07 【实操篇-文件权限和搜索查找类命令】

1.文件权限类

1.1 权限管理的重要性

和 Windows 系统不同，Linux 系统为每个文件都添加了很多的属性，最大的作用就是维护数据的安全。举个简单的例子，在你的 Linux 系统中，和系统服务相关的文件通常只有 root 用户才能读或写，就拿 /etc/shadow 这个文件来说，此文件记录了系统中所有用户的密码数据，非常重要，因此绝不能让任何人读取（否则密码数据会被窃取），只有 root 才可以有读取权限。

此外，如果你有一个软件开发团队，你希望团队中的每个人都可以使用某一些目录下的文件，而非团队的其他人则不予以开放。通过前面章节的学习我们知道，只需要将团队中的所有人加入新的群组，并赋予此群组读写目录的权限，即可实现要求。反之，如果你的目录权限没有做好，就很难防止其他人在你的系统中乱搞。

比如说，本来 root 用户才能做的开关机、ADSL 拨接程序，新增或删除用户等命令，一旦允许任何人拥有这些权限，系统很可能会经常莫名其妙的挂掉。而且，万一 root 用户的密码被其他人获取，他们就可以登录你的系统，从事一些只有 root 用户才能执行的操作，这是绝对不允许发生的。

因此，在服务器上，绝对不是所有的用户都使用 root 身份登录，而要根据不同的工作需要和职位需要，合理分配用户等级和权限等级。

在Linux中我们可以使用ll或者ls -l命令来显示一个文件的属性以及文件所属 的用户和组。

```
[root@localhost ~]# ls -al
total 156
drwxr-x---.   4    root   root     4096   Sep  8 14:06 .
drwxr-xr-x.  23    root   root     4096   Sep  8 14:21 ..
-rw-------.   1    root   root     1474   Sep  4 18:27 anaconda-ks.cfg
-rw-------.   1    root   root      199   Sep  8 17:14 .bash_history
-rw-r--r--.   1    root   root       24   Jan  6  2007 .bash_logout
...
```

1.2 权限位

Linux 系统，最常见的文件权限有 3 种，即对文件的读（用 r 表示）、写（用 w 表示）和执行（用 x 表示，针对可执行文件或目录）权限。在 Linux 系统中，每个文件都明确规定了不同身份用户的访问权限，通过 ls 命令即可看到。

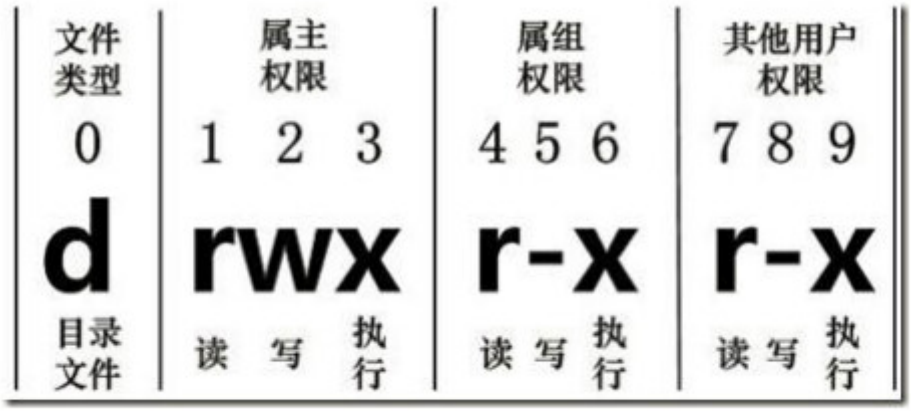
> 除此之外，我们有时会看到 s（针对可执行文件或目录，使文件在执行阶段，临时拥有文件所有者的权限）和 t（针对目录，任何用户都可以在此目录中创建文件，但只能删除自己的文件），文件设置 s 和 t 权限，会占用 x 权限的位置。

例如，我们以 root 的身份登陆 Linux，并执行如下指令：

```
[root@localhost ~]# ls -al
total 156
drwxr-x---.   4    root   root     4096   Sep  8 14:06 .
drwxr-xr-x.  23    root   root     4096   Sep  8 14:21 ..
-rw-------.   1    root   root     1474   Sep  4 18:27 anaconda-ks.cfg
-rw-------.   1    root   root      199   Sep  8 17:14 .bash_history
-rw-r--r--.   1    root   root       24   Jan  6  2007 .bash_logout
...
```

可以看到，每行的第一列表示的就是各文件针对不同用户设定的权限，一共 11 位，但第 1 位用于表示文件的具体类型，最后一位此文件受 SELinux 的安全规则管理，不是本节关心的内容。

因此，为文件设定不同用户的读、写和执行权限，仅涉及到 9 位字符，以 ls 命令输出信息中的 .bash_logout 文件为例，设定不同用户的访问权限是 rw-r--r--，各权限位的含义如图所示。



从图中可以看到，Linux 将访问文件的用户分为 3 类，分别是文件的所有者，所属组（也就是文件所属的群组）以及其他人。

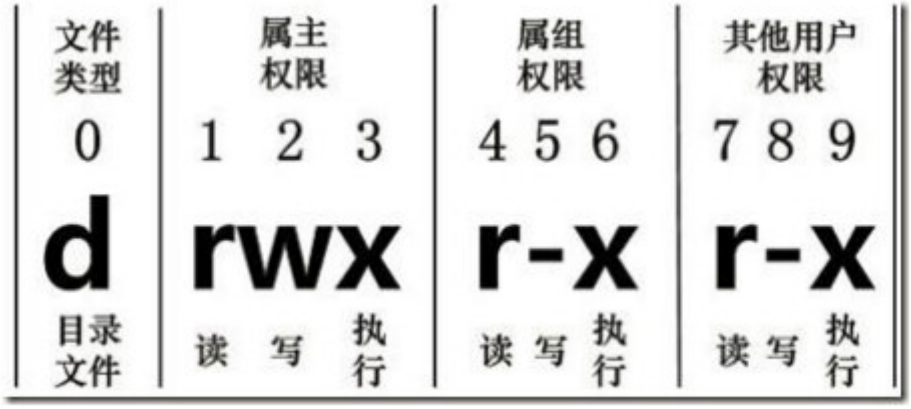
> 除了所有者，以及所属群组中的用户可以访问文件外，其他用户（其他群组中的用户）也可以访问文件，这部分用户都归为其他人范畴。

很显然，Linux 系统为 3 种不同的用户身份，分别规定了是否对文件有读、写和执行权限。拿图 1 来说，文件所有者拥有对文件的读和写权限，但是没有执行权限；所属群组中的用户只拥有读权限，也就是说，这部分用户只能读取文件内容，无法修改文件；其他人也是只能读取文件。

Linux 系统中，多数文件的文件所有者和所属群组都是 root（都是 root 账户创建的），这也就是为什么，root 用户是超级管理员，权限足够大的原因。

1.3 读写执行权限（-r、-w、-x）的含义

从左到右的 10 个字符表示



如果没有权限，就会出现减号[-]而已。从左至右用0-9这些数字来表示:

（1）0 首位表示类型

在Linux中第一个字符代表这个文件是目录、文件或链接文件等等

> - \- 代表文件
> - d 代表目录
> - l 链接文档(link file)；

（2）第1-3位确定属主（该文件的所有者）拥有该文件的权限。---User

（3）第4-6位确定属组（所有者的同组用户）拥有该文件的权限，---Group

（4）第7-9位确定其他用户拥有该文件的权限 ---Other

rwx 作用文件和目录的不同解释

（1）作用到文件：

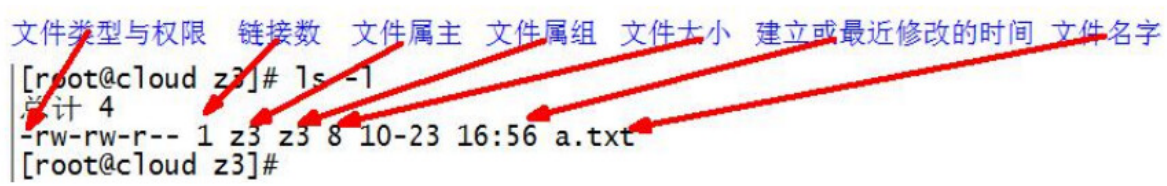
- [r]代表可读(read): 可以读取，查看
- [w]代表可写(write): 可以修改，但是不代表可以删除该文件，删除一个文件的前提条件是对该文件所在的目录有写权限，才能删除该文件
- [x]代表可执行(execute):可以被系统执行

（2）作用到目录：

- [r]代表可读(read): 可以读取，ls查看目录内容
- [w]代表可写(write): 可以修改，目录内创建+删除+重命名目录
- [x]代表可执行(execute):可以进入该目录

案例实操

```
[root@hadoop101 ~]# ll
总用量 104
-rw-------. 1 root root 1248 1 月  8 17:36 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4096 1 月 12 14:02 dssz
lrwxrwxrwx. 1 root root 20 1 月 12 14:32 houzi -> xiyou/dssz/houge.tx
```



（1）如果查看到是文件：链接数指的是硬链接个数。

（2）如果查看的是文件夹：链接数指的是子文件夹个数。

1.4 chmod 改变权限

既然我们已经知道文件权限对于一个系统的重要性，也知道每个文件都设定了针对不同用户的访问权限，那么，是否可以手动修改文件的访问权限呢？

可以，通过 chmod 命令即可。chmod 命令设定文件权限的方式有 2 种，分别可以使用数字或者符号来进行权限的变更。

1.4.1 chmod命令使用数字修改文件权限

Linux 系统中，文件的基本权限由 9 个字符组成，以 `rwxrw-r-x` 为例，我们可以使用数字来代表各个权限，各个权限与数字的对应关系如下：

```
r --> 4
w --> 2
x --> 1
```

由于这 9 个字符分属 3 类用户，因此每种用户身份包含 3 个权限（r、w、x），通过将 3 个权限对应的数字累加，最终得到的值即可作为每种用户所具有的权限。

拿 rwxrw-r-x 来说，所有者、所属组和其他人分别对应的权限值为：

```
所有者 = rwx = 4+2+1 = 7
所属组 = rw- = 4+2 = 6
其他人 = r-x = 4+1 = 5
```

所以，此权限对应的权限值就是 765。

使用数字修改文件权限的 chmod 命令基本格式为：

`[root@localhost ~]# chmod [-R] 权限值 文件名`

- -R（注意是大写）选项表示连同子目录中的所有文件，也都修改设定的权限。

例如，使用如下命令，即可完成对 .bashrc 目录文件的权限修改：

```
[root@localhost ~]# ls -al .bashrc
-rw-r--r--. 1 root root 176 Sep 22 2004 .bashrc
[root@localhost ~]# chmod 777 .bashrc
[root@localhost ~]# ls -al .bashrc
-rwxrwxrwx. 1 root root 176 Sep 22 2004 .bashrc
```

再举个例子，通常我们以 Vim 编辑 Shell 文件批处理文件后，文件权限通常是 rw-r--r--（644），那么，如果要将该文件变成可执行文件，并且不让其他人修改此文件，则只需将此文件的权限该为 rwxr-xr-x（755）即可。

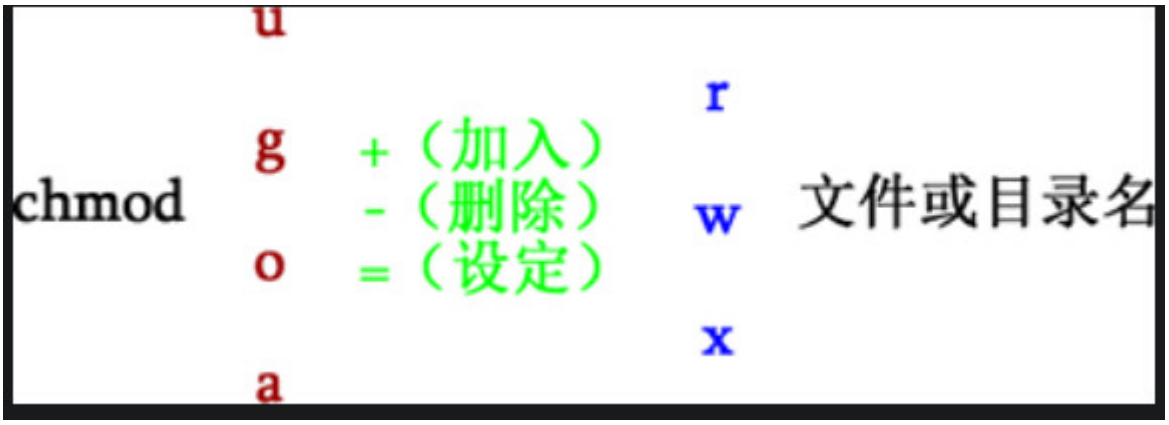
修改整个文件夹里面的所有文件的所有者、所属组、其他用户都具有可读可写可 执行权限。

```
[root@localhost ~]# chmod -R 777 .bashrc
```

1.4.2 chmod命令使用字母修改文件权限

既然文件的基本权限就是 3 种用户身份（所有者、所属组和其他人）搭配 3 种权限（rwx），chmod 命令中用 u、g、o 分别代表 3 种身份，还用 a 表示全部的身份（all 的缩写）。另外，chmod 命令仍使用 r、w、x 分别表示读、写、执行权限。

使用字母修改文件权限的 chmod 命令，其基本格式如图所示。



例如，如果我们要设定 .bashrc 文件的权限为 rwxr-xr-x，则可执行如下命令：

```
[root@localhost ~]# chmod u=rwx,go=rx .bashrc
[root@localhost ~]# ls -al .bashrc
-rwxr-xr-x. 1 root root 176 Sep 22 2004 .bashrc
```

再举个例子，如果想要增加 .bashrc 文件的每种用户都可做写操作的权限，可以使用如下命令：

```
[root@localhost ~]# ls -al .bashrc
-rwxr-xr-x. 1 root root 176 Sep 22 2004 .bashrc
[root@localhost ~]# chmod a+w .bashrc
[root@localhost ~]# ls -al .bashrc
-rwxrwxrwx. 1 root root 176 Sep 22 2004 .bashrc
```

> 实际还是以数字修改为主

1.5 chown 改变所有者和所属组

chown 命令，可以认为是 "change owner" 的缩写，主要用于修改文件（或目录）的所有者，除此之外，这个命令也可以修改文件（或目录）的所属组。

当只需要修改所有者时，可使用如下 chown 命令的基本格式：

`[root@localhost ~]# chown [-R] 所有者 文件或目录`

- -R（注意大写）选项表示连同子目录中的所有文件，都更改所有者。

如果需要同时更改所有者和所属组，chown 命令的基本格式为：

`[root@localhost ~]# chown [-R] 所有者:所属组 文件或目录`

注意，在 chown 命令中，所有者和所属组中间也可以使用点（.），但会产生一个问题，如果用户在设定账号时加入了小数点（例如 zhangsan.temp），就会造成系统误判。因此，建议大家使用冒号连接所有者和所属组。

当然，chown 命令也支持单纯的修改文件或目录的所属组，例如 `chown :group install.log` 就表示修改 install.log 文件的所属组，但修改所属组通常使用 chgrp 命令，因此并不推荐大家使用 chown 命令。

另外需要注意的一点是，使用 chown 命令修改文件或目录的所有者（或所属者）时，要保证使用者用户（或用户组）存在，否则该命令无法正确执行，会提示 "invalid user" 或者 "invaild group"。

【例 1】
其实，修改文件的所有者，更多时候是为了得到更高的权限，举一个实例：

```
[root@localhost ~]# touch file
#由root用户创建file文件
[root@localhost ~]# ll file
-rw-r--r--. 1 root root 0 Apr 17 05:12 file
#文件的所有者是root，普通用户user对这个文件拥有只读权限
[root@localhost ~]# chown user file
#修改文件的所有者
[root@localhost ~]# ll file
-rw-r--r--. 1 user root 0 Apr 17 05:12 file
#所有者变成了user用户，这时user用户对这个文件就拥有了读、写权限
```

可以看到，通过修改 file 文件的所有者，user 用户从其他人身份（只对此文件有读取权限）转变成了所有者身份，对此文件拥有读和写权限。

【例 2】
Linux 系统中，用户等级权限的划分是非常清楚的，root 用户拥有最高权限，可以修改任何文件的权限，而普通用户只能修改自己文件的权限（所有者是自己的文件），例如：

```
[root@localhost ~]# cd /home/user
#进入user用户的家目录
[root@localhost user]# touch test
#由root用户新建文件test
[root@localhost user]# ll test
-rw-r--r--. 1 root root 0 Apr 17 05:37 test
#文件所有者和所属组都是root用户
[root@localhost user]# su - user
#切换为user用户
[user@localhost ~]$ chmod 755 test
chmod:更改"test"的权限：不允许的操作 #user用户不能修改test文件的权限
[user@localhost ~]$ exit
#退回到root身份
[root@localhost user]# chown user test
#由root用户把test文件的所有者改为user用户
[root@localhost user]# su - user
#切换为user用户
[user@localhost ~]$ chmod 755 test
#user用户由于是test文件的所有者，所以可以修改文件的权限
[user@localhost ~]$ ll test
-rwxr-xr-x. 1 user root 0 Apr 17 05:37 test
#查看权限
```

可以看到，user 用户无权更改所有者为 root 用户文件的权限，只有普通用户是这个文件的所有者，才可以修改文件的权限。

【例 3】

```
[root@localhost ~]# chown user:group file
[root@localhost ~]# ll file
-rw-r--r--. 1 user group 0 Apr 17 05:12 file
```

1.6 chgrp 改变所属组

hgrp 命令用于修改文件（或目录）的所属组。

为了方便初学者记忆，可以将 chgrp 理解为是 "change group" 的缩写。

chgrp 命令的用法很简单，其基本格式为：

`[root@localhost ~]# chgrp [-R] 所属组 文件名（目录名）`

- -R（注意是大写）选项长作用于更改目录的所属组，表示更改连同子目录中所有文件的所属组信息。

使用此命令需要注意的一点是，要被改变的群组名必须是真实存在的，否则命令无法正确执行，会提示 "invaild group name"。

举个例子，当以 root 身份登录 Linux 系统时，主目录中会存在一个名为 install.log 的文件，我们可以使用如下方法修改此文件的所属组：

```
[root@localhost ~]# groupadd group1
#新建用于测试的群组 group1
[root@localhost ~]# chgrp group1 install.log
#修改install.log文件的所属组为group1
[root@localhost ~]# ll install.log
-rw-r--r--. 1 root group1 78495 Nov 17 05:54 install.log
#修改生效
[root@localhost ~]# chgrp testgroup install.log
chgrp: invaild group name 'testgroup'
```

可以看到，在具有 group1 群组的前提下，我们成功修改了 install.log 文件的所属组，但我们再次试图将所属组修改为 testgroup 时，命令执行失败，就是因为系统的 /etc/group 文件中，没有 testgroup 群组。

2.搜索查找类

2.1 find 查找文件或者目录

Linux find 命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则 find 命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。find 命令有非常大的灵活性，可以向其指定丰富的搜索条件（如文件权限、属主、属组、文件类型、日期和大小等）来定位系统中的文件和目录。此外，find 还支持对搜索到的结果进行多种类型的命令操作。

1）基本语法

find [搜索范围] [选项]

2）选项说明

- -name<查询方式> 按照指定的文件名查找模式查找文件
- -user<用户名> 查找属于指定用户名所有文件

- -size<文件大小> 按照指定的文件大小查找文件,单位为
  - b —— 块（512 字节）
  - c —— 字节
  - w —— 字（2 字节）
  - k —— 千字节
  - M —— 兆字节
  - G —— 吉字节

3）案例实操

（1）按文件名：根据名称查找/目录下的filename.txt文件。

```
[root@hadoop101 ~]# find xiyou/ -name "*.txt"
```

(2) 将当前目录及其子目录下所有文件后缀为 **.c** 的文件列出来

```
find . -name "*.c"
```

（3）按拥有者：查找/opt目录下，用户名称为-user的文件

```
[root@hadoop101 ~]# find xiyou/ -user atguigu
```

（4）按文件大小：在/home目录下查找大于200m的文件（+n 大于 -n小于 n等于）

```
[root@hadoop101 ~]find /home -size +204800
```

2.2 locate 快速定位文件路径

Linux locate命令用于查找符合条件的文档，他会去保存文档和目录名称的数据库内，查找合乎范本样式条件的文档或目录。

一般情况我们只需要输入 **locate your_file_name** 即可查找指定文件。

Locate 指令无需遍历整个文件系统，查询速度较快。为了保证查询结果的准确 度，管理员必须定期更新 locate 时刻。

1）基本语法

locate 搜索文件

2）经验技巧

locate 与 find 不同: find 是去硬盘找，locate 只在 /var/lib/slocate 资料库中找。

locate 的速度比 find 快，它并不是真的查找，而是查数据库，一般文件数据库在 /var/lib/slocate/slocate.db 中，所以 locate 的查找并不是实时的，而是以数据库的更新为准，一般是系统自己维护，也可以手工升级数据库 ，命令为：

```
updatedb
```

3）案例实操

查询文件夹

```
[root@hadoop101 ~]# updatedb
[root@hadoop101 ~]#locate tmp
```

查找 passwd 文件，输入以下命令：

```
[root@hadoop101 ~]# locate passwd
```

2.3 grep 过滤查找

Linux grep 命令用于查找文件里符合条件的字符串。

grep 指令用于查找内容包含指定的范本样式的文件，如果发现某文件的内容符合所指定的范本样式，预设 grep 指令会把含有范本样式的那一列显示出来。若不指定任何文件名称，或是所给予的文件名为 -，则 grep 指令会从标准输入设备读取数据。

基本语法

`grep 选项 查找内容 源文件`

- **-n 或 --line-number** : 在显示符合样式的那一行之前，标示出该行的列数编号。

实例

1、在当前目录中，查找后缀有 file 字样的文件中包含 test 字符串的文件，并打印出该字符串的行。此时，可以使用如下命令：

```
grep test *file
```

查找前缀有“test”的文件包含“test”字符串的文件

```
$ grep test test* #查找前缀有“test”的文件包含“test”字符串的文件
testfile1:This a Linux testfile! #列出testfile1 文件中包含test字符的行
testfile_2:This is a linux testfile! #列出testfile_2 文件中包含test字符的行
testfile_2:Linux test #列出testfile_2 文件中包含test字符的行
```

2、系统报警显示了时间，但是日志文件太大无法直接 cat 查看。(查询含有特定文本的文件，并拿到这些文本所在的行)

```
grep -n '2021-10-24 00:01:11' *.log
```

2.4 “|”管道符

> 管道符主要用于多重命令处理，前面命令的打印结果作为后面命令的输入。简单点说就是，就像工厂的流水线一样，进行完一道工序后，继续传送给下一道工序处理...

`# cat /etc/passwd | grep /bin/bash`

这条命令使用了两个管道，利用第一个管道将cat命令（显示passwd文件的内容）的输出送给grep命令，grep命令找出含有`/bin/bash`的所有行；

例

查找某文件在第几行

`[root@hadoop101 ~]# ls | grep -n test`

2.5 wc 计算字数

Linux wc命令用于计算字数。

利用wc指令我们可以计算文件的Byte数、字数、或是列数，若不指定文件名称、或是所给予的文件名为"-"，则wc指令会从标准输入设备读取数据。

语法

```
wc [-clw][--help][--version][文件...]
```

- -c或--bytes或--chars 只显示Bytes数。
- -l或--lines 显示行数。
- -w或--words 只显示字数。
- --help 在线帮助。
- --version 显示版本信息。

例1

在默认的情况下，wc将计算指定文件的行数、字数，以及字节数。使用的命令为：

```
wc testfile
```

先查看testfile文件的内容，可以看到：

```
$ cat testfile
Linux networks are becoming more and more common, but scurity is often an
overlooked
issue. Unfortunately, in today’s environment all networks are potential hacker
targets,
froOm tp-secret military research networks to small home LANs.
Linux Network Securty focuses on securing Linux in a networked environment,
where the
security of the entire network needs to be considered rather than just isolated
machines.
It uses a mix of theory and practicl techniques to teach administrators how to
install and
use security applications, as well as how the applcations work and why they are
necesary.
```

使用 wc统计，结果如下：

```
$ wc testfile           # testfile文件的统计信息
3 92 598 testfile       # testfile文件的行数为3、单词数92、字节数598
```

其中，3 个数字分别表示testfile文件的行数、单词数，以及该文件的字节数。

例2

如果想同时统计多个文件的信息，例如同时统计testfile、testfile_1、testfile_2，可使用如下命令：

```
wc testfile testfile_1 testfile_2   #统计三个文件的信息
```

输出结果如下：

```
$ wc testfile testfile_1 testfile_2  #统计三个文件的信息
3 92 598 testfile                    #第一个文件行数为3、单词数92、字节数598
9 18 78 testfile_1                   #第二个文件的行数为9、单词数18、字节数78
3 6 32 testfile_2                    #第三个文件的行数为3、单词数6、字节数32
15 116 708 总用量                   #三个文件总共的行数为15、单词数116、字节数708
```