

Linux命令

1. <https://wangchujiang.com/linux-command>
2. <https://jaywcjlove.gitee.io/linux-command>



Linux

Linux 命令搜索

搜索

添加命令 | 命令列表 | [Chrome 插件](#) | [Alfred](#) | [Dash](#) | [Krunner](#) | [开源中国Web版](#)
[Github](#) | 短地址: <https://git.io/linux>
收藏本站请使用Ctrl+D或者Command+d
共搜集到 573 个Linux命令



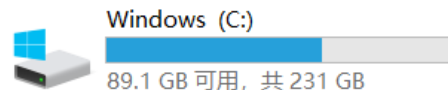
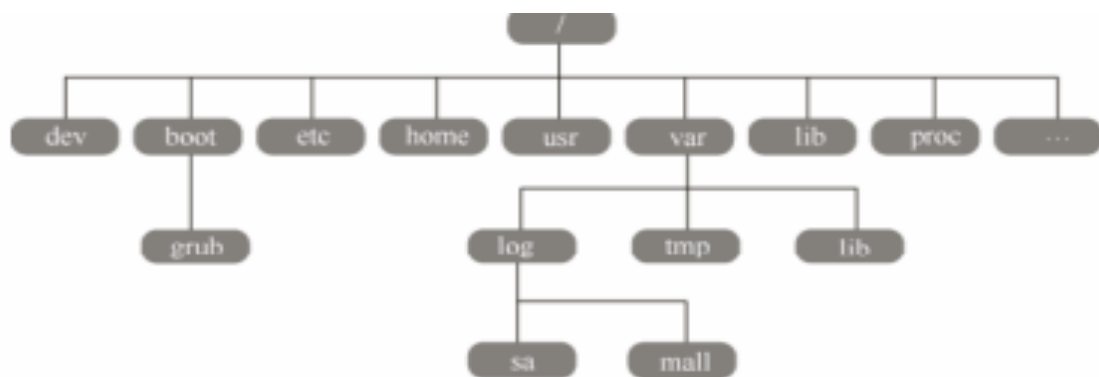
3.1 Linux目录结构

Linux 所有文件采用树形结构组织起来。

最顶层有一个“根 (/)”，然后在根上分支“(directory)”，任何一个分支上都可以再分支，分支也可以长出“叶子”。“根”和“分支”在**Linux**中被称为“目录”。而“叶”节点则是文件。

这种结构的文件系统效率高，现代操作系统基本都采用这种结构方式。

通常**Linux** 系统在安装后都会默认创建系统目录，以存放和整个操作系统相关的文件。**Linux** 系统树状目录结构如下：



3.1 Linux目录结构

系统目录及其说明如下。

1. /

/根目录 超级用户的主目录是**/root**。“**/**”位于 **linux** 文件系统目录结构的顶层，它是整个系统最重要的目录，因为所有的目录都是由根目录衍生出来，它是 **Linux** 文件系统的入口，是最高一级的目录。

2. /dev

/dev 是 **device** 的缩写，这个目录下保存所有的设备文件，用户可以通过这些文件访问外部设备，如 **sda** 文件表示硬盘设备。并且该目录下有一些由 **Linux** 内核创建的用来控制硬件设备的特殊文件。

3. /boot

/boot 叫作引导目录，主要放置开机时会使用到的文档，即该目录下存放系统的内核文件和引导装载程序文件，例如，系统中非常重要的 **Linux** 内核二进制文件 **vmlinux** 就放在该目录下。

4. /etc

/etc 保存绝大部分的系统配置文件，基本都是纯文本的，一般以扩展名 **.conf** 或 **.cnf** 结尾，如 **passwd**、**inittab**、**group** 等。

3.1 Linux目录结构

5. /home

即用户的主目录，每一个用户都有一个文件夹，保存该用户的私有数据。默认情况下，除 **root** 外的用户，主目录都会放在 **/home** 目录下。在 **Linux** 下，可以通过 **#cd ~** 来切换至自己的主目录。

6. /usr

该目录是系统存放程序的目录，如 **/usr/src** 中存放着 **Linux** 内核的源代码，**/usr/include** 中存放着 **Linux** 下开发和编译应用程序需要的头文件。

7. /var

存放系统产生的文件，该目录的内容经常变动。例如，**/var/tmp** 就是用来存储临时文件的，**/var/log** 日志文件。

8. /lib

/lib 启动时需要用到的库文件都放在该目录下，相当于 **Windows** 下的动态链接库 **dll** 文件。而非启动用的库文件都会放在 **/usr/lib** 目录下。内核模块是放在 **/lib/modules**（内核版本）下的。

3.1 Linux目录结构

9. /proc

这个目录在磁盘中是不存在的，它是存放在内存中的一个虚拟的文件夹，是启动**Linux**系统时创建的，里面的文件都是关于当前系统的实时状态信息，包括正在运行的进程、硬件状态、内存使用信息等。

10. /tmp

临时文件目录，有时用户运行程序时，会产生临时文件。因为**/tmp**会自动删除文件，所以有用的文件不要放在该目录下。**/var/tmp**目录和这个目录相似。

11. /mnt

该目录一般用于存放挂载储存设备的挂载目录，它是安装软盘、光盘、**U**盘的挂载点。**/media**是自动挂载，与**/mnt**相同。

3.1 Linux 目录结构

12. /bin

/bin 是 **binary** 的缩写，二进制文件，即可执行程序。里面保存的是基础系统所需的最基础的、最常用的命令，如 **ls**、**cp**、**mkdir** 等命令，功能和 **/usr/bin** 类似。这个目录中的文件都是可执行的，并且是普通用户都可以使用的命令。

13. /sbin

/sbin 是 **super binary** 的缩写，存放的大多是涉及**系统管理的命令**，存储的也是二进制文件，但只有超级用户 **root** 才可以使用，普通用户无权执行这个目录下的命令，这个目录和 **/usr/sbin**、**/usr/lib/debug/sbin** 或 **/usr/local/sbin** 目录相似。目录 **sbin** 中包含的命令只有具有 **root** 权限才能执行的。

3.2 Linux文件类型

Linux 中常用的文件类型有 **5** 种：普通文件、目录文件、链接文件、设备文件和管道文件。

1. 普通文件

Linux 的普通文件是指以字节为单位的数据流类型文件，它是最常用的一类文件，其特点是不包含文件系统的结构信息。如图形文件、数据文件、文档文件、声音文件等都属于普通文件。这种类型的文件按其内部结构又可细分为文本文件和二进制文件。

2. 目录文件

目录文件不存放常规数据，它是用来组织、访问其他文件的。它是内核组织文件系统的节点。目录文件可以包含下一级目录文件或普通文件。在 **Linux** 中，目录文件是一种文件，与其他操作系统中“目录”的概念不同，它是 **Linux** 文件中的一种。

3. 链接文件

链接文件是一种特殊的文件，实际上是指向一个真实存在的文件链接，类似于 **Windows** 下的快捷方式。

3.2 Linux 文件类型

4. 设备文件

设备文件是 **Linux** 中最特殊的文件。**Linux** 系统为外部设备提供一种标准接口，将外部设备视为一种特殊的文件。用户可以像访问普通文件一样访问任何外部设备，使 **Linux** 系统可以很方便地适应不断变化的外部设备。

5. 管道文件

管道文件是一种很特殊的文件，主要用于不同进程的信息传递。当两个进程间需要传递数据或信息时，可以使用管道文件。一个进程将需传递的数据或信息写入管道的一端，另一进程则从管道的另一端取得所需的数据或信息。

3.3 工作目录与目录的切换

Linux 系统使用 **cd** (**change directory**) 命令来切换工作目录，作用是改变当前工作目录。

cd 的命令格式为：**cd [directory]**

该命令将当前目录改变为 **directory** 指定的目录。若没有制定 **directory**，则回到用户的主目录，“~”是 **home** 目录的意思。主目录是当前用户的 **home** 目录，是添加用户时指定的。一般用户默认的 **home** 目录是 **/home/xxx** (**xxx** 是用户名)，**root** 的默认 **home** 目录是 **/root**。

要改变到指定目录，用户必须拥有对指定目录的执行和读权限。该命令可以使用通配符。

如，假设用户当前的目录是 **/root/working**，要更换到 **/user/src** 目录下，可使用如下命令。

```
[root@localhost working]# cd /user/src
```

3.3.1 ls命令

ls (**list**) 命令是用户最常用的命令之一。对于目录，**ls** 命令将输出该目录下的所有子目录与文件；对于文件，**ls** 命令将输出其文件名以及要求的其他信息。该命令类似于**DOS** 下的 **dir** 命令。默认情况下，输出条目按字母顺序排序。

ls 的命令格式为：**ls [option] [names]**

其中，**option** 选项可以省略，常用的参数如表所示。

选项	功能描述
-a	显示指定目录下的所有子目录与文件，包括隐藏文件
-A	显示指定目录下的所有子目录与文件，包括隐藏文件，但不列出“.”和“..”
-d	列出目录文件本身的状态，而不是列出目录下包括的文件内容。常与-l选项联用，以得到目录的详细信息
-l	以长格式显示文件的详细信息。这个选项最为常用，每行列出的信息：文件类型与权限、链接数、文件属主（属主就是所属的主人，即 owner）、文件属组（属组就是 owner 所在的 group）、文件大小、建立或最近修改的时间名字
-L	若指定的名称为一个符号链接文件，则显示链接指向的文件
-n	输出格式与-l选项相同，只不过在输出中文件属主和属组是用相应的 UID 号和 GID 号来表示，而不是实际的名称
-R	递归地列出其中包含的子目录中的文件信息及内容

3.3.2 目录的创建和删除

下面介绍 **Linux** 系统中的目录创建与删除命令。

1. **mkdir** 命令

创建目录需要使用 **mkdir** 命令。

mkdir 的命令格式为: **mkdir [option] [dirname]**

其中, **option** 选项可以省略, **dirname** 是要创建的目录名称。

2. **rmdir** 命令

rmdir 命令只能用来删除空目录, 若目录中存在文件, 就要使用 **rm** 命令删除文件后再删除目录, 后面会详细介绍 **rm** 命令。

rmdir 的命令格式为: **rmdir [option] [dirname]**

其中, **option** 选项可以省略。 **dirname** 表示目录名。

3.3.3 文件的创建、复制、移动和删除命令

文件的创建、复制、移动和删除操作在 **Linux** 系统中使用得相当频繁，下面详细介绍这些操作命令。

1. **touch** 命令

touch 命令有两个功能：一是用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来；二是用来创建新的空文件。

touch 的命令格式为：**touch [option] filename**

其中，**option** 选项可以省略。**filename** 是将要创建的文件的名称。

2. **cp** 命令

该命令的功能是将给出的文件或目录复制到另一文件或目录中。

cp 的命令格式为：**cp [option] source dest**

其中，**option** 选项可以省略。**source** 表示需要复制的文件，**dest** 表示需要复制到的目录。

3. **mv** 命令

用户可以使用 **mv** 命令为文件或目录改名或者将文件由一个目录移动到另一个目录中。

mv 的命令格式为：**mv [option] source dest**

3.3.4 文件的创建、复制、移动和删除命令

4. **rm** 命令

在 **Linux** 系统中，可以使用 **rm** 命令将无用文件删除。该命令的功能是删除一个目录中的一个或者多个文件，也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

rm 的命令格式为：**rm [option] filename**

3.3.5 其他操作

下面介绍 **Linux** 系统中常用的其他操作命令。

1. **sort** 命令

该命令的功能是对文件中的各行进行排序。**sort** 将文件的每一行作为一个单位，相互比较，比较原则是从首字符向后，依次按 **ASCII** 码值进行比较，最后将它们按升序输出。

sort 的命令格式为：**sort [option] filename**

其中，**option** 选项可以省略，**filename** 是操作对象的文件名称。

2. **cat** 命令

该命令的主要功能是用来显示文件内容，依次读取其后所指文件的内容并将其输出到标准输出设备上。还能够用来连接两个或者多个文件，形成新文件。

cat 的命令格式为：**cat [option] filename**

其中，**option** 选项可以省略，**filename** 是操作对象的文件名称。

3. **more** 命令

在查看文件的过程中，可以使用 **more** 命令一次只显示一屏文本，并在终端底部打出“**--more--**”，系统还将同时显示已显示文本占全部文本的百分比。如果要继续显示，可以按回车键或空格键。

more 的命令格式为：**more [option] filename**

3.3.5 其他操作

4. **file** 命令

file 命令用于辨识文件类型。

file 的命令格式为: **file [option] filename**

其中, **option** 选项可以省略, **filename** 是操作对象的文件名称。

3.3.6 文件内容查找命令

文件内容查询命令主要是指 **grep**、**egrep** 等命令。这组命令以指定的查找模式搜索文件，通知用户在什么文件中搜索到与指定模式匹配的字符串，并且打印出所有包含该字符的文本行，该文本行的最前面是该行所在的文件名。这 **3** 个命令的含义分别如下。

（1）**grep** 命令：是最早的文本匹配程序，使用 **POSIX** 定义的基本正则表达式（**BRE**）来匹配文本。该命令一次只能搜索一个指定的模式。

（2）**egrep** 命令：扩展式 **grep**，其使用扩展式表达式（**ERE**）匹配文本。

3.3.7 find命令

Linux 下 **find** 命令在目录结构中搜索文件，并执行指定的操作。该命令的功能是从指定的目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件并对其采取相关的操作。因为此命令提供了相当多的查找条件，功能很强大，所以它的选项也很多。

find 的命令格式为：**find [option] filename**

find 命令提供的寻找条件可以使一个用逻辑运算符 **not**、**and** 和 **or** 组成的复合条件。逻辑运算符 **not**、**and** 和 **or** 的含义如下。

and：逻辑与，在命令中用 “**-a**”表示，是系统默认的选项，表示只有当所给的条件都满足时，寻找条件才算满足。

or：逻辑或，在命令中用 “**-o**”表示。该运算符表示只要所给的条件中有一个满足，寻找条件就算满足。

not：逻辑非，在命令中用 “**!**”表示。该运算符表示查找不满足所给条件的文件。

3.3.8 locate命令

该命令的功能也是查找文件，比 **find** 命令的搜索速度快，原因在于它不搜索具体目录，而是搜索一个数据库（**/var/lib/located**），这个数据库中含有本地所有文件信息。**Linux** 系统自动创建这个数据库，并且每天自动更新一次，所以使用 **locate** 命令查不到最新变动过的文件。为了避免这种情况，可以在使用 **locate** 之前，先使用 **updatedb** 命令，手动更新数据库。

locate 的命令格式为：**locate [option] filename**

例如，搜索 **etc** 目录下所有以 **sh** 开头的文件，忽略大小写区别。

```
[user@localhost ~]$ locate -i /etc/sh
```

```
/etc/shadow
```

```
/etc/shadow-
```

```
/etc/shells
```

3.3.9 whereis命令

whereis 命令用于查找文件。该指令会在特定目录中查找符合条件的文件。这些文件应属于原始代码、二进制文件，或帮助文件。该指令只能用于查找二进制文件、源代码文件和 **man** 手册页，一般文件的定位需使用 **locate** 命令。

whereis 的命令格式为：**whereis [option] filename**

例如，使用命令 **whereis** 查看 **bash** 命令的位置，输入如下命令。

```
[user@localhost ~]$ whereis bash
```

上面的指令执行后，输出信息如下。

```
bash:/bin/bash/etc/bash.bashrc/usr/share/man/man1/bash.1.gz
```

3.3.10 tar命令

tar 是一个归档程序，也就是说，**tar** 命令可以将许多文件打包成为一个归档文件或者把它们写入备份设备，如一个磁带驱动器。所以通常 **Linux** 下，保存文件都是先用 **tar** 命令将目录或者文件打成 **tar** 归档文件（也称 **tar** 包），然后进行压缩。

tar 的命令格式为：**tar [option] filename**

例如，使用 **touch** 命令创建一个文件名为**a.c**的文件。

```
[user@localhost ~]$ touch a.c
```

压缩 **a.c** 文件为 **test.tar.gz**。

```
[user@localhost ~]$ tar -czvf test.tar.gz a.c
```

解压压缩文件内容。

```
[user@localhost ~]$ tar -xzvf test.tar.gz
```

3.3.11 zip命令

zip 命令可以用来解压缩文件，或者对文件进行打包操作。**zip** 是个使用广泛的压缩程序，文件经它压缩后会另外产生具有“**.zip**”扩展名的压缩文件。

zip 的命令格式为：**zip [option] filename**

例，将 **/home/Blinux/html/** 目录下的所有文件和文件夹打包为当前目录下的 **html.zip**。

```
[user@localhost ~]$ zip -r html.zip /home/Blinux/html
```

```
zip -e test.zip test.c
```

```
unzip test.zip
```

3.3.12 常用压缩格式

1. 文件压缩——**gzip** 命令

gzip 命令用于压缩一个或更多文件。执行命令后，原文件会被其压缩文件取代。

gzip的命令格式为：**gzip [option] filename**

例如，压缩 **hello.c**，压缩后，文件以 **gz** 结尾，原始文件已删除。

```
[user@localhost ~]$ gzip hello.c
```

```
[user@localhost ~]$ ls  
hello.c.gz
```

3.3.13 符号链接

符号链接是为了克服硬链接的局限性而创建的。符号链接是通过创建一个特殊类型的文件来起作用的，该文件包含了指向引用文件或目录的文本指针。就这点来看，符号链接与 **Windows** 系统下的快捷方式非常相似，但是，符号链接要早与 **Windows** 的快捷方式很多年出现。

符号链接指向的文件与符号链接自身几乎没有区别。例如，将一些东西写进符号链接里，这些东西同样也写进了引用文件。而当删除一个符号链接时，删除的只是符号链接而没有删除文件本身。如果先于符号链接之前删除文件，那么这个链接依然存在，但不指向任何文件。此时，这个链接就称为坏链接。在很多实现中，**ls** 命令会用不同的颜色来显示坏链接，如红色。

例：

```
ln -s test.c test
```

3.4.1 交换空间

当系统的物理内存不够用时，就需要将物理内存中的一部分空间释放出来，以供当前运行的程序使用。那些被释放的空间可能来自一些很长时间都没有什么操作的程序，这些释放的空间被临时保存到 **Swap** 空间中，等到那些程序要运行时，再从 **Swap** 中恢复保存的数据到内存中。这样，系统总是在物理内存不够时，才进行 **Swap** 交换。其实，**Swap** 的调整对 **Linux** 服务器，特别是 **Web** 服务器的性能至关重要。调整 **Swap**，有时可以越过系统性能瓶颈，节省系统升级费用。

Swap 空间有两种形式：交换分区和交换文件。总之对 **Swap** 的读写都是磁盘操作。

增加交换空间有以下两种方法（严格来说，在系统安装完后，只有一种方法可以增加 **Swap**，那就是下面介绍的第二种方法，至于第一种方法，应该是安装系统时设置交换区）。

方法一：使用分区：在安装 **OS** 时划分出专门的交换分区，空间大小要事先规划好，启动系统时自动进行 **mount**。这种方法只能在安装 **OS** 时设定，一旦设定好，就不容易改变，除非重装系统。

3.4.2 交换空间

方法二：使用 **swapfile**（或者是整个空闲分区）：新建临时 **swapfile** 或者空闲分区，在需要时设定为交换空间，最多可以增加 **8** 个 **swapfile**。交换空间的大小与 **CPU** 密切相关，如在 **i386** 系中，最多可以使用 **2GB** 的空间。在系统启动后，根据需要在 **2GB** 的总容量下增减。这种方法比较灵活，也比较方便，缺点是启动系统后需要手工设置。

运用 **swapfile** 增加交换空间涉及命令如下。

(1) **free**: 查看内存状态命令，可以显示 **memory**、**swap**、**buffer cache** 等的大小及使用状况。

(2) **dd**: 读取，转换并输出数据命令。

(3) **mkswap**: 设置交换区。

(4) **swapon**: 启用交换区，相当于 **mount**。

(5) **swapoff**: 关闭交换区，相当于 **umount**。

例：

```
mkdir /data
```

```
dd if=/dev/zero of=/data/swap bs=1M count=4096
```

```
mkswap /data/swap
```

```
swapon /data/swap
```

4 Linux用户及权限机制

■ 4.1 用户与用户组

- 4.1.1 用户的管理
- 4.1.2 用户组的管理
- 4.1.3 用户配置文件

■ 4.2 文件权限管理

- 4.2.1 所有者、所在组和其他用户
 - 4.2.2 读、写和执行操作
 - 4.2.3 umask属性和特殊权限
 - 4.2.4 文件属性控制
-

4.1.1 用户的管理

Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以该账号的身份进入系统。每个用户账号都拥有一个唯一的用户名和口令，同时系统会为每个用户账号分配一个用户 **ID** (**uid**) 来标识用户。用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

根据用户 **ID** 的不同，在 **Linux** 系统中，用户可分为以下 **3** 种类型。

(1) **root** 用户：又称为超级用户，**ID** 为 **0**，拥有最高权限。

(2) **系统用户**：又称为虚拟用户、伪用户或假用户，不具有登录**Linux** 系统的能力，但却是系统运行不可缺少的用户，一般 **ID** 为 **1~499**，本书中使用的**Centos7** 为 **1~999**。

(3) **普通用户**：**ID** 为 **500** 以上，**Centos7** 为 **1000** 以上。可以登录**Linux** 系统，但是使用的权限有限，由管理员创建。

用户管理的常用命令包括：**useradd**、**passwd**、**usermod**、**userdel**。**useradd** 用来添加用户，**passwd** 修改用户口令，**usermod** 修改用户信息，**userdel** 删除用户。

4.1.1 用户的管理

1. 添加用户

添加用户就是在系统中创建一个新账号，并为新账号分配用户 **ID**、用户组、主目录和登录 **Shell** 等资源。通过 **useradd** 命令添加用户。

useradd 的命令格式为：**useradd [option] username**
username 表示新账号的登录名。

2. 修改用户口令

用户账号刚创建时没有口令，被系统锁定无法使用，必须为其指定口令后才可以使⽤，即使是空口令。使⽤ **passwd** 命令指定和修改用户口令。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。

passwd 的命令格式为：**passwd [option] [username]**

其中，**option** 选项可以缺省，主要对 **/etc/shadow** 文件的字段产生影响，**username** 参数也可以缺省，没有指定该参数时，表示修改当前用户的口令；如果指定了该参数，则表示修改指定用户的口令，只有 **root** 用户才有修改指定用户口令的权限。

4.1.1 用户的管理

3. 修改用户信息

修改用户信息就是更改用户的属性，如用户 **ID**、主目录、用户所在组、登录 **Shell** 等。通过 **usermod** 命令修改用户信息。

usermod 的命令格式为：**usermod [option] username**
username 表示用户名。

4. 删除用户

如果一个用户账号不再使用，可以从系统中删除。删除用户就是删除与用户有关的系统配置文件中的记录（如 **/etc/passwd**）。通过 **userdel** 命令删除用户。

userdel 的命令格式为：**userdel [option] username**

其中，**option** 选项最常用的参数是 **-r**，表示同时删除用户的主目录。

username 表示要删除的用户。例如，删除用户 **user2**，执行如下命令。

```
[root@localhost Desktop]# userdel -r user2
```

此时再查看 **/etc/passwd** 文件中的信息，将不会找到与 **user2** 有关的信息行。

4.1.2 用户配置文件

与用户相关的系统配置文件主要有 **/etc/passwd**、**/etc/shadow**、**/etc/group**。**/etc/passwd** 文件保存用户信息，**/etc/shadow** 文件保存加密的用户密码，**/etc/group** 文件保存用户组信息。

1. **/etc/passwd** 文件

系统中所有的用户信息都会记录到**/etc/passwd** 文件中，是系统识别用户的一个文件。当用户登录时，系统首先查阅 **/etc/passwd** 文件。假设用户名为 **user**，则会在**/etc/passwd** 文件中查看是否有该账号，然后确定 **user** 的 **uid**，通过 **uid** 确认用户和身份。在**/etc/passwd** 文件中，每一行都表示一个用户的信息。每行有 **7** 个字段，字段之间通过“:”分隔。例如：

```
[root@localhost Desktop]# cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
user:x:1001:1001::/home/user:/bin/bash
```

4.1.2 用户配置文件

2. **/etc/shadow** 文件

/etc/shadow 与 **/etc/passwd** 文件是互补的，由于 **/etc/passwd** 文件所有用户都可以访问，为保证安全，将密码和其他 **/etc/passwd** 文件不能包括的信息（如有效期限）单独保存在 **/etc/shadow** 中，此文件只有 **root** 用户有权查看。例如，用 **user** 用户查看该文件时，会出现提示 “**Permission denied**（权限不足）”。

```
[user@localhost Desktop]$ cat /etc/shadow  
cat: /etc/shadow: Permission denied
```

3. **/etc/group** 文件

/etc/group 文件是用户组的配置文件，可以直观看出用户组中包括哪些用户。每个用户组是一条记录，每个记录包含 **4** 个字段，字段之间通过 “:” 分隔，具体内容如下。

```
[root@localhost Desktop]# cat /etc/group  
root:x:0:myu  
user:x:1001:  
user1:x:1002:  
myuser:x:1003:  
myGroup:x:1004:
```

4.2.1 所有者、所在组和其他用户

有 **3** 种类型的用户可以访问文件或目录：文件所有者、同组用户、其他用户。所有者一般是文件的创建者，对该文件的访问权限拥有控制权。所有者可以允许同组用户有权访问文件，还可以将文件的访问权限赋予系统中的其他用户。通过 **chown** 和 **chgrp** 命令可以改变文件的所属用户和所属组。

1. 改变文件所属用户

通过 **chown** 命令将文件的所有者修改为指定的用户，普通用户不能将自己的文件改变成其他的拥有者，超级用户才拥有此权限。

chown 的命令格式为：**chown [option] [owner][:[group]] file**。

owner 表示文件的所有者，可以是用户名，也可以是用户 **ID**。**group** 表示文件的所在组，组名或者组 **ID** 均可。**file** 是文件的名称。

2. 改变文件所在组

通过 **chgrp** 命令变更目录和文件的所属组，只有超级用户才拥有此权限。

chgrp 的命令格式为：**chgrp [option] group file**。

group 表示目录或文件的所在组，可以是组名或者组 **ID**。**file** 是目录或文件的名称。

4.2.2 读、写和执行操作

每一文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写、执行权限，与属主同组用户的读、写和执行权限，系统中其他用户的读、写和执行权限。文件被创建时，文件的所有者自动拥有对该文件的读、写、执行权限，以便于阅读和修改文件。用户也可根据需要把访问权限设置为需要的任何组合，目录必须拥有执行权限，否则无法查看其内容。**r**表示读权限，**w**表示写权限，**x**表示执行权限。三种权限模式对文件和目录的影响如表所示。

权限	对文件的影响	对目录的影响
r (读)	可读取文件内容	可列出目录内容
w (写)	可修改文件内容	可在目录中创建、删除文件
x (执行)	可作为命令执行	可访问目录内容

当用 **ls -l** 命令显示文件或目录的详细信息时，最左边一列为文件的访问权限。

/etc/passwd 文件的详细信息如下。

```
[root@localhost Desktop]# ls -l /etc/passwd
```

```
-rw-r--r--. 1 root root 2418 Mar 22 10:48 /etc/passwd
```

4.2.2 读、写和执行操作

输出结果的前十个字符 “**-rw-r--r--**”表示文件属性，第一个字符表示文件类型，剩下的 **9** 个字符（三个一组）分别表示文件所有者、文件所在组以及其他用户对该文件的读、写和执行权限，具体含义如图所示。



通过 **chmod** 命令改变不同用户对文件或目录的访问权限，文件或目录的所有者和超级用户拥有修改权限。该命令有以下两种使用方法：表达式法和数字法。

4.2.2 读、写和执行操作

1. 表达式法

表达式法的**chmod**的命令格式为：**chmod [who] [operator] [mode] file**。

其中，**who** 指定用户身份，若此参数省略，则表示对所有用户进行操作。**operator** 表示添加或取消某个权限，取值为“+”或“-”。**mode** 指定读、写、执行权限，取值为 **r**、**w**、**x** 的任意组合。

2. 数字法

chmod 支持以数字方式修改权限，读、写、执行权限分别由 **3** 个数字表示：**r**（读）=**4**；**w**（写）=**2**；**x**（执行）=**1**。每组权限分别为对应数字之和，如“**rw-rw-r--**”表示为“**664**”（其中的-表示没有某一个权限，取值为**0**）。当前**a**文件的访问权限为“**rw-rw-rw-**”，将其恢复到原来的“**rw-rw-r--**”，只需要执行命令“**chmod 664 a**”，结果如下。

```
[root@localhost four]# chmod 664 a
```

```
[root@localhost four]# ls -ld a
```

```
-rw-rw-r--. 1 myUser myGroup 0 Mar 28 15:20 a
```