etc/profile文件修改
[root@test ~]# vim /etc/profile
...
46 HISTSIZE=100
```

date日期时间管理

- **date命令用于显示或设置系统日期与时间**
- **命令格式：date [-选项] [+格式符]　　#查看系统日期时间**
- **命令格式：date [-选项]　　#设置日期时间**
- **常用选项：-s 设置日期时间**
- **格式符：**
  - **+%Y 年份**
  - **+%B 月份**
  - **+%d 日**

- **+%H 时**
  - **+%M 分**
  - **+%S 秒**
  - **+%F 年-月-日**
  - **+%X 时：分：秒**

```
#显示系统日期与时间
[root@test ~]# date
2021年 03月 28日 星期日 17:08:34 CST

#只显示年分
[root@test ~]# date +%Y
2021

#只显示月份
[root@test ~]# date +%B
三月

#只显示几号
[root@test ~]# date +%d
28

#只显示小时
[root@test ~]# date +%H
17

#只显示分钟
[root@test ~]# date +%M
10

#只显示秒
[root@test ~]# date +%S
24

#显示年月日
[root@test ~]# date +%F
2021-03-28

#显示时分秒
[root@test ~]# date +%X
17时12分10秒

#显示年月日时分秒
[root@test ~]# date +%F%X
2021-03-2817时12分39秒

#可以自定义分隔符“-”
[root@test ~]# date +%F-%X
2021-03-28-17时13分38秒

[root@test ~]# date +%F:%X
2021-03-28:17时13分55秒
```

```
#修改系统年月日
[root@test ~]# date -s 2020-03-28
2020年 03月 28日 星期六 00:00:00 CST

#修改系统时分秒
[root@test ~]# date -s 17:16:00
2020年 03月 28日 星期六 17:16:00 CST

#修改年月日时分秒
[root@test ~]# date -s '2021-03-28 17:17:00'
2021年 03月 28日 星期日 17:17:00 CST
#解释:
''单引号:引用整体,屏蔽特殊符号的功能
""双引号:引用整体,不会屏蔽特殊符号的功能

#Linux的两种时钟
系统时钟:内核通过CPU的工作频率去计算的时间
硬件时钟:

#显示硬件时间
[root@test ~]# clock
2021年03月28日 星期日 17时23分42秒  -0.945549 秒

#显示并同步系统与硬件时钟
[root@test ~]# man hwclock
-s:把系统时间设置成与硬件时间相同
-w:把硬件时间设置成与系统时间相同
[root@test ~]# hwclock -w
[root@test ~]# date
2021年 03月 28日 星期日 17:27:18 CST

#cal显示日历
[root@test ~]# cal
      三月 2021
日 一 二 三 四 五 六
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

#显示指定的全年月份
[root@test ~]# cal 2021
```

wc统计命令

- **wc 用于统计文件的字节数、行数,并将统计的结果输出到屏幕**
- **命令格式:wc [-选项] 文件名**
- **常用选项:**
  - **-c #统计字节数**

- **-l #统计行数**

```
[root@test ~]# wc /etc/passwd
43   87 2259 /etc/passwd
行数 单词 字节  文件名

#统计文件字节数
[root@test ~]# wc -c /etc/passwd
2259 /etc/passwd

#统计文件行数
[root@test ~]# wc -l /etc/passwd
43 /etc/passwd

[root@test ~]# wc -l /etc/fstab
11 /etc/fstab
```

管道符

- **管道符“|”：将命令的输出结果交给另外一条命令作为参数继续处理**

```
[root@test ~]# head -10 /etc/passwd |tail -5

[root@test ~]# head -10 /etc/passwd |tail -5 |wc -l
5

root@test ~]# cat -n /etc/passwd |head -10|tail -5
     6    sync:x:5:0:sync:/sbin:/bin/sync
     7    shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8    halt:x:7:0:halt:/sbin:/sbin/halt
     9    mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10    operator:x:11:0:operator:/root:/sbin/nologin

[root@test ~]# ifconfig ens32 |head -2
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.50  netmask 255.255.255.0  broadcast 192.168.0.255
```

重定向操作

- **重定向操作：将前面命令的输出结果，写入到其他的文本文件中**
- **重定向的表示符号**
  -

** #重定向输出（覆盖）**

  -

> #重定向输出（追加）

  - **< #输入重定向（覆盖）**
  - **<< #输入重定向（追加）**

-

** 只收集正确的输出结果**

- **2> 只收集错误的输出结果（覆盖）**
- **2>> 只收集错误的输出结果（追加）**
- **&> 正确错误都收集（覆盖）**
- **&>> 正确错误都收集（追加）**

```
#将命令的输出结果以覆盖的方式重定向到文件中，（>附带创建文件功能）
[root@test ~]# ifconfig ens32 |head -2 > /opt/ens32.bak
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.50  netmask 255.255.255.0  broadcast 192.168.0.255

[root@test ~]# cat /etc/hostname > /opt/ens32.bak
[root@test ~]# cat /opt/ens32.bak
test

[root@test ~]# free -h > /opt/free.bak
[root@test ~]# cat /opt/free.bak
              total        used        free      shared  buff/cache   available
Mem:           972M        414M        123M         15M        435M        336M
Swap:          2.0G          0B        2.0G

#将命令的输出结果以追加的方式重定向到文件中
[root@test ~]# cat /etc/hostname >> /opt/free.bak
[root@test ~]# cat /opt/free.bak

#“>”只收集正确的输出结果，不收集错误的输出结果
[root@test ~]# ls xxooooxx > /opt/xx.txt
ls: 无法访问xxooooxx: 没有那个文件或目录

#“2>”只收集错误的输出结果，不收集正确的输出结果
[root@test ~]# ls xxooooxx 2> /opt/xx.txt
[root@test ~]# cat /opt/xx.txt
ls: 无法访问xxooooxx: 没有那个文件或目录

#“2>”以覆盖的方式将输出结果重定向到文件中
[root@test ~]# cat /etc/abc 2> /opt/ens32.bak
[root@test ~]# cat /opt/ens32.bak
cat: /etc/abc: 没有那个文件或目录

#“2>>”以追加的方式将输出结果重定向到文件中
[root@test ~]# ls /etc/abcd 2>> /opt/ens32.bak
[root@test ~]# cat /opt/ens32.bak
cat: /etc/abc: 没有那个文件或目录
ls: 无法访问/etc/abcd: 没有那个文件或目录

#“&>”以覆盖的方式将正确输出与错误输出重定向到文件中
[root@test ~]# lscat &> /opt/abc.txt
[root@test ~]# cat /opt/abc.txt

[root@test ~]# ls /etc/passwd &> /opt/pass.bak
```

```
[root@test ~]# cat /opt/pass.bak

[root@test ~]# free -h &> /opt/pass.bak
[root@test ~]# cat /opt/pass.bak

#“&>”以追加的方式将正确输出与错误输出重定向到文件中
[root@test ~]# ifconfig ens32 | head -2 &>> /opt/pass.bak
[root@test ~]# cat /opt/pass.bak

#以覆盖方式将正确输出与错误输出重定向到不同文件中
[root@test ~]# ll -d /root/  bcd >a.txt 2>b.txt
[root@test ~]# cat a.txt
dr-xr-x---. 24 root root 4096 3月  28 18:07 /root/
[root@test ~]# cat b.txt
ls: 无法访问bcd: 没有那个文件或目录
```

echo命令与sleep命令

- **echo命令用于输出指定的字符串和变量**
- **命令格式：echo [-选项] [参数]**

```
[root@test ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@test ~]# echo xxoo
xxoo

[root@test ~]# echo abc
abc

[root@test ~]# echo 男人好难
男人好难

[root@test ~]# echo 123
123

[root@test ~]# cat /etc/hostname
test

[root@test ~]# echo localhost > /etc/hostname
[root@test ~]# cat /etc/hostname
localhost
```

- **sleep命令可以用来将目前动作延迟一段时间**
- **命令格式：sleep 时间**
- **常用选项： s 秒 m 分钟 h 小时 d 日**

```
[root@test ~]# sleep 3
```

课后练习

1.查看当前系统内核名称及版本信息

uname -sr

2.请写系统存放cpu配置文件

/proc/cpuinfo

3.请写出查看cpu信息命令

cat /proc/cpuinfo

lscpu

4.请写出系统存放内存配置文件

/proc/meminfo

5.请写出查看内存命令（以人类易读方式显示）

free -h

6.请写出系统存放网卡配置文件路径

/etc/sysconfig/network-scripts/

7.请写出查看网卡配置信息命令

ifconfig（如果系统最小化安装，需要安装net-tools）

ip a （ip address）

8.请写出系统存放主机名配置文件

/etc/hostname

9.请写出查看主机名命令

cat /etc/hostname

hostname

10.将主机名修改为student（永久修改）

hostnamectl set-hostname student

vim /etc/hostname

echo student > /etc/hostname

11.请写出vim的三种模式

命令模式

输入模式

底线命令模式（末行模式）

12.将/etc/passwd文件复制到/opt目录，使用vim打开文件并显示行号

cp /etc/passwd /opt

vim /opt/passwd

:set nu

13.使用vim在/opt/passwd文件中搜索包含root关键字的行

/root

14.使用vim在/opt/passwd文件中将光标快速跳转到第10行，并将光标跳转到行尾

:10 $

15.使用vim在/opt/passwd文件中快速跳转到文件最后一行并删除，在将光标跳转到文件第一行，刚刚删除的行复制到文件第二行

G dd p

16.使用vim将/etc/hostname文件内容读入到/opt/passwd文件最后一行下

:r /etc/hostname

17.使用vim在/opt/passwd文件中复制前5行内容并粘贴到文件最后一行下

5yy p

18.将本次vim的修改恢复至初始状态，并保存退出

**u **

:wq

19.将本机IP地址修改为192.168.0.100，并重启动网卡

vim /etc/sysconfig/network-scripts/ifcfg-ens32

systemctl restart network

20.如何获取一个域名所对应的IP地址

**host **www.baidu.com

21.如何检测本机使用的DNS是否可用

**nslookup **www.jd.com

22.请将hostname命令设置别名为hn（临时设置）

alias hn=hostname

23.取消hostname命令别名

unalias hn

24.如何查看本机历史命令

history

25.执行命令历史中第20条命令

!20

26.删除命令历史中第5条命令

history -d 5

27.清空所有历史命令

history -c

rm -rf .bash_history

28.查看本机当前系统日期与时间

date

29.将本机日期时间设置与你当前时间一致

date -s '2021-04-10 14:32:00'

30.统计/etc/passwd文件行数，并将命令输出结果重定向至/opt/pass.bak文件中

wc -l /etc/passwd > /opt/pass.bak

31.显示/etc/passwd文件末尾10行的前5行内容，并将输出结果追加至/opt/pass.bak文件中

** tail -10 /etc/passwd | head -5 > /opt/pass.bak**

** cat -n /etc/passwd | tail -10 | head -5 >> /opt/pass.bak **

用户账号管理

- **用户账号的作用：用户账号可用来登录系统，可以实现访问控制**
- **用户模板目录：/etc/skel/**

```
[root@localhost ~]# ls -a /etc/skel/
.  ..  .bash_logout  .bash_profile  .bashrc  .mozilla

[root@localhost ~]# cd /etc/skel/
[root@localhost skel]# vim prompt
```

useradd创建用户

- **useradd 命令用于创建新的用户**
- **命令格式：useradd [-选项] 用户名**

- **常用选项：**
  - **-u 指定用户UID**
  - **-d 指定用户家目录（了解）**
  - **-c 用户描述信息**
  - **-g 指定用户基本组（了解）**
  - **-G 指定用户附加组**
  - **-s 指定用户的解释器程序**

```
[root@localhost ~]# useradd user1

#创建用户并指定用户的UID
[root@localhost ~]# useradd -u 1100 user2

#创建用户并指定用户的家目录
root@localhost ~]# useradd -d /opt/user3 user3

#创建用户并指定UID与用户描述信息
[root@localhost ~]# useradd -u 1400  -c yunwei user4

#创建test组
[root@localhost ~]# groupadd test

#创建用户指定用户UID、描述信息、基本组
[root@localhost ~]# useradd -u 1500 -c xxoo@163.com -g test user5
[root@localhost ~]# id user5

#创建用户指定用户UID、描述信息、附加组
[root@localhost ~]# useradd -u 1600 -c yunwei -G test xiaozhang
[root@localhost ~]# id xiaozhang
uid=1600(xiaozhang) gid=1600(xiaozhang) 组=1600(xiaozhang),1401(test)

#/sbin/nologin ： 禁止用户登录系统
[root@localhost ~]# useradd -u 1800 -c test -s /sbin/nologin user8
user8:x:1800:1800:test:/home/user8:/sbin/nologin
```

id命令

- **id 命令用于查看系统用户和用户所在组的信息**
- **命令格式：id [-选项] [用户名]**

```
[root@localhost ~]# id user1
uid=1001(user1) gid=1001(user1) 组=1001(user1)
```

/etc/passwd用户信息文件

用户的基本信息存放在/etc/passwd文件

```
[root@localhost ~]# vim /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

#每个字段含义解释：用户名:密码占位符:UID:基本组GID:用户描述信息:家目录:解释器程序
UID：0 超级用户
UID：1-999 系统伪用户，不能登录系统
UID：1000-65535 普通用户，管理员创建的用户

组：

基本组（初始组）：一个用户只允许有一个基本组

附加组（在基本组之外组）：一个用户可以允许有多个附加组

用户--->shell程序--->内核--->硬件

/etc/default/useradd文件

/etc/default/useradd 存放用户默认值信息

[root@localhost ~]# vim /etc/default/useradd
# useradd defaults file
GROUP=100 #用户默认组
HOME=/home #用户家目录
INACTIVE=-1 #密码过期宽限天数（/etc/shadow文件第7个字段）
EXPIRE= #密码失效时间（/etc/shadow文件第8个字段）
SHELL=/bin/bash #默认使用的
SKEL=/etc/skel #模板目录
CREATE_MAIL_SPOOL=yes #是否建立邮箱

/var/spool/mail/用户邮件目录

[root@localhost ~]# ls /var/spool/mail/
laowang lisi rpc user1 user2 user3 user4 user5 user8 xiaozhang

#查看邮件
[root@localhost ~]# mail

passwd设置用户密码

- **passwd命令用于设置用户密码**
- **命令格式：passwd [-选项] [用户名]**
- **密码规范：长度不能少于8个字符，复杂度（数字、字母区分大小写，特殊字符）**
- **密码规范：本次修改的密码不能和上次修改的密码太相近 **
- **常用选项**
  - **-S 查看密码信息**
  - **-l 锁定用户密码**
  - **-u 解锁用户密码**
  - **-d 删除密码**
  - **--stdin 通过管道方式设置用户密码**

- **非交互设置用户密码**
- **命令格式：echo '密码' | passwd --stdin 用户名 **

```
#设置用户密码
[root@localhost ~]# passwd user1
更改用户 user1 的密码 。
新的 密码：1
无效的密码： 密码是一个回文
重新输入新的 密码：1
passwd：所有的身份验证令牌已经成功更新。

#使用user1用户登录系统
[user1@localhost ~]$ ls
prompt
[user1@localhost ~]$ cat prompt
不允许随便修改系统xx文件!
有问题可联系管理员邮箱：xxoo@163.com

#查看用户密码信息
[root@localhost ~]# passwd -S user1

#锁定用户当密码
[root@localhost ~]# passwd -l user2
锁定用户 user2 的密码 。
passwd: 操作成功

[root@localhost ~]# passwd -S user2
user2 LK 2021-04-10 0 99999 7 -1 (密码已被锁定。)

#解锁用户密码
[root@localhost ~]# passwd -u user2
解锁用户 user2 的密码。
passwd: 操作成功

#删除用户密码
[root@localhost ~]# passwd -d user2
清除用户的密码 user2。
passwd: 操作成功

#非交互设置用户密码
[root@localhost ~]# echo 1 | passwd --stdin laowang
更改用户 laowang 的密码 。
passwd：所有的身份验证令牌已经成功更新。
```

/etc/shadow用户密码文件

- **用户的密码信息存放在/etc/shadow文件中，该文件默认任何人都没有任何权限（不包括root）**

```
[root@localhost ~]# vim /etc/shadow
root:$6$1ji5e8yglrZWAcI6$FONKr3qebZufQ.u0Mf/MbipzGw/MVvxS.vgXcy/duc4b/GU0U7tfe3
wPQ4XJEXstqBuwvaJqq2/kY/g/783u/::0:99999:7:::
#每个字段含义解释：
第一字段：用户名
```

第二字段：密码加密字符串，加密算法为SHA512散列加密算法，如果密码位是“*”或者“!!”表示码已过期
第三个字段：密码最后一次修改日期，日期从1970年1月1日起，每过一天时间戳加1
第四个字段：密码修改的期限，如果该字段为0表示随时可以修改密码，例如：该字段为10，代表10之内不可以修改密
第五个字段：密码有效期
第六个字段：密码到期前警告时间（和第五个字段相比）
第七个字段：密码过期后的宽限天数（和第五个字段相比）
第八个字段：账号失效时间，日期从1970年1月1日起
第九个字段：保留

#chage命令用于修改/etc/shadow文件信息，修改文件内容第三个字段（密码最后一次修改时间）
[root@localhost ~]# chage -d 0 user8

su命令

- **su命令用于切换当前用户身份到其他用户身份**
- **命令格式：su [-选项] [用户名]**

#只切换用户身份，环境没有改变
[root@localhost ~]# su user1
[user1@localhost root]$ ls
ls: 无法打开目录.: 权限不够
[user1@localhost root]$ cd
[user1@localhost ~]$ exit
exit

#切换用户身份，连同环境一起切换
[root@localhost ~]# su - user1
上一次登录：六 4月 10 16:54:40 CST 2021pts/1 上
[user1@localhost ~]$ pwd
/home/user1

#普通用户切换为root（需要输入root用户的密码）
[user1@localhost ~]$ su - root
密码：
上一次登录：六 4月 10 16:05:17 CST 2021从 192.168.0.1pts/2 上

usermod修改用户属性

- **usermod 命令用于修改已存在用户的基本信息**
- **命令格式：usermod [-选项] 用户名**
- **常用选项：**
  - **-u 修改用户UID**
  - **-d 修改用户家目录**
  - **-g 修改用户基本组**
  - **-c 修改用户描述信息**
  - **-G 添加用户附加组**

- **-s 修改用户shell**

```
#修改用户UID（用户如果以登录系统，不允许修改）
[root@localhost ~]# usermod -u 1111 user1
[root@localhost ~]# id user1
uid=1111(user1) gid=1001(user1) 组=1001(user1)

#修改用户描述信息
[root@localhost ~]# usermod -c xxoo@163.com user8

#修改用户的附加组
[root@localhost ~]# usermod -G test user8
[root@localhost ~]# id user8
uid=1800(user8) gid=1800(user8) 组=1800(user8),1401(test)

#修改用户的解释器
[root@localhost ~]# usermod -s /bin/bash user8
```

userdel删除用户

- **userdel 用于删除给定的用户以及与用户相关的文件，该命令若不加选项仅删除用户账号，不删除用户的家目录**
- **命令格式：userdel [-选项] 用户名**
- **常用选项：**
  - **-r 删除用户同时，并删除用户的家目录**

```
#删除用户，仅删除账号，不删除家目录
[root@localhost ~]# userdel user8
[root@localhost ~]# ls /home
laowang  lisi  user1  user2  user4  user5  user8  xiaozhang
[root@localhost ~]# id user8
id: user8: no such user

#删除用户，连同用户家目录一并删掉
[root@localhost ~]# userdel -r user4
[root@localhost ~]# ls /home
laowang  lisi  user1  user2  user5  user8  xiaozhang
[root@localhost ~]# id user4
id: user4: no such user
```

groupadd添加新组

- **groupadd 用于创建一个新的工作组，新组的信息将被添加到/etc/group文件中**
- **命令格式：groupadd [-选项] 组名**
- **常用选项：**
  - **-g GID　　#指定组的GID**

```
#创建组
```

```
[root@localhost ~]# groupadd -g 1555 student
[root@localhost ~]# cat /etc/group
```

/etc/group组信息文件

- **组信息存放在/etc/group文件中**

```
[root@localhost ~]# vim /etc/group
root:x:0:
#每个字段含义解释：组名:组密码占位符:GID:组中附加用户
```

/etc/gshadow组密码文件

- **组密码信息存放在/etc/gshadow文件中**

```
[root@localhost ~]# vim /etc/gshadow
root:::
#每个字段含义解释：组名:组密码:组内管理员:组中附加用户
```

groupmod修改组属性

- **groupmod 用于修改指定工作组属性**
- **命令格式：groupmod [-选项] 组名**
- **常用选项：**
  - **-g GID　　#修改组的GID**
  - **-n 新组名　#修改组名**

```
#修改组名
[root@localhost ~]# groupmod -n stugrp student

#修改组GID
root@localhost ~]# groupmod -g 1666 stugrp
```

gpasswd组管理命令

- **gpasswd 是Linux工作组文件/etc/group和/etc/gshadow管理工具，用于将用户添加到组或从中删除**
- **命令格式：gpasswd [-选项] 用户名 组名**
- **常用选项：**
  - **-a　#将用户添加到工作组**
  - **-d　#将用户从工作组中删除**

```
#创建用户
[root@localhost ~]# useradd hary
[root@localhost ~]# useradd tom
[root@localhost ~]# useradd natasha
```

```
[root@localhost ~]# useradd kenji
[root@localhost ~]# useradd jack

#讲用户加入到组
[root@localhost ~]# gpasswd -a hary stugrp
正在将用户“hary”加入到“stugrp”组中
[root@localhost ~]# gpasswd -a tom stugrp
正在将用户“tom”加入到“stugrp”组中
[root@localhost ~]# gpasswd -a kenji stugrp
正在将用户“kenji”加入到“stugrp”组中
[root@localhost ~]# gpasswd -a natasha stugrp
正在将用户“natasha”加入到“stugrp”组中
[root@localhost ~]# gpasswd -a jack stugrp
正在将用户“jack”加入到“stugrp”组中
[root@localhost ~]#

#查看组文件信息
[root@localhost ~]# cat /etc/group
stugrp:x:1666:hary,tom,kenji,natasha,jack

#将用户从组中删除
root@localhost ~]# gpasswd -d tom stugrp
[root@localhost ~]# gpasswd -d hary stugrp
正在将用户“hary”从“stugrp”组中删除
[root@localhost ~]# gpasswd -d jack stugrp
正在将用户“jack”从“stugrp”组中删除
[root@localhost ~]# gpasswd -d kenji stugrp
正在将用户“kenji”从“stugrp”组中删除
[root@localhost ~]# cat /etc/group
```

groupdel删除组

- **groupdel 用于删除指定工作组**
- **命令格式：groupdel 组名**

```
[root@localhost ~]# groupdel stugrp
```

chmod权限管理

- **chmod（英文全拼：change mode）设置用户对文件的权限**
- **命令格式：chmod [-选项] 归属关系+-=权限类别 文件...**
- **常用选项：**
  - **-R 递归修改目录下所有的子文件与子目录的权限与父目录相同**
- **归属关系：u 所有者 g 所属组 o 其他人**
- **权限类别： r 读取 w 写入 x 执行 - 没有权限 **
- **操作： + 添加权限 - 去除权限 = 重新定义权限**
- **权限数字表示：r ---- 4 w ---- 2 x ---- 1 0 没有权限**

```
#查看文件详细属性
[root@localhost ~]# ll hello
-rw-r--r--. 1 root root 426 3月  28 15:00 hello

#为文件所有者添加执行权限
[root@localhost ~]# chmod u+x hello
[root@localhost ~]# ll hello
-rwxr--r--. 1 root root 426 3月  28 15:00 hello

#为文件所属组添加写权限
[root@localhost ~]# chmod g+w hello
[root@localhost ~]# ll hello
-rwxrw-r--. 1 root root 426 3月  28 15:00 hello

#为文件其他人添加写权限
[root@localhost ~]# chmod o+w hello
[root@localhost ~]# ll hello
-rwxrw-rw-. 1 root root 426 3月  28 15:00 hello

#使用（逗号）可以同时为多个用户授权
[root@localhost ~]# chmod g+x,o+x hello
[root@localhost ~]# ll hello
-rwxrwxrwx. 1 root root 426 3月  28 15:00 hello

#去除所有者执行权限
[root@localhost ~]# chmod u-x hello
[root@localhost ~]# ll hello
-rw-rwxrwx. 1 root root 426 3月  28 15:00 hello

#去除所属组执行权限
[root@localhost ~]# chmod g-x hello
[root@localhost ~]# ll hello
-rw-rw-rwx. 1 root root 426 3月  28 15:00 hello

#去除其他人执行权限
[root@localhost ~]# chmod o-x hello
[root@localhost ~]# ll hello
-rw-rw-rw-. 1 root root 426 3月  28 15:00 hello

#同时去除ugo写权限
[root@localhost ~]# chmod u-w,g-w,o-w hello
[root@localhost ~]# ll hello
-r--r--r--. 1 root root 426 3月  28 15:00 hello

#重新定义所有者权限
[root@localhost ~]# chmod u=rwx hello
[root@localhost ~]# ll hello
-rwxr--r--. 1 root root 426 3月  28 15:00 hello

#重新定义所属组权限
[root@localhost ~]# chmod g=rwx hello
[root@localhost ~]# ll hello
-rwxrwxr--. 1 root root 426 3月  28 15:00 hello
```

```
#重新定义其他人权限
[root@localhost ~]# chmod o=rwx hello
[root@localhost ~]# ll hello
-rwxrwxrwx. 1 root root 426 3月  28 15:00 hello

#创建目录并设置目录权限
[root@localhost ~]# mkdir /test
[root@localhost ~]# ll -d /test
drwxr-xr-x. 2 root root 6 4月  11 14:30 /test

#为目录所属组添加写权限
[root@localhost ~]# chmod g+w /test
[root@localhost ~]# ll -d /test
drwxrwxr-x. 2 root root 6 4月  11 14:30 /test

#为目录其他人添加写权限
[root@localhost ~]# chmod o+w /test
[root@localhost ~]# ll -d /test
drwxrwxrwx. 2 root root 6 4月  11 14:30 /test
[root@localhost ~]#

#重新定义所有用户权限
[root@localhost ~]# chmod u=rwx,g=rx,o=rx /test
[root@localhost ~]# ll -d /test
drwxr-xr-x. 2 root root 6 4月  11 14:30 /test

#同时为所有用户定义相同权限
[root@localhost ~]# chmod ugo=rwx /test
[root@localhost ~]# ll -d /test
drwxrwxrwx. 2 root root 21 4月  11 14:37 /test

#权限数字定义方式
[root@localhost ~]# ll hello
-rwxrwxrwx. 1 root root 426 3月  28 15:00 hello
所有者：rwx   4+2+1=7
所属组：r     4
其他人：r     4
[root@localhost ~]# chmod 744 hello
[root@localhost ~]# ll hello
-rwxr--r--. 1 root root 426 3月  28 15:00 hello

所有者：rw 4+2=6
所属组：rw 4+2=6
其他人：--- 0
[root@localhost ~]# chmod 660 hello
[root@localhost ~]# ll hello
-rw-rw----. 1 root root 426 3月  28 15:00 hello

所有者：rwx 4+2+1=7
所属组：wx  2+1=3
其他人：--- 0
[root@localhost ~]# touch /hello.txt
[root@localhost ~]# ll /hello.txt
-rw-r--r--. 1 root root 0 4月  11 14:45 /hello.txt
```

```
[root@localhost ~]# chmod 730 /hello.txt
[root@localhost ~]# ll /hello.txt
-rwx-wx---. 1 root root 0 4月  11 14:45 /hello.txt

#去除所有用户权限
[root@localhost ~]# chmod 000 /hello.txt
[root@localhost ~]# ll /hello.txt
----------. 1 root student 0 4月  11 14:45 /hello.txt

#递归修改目录下所有子文件与子目录权限
[root@localhost ~]# ll -d /test
drwxrwxrwx. 2 root root 21 4月  11 14:37 /test

[root@localhost ~]# mkdir /test/xxoo
[root@localhost ~]# ll -d /test/xxoo/
drwxr-xr-x. 2 root root 6 4月  11 14:54 /test/xxoo/

[root@localhost ~]# ll /test/abc.txt
-rw-r--r--. 1 root root 0 4月  11 14:37 /test/abc.txt
#默认用户在该目录下创建文件权限与父目录不一致

#递归修改目录下所有子文件与子目录权限
[root@localhost ~]# chmod -R 777 /test
[root@localhost ~]# ll /test/abc.txt
-rwxrwxrwx. 1 root root 0 4月  11 14:37 /test/abc.txt
[root@localhost ~]# ll -d /test/xxoo
drwxrwxrwx. 2 root root 6 4月  11 14:54 /test/xxoo

#深入理解权限，
[root@localhost ~]# mkdir /test1
[root@localhost ~]# chmod 777 /test1
[root@localhost ~]# ll -d /test1
drwxrwxrwx. 2 root root 6 4月  11 14:57 /test1

#在该目录下创建文件与目录
[root@localhost ~]# touch /test1/root.txt
[root@localhost ~]# mkdir /test1/rootbak
[root@localhost ~]# chmod o=rx /test1
[root@localhost ~]# ll -d /test1
drwxrwxr-x. 2 root root 6 4月  11 14:59 /test1
[root@localhost ~]# touch /test1/root.txt

#普通用户对该目录如果拥有rwx权限是可以删除该目录下任何用户创建的文件（包括root）
[user1@localhost ~]$ cd /test1
[user1@localhost test1]$ ls
root.txt
[user1@localhost test1]$ ll root.txt
-rw-r--r--. 1 root root 0 4月  11 14:57 root.txt
[user1@localhost test1]$ rm -rf root.txt
[user1@localhost test1]$ ls
rootbak
[user1@localhost test1]$ rm -rf rootbak/
[user1@localhost test1]$ ls
[user1@localhost test1]$ ll -d /test1
```

drwxrwxrwx. 2 root root 6 4月 11 14:59 /test1

总结：
1.用户对文件拥有rwx权限
r：查看文件内容
w：对文件内容拥有增删改权限，并不能删除文件，删除文件取决于对文件的父目录有没有rwx权限
x：执行文件写权限可以增加/修改/删除文件里内容

2.用户对目录拥有rwx权限
r：查看目录下内容
w：在该目录创建文件，修改文件属性，删除任何用户的文件（包括root）
x：可以切换到该目录

umask预设权限

- **umask用于显示或设置创建文件的权限掩码**
- **命令格式：umask [-p] [-S] [mode]**

```
root@localhost ~]# mkdir /test2
[root@localhost ~]# ll -d /test2
drwxr-xr-x. 2 root root 6 4月  11 15:05 /test2
[root@localhost ~]# umask --help
umask: 用法:umask [-p] [-S] [模式]

#查看目录默认权限掩码，以数字形式显示
[root@localhost ~]# umask -p
umask 0022

#查看目录默认权限掩码，以字母形式显示
[root@localhost ~]# umask -S
u=rwx,g=rx,o=rx

#设置目录默认权限掩码，为所属组添加写权限
[root@localhost ~]# umask g+w
[root@localhost ~]# mkdir /test3
[root@localhost ~]# ll -d /test3
drwxrwxr-x. 2 root root 6 4月  11 15:09 /test3

#去除目录默认权限掩码
[root@localhost ~]# umask g-w
[root@localhost ~]# mkdir /test4
[root@localhost ~]# ll -d /test4
drwxr-xr-x. 2 root root 6 4月  11 15:10 /test4
```

chown归属关系管理

- **chown（英文全拼：change owner）用于设置文件的所有者和所属组关系**
- **命令格式：**
  - **chown [-选项] 所有者:所属组 文档　　#同时修改所有者和所属组身份**
  - **chown [-选项] 所有者 文档　　#只修改所有者身份**

- **chown [-选项] :所属组 文档　　#只修改所属组身份**
- **常用选项：**
  - **-R 递归修改**

```
#创建文件
[root@localhost ~]# chmod 744 /hello.txt
[root@localhost ~]# ll /hello.txt
-rwxr--r--. 1 root student 0 4月  11 14:45 /hello.txt

#修改文件所有者为user1用户
[root@localhost ~]# chown user1 /hello.txt
[root@localhost ~]# ll /hello.txt
-rwxr--r--. 1 user1 student 0 4月  11 14:45 /hello.txt

#修改文件所有者与所属组为lisi
[root@localhost ~]# chown lisi:lisi /hello.txt
[root@localhost ~]# ll /hello.txt
-rwxr--r--. 1 lisi lisi 4 4月  11 15:26 /hello.txt

#创建目录
[root@localhost ~]# mkdir /test5
[root@localhost ~]# ll -d /test5
drwxr-xr-x. 2 root root 6 4月  11 15:30 /test5

#修改目录所有者与所属组为lisi
[root@localhost ~]# chown lisi:lisi /test5
[root@localhost ~]# ll -d /test5
drwxr-xr-x. 2 lisi lisi 6 4月  11 15:30 /test5

[root@localhost ~]# touch /test5/root.txt
[root@localhost ~]# ll /test5/root.txt
-rw-r--r--. 1 root root 0 4月  11 15:31 /test5/root.txt

#递归修目录下所有子文件与子目录归属关系
[root@localhost ~]# chown -R lisi:lisi /test5
[root@localhost ~]# ll /test5/root.txt
-rw-r--r--. 1 lisi lisi 0 4月  11 15:31 /test5/root.txt
```

SetUID特殊权限

- **SetUID（SUID）：对于一个可执行的文件用了SUID权限后，普通用户在执行该文件后，临时拥文件所有者的身份，该权限只在程序执行过程中有效，程序执行完毕后用户恢复原有身份**
- **SetUID权限会附加在所有者的 x 权限位上，所有者的 x 权限标识会变成 s**
- **设置SetUID命令格式：chmod u+s 文件名**

```
#搜索命令绝对路径
[root@localhost ~]# which passwd
/usr/bin/passwd
[root@localhost ~]# ll /usr/bin/passwd
-rwsr-xr-x. 1 root root 27832 6月  10 2014 /usr/bin/passwd
```

```
[root@localhost ~]# which cat
/usr/bin/cat
[root@localhost ~]# ll /usr/bin/cat
-rwxr-xr-x. 1 root root 54160 10月 31 2018 /usr/bin/cat

#普通用户使用cat命令是默认无法查看/etc/shadow文件内容
[lisi@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: 权限不够

#设置SUID权限
[root@localhost ~]# chmod u+s /usr/bin/cat
[root@localhost ~]# ll /usr/bin/cat
-rwsr-xr-x. 1 root root 54160 10月 31 2018 /usr/bin/cat

#普通用户再次使用cat命令时临时获取文件所有者身份
[lisi@localhost ~]$ cat /etc/shadow

#去除SUID权限
[root@localhost ~]# chmod u-s /usr/bin/cat
[root@localhost ~]# ll /usr/bin/cat
-rwxr-xr-x. 1 root root 54160 10月 31 2018 /usr/bin/cat

[root@localhost ~]# which vim
/usr/bin/vim

[root@localhost ~]# ll /usr/bin/vim
-rwxr-xr-x. 1 root root 2294208 10月 31 2018 /usr/bin/vim

#为vim设置SUID权限
[root@localhost ~]# chmod u+s /usr/bin/vim
[root@localhost ~]# ll /usr/bin/vim
-rwsr-xr-x. 1 root root 2294208 10月 31 2018 /usr/bin/vim

[root@localhost ~]# ll /etc/passwd
-rw-r--r--. 1 root root 2737 4月  10 17:26 /etc/passwd

[root@localhost ~]# chmod u-s /usr/bin/vim
[root@localhost ~]# vim /etc/passwd
```

SetGID特殊权限

- **SetGID（SGID）：当对一个可执行的程序文件设置了SGID后，普通用户在执行该文件时临时拥其所属组的权限，该权限只在程序执行过程中有效，程序执行完毕后用户恢复原有组身份**
- **当对一个目录作设置了SGID权限后，普通用户在该目录下创建的文件的所属组，均与该目录的所组相同**
- **SetGID权限会附加在所属组的 x 权限位上，所属组的 x 权限标识会变成 s**
- **设置SetGID命令格式：chmod g+s 文件名**

```
[root@localhost ~]# mkdir /test6
[root@localhost ~]# chmod 777 /test6
[root@localhost ~]# ll -d /test6
```

```
drwxrwxrwx. 2 root root 6 4月  11 15:59 /test6

#为目录设置SGID权限
[root@localhost ~]# chmod g+s /test6
[root@localhost ~]# ll -d /test6
drwxrwsrwx. 2 root root 6 4月  11 15:59 /test6
#SGID权限会附加在所属组执行权限位，所属组执行权限变为s

[root@localhost ~]# touch /test6/1.txt
[root@localhost ~]# ll /test6/1.txt
-rw-r--r--. 1 root root 0 4月  11 16:00 /test6/1.txt

#修改目录所属组为lisi组
[root@localhost ~]# chown :lisi /test6
[root@localhost ~]# ll -d /test6
drwxrwsrwx. 2 root lisi 19 4月  11 16:00 /test6

#SGID对目录设置后，在该目录下创建的任何文件都会继承父目录的所属组
[root@localhost ~]# touch /test6/2.txt
[root@localhost ~]# ll /test6/2.txt
-rw-r--r--. 1 root lisi 0 4月  11 16:01 /test6/2.txt
```

Sticky BIT特殊权限

- **Sticky BIT（SBIT）：该权限只针对于目录有效，当普通用户对一个目录拥有rwx权限时，普通用可以在此目录下拥有增删改的权限，应为普通用户对目录拥有rwx权限时，是可以删除此目录下的所文件**
- **如果对一个目录设置了SBIT权限，除了root可以删除所有文件以外，普通用户就算对该目录拥有rw权限，也只能删除自己建立的文件，不能删除其他用户建立的文件**
- **SBIT权限会附加在其他人的 x 权限位上，其他人的 x 权限标识会变成 t**
- **设置SBIT命令格式：chmod o+t 目录名**

```
#为目录设置SBIT
[root@localhost ~]# chmod o+t /test
[root@localhost ~]# ll -d /test
drwxrwxrwt. 2 root root 6 4月  11 16:07 /test

[lisi@localhost test]$ ls
kenji.txt  laowang.txt  lisi.txt

[lisi@localhost test]$ rm -rf *
rm: 无法删除"kenji.txt": 不允许的操作
rm: 无法删除"laowang.txt": 不允许的操作
```

FACL访问控制列表

- **FACL（Filesystemctl Access Control List）文件系统访问控制列表：利用文件扩展属性保存额的访问控制权限，单独为每一个用户量身定制一个权限**
- **命令格式：setfacl 选项 归属关系:用户名:权限 文档**
- **常用选项：**

- **-m 设置权限**
  - **-x 删除指定用户权限**
  - **-b 删除所有用户权限**

```
#为natasha用户设置ACL权限
[root@localhost ~]# setfacl -m u:natasha:rx /yunwei/
[root@localhost ~]# ll -d /yunwei/
drwxrwx---+ 2 root yunwei 54 4月  11 16:43 /yunwei/
[root@localhost ~]# ll -d /test
drwxrwxrwt. 2 root root 42 4月  11 16:11 /test

#查看目录ACL权限
[root@localhost ~]# getfacl /yunwei
getfacl: Removing leading '/' from absolute path names
# file: yunwei
# owner: root
# group: yunwei
user::rwx
user:natasha:r-x
group::rwx
mask::rwx
other::---

#用户测试权限
[natasha@localhost ~]$ ls /yunwei/
hell.sh  kenji.txt  lisi.txt
[natasha@localhost yunwei]$ rm -rf kenji.txt
rm: 无法删除"kenji.txt": 权限不够
[natasha@localhost yunwei]$ touch natasha.txt
touch: 无法创建"natasha.txt": 权限不够
[natasha@localhost yunwei]$ vim kenji.txt


[root@localhost ~]# setfacl -m u:tom:rx /yunwei
[root@localhost ~]# setfacl -m u:jack:rx /yunwei
[root@localhost ~]# setfacl -m u:hary:rx /yunwei
[root@localhost ~]# getfacl /yunwei
getfacl: Removing leading '/' from absolute path names
# file: yunwei
# owner: root
# group: yunwei
user::rwx
user:hary:r-x
user:tom:r-x
user:natasha:r-x
user:jack:r-x
group::rwx
mask::rwx
other::---

#删除指定用户ACL权限
[root@localhost ~]# setfacl -x u:tom /yunwei
[root@localhost ~]# getfacl /yunwei
```

```
getfacl: Removing leading '/' from absolute path names
# file: yunwei
# owner: root
# group: yunwei
user::rwx
user:hary:r-x
user:natasha:r-x
user:jack:r-x
group::rwx
mask::rwx
other::---

#删除所有用户ACL权限
[root@localhost ~]# setfacl -b /yunwei
[root@localhost ~]# getfacl /yunwei
getfacl: Removing leading '/' from absolute path names
# file: yunwei
# owner: root
# group: yunwei
user::rwx
group::rwx
other::---
```

课后练习

1.创建test1用户，并指定用户UID为6666，指定用户描述信息为test1@163.com，指定用户解释器/sbin/nologin

test1□x** 6666:6666:test1@163.com:/home/test1:/sbin/nologin**

2.创建名为stugrp组，将test1用户加入到stugrp组

[root@localhost ~]# groupadd stugrp

[root@localhost ~]# gpasswd -a test1 stugrp

3.请写出/etc/passwd文件中每个字段含义

用户名 密码占位符 UID GID 描述信息 家目录 解释器

4.创建test2用户，并设置密码为123456

[root@localhost ~]# useradd test2**

****[root@localhost ~]# passwd test2**

5.修改root用户密码为123456

[root@localhost ~]# passwd

6.请写出Linux系统下存放用户密码信息文件

/etc/shadow

7.设置test2用户首次登录系统需要修改密码

[root@localhost ~]# chage -d 0 test2

8.使用root切换为test1用户身份

su - 用户名

9.将test2用户添加至stugrp组，并锁定用户密码

[root@localhost ~]# gpasswd -a test2 stugrp

[root@localhost ~]# passwd -l test2

10.删除test1用户，连同用户家目录一并删除

root@localhost ~]# userdel -r test1

11.请写出Linux系统存放组信息文件，与组密码信息文件

[root@localhost ~]# ls /etc/group

[root@localhost ~]# ls /etc/gshadow

12.将test2用户从stugrp组中删除

[root@localhost ~]# gpasswd -d test2 stugrp

13.在根下创建upload目录，并修改目录所有者为test2用户，所属组为stugrp组，并将lisi用户加入stugrp组，修改所有者权限rwx，修改所属组权限为rwx，设置其他人没有任何权限

[root@localhost ~]# mkdir /upload

[root@localhost ~]# chown test2:stugrp /upload/

[root@localhost ~]# gpasswd -a lisi stugrp

[root@localhost ~]# chmod 770 /upload/

14.创建test3用户，非交互式设置用户密码为123456，并设置test3用户可以对upload目录拥有rx限

[root@localhost ~]# useradd test3**

****[root@localhost ~]# echo 123456 | passwd --stdin test3**

[root@localhost ~]# setfacl -m u:test3:rx /upload/**

****[root@localhost ~]# getfacl /upload/**

15.在根下创建shared目录，并同时设置所有人都有完全权限（至少两种方法设置），要求所有普通户在该目录下只能修改自己创建的文件

[root@localhost ~]# mkdir /shared**

[root@localhost ~]# chmod ugo=rwx /shared/

[root@localhost ~]# chmod 777 /shared/

****[root@localhost ~]# ll -d /shared/**

[root@localhost ~]# chmod o+t /shared/

常用特殊符号的使用

Linux系统下特殊符号起到了很大的作用，特殊符号可以完成一些特殊的功

***常用的特殊符号，在文件名上，用来代表任意多个任意字符**

? 常用的特殊符号，在文件名上，用来代表任意单个任意字符

[0-9] #在文件名上，用来代表多个字符或连续范围中的一个，若无则忽略

{a,b,cd,abcd} #在文件名上，用来代表多组不同的字符串，全匹配

- **范例**

```
#查找以tab结尾的文件
[root@localhost ~]# ls /etc/*tab
[root@localhost ~]# ls /etc/*wd
[root@localhost ~]# ls /etc/*.conf
[root@localhost ~]# ls /etc/redhat*
[root@localhost ~]# ls /etc/*ss*

#查找以tty开头的文件，结尾以一个任意字符结尾
[root@localhost ~]# ls /dev/tty?
[root@localhost ~]# ls /etc/host?
[root@localhost ~]# ls /etc/pass??

#查找tty开头结尾以1-5连续字符结尾
[root@localhost ~]# ls /dev/tty[1-5]
[root@localhost ~]# ls /dev/tty[4-9]
[root@localhost ~]# ls /dev/tty[1,3,5,7,9,15,20,30]

#查找tty开头结尾为不连续字符结尾
[root@localhost ~]# ls /dev/tty{1,3,5,7,9,15,20,30}
[root@localhost ~]# ls /dev/tty{1..9}
[root@localhost ~]# ls /dev/tty{1..10}
[root@localhost ~]# ls /dev/tty[1-10]
```

grep文件内容过滤

- **grep用于查找文件中符合条件的字符串，它能利用正则表达式搜索文件中的字符串，并把匹配到的符串的行打印出来**
- **命令格式：grep [-选项] "查找条件" 目标文件**
- **常用选项：**
  - **-n #以行号形式输出**
  - **-i #忽略字符串大小写**
  - **-v #显示不包含匹配的行（排除）**
- **常用正则表达式符号**

- **^字符串　　#显示以该字符串开头的行(匹配以什么什么开头的行)**
  - **字符串$　　#显示以该字符串结尾的行(匹配以什么什么结尾的行)**
  - **^$　　#显示空行**

- **grep命令示例**

```
#过滤包含root关键字的行
[root@localhost ~]# grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

#以行号形式过滤包含root关键字的行
[root@localhost ~]# grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin

[root@localhost ~]# grep -n bash /etc/passwd
[root@localhost ~]# grep -n : /etc/passwd

#忽略大小写过滤
[root@localhost ~]# grep -i -n ssh /etc/passwd
38:sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

#排除包含#号的行
[root@localhost ~]# grep -n -v '^#' /etc/fstab

#过滤以root开头的行
[root@localhost ~]# grep ^root /etc/passwd

#过滤以root结尾的行
[root@localhost ~]# grep -n 'root$' /etc/passwd
[root@localhost ~]# grep -n 'bash$' /etc/passwd

#语法错误示范
[root@localhost ~]# grep -n -v '^#' ^$  /etc/fstab
grep: ^$: 没有那个文件或目录
/etc/fstab:1:
/etc/fstab:9:/dev/mapper/centos-root /                       xfs     defaults        0 0
/etc/fstab:10:UUID=ae55ec6b-973b-498e-a366-f35e14b3d153 /boot                   xfs     defaults
        0 0
/etc/fstab:11:/dev/mapper/centos-swap swap

#语法错误示范
[root@localhost ~]# grep -n -v '^#' /etc/fstab | grep -v ^$
1:
9:/dev/mapper/centos-root /                       xfs     defaults        0 0
10:UUID=ae55ec6b-973b-498e-a366-f35e14b3d153 /boot                   xfs     defaults        0 0
11:/dev/mapper/centos-swap swap                    swap    defaults        0 0

#正确语法
[root@localhost ~]# grep  -v '^#' /etc/fstab | grep -v ^$ -n
2:/dev/mapper/centos-root /                       xfs     defaults        0 0
3:UUID=ae55ec6b-973b-498e-a366-f35e14b3d153 /boot                   xfs     defaults        0 0
```

```
4:/dev/mapper/centos-swap swap                    swap    defaults        0 0

#显示该文件内有效配置的行
[root@localhost ~]# grep -v '^#' /etc/login.defs | grep -v ^$ -n | wc -l
```

find文件/目录查找命令

- **find 命令根据预设的条件递归查找文件或目录所在位置**
- **命令格式：find 查找路径 查找条件1 查找条件2 .. [-exec 处理命令 {} ;] **
  - **–exec 可接额外的命令来处理查找到结果**
  - **{} 代表find查找到的内容被放置{}中**
  - **; 代表额外处理命令结束**
- **常用查找条件**
  - **-type 类型（f文件 d目录 l链接文件）**
  - **-name “文件名” **
  - **-iname 按文件名查找忽略大小写**
  - **-size 文件大小（k、M、G + 大于 - 小于）**
  - **-a （and并且）两个条件同时满足**
  - **-o （or或者）两个条件满足任意一个即可**
  - **-user 用户名**
  - **-mtime 按日期查找（+ 代表多少天之前 - 代表多少天之内，0代表24小时之内）**
- **find命令范例**

```
[root@localhost ~]# ls /var/log

#按照类型查找，类型为文件
[root@localhost ~]# find /var/log -type f
[root@localhost ~]# ll boot.log-20210417
[root@localhost ~]# ll /var/log/boot.log-20210417
[root@localhost ~]# ll /var/log/vmware-network.2.log

#按照类型查找，类型为目录
[root@localhost ~]# find /var/log -type d
[root@localhost ~]# ll -d /var/log/tuned
[root@localhost ~]# ll -d /var/log/qemu-ga

#按照类型查找，类型为链接文件
[root@localhost ~]# find /var/log -type l
[root@localhost ~]# fin /etc/ -type l
[root@localhost ~]# find /etc/ -type l
[root@localhost ~]# ll /etc/scl/conf

#按照名字查找
[root@localhost ~]# find /etc/ -name passwd
/etc/passwd
```

```
/etc/pam.d/passwd

#按照名字查找，类型为文件
[root@localhost ~]# find /etc/ -name passwd -type f

#按照名字查找，以tab结尾，类型为文件
[root@localhost ~]# find /etc/ -name '*tab' -type f

#按照名字查找，以pass开头，类型为文件
[root@localhost ~]# find /etc/ -name 'pass*' -type f
[root@localhost etc]# find . -name '*.conf' -type f

[root@localhost ~]# find /etc/ -name '*tab*' -type f

#按照名字忽略大小写查找，类型为文件
[root@localhost ~]# find /etc/ -iname FSTAB -type f
/etc/fstab
[root@localhost ~]# find /etc/ -name FSTAB -type f

#查找大于10k的文件
[root@localhost ~]# find /var/log -size +10k -type f
[root@localhost ~]# du -h /var/log/boot.log-20210417
16K    /var/log/boot.log-20210417

#查找大于1M的文件
[root@localhost ~]# find /var/log -size +1M -type f
[root@localhost ~]# du -h /var/log/audit/audit.log
2.4M    /var/log/audit/audit.log

[root@localhost ~]# find /home -size +1M -type f

#查找小于1M的文件
[root@localhost ~]# find /var/log -size -1M -type f
[root@localhost ~]# du -h /var/log/spooler
0    /var/log/spooler

#查找大于10k并且小于20k，类型为文件
[root@localhost ~]# find /var/log -size +10k -a -size -20k -type f

#-o或者，当有多个条件时，满足任意其中一个即可
[root@thinkmo ~]# find /var/log -name "*.log" -o -size -10k -type f


#查找属于lisi用户的文件/目录
[root@localhost ~]# find /home -user lisi

#查找30天之前被修改过，类型为文件
[root@localhost ~]# find /var/log -mtime +30 -type f
[root@localhost ~]# find /var/log -mtime +10 -type f

#查找10天之内被修改过，类型为文件
[root@localhost ~]# find /var/log -mtime -10 -type f
root@localhost ~]# find /var/log -mtime -30 -type f
```

#查找30之前被修改过，类型为文件，拷贝到/opt目录下
[root@localhost ~]# find /var/log -mtime -30 -type f -exec cp {} /opt \;

题型：

- **查找/etc/目录下以.conf结尾的文件（只能在/etc这一层目录去查找）**

[root@localhost ~]# ls /etc/*.conf

- **查找/etc/目录下以.conf结尾的文件（包含所有的子目录）**

[root@localhost ~]# find /etc/ -name '*.conf' -type f

百度：多查--多查--多查

查找/var/log/messages 文件，清空文件内容，使用find实现

[root@localhost ~]# find /var/log/ -name messages -type f -exec cp /dev/null {} ;

查找/var/log以.log结尾的文件，清空文件内容，使用find实现

*[root@localhost ~]# find /var/log -name .log -type f -a -mtime +10 -exec cp /dev/null {} ;

压缩与解压缩

- **Linux独有压缩格式及命令工具:**
  - **gzip---> .gz **
  - **bzip2---> .bz2**
  - **xz---> .xz **
- **压缩命令格式**
  - **gzip [选项...] 文件名**
    - **常用选项：-d 解压缩 **
  - **bzip2 [选项...] 文件名**
    - **常用选项：-d 解压缩**
  - **xz [选项...] 文件名**
    - **常用选项：-d 解压缩**
- **查看压缩文件内容**
  - **zcat [选项...] 文件名 #查看gzip格式压缩文件**
  - **bzcat [选项...] 文件名 **
  - **xzcat [选项...] 文件名**

[root@localhost ~]# cp /etc/services /opt
[root@localhost ~]# cd /opt
[root@localhost opt]# ll services
-rw-r--r--. 1 root root 670293 4月 17 17:06 services

```
[root@localhost opt]# ll -h services
-rw-r--r--. 1 root root 655K 4月  17 17:06 services

#使用gzip格式对文件进行压缩
[root@localhost opt]# gzip services
[root@localhost opt]# ls
services.gz
[root@localhost opt]# ll -h services.gz
-rw-r--r--. 1 root root 133K 4月  17 17:06 services.gz

#不解压查看压缩文件内容
[root@localhost opt]# zcat services.gz

#解压文件
[root@localhost opt]# gzip -d services.gz

#使用bzip2格式对文件进行压缩
[root@localhost opt]# bzip2 services
[root@localhost opt]# ls
services.bz2
[root@localhost opt]# ll -h services.bz2
-rw-r--r--. 1 root root 122K 4月  17 17:06 services.bz2

#不解压查看文件内容
[root@localhost opt]# bzcat services.bz2

#解压文件
[root@localhost opt]# bzip2 -d services.bz2

#使用xz格式对文件进行压缩
[root@localhost opt]# xz services
[root@localhost opt]# ls
services.xz
[root@localhost opt]# ll -h services.xz
-rw-r--r--. 1 root root 98K 4月  17 17:06 services.xz

#解压文件
[root@localhost opt]# xz -d services.xz
```

tar打包工具

- **tar命令用在linux下用于对文件/目录打包，使用 tar 程序打出来的包常称为 tar 包，tar 包文件通都是以 .tar 结尾 **
- **tar 命令格式：tar 选项 打包后名字 被打包文件**
- **常用选项：**
  - **-c 创建打包文件**
  - **-f 指定打包后的文件名称**
  - **-z 调用gzip压缩工具 -J 调用xz压缩工具 -j 调用bzip2压缩工具**
  - **-t 列出打包文档内容**
  - **-x 解压打包文件**

- **-C 指定解压路径**
  - **-v 显示详细信息**
- **tar命令范例**

```
#同时打包多个文件/目录并使用gzip格式压缩
[root@localhost opt]# tar -czf xxx.tar.gz /etc/passwd /etc/fstab /home

#将压缩包数据解压到/media目录
[root@localhost opt]# tar -xf xxx.tar.gz -C /media/
[root@localhost opt]# ls /media/etc
[root@localhost opt]# rm -rf xxx.tar.gz

#同时打包多个文件/目录并使用xz格式压缩
[root@localhost opt]# tar -cJf xx.tar.xz /etc/hostname /etc/services /home

#错误语法，f选项要放到所有选项右边
[root@localhost opt]# tar -ft xx.tar.xz
tar: 您必须从"-Acdtrux"或是"--test-label"选项中指定一个
请用“tar --help”或“tar --usage”获得更多信息。

#不解压查看压缩包数据
[root@localhost opt]# tar -tf xx.tar.xz
etc/hostname

#将压缩包数据解压到/tmp目录
[root@localhost opt]# tar -vxf xx.tar.xz -C /tmp
[root@localhost opt]# ls /tmp

#同时打包多个文件/目录并使用bzip2格式压缩
[root@localhost opt]# tar -cjf abc.tar.bz2 /etc/hostname /etc/group /home

#解压缩
[root@localhost opt]# tar -xf abc.tar.bz2 -C /media/
```

磁盘介绍

分区过程

添加新硬盘--分区--格式化文件系统--挂载使用

扇区是磁盘存储数据的最小单元，默认一个扇区可以存储512字节的数据

磁盘类型介绍

- **IDE接口类型：主要用于个人家用计算机领域，优点价格便宜，缺点数据传输速度慢**
- **SCSI接口类型：早期主要用于服务器理领域，数据传输速度快**
- **SAS接口类型：目前在服务器领域比较流行**
- **SATA接口类型：串口磁盘，主要用于个人家用计算机领域，偶尔也应用在服务器领域**
- **SSD接口类型：固态硬盘接口，价格昂贵，数据传输速度快，利用内存的机制读写数据，主要应用**

个人电脑

- **NVMe接口类型：固态硬盘接口，价格昂贵，数据传输速度快，利用内存的机制读写数据**

Linux常用分区格式

- **MBR分区格式：比较古老的分区格式，最初只能划分4个主分区，后来新增加扩展分区（容器）功，可在扩展分区内划分更多逻辑分区，最大支持2.2T磁盘容量**
  - **IDE接口硬盘逻辑分区最多可以划分59个**
  - **SCSI接口硬盘逻辑分区最多可以划分11个**
  - **最大支持2.2T以内磁盘容量**
- **GPT分区格式：可划分128个主分区，最大支持18EB磁盘容量（1EB=1024PB，1PB=1024TB，1B=1024GB）**

文件系统类型详解

- **文件管理系统，赋予分区文件系统，分区才可以正常的使用，根文件系统，多少个多少个文件系统**
- **CentOS5：分区默认使用文件系统类型ext3**
- **CentOS6：分区默认使用文件系统类型ext4**
  - **ext4日志记录功能，意外宕机，通过日志记录把没有保存的数据，在系统再次重启时快速恢复来**
  - **单个文件系统最大支持1EB的分区容量，单个文件最大可以存储16TB数据**
- **CentOS7：分区默认使用文件系统类型xfs**
  - **xfs开启了日志记录的功能，意外宕机，通过日志记录把没有保存的数据，在系统再次重启时快恢复回来，数据恢复的速度比ext4文件系统快**
  - **单个文件系统最大支持8EB分区容量，单个文件最大可以存储500TB的数据**
  - **单个文件每秒读写数据的速度可以达到4G**
- **swap文件系统：交换分区，硬盘空间去充当内存去使用**

挂载

- **在Linux系统中用户无法直接使用硬件设备的，硬件设备在系统中都是以只读的方式存在的，必须载**
- **挂载就是给我们用户提供一个可以使用设备的一个接口**
- **挂载注意事项：**
  - **挂载点必须是一个目录，理论上还得是一个空目录**
  - **一个文件系统不允许重复挂载到多个目录下**
  - **一个目录不允许重复挂载多个文件系统**

lsblk查看系统所有磁盘信息

- **lsblk（英文全拼：list block）用于列出当前系统所有磁盘与磁盘内的分区信息**
- **命令格式：lsblk [选项...] [设备名]**
- **常用选项：**
  - **-d #仅显示磁盘本身，不会列出磁盘的分区数据**
  - **-f #列出磁盘分区使用的文件系统类型**
- **lsblk命令示例**

```
#列出当前系统所有磁盘与磁盘内的分区信息
[root@localhost ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda               8:0    0   20G  0 disk
├─sda1            8:1    0    1G  0 part /boot
└─sda2            8:2    0   19G  0 part
  ├─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sr0              11:0    1  4.3G  0 rom  /mnt/centos

#sda1：sd代表SCSI磁盘，a代表第一块磁盘，1代表第一个分区
#sdb：sd代表SCSI磁盘，b代表第二块磁盘，1代表第一个分区
#解释：
NAME        #设备名称
MAJ:MIN        #主设备号:次设备号，内核通过主次设备号识别磁盘
RM            #是否为可卸载设备，1可卸载，0不可卸载
SIZE        #设备的容量大小
RO            #表示设备是否为只读，0非只读设备，1只读设备
TYPE        #表示设备类型（disk为磁盘，part为分区，lvm逻辑卷，rom只读）
MOUNTPOINT    #设备挂载点（SWAP没有挂载点）

#列出指定的磁盘信息
[root@localhost ~]# lsblk -d /dev/sda
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda    8:0    0  20G  0 disk

#列出所有磁盘分区内使用的文件系统类型
[root@localhost ~]# lsblk -f
NAME            FSTYPE      LABEL           UUID                                   MOUNTPOINT
sda
├─sda1          xfs                         4cb9bb38-c34a-4415-9614-ba38642bb86d   /boot
└─sda2          LVM2_member                 cKn0jP-z8Bq-SNvl-BsNa-7vTg-GBU2-OiHCro
  ├─centos-root xfs                         55dad88d-a600-42d1-b387-236db62ce396   /
  └─centos-swap swap                        2e91599a-6d72-483d-add8-6dfb84296170   [SWAP]
sr0             iso9660     CentOS 7 x86_64 2018-11-25-23-54-16-00                /mnt/centos

#列出指定分区的文件系统类型
[root@localhost ~]# lsblk -df /dev/sda1
NAME FSTYPE LABEL UUID                                 MOUNTPOINT
sda1 xfs         4cb9bb38-c34a-4415-9614-ba38642bb86d /boot
```

df查看分区使用情况

- **df命令用于查看文件系统使用情况**
- **命令格式：df [选项...] [参数...]**
- **常用选项：**
  - **-h 以人类易读方式显示文件系统容量**
  - **T　显示文件系统类型**
- **df 命令示例**

```
[root@localhost ~]# df
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/centos-root  17811456 3746320  14065136  22% /
devtmpfs                   480884       0    480884   0% /dev
tmpfs                      497948       0    497948   0% /dev/shm
tmpfs                      497948    8340    489608   2% /run
tmpfs                      497948       0    497948   0% /sys/fs/cgroup
/dev/sr0                  4480476 4480476         0 100% /mnt
/dev/sda1                 1038336  169448    868888  17% /boot
tmpfs                      99592      12     99580   1% /run/user/42
tmpfs                      99592       0     99592   0% /run/user/0

[root@localhost ~]# df -h /
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/centos-root   17G  3.6G   14G  22% /
```

du统计文件/目录大小

- **du命令用于统计磁盘下目录或文件大小**
- **命令格式：du [选项...] [参数...]**
- **常用选项：**
  - **-h　　#以人类易读方式（Kb，MB，GB）显示文件大小**
  - **-s　　#只统计每个参数的总数**
- **du 命令示例**

```
[root@localhost ~]# du -h /etc/services

[root@localhost ~]# du -hs /etc
38M    /etc
```

- **/dev/目录下文件详解**

```
[root@localhost ~]# ls /dev
hd[a-t]:IDE设备
sd[a-z]:SCSI设备
fd[0-7]：软盘驱动设备
md[0-32]：软RAID设备
loop[0-7]：本地回环设设备
lp[0-3]:打印机设备
mem：内存设备
```

null：空设备，也称为黑洞，任何写入的数据都将被丢弃
zero：零资源设备，任何写入的数据都将被丢弃
full：满设备，任何写入的数据都将失败
tty[0-63]：虚拟终端设备
random：随机数设备
urandom：随机数设备
port：存取I/O端口

blkid查看设备属性

- **blkid命令显示块设备属性信息（设备名称，设备UUID，文件系统类型）**
- **命令格式：blkid [选项...] [参数...]**
- **blkid命令示例**

```
#显示系统所有块设备属性信息
[root@localhost ~]# blkid
/dev/sda1: UUID="4cb9bb38-c34a-4415-9614-ba38642bb86d" TYPE="xfs"
/dev/sda2: UUID="cKn0jP-z8Bq-SNvl-BsNa-7vTg-GBU2-OiHCro" TYPE="LVM2_member"
/dev/sr0: UUID="2018-11-25-23-54-16-00" LABEL="CentOS 7 x86_64" TYPE="iso9660" PTTYP
="dos"
/dev/mapper/centos-root: UUID="55dad88d-a600-42d1-b387-236db62ce396" TYPE="xfs"
/dev/mapper/centos-swap: UUID="2e91599a-6d72-483d-add8-6dfb84296170" TYPE="swap"

#查看执行分区属性信息
root@localhost ~]# blkid /dev/sda1
/dev/sda1: UUID="4cb9bb38-c34a-4415-9614-ba38642bb86d" TYPE="xfs"
```

MBR分区格式

- **fdisk命令用于查看磁盘使用情况和磁盘分区（MBR分区格式）**
- **命令格式：fdisk [选项...] [设备路径]**
- **常用选项：-l 列出磁盘分区表类型与分区信息**
- **分区**

```
[root@localhost ~]# fdisk /dev/sdb
m    #获取命令帮助        ※
p    #显示磁盘分区表    ※
n    #新增加一个分区    ※
q    #不保存分区退出    ※
d    #删除一个分区      ※
w    #保存分区退出      ※
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
l    #显示已知的文件系统类型，82为swap交换分区，83为Linux分区
o    #建立空白DOS分区表
s    #新建空白SUN磁盘标签
t    #改变分区的系统ID
u    #改变显示记录单位
v    #验证分区表
```

```
x   #附加功能

命令(输入 m 获取帮助)：m
命令(输入 m 获取帮助)：p

#划分第一个主分区
命令(输入 m 获取帮助)：n
Select (default p):  回车
分区号 (1-4，默认 1)：回车
起始 扇区 (2048-209715199，默认为 2048)：回车
Last 扇区, +扇区 or +size{K,M,G} (2048-209715199，默认为 209715199)：+10G  #指定大小（K
M,G)
分区 1 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：p
磁盘标签类型：dos
磁盘标识符：0xefc65503

   设备 Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048    20973567    10485760   83  Linux

#划分第二个主分区
命令(输入 m 获取帮助)：n
Select (default p):
分区号 (2-4，默认 2):
起始 扇区 (20973568-209715199，默认为 20973568):
Last 扇区, +扇区 or +size{K,M,G} (20973568-209715199，默认为 209715199)：+10G  #指定分
大小

#划分第三个主分区
命令(输入 m 获取帮助)：n
Select (default p):
分区号 (3,4，默认 3):
起始 扇区 (41945088-209715199，默认为 41945088):
Last 扇区, +扇区 or +size{K,M,G} (41945088-209715199，默认为 209715199)：+10G

#查看分区信息
命令(输入 m 获取帮助)：p
磁盘标签类型：dos
磁盘标识符：0xefc65503

   设备 Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048    20973567    10485760   83  Linux
/dev/sdb2        20973568    41945087    10485760   83  Linux
/dev/sdb3        41945088    62916607    10485760   83  Linux

#划分第四个分区
命令(输入 m 获取帮助)：n
Select (default e): p
起始 扇区 (62916608-209715199，默认为 62916608):
Last 扇区, +扇区 or +size{K,M,G} (62916608-209715199，默认为 209715199)：+10G

#继续划分分区
命令(输入 m 获取帮助)：n
```

If you want to create more than four partitions, you must replace a
primary partition with an extended partition first.
#提示如果想要创建更多的分区，先将一个主分区替换为扩展分区

#删除分区
命令(输入 m 获取帮助)：d4
分区号 (1-4，默认 4)：
分区 4 已删除

命令(输入 m 获取帮助)：d
分区号 (1-3，默认 3)：3
分区 3 已删除

命令(输入 m 获取帮助)：p
磁盘标签类型：dos
磁盘标识符：0xefc65503

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	20973567	10485760	83	Linux
/dev/sdb2		20973568	41945087	10485760	83	Linux

#创建主分区
命令(输入 m 获取帮助)：n
Select (default p):
分区号 (3,4，默认 3)：
起始 扇区 (41945088-209715199，默认为 41945088)：
Last 扇区, +扇区 or +size{K,M,G} (41945088-209715199，默认为 209715199)：+10G

#创建按扩展分区
命令(输入 m 获取帮助)：n
Select (default e):
Using default response e
已选择分区 4
起始 扇区 (62916608-209715199，默认为 62916608)：
Last 扇区, +扇区 or +size{K,M,G} (62916608-209715199，默认为 209715199)：
分区 4 已设置为 Extended 类型，大小设为 70 GiB

#创建逻辑分区
命令(输入 m 获取帮助)：n
添加逻辑分区 5
起始 扇区 (62918656-209715199，默认为 62918656)：
Last 扇区, +扇区 or +size{K,M,G} (62918656-209715199，默认为 209715199)：+10G
分区 5 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：p
磁盘 /dev/sdb：107.4 GB, 107374182400 字节，209715200 个扇区
磁盘标签类型：dos
磁盘标识符：0xefc65503

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	20973567	10485760	83	Linux
/dev/sdb2		20973568	41945087	10485760	83	Linux
/dev/sdb3		41945088	62916607	10485760	83	Linux
/dev/sdb4		62916608	209715199	73399296	5	Extended

```
/dev/sdb5       62918656    83890175    10485760   83  Linux
命令(输入 m 获取帮助)：w
```

格式化文件系统

- **mkfs命令用于在分区上建立文件系统**
- **常用文件系统类型**
  - **ext4，xfs**
- **命令格式：**
  - **mkfs.xfs 分区设备路径　#格式化为xfs类型文件系统**
  - **mkfs.ext4 分区设备路径 #格式化为ext4类型文件系统**

```
#格式化文件系统
[root@localhost ~]# mkfs.xfs /dev/sdb1

#查看文件系统类型
[root@localhost ~]# blkid /dev/sdb1
/dev/sdb1: UUID="3bb79b0b-3f17-4ad9-ad47-f00dcb6a5afa" TYPE="xfs"
```

mount挂载

- **mount文件系统挂载命令**
- **命令格式：mount 设备路径 挂载点目录**

```
#创建挂载点目录
[root@localhost ~]# mkdir /mybak

#挂载文件系统
[root@localhost ~]# mount /dev/sdb1 /mybak

#查看正在使用中的分区信息
[root@localhost ~]# df -Th

[root@localhost ~]# df -Th /mybak
文件系统       类型  容量  已用  可用 已用% 挂载点
/dev/sdb1      xfs    10G   33M   10G    1% /mybak
```

总结：

- **添加硬盘---查看系统是否识别新硬盘 lsblk**
- **划分分区---fdisk 设备路径**
- **格式化文件系统---mkfs.xfs**
- **挂载---创建挂载点目录--挂载 mount 设备路径 挂载点目录**
- **查看分区使用情况 df -hT**

umount卸载

- **umount命令用于卸载文件系统**
- **命令格式：umount 挂载点目录**

```
#卸载文件系统
[root@localhost ~]# umount /mybak
[root@localhost ~]# df -h
```

开机自动挂载

- **/etc/fstab用于存放文件系统信息，当系统启动时，系统会自动读取文件内容将指定的文件系统挂到指定的目录**
- **文件内容详解**

```
[root@localhost ~]# vim /etc/fstab
/dev/mapper/centos-root /                       xfs     defaults        0 0
UUID=5d36a8b5-5a58-450f-acf9-81fcddaa62de /boot                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
#解释：该文件内容为6个字段，每个字段详解如下
第一个字段：要挂载的设备路径
第二个字段：挂载点目录
第三个字段：设备文件系统类型
第四个字段：挂载参数，参数如下↓
sync，async：  此文件系统是否使用同步写入 (sync) 或异步 (async) 的内存机制，默认为异步（asyn
)
atime，noatime：更新访问时间/不更新访问时间，访问分区时，是否更新文件的访问时间，默认为
新
ro，rw：挂载文件为只读（ro）或读写（rw），默认为rw
auto，noauto：自动挂载/手动挂载，执行mount -a时，是否自动挂载/etc/fstab文件内容，默认为
动（auto）
dev，nodev：是否允许此文件系统上，可建立装置文件，默认为允许（dev）
suid，nosuid：是否允许文件系统上含有SUID与SGID特殊权限，默认为允许（SUID）
exec，noexec：是否允许文件系统上拥有可执行文件，默认为允许（exec）
user，nouser：是否允许普通用户执行挂载操作，默认为不允许（nouser），只有root用户可以挂
分区
defaults默认值：代表async，rw，auto，dev，suid，exec，nouser七个选项
第五个字段：是否对文件系统进行备份，0不备份，1为备份
第六个字段：是否检查文件系统顺序，允许的数字是0，1，2，0表示不检查，1的优先权最高

/dev/mapper/centos-root /                       xfs     defaults        0 0
UUID=ae55ec6b-973b-498e-a366-f35e14b3d153 /boot                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
/dev/sdb1 /mybak xfs defaults 0 0     #手动添加
```

- **mount常用选项：**
  - **-a：依照配置文件/etc/fstab的数据将所有未挂载的磁盘都挂载上来**
  - **-o：该选项后边可跟挂载时额外参数**
- **remount命令：重新挂载文件系统，在文件系统出错时或重新挂载文件系统时非常重要**

```
[root@localhost ~]# mount -a
```

GPT分区格式

- **gdisk命令用于查看磁盘使用情况和磁盘分区（GPT分区格式）**
- **命令格式：gdisk [选项...] [设备路径]**
- **常用选项：-l 列出磁盘分区表类型与分区信息**

```
[root@localhost ~]# gdisk /dev/sdc
GPT fdisk (gdisk) version 0.8.10 #GPT版本

Partition table scan:   #分区表扫描
  MBR: not present        #MBR分区不存在
  BSD: not present        #BSD分区不存在
  APM: not present        #APM分区不存在
  GPT: not present        #GPT分区不存在

Creating new GPT entries.  #创建新的GPT分区

Command (? for help): ?   #输入？号获取命令帮助
p    #显示磁盘分区表   ※
n    #新增加一个分区   ※
q    #不保存分区退出   ※
d    #删除一个分区     ※
w    #保存分区退出     ※

#创建新的分区
Command (? for help): n
Partition number (1-128, default 1): 回车
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 回车  #输入起始扇区，默认204
开始
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}: +20G #输入新增分区
大小，可以通过扇区数来增加，也可以通过+size{KMGTP}方式来增加
Hex code or GUID (L to show codes, Enter = 8300):  #这里要求输入分区的类型，直接回车就行

#查看分区类型
Command (? for help): p  #输入p查看创建的分区
Disk /dev/sdc: 209715200 sectors, 100.0 GiB  #磁盘总容量
...
Total free space is 167772093 sectors (80.0 GiB)  #磁盘剩余容量

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        41945087   20.0 GiB    8300  Linux filesystem
#以创建的分区

Command (? for help): w   #输入w保存配置，如果不想保存可以输入q退出
Do you want to proceed? (Y/N): y  #问你是否相想继续，输入y继续
OK; writing new GUID partition table (GPT) to /dev/sdc.
The operation has completed successfully.   #写入成功

#格式化文件系统
[root@localhost ~]# mkfs.xfs /dev/sdc1

#查看文件系统类型
[root@localhost ~]# blkid /dev/sdc1
```

```
/dev/sdc1: UUID="c57746eb-8170-4c86-82ad-6aae95de19f3" TYPE="xfs"

#创建挂载点
[root@localhost ~]# mkdir /webbak
[root@localhost ~]# mount /dev/sdc1 /webbak
[root@localhost ~]# df -hT
/dev/sdc1              xfs        20G   33M   20G    1% /webbak

#开机自动挂载
[root@localhost ~]# vim /etc/fstab
/dev/mapper/centos-root /                       xfs     defaults        0 0
UUID=ae55ec6b-973b-498e-a366-f35e14b3d153 /boot                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
/dev/sdb1               /mybak                  xfs     defaults        0 0
/dev/sdc1               /webbak                 xfs     defaults        0 0   #手动添加

[root@localhost ~]# mount -a
```

LVM逻辑卷

- **逻辑卷：LVM（Logical Volume Manager）逻辑卷管理系统 **
- **逻辑卷可以实现将底层的物理分区整合成一个大的虚拟硬盘**
- **逻辑卷技术是通过Linux系统内核dm（device mapper）设备映射组**

创建卷组

- **创建卷组思路：将创建好的物理卷组成卷组（或者直接创建卷组）**
- **命令格式：vgcreate 卷组名 设备路径1 设备路径2...**

```
#创建卷组
[root@localhost ~]# vgcreate systemvg  /dev/sdb2 /dev/sdb3

#详细显示卷组信息
[root@localhost ~]# vgdisplay  systemvg
  --- Volume group ---
  VG Name               systemvg  #卷组名字
  System ID
  Format                lvm2      #卷组格式
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               19.99 GiB  #卷组大小
  PE Size               4.00 MiB
  Total PE              5118
```

```
  Alloc PE / Size       0 / 0
  Free  PE / Size       5118 / 19.99 GiB
  VG UUID               KEP7XS-wrkI-rTUY-RqBa-UJA6-YRkK-iKDabR  #卷组UUID

#简要显示卷组信息
[root@localhost ~]# vgs systemvg
  VG       #PV #LV #SN Attr   VSize  VFree
  systemvg   2   0   0 wz--n- 19.99g 19.99g
```

创建逻辑卷

- **创建逻辑卷思路：从创建好的卷组中创建逻辑卷**
- **命令格式：lvcreate -L 大小 -n 逻辑卷名 卷组名**

```
#创建逻辑卷
[root@localhost ~]# lvcreate -L 10G -n mylv systemvg
  Logical volume "mylv" created.

#简要查看逻辑卷信息
[root@localhost ~]# lvs
  LV   VG       Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root centos   -wi-ao---- <17.00g
  swap centos   -wi-ao----   2.00g
  mylv systemvg -wi-a-----  10.00g
[root@localhost ~]# lvs /dev/systemvg/mylv
  LV   VG       Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  mylv systemvg -wi-a----- 10.00g

 #查看卷组信息，卷组信息以变小
 [root@localhost ~]# vgs
  VG       #PV #LV #SN Attr   VSize   VFree
  centos     1   2   0 wz--n- <19.00g    0
  systemvg   2   1   0 wz--n-  19.99g 9.99g
```

格式化文件系统

```
#格式化文件系统
[root@localhost ~]# mkfs.xfs /dev/systemvg/mylv

#查看文件系统类型
[root@localhost ~]# blkid /dev/systemvg/mylv
/dev/systemvg/mylv: UUID="7f08daf8-ae3c-40b2-a282-4514a6f37111" TYPE="xfs"

#挂载使用
[root@localhost ~]# mkdir /dbbak
[root@localhost ~]# mount /dev/systemvg/mylv /dbbak
[root@localhost ~]# df -hT
/dev/mapper/systemvg-mylv xfs        10G   33M   10G    1% /dbbak
```

扩展逻辑卷

- **逻辑卷支线上扩容，逻辑卷的空间来源于卷组，当卷组有足够的空间时，才可以扩展逻辑卷**

- **扩展命令：lvextend**

```
#扩容逻辑卷
[root@localhost ~]# lvextend -L +9G /dev/systemvg/mylv

#查看逻辑卷信息
[root@localhost ~]# lvs
  LV   VG       Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root centos   -wi-ao---- <17.00g
  swap centos   -wi-ao----   2.00g
  mylv systemvg -wi-ao----  19.00g    #扩容成功
```

扩展文件系统

- **当逻辑卷扩大以后，也需要对逻辑卷的文件系统进行扩展**
- **扩展文件系统容量：**
  - **xfs_growfs　#用于扩容XFS设备**
  - **resize2fs　　#用于扩容EXT3/EXT4设备（了解）resize2fs /dev/systemvg/xxoo **

```
#扩展文件系统
[root@localhost ~]# xfs_growfs /dbbak

#[root@localhost ~]# df -hT
/dev/mapper/systemvg-mylv xfs        19G   33M   19G    1% /dbbak

#查看卷组信息
[root@localhost ~]# vgs
  VG       #PV #LV #SN Attr   VSize   VFree
  centos     1   2   0 wz--n- <19.00g       0
  systemvg   2   1   0 wz--n-  19.99g 1016.00m
```

扩展卷组

- **卷组的空间来源于物理分区，当卷组没有足够空间提供给逻辑卷时，须扩容卷组**
- **扩展卷组命令：vgextend**

```
[root@localhost ~]# vgextend systemvg /dev/sdb5 /dev/sdb6 /dev/sdb7 /dev/sdb8

[root@localhost ~]# vgs
  VG       #PV #LV #SN Attr   VSize   VFree
  centos     1   2   0 wz--n- <19.00g       0
  systemvg   6   1   0 wz--n- <59.98g <40.98g

#扩容逻辑卷
[root@localhost ~]# lvextend -L +40G /dev/systemvg/mylv


[root@localhost ~]# lvs
  LV   VG       Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root centos   -wi-ao---- <17.00g
```

```
  swap centos   -wi-ao----   2.00g
  mylv systemvg -wi-ao----  59.00g

#扩展文件系统
[root@localhost ~]# xfs_growfs /dbbak
/dev/mapper/systemvg-mylv   59G   34M   59G    1% /dbbak
```

课后作业

1.查看/var/log目录下以包含log的文件

[root@localhost ~]# ls /var/log/log**

2.查看/var/log目录下以数字结尾的文件

[root@localhost ~]# ls /var/log/*[0-9]

3.查看/var/log目录下以字母结尾的文件（包括大写）

[root@localhost ~]# ls /var/log/*[a-Z]

4.过滤/etc/sudoers文件以root开头的行

root@localhost ~]# grep ^root /etc/sudoers**

root   ****ALL=(ALL) **   **ALL**

5.看/etc/sudoers文件有效的配置

[root@localhost ~]# grep -v '^#' /etc/sudoers | grep -v '^$' -n

6.查找/etc/目录下crontab文件存放位置，并查看文件内容

[root@localhost ~]# find /etc/ -name crontab -type f

[root@localhost ~]# cat /etc/crontab

[root@localhost ~]# find /etc/ -name crontab -type f -exec cat {} ;

7.查找10分钟内被修改的文件

[root@localhost ~]# find / -cmin -10 -type f

8.查找/var/log目录下30天之前被修改且大于1M的文件，清空文件内容

[root@localhost ~]# find /var/log -mtime +30 -type f -size +10k -exec cp /dev/null {} ;

9.Linux下你常熟悉的压缩格式有哪些?

gzip bzip2 xz

10.对/home目录打包并压缩，打包后名为home.tar.gz

[root@localhost ~]# tar -czf home.tar.gz /home

11.将home.tar.gz压缩包内容解压至/homebak目录下

[root@localhost ~]# tar -xvf home.tar.gz -C /homebak/

12.MBR分区格式可以划分多少个主分区？支持多大容量磁盘？

4个主分区，2.2T

13.GPT分区格式可以划分多少个主分区？支持多大容量磁盘？

128主分区，18EB

14.CentOS7分区默认使用的文件系统类型是什么？

xfs

15.如何查看一块磁盘的分区格式？及扩展分区大小？

[root@localhost ~]# fdisk -l /dev/sdc

磁盘标签类型：gpt

16如何查看一块磁盘剩余容量？

[root@localhost ~]# lsblk /dev/sdc

17.linux下开机自动挂载文件是哪个？

/etc/fstab

18.如何查看一个分区文件系统类型？及使用情况？

[root@localhost ~]# df -hT

19.为根分区扩容40G空间（添加硬盘、分区）

```
#查看根分区卷组
[root@localhost ~]# vgs
  VG       #PV #LV #SN Attr   VSize   VFree
  centos     1   2   0 wz--n- <19.00g       0

#扩容根分区卷组
[root@localhost ~]# vgextend centos /dev/sdc2 /dev/sdc3

#查看根分区逻辑卷信息
[root@localhost ~]# lvs
  LV   VG       Attr       LSize
  root centos   -wi-ao---- <17.00g

#扩容逻辑卷
[root@localhost ~]# lvextend -L +39G /dev/centos/root

#查看逻辑卷信息
[root@localhost ~]# lvs
  root centos   -wi-ao---- <56.00g
```

```
#查看正在使用的分区信息
[root@localhost ~]# df -hT
文件系统                类型      容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root xfs        17G  4.4G   13G   26% /

#扩容文件系统
[root@localhost ~]# xfs_growfs /

#查看使用情况
[root@localhost ~]# df -h
文件系统                容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root   56G  4.4G   52G    8% /
```

删除逻辑卷

- **逻辑卷的删除不允许联机操作，需要先卸载，在执行删除**
- **在执行删除操作时，首先删除LV逻辑卷，在删除VG卷组，最后删除PV物理卷**
- **删除命令：lvremove**

```
#删除逻辑卷错误示范
[root@localhost ~]# lvremove /dev/systemvg/mylv
  Logical volume systemvg/mylv contains a filesystem in use.  #提示文件正在使用中

#需要先卸载
[root@localhost ~]# umount /dblod/

#删除逻辑卷
[root@localhost ~]# lvremove /dev/systemvg/mylv
Do you really want to remove active logical volume systemvg/mylv? [y/n]: y
  Logical volume "mylv" successfully removed

#删除卷组
[root@localhost ~]# vgremove systemvg
  Volume group "systemvg" successfully removed

#删除物理卷后将恢复至普通分区
#查看物理卷
[root@thinkmo ~]# pvs

#删除物理卷
[root@thinkmo ~]# pvremove /dev/sdb2 /dev/sdb3 /dev/sdb5 /dev/sdb6 /dev/sdb6 /dev/sd
7 /dev/sdb8
```

逻辑卷的缩减

- **命令lvreduce**
- **不允许联机缩减**
- **先缩减文件系统的空间，在缩减逻辑卷的空间**

RAID磁盘阵列

- RAID中文全称：独立磁盘冗余阵列 ，简称磁盘阵列
- RAID可通过技术（软件/硬件）将多个独立的磁盘整合成一个巨大容量大逻辑磁盘使用
- RAID可以提高数据I/O（读写）速度，和冗余数据的功能

RAID级别

RAID0：等量存储，至少由2块磁盘组成，同一个文档等量存放在不同的磁盘并行写入数据来提高效，但只是单纯的提高效率，并没有冗余功能，如果其中一块盘故障，数据会丢失，不适合存放重要数据

RAID1：完整备份，至少由两块磁组成，同一个文档复制成多份存储到不同磁盘提高可靠性，读写速没有提升，适合存储重要的数据

RAID2：至少由3块磁盘组成，数据分散存储在不同磁盘，在读写数据时需要对数据时时校验，由于用的校验算法复杂，数据量比原有数据增大，而且导致硬件开销较大

RAID3：至少由三块磁盘组成，同一份文档分散写入不同的磁盘，校验数据单独存放在另外一块磁盘由于每次读写操作都会访问校验盘，容易导致校验盘长时间高负荷工作而挂掉，如果校验盘损坏数据无法恢复

RAID4：与RAID3类似，至少由3块磁盘组成，同一份文档分散存写入不同磁盘，校验数据单独存放另外一块磁盘，由于每次读写操作都会访问校验盘，容易导致校验盘长时间高负荷工作而挂掉，如果验盘损坏数据将无法恢复，与RAID3的区别是数据分割方式不一样

RAID5：至少由3块磁盘组成，同一份文档分散写入不同磁盘，每个硬盘都有校验数据，其中校验数会占用磁盘三分之一的空间，三分之二的空间存放原始数据，允许同时坏一块磁盘，当一块磁盘损坏其他磁盘里的数据配合校验信息可将数据恢复回来

RAID6：至少由4块磁盘组成，同一份文档分散写入不同磁盘，每个磁盘都有校验数据，由于采用双验算法，所以校验数据量是RAID5的两倍，需要占用2块磁盘空间存放校验数据，两块盘存放原始数，由于数据校验的算法计算量偏大，所以在速写速度上没有RAID5快，允许同时坏2块磁盘

RAID7：美国SCC公司专利，花钱

RAID10：RAID10=RAID1+RAID0合二为一，最少需要4块磁盘，先将4块硬盘组成两组RAID1，将两组RAID1组成一个RAID0，既提高数据读写速度，又能保障数据安全性，缺点是可用容量是总量的一半

实现RAID方式

- 实现RAID通常有三种方式，通过软件技术实现RAID功能（软RAID），不稳定
- 外接式磁盘阵列柜，被常用在大型服务器上，不过这类产品价格昂贵
- RAID磁盘阵列卡，分为服务器自带和额外安装，硬RAID比软RAID更安全稳定，RAID卡带有缓存能可实现数据自动恢复，RAID卡有电池
- 配置硬RAID方式

进程管理

- 什么是程序：用计算机语言编写的命令序列集合，用来实现特定的目标或解决特定的问题，程序占磁盘空间，程序是静态并且是永久的
- 什么是进程：正在运行中的程序叫进程，占用内存空间，进程是动态的，进程是有生命周期的，进

有自己的独立内存空间，每启动一个进程，系统就会为它分配内存空间并分配一个PID号，每个进程会对应一个父进程，而父进程可以复制多个子进程，每种进程都有两种方式存在，前台与后台，一般程都是以后台方式运

- **什么是线程：线程也被称为轻量级进程，被包含在进程中，是进程的一个子集，是进程中的实际运单位，一个进程中可以并发多个线程，每条线程并行执行不同的任务，每个线程都是独立的，线程之共享进程的内存空间，在多线程的程序中，由于线程很“轻”，故线程的切换非常迅速且开销小（在一进程中）**

查看进程树

- **pstree以树状结构显示进程信息，包括进程之间的关系**
- **命令格式：pstree [选项...] [参数...]**
- **常用选项：**
  - **-p #显示进程PID**
  - **-a #显示完整的命令行**
  - **-u #列出每个进程所属账号名称**

```
#查看进程树
[root@localhost ~]# pstree
systemd─┬─ModemManager───2*[{ModemManager}]
CentOS7版本：天父进程systemd
CentOS6版本：天父进程init，Apstart
CentOS5版本：天赋进程init

#以PID形式显示进程信息
[root@localhost ~]# pstree -p
systemd(1)─┬─ModemManager(6714)─┬─{ModemManager}(6739)

#查看系统用户的进程信息
[root@localhost ~]# pstree -p lisi
sshd(15086)───bash(15089)───vim(15244)
[root@localhost ~]# pstree -pa lisi
sshd,15086
  └─bash,15089
      └─vim,15244 1.txt

#查看系统所有用户的进程
root@localhost ~]# pstree -up
...
        ├─smartd(6726)
        ├─sshd(7337)─┬─sshd(8880)───bash(8887)───pstree(15395)
        │            └─sshd(15066)───sshd(15086,lisi)───bash(15089)───vim(15244)
```

- **ps aux：unix格式静态查看系统进程，查看系统所有进程信息**
  - **a #显示当前终端所有进程**
  - **u #以用户格式输出**
  - **x #当前用户在所有终端下的进程**

- **ps -ef：Linux格式静态查看系统进程，查看系统所有进程信息**
  - **-e　#显示系统所有进程**
  - **-l　#以长格式输出信息**
  - **-f　#显示最完整的进程信息**

```
#查看系统所有进程信息
[root@localhost ~]# ps aux
USER PID %CPU %MEM VSZ　RSS TTY STAT START TIME COMMAND
root　1　2.2　0.3 127992 6576 ？ Ss　09:08 0:01 /usr/lib/systemd/systemd --switched-root
#个字段含义如下：
user：进程属于那个用户
PID ： 进程PID号
%CPU：进程占用CPU资源百分比
%MEM：进程占用物理内存百分比
VSZ ： 进程使用掉的虚拟内存量（单位：Kb）
RSS ： 进程占用固定内存量（单位：Kb）
TTY ： 进程在那个终端运行，如果内核直接调用则显示“？”，tty1-tty6表示本机终端登录的用户进，pts/0-255则表示远程终端登录用户的进程
STAT：进程状态：R（Running）运行，S（Sleep）休眠，s包含子进程，T（stop）停止，Z（Zombe）僵尸，+后台进程
START：进程启动时间
TIME ： 占用CPU运算时间
COMMAND：产生进程的命令

#查看系统所有进程信息
[root@localhost ~]# ps -ef
UID　PID PPID　C STIME TTY　　TIME　CMD
root 1　　0　0 09:08 ？ 00:00:01 /usr/lib/systemd/systemd --switched-root --system --dese
#PPID ： 该进程的父进程ID号
```