第四章：常用命令

尚硅谷云计算 Linux 课程

版本：V1.0

讲师：沈超

一　命令的基本格式

1. 命令的提示符

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

- []：这是提示符的分隔符号，没有特殊含义。
- root：显示的是当前的登录用户，超哥现在使用的是 root 用户登录。
- @：分隔符号，没有特殊含义。
- localhost：当前系统的简写主机名（完整主机名是 localhost.localdomain）。
- ~：代表用户当前所在的目录，此例中用户当前所在的目录是家目录。
- #：命令提示符。超级用户是#，普通用户是$

2. 命令的基本格式

```
[root@localhost ~]# 命令 [选项] [参数]
```

ls 是最常见的目录操作命令，主要作用是显示目录下的内容。

- 命令名称：ls。
- 英文原意：list。
- 所在路径：/bin/ls。
- 执行权限：所有用户。
- 功能描述：显示目录下的内容。

```
[root@localhost ~]#ls [选项] [文件名或目录名]
选项：
   -a:            显示所有文件
   --color=when: 支持颜色输出，when 的值默认是 always（总显示颜色），也可以是
                      never（从不显示颜色）和 auto（自动）
   -d：              显示目录信息，而不是目录下的文件
   -h：              人性化显示，按照我们习惯的单位显示文件大小
   -i：              显示文件的 i 节点号
   -l：              长格式显示
```

举几个例子：

```
[root@localhost ~]# ls -l
总用量 44
-rw-------. 1 root root  1207 1月  14 18:18 anaconda-ks.cfg
#权限     引用计数  所有者 所属组  大小 文件修改时间     文件名
```

我们已经知道“-l”选项用于显示文件的详细信息，那么“-l”选项显示的这 7 列分别是什么含义？

- 第一列：权限。具体权限的含义将在 4.5 节中讲解。
- 第二列：引用计数。文件的引用计数代表该文件的硬链接个数，而目录的引用计数代表该目录有多少个一级子目录。
- 第三列：所有者，也就是这个文件属于哪个用户。默认所有者是文件的建立用户
- 第四列：所属组。默认所属组是文件建立用户的有效组，一般情况下就是建立用户的所在组。
- 第五列：大小。默认单位是字节。
- 第六列：文件修改时间。文件状态修改时间或文件数据修改时间都会更改这个时间，注意这个时间不是文件的创建时间。

第七列：文件名。

选项：是用于调整命令的功能的。

参数：是命令的操作对象，如果省略参数，是因为有默认参数

二　目录操作命令

1. ls 命令

见前一小节的内容。

2. cd 命令

cd 是切换所在目录的命令，这个命令的基本信息如下。

- 命令名称：cd。
- 英文原意：change directory。
- 所在路径：Shell 内置命令。
- 执行权限：所有用户。

功能描述：切换所在目录。

2.1 cd 命令的简化用法

特殊符号	作　用
~	代表用户的家目录
-	代表上次所在目录
.	代表当前目录
..	代表上级目录

2.2 绝对路径和相对路径

绝对路径：以跟目录为参照物，从根目录开始，一级一级进入目录

相对路径：以当前目录作为参照物，进行目录查找

3. pwd 命令

pwd 命令是查询所在目录的命令，基本信息如下：

- 命令名称：pwd
- 英文原意：print name of current/working directory
- 所在路径：/bin/pwd
- 执行权限：所有用户。
- 功能描述：查询所在的工作目录。

4. mkdir 命令

mkdir 是创建目录的命令，其基本信息如下。

- 命令名称：mkdir。
- 英文原意：make directories。
- 所在路径：/bin/mkdir。
- 执行权限：所有用户。
- 功能描述：创建空目录。

命令格式

```
[root@localhost ~]# mkdir [选项] 目录名
选项：
    -p：     递归建立所需目录
```

行。

5. rmdir 命令

既然有建立目录的命令，就一定会有删除目录的命令 rmdir，其基本信息如下。

- 命令名称：rmdir。
- 英文原意：remove empty directories。
- 所在路径：/bin/rmdir。
- 执行权限：所有用户。
- 功能描述：删除空目录。

命令格式

```
[root@localhost ~]# rmdir [选项] 目录名
选项：
    -p：    递归删除目录
```

rmdir 命令的作用十分有限，因为只能删除空目录，所以一旦目录中有内容，就会报错。

这个命令比较“笨”，所以我们不太常用。后续我们不论删除的是文件还是目录，都会使用 rm 命令

三　文件操作命令

1. touch 命令

创建空文件或修改文件时间，这个命令的基本信息如下。

- 命令名称：touch。
- 英文原意：change file timestamps。
- 所在路径：/bin/touch。
- 执行权限：所有用户。
- 功能描述：修改文件的时间戳。

2. stat 命令

stat 是查看文件详细信息的命令，而且可以看到文件的这三个时间，其基本信息如下。

- 命令名称：stat。
- 英文原意：display file or file system status。
- 所在路径：/usr/bin/stat。
- 执行权限：所有用户。

功能描述：显示文件或文件系统的详细信息。

```
[root@localhost ~]# stat  anaconda-ks.cfg
  文件："anaconda-ks.cfg"
  大小：1453          块：8          IO 块：4096   普通文件
设备：803h/2051dInode：33574991    硬链接：1
权限：(0600/-rw-------)  Uid：(    0/    root)   Gid：(    0/    root)
环境：system_u:object_r:admin_home_t:s0
最近访问：2018-11-06 23:22:23.409038121 +0800
最近更改：2018-10-24 00:53:08.760018638 +0800        #数据修改时间
最近改动：2018-10-24 00:53:08.760018638 +0800        #状态修改时间
创建时间：-
```

3. cat 命令

cat 命令用来查看文件内容。这个命令的基本信息如下。

- 命令名称：cat。
- 英文原意：concatenate files and print on the standard output。
- 所在路径：/bin/cat。
- 执行权限：所有用户。

功能描述：合并文件并打印输出到标准输出

命令格式

```
[root@localhost ~]# cat [选项] 文件名
选项：
    -A：相当于-vET 选项的整合，用于列出所有隐藏符号
    -E：列出每行结尾的回车符$
    -n：显示行号
    -T：把 Tab 键用^I 显示出来
    -v：列出特殊字符
```

4. more 命令

more 是分屏显示文件的命令，其基本信息如下。

- 命令名称：more。
- 英文原意：file perusal filter for crt viewin。
- 所在路径：/bin/more。
- 执行权限：所有用户。
- 功能描述：分屏显示文件内容。

more 命令比较简单，一般不用什么选项，命令会打开一个交互界面，可以识别一些交互命令。常用的交互命令如下。

- 空格键：向下翻页。
- b：向上翻页。
- 回车键：向下滚动一行。
- /字符串：搜索指定的字符串。
- q：退出。

5. less 命令

less 命令和 more 命令类似，只是 more 是分屏显示命令，而 less 是分行显示命令，其基本信息如下。

- 命令名称：less。
- 英文原意：opposite of more。
- 所在路径：/usr/bin/less。

- 执行权限：所有用户。
- 功能描述：分行显示文件内容

6. head 命令

head 是用来显示文件开头的命令，其基本信息如下。

- 命令名称：head。
- 英文原意：output the first part of files。
- 所在路径：/usr/bin/head。
- 执行权限：所有用户。
- 功能描述：显示文件开头的内容。

1. 命令格式

```
[root@localhost ~]# head [选项] 文件名
选项：
    -n 行数：      从文件头开始，显示指定行数
    -v：           显示文件名
```

7. tail 命令

既然有显示文件开头的命令，就会有显示文件结尾的命令。tail 命令的基本信息如下。

- 命令名称：tail。
- 英文原意：output the last part of files。
- 所在路径：/usr/bin/tail。
- 执行权限：所有用户。
- 功能描述：显示文件结尾的内容。

命令格式

```
[root@localhost ~]# tail [选项] 文件名
选项：
    -n 行数：      从文件结尾开始，显示指定行数
    -f：           监听文件的新增内容
```

8. ln 命令

我们来看看 ln 命令的基本信息。

- 命令名称：ln。
- 英文原意：make links between file。
- 所在路径：/bin/ln。
- 执行权限：所有用户。
- 功能描述：在文件之间建立链接。

8.1 ln 命令的基本格式如下：

```
[root@localhost ~]# ln [选项] 源文件 目标文件
选项：
    -s：建立软链接文件。如果不加“-s”选项，则建立硬链接文件
    -f：强制。如果目标文件已经存在，则删除目标文件后再建立链接文件
```

如果创建硬链接：

```
[root@localhost ~]# touch cangls
[root@localhost ~]# ln /root/cangls /tmp/
#建立硬链接文件，目标文件没有写文件名，会和原名一致
#也就是/root/cangls 和/tmp/cangls 是硬链接文件
```

如果创建软链接：

```
[root@localhost ~]# touch bols
[root@localhost ~]# ln -s /root/bols  /tmp/
#建立软链接文件
```

8.2 硬链接与软连接的特征

硬链接特征：

- 源文件和硬链接文件拥有相同的 Inode 和 Block
- 修改任意一个文件，另一个都改变
- 删除任意一个文件，另一个都能使用
- 硬链接标记不清，很难确认硬链接文件位置，不建议使用
- 硬链接不能链接目录
- 硬链接不能跨分区

软链接特征：

- 软链接和源文件拥有不同的 Inode 和 Block
- 两个文件修改任意一个，另一个都改变
- 删除软链接，源文件不受影响；删除源文件，软链接不能使用
- 软链接没有实际数据，只保存源文件的 Inode，不论源文件多大，软链接大小不变
- 软链接的权限是最大权限 lrwxrwxrwx.，但是由于没有实际数据，最终访问时需要参考源文件权限
- 软链接可以链接目录
- 软链接可以跨分区
- 软链接特征明显，建议使用软连接

四　目录和文件都能操作的命令

1. rm 命令

rm 是强大的删除命令，不仅可以删除文件，也可以删除目录。这个命令的基本信息如下。

- 命令名称：rm。
- 英文原意：remove files or directories。
- 所在路径：/bin/rm。

- 执行权限：所有用户。
- 功能描述：删除文件或目录。

命令格式

```
[root@localhost ~]# rm [选项] 文件或目录
选项：
    -f：    强制删除（force）
    -i：    交互删除，在删除之前会询问用户
```

-r：　　递归删除，可以删除目录（recursive）

2. cp 命令

cp 是用于复制的命令，其基本信息如下：

- 命令名称：cp。
- 英文原意：copy files and directories。
- 所在路径：/bin/cp。
- 执行权限：所有用户。
- 功能描述：复制文件和目录。

命令格式

```
[root@localhost ~]# cp [选项] 源文件 目标文件
选项：
    -a：    相当于-dpr 选项的集合，这几个选项我们一一介绍
    -d：    如果源文件为软链接（对硬链接无效），则复制出的目标文件也为软链接
    -i：    询问，如果目标文件已经存在，则会询问是否覆盖
    -p：    复制后目标文件保留源文件的属性（包括所有者、所属组、权限和时间）
    -r：    递归复制，用于复制目录
```

3. mv 命令

mv 是用来剪切的命令，其基本信息如下。

- 命令名称：mv。
- 英文原意：move (rename) files。
- 所在路径：/bin/mv。
- 执行权限：所有用户。
- 功能描述：移动文件或改名。

命令格式

```
[root@localhost ~]# mv [选项] 源文件 目标文件
选项：
    -f：    强制覆盖，如果目标文件已经存在，则不询问，直接强制覆盖
    -i：    交互移动，如果目标文件已经存在，则询问用户是否覆盖（默认选项）
    -v：    显示详细信息
```

五　基本权限管理

1．权限的介绍

权限位的含义

前面讲解 ls 命令时，我们已经知道长格式显示的第一列就是文件的权限，例如：

```
[root@localhost ~]# ls -l install.log
-rw-r--r--. 1 root root 24772 1月  14 18:17 install.log
```

第一列的权限位如果不计算最后的“.”（这个点的含义我们在后面解释），则共有 10 位，这 10 位权限位的含义如图 4-4 所示。

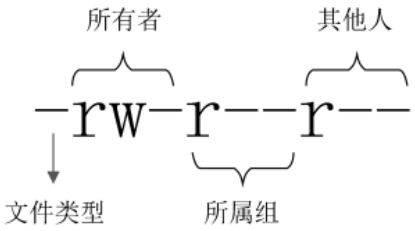


图 4-4　权限位的含义

- 第 1 位代表文件类型。Linux 不像 Windows 使用扩展名表示文件类型，而是使用权限位的第 1 位表示文件类型。虽然 Linux 文件的种类不像 Windows 中那么多，但是分类也不少，详细情况可以使用“info ls”命令查看。超哥在这里只讲一些常见的文件类型。
  - “-”：普通文件。
  - “b”：块设备文件。这是一种特殊设备文件，存储设备都是这种文件，如分区文件/dev/sda1 就是这种文件。
  - “c”：字符设备文件。这也是特殊设备文件，输入设备一般都是这种文件，如鼠标、键盘等。
  - “d”：目录文件。Linux 中一切皆文件，所以目录也是文件的一种。
  - “l”：软链接文件。
  - “p”：管道符文件。这是一种非常少见的特殊设备文件。
  - “s”：套接字文件。这也是一种特殊设备文件，一些服务支持 Socket 访问，就会产生这样的文件。
- 第 2～4 位代表文件所有者的权限。
  - r：代表 read，是读取权限。
  - w：代表 write，是写权限。
  - x：代表 execute，是执行权限。

如果有字母，则代表拥有对应的权限；如果是“-”，则代表没有对应的权限。

- 第 5～7 位代表文件所属组的权限，同样拥有“rwx”权限。
- 第 8～10 位代表其他人的权限，同样拥有“rwx”权限。

2. 基本权限命令

首先来看修改权限的命令 chmod，其基本信息如下。

- 命令名称：chmod。
- 英文原意：change file mode bits。
- 所在路径：/bin/chmod。
- 执行权限：所有用户。
- 功能描述：修改文件的权限模式。

2.1. 命令格式

```
[root@localhost ~]# chmod [选项] 权限模式 文件名
选项：
    -R：    递归设置权限，也就是给子目录中的所有文件设定权限
```

2.2. 权限模式

chmod 命令的权限模式的格式是“[ugoa][[+-=][perms]]”，也就是“[用户身份][[赋予方式][权限]]”的格式，我们来解释一下。

- 用户身份。
  - u：代表所有者（user）。
  - g：代表所属组（group）。
  - o：代表其他人（other）。
  - a：代表全部身份（all）。
- 赋予方式。
  - +：加入权限。
  - -：减去权限。
  - =：设置权限。
- 权限。
  - r：读取权限（read）。
  - w：写权限（write）。
  - x：执行权限（execute）。

2.3. 数字权限

数字权限的赋予方式是最简单的，但是不如之前的字母权限好记、直观。我们来看看这些数字权限的含义。

- 4：代表“r”权限。
- 2：代表“w”权限。
- 1：代表“x”权限。

2.4. 常用权限

数字权限的赋予方式更加简单，但是需要用户对这几个数字更加熟悉。其实常用权限也并不多，只有如下几个。

- 644：这是文件的基本权限，代表所有者拥有读、写权限，而所属组和其他人拥有只读权限。

- 755：这是文件的执行权限和目录的基本权限，代表所有者拥有读、写和执行权限，而所属组和其他人拥有读和执行权限。
- 777：这是最大权限。在实际的生产服务器中，要尽力避免给文件或目录赋予这样的权限，这会造成一定的安全隐患。

3. 基本权限的作用

3.1. 权限含义的解释

首先，读、写、执行权限对文件和目录的作用是不同的。

- 权限对文件的作用。
  - 读（r）：对文件有读（r）权限，代表可以读取文件中的数据。如果把权限对应到命令上，那么一旦对文件有读（r）权限，**就可以对文件执行 cat、more、less、head、tail 等文件查看命令。**
  - 写（w）：对文件有写（w）权限，代表可以修改文件中的数据。如果把权限对应到命令上，那么一旦对文件有写（w）权限，就可以对文件执行 **vim、echo 等修改文件数据的命令。注意：对文件有写权限，是不能删除文件本身的，只能修改文件中的数据。如果要想删除文件，则需要对文件的上级目录拥有写权限。**
  - 执行（x）：对文件有执行（x）权限，代表文件拥有了执行权限，可以运行。在 Linux 中，只要文件有执行（x）权限，这个文件就是执行文件了。只是这个文件到底能不能正确执行，不仅需要执行（x）权限，还要看文件中的代码是不是正确的语言代码。对文件来说，执行（x）权限是最高权限。

- 权限对目录的作用。
  - 读（r）：对目录有读（r）权限，代表可以查看目录下的内容，也就是可以查看目录下有哪些子文件和子目录。如果把权限对应到命令上，那么一旦对目录拥有了读（r）权限，就可以在目录下执行 **ls 命令**，查看目录下的内容了。
  - 写（w）：对目录有写（r）权限，代表可以修改目录下的数据，也就是可以在目录中新建、删除、复制、剪切子文件或子目录。如果把权限对应到命令上，那么一旦对目录拥有了写（w）权限，就可以在目录下执行 **touch、rm、cp、mv 命令。**对目录来说，写（w）权限是最高权限。
  - 执行（x）：目录是不能运行的，那么对目录拥有执行（x）权限，代表可以进入目录。如果把权限对应到命令上，那么一旦对目录拥有了执行（x）权限，就可以对目录执行 **cd 命令**，进入目录。

3.1. 目录的可用权限

目录的可用权限其实只有以下几个。

- 0：任何权限都不赋予。
- 5：基本的目录浏览和进入权限。
- 7：完全权限。

4．所有者和所属组命令

4.1．chown 命令

chown 是修改文件和目录的所有者和所属组的命令，其基本信息如下。

- 命令名称：chown。
- 英文原意：change file owner and group。
- 所在路径：/bin/chown。
- 执行权限：所有用户。
- 功能描述：修改文件和目录的所有者和所属组。

1）命令格式

```
[root@localhost ~]# chown [选项] 所有者:所属组 文件或目录
选项：
    -R：      递归设置权限，也就是给子目录中的所有文件设置权限
```

普通用户不能修改文件的所有者，哪怕自己是这个文件的所有者也不行。

普通用户可以修改所有者是自己的文件的权限。

4.2．chgrp 命令

chgrp 是修改文件和目录的所属组的命令，其基本信息如下。

- 命令名称：chgrp。
- 英文原意：change group ownership。
- 所在路径：/bin/chgrp。
- 执行权限：所有用户。
- 功能描述：修改文件和目录的所属组。

5．umask 默认权限

5.1 查看系统的 umask 权限

```
[root@localhost ~]# umask
0022
#用八进制数值显示 umask 权限
[root@localhost ~]# umask -S
u=rwx,g=rx,o=rx
#用字母表示文件和目录的初始权限
```

.2　umask 权限的计算方法

我们需要先了解一下新建文件和目录的默认最大权限。

- 对文件来讲，新建文件的默认最大权限是 666，没有执行（x）权限。这是因为执行权限对文件来讲比较危险，不能在新建文件的时候默认赋予，而必须通过用户手工赋予。
- 对目录来讲，新建目录的默认最大权限是 777。这是因为对目录而言，执行（x）权限仅仅代表进入目录，所以即使建立新文件时直接默认赋予，也没有什么危险。

按照官方的标准算法，umask 默认权限需要使用二进制进行逻辑与和逻辑非联合运算才可以得到正确的新建文件和目录的默认权限。这种方法既不好计算，也不好理解，超哥并不推荐。

我们在这里还是按照权限字母来讲解 umask 权限的计算方法。我们就按照默认的 umask 值是 022 来分别计算一下新建文件和目录的默认权限吧。

- 文件的默认权限最大只能是 666，而 umask 的值是 022
  “-rw-rw-rw-”减去 “-----w--w-”等于“-rw-r--r—”
- 目录的默认权限最大可以是 777，而 umask 的值是 022
  “drwxrwxrwx”减去“d----w--w-”等于“drwx-r-xr-x”

注意：umask 默认权限的计算绝不是数字直接相减。

例如 umask 是 033 呢？

- 文件的默认权限最大只能是 666，而 umask 的值是 033

“-rw-rw-rw-”减去“-----wx-wx”等于“-rw-r--r—”

六　帮助命令

1．man 命令

man 是最常见的帮助命令，也是 Linux 最主要的帮助命令，其基本信息如下。

- 命令名称：man。
- 英文原意：format and display the on-line manual pages。
- 所在路径：/usr/bin/man。
- 执行权限：所有用户。
- 功能描述：显示联机帮助手册。

1.1．命令格式

```
[root@localhost ~]# man [选项] 命令
选项：
    -f：    查看命令拥有哪个级别的帮助
    -k：    查看和命令相关的所有帮助
```

1.2．man 命令的快捷键

快 捷 键	作　用
上箭头	向上移动一行
下箭头	向下移动一行
PgUp	向上翻一页
PgDn	向下翻一页
g	移动到第一页

G	移动到最后一页
q	退出
/字符串	从当前页向下搜索字符串
?字符串	从当前页向上搜索字符串
n	当搜索字符串时，可以使用 n 键找到下一个字符串
N	当搜索字符串时，使用 N 键反向查询字符串。也就是说，如果使用“/字符串”方式搜索，则 N 键表示向上搜索字符串；如果使用“?字符串”方式搜索，则 N 键表示向下搜索字符串

1.3. man 命令的帮助级别

级　别	作　用
1	普通用户可以执行的系统命令和可执行文件的帮助
2	内核可以调用的函数和工具的帮助
3	C 语言函数的帮助
4	设备和特殊文件的帮助
5	配置文件的帮助
6	游戏的帮助（个人版的 Linux 中是有游戏的）
7	杂项的帮助
8	超级用户可以执行的系统命令的帮助
9	内核的帮助

man -f 命令 或 whatis 命令
#查看命令拥有哪个级别的帮助

man -k 命令 或 apropos 命令
#查看和命令相关的所有帮助

2. info 命令

info 命令的帮助信息是一套完整的资料，每个单独命令的帮助信息只是这套完整资料中的某一个小章节。

快 捷 键	作　用
上箭头	向上移动一行
下箭头	向下移动一行
PgUp	向上翻一页
PgDn	向下翻一页
Tab	在有“*”符号的节点间进行切换
回车	进入有“*”符号的子页面，查看详细帮助信息
u	进入上一层信息（回车是进入下一层信息）
n	进入下一小节信息
p	进入上一小节信息
?	查看帮助信息
q	退出 info 信息

3. help 命令

help 只能获取 Shell 内置命令的帮助

help 命令的基本信息如下。

- 命令名称：help。
- 英文原意：help。
- 所在路径：Shell 内置命令。
- 执行权限：所有用户。

功能描述：显示 Shell 内置命令的帮助。可以使用 type 命令来区分内置命令与外部命令

shell 是 Linux 的命令解释器。

4. --help 选项

绝大多数命令都可以使用“--help”选项来查看帮助，这也是一种获取帮助的方法。例如：

```
[root@localhost ~]# ls --help
```

这种方法非常简单，输出的帮助信息基本上是 man 命令的信息简要版。

对于这 4 种常见的获取帮助的方法，大家可以按照自己的习惯任意使用。

七　搜索命令

1. whereis 命令

whereis 是搜索系统命令的命令（像绕口令一样），也就是说，whereis 命令不能搜索普通文件，而只能搜索系统命令。whereis 命令的基本信息如下。

- 命令名称：whereis。
- 英文原意：locate the binary, source, and manual page files for a command。
- 所在路径：/usr/bin/whereis。
- 执行权限：所有用户。
- 功能描述：查找二进制命令、源文件和帮助文档的命令。

2. which 命令

which 也是搜索系统命令的命令。和 whereis 命令的区别在于：

- whereis 命令可以在查找到二进制命令的同时，查找到帮助文档的位置；
- 而 which 命令在查找到二进制命令的同时，如果这个命令有别名，则还可以找到别名命令。

which 命令的基本信息如下。

命令名称：which。

- 英文原意：shows the full path of (shell) commands。
- 所在路径：/usr/bin/which。
- 执行权限：所有用户。
- 功能描述：列出命令的所在路径。

3. locate 命令

3.1 基本用法

locate 命令才是可以按照文件名搜索普通文件的命令。

- 优点：按照数据库搜索，搜索速度快，消耗资源小。数据库位置/var/lib/mlocate/mlocate.db，可以使用 updatedb 命令强制更新数据库。
- 缺点：只能按照文件名来搜索文件，而不能执行更复杂的搜索，比如按照权限、大小、修改时间等搜索文件。

locate 命令的基本信息如下。

- 命令名称：locate。
- 英文原意：find files by name。
- 所在路径：/usr/bin/locate。
- 执行权限：所有用户。
- 功能描述：按照文件名搜索文件。

3.2 配置文件

```
[root@localhost ~]# vi /etc/updatedb.conf
PRUNE_BIND_MOUNTS = "yes"
#开启搜索限制，也就是让这个配置文件生效
PRUNEFS = "……"
#在 locate 执行搜索时，禁止搜索这些文件系统类型
PRUNENAMES = "……"
#在 locate 执行搜索时，禁止搜索带有这些扩展名的文件
PRUNEPATHS = "……"
#在 locate 执行搜索时，禁止搜索这些系统目录
```

4. find 命令

find 命令的基本信息如下。

- 命令名称：find。
- 英文原意：search for files in a directory hierarchy。
- 所在路径：/bin/find。
- 执行权限：所有用户。

- 功能描述：在目录中搜索文件。

4.1 按照文件名搜索

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -name：      按照文件名搜索
    -iname：     按照文件名搜索，不区分文件名大小写
    -inum：      按照 inode 号搜索
```

4.2. 按照文件大小搜索

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -size [+|-]大小：        按照指定大小搜索文件
```

这里的“+”的意思是搜索比指定大小还要大的文件，“-”的意思是搜索比指定大小还要小的文件。

find 命令的单位：

```
[root@localhost ~]# man find
      -size n[cwbkMG]
            File uses n units of space.  The following suffixes can be used:

            'b'    for 512-byte blocks (this is the default if no suffix is used)
               #这是默认单位，如果单位为 b 或不写单位，则按照 512 Byte 搜索
            'c'    for bytes
               #搜索单位是 c，按照字节搜索
            'w'    for two-byte words
               #搜索单位是 w，按照双字节（中文）搜索
            'k'    for Kilobytes (units of 1024 bytes)
               #按照 KB 单位搜索，必须是小写的 k
            'M'    for Megabytes (units of 1048576 bytes)
               #按照 MB 单位搜索，必须是大写的 M
            'G'    for Gigabytes (units of 1073741824 bytes)
               #按照 GB 单位搜索，必须是大写的 G
```

4.3 按照修改时间搜索

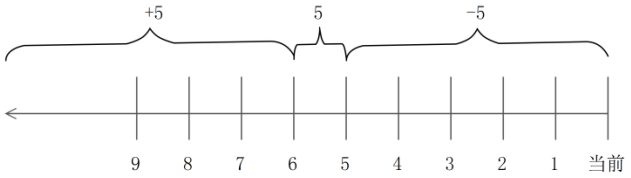
Linux 中的文件有访问时间（atime）、数据修改时间（mtime）、状态修改时间（ctime）这三个时间，我们也可以按照时间来搜索文件。

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -atime [+|-]时间：   按照文件访问时间搜索
    -mtime [+|-]时间：   按照文件数据修改时间搜索
    -ctime [+|-]时间：   按照文件状态修改时间搜索
```

这三个时间的区别我们在 stat 命令中已经解释过了，这里用 mtime 数据修改时间来举例，重点说说“[+-]”时间的含义。

- -5：代表 5 天内修改的文件。
- 5：代表前 5～6 天那一天修改的文件。
- +5：代表 6 天前修改的文件。

我们画一个时间轴，来解释一下，如图 4-6 所示。



4.4 按照权限搜索

命令格式。

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -perm 权限模式：  查找文件权限刚好等于“权限模式”的文件
    -perm -权限模式：查找文件权限全部包含“权限模式”的文件
    -perm +权限模式：查找文件权限包含“权限模式”的任意一个权限的文件
```

4.5 按照所有者和所属组搜索

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -uid 用户 ID：        按照用户 ID 查找所有者是指定 ID 的文件
    -gid 组 ID：          按照用户组 ID 查找所属组是指定 ID 的文件
    -user 用户名：        按照用户名查找所有者是指定用户的文件
    -group 组名：         按照组名查找所属组是指定用户组的文件
    -nouser：             查找没有所有者的文件
```

按照所有者和所属组搜索时，“-nouser”选项比较常用，主要用于查找垃圾文件。

只有一种情况例外，那就是外来文件。比如光盘和 U 盘中的文件如果是由 Windows 复制的，在 Linux 中查看就是没有所有者的文件；再比如手工源码包安装的文件，也有可能没有所有者

4.6 按照文件类型搜索

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
选项：
    -type d：        查找目录
    -type f：        查找普通文件
    -type l：        查找软链接文件
```

4.7 逻辑运算符

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容
```

```
选项：
    -a：      and 逻辑与
    -o：      or 逻辑或
    -not：    not 逻辑非
```

1）-a：and 逻辑与

find 命令也支持逻辑运算符选项，其中-a 代表逻辑与运算，也就是-a 的两个条件都成立，find 搜索的结果才成立。举个例子：

```
[root@localhost ~]# find . -size +2k -a -type f
#在当前目录下搜索大于 2KB，并且文件类型是普通文件的文件
```

2）-o：or 逻辑或

-o 选项代表逻辑或运算，也就是-o 的两个条件只要其中一个成立，find 命令就可以找到结果。例如：

```
[root@localhost ~]# find . -name cangls -o -name bols
./cangls
./bols
#在当前目录下搜索文件名要么是 cangls 的文件，要么是 bols 的文件
```

3）-not：not 逻辑非

-not 是逻辑非，也就是取反的意思。举个例子：

```
[root@localhost ~]# find . -not -name cangls
#在当前目录下搜索文件名不是 cangls 的文件
```

4.8 其他选项

1）-exec 选项

这里我们主要讲解两个选项“-exec”和“-ok”，这两个选项的基本作用非常相似。我们先来看看“-exec”选项的格式。

```
[root@localhost ~]# find 搜索路径 [选项] 搜索内容 -exec 命令 2 {} \;
```

其次，这个选项的作用其实是把 find 命令的结果交给由“-exec”调用的命令 2 来处理。“{}”就代表 find 命令的查找结果。

2）-ok 选项

“-ok”选项和“-exec”选项的作用基本一致，区别在于：“-exec”的命令 2 会直接处理，而不询问；“-ok”的命令 2 在处理前会先询问用户是否这样处理，在得到确认命令后，才会执行。

5. grep 命令：补充命令

grep 的作用是在文件中提取和匹配符合条件的字符串行。命令格式如下：

```
[root@localhost ~]# grep [选项] "搜索内容" 文件名
选项：
    -i：            忽略大小写
```

```
-n：            输出行号
-v：            反向查找
--color=auto:   搜索出的关键字用颜色显示
```

find 也是搜索命令，那么 find 命令和 grep 命令有什么区别呢？

1）find 命令

find 命令用于在系统中搜索符合条件的文件名，如果需要模糊查询，则使用通配符进行匹配，通配符是完全匹配（find 命令可以通过-regex 选项，把匹配规则转为正则表达式规则，但是不建议如此）。

2）grep 命令

grep 命令用于在文件中搜索符合条件的字符串，如果需要模糊查询，则使用正则表达式进行匹配，正则表达式是包含匹配。

3）通配符与正则表达式的区别

通配符：用于匹配文件名，完全匹配

通 配 符	作　用
?	匹配一个任意字符
*	匹配 0 个或任意多个任意字符，也就是可以匹配任何内容
[]	匹配中括号中任意一个字符。例如，[abc]代表一定匹配一个字符，或者是 a，或者是 b，或者是 c
[-]	匹配中括号中任意一个字符，-代表一个范围。例如，[a-z]代表匹配一个小写字母
[^]	逻辑非，表示匹配不是中括号内的一个字符。例如，[^0-9]代表匹配一个不是数字的字符

正则表达式：用于匹配字符串，包含匹配

正 则 符	作　用
?	匹配前一个字符重复 0 次，或 1 次（?是扩展正则，需要使用 egrep 命令）
*	匹配前一个字符重复 0 次，或任意多次
[]	匹配中括号中任意一个字符。例如，[abc]代表一定匹配一个字符，或者是 a，或者是 b，或者是 c
[-]	匹配中括号中任意一个字符，-代表一个范围。例如，[a-z]代表匹配一个小写字母
[^]	逻辑非，表示匹配不是中括号内的一个字符。例如，[^0-9]代表匹配一个不是数字的字符
^	匹配行首
$	匹配行尾

6. 管道符：补充命令

命令格式：　　命令 1 | 命令 2

命令 1 的正确输出作为命令 2 的操作对象

1）例子 1：

举个例子，我们经常需要使用“ll”命令查看文件的长格式，不过在有些目录中文件众多，比如/etc/

目录，使用“ll”命令显示的内容就会非常多，只能看到最后的内容，而不能看到前面输出的内容。这时我们马上想到 more 命令可以分屏显示文件内容，可是怎么让 more 命令分屏显示命令的输出呢？我想到了一种笨办法：

```
[root@localhost ~]# ll -a /etc/ > /root/testfile
#用输出重定向，把 ll 命令的输出保存到/root/testfile 文件中
[root@localhost ~]# more /root/testfile
#既然 testfile 是文件，当然可以用 more 命令分屏显示了
总用量 1784
drwxr-xr-x. 105 root root  12288 10 月 21 12:49 .
dr-xr-xr-x.  26 root root   4096 6 月   5 19:06 ..
...省略部分输出...
-rwxr-xr-x.   1 root root    687 6 月  22 2012 auto.smb
--More--(7%)
```

可是这样操作实在不方便，这时就可以利用管道符了。命令如下：

```
[root@localhost ~]# ll -a /etc/  | more
```

2）例子 2：

我想在命令 ll /etc/的结果中搜索 yum 的文件名，应该使用 find 命令？还是 grep 命令？

```
[root@localhost ~]# ll -a /etc/  | grep  yum
```

3）例子 3：

netstat 命令（CentOS 7 中，需要安装 net-snmp.x86_64，net-tools.x86_64 两个包才有此命令。7.5 系统中已经自动安装）格式如下：

```
[root@localhost ~]# netstat [选项]
选项：
    -a：          列出所有网络状态，包括 Socket 程序
    -c 秒数：     指定每隔几秒刷新一次网络状态
    -n：          使用 IP 地址和端口号显示，不使用域名与服务名
    -p：          显示 PID 和程序名
    -t：          显示使用 TCP 协议端口的连接状况
    -u：          显示使用 UDP 协议端口的连接状况
    -l：          仅显示监听状态的连接
    -r：          显示路由表
```

```
[root@localhost ~]# netstat -an | grep "ESTABLISHED" | wc -l
#如果想知道具体的网络连接数量，就可以再使用 wc 命令统计行数
```

统计正在连接的网络连接数量

7. 命令的别名：补充命令

命令的别名，就是命令的小名，主要是用于照顾管理员使用习惯的。

```
命令格式：
[root@localhost ~]# alias
#查询命令别名
[root@localhost ~]# alias 别名='原命令'
#设定命令别名
例如：
[root@localhost ~]# alias ser='service network restart'
#用 ser 别名，替代 service network restart 命令
```

用命令定义的别名，是临时生效的，要想永久生效，需要写入环境变量配置文件~/.bashrc

8. 常用快捷键：补充命令

快捷键	作　　用
Tab 键	命令或文件补全
ctrl+A	把光标移动到命令行开头。如果我们输入的命令过长，想要把光标移动到命令行开头时使用。
ctrl+E	把光标移动到命令行结尾。
ctrl+C	强制终止当前的命令。
ctrl+L	清屏，相当于 clear 命令。
ctrl+U	删除或剪切光标之前的命令。我输入了一行很长的命令，不用使用退格键一个一个字符的删除，使用这个快捷键会更加方便
ctrl+Y	粘贴 ctrl+U 剪切的内容。

八　压缩和解压缩命令

在 Linux 中可以识别的常见压缩格式有十几种，比如“.zip”“.gz”“.bz2”“.tar”“.tar.gz”“.tar.bz2”等。

1. “.zip”格式

“.zip”是 Windows 中最常用的压缩格式，Linux 也可以正确识别“.zip”格式，这可以方便地和 Windows 系统通用压缩文件。

1.1. “.zip”格式的压缩命令

压缩命令就是 zip，其基本信息如下。

- 命令名称：zip。
- 英文原意：package and compress (archive) files。
- 所在路径：/usr/bin/zip。
- 执行权限：所有用户。

- 功能描述：压缩文件或目录。

命令格式如下：

```
[root@localhost ~]# zip [选项] 压缩包名 源文件或源目录
选项：
    -r：    压缩目录
例如：
[root@localhost ~]# zip ana.zip anaconda-ks.cfg
```

1.2. “.zip”格式的解压缩命令

“.zip”格式的解压缩命令是 unzip，其基本信息如下。

- 命令名称：unzip。
- 英文原意：list, test and extract compressed files in a ZIP archive。
- 所在路径：/usr/bin/unzip。
- 执行权限：所有用户。
- 功能描述：列表、测试和提取压缩文件中的文件。

命令格式如下：

```
[root@localhost ~]# unzip [选项] 压缩包名
选项：
    -d：    指定解压缩位置
例如：
[root@localhost ~]# unzip -d /tmp/ ana.zip
#把压缩包解压到指定位置
```

2. “.gz”格式　　不会打包

2.1. “.gz”格式的压缩命令

“.gz”格式是 Linux 中最常用的压缩格式，使用 gzip 命令进行压缩，其基本信息如下。

- 命令名称：gzip。
- 英文原意：compress or expand files。
- 所在路径：/bin/gzip。
- 执行权限：所有用户。
- 功能描述：压缩文件或目录。

这个命令的格式如下：

```
[root@localhost ~]# gzip [选项] 源文件
选项：
    -c：    将压缩数据输出到标准输出中，可以用于保留源文件
    -d：    解压缩
    -r：    压缩目录
```

```
[root@localhost ~]# gzip -c anaconda-ks.cfg > anaconda-ks.cfg.gz
#使用-c 选项，但是不让压缩数据输出到屏幕上，而是重定向到压缩文件中
#这样可以在压缩文件的同时不删除源文件
```

2. “.gz”格式的解压缩命令

如果要解压缩“.gz”格式，那么使用“gzip -d 压缩包”和“gunzip 压缩包”命令都可以。我们先看看 gunzip 命令的基本信息。

- 命令名称：gunzip。
- 英文原意：compress or expand files。
- 所在路径：/bin/gunzip。
- 执行权限：所有用户。
- 功能描述：解压缩文件或目录。

```
例如：
[root@localhost ~]# gunzip install.log.gz
[root@localhost ~]# gzip -d anaconda-ks.cfg.gz
```

两个命令都可以解压缩“.gz”格式

3. “.bz2”格式　　不能压缩目录

3.1. “.bz2”格式的压缩命令

“.bz2”格式是 Linux 的另一种压缩格式，从理论上来讲，“.bz2”格式的算法更先进、压缩比更好；而“.gz”格式相对来讲压缩的时间更快。

“.bz2”格式的压缩命令是 bzip2，我们来看看这个命令的基本信息。

- 命令名称：bzip2。
- 英文原意：a block-sorting file compressor。
- 所在路径：/usr/bin/bzip2。
- 执行权限：所有用户。
- 功能描述：.bz2 格式的压缩命令。

来看看 bzip2 命令的格式。

```
[root@localhost ~]# bzip2 [选项] 源文件
选项：
    -d：    解压缩
    -k：    压缩时，保留源文件
    -v：    显示压缩的详细信息
例如：
[root@localhost ~]# bzip2 anaconda-ks.cfg
#压缩成.bz2 格式
[root@localhost ~]# bzip2 -k install.log.syslog
#保留源文件压缩
```

3.2. “.bz2”格式的解压缩命令

“.bz2”格式可以使用“bzip2 -d 压缩包”命令来进行解压缩，也可以使用“bunzip2 压缩包”命令来进行解压缩。先看看 bunzip2 命令的基本信息。

- 命令名称：bunzip2。
- 英文原意：a block-sorting file compressor。
- 所在路径：/usr/bin/bunzip2。
- 执行权限：所有用户。
- 功能描述：.bz2 格式的解压缩命令。

```
[root@localhost ~]# bunzip2 anaconda-ks.cfg.bz2
[root@localhost ~]# bzip2 -d install.log.syslog.bz2
#两个命令都可以解压缩
```

4. “.tar”格式　　打包不会压缩

4.1. “.tar”格式的打包命令

“.tar”格式的打包和解打包都使用 tar 命令，区别只是选项不同。我们先看看 tar 命令的基本信息。

- 命令名称：tar。
- 英文原意：tar。
- 所在路径：/bin/tar。
- 执行权限：所有用户。
- 功能描述：打包与解打包命令。

命令的基本格式如下：

```
[root@localhost ~]# tar [选项] [-f 压缩包名] 源文件或目录
选项：
    -c：    打包
    -f：    指定压缩包的文件名。压缩包的扩展名是用来给管理员识别格式的，所以一定
            要正确指定扩展名
    -v：    显示打包文件过程
```

```
[root@localhost ~]# tar -cvf anaconda-ks.cfg.tar anaconda-ks.cfg
#打包，不会压缩
```

4.2. “.tar”格式的解打包命令

“.tar”格式的解打包也需要使用 tar 命令，但是选项不太一样。命令格式如下：

```
[root@localhost ~]# tar [选项] 压缩包
选项：
    -x：        解打包
    -f：        指定压缩包的文件名
    -v：        显示解打包文件过程
    -t：        测试，就是不解打包，只是查看包中有哪些文件
```

```
    -C(大) 目录：指定解打包位置
```

```
例如
[root@localhost ~]# tar -xvf anaconda-ks.cfg.tar
#解打包到当前目录下
```

5. “.tar.gz”和“.tar.bz2”格式

使用 tar 命令直接打包压缩。命令格式如下：

```
[root@localhost ~]# tar [选项] 压缩包 源文件或目录
选项：
    -z：     压缩和解压缩“.tar.gz”格式
    -j：     压缩和解压缩“.tar.bz2”格式
```

```
例如：.tar.gz 格式
[root@localhost ~]# tar -zcvf tmp.tar.gz /tmp/
#把/tmp/目录直接打包压缩为“.tar.gz”格式
[root@localhost ~]# tar -zxvf tmp.tar.gz
#解压缩与解打包“.tar.gz”格式
```

```
例如：.tar.bz2 格式
[root@localhost ~]# tar -jcvf tmp.tar.bz2 /tmp/
#打包压缩为“.tar.bz2”格式，注意压缩包文件名
[root@localhost ~]# tar -jxvf tmp.tar.bz2
#解压缩与解打包“.tar.bz2”格式
```

再举几个例子：

```
[root@localhost ~]# mkdir test
[root@localhost ~]# touch  test/abc
[root@localhost ~]# touch  test/bcd
[root@localhost ~]# touch  test/cde
#建立测试目录和测试文件

[root@localhost ~]# tar -zcvf  test.tar.gz  test/
#压缩
[root@localhost ~]# tar  -ztvf  test.tar.gz
#只查看，不解压
[root@localhost ~]# tar  -zxvf  test.tar.gz -C /tmp
#解压缩到指定位置
[root@localhost ~]# tar  -zxvf  test.tar.gz -C /tmp  test/cde
#只解压压缩包中的特定文件，到指定位置
```

九　关机和重启命令

1. sync 数据同步

sync 命令的基本信息如下。

- 命令名称：sync。
- 英文原意：flush file system buffers。
- 所在路径：/bin/sync。
- 执行权限：所有用户。
- 功能描述：刷新文件系统缓冲区。

2. shutdown 命令

shutdown 命令的基本信息如下。

- 命令名称：shutdown。
- 英文原意：bring the system down。
- 所在路径：/sbin/shutdown。
- 执行权限：超级用户。
- 功能描述：关机和重启

命令的基本格式如下：

```
[root@localhost ~]# shutdown [选项] 时间 [警告信息]
选项：
    -c：     取消已经执行的 shutdown 命令
    -h：     关机
    -r：     重启
```

3. reboot 命令

在现在的系统中，reboot 命令也是安全的，而且不需要加入过多的选项。

```
[root@localhost ~]# reboot
#重启
```

4. halt 和 poweroff 命令

这两个都是关机命令，直接执行即可。这两个命令不会完整关闭和保存系统的服务，不建议使用。

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

5. init 命令

init 是修改 Linux 运行级别的命令，也可以用于关机和重启。这个命令并不安全，不建议使用。

```
[root@localhost ~]# init 0
```

```
#关机，也就是调用系统的 0 级别
[root@localhost ~]# init 6
#重启，也就是调用系统的 6 级别
```

十　常用网络命令

1. 配置 IP 地址

1.1　配置 IP 地址

IP 地址是计算机在互联网中唯一的地址编码。每台计算机如果需要接入网络和其他计算机进行数据通信，就必须配置唯一的公网 IP 地址。

配置 IP 地址有两种方法：

1）setup 工具

2）vi　/etc/sysconfig/network-scripts/ifcfg-eth0

手工修改配置文件

1.2　重启网络服务

```
[root@localhost ~]# service  network  restart
#重启网络服务
```

1.3　虚拟机需要桥接到有线网卡，病重启网络服务

1.4　复制镜像有可能需要重置 UUID（唯一识别符）

```
[root@localhost ~]# vi  /etc/sysconfig/network-scripts/ifcfg-eth0
#删除 MAC 地址行

[root@localhost ~]# rm  -rf  /etc/udev/rules.d/70-persistent-net.rules
#删除 MaC 地址和 UUID 绑定文件

[root@localhost ~]# reboot
#重启 Linux
```

2. ifconfig 命令

- 命令名称：ifconfig。
- 英文原意：configure a network interface。
- 所在路径：/sbin/ifconfig。
- 执行权限：超级用户。

- 功能描述：配置网络接口。

ifconfig 命令最主要的作用就是查看 IP 地址的信息，直接输入 ifconfig 命令即可。

```
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        #标志                                          最大传输单元
        inet 192.168.252.20  netmask 255.255.255.0  broadcast 192.168.252.255
        #IP 地址             子网掩码                  广播地址
        inet6 fe80::546e:994b:30c:e2f7  prefixlen 64  scopeid 0x20<link>
        #IPv6 地址（目前没有生效）
        ether 00:0c:29:aa:d2:96  txqueuelen 1000  (Ethernet)
        #MAC 地址
        RX packets 3728  bytes 310958 (303.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        #接收的数据包情况
        TX packets 3051  bytes 1495119 (1.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        #发送的数据包情况

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
#本地回环网卡
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 696 (696.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 696 (696.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. ping 命令

ping 是常用的网络命令，主要通过 ICMP 协议进行网络探测，测试网络中主机的通信情况。ping 命令的基本信息如下。

- 命令名称：ping。
- 英文原意：send ICMP ECHO_REQUEST to network hosts。
- 所在路径：/bin/ping。
- 执行权限：所有用户。
- 功能描述：向网络主机发送 ICMP 请求。

命令的基本格式如下：

```
[root@localhost ~]# ping [选项] IP
选项：
    -b：          后面加入广播地址，用于对整个网段进行探测
    -c 次数：     用于指定 ping 的次数
    -s 字节：     指定探测包的大小
```

例子：探测网段中的可用主机

在 ping 命令中，可以使用“-b”选项，后面加入广播地址，探测整个网段。我们可以使用这个选

项知道整个网络中有多少主机是可以和我们通信的，而不用一个一个 IP 地址地进行探测。例如：

```
[root@localhost ~]# ping -b -c 3 192.168.103.255
WARNING: pinging broadcast address
PING 192.168.103.255 (192.168.103.255) 56(84) bytes of data.
64 bytes from 192.168.103.199: icmp_seq=1 ttl=64 time=1.95 ms
64 bytes from 192.168.103.168: icmp_seq=1 ttl=64 time=1.97 ms (DUP!)
64 bytes from 192.168.103.252: icmp_seq=1 ttl=64 time=2.29 ms (DUP!)
…省略部分内容…
```

#探测 192.168.103.0/24 网段中有多少可以通信的主机

4. netstat 命令

netstat 是网络状态查看命令，既可以查看到本机开启的端口，也可以查看有哪些客户端连接。在 CentOS 7.x 中 netstat 命令默认没有安装，如果需要使用，需要安装 net-snmp 和 net-tools 软件包。

netstat 命令的基本信息如下。

- 命令名称：netstat。
- 英文原意：Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships。
- 所在路径：/bin/netstat。
- 执行权限：所有用户。
- 功能描述：输出网络连接、路由表、接口统计、伪装连接和组播成员。

命令格式如下：

```
[root@localhost ~]# netstat [选项]
选项：
    -a：          列出所有网络状态，包括 Socket 程序
    -c 秒数：     指定每隔几秒刷新一次网络状态
    -n：          使用 IP 地址和端口号显示，不使用域名与服务名
    -p：          显示 PID 和程序名
    -t：          显示使用 TCP 协议端口的连接状况
    -u：          显示使用 UDP 协议端口的连接状况
    -l：          仅显示监听状态的连接
    -r：          显示路由表
```

例子 1：查看本机开启的端口

这是本机最常用的方式，使用选项“-tuln”。因为使用了“-l”选项，所以只能看到监听状态的连接，而不能看到已经建立连接状态的连接。例如：

```
[root@localhost ~]# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:11211           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 :::11211                :::*                    LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
```

```
tcp        0      0 :::22                   :::*                    LISTEN
udp        0      0 0.0.0.0:11211           0.0.0.0:*
udp        0      0 :::11211                :::*
#协议 接收队列 发送队列 本机的 IP 地址及端口号      远程主机的 IP 地址及端口号       状态
```

这个命令的输出较多。

- Proto：网络连接的协议，一般就是 TCP 协议或者 UDP 协议。
- Recv-Q：表示接收到的数据，已经在本地的缓冲中，但是还没有被进程取走。
- Send-Q：表示从本机发送，对方还没有收到的数据，依然在本地的缓冲中，一般是不具备 ACK 标志的数据包。
- Local Address：本机的 IP 地址和端口号。
- Foreign Address：远程主机的 IP 地址和端口号。
- State：状态。常见的状态主要有以下几种。
  - LISTEN：监听状态，只有 TCP 协议需要监听，而 UDP 协议不需要监听。
  - ESTABLISHED：已经建立连接的状态。如果使用“-l”选项，则看不到已经建立连接的状态。
  - SYN_SENT：SYN 发起包，就是主动发起连接的数据包。
  - SYN_RECV：接收到主动连接的数据包。
  - FIN_WAIT1：正在中断的连接。
  - FIN_WAIT2：已经中断的连接，但是正在等待对方主机进行确认。
  - TIME_WAIT：连接已经中断，但是套接字依然在网络中等待结束。
  - CLOSED：套接字没有被使用。

在这些状态中，我们最常用的就是 LISTEN 和 ESTABLISHED 状态，一种代表正在监听，另一种代表已经建立连接。

例子 2：查看本机有哪些程序开启的端口

如果使用“-p”选项，则可以查看到是哪个程序占用了端口，并且可以知道这个程序的 PID。例如：

```
[root@localhost ~]# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address    Foreign Address  State    PID/Program name
tcp        0      0 0.0.0.0:3306    0.0.0.0:*        LISTEN   2359/mysqld
tcp        0      0 0.0.0.0:11211   0.0.0.0:*        LISTEN   1563/memcached
tcp        0      0 0.0.0.0:22      0.0.0.0:*        LISTEN   1490/sshd
tcp        0      0 :::11211        :::*             LISTEN   1563/memcached
tcp        0      0 :::80           :::*             LISTEN   21025/httpd
tcp        0      0 :::22           :::*             LISTEN   1490/sshd
udp        0      0 0.0.0.0:11211   0.0.0.0:*                 1563/memcached
udp        0      0 :::11211        :::*                      1563/memcached
#比之前的命令多了一个“-p”选项，结果多了“PID/程序名”，可以知道是哪个程序占用了端口
```

例子 3：查看所有连接

使用选项“-an”可以查看所有连接，包括监听状态的连接（LISTEN）、已经建立连接状态的连接（ESTABLISHED）、Socket 程序连接等。因为连接较多，所以输出的内容有很多。例如：

```
[root@localhost ~]# netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address         State
tcp       0      0 0.0.0.0:3306           0.0.0.0:*               LISTEN
tcp       0      0 0.0.0.0:11211          0.0.0.0:*               LISTEN
tcp       0      0 117.79.130.170:80      78.46.174.55:58815      SYN_RECV
tcp       0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp       0      0 117.79.130.170:22      124.205.129.99:10379    ESTABLISHED
tcp       0      0 117.79.130.170:22      124.205.129.99:11811    ESTABLISHED
…省略部分内容…
udp       0      0 0.0.0.0:11211          0.0.0.0:*
udp       0      0 :::11211               :::*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags    Type    State       I-Node Path
unix  2      [ ACC ]  STREAM  LISTENING   9761  @/var/run/hald/dbus-fr41WkQn1C
…省略部分内容…
```

从“Active UNIX domain sockets”开始，之后的内容就是 Socket 程序产生的连接，之前的内容都是网络服务产生的连接。我们可以在“-an”选项的输出中看到各种网络连接状态，而之前的“-tuln”选项则只能看到监听状态。

5. write 命令

write 命令的基本信息如下。

- 命令名称：write。
- 英文原意：send a message to another user。
- 所在路径：/usr/bin/write。
- 执行权限：所有用户。
- 功能描述：向其他用户发送信息。

```
[root@localhost ~]#write user1 pts/1
hello
I will be in 5 minutes to restart, please save your data
#向在pts/1（远程终端1）登录的user1用户发送信息，使用"Ctrl+D"快捷键保存发送的数据
```

6. wall 命令

write 命令用于给指定用户发送信息，而 wall 命令用于给所有登录用户发送信息，包括你自己。执行时，在 wall 命令后加入需要发送的信息即可，例如：

```
[root@localhost ~]# wall "I will be in 5 minutes to restart, please save your data"
```

7. mail 命令

mail 是 Linux 的邮件客户端命令，可以利用这个命令给其他用户发送邮件。mail 命令的基本信息如下。

- 命令名称：mail。

- 英文原意：send and receive Internet mail。
- 所在路径：/bin/mail。
- 执行权限：所有用户。
- 功能描述：发送和接收电子邮件。

例子 1：发送邮件

如果我们想要给其他用户发送邮件，则可以执行如下命令：

```
[root@localhost ~]# mail user1
Subject: hello              <-  邮件标题
Nice to meet you!           <-  邮件具体内容
.                           <-  使用“.”来结束邮件输入
#发送邮件给 user1 用户
```

我们接收到的邮件都保存在“/var/spool/mail/用户名”中，每个用户都有一个以自己的用户名命名的邮箱。

例子 2：发送文件内容

如果我们想把某个文件的内容发送给指定用户，则可以执行如下命令：

```
[root@localhost ~]# mail -s "test mail" root < /root/anaconda-ks.cfg
选项：
    -s：    指定邮件标题
#把/root/anaconda-ks.cfg 文件的内容发送给 root 用户
```

我们在写脚本时，有时需要脚本自动发送一些信息给指定用户，把要发送的信息预先写到文件中，是一个非常不错的选择。

例子 3：查看已经接收的邮件

我们可以直接在命令行中执行 mail 命令，进入 mail 的交互命令中，可以在这里查看已经接收到的邮件。例如：

```
[root@localhost ~]# mail
Heirloom Mail version 12.4 7/29/08.Type ?for help.
"/var/spool/mail/root": 1 message 1 new
>N  1 root               Mon Dec  5 22:45  68/1777  "test mail"<-之前收到的邮件
>N  2 root               Mon Dec  5 23:08  18/602   "hello"
#未阅读 编号 发件人        时间                        标题
&                                                        <-等待用户输入命令
```

可以看到已经接收到的邮件列表，“N”代表未读邮件，如果是已经阅读过的邮件，则前面是不会有这个“N”的；之后的数字是邮件的编号，我们主要通过这个编号来进行邮件的操作。如果我们想要查看第一封邮件，则只需输入邮件的编号“1”就可以了。

在交互命令中执行“？”，可以查看这个交互界面支持的命令。例如：

```
& ?                         <-输入命令
mail commands
type<message list>          type messages
```

```
next                        goto and type next message
from<message list>          give head lines of messages
headers                     print out active message headers
delete<message list>            delete messages
undelete<message list>          undelete messages
save<message list> folder       append messages to folder and mark as saved
copy<message list> folder       append messages to folder without marking them
write<message list> file        append message texts to file, save attachments
preserve<message list>          keep incoming messages in mailbox even if saved
Reply <message list>            reply to message senders
reply<message list>             reply to message senders and all recipients
mail addresses                  mail to specific recipients
file folder                     change to another folder
quit                            quit and apply changes to folder
xit                             quit and discard changes made to folder
!                               shell escape
cd<directory>                   chdir to directory or home if none given
list                            list names of all available commands
```

这些交互命令是可以简化输入的，比如“headers”命令，就可以直接输入“h”，这是列出邮件标题列表的命令。我们解释一下常用的交互命令。

- headers：列出邮件标题列表，直接输入“h”命令即可。
- delete：删除指定邮件。比如想要删除第二封邮件，可以输入“d 2”。
- save：保存邮件。可以把指定邮件保存成文件，如“s 2 /tmp/test.mail”。
- quit：退出，并把已经操作过的邮件进行保存。比如移除已删除邮件、保存已阅读邮件等。
- exit：退出，但是不保存任何操作。

十一　系统痕迹命令

系统中有一些重要的痕迹日志文件，如/var/log/wtmp、/var/run/utmp、/var/log/btmp、/var/log/lastlog 等日志文件，如果你用 vim 打开这些文件，你会发现这些文件是二进制乱码。这是由于这些日志中保存的是系统的重要登录痕迹，包括某个用户何时登录了系统，何时退出了系统，错误登录等重要的系统信息。这些信息要是可以通过 vim 打开，就能编辑，这样痕迹信息就不准确，所以这些重要的痕迹日志，只能通过对应的命令来进行查看。

1．w 命令

w 命令是显示系统中正在登陆的用户信息的命令，这个命令查看的痕迹日志是/var/run/utmp。这个命令的基本信息如下：

- 命令名称：w
- 英文原意：Show who is logged on and what they are doing.

- 所在路径：/usr/bin/w
- 执行权限：所有用户。
- 功能描述：显示灯用户，和他正在做什么。

例如：

```
[root@localhost ~]# w
 00:06:11 up  5:47,  2 users,  load average: 0.00, 0.01, 0.05
#系统时间     持续开机时间   登陆用户       系统在 1 分钟，5 分钟，15 分钟前的平均负载
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1                      23:59    7:07   0.08s  0.08s -bash
root     pts/2    192.168.252.1    23:42    3.00s  0.44s  0.06s w
```

第一行信息，内容如下：

内　　容	说　　明
12:26:46	系统当前时间
up 1 day, 13:32	系统的运行时间，本机已经运行 1 天 13 小时 32 分钟
2 users	当前登录了两个用户
load average: 0.00, 0.00, 0.00	系统在之前 1 分钟、5 分钟、15 分钟的平均负载。如果 CPU 是单核的，则这个数值超过 1 就是高负载；如果 CPU 是四核的，则这个数值超过 4 就是高负载 （这个平均负载完全是依据个人经验来进行判断的，一般认为不应该超过服务器 CPU 的核数）

第二行信息，内容如下：

内　　容	说　　明
USER	当前登陆的用户
TTY	登陆的终端： tty1-6：本地字符终端（alt+F1-6 切换） tty7：本地图形终端（ctrl+alt+F7 切换，必须安装启动图形界面） pts/0-255:远程终端
FROM	登陆的 IP 地址，如果是本地终端，则是空
LOGIN@	登陆时间
IDLE	用户闲置时间
JCPU	所有的进程占用的 CPU 时间
PCPU	当前进程占用的 CPU 时间
WHAT	用户正在进行的操作

2. who 命令

who 命令和 w 命令类似，用于查看正在登陆的用户，但是显示的内容更加简单，也是查看/var/run/utmp 日志。

```
[root@localhost ~]# who
root     tty1         2018-11-12 23:59
root     pts/2        2018-11-12 23:42 (192.168.252.1)
#用户名  登陆终端      登陆时间（来源 IP）
```

3. last 命令

last 命令是查看系统所有登陆过的用户信息的，包括正在登陆的用户和之前登陆的用户。这个命令查看的是/var/log/wtmp 痕迹日志文件。

```
[root@localhost ~]# last
root     tty1                      Mon Nov 12 23:59   still logged in
root     pts/2        192.168.252.1    Mon Nov 12 23:42   still logged in
root     pts/1        192.168.252.1    Mon Nov 12 23:37 - 23:59  (00:22)
root     tty1                      Mon Nov 12 19:17 - 23:58  (04:41)
root     pts/0        192.168.252.1    Mon Nov 12 18:20 - 23:52  (05:32)
reboot   system boot  3.10.0-862.el7.x Mon Nov 12 18:18 - 00:22  (06:03)
#系统重启信息记录
root     pts/1        192.168.252.1    Mon Nov 12 08:48 - down   (01:29)
root     pts/1        192.168.252.1    Thu Nov  8 21:04 - 22:29  (01:25)
#用户名  终端号        来源 IP 地址       登陆时间  -    退出时间
```

4. lastlog 命令

lastlog 命令是查看系统中所有用户最后一次的登陆时间的命令，他查看的日志是/var/log/lastlog 文件。

```
[root@localhost ~]# lastlog
Username         Port     From             Latest
root             tty1                      Mon Nov 12 23:59:03 +0800 2018
bin                                       **Never logged in**
daemon                                     **Never logged in**
adm                                       **Never logged in**
lp                                        **Never logged in**
sync                                       **Never logged in**
…省略部分内容…
#用户名           终端      来源 IP        登陆时间
```

5. lastb 命令

lastb 命令是查看错误登陆的信息的，查看的是/var/log/btmp 痕迹日志：

```
[root@localhost ~]# lastb
(unknown tty1                        Mon Nov 12 23:58 - 23:58  (00:00)
root     tty1                       Mon Nov 12 23:58 - 23:58  (00:00)
#错误登陆用户      终端                    尝试登陆的时间
```

十二　挂载命令

1. mount 命令基本格式

linux 所有存储设备都必须挂载使用，包括硬盘

命令名称：mount

命令所在路径：/bin/mount

执行权限：所有用户

说了这么多，命令的具体格式如下：

```
[root@localhost ~]# mount [-l]
#查询系统中已经挂载的设备，-l 会显示卷标名称
[root@localhost ~]# mount  -a
#依据配置文件/etc/fstab 的内容，自动挂载
[root@localhost ~]# mount [-t 文件系统] [-L 卷标名] [-o 特殊选项]  \
设备文件名 挂载点
#\代表这一行没有写完，换行
选项：
    -t 文件系统：     加入文件系统类型来指定挂载的类型，可以 ext3、ext4、iso9660
                等文件系统。具体可以参考表 9-1
    -L 卷标名：       挂载指定卷标的分区，而不是安装设备文件名挂载
    -o 特殊选项：     可以指定挂载的额外选项，比如读写权限、同步异步等，如果不指定
                则默认值生效。具体的特殊选项，见表 9-4：
```

参数	说明
atime/noatime	更新访问时间/不更新访问时间。访问分区文件时，是否更新文件的访问时间，默认为更新
async/sync	异步/同步，默认为异步
auto/noauto	自动/手动，mount -a 命令执行时，是否会自动安装/etc/fstab 文件内容挂载，默认为自动
defaults	定义默认值，相当于 rw, suid, dev, exec, auto, nouser, async 这七个选项
exec/noexec	执行/不执行，设定是否允许在文件系统中执行可执行文件，默认是 exec 允许
remount	重新挂载已经挂载的文件系统，一般用于指定修改特殊权限

rw/ro	读写/只读，文件系统挂载时，是否具有读写权限，默认是 rw
suid/nosuid	具有/不具有 SUID 权限，设定文件系统是否具有 SUID 和 SGID 的权限，默认是具有
user/nouser	允许/不允许普通用户挂载，设定文件系统是否允许普通用户挂载，默认是不允许，只有 root 可以挂载分区
usrquota	写入代表文件系统支持用户磁盘配额，默认不支持
grpquota	写入代表文件系统支持组磁盘配额，默认不支持

举例

```
例 1：
[root@localhost ~]# mount
#查看系统中已经挂载的文件系统，注意有虚拟文件系统
/dev/sda3 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
#命令结果是代表：/dev/sda3 分区挂载到/目录，文件系统是 ext4，权限是读写

例 2：修改特殊权限
[root@localhost ~]# mount
#我们查看到/boot 分区已经被挂载，而且采用的 defaults 选项，那么我们重新挂载分区，并采用 noexec
#权限禁止执行文件执行，看看会出现什么情况（注意不要用/分区做试验，#不然系统命令也不能执行了）。
…省略部分输出…
/dev/sda1 on /boot type ext4 (rw)
…省略部分输出…
[root@localhost ~]# mount -o remount,noexec /boot
#重新挂载/boot 分区，并使用 noexec 权限
[root@localhost sh]# cd /boot/
[root@localhost boot]# vi hello.sh
#写个 shell 吧
#!/bin/bash
echo "hello!!"
[root@localhost boot]# chmod 755 hello.sh
[root@localhost boot]# ./hello.sh
-bash: ./hello.sh: 权限不够
#虽然赋予了 hello.sh 执行权限，但是任然无法执行
[root@localhost boot]# mount -o remount,exec /boot
#记得改回来啊，要不会影响系统启动的
```

如果我们做试验修改了特殊选项，一定要记得住，而且确定需要修改，否则非常容易出现系统问

题，而且还找不到哪里出现了问题。

例 3：挂载分区

```
[root@localhost ~]# mkdir /mnt/disk1
#建立挂载点目录
[root@localhost ~]# mount /dev/sdb1 /mnt/disk1
#挂载分区
```

2. 光盘挂载

光盘挂载的前提依然是指定光盘的设备文件名，不同版本的 Linux，设备文件名并不相同：

- CentOS 5.x 以前的系统，光盘设备文件名是/dev/hdc
- CentOS 6.x 以后的系统，光盘设备文件名是/dev/sr0

不论哪个系统都有软连接/dev/cdrom，与可以作为光盘的设备文件名

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

用完之后记得卸载：

```
[root@localhost ~]# umount /dev/sr0
[root@localhost ~]# umount /mnt/cdrom
#因为设备文件名和挂载点已经连接到一起，卸载哪一个都可以
```

注意：卸载的时候需要退出光盘目录，才能正常卸载

3. 挂载 U 盘

U 盘会和硬盘共用设备文件名，所以 U 盘的设备文件名不是固定的，需要手工查询，查询命令：

```
[root@localhost ~]# fdisk -l
#查询硬盘
```

然后就是挂载了，挂载命令如下：

```
[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/
#挂载 U 盘。因为是 Windows 分区，所以是 vfat 文件系统格式
```

如果 U 盘中有中文，会发现中文是乱码。Linux 要想正常显示中文，需要两个条件：

- 安装了中文编码和中文字体
- 操作终端需要支持中文显示（纯字符终端，是不支持中文编码的）

而我们当前系统是安装了中文编码和字体，而 xshell 远程终端是 Windows 下的程序，当然是支持中文显示的。那之所以挂载 U 盘还出现乱码，是需要在挂载的时候，手工指定中文编码，例如：

```
[root@localhost ~]# mount -t vfat -o iocharset=utf8 /dev/sdb1 /mnt/usb/
#挂载 U 盘，指定中文编码格式为 UTF-8
```

如果需要卸载，可以执行以下命令：

```
[root@localhost ~]# umount /mnt/usb/
```

4、挂载 NTFS 分区

4.1 Linux 的驱动加载顺序：

- 驱动直接放入系统内核之中。这种驱动主要是系统启动加载必须的驱动，数量较少。
- 驱动以模块的形式放入硬盘。大多数驱动都已这种方式保存，保存位置在/lib/modules/3.10.0-862.el7.x86_64/kernel/中。
- 驱动可以被 Linux 识别，但是系统认为这种驱动一般不常用，默认不加载。如果需要加载这种驱动，需要重新编译内核，而 NTFS 文件系统的驱动就属于这种情况。
- 硬件不能被 Linux 内核识别，需要手工安装驱动。当然前提是厂商提供了该硬件针对 Linux 的驱动，否则就需要自己开发驱动了☺。

4.2 使用 NTFS-3G 安装 NTFS 文件系统模块

✧ 下载 NTFS-3G 插件

我们从网站 http://www.tuxera.com/community/ntfs-3g-download/下载 NTFS-3G 插件到 Linux 服务器上。

✧ 安装 NTFS-3G 插件

在编译安装 NTFS-3G 插件之前，要保证 gcc 编译器已经安装。具体安装命令如下：

```
[root@localhost ~]# tar -zxvf ntfs-3g_ntfsprogs-2013.1.13.tgz
#解压
[root@localhost ~]# cd ntfs-3g_ntfsprogs-2013.1.13
#进入解压目录
[root@localhost ntfs-3g_ntfsprogs-2013.1.13]# ./configure
#编译器准备。没有指定安装目录，安装到默认位置中
[root@localhost ntfs-3g_ntfsprogs-2013.1.13]# make
#编译
[root@localhost ntfs-3g_ntfsprogs-2013.1.13]# make install
#编译安装
```

安装就完成了，已经可以挂载和使用 Windows 的 NTFS 分区了。不过需要注意挂载分区时的文件系统不是 ntfs，而是 ntfs-3g。挂载命令如下：

```
[root@localhost ~]# mount -t ntfs-3g 分区设备文件名 挂载点

例如：
[root@localhost ~]# mount  -t ntfs-3g /dev/sdb1 /mnt/win
```