

## 第 3 章 Linux 文件与目录管理

### 学习目标

- (1) 了解 Linux 常用文件类型。
- (2) 了解 Linux 目录的结构。
- (3) 掌握 Linux 文件权限与属主的设置。
- (4) 掌握 Linux 常用的基本命令的用法。

文件管理是学习和使用 Linux 系统的基础,也是 Linux 系统管理与维护中最重要的部分之一。本章将对 Linux 系统的文件与目录的基础知识,以及文件管理操作中的一些重要或者常见的命令做较为详细的介绍。

### 3.1 Linux 文件与目录的基本命令

在 UNIX 系统中有一个重要的概念:一切都是文件。其实这是 UNIX 哲学的一个体现,而 Linux 是重写 UNIX 而来,所以这个概念也就传承了下来。在 UNIX 系统中,把一切资源都看作是文件,包括硬件设备。UNIX 系统把每一个硬件都看成是一个文件,通常称为设备文件,这样用户就可以用读写文件的方式实现对硬件的访问。

在 Linux 系统中,任何软件和 I/O 设备都被视为文件。Linux 系统的文件名最大支持 256 个字符,分别可以用 A~Z、a~z、0~9 等字符来命名。与 Windows 不同,Linux 系统文件名是区分大小写的,所有的 UNIX 系列操作系统都遵循这个规则。Linux 下也没有盘符的概念(如 Windows 下的 C 盘、D 盘),而不同的磁盘分区是被挂载在不同的目录下的。

Linux 系统的文件没有扩展名,所以 Linux 系统的文件名称和它的种类没有任何关系。例如,abc.exe 可以是文本文件,而 abc.txt 也可以是可执行文件。Linux 系统的文件可以分为五种不同的类型: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

普通文件一般是用一些相关的应用程序创建的,例如图像、文件、压缩包等。依照文件的内容,通常用“-”符号标识,具体有以下两种。

(1) 文本文件:它是以文本的 ASCII 编码形式存储,是 Linux 系统中最多的一种文件类型,又称为纯文字文件,是因为可以从文件中直接读到如数字、字母等数据。几乎所有有系统配置的文件都属于这一文件类型。

(2) 二进制文件：它是以文本的二进制形式存储，文件内容不可读，仅仅系统可以执行这些二进制文件，Linux 系统所有的命令都是这种格式。

## 2. 目录文件

目录文件简称目录，存储一组相关文件的位置、大小等信息。通常用“d”标识，例如 drwxrwxrwx 代表目录文件。

## 3. 链接文件

链接文件类似 Windows 系统中的快捷方式，用 l 标识，例如 lrwxrwxrwx 代表链接文件。链接文件可以分为以下两种。

(1) 硬链接文件：保留所链接文件的索引节点(磁盘的具体位置)信息。

(2) 符号链接文件：类似于 Windows 中的快捷方式，其本身并不保存文件的内容，而只记录所连接文件的路径。

## 4. 设备文件

设备文件是存储 I/O 设备信息的文件，Linux 系统的每个设备都用一个设备文件来表示。设备文件通常集中在 /dev 目录下，通常又分为以下两种。

(1) 区块设备文件：就是存储一些数据，以提供系统随机存取的接口设备。例如磁盘就可以随机在其不同区块读写，通常用 b 来标识这类文件。

(2) 字符设备文件：是一些串行端口的接口设备，例如键盘、鼠标等。这些设备的特点就是一次性读取，不能够截断输出，通常用 c 来标识这类文件。

## 5. 管道文件

管道文件是一种特殊的文件，主要的目的是解决多个程序同时存取一个文件所造成的错误问题，通常用 p 来标识这类文件。管道是 Linux 系统很重要的一种通信方式，以间隔两个程序，从而实现把前面一个程序的输出直接连接到后面另一个程序的输入。

Linux 系统中，文件与目录的操作是最基本、最重要的技术。用户可以方便、高效地通过系统提供的命令对文件和目录进行操作，为了更好地介绍本章内容，本节将分别对其最常用的几个基本命令进行介绍。

### 3.1.1 查看文件与目录命令 ls

ls 命令是英文单词 list 的简写，这是用户最常用的命令之一，该命令的功能是输出指定目录中所有的子目录与文件的文件名以及所要求的其他信息，在默认情况下，输出条目按字母顺序排序。当未给出目录名或文件名时，就显示当前目录的信息。ls 命令常用格式如下。



```
ls [选项][dirname|filename]
```

该命令中的[选项]含义如下。

-a: 显示指定目录下所有子目录与文件,包括隐藏文件。

-A: 显示指定目录下所有子目录与文件,包括隐藏文件,但不列出“.”和“..”。

-b: 对文件名中的不可显示字符用八进制字符显示。

-c: 按文件的修改时间排序。

-C: 分成多列显示各项。

-d: 如果参数是目录,只显示其名称而不显示其下的各文件。往往与 l 选项一起使用,以得到目录的详细信息。

-l: 以长格式来显示文件的详细信息,这个选项最常用。每行列出的信息依次是文件类型与权限、链接数、文件属主、文件属组、文件大小、建立或最近修改的时间以及名字,运行效果等同于命令 ll。

```
[root@localhost ~]# ls
公共模板视频图片文件下载音乐桌面  anaconda-ks.cfg
[root@localhost ~]# ls -a
.      模板文件桌面      .bash_logout  .cache  .esd_auth  .pki
..     视频下载  anaconda-ks.cfg  .bash_profile  .config  .ICEauthority  .tcshrc
公共图片音乐  .bash_history  .bashrc      .cshrc  .local
[root@localhost ~]# ls -A
公共图片音乐      .bash_history  .bashrc  .cshrc      .local
模板文件桌面      .bash_logout  .cache  .esd_auth  .pki
视频下载  anaconda-ks.cfg  .bash_profile  .config  .ICEauthority  .tcshrc
// 以符号“.”开头的文件均为隐藏文件,均为系统配置文件,所以默认对用户不可见

[root@localhost ~]# ls -d /root
/root
[root@localhost ~]# ls -dl /root
dr-xr-x---. 15 root root 4096 11月  9 22:04 /root
```

### 3.1.2 目录的相关操作命令 pwd、cd、mkdir

#### 1. pwd 命令

pwd 命令是英文词组 print working directory 的缩写,作用是显示当前工作目录的路径。该命令无参数和选项,在 Linux 层次目录结构中,用户可以在被授权的任意目录下用 mkdir 命令创建新目录,也可以用 cd 命令从一个目录切换到另一个目录。然而,没有提示符来告知用户目前处于哪一个目录中,要想知道当前所处的目录,可以用 pwd 命令,该命令显示整个路径名。其实该命令显示的是当前工作目录的绝对路径。

## 2. cd 命令

cd 命令是英文词组 change directory 的缩写,作用是改变当前工作目录。cd 命令的常用格式如下。

```
cd [directory]
```

该命令将当前目录改变至 directory 所指定的目录。若没有指定 directory,则回到用户的主目录。为了切换到指定目录,用户必须拥有对指定目录的执行和读权限。该命令也可以使用通配符。最重要的是区分出以下命令的不同。

```
cd ..: 回到上层目录。
cd .: 什么也不做。
cd -: 回到刚才目录。
cd ~: 回到自己的主目录。
cd: 回到自己的主目录。
```

从前两个命令可以明确看出符号“.”和“..”的含义:前者指向当前目录,即 pwd 命令所返回的目录;而后者代表当前目录的上一级目录。

## 3. mkdir 命令

创建目录需要使用 mkdir 命令。mkdir 命令的常用格式如下。

```
mkdir [选项] [dirname]
```

该命令创建名为 dirname 的目录,mkdir 命令要求创建目录的用户在当前的目录(即 dirname 的父目录)中具有写权限,并且 dirname 不能是当前目录中已有的目录或文件名称,否则就需要加入参数-p。

参数-p: 实现递归创建目录,而如果待创建目录已经存在,则不报错误。

```
t@localhost ~]# cd
[root@localhost ~]# mkdir test
[root@localhost ~]# cd test
[root@localhost test]# pwd
/root/test
[root@localhost test]# mkdir test2/test3
mkdir: 无法创建目录“test2/test3”: No such file or directory
[root@localhost test]# mkdir -p test2/test3
[root@localhost test]# cd test2/test3/
[root@localhost test3]# pwd
/root/test/test2/test3
//要想直接创建多级目录,就必须加入参数-p,才能实现递归创建,否则报错

[root@localhost test3]# cd ..
[root@localhost test2]# pwd
```

```

/root/test/test2
[root@localhost test2]# cd ..
[root@localhost test]# pwd
/root/test
[root@localhost test]# cd ..
[root@localhost ~]# pwd
/root
//通过三次回退到上级目录,实现从 /root/test/test2/test3 到 /root

[root@localhost ~]# mkdir test
mkdir: 无法创建目录“test”: File exists
//mkdir 创建的目录如果已经存在,则报错

[root@localhost ~]# mkdir -p test
//如果加入参数-p,则不会报错

```

### 3.1.3 复制、移动与删除命令 cp、mv、rm

#### 1. cp 命令

该命令的功能是将给出的文件或目录复制到另一个文件或目录。cp 命令的常用格式如下。

```
cp [选项] [源文件或目录] [目标文件或目录]
```

该命令的[选项]含义如下。

- a: 该选项通常在复制目录时使用,它保留链接和文件属性,并递归地复制目录,其作用等于 dpr 选项的组合。
- p: 此时 cp 除复制源文件的内容外,还将修改时间和访问权限也复制到新文件中。
- d: 若源文件为链接文件,则复制链接文件的属性而非文件本身。
- r: 若源文件是目录文件,此时 cp 将递归复制该目录下所有的子目录和文件,此时目标文件必须是一个目录名。
- f: 不询问,直接覆盖已经存在的目标文件而不提示。
- i: 和-f 选项相反,在覆盖目标文件之前将给出提示要求用户确认,回答 y 时目标文件将会覆盖,是交互式复制。

需要说明的是,如果用户指定的目标文件名是一个已存在的文件名,用 cp 命令复制文件后,这个文件就会被新复制的源文件覆盖。因此,系统为防止用户在不经意的情况下用 cp 命令破坏另一个文件,默认设置了-i 选项。

```

[root@localhost ~]# mkdir -p user/shiyan
[root@localhost ~]# touch bbb.txt

```



```
//创建一个空白文件

[root@localhost ~]#cp bbb.txt user/shiyan/
//表示将 bbb.txt 文件复制到 user/shiyan 目录下

[root@localhost ~]#cd user/shiyan
[root@localhost shiyan]#ls -l
总用量 0
-rw-r--r-- 1 root root 0  2月  8 01:35 bbb.txt
```

## 2. mv 命令

用户可以使用 mv 命令为文件或目录改名或将文件由一个目录移入另一个目录中。mv 命令的常用格式如下。

```
mv [选项][源文件或目录][目标文件或目录]
```

该命令的[选项]含义如下。

-i: 交互式操作,如果 mv 操作导致对已存在的目标文件的覆盖,此时系统询问是否重写,要求用户回答 y 或者 n,这样可以避免覆盖文件。

-f: 禁止交互操作,在 mv 操作中覆盖已有的目标文件不给任何指示,指定此选项后,-i 选项将不再起作用。

如果所给的目标文件(不是目录)已存在,此时该文件的内容将被新文件覆盖。为防止用户在不经意的情况下用 mv 命令破坏另一个文件,建议用户在使用 mv 命令移动文件时,最好使用-i 选项。需要注意的是,mv 和 cp 的结果不同,mv 好像文件搬家,文件个数并未增加;而 cp 则是对文件进行复制,文件个数增加了。

```
[root@localhost ~]#mkdir -p test
[root@localhost ~]#cd test
[root@localhost test]#cp /proc/cpuinfo cpuinfo2
[root@localhost test]#ll
总用量 4
-r--r--r-- . 1 root root 1091 11月 17 10:40 cpuinfo2
[root@localhost test]#mv cpuinfo2 cpuinfo3
[root@localhost test]#ll
总用量 4
-r--r--r-- . 1 root root 1091 11月 17 10:40 cpuinfo3
```

## 3. rm 命令

对于无用的文件,用户可以用 rm 命令将其删除,该命令的功能为删除一个目录中的一个或多个文件,它也可以将某个目录及其下的所有文件、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 命令的常用格式如下。

```
rm [选项] 文件名
```

该命令的[选项]含义如下。

-f: 忽略不存在的文件,从不给出提示。

-r: 指示 rm 将参数中列出的全部目录和子目录均递归地删除。

-i: 进行交互式删除。

使用 rm 命令要格外小心,因为一旦一个文件被删除,它是不可能被恢复的,所以系统默认设置了-i 这个选项。

```
[root@localhost ~]#rm bbb.txt
rm:是否删除普通空文件 "bbb.txt"?y
[root@localhost ~]#ls -l bbb.txt
ls:无法访问 bbb.txt:没有那个文件或目录
[root@localhost ~]#rm -rf user
[root@localhost ~]#ls -l user
ls:无法访问 user:没有那个文件或目录
```

## 3.2 Linux 目录与目录配置

在计算机系统中存在大量的文件,如何有效地组织和管理它们,并为用户提供一个方便的接口是文件系统的主要任务。Linux 系统以文件目录的方式来组织和管理系统中的所有文件。所谓文件目录就是将所有文件的说明信息采用树状结构组织起来,目录也是其中一种类型的文件,各个目录节点之下都会有一些文件和子目录,所以 Linux 目录为管理文件提供了一个方便的途径。

### 3.2.1 Linux 的目录树

因为利用 Linux 开发应用软件产品或系统发行版的团队实在太多,如果每个人都用自己的想法来配置文件放置的目录,那么将可能造成很多管理上的困扰。所以,Linux 目录的配置依据 FHS(Filesystem Hierarchy Standard)规范。根据 FHS 的官方文件指出,主要目的是希望让用户可以了解到数据通常放置于哪个目录下。也就是说,FHS 的重点在于规范每个特定的目录下应该要放置什么样的数据。FHS 定义出两层规范:第一层是根目录(/)底下的各个目录应该要放置什么样内容的文件数据,例如,/etc 应该要放置系统配置文件,/bin 和/sbin 则应该放置可执行程序等;第二层则是针对/user 和/var 这两个目录的次目录来定义的,例如,/var/log 放置系统登录文件,/user/share 放置共享数据等。

在 Linux 系统中,所有的文件与目录都是从根目录(/)开始的,然后再一个一个地分支下来,像是树枝状。因此,人们也称这种目录配置方式为目录树(Directory Tree)。这个目录树具有以下特性。

(1) 目录树的起始点为根目录(/)。

(2) 每一个目录不仅能使用本地端分区的文件系统,也可以使用网络上的文件系统。例如可以利用 Network File System(NFS)服务器挂载某特定目录等。

(3) 每一个文件在此目录树中的文件名(包括完整路径)都是独一无二的。

目录可以分成四种类型。

(1) 可分享的目录:指的是可以分享给其他系统挂载的目录。

(2) 不可分享的目录:系统或程序相关的配置文件等,由于与自身机器有关,所以不适合分享给其他主机。

(3) 不变的目录:不变的数据。

(4) 可变的目录:可变的数据。

FHS 规范将目录分为四种交互作用的形态,如表 3.1 所示。

表 3.1 目录四种交互作用形态

种 类	可分享的(shareable)	不可分享的(unshareable)
不变的(static)	/usr(常用软件)	/etc(配置文件)
	/opt(第三方软件)	/boot(与系统启动有关的文件)
可变的(variable)	/var/mail(使用者邮件信箱)	/var/run(程序相关)
	/var/spool/news(新闻组)	/var/lock(程序相关)

目录树的形式如图 3.1 所示,而从根目录(/)分支下来的众多目录应放置的文件内容如表 3.2 所示,其中有 5 个目录不可与根目录分开放在不同的分区。这 5 个目录分别为 etc、bin、dev、lib、sbin。

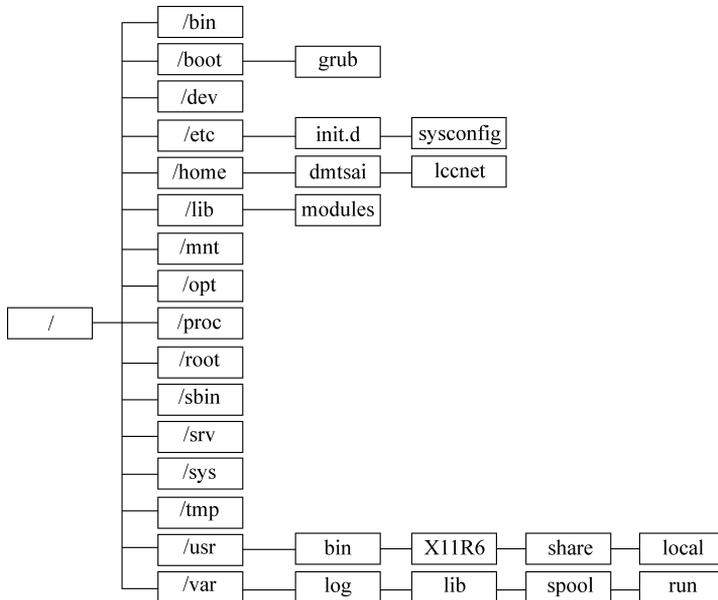


图 3.1 Linux 的目录树

表 3.2 各目录应放置的文件内容

目录	应放置的文件内容
/bin	系统有很多放置执行文件的目录,但是 bin 比较特殊,因为 bin 放置的是在单用户维护模式下还能被操作的命令。在 bin 下面的命令可以被 root 与一般账号所使用,主要有 cat、chmod、date、mv、mkdir、cp、bash 等常用命令
/boot	主要放置开机会使用到的文件,包括 Linux 内核文件以及开机菜单与开机所需配置文件等,如常用文件 vmlinux。如果使用的是 grub 引导装载程序,还会存在 /boot/grub 这个目录
/dev	在 Linux 系统中,任何设备与接口设备都是以文件的形式存储在这个目录中,只要通过访问这个目录下面的文件,就等于访问某个设备,比较重要的文件有 null、zero、tty、lp *、hd *、sd * 等
/etc	系统的主要配置文件几乎都放在这个目录内,例如人员的账号、密码文件,各种服务的起始文件等。一般来说,这个目录下的各文件属性可以让一般用户查阅,但只有 root 才有权利修改。FHS 建议不要放置可执行文件(binary),在这个目录中,比较重要的有 inittab、init.d、modprobe.conf、/X11、fatab、sysconifg 等。①init.d: 所有服务的默认启动脚本都放在这里,例如要启动或者关闭 iptables: /etc/init.d/iptables start stop。②xinetd.d: super daemon 管理的各项服务配置文件目录。③X11: 与 X Window 有关的各种文件都放在这里,尤其是 xorg 和 XServer 的配置文件 conf
/home	系统默认的用户主文件夹。当创建一个一般用户账号时,默认的用户文件夹都会放置在这里。~代表目前这个用户的主文件夹
/lib	系统的函数库非常多,而 lib 放置的则是开机时用到的函数库,以及 bin 或者/sbin 下面的命令会调用的函数库。modules 这个目录会放置内核相关的模板(驱动程序)
/media	放置可删除的设备,例如软盘、光盘、DVD 等
/mnt	如果想暂时挂载某些额外的设备,一般建议放置在此目录中
/opt	放置第三方软件的目录,以前习惯放置在 /uer/local 中
/root	系统管理员 root 的主文件夹。之所以放在这里,是因为如果进入单用户维护模式而挂载根目录时,该目录能够拥有 root 的主文件夹。建议放在根目录下
/sbin	放在/sbin 下面的为开机过程所需要的,里面包括的开机、修复、还原系统所需要的命令。例如 fdisk、fsck、ifconifg、mkfs、init、mount。sbin 目录下的命令普通用户都执行不了
/srv	服务启动之后需要访问的数据目录,如 WWW 服务需要访问的网页数据放置在 /srv/www 内
/tmp	放置临时文件目录,一些命令和应用程序会用到这个目录。该目录下的所有文件会被定时删除,会避免临时文件占满整个磁盘。临时文件,在系统重启时目录中文件不会保留

### 3.2.2 Linux 的目录基本概念

在了解 Linux 的目录时,要了解一些常见的基本概念。下面分别介绍路径、根目录、工作目录、用户主目录等一些相关的基本概念。

#### 1. 路径

首先了解路径的概念,掌握了路径就可以随心所欲地进入任何目录,进行各种各样的工作。所谓的路径是指从树状目录中的某个目录层次到某个文件的一条道路,路径

是由目录或目录和文件名构成的,例如/etc/x11 就是一个路径,而/etc/x11/xorg.conf 也是一个路径。也就是说,路径可以是目录的组合,分级深入进去,也可以由目录+文件构成。

路径有绝对路径和相对路径之分。在 Linux 系统中,绝对路径是从根目录(/)开始写起的文件名或目录名称,例如/user、/etc/x11 等。进入绝对路径的举例如下。

```
[root@localhost ~]#pwd
/root
[root@localhost ~]#cd /proc
[root@localhost proc]#pwd
/proc
```

相对路径是以“.”或“..”开始的,“.”表示用户当前操作所处的位置,而“..”表示上级目录,在路径中,“.”表示用户当前操作所处的目录,而“..”表示上级目录,要把“.”和“..”当成目录来看。进入相对路径的举例如下。

```
[root@localhost ~]#pwd
/root
[root@localhost ~]#cd .
[root@localhost ~]#pwd
/root
//还在当前目录下,位置没有发生变化
[root@localhost ~]#cd ..
[root@localhost /]#pwd
/
```

## 2. 根目录

Linux 的根目录(/)是 Linux 系统中最特殊的目录,因为所有的目录都是由根目录衍生出来的,同时根目录也与开机、还原、系统修复等动作有关。因此,FHS 标准建议:根目录(/)所在的分区越小越好,且各种应用软件最好不要与根目录放在一个分区,以保持根目录越小越好,如此不但效能最佳,根目录所在的文件系统也不容易出现问題。

## 3. 工作目录

从逻辑上讲,用户登录 Linux 系统之后,每时每刻都处在某个目录之中,此目录被称作工作目录或当前目录。工作目录是可以随时改变的,用“.”表示,其父目录用“..”表示。

## 4. 用户主目录

用户主目录是系统管理员在增加用户时为该用户建立起来的目录,每个用户都有自己的主目录,不同用户的主目录一般互不相同,也可以改变用户的主目录。在默认情况下,用户主目录是/home 目录下与用户名相同的目录。用户刚登录系统时,工作目录便

是该用户的主目录,用户可以通过一个“~”字符来引用自己的主目录,例如,用户 lhy 的主目录/home/lhy,那么命令“ls ~/files”和命令“ls /home/lhy/files”的意义相同。

### 3.2.3 全局变量 PATH

命令 ls 的绝对路径是/bin/ls,但是为什么在任何目录都可以直接只输入 ls,而不用每次输入其绝对路径呢?这就是全局变量 PATH 在起作用。全局变量 PATH 决定了 Shell 将到哪些目录中寻找命令或程序,PATH 的值是一系列的目录,当运行一个程序时,Linux 在这些目录下进行搜寻编译连接。其格式如下。



```
PATH=<PATH 1>:<PATH 2>:<PATH 3>:...:<PATH N>
```

查看 PATH 环境变量,可用:

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

需要注意的是,不同用户的 \$PATH 是不一样的,root 最完整,具体设置 PATH 变量的值请参见 5.4 节。

一个 PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/root/bin 环境变量告诉系统,当接到用户送入的命令时,依次检索/usr/local/sbin、/usr/sbin、/sbin 等目录,直到找到用户输入的命令为止,或者遍历后没有检索成功则报错退出。

```
[root@localhost ~]#lstest
//命令未找到

[root@localhost ~]#mkdir -p /root/bin
[root@localhost ~]#cp /bin/ls /root/bin/lstest
[root@localhost ~]#lstest
bak bin 公共的模板 视频图片文件下载 音乐桌面
//把 ls 命令复制到变量 PATH 所包含的目录/root/bin 并改名为 lstest 就可以运行
```

## 3.3 Linux 的文件权限设置

### 3.3.1 Linux 的文件权限

Linux 系统是一个典型的多用户系统,不同的用户处于不同的地位。为了保护系统的安全性,Linux 系统对不同用户访问同一文件的权限做了不同的规定。文件的访问权限包括读取权限(r)、写入权限(w)和执行权限(x),下面分别对这三种权限做具体的

介绍。

(1) 读取权限(r)是浏览文件或者目录中内容的权限,所以当具有读取一个目录的权限时,就可以使用 ls 这个命令将该目录的文件列表显示出来;而当具有读取一个文件的权限时,就可以显示该文件的具体内容。

(2) 写入权限(w)是修改文件内容的权限,而对于目录而言是删除、添加和重命名目录内文件的权限。

(3) 执行权限(x)对可执行文件而言是允许执行的权限,而对目录来讲是获得进入该目录的权限。

与文件权限相关的用户分类总共可分为三类:文件的所有者(Owner)、同组用户(Group)和其他用户(Other)。

```
[root@localhost ~]#ls -l
总用量 40
drwxr-xr-x. 2 root root 4096 4月 10 23:02 公共
drwxr-xr-x. 2 root root 4096 4月 10 23:02 模板
drwxr-xr-x. 2 root root 4096 4月 10 23:02 视频
drwxr-xr-x. 2 root root 4096 4月 10 23:02 图片
drwxr-xr-x. 2 root root 4096 4月 10 23:02 文档
drwxr-xr-x. 2 root root 4096 4月 10 23:02 下载
drwxr-xr-x. 2 root root 4096 4月 10 23:02 音乐
drwxr-xr-x. 2 root root 4096 4月 10 23:02 桌面
-rw----- . 1 root root 899 4月 5 01:00 anaconda-ks.cfg
drwxr-xr-x. 3 root root 4096 4月 11 10:37 test
/* [ 1 ] [2] [3] [4] [5] [ 6 ] [7] * /
```

命令的运行结果分 7 栏显示。

(1) 第一栏代表了十个属性,第一个属性代表文件的类型,例如目录标注为 d,而普通文件标注为-;剩下的三个一组:r 代表可读,w 代表可写,x 代表可执行。第一组是拥有人的权限,第二组是同群组的权限,第三组是其他用户的权限。

(2) 第二栏为连接数,表示有多少文件名连接到该节点。

(3) 第三栏代表这个文件的拥有人。

(4) 第四栏代表拥有人的群组。

(5) 第五栏代表文件的大小。

(6) 第六栏代表这个文件的创建时间或者是最后被修改的时间。

(7) 第七栏代表目录或文件的名字。

文件属性介绍示意图如图 3.2 所示。

Linux 文件的基本属性一共有九个,分别是 owner/group/other 的读取权限、写入权限和执行权限。需要注意的是,文件初始访问权限在创建时由系统赋予,文件所有者或超级用户可以修改文件权限。文件的访问权限的表示有两种方法:字母表示法和数字表示法。

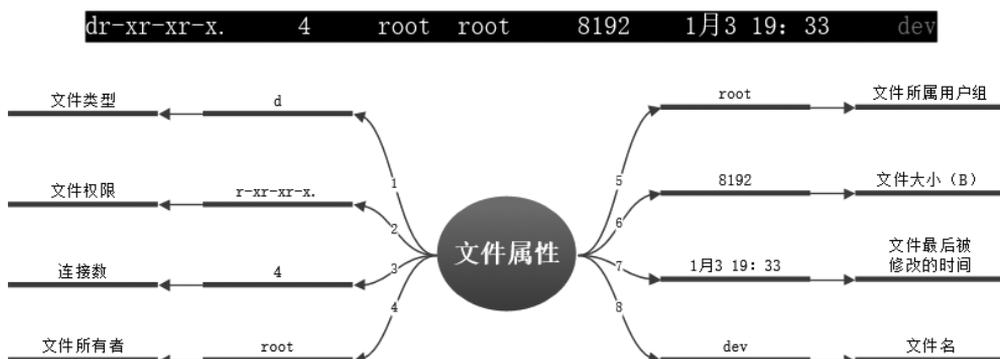


图 3.2 文件属性介绍示意图

### 1. 字母表示法

字母表示法是文件权限表示的一种常用的方法,具体示意如图 3.3 所示。

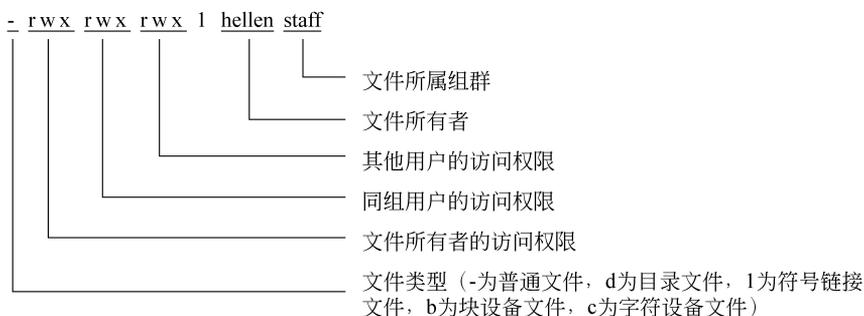


图 3.3 文件权限字母表示法

### 2. 数字表示法

数字表示法就是用数字表示各属性,各属性与数字的对照为 r:4、w:2、x:1。不同的组合会代表不同的数字,一共会有从 0~7 共 8 种组合: --表示 0,--x 表示 1,-w-表示 2,-wx 表示 3,r--表示 4,r-x 表示 5,rw-表示 6,rwx 表示 7。例如,当文件权限字母法表示为 -rwxrwx--,即 owner=rwx=7,group=rwx=7,other=---=0,对应的数字表示法的属性就为 770。

#### 3.3.2 桌面环境改变文件权限

桌面环境下选中要修改文件权限的文件右击,在弹出的快捷菜单中选择“属性”命令,在弹出文件的“属性”对话框中单击“权限”选项卡。具体的操作步骤如图 3.4 和图 3.5 所示。



图 3.4 桌面环境下进入文件属性



图 3.5 桌面环境下改变文件权限

### 3.3.3 修改文件权限的 Shell 命令

确定了一个文件的访问权限后,用户可以利用 Linux 系统提供的 `chmod` 命令来重新设定不同的访问权限,也可以利用 `chown` 命令来更改某个文件或目录的所有者,利用 `chgrp` 命令来更改某个文件或目录的用户组。

#### 1. `chmod` 命令

`chmod` 命令用于改变文件或目录的访问权限。该命令有两种用法:一种是包含字母和操作符表达式的文字设定方法;另外一种包含数字的数字设定法。所以会有如下两种格式。

```
chmod 数字模式 文件名
chmod 功能模式 文件名
```



数字模式为一组三个数字,在 3.3.2 节介绍的数字表示法中就已经解释了该如何用数字表示文件的访问属性。功能模式可由三部分组成,包括对象、操作符和权限。对象包括文件所有者(user)、同组用户(group)和其他用户(other),操作符包括增加权限(+),删除权限(-)和赋予给定权限(=),权限有读取权限(r)、写入权限(w)和执行权限(x)。

在一个命令行中可给出多个权限方式,其间用逗号隔开。例如,`chmod g+r,o+r aaa.txt` 表示同组和其他用户对文件 `aaa` 有读的权限。另外在命令行中加入 `-R`,就是对目录下的所有文件与子目录进行相同的权限变更(以递归的方式逐个变更)。

下面的两种方法,都让对应文件所有人均具有读、写、执行的权限。

(1) 变更属性的指令 `chmod` 的数字用法如下。

```
[root@localhost ~]# cd
[root@localhost ~]# cp anaconda-ks.cfg test1.cfg
[root@localhost ~]# ll test1.cfg
-rw----- . 1 root root 899 4月 16 09:46 test1.cfg
[root@localhost ~]# chmod 777 test1.cfg
[root@localhost ~]# ll test1.cfg
-rwxrwxrwx. 1 root root 899 4月 16 09:46 test1.cfg
```

(2) 变更属性的指令 `chmod` 的文字用法如下:

```
[root@localhost ~]# cp anaconda-ks.cfg test2.cfg
[root@localhost ~]# ll test2.cfg
-rw----- . 1 root root 899 4月 16 09:46 test2.cfg
[root@localhost ~]# chmod u+x,g+r+w+x,o+r+w+x test2.cfg
[root@localhost ~]# ll test2.cfg
-rwxrwxrwx. 1 root root 899 4月 16 09:46 test2.cfg
```

## 2. chown 命令

文件与目录不仅可以改变权限,其所有权及所属用户组也能修改,与设置权限类似,用户可以通过图形界面来设置或者执行 chown 命令来修改。

chown 用法的格式如下。

```
chown 文件所有者[:组群]文件名
```

chown 的功能可以改变文件的所有者,并且如果有[:组群]可以一并修改文件的所属组群。如果在命令中加-R,就可以把所要修改的文件下的所有文件的群组都修改。

```
[root@localhost ~]#ls -l bbb.txt
-rw-r--r-- 1 root root 0  2月  4 04:31 bbb.txt
[root@localhost ~]#chown test:test bbb.txt
[root@localhost ~]#ls -l bbb.txt
-rw-r--r-- 1 test test 0  2月  4 04:31 bbb.txt
/* */
```



请注意上面 chmod 命令用到的普通用户 test 需要是系统真实存在的用户,否则命令会报错“无效的用户”。具体用户创建方法如图 3.6 所示,在主界面右上角,单击进入账号设置界面,如图 3.7 所示,再单击右上角“添加用户”按钮实现新用户的添加。

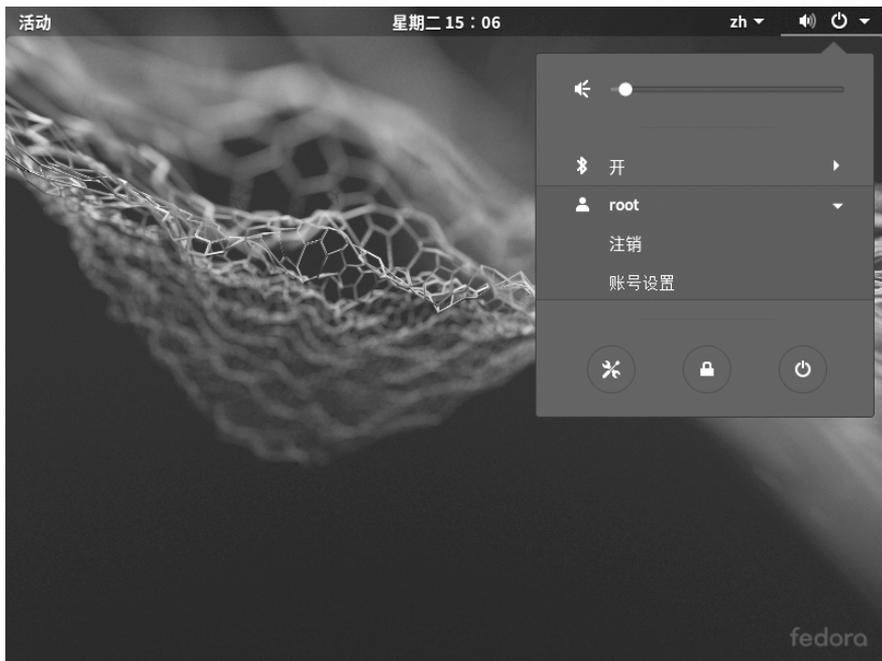


图 3.6 系统账号设置入口界面



图 3.7 系统添加用户界面

### 3. chgrp 命令

chgrp 命令用来改变文件或目录所属的用户群组,其中,组名可以是用户的 ID,也可以是用户组的组名,使用 chgrp 命令的格式如下。

```
chgrp  组群  文件名
```

文件名可以由空格分开的要改变属组的文件列表,也可以是由通配符描述的文件集合。不过要改变的群组名称必须要在 /etc/group 里面存在,否则会显示错误。也可以加 -R 这个参数,此参数的用法和功能与介绍 chmod 时介绍的 -R 是相同的。可以看到该命令的功能可以通过 chown 命令实现,所以该命令一般很少使用。

### 4. umask 命令

我们知道如何创建或者改变一个目录或文件的属性,不过当创建一个新的文件或目录时,它的默认权限是如何指定的,这就与 umask 命令有关了。系统管理员必须要设置一个合理的 umask 值,以确保创建的文件具有所希望的默认权限,防止其他非同组用户对文件具有写权限。在已经登录之后,可以按照个人的偏好使用 umask 命令来改变文件创建的默认权限。相应的改变直到退出该 Shell 或使用另外的 umask 命令之前一直有效。使用 umask 命令的格式如下。

```
umask [选项] [掩码]
```

基本上 umask 就是指定目前用户新建文件或目录时默认的权限默认值。查看权限

默认值有以下两种方法：

(1) 直接输入 `umask`，就可以看到数字形态的权限设置数值，`umask` 有四组数，第一组是特殊权限用的，直接看后三组即可。

```
[root@localhost ~]#umask
0022
```

(2) 加入 `-S` 这个参数，就会以符号类型的方式来显示权限。

```
[root@localhost ~]#umask -S
u=rwx,g=rx,o=rx
```

在默认权限属性上目录与文件是不一样的，执行权限对目录是非常重要的，但是一般文件创建时则不应该有执行的权限，因为一般的文件通常用于数据的记录。用户创建文件默认没有可执行权限，即只有读写权限，也就是最大权限是 666：`-rw-rw-rw-`。若用户创建目录时，由于可执行权限与是否可进入目录有关，因此默认为所有权限均开放，即最大权限是 777：`drwxrwxrwx`。

`umask` 设置了用户创建文件的默认权限，它与 `chmod` 的效果刚好相反，`chmod` 设置的是文件权限码，而 `umask` 设置的是权限“补码”，即 `umask` 是从权限中“拿走”相应的位，即文件创建时不能赋予执行权限。因此，`umask` 的数值指的是该默认权限值需要减掉的权限，当前 `umask` 的值为 022，则当前文件的权限变为文件的最大权限值 (`-rw-rw-rw-`) - (`----w--w-`) = `-rw-r--r--`，目录的权限变为目录的最大权限值 (`drwxrwxrwx`) - (`----w--w-`) = `drwxr-xr-x`。而如果运行命令 `umask 003` 进行权限的修改，003 表示在当前的基础上减去该值，文件的权限变为文件的最大权限值 (`-rw-rw-rw-`) - (`-----wx`) = `-rw-rw-r--`，目录的权限变为目录的最大权限值 (`drwxrwxrwx`) - (`-----wx`) = `drwxrwxr--`。

```
[root@localhost ~]#umask
0022
[root@localhost ~]#mkdir -p test
[root@localhost ~]#cd test/
[root@localhost test]#umask -S
u=rwx,g=rx,o=rx
[root@localhost test]#mkdir linux
[root@localhost test]#ll
总用量 4
drwxr-xr-x. 2 root root 4096 4月 16 10:46 linux

//新建的子目录 linux 的权限确为 drwxr-xr-x

[root@localhost test]#umask 003
[root@localhost test]#mkdir linux2
[root@localhost test]#ll
```

```
总用量 8
drwxr-xr-x. 2 root root 4096 4月 16 10:46 linux
drwxrwxr--. 2 root root 4096 4月 16 13:10 linux2
//请通过这两个目录的权限的比较理解、掌握 umask 的使用
```

### 3.3.4 Linux 文件的时间属性

Linux 系统的文件除了对应的权限外还有三个主要的时间属性,分别是 ctime (change time)、atime(access time)和 mtime(modify time)。文件的 ctime 是在写入文件、更改所有者、权限或链接设置时随 inode 的内容更改而更改的,其实它记录该文件的 inode 节点被修改的时间,chmod 和 chown 等命令也能改变该值;文件的 atime 是在读取文件或者执行文件时更改的;文件的 mtime 是在写入文件时随文件内容的更改而更改的。可以理解为如下。

- (1) ctime: 最后一次改变文件或目录(改变的是原数据即属性)的时间。
- (2) atime: 最后一次访问文件或目录的时间。
- (3) mtime: 最后一次修改文件或目录的时间。

使用 stat 可以查看文件的 ctime、atime 和 mtime。

```
[root@localhost ~]# cd
[root@localhost ~]# stat anaconda-ks.cfg
文件:anaconda-ks.cfg
大小:899          块:8          IO 块:4096   普通文件
设备:fd00h/64768d  Inode:786627   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:2019-04-16 09:46:34.082816373 +0800
最近更改:2019-04-05 01:00:09.325027749 +0800
最近改动:2019-04-05 01:00:09.325027749 +0800
创建时间:-
```

也可以使用 ls 查看文件的 ctime、atime 和 mtime。

ls -lc filename 列出文件的 ctime。

```
[root@localhost ~]# ls -lc anaconda-ks.cfg
-rw-----. 1 root root 899 4月 5 01:00 anaconda-ks.cfg
```

ls -lu filename 列出文件的 atime。

```
[root@localhost ~]# ls -lu anaconda-ks.cfg
-rw-----. 1 root root 899 4月 16 09:46 anaconda-ks.cfg
```

ls -l filename 列出文件的 mtime。

```
[root@localhost ~]#ls -l anaconda-ks.cfg
-rw-----. 1 root root 899 4月 5 01:00 anaconda-ks.cfg
```

## 3.4 Linux 文件的查阅与创建

### 3.4.1 文本文件查阅命令 cat、more、less、head、tail

用户要查看一个文本文件的内容时,可以根据显示要求的不同选用以下命令。

#### 1. cat 命令

该命令的主要功能是用来显示文件,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。另外,还能够用来连接两个或多个文件形成新的文件,该命令的常用格式如下。

```
cat [选项]文件名
```

该命令的[选项]含义如下。

-n: 由 1 开始对所有输出的行数编号。

-b: 和-n 相似,只不过对于空白行不编号。

-s: 当遇到有连续两行以上的空白行时,就代换为一行的空白行。

其实 cat 主要有以下三大功能。

(1) 一次显示整个文件的内容。

```
[root@localhost ~]#cat .bashrc
#.bashrc
#User specific aliases and functions
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
#Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
PATH=/usr/local/arm/4.3.1-eabi-armv6/usr/bin/:$PATH:$HOME/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/arm/4.3.1-eabi-armv6/gmp/lib:/usr/local/arm/4.3.1-eabi-armv6/mpfr/lib
export PATH
export LD_LIBRARY_PATH
```



(2) 从键盘创建一个文件: # cat > filename(注意: 只能创建新文件,不能编辑已