

## 项目5

# Linux配置与管理

### 学习目标

- 理解 Linux 的发展历史、Linux 的特性、Linux 基本命令、Vi、Vim 编辑器的使用、磁盘配置与管理、逻辑卷配置与管理、RAID 基础知识以及网络配置管理等相关理论知识。
- 掌握 RAID 配置基本命令、RAID5 阵列实例配置以及磁盘扩容等相关知识与技能。

## 5.1 项目陈述

回顾 Linux 的历史,可以说它是“踩着巨人的肩膀”逐步发展起来的, Linux 在很大程度上借鉴了 UNIX 操作系统的成功经验,继承并发展了 UNIX 的优良传统。由于 Linux 具有开源的特性,因此一经推出便得到了广大操作系统开发爱好者的积极响应和支持,这也是 Linux 得以迅速发展的关键因素之一。本章讲解 Linux 的发展历史、Linux 的特性、Linux 基本命令以及 Vi、Vim 编辑器的使用等相关理论知识,项目实践部分讲解 RAID 配置基本命令、RAID5 阵列实例配置以及磁盘扩容配置等相关知识与技能。

## 5.2 必备知识



### 5.2.1 Linux 的发展历史

Linux 操作系统是一种类 UNIX 的操作系统,UNIX 是一种主流经典的操作系统, Linux 操作系统来源于 UNIX,是 UNIX 在计算机上的完整实现。UNIX 操作系统是 1969 年由肯·汤普森(K. Thompson)在美国贝尔实验室开发的一种操作系统,1972 年,其与丹尼斯·里奇(D. Ritchie)一起用 C 语言重写了 UNIX 操作系统,大幅增加了其可移植性。由于 UNIX 具有良好而稳定的性能,又在几十年中不断地改进和迅速发展,因此在计算机领域中得到了广泛应用。

由于美国电话电报公司的政策改变,在 Version 7 UNIX 推出之后,其发布了新的使用条款,

将 UNIX 源代码私有化,在大学中不能再使用 UNIX 源代码。1987 年,荷兰的阿姆斯特丹自由大学计算机科学系的安德鲁·塔能鲍姆(A. Tanenbaum)教授为了能在课堂上教授学生操作系统运作的实务细节,决定在不使用任何美国电话电报公司的源代码的前提下,自行开发与 UNIX 兼容的操作系统,以避免版权上的争议。他以小型 UNIX(mini-UNIX)之意将此操作系统命名为 MINIX。MINIX 是一种基于微内核架构的类 UNIX 计算机操作系统,除了启动的部分用汇编语言编写以外,其他大部分是用 C 语言编写的,其内核系统分为内核、内存管理及文件管理 3 部分。

MINIX 最有名的学生用户是芬兰人李纳斯·托沃兹(L. Torvalds),他在芬兰的赫尔辛基技术大学用 MINIX 操作系统搭建了一个新的内核与 MINIX 兼容的操作系统。1991 年 10 月 5 日,他在一台 FTP 服务器上发布了这个消息,将此操作系统命名为 Linux,标志着 Linux 操作系统的诞生。在设计理念上,Linux 和 MINIX 大相径庭,MINIX 在内核设计上采用了微内核的原则,但 Linux 和原始的 UNIX 相同,都采用了宏内核的设计。

Linux 操作系统增加了很多功能,被完善并发布到互联网中,所有人都可以免费下载、使用它的源代码。Linux 的早期版本并没有考虑用户的使用,只提供了最核心的框架,使得 Linux 编程人员可以享受编制内核的乐趣,这也促成了 Linux 操作系统内核的强大与稳定。随着互联网的发展与兴起,Linux 操作系统迅速发展,许多优秀的程序员都加入了 Linux 操作系统的编写行列之中,随着编程人员的扩充和完整的操作系统基本软件的出现,Linux 操作系统开发人员认识到 Linux 已经逐渐变成一个成熟的操作系统平台,1994 年 3 月,其内核 5.0 的推出,标志着 Linux 第一个版本的诞生。

Linux 一开始要求所有的源码必须公开,且任何人均不得从 Linux 交易中获利。然而,这种纯粹的自由软件的理想对于 Linux 的普及和发展是不利的,于是 Linux 开始转向通用公共许可证(General Public License,GPL)项目,成为 GNU(GNU's Not UNIX)阵营中的主要一员。GNU 项目是由理查德·斯托曼(R. Stallman)于 1984 年提出的,他建立了自由软件基金会,并提出 GNU 项目的目的是开发一种完全自由的、与 UNIX 类似但功能更强大的操作系统,以便为所有计算机用户提供一种功能齐全、性能良好的基本系统。

Linux 凭借优秀的设计、不凡的性能,加上 IBM、Intel、CA、Core、Oracle 等国际知名企业的大力支持,市场份额逐步扩大,逐渐成为主流操作系统之一。

## 5.2.2 Linux 的特性

Linux 操作系统是目前发展最快的操作系统,这与 Linux 具有的良好特性是分不开的。它包含 UNIX 的全部功能和特性。Linux 操作系统作为一款免费、自由、开放的操作系统,发展势不可挡,它高效、安全、稳定,支持多种硬件平台,用户界面友好,网络功能强大,支持多任务、多用户。

(1) 开放性。Linux 操作系统遵循世界标准规范,特别是遵循开放系统互连(Open System Interconnect,OSI)国际标准,凡遵循国际标准所开发的硬件和软件都能彼此兼容,可方便地实现互连。另外,源代码开放的 Linux 是免费的,使 Linux 的获得非常方便,且使用 Linux 可节省花销。使用者能控制源代码,即按照需求对部件进行配置,以及自定义建设系统安全设置等。

(2) 多用户。Linux 操作系统资源可以被不同用户使用,每个用户对自己的资源(如文件、设备)有特定的权限,互不影响。

(3) 多任务。使用 Linux 操作系统的计算机可同时执行多个程序,而各个程序的运行互相独立。



(4) 良好的用户界面。Linux 操作系统为用户提供了图形用户界面。它利用鼠标、菜单、窗口、滚动条等元素,给用户呈现一个直观、易操作、交互性强的友好的图形化界面。

(5) 设备独立性强。Linux 操作系统将所有外部设备统一当作文件来看待,只要安装它们的驱动程序,任何用户都可以像使用文件一样操纵、使用这些设备,而不必知道它们的具体存在形式。Linux 是具有设备独立性的操作系统,它的内核具有高度适应能力。

(6) 提供了丰富的网络功能。Linux 操作系统是在 Internet 基础上产生并发展起来的,因此,完善的内置网络是 Linux 的一大特点,Linux 操作系统支持 Internet、文件传输和远程访问等。

(7) 可靠的安全系统。Linux 操作系统采取了许多安全技术措施,包括读写控制、带保护的子系统、审计跟踪、核心授权等,这为网络多用户环境中的用户提供了必要的安全保障。

(8) 良好的可移植性。Linux 操作系统从一个平台转移到另一个平台时仍然能用其自身的方式运行。Linux 是一种可移植的操作系统,能够在从微型计算机到大型计算机的任何环境和任何平台上运行。

(9) 支持多文件系统。Linux 操作系统可以把许多不同的文件系统以挂载形式连接到本地主机上,包括 Ext2/3、FAT32、NTFS、OS/2 等文件系统,以及网络中其他计算机共享的文件系统等,是数据备份、同步、复制的良好平台。

### 5.2.3 Linux 基本命令

Linux 操作系统的 Shell 作为操作系统的外壳,为用户提供使用操作系统的接口。它是命令语言、命令解释程序及程序设计语言的统称。

Shell 是用户和 Linux 内核之间的接口程序,如果把 Linux 内核想象成一个球体的中心,Shell 就是围绕内核的外层。当从 Shell 或其他程序向 Linux 传递命令时,内核会做出相应的反应。

#### 1. Shell 命令的基本格式

在 Linux 操作系统中看到的命令其实就是 Shell 命令,Shell 命令的基本格式如下。

```
command [选项] [参数]
```

(1) command 为命令名称,例如,查看当前文件夹下文件或文件夹的命令是 ls。

(2) [选项]表示可选,是对命令的特别定义,以连接符“-”开始,多个选项可以用一个连接符“-”连接起来,例如,ls -l -a 与 ls -la 的作用是相同的,有些命令不写选项和参数也能执行,有些命令在必要时可以附带选项和参数。

ls 是常用的一个命令,属于目录操作命令,用来列出当前目录下的文件和文件夹。ls 命令后可以加选项,也可以不加选项,不加选项的写法如下。

```
[root@localhost ~]# ls
anaconda - ks.cfg initial - setup - ks.cfg 公共 模板 视频 图片 文档 下载 音乐 桌面
[root@localhost ~]#
```

ls 命令之后不加选项和参数也能执行,但只能执行最基本的功能,即显示当前目录下的文件名。那么,加入一个选项后,会出现什么结果?

```
[root@localhost ~]# ls -l
总用量 8
```

```

-rw----- . 1 root root 1647 6月 8 01:27 anaconda-ks.cfg
-rw-r--r-- . 1 root root 1695 6月 8 01:30 initial-setup-ks.cfg
drwxr-xr-x. 2 root root 6 6月 8 01:41 公共
... (省略部分内容)
drwxr-xr-x. 2 root root 40 6月 8 01:41 桌面
[root@localhost ~]#

```

如果加-l选项,则可以看到显示的内容明显增多了。-l是长格式(long list)的意思,即显示文件的详细信息。

可以看到,选项的作用是调整命令功能。如果没有选项,那么命令只能执行最基本的功能;而一旦有选项,就能执行更多功能,或者显示更加丰富的数据。

Linux的选项又分为短格式选项和长格式选项两类。

短格式选项是长格式选项的简写,用一个“-”和一个字母表示,如“ls -l”。

长格式选项是完整的英文单词,用两个“-”和一个单词表示,如“ls --all”。

一般情况下,短格式选项是长格式选项的缩写,即一个短格式选项会有对应的长格式选项。当然也有例外,例如,“ls”命令的短格式选项“-l”就没有对应的长格式选项,所以具体的命令选项需要通过帮助手册来查询。

(3) [参数]为跟在可选项后的参数,或者是command的参数,参数可以是文件,也可以是目录,可以没有,也可以有多个,有些命令必须使用多个操作参数,例如,cp命令必须指定源操作对象和目标对象。

(4) command [选项] [参数]等项目之间以空格隔开,无论有几个空格,Shell都将其视为一个空格。

## 2. 输入命令时键盘操作的一般规律

- (1) 命令、文件名、参数等都要区分英文大小写,例如,md与MD是不同的。
- (2) 命令、选项、参数之间必须有一个或多个空格。
- (3) 命令太长时,可以使用“\”符号来转义Enter符号,以实现一条命令跨多行。

```

[root@localhost ~]# hostnamectl set - hostname \           //输入"\”符号来转义Enter符号
> test1                                                  //输入主机名为test1
[root@localhost ~]# bash                                  //bash执行命令
[root@test1 ~]#

```

(4) 按Enter键以后,该命令才会被执行。

## 3. 配置显示系统的常用命令

(1) cat命令用于查看Linux内核版本,执行命令如下。

```

[root@localhost ~]# cat /proc/version
Linux version 3.10.0-957.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 5.8.5
20150623 (Red Hat 5.8.5-36) (GCC) ) #1 SMP Thu Nov 8 23:39:32 UTC 2018
[root@localhost ~]#
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 7.6.1810 (Core)
[root@localhost ~]#

```

cat 命令的作用是连接文件或标准输入并输出。这个命令常用来显示文件内容,或者将几个文件连接起来显示,或者从标准输入读取内容并显示,常与重定向符号配合使用。其命令格式如下。

```
cat [选项] 文件名
```

cat 命令各选项及其功能说明,如表 5.1 所示。

表 5.1 cat 命令各选项及其功能说明

选 项	功 能 说 明
-A	等价于-vET
-b	对非空输出行编号
-e	等价于-vE
-E	在每行结束处显示 \$
-n	对输出的所有行编号,由 1 开始对所有输出的行数进行编号
-s	当有连续两行以上的空白行时,将其替换为一行的空白行
-t	与-vT 等价
-T	将跳格字符显示为^I
-v	使用^和 M-引用,除了 Tab 键之外

使用 cat 命令来显示文件内容时,执行命令如下。

```
[root@localhost ~]# dir
a1 - test05.txt  history.txt  mkfs.ext2  mkrfc2734  公共  图片  音乐
anaconda - ks.cfg  initial - setup - ks.cfg  mkfs.msdos  mnt  模板  文档  桌面
font.map  mkfontdir  mkinitrd  user01  视频  下载
[root@localhost ~]# cat a1 - test05.txt //显示 a1 - test05.txt 文件的内容
aaaaaaaaaaaaaa
bbbbbbbbbbbbbbb
ccccccccccccccc
[root@localhost ~]# cat - nE a1 - test05.txt //显示 a1 - test05.txt 文件的内容,对输出的
//所有行编号,由 1 开始对所有输出的行数进行编号,在每行结束处显示 $
 1 aaaaaaaaaaaaaa $
 2 bbbbbbbbbbbbbbb $
 3 ccccccccccccccc $
[root@localhost ~]#
```

(2) tac 命令反向显示文件内容。

tac 命令与 cat 命令相反,只适合用于显示内容较少的文件。其命令格式如下。

```
tac [选项] 文件名
```

tac 命令各选项及其功能说明,如表 5.2 所示。

表 5.2 tac 命令各选项及其功能说明

选 项	功 能 说 明
-b	在行前而非行尾添加分隔标志
-r	分隔标志视作正则表达式来解析
-s	使用指定字符串代替换行作为分隔标志

使用 tac 命令来反向显示文件内容时,执行命令如下。

```
[root@localhost ~]# tac -r a1 - test05.txt
cccccccccccccccc
bbbbbbbbbbbbbbbb
aaaaaaaaaaaaaaaa
[root@localhost ~]#
```

(3) head 命令查看文件的 n 行。

head 命令用于查看具体文件的前几行的内容,默认情况下显示文件前 10 行的内容。其命令格式如下。

```
head [选项] 文件名
```

head 命令各选项及其功能说明,如表 5.3 所示。

表 5.3 head 命令各选项及其功能说明

选 项	功 能 说 明
-c	显示文件的前 n 个字节,如-c5,表示文件内容的前 5 个字节
-n	后面接数字,表示显示几行
-q	不显示包含给定文件名的文件头
-v	总是显示包含给定文件名的文件头

使用 head 命令来查看具体文件的前几行的内容时,执行以下命令。

```
[root@localhost ~]# head -n5 -v /etc/passwd
==> /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
[root@localhost ~]#
```

(4) tail 命令查看文件的最后 n 行。

tail 命令用于查看具体文件的最后几行的内容,默认情况下显示文件最后 10 行的内容,可以使用 tail 命令来查看日志文件被更改的过程。其命令格式如下。

```
tail [选项] 文件名
```

tail 命令各选项及其功能说明,如表 5.4 所示。

表 5.4 tail 命令各选项及其功能说明

选 项	功 能 说 明
-c	显示文件的前 n 个字节,如-c5,表示文件内容前 5 个字节,其他文件内容不显示
-f	随着文件的生长输出附加数据,即实时跟踪文件,显示一直继续,直到按 Ctrl+C 组合键才停止显示
-F	实时跟踪文件,如果文件不存在,则继续尝试

续表

选 项	功 能 说 明
-n	后面接数字时,表示显示几行
-q	不显示包含给定文件名的文件头
-v	总是显示包含给定文件名的文件头

使用 tail 命令来查看具体文件的最后几行的内容时,执行以下命令。

```
[root@localhost ~]# tail -n5 -v /etc/passwd
==> /etc/passwd <==
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
csg:x:1000:1000:root:/home/csg:/bin/bash
user01:x:1001:1001:user01:/home/user01:/bin/bash
user0:x:1002:1002:user01:/home/user0:/bin/bash
[root@localhost ~]#
```

(5) echo 命令将显示内容输出到屏幕上。

echo 命令非常简单,如果命令的输出内容没有特殊含义,则原内容输出到屏幕上;如果命令的输出内容有特殊含义,则输出其含义。其命令格式如下。

```
echo [选项] [输出内容]
```

echo 命令各选项及其功能说明,如表 5.5 所示。

表 5.5 echo 命令各选项及其功能说明

选 项	功 能 说 明
-n	取消输出后行末的换行符号(内容输出后不换行)
-e	支持反斜线控制的字符转换

在 echo 命令中,如果使用了-n 选项,则表示输出文字后不换行;字符串可以加引号,也可以不加引号,用 echo 命令输出加引号的字符串时,将字符串原样输出;用 echo 命令输出不加引号的字符串时,字符串中的各个单词作为字符串输出,各字符串之间用一个空格分隔。

如果使用了-e 选项,则可以支持控制字符,会对其进行特别处理,而不会将它当作一般文字输出。控制字符如表 5.6 所示。

表 5.6 控制字符

控 制 字 符	功 能 说 明
\\	输出\本身
\a	输出警告音
\b	退格键,即向左删除键
\c	取消输出行末的换行符。和-n 选项一致
\e	Esc 键
\f	换页符
\n	换行符
\r	回车键
\t	制表符,即 Tab 键

续表

控制字符	功能说明
\v	垂直制表符
\0nnn	按照八进制 ASCII 表输出字符。其中,0 为数字 0,nnn 是三位八进制数
\xhh	按照十六进制 ASCII 表输出字符。其中,hh 是两位十六进制数

使用 echo 命令输出相关内容到屏幕上,执行以下命令。

```
[root@localhost ~]# echo -en "hello welcome\n"           //换行输出
hello welcome
[root@localhost ~]# echo -en "1 2 3\n"                   //整行换行输出
1 2 3
[root@localhost ~]# echo -en "1\n2\n3\n"                //每个字符换行输出
1
2
3
[root@localhost ~]# echo -n aaa                           //字符串不加引号,不换行输出
aaa[root@localhost ~]# echo -n 123
123[root@localhost ~]#
```

echo 命令也可以把显示输出的内容输入到一个文件中,命令如下。

```
[root@localhost ~]# echo "hello everyone welcome to here" > welcome.txt //写入或替换
[root@localhost ~]# echo "hello everyone" >> welcome.txt //追加写入
[root@localhost ~]# cat welcome.txt
hello everyone welcome to here
hello everyone
[root@localhost ~]#
```

(6) shutdown 命令可以安全地关闭或重启 Linux 操作系统,它在系统关闭之前给系统中的所有登录用户发送一条警告信息。

该命令还允许用户指定一个时间参数,用于指定什么时间关闭,时间参数可以是一个精确的时间,也可以是从现在开始的一个时间段。

精确时间的格式是 hh:mm,表示小时和分钟,时间段由小时和分钟数表示。系统执行该命令后会自动进行数据同步的工作。

该命令的一般格式如下。

```
shutdown [选项] [时间] [警告信息]
```

shutdown 命令各选项及含义,如表 5.7 所示。

表 5.7 shutdown 命令各选项及含义

选项	含义
-k	并不真正关机,而只是发出警告信息给所有用户
-r	关机后立即重新启动系统
-h	关机后不重新启动系统
-f	快速关机重新启动时跳过文件系统检查
-n	快速关机且不经过 init 程序
-c	取消一个已经运行的 shutdown 操作

需要特别说明的是,该命令只能由 root 用户使用。

halt 是最简单的关机命令,其实际上是调用 shutdown -h 命令。halt 命令执行时,会结束应用程序,文件系统写操作完成后会停止内核。

```
[root@localhost ~]# shutdown -h now           //立刻关闭系统
```

reboot 的工作过程与 halt 类似,其作用是重新启动系统,而 halt 是关机。其参数也与 halt 类似, reboot 命令重启系统时是删除所有进程,而不是平稳地终止它们。因此,使用 reboot 命令可以快速地关闭系统,但当还有其他用户在该系统中工作时,会引起数据的丢失,所以使用 reboot 命令的场合主要是单用户模式。

```
[root@localhost ~]# reboot                   //立刻重启系统
[root@localhost ~]# shutdown -r 00:05       //5 min 后重启系统
[root@localhost ~]# shutdown -c            //取消 shutdown 操作
```

退出终端窗口命令 exit。

```
[root@localhost ~]# exit                     //退出终端窗口
```

(7) whoami 命令用于显示当前的操作用户的用户名,执行命令如下。

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

(8) hostnamectl 命令用于设置当前系统的主机名,执行命令如下。

```
[root@localhost ~]# hostnamectl set -hostname test1 //设置当前系统的主机名为 test1
[root@localhost ~]# bash                             //执行命令
[root@test1 ~]#
[root@test1 ~]# hostname                             //查看当前主机名
test1
[root@test1 ~]#
```

(9) date 命令用于显示当前时间/日期,执行命令如下。

```
[root@localhost ~]# date
2022年05月10日 星期二 19:13:05 CST
[root@localhost ~]#
```

(10) cal 命令用于显示日历信息,执行命令如下。

```
[root@localhost ~]# cal
      五月 2022
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
[root@localhost ~]#
```

(11) clear 命令相当于 DOS 下的 cls 命令,执行命令如下。

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

(12) history 命令可以用来显示和编辑历史命令,显示最近 5 个历史命令,执行命令如下。

```
[root@localhost ~]# history 5
 14 uname -a
 15 whoami
 16 date
 17 cal
 18 history 5
[root@localhost ~]#
```

(13) pwd 命令显示当前工作目录,执行命令如下。

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



(14) cd 命令改变当前工作目录。

cd 是 change directory 的缩写,用于改变当前工作目录。其命令格式如下。

```
cd [绝对路径或相对路径]
```

路径是目录或文件在系统中的存放位置,如果想要编辑 ifcfg-ens33 文件,则先要知道此文件的所在位置,此时就需要用路径来表示。

路径是由目录和文件名构成的,例如,/etc 是一个路径,/etc/sysconfig 是一个路径,/etc/sysconfig/network-scripts/ifcfg-ens33 也是一个路径。

路径的分类如下。

① 绝对路径:从根目录(/)开始的路径,如/usr、usr/local/、usr/local/etc 等是绝对路径,它指向系统中一个绝对的位置。

② 相对路径:路径不是由“/”开始的,相对路径的起点为当前目录,如果现在位于/usr 目录,那么相对路径 local/etc 所指示的位置为/usr/local/etc。也就是说,相对路径所指示的位置,除了相对路径本身之外,还受到当前位置的影响。

Linux 操作系统中常见的目录有/bin、usr/bin、usr/local/bin,如果只有一个相对路径 bin,那么它指示的位置可能是上面 3 个目录中的任意一个,也可能是其他目录。特殊符号表示的目录,如表 5.8 所示。

表 5.8 特殊符号表示的目录

特殊符号	表示的目录	特殊符号	表示的目录
~	代表当前登录用户的主目录	.	代表当前目录
~用户名	表示切换至指定用户的主目录	..	代表上级目录
-	代表上次所在目录		

如果只输入 cd,未指定目标目录名,则返回到当前用户的主目录,等同于 cd~,一般用户的主

目录默认在/root下,如root用户的默认主目录为/root,为了能够进入指定的目录,用户必须拥有对指定目录的执行和读权限。

以root身份登录到系统中,进行目录切换等操作,执行命令如下。

```
[root@localhost ~]# pwd //显示当前工作目录
/root
[root@localhost ~]# cd /etc //以绝对路径进入 etc 目录
[root@localhost etc]# cd yum.repos.d //以相对路径进入 yum.repos.d 目录
[root@localhost yum.repos.d]# pwd
/etc/yum.repos.d
[root@localhost yum.repos.d]# cd . //当前目录
[root@localhost yum.repos.d]# cd .. //上级目录
[root@localhost etc]# pwd
/etc
[root@localhost etc]# cd ~ //当前登录用户的主目录
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd - //上次所在目录
/etc
[root@localhost etc]#
```

(15) ls 命令显示目录文件。

ls 是 list 的缩写,不加参数时,ls 命令用来显示当前目录清单,是 Linux 中最常用的命令之一,通过 ls 命令不仅可以查看 Linux 文件夹包含的文件,还可以查看文件及目录的权限、目录信息等。其命令格式如下。

```
ls [选项] 目录或文件名
```

ls 命令各选项及其功能说明,如表 5.9 所示。

表 5.9 ls 命令各选项及其功能说明

选 项	功 能 说 明
-a	显示所有文件,包括隐藏文件,如“.”“..”
-d	仅可以查看目录的属性参数及信息
-h	以易于阅读的格式显示文件或目录的大小
-i	查看任意一个文件的节点
-l	长格式输出,包含文件属性,显示详细信息
-L	递归显示,即列出某个目录及子目录的所有文件和目录
-t	以文件和目录的更改时间排序显示

使用 ls 命令,进行显示目录文件相关操作,执行命令如下。

① 显示所有文件,包括隐藏文件,如“.”“..”。

```
[root@localhost ~]# ls -a
.          .bash_profile  .esd_auth      mkfontdir     .tcshrc      文档
..         .bashrc        font.map        mkfs.ext2     .Viminfo     下载
aa.txt     .cache         history.txt     mkfs.msdos    公共         音乐
anaconda-ks.cfg .config       .ICEauthority  mkinitrd      模板         桌面
... (省略部分内容)
[root@localhost ~]#
```

② 长格式输出,包含文件属性,显示详细信息。

```
[root@localhost ~]# ls -l
总用量 16
-rw-r--r--. 1 root root 85 6月 25 14:04 aa.txt
-rw-----. 1 root root 1647 6月 8 01:27 anaconda-ks.cfg
-rw-r--r--. 1 root root 0 6月 20 22:37 font.map
...(省略部分内容)
[root@localhost ~]#
```

(16) touch 命令创建文件或修改文件。

touch 命令可以用来创建文件或修改文件的存取时间,如果指定的文件不存在,则会生成一个空文件。其命令格式如下。



```
touch [选项] 目录或文件名
```

touch 命令各选项及其功能说明,如表 5.10 所示。

表 5.10 touch 命令各选项及其功能说明

选 项	功 能 说 明
-a	只把文件存取时间修改为当前时间
-d	把文件的存取/修改时间格式改为 yyyymmdd
-m	只把文件的修改时间修改为当前时间

使用 touch 命令创建一个或多个文件时,执行以下命令。

```
[root@localhost ~]# cd /mnt //切换目录
[root@localhost mnt]# touch file05.txt //创建一个文件
[root@localhost mnt]# touch file0{5..4}.txt //创建多个文件
[root@localhost mnt]# touch * //把当前目录下所有文件的存取和修改时间修改为当前时间
[root@localhost mnt]# ls -l //查看修改结果
总用量 0
-rw-r--r--. 1 root root 0 5月 10 19:35 file05.txt
-rw-r--r--. 1 root root 0 5月 10 19:35 file05.txt
-rw-r--r--. 1 root root 0 5月 10 19:35 file03.txt
-rw-r--r--. 1 root root 0 5月 10 19:35 file05.txt
[root@localhost mnt]#
```

使用 touch 命令把目录/mnt下的所有文件的存取和修改时间修改为 2022 年 6 月 26 日,执行以下命令。

```
[root@localhost mnt]# touch -d 20220626 /mnt/*
[root@localhost mnt]# ls -l
总用量 0
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file03.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
[root@localhost mnt]#
```

(17) mkdir 命令创建新目录。

mkdir 命令用于创建指定的目录名,要求创建的用户在当前目录中具有写权限,并且指定的目录名不能是当前目录中已有的目录,目录可以是绝对路径,也可以是相对路径。其命令格式如下。

```
mkdir [选项] 目录名
```

mkdir 命令各选项及其功能说明,如表 5.11 所示。

表 5.11 mkdir 命令各选项及其功能说明

选 项	功 能 说 明
-p	创建目录时,递归创建,如果父目录不存在,则此时可以与子目录一起创建,即可以一次创建多个层次的目录
-m	给创建的目录设定权限,默认权限是 drwxr-xr-x
-v	输入目录创建的详细信息

使用 mkdir 命令创建新目录时,执行命令如下。

```
[root@localhost mnt]# mkdir user01 //创建新目录 user01
[root@localhost mnt]# ls -l
总用量 0
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file03.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
drwxr-xr-x. 2 root root 6 5月 10 19:38 user01
[root@localhost mnt]# mkdir -v user02 //创建新目录 user02
mkdir: 已创建目录 "user02"
[root@localhost mnt]# ls -l
总用量 0
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file03.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file05.txt
drwxr-xr-x. 2 root root 6 5月 10 19:38 user01
drwxr-xr-x. 2 root root 6 5月 10 19:40 user02
[root@localhost mnt]# mkdir -p /mnt/user03/a01 /mnt/user03/a02
//在 user03 目录下,同时创建目录 a01 和目录 a02
[root@localhost mnt]# ls -l /mnt/user03
总用量 0
drwxr-xr-x. 2 root root 6 5月 10 19:41 a01
drwxr-xr-x. 2 root root 6 5月 10 19:41 a02
[root@localhost mnt]#
```

(18) rmdir 命令删除目录。

rmdir 是常用的命令,该命令的功能是删除空目录,一个目录被删除之前必须是空的,删除某目录时必须具有对其父目录的写权限。其命令格式如下。

```
rmdir [选项] 目录名
```

rmmdir 命令各选项及其功能说明,如表 5.12 所示。

表 5.12 rmdir 命令各选项及其功能说明

选 项	功 能 说 明
-p	递归删除目录,当子目录删除后其父目录为空时,父目录也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径,则系统在标准输出上显示相应的信息
-v	显示指令执行过程

使用 rmdir 命令删除目录时,执行命令如下。

```
[root@localhost mnt]# rmdir -v /mnt/user03/a01
rmdir: 正在删除目录 "/mnt/user03/a01"
[root@localhost mnt]# ls -l /mnt/user03
总用量 0
drwxr-xr-x. 2 root root 6 5月 10 19:41 a02
[root@localhost mnt]#
```

(19) rm 命令删除文件或目录。

rm 既可以删除一个目录中的一个文件或多个文件或目录,又可以将某个目录及其下的所有文件及子目录都删除,功能非常强大。其命令格式如下。

```
rm [选项] 目录或文件名
```

rm 命令各选项及其功能说明,如表 5.13 所示。

表 5.13 rm 命令各选项及其功能说明

选 项	功 能 说 明
-f	强制删除,删除文件或目录时不提示用户
-i	在删除前会询问用户是否操作
-r	删除某个目录及其中的所有的文件和子目录
-d	删除空文件或目录
-v	显示指令执行过程

使用 rm 命令删除文件或目录时,执行命令如下。

```
[root@localhost ~]# ls -l /mnt //显示目录下的信息
总用量 0
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file03.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file05.txt
drwxr-xr-x. 2 root root 6 5月 10 19:38 user01
drwxr-xr-x. 2 root root 6 5月 10 19:40 user02
drwxr-xr-x. 3 root root 17 5月 10 19:44 user03
[root@localhost ~]# rm -r -f /mnt/* //强制删除目录下的所有文件和目录
[root@localhost /]# ls -l /mnt //显示目录下的信息
总用量 0
[root@localhost /]#
```

(20) cp 命令复制文件或目录。

要将一个文件或目录复制到另一个文件或目录下,可以使用 cp 命令,该命令的功能非常强大,参数也很多,除了单纯的复制之外,还可以建立连接文件,复制整个目录,在复制的同时可以对文件进行改名操作等,这里仅介绍几个常用的参数选项。其命令格式如下。

```
cp [选项] 源目录或文件名 目标目录或文件名
```

cp 命令各选项及其功能说明,如表 5.14 所示。

表 5.14 cp 命令各选项及其功能说明

选 项	功 能 说 明
-a	将文件的属性一起复制
-f	强制复制,无论目标文件或目录是否已经存在,如果目标文件或目录存在,则先删除它们再复制(即覆盖),并且不提示用户
-i	i 和 f 选项正好相反,如果目标文件或目录存在,则提示是否覆盖已有的文件
-n	不要覆盖已存在的文件(使-i 选项失效)
-p	保持指定的属性,如模式、所有权、时间戳等,与-a 类似,常用于备份
-r	递归复制目录,即包含目录下的各级子目录的所有内容
-s	只创建符号链接而不复制文件
-u	只在源文件比目标文件新或目标文件不存在时才进行复制
-v	显示指令执行过程

使用 cp 命令复制文件或目录时,执行命令如下。

```
[root@localhost ~]# cd /mnt
[root@localhost mnt]# touch a0{5..3}.txt
[root@localhost mnt]# mkdir user0{5..3}
[root@localhost mnt]# dir
a05.txt a05.txt a03.txt user01 user02 user03
[root@localhost mnt]# ls -l
总用量 0
-rw-r--r--. 1 root root 0 5月 10 19:49 a05.txt
-rw-r--r--. 1 root root 0 5月 10 19:49 a05.txt
-rw-r--r--. 1 root root 0 5月 10 19:49 a03.txt
drwxr-xr-x. 2 root root 6 5月 10 19:49 user01
drwxr-xr-x. 2 root root 6 5月 10 19:49 user02
drwxr-xr-x. 2 root root 6 5月 10 19:49 user03
[root@localhost mnt]# cd ~
[root@localhost ~]# cp -r /mnt/a05.txt /mnt/user01/
[root@localhost ~]# ls -l /mnt/user01
总用量 0
-rw-r--r--. 1 root root 0 5月 10 19:51 a05.txt
[root@localhost ~]#
```

(21) mv 命令移动文件或目录。

使用 mv 命令可以为文件或目录重命名或将文件由一个目录移入另一个目录,如果在同一目录下移动文件或目录,则该操作可理解为给文件或目录重命名。其命令格式如下。

```
mv [选项] 源目录或文件名 目标目录或文件名
```

mv 命令各选项及其功能说明,如表 5.15 所示。

表 5.15 mv 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	覆盖前不询问	-n	不覆盖已存在文件
-i	覆盖前询问	-v	显示指令执行过程

使用 mv 命令移动文件或目录时,执行命令如下。

```
[root@localhost ~]# ls -l /mnt //显示/mnt 目录信息
总用量 0
-rw-r--r--. 1 root root 0 6月 25 20:27 a05.txt
-rw-r--r--. 1 root root 0 6月 25 20:27 a05.txt
-rw-r--r--. 1 root root 0 6月 25 20:27 a03.txt
drwxr-xr-x. 2 root root 24 6月 25 20:29 user01
drwxr-xr-x. 2 root root 24 6月 25 20:30 user02
drwxr-xr-x. 6 root root 104 6月 25 20:37 user03
[root@localhost ~]# mv -f /mnt/a05.txt /mnt/test05.txt //将 a05.txt 重命名为 test05.txt
[root@localhost ~]# ls -l /mnt //显示/mnt 目录信息
总用量 0
-rw-r--r--. 1 root root 0 6月 25 20:27 a05.txt
-rw-r--r--. 1 root root 0 6月 25 20:27 a03.txt
-rw-r--r--. 1 root root 0 6月 25 20:27 test05.txt
drwxr-xr-x. 2 root root 24 6月 25 20:29 user01
drwxr-xr-x. 2 root root 24 6月 25 20:30 user02
drwxr-xr-x. 6 root root 104 6月 25 20:37 user03
[root@localhost ~]#
```

(22) tar 命令打包、归档文件或目录。

使用 tar 命令可以把整个目录的内容归并为一个单一的文件,而许多用于 Linux 操作系统的程序就是打包为 TAR 文件的形式,tar 命令是 Linux 中最常用的备份命令之一。

tar 命令可用于建立、还原、查看、管理文件,也可以方便地追加新文件到备份文件中,或仅更新部分备份文件,以及解压、删除指定的文件。这里仅介绍几个常用的参数选项,以便于日常的系统管理工作。其命令格式如下。

```
tar [选项] 文件目录列表
```

tar 命令各选项及其功能说明如表 5.16 所示。

表 5.16 tar 命令各选项及其功能说明

选 项	功 能 说 明
-c	创建一个新归档,如果备份一个目录或一些文件,则使用这个选项
-f	使用归档文件或设备,这个选项通常是必选的,选项后面一定要跟文件名
-z	用 gzip 压缩/解压缩文件,加上该选项后可以对文件进行压缩,还原时也一定要使用该选项进行解压缩
-v	详细地列出处理的文件信息,如无此选项,则 tar 命令不报告文件信息
-r	把要存档的文件追加到档案文件的末尾,使用该选项时,可将忘记的目录或文件追加到备份文件中
-t	列出归档文件的内容,可以查看哪些文件已经备份
-x	从归档文件中释放文件

使用 tar 命令打包、归档文件或目录。

① 将/mnt 目录打包为一个文件 test05.tar,将其压缩为文件 test05.tar.gz,并存放在/root/user01 目录下作为备份,执行以下命令。

```
[root@localhost ~]# rm -rf /mnt/* //删除/mnt 目录下的所有目录或文件
[root@localhost ~]# ls -l /mnt
总用量 0
[root@localhost ~]# touch /mnt/a0{5..2}.txt //新建两个文件
[root@localhost ~]# mkdir /mnt/test0{5..2} //新建两个目录
[root@localhost ~]# ls -l /mnt
总用量 0
-rw-r--r-- . 1 root root 0 6月 25 22:32 a05.txt
-rw-r--r-- . 1 root root 0 6月 25 22:32 a05.txt
drwxr-xr-x. 2 root root 6 6月 25 22:46 test01
drwxr-xr-x. 2 root root 6 6月 25 22:46 test02
[root@localhost ~]# mkdir /root/user01 //新建目录
[root@localhost ~]# tar -cvf /root/user01/test05.tar /mnt
//将/mnt 下的所有文件归并为文件 test05.tar
tar: 从成员名中删除开头的"/"
/mnt/
/mnt/a05.txt
/mnt/a05.txt
/mnt/test01
/mnt/test02
[root@localhost ~]# ls /root/user01
test05.tar
[root@localhost ~]#
```

② 在/root/user01 目录下生成压缩文件 test05.tar,使用 gzip 命令可对单个文件进行压缩,原归档文件 test05.tar 就没有了,并生成压缩文件 test05.tar.gz,执行以下命令。

```
[root@localhost ~]# gzip /root/user01/test05.tar
[root@localhost ~]# ls -l /root/user01
总用量 8
-rw-r--r-- . 1 root root 190 6月 25 22:36 test05.tar.gz
[root@localhost ~]#
```

③ 在/root/user01 目录下生成压缩文件 test05.tar.gz,可以一次性完成归档和压缩,把两步合并为一步,执行以下命令。

```
[root@localhost ~]# tar -zcvf /root/user01/test05.tar.gz /mnt
tar: 从成员名中删除开头的"/"
/mnt/
/mnt/a05.txt
/mnt/a05.txt
/mnt/test01
/mnt/test02
[root@localhost ~]# ls -l /root/user01
总用量 16
-rw-r--r-- . 1 root root 10240 6月 25 22:36 test05.tar.gz
[root@localhost ~]#
```

④ 对文件 test05.tar.gz 进行解压缩,执行以下命令。

```
[root@localhost ~]# cd /root/user01
[root@localhost user01]# ls -l
总用量 4
4 -rw-r--r--. 1 root root 175 6月 25 23:13 test05.tar.gz
[root@localhost user01]# gzip -d test05.tar.gz
[root@localhost user01]# tar -xf test05.tar
```

也可以一次完成解压缩,把两步合并为一步,执行以下命令。

```
[root@localhost user01]# tar -zxf test05.tar.gz
[root@localhost user01]# ls -l
总用量 4
drwxr-xr-x. 4 root root 64 6月 25 23:13 mnt
-rw-r--r--. 1 root root 175 6月 25 23:13 test05.tar.gz
[root@localhost user01]# cd mnt
[root@localhost mnt]# ls -l
总用量 0
-rw-r--r--. 1 root root 0 6月 25 23:12 a05.txt
-rw-r--r--. 1 root root 0 6月 25 23:12 a05.txt
drwxr-xr-x. 2 root root 6 6月 25 23:13 test01
drwxr-xr-x. 2 root root 6 6月 25 23:13 test02
[root@localhost mnt]#
```

可查看用户目录下的文件列表,检查执行的情况,参数 f 之后的文件名是由用户自己定义的,通常应命名为便于识别的名称,并加上相对应的压缩名称,如 xxx.tar.gz。在前面的实例中,如果加上 z 参数,则调用 gzip 进行压缩,通常以 .tar.gz 来代表 gzip 压缩过的 TAR 文件,注意,在压缩时自身不能处于要压缩的目录及子目录内。

(23) whereis 命令查找文件位置。

whereis 命令用于查找可执行文件、源代码文件、帮助文件在文件系统中的位置。其命令格式如下。

```
whereis [选项] 文件
```

whereis 命令各选项及其功能说明,如表 5.17 所示。

表 5.17 whereis 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-b	只搜索二进制文件	-S<目录>	定义源代码查找路径
-B<目录>	定义二进制文件查找路径	-f	终止<目录>参数列表
-m	只搜索 man 手册	-u	搜索不常见记录
-M<目录>	定义 man 手册查找路径	-l	输出有效查找路径
-s	只搜索源代码		

使用 whereis 命令查找文件位置时,执行以下命令。

```
[root@localhost ~]# whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man5/passwd.5.gz /usr/share/man/man1/
```

```
passwd.5.gz
[root@localhost ~]#
```

(24) locate 命令查找绝对路径中包含指定字符串的文件的位置。

locate 命令用于查找可执行文件、源代码文件、帮助文件在文件系统中的位置。其命令格式如下。

```
locate [选项] 文件
```

locate 命令各选项及其功能说明,如表 5.18 所示。

表 5.18 locate 命令各选项及其功能说明

选 项	功 能 说 明
-b	仅匹配路径名的基名
-c	只输出找到的数量
-d	使用 DBPATH 指定的数据库,而不是默认数据库/var/lib/mlocate/mlocate.db
-e	仅输出当前现有文件的条目
-L	当文件存在时,跟随蔓延的符号链接(默认)
-h	显示帮助
-i	忽略字母大小写
-l	LIMIT 限制为 LIMIT 项目的输出(或计数)
-q	安静模式,不会显示任何错误信息
-r	使用基本正则表达式
-w	匹配整个路径名(默认)

使用 locate 命令查找文件位置时,执行命令如下。

```
[root@localhost ~]# locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
/etc/security/opasswd
/usr/bin/gpasswd
...
[root@localhost ~]# locate -c passwd //只输出找到的数量
153
[root@localhost ~]# locate firefox | grep rpm //查找 firefox 文件的位置
/var/cache/yum/x86_64/7/updates/packages/firefox-68.15.0-5.el7.CentOS.x86_65.rpm
```

(25) find 命令文件查找。

find 命令用于文件查找,其功能非常强大,对于文件和目录的一些比较复杂的搜索操作,可以灵活应用最基本的通配符和搜索命令 find 来实现,其可以在某一目录及其所有的子目录中快速搜索具有某些特征的目录或文件。其命令格式如下。

```
find [路径] [匹配表达式] [-exec command]
```

find 命令各匹配表达式及其功能说明,如表 5.19 所示。

表 5.19 find 命令各匹配表达式及其功能说明

匹配表达式	功能说明
-name filename	查找指定名称的文件
-user username	查找属于指定用户的文件
-group groupname	查找属于指定组的文件
-print	显示查找结果
-type	查找指定类型的文件。文件类型有：b(块设备文件)、c(字符设备文件)、d(目录)、p(管道文件)、l(符号链接文件)、f(普通文件)
-mtime n	类似于 atime,但查找的是文件内容被修改的时间
-ctime n	类似于 atime,但查找的是文件索引节点被改变的时间
-newer file	查找比指定文件新的文件,即文件的最后修改时间离现在较近
-perm mode	查找与给定权限匹配的文件,必须以八进制的形式给出访问权限
-exec command {} \;	对匹配指定条件的文件执行 command 命令
-ok command {} \;	与 exec 相同,但执行 command 命令时请用户确认

使用 find 命令查找文件时,执行命令如下。

```
[root@localhost ~]# find /etc - name passwd
/etc/pam.d/passwd
/etc/passwd
[root@localhost ~]# find / - name "firefox * .rpm"
/var/cache/yum/x86_64/7/updates/packages/firefox-68.15.0-5.el7.CentOS.x86_65.rpm
[root@localhost ~]#
[root@localhost ~]# find /etc - type f - exec ls -l {} \;
-rw-r--r--. 1 root root 465 6月 8 01:15 /etc/fstab
-rw-----. 1 root root 0 6月 8 01:15 /etc/crypttab
-rw-r--r--. 1 root root 49 6月 26 09:38 /etc/resolv.conf
... (省略部分内容)
[root@localhost ~]#
```

(26) which 命令确定程序的具体位置。

which 命令用于查找并显示给定命令的绝对路径,环境变量 PATH 中保存了查找命令时需要遍历的目录,which 命令会在环境变量 PATH 设置的目录中查找符合条件的文件,也就是说,使用 which 命令可以看到某个系统指令是否存在,以及执行的命令的位置。其命令格式如下。

```
which [选项] [--] COMMAND
```

which 命令各选项及其功能说明,如表 5.20 所示。

表 5.20 which 命令各选项及其功能说明

选 项	功能说明
--version	输出版本信息
--help	输出帮助信息
--skip-dot	跳过以点开头的路径中的目录
--show-dot	不将点扩展到输出的当前目录中
--show-tilde	输出一个目录的非根
--tty-only	如果不处于 TTY 模式,则停止右侧的处理选项

续表

选 项	功 能 说 明
--all, -a	输出所有的匹配项,但不输出第一个匹配项
--read-alias, -i	从标准输入读取别名列表
--skip-alias	忽略选项--read-alias,不读取标准输入
--read-functions	从标准输入读取 shell 方法
--skip-functions	忽略选项--read-functions

使用 which 命令查找文件位置时,执行命令如下。

```
[root@localhost ~]# which find
/usr/bin/find
[root@localhost ~]# which -- show - tilde pwd
/usr/bin/pwd
[root@localhost ~]# which -- version bash
GNU which v5.20, Copyright (C) 1999 - 2008 Carlo Wood.
GNU which comes with ABSOLUTELY NO WARRANTY;
This program is free software; your freedom to use, change
and distribute this program is protected by the GPL.
[root@localhost ~]#
```

(27) grep 命令查找文件中包含指定字符串的行。

grep 是一个强大的文本搜索命令,它能使用正则表达式搜索文本,并把匹配的行输出。在 grep 命令中,字符“^”表示行的开始,字符“\$”表示行的结束,如果要查找的字符串中带有空格,则可以用单引号或双引号将其引起来。其命令格式如下。

```
grep [选项] [正则表达式] 文件名
```

grep 命令各选项及其功能说明,如表 5.21 所示。

表 5.21 grep 命令各选项及其功能说明

选 项	功 能 说 明
-a	对二进制文件以文本文件的方式搜索数据
-c	对匹配的行计数
-i	忽略字母大小写的不同
-l	只显示包含匹配模式的文件名
-n	每个匹配行只按照相对的行号显示
-v	反向选择,列出不匹配的行

使用 grep 命令查找文件位置时,执行以下命令。

```
[root@localhost ~]# grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
csg:x:1000:1000:root:/home/csg:/bin/bash
[root@localhost ~]# grep -il "root" /etc/passwd
/etc/passwd
[root@localhost ~]#
```

grep 与 find 命令的差别在于, grep 是在文件中搜索满足条件的行, 而 find 是在指定目录下根据文件的相关信息查找满足指定条件的文件。

(28) sort 命令对文本文件内容进行排序。

sort 命令用于将文本文件内容加以排序。其命令格式如下。

```
sort [选项] 文件名
```

sort 命令各选项及其功能说明, 如表 5.22 所示。

表 5.22 sort 命令各选项及其功能说明

选 项	功 能 说 明
-b	忽略前导的空白区域
-c	检查输入是否已排序, 若已排序, 则不进行操作
-d	只考虑空白区域和字母字符
-f	忽略字母大小写
-i	除了 040~176 中的 ASCII 字符外, 忽略其他的字符
-m	将几个排序好的文件合并
-M	将前面 3 个字母依照月份的缩写进行排序
-n	依照数值的大小进行排序
-o	将结果写入文件而非标准输出
-r	逆序输出排序结果
-s	禁用 last-resort 比较, 以稳定比较算法
-t	使用指定的分隔符代替非空格到空格的转换
-u	配合 -c 时, 严格校验排序; 不配合 -c 时, 只输出一次排序结果
-z	以 0 字节而非新行作为行尾标志

使用 sort 命令时, 可针对文本文件的内容, 以行为单位来进行排序, 执行命令如下。

```
[root@localhost ~]# cat testfile05.txt //查看 testfile05.txt 文件的内容
test 10
open 20
hello 30
welcome 40
[root@localhost ~]# sort testfile05.txt //排序 testfile05.txt 文件的内容
hello 30
open 20
test 10
welcome 40
[root@localhost ~]#
```

sort 命令将以默认的方式使文本文件的第一列以 ASCII 码的次序排列, 并将结果输出到标准输出。

## 5.2.4 Vi、Vim 编辑器的使用

可视化接口 (Visual interface, Vi) 也称为可视化界面, 它为用户提供了一个全屏幕的窗口编辑器, 窗口中一次可以显示一屏的编辑内容, 并可以上下滚动。Vi 是所有 UNIX 和 Linux 操作系统中的标准编辑器, 类似于 Windows 操作系统中的记事本, 对于 UNIX 和 Linux 操作系统中的任何



版本,Vi 编辑器都是完全相同的,因此可以在其他任何介绍 Vi 的地方进一步了解它,Vi 也是 Linux 中最基本的文本编辑器,学会它后,可以在 Linux,尤其是在终端中畅通无阻。

Vim(Visual interface improved,Vim)可以看作 Vi 的改进升级版,Vi 和 Vim 都是 Linux 操作系统中的编辑器,不同的是,Vim 比较高级,Vi 用于文本编辑,但 Vim 更适用于面向开发者的云端开发平台。

Vim 可以执行输出、移动、删除、查找、替换、复制、粘贴、撤销、块操作等众多文件操作,而且用户可以根据自己的需要对其进行定制,这是其他编辑程序没有的。但 Vim 不是一个排版程序,它不像 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排,它只是一个文件编辑程序,Vim 是全屏幕文件编辑器,没有菜单,只有命令。

在命令行中执行命令 `vim filename`,如果 `filename` 已经存在,则该文件被打开且显示其内容;如果 `filename` 不存在,则 Vim 在第一次存盘时自动在硬盘中新建 `filename` 文件。

Vim 有 3 种基本工作模式:命令模式、编辑模式、末行模式。考虑到各种用户的需要,采用状态切换的方法可以实现工作模式的转换,切换只是习惯性的问题,一旦熟练使用 Vim,就会觉得它非常易于使用。

### 1. 命令模式

命令模式(其他模式→Esc)是用户进入 Vim 的初始状态,在此模式下,用户可以输入 Vim 命令,使 Vim 完成不同的工作任务,如光标移动、复制、粘贴、删除等,也可以从其他模式返回到命令模式,在编辑模式下按 Esc 键或在末行模式下输入错误命令都会返回到命令模式。Vim 命令模式的光标移动命令,如表 5.23 所示;Vim 命令模式的复制和粘贴命令,如表 5.24 所示;Vim 命令模式的删除操作命令,如表 5.25 所示;Vim 命令模式的撤销与恢复操作命令,如表 5.26 所示。

表 5.23 Vim 命令模式的光标移动命令

操 作	功 能 说 明
gg	将光标移动到文章的首行
G	将光标移动到文章的尾行
w 或 W	将光标移动到下一个单词
H	将光标移动到该屏幕的顶端
M	光标移动到该屏幕的中间
L	将光标移动到该屏幕的底端
h(←)	将光标向左移动一格
l(→)	将光标向右移动一格
j(↓)	将光标向下移动一格
k(↑)	将光标向上移动一格
0(Home)	数字 0,将光标移至行首
\$(End)	将光标移至行尾
Page Up/Page Down	(Ctrl+B/Ctrl+F)上下翻页

表 5.24 Vim 命令模式的复制和粘贴命令

操 作	功 能 说 明
yy 或 Y(大写)	复制光标所在的整行
3yy 或 y3y	复制 3 行(含当前行,后 3 行),如复制 5 行,则使用 5yy 或 y5y 即可

续表

操 作	功 能 说 明
y1G	复制至行文件首
yG	复制至行文件尾
yw	复制一个单词
y2w	复制两个字符
p(小写)	粘贴到光标的后(下)面,如果复制的是整行,则粘贴到光标所在行的下一行
P(大写)	粘贴到光标的前(上)面,如果复制的是整行,则粘贴到光标所在行的上一行

表 5.25 Vim 命令模式的删除操作命令

操 作	功 能 说 明
dd	删除当前行
3dd 或 d3d	删除 3 行(含当前行,后 3 行),如删除 5 行,则使用 5dd 或 d5d 即可
d1G	删除至文件首
dG	删除至文件尾
D 或 d\$	删除至行尾
dw	删除至词尾
ndw	删除后面的 n 个词

表 5.26 Vim 命令模式的撤销与恢复操作命令

操 作	功 能 说 明
U(小写)	取消上一个更改(常用)
U(大写)	取消一行内的所有更改
Ctrl+R	重做一个动作(常用),通常与“u”配合使用,将会为编辑提供很多方便
.	重复前一个动作,如果想要重复删除、复制、粘贴等,则按“.”键即可

## 2. 编辑模式

在编辑模式(命令模式→a/A,i/I 或 o/O)下,可对编辑的文件添加新的内容并进行修改,这是该模式的唯一功能。进入该模式时,可按 a/A,i/I 或 o/O 键。Vim 编辑模式命令,如表 5.27 所示。

表 5.27 Vim 编辑模式命令

操 作	功 能 说 明
a(小写)	在光标之后插入内容
A(大写)	在光标当前行的末尾插入内容
i(小写)	在光标之前插入内容
I(大写)	在光标当前行的开始部分插入内容
o(小写)	在光标所在行的下面新增一行
O(大写)	在光标所在行的上面新增一行

## 3. 末行模式

末行模式(命令模式→: 或/与?)主要用来进行一些文字编辑辅助功能,如查找、替换、文件保存等,在命令模式下输入“:”字符,即可进入末行模式,若输入命令完成或命令出错,则会退出 Vim 或返回到命令模式。Vim 末行模式命令,如表 5.28 所示,按 Esc 键可返回命令模式。

表 5.28 Vim 末行模式命令

操 作	功 能 说 明
ZZ(大写)	保存当前文件并退出
:wq 或 :x	保存当前文件并退出
:q	结束 Vim 程序,如果文件有过修改,则必须先保存文件
:q!	强制结束 Vim 程序,修改后的文件不会保存
:w[文件路径]	保存当前文件,将其保存为另一个文件(类似于另存为新文件)
:r[filename]	在编辑的数据中,读入另一个文件的数据,即将 filename 文件的内容追加到光标所在行的后面
!:command	暂时退出 Vim 到命令模式下执行 command 的显示结果,如“:!ls/home”表示可在 Vim 中查看/home 下以 ls 输出的文件信息
:set nu	显示行号,设定之后,会在每一行的前面显示该行的行号
:set nonu	与:set nu 相反,用于取消行号

在命令模式下输入“:”字符,即可进入末行模式,在末行模式下可以进行查找与替换操作,其命令格式如下。

```
: [range] s/pattern/string/[c,e,g,i]
```

查找与替换操作各选项及其功能说明,如表 5.29 所示。

表 5.29 查找与替换操作各选项及其功能说明

选 项	功 能 说 明
range	指的是范围,如“1,5”指从第 1 行至第 5 行,“1,\$”指从首行至最后一行,即整篇文章
s(search)	表示查找搜索
pattern	要被替换的字符串
string	将用 string 替换 pattern 的内容
c(confirm)	每次替换前会询问
e(error)	不显示 error
g(globe)	不询问,将做整行替换
i(ignore)	不区分字母大小写

在命令模式下输入“/”或“?”字符,即可进入末行模式,在末行模式下可以进行查找操作,其命令格式如下。

```
/word 或 ?word
```

查找操作各选项及其功能说明,如表 5.30 所示。

表 5.30 查找操作各选项及其功能说明

选 项	功 能 说 明
/word	向光标之下寻找一个名称为 word 的字符串。例如,要在文件中查找“welcome”字符串,则输入/welcome 即可
?word	向光标之上寻找一个名称为 word 的字符串
n	代表英文按键,表示重复前一个查找的动作。例如,如果刚刚执行了/welcome 命令向下查找 welcome 字符串,则按下 n 键后,会继续向下查找下一个名称为 welcome 的字符串;如果执行了?welcome 命令,那么按 n 键会向上查找名称为 welcome 的字符串

续表

选 项	功 能 说 明
N	代表英文按键,与 n 刚好相反,为反向进行前一个查找动作。例如,执行/welcome 命令后,按 N 键表示向上查找 welcome

Vim 编辑器的使用如下。

(1) 在当前目录下新建文件 newtest.txt,输入文件内容,执行以下命令。

```
[root@localhost ~]# vim newtest.txt //创建新文件 newtest.txt
```

在命令模式下按 a/A,i/I 或 o/O 键,进入编辑模式,完成以下内容的输入。

```
1 hello
2 everyone
3 welcome
4 to
5 here
```

输入以上内容后,按 Esc 键,从编辑模式返回到命令模式,再输入大写字母“ZZ”,退出并保存文件内容。

(2) 复制第 2 行与第 3 行文本到文件尾,同时删除第 1 行文本。

按 Esc 键,从编辑模式返回到命令模式,将光标移动到第 2 行,在键盘上连续按 2yy 键,再按 G 键,将光标移动到文件最后一行,按 p 键,复制第 2 行与第 3 行文本到文件尾,按 gg 键,将光标移动到文件首行,按 dd 键,删除第 1 行文本,执行以上操作命令后,显示的文件内容如下。

```
2 everyone
3 welcome
4 to
5 here
2 everyone
3 welcome
```

(3) 在命令模式下,输入“:”字符,进入末行模式,在末行模式下进行查找与替换操作,执行以下命令。

```
:1,$ s/everyone/myfriend/g
```

对整个文件进行查找,用 myfriend 字符串替换 everyone,无询问进行替换操作,执行命令后的操作结果如下。

```
2 myfriend
3 welcome
4 to
5 here
2 myfriend
3 welcome
```

(4) 在命令模式下,输入“?”或“/”,进行查询,执行以下命令。

```
/welcome
```

按 Enter 键后,可以看到光标位于第 2 行, welcome 闪烁显示,按 n 键,可以继续查找,可以看到光标已经移动到最后一行 welcome 处进行闪烁显示。按 a/A、i/I 或 o/O 键,进入编辑模式,按 Esc 键返回命令模式,再输入“ZZ”,保存文件并退出 Vim 编辑器。

## 5.2.5 磁盘配置与管理

从广义上来讲,硬盘、光盘和 U 盘等用来保存数据信息的存储设备都可以称为磁盘。其中,硬盘是计算机的重要组件,无论是在 Windows 操作系统还是在 Linux 操作系统中,都要使用硬盘。因此,规划和管理磁盘是非常重要的工作。



### 1. Linux 操作系统中的设备命名规则

在 Linux 操作系统中,每个硬件设备都有一个称为设备名称的特别名称,例如,对于接在 IDE1 的第一个硬盘(主硬盘),其设备名称为/dev/hda,也就是说,可以用“/dev/hda”来代表此硬盘。对于以下信息,相信读者能够一目了然。下面介绍硬盘设备在 Linux 操作系统中的命名规则。

IDE1 的第 1 个硬盘(master)/dev/hda;

IDE1 的第 2 个硬盘(slave) /dev/hdb;

.....

IDE2 的第 1 个硬盘(master)/dev/hdc;

IDE2 的第 2 个硬盘(slave) /dev/hdd;

.....

SCSI 的第 1 个硬盘 /dev/sda;

SCSI 的第 2 个硬盘 /dev/sdb;

.....

在 Linux 操作系统中,分区概念和 Windows 中的概念更加接近,按照功能的不同,硬盘分区可以分为以下几类。

(1) 主分区。在划分硬盘的第 1 个分区时,会指定其为主分区, Linux 最多可以让用户创建 4 个主分区,其主要用来存放操作系统的启动或引导程序,/boot 分区最好放在主分区中。

(2) 扩展分区。Linux 下的一个硬盘最多只允许有 4 个主分区,如果用户想要创建更多的分区,应该怎么办? 这就有了扩展分区概念。用户可以创建一个扩展分区,并在扩展分区中创建多个逻辑分区,从理论上来说,其逻辑分区没有数量限制。需要注意的是,创建扩展分区时,会占用一个主分区的位置,因此,如果创建了扩展分区,则一个硬盘中最多只能创建 3 个主分区和 1 个扩展分区。扩展分区不是用来存放数据的,它的主要功能是创建逻辑分区。

(3) 逻辑分区。逻辑分区不能被直接创建,它必须依附在扩展分区下,容量受到扩展分区大小的限制,逻辑分区通常用于存放文件和数据。

大部分设备的前缀名后面跟有一个数字,它唯一指定了某一设备;硬盘驱动器的前缀名后面跟有一个字母和一个数字,字母用于指明设备,而数字用于指明分区。因此,/dev/sda2 指定了硬盘上的一个分区,/dev/pts/10 指定了一个网络终端会话。设备节点前缀及设备类型说明,如表 5.31 所示。

表 5.31 设备节点前缀及设备类型说明

设备节点前缀	设备类型说明	设备节点前缀	设备类型说明
fb	FFrame 缓冲	ttyS	串口
fd	软盘	scd	SCSI 音频光驱
hd	IDE 硬盘	sd	SCSI 硬盘
lp	打印机	sg	SCSI 通用设备
par	并口	sr	SCSI 数据光驱
pt	伪终端	st	SCSI 磁带
tty	终端	md	磁盘阵列

一些 Linux 发行版用 SCSI 层访问所有固定硬盘,因此,虽然硬盘有可能并不是 SCSI 硬盘,但仍可以通过存储设备进行访问。

有了磁盘命名和分区命名的概念,理解诸如/dev/hda1 之类的分区名称就不难了,分区命名规则如下。

IDE1 的第 1 个硬盘(master)的第 1 个主分区 /dev/hda1;

IDE1 的第 1 个硬盘(master)的第 2 个主分区 /dev/hda2;

IDE1 的第 1 个硬盘(master)的第 1 个逻辑分区 /dev/hda5;

IDE1 的第 1 个硬盘(master)的第 2 个逻辑分区 /dev/hda6;

.....

IDE1 的第 2 个硬盘(slave)的第 1 个主分区 /dev/hdb1;

IDE1 的第 2 个硬盘(slave)的第 2 个主分区 /dev/hdb2;

.....

SCSI 的第 1 个硬盘的第 1 个主分区 /dev/sda1;

SCSI 的第 1 个硬盘的第 2 个主分区 /dev/sda2;

.....

SCSI 的第 2 个硬盘的第 1 个主分区 /dev/sdb1;

SCSI 的第 2 个硬盘的第 2 个主分区 /dev/sdb2;

.....

## 2. 添加新磁盘管理

新购置的物理硬盘,不管是用于 Windows 操作系统还是用于 Linux 操作系统,都要进行如下操作。

(1) 分区: 可以是一个分区或多个分区。

(2) 格式化: 分区必须经过格式化才能创建文件系统。

(3) 挂载: 被格式化的磁盘分区必须挂载到操作系统相应的文件目录下。

Windows 操作系统自动帮助用户完成了挂载分区到目录的工作,即自动将磁盘分区挂载到盘符; Linux 操作系统除了会自动挂载根分区启动项外,其他分区都需要用户自己配置,所有的磁盘都必须挂载到文件系统相应的目录下。

为什么要将一个硬盘划分成多个分区,而不是直接使用整个硬盘呢? 主要有如下原因。

(1) 方便管理和控制。可以将系统中的数据(包括程序)按不同的应用分成几类,之后将不同类型的数据分别存放在不同的磁盘分区中。由于在每个分区中存放的都是类似的数据或程序,因此管理和维护会简单很多。



(2) 提高系统的效率。给硬盘分区后,可以直接缩短系统读写磁盘时磁头移动的距离,也就是说,缩小了磁头搜寻的范围;反之,如果不使用分区,则每次在硬盘中搜寻信息时可能要搜寻整个硬盘,搜寻速度会很慢。另外,硬盘分区可以减轻碎片(文件不连续存放)所造成的系统效率下降的问题。

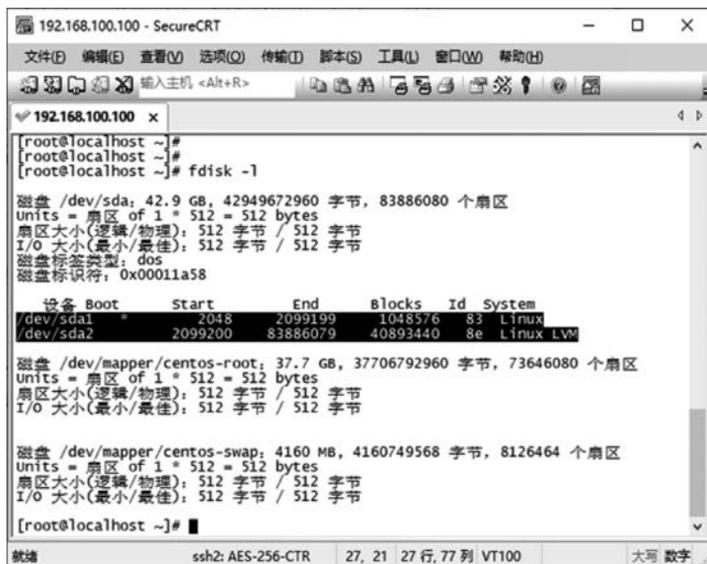
(3) 使用磁盘配额的功能限制用户使用的磁盘量。由于限制了用户使用磁盘配额的功能,即只能在分区一级上使用,所以为了限制用户使用磁盘的总量,防止用户浪费磁盘空间(甚至将磁盘空间耗光),最好先对磁盘进行分区,再分配给一般用户。

(4) 便于备份和恢复。硬盘分区后,可以只对所需的分区进行备份和恢复操作,这样备份和恢复的数据量会大大下降,操作也更简单和方便。

在进行分区、格式化和挂载操作之前,要先进行查看分区信息和在虚拟机中添加磁盘操作。

### 3. 查看分区信息

可以使用 `fdisk -l` 命令查看当前系统所有磁盘设备及其分区的信息,如图 5.1 所示。



```

192.168.100.100 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 窗口(W) 帮助(H)
输入主机 <Alt+R>
192.168.100.100 x
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# fdisk -l

磁盘 /dev/sda: 42.9 GB, 42949672960 字节, 83886080 个扇区
units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x00011a58

 设备 Boot      Start          End      Blocks   Id  System
/dev/sda1 *    2048          2099199    1048576   83  Linux
/dev/sda2      2099200      83886079    40893440   8e  Linux LVM

磁盘 /dev/mapper/centos-root: 37.7 GB, 37706792960 字节, 73646080 个扇区
units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘 /dev/mapper/centos-swap: 4160 MB, 4160749568 字节, 8126464 个扇区
units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

[root@localhost ~]#
ssh2: AES-256-CTR 27, 21 27 行, 77 列 VT100 大写 数字

```

图 5.1 查看当前系统所有磁盘设备及其分区的信息

从图 5.1 中可以看出,安装系统时,硬盘分为 `/root` 分区、`/boot` 分区和 `/swap` 分区,其中分区信息的各字段的含义如下。

- (1) 设备: 分区的设备文件名称,如 `/dev/sda`。
- (2) Boot: 是否为引导分区。若是,则带有“\*”标识 `/dev/sda1 *`。
- (3) Start: 该分区在硬盘中的起始位置(柱面数)。
- (4) End: 该分区在硬盘中的结束位置(柱面数)。
- (5) Blocks: 分区大小。
- (6) Id: 分区类型的 Id。ext4 分区的 Id 为 83,LVM 分区的 Id 为 8e。
- (7) System: 分区类型。其中,“Linux”代表 ext4 文件系统,“Linux LVM”代表逻辑卷。

### 4. 在虚拟机中添加硬盘

练习硬盘分区操作时,需要先在虚拟机中添加一块新的硬盘,由于 SCSI 接口的硬盘支持热插拔,因此可以在虚拟机开机的状态下直接添加硬盘。

(1) 打开虚拟机软件,选择“虚拟机”→“设置”选项,如图 5.2 所示。



图 5.2 选择“虚拟机”→“设置”选项

(2) 弹出“虚拟机设置”对话框,如图 5.3 所示。



图 5.3 “虚拟机设置”对话框

(3) 单击“添加”按钮,弹出“添加硬件向导”对话框,如图 5.4 所示。

(4) 在硬件类型界面中,选择“硬盘”选项,单击“下一步”按钮,进入选择磁盘类型界面,如图 5.5 所示。

(5) 选中 SCSI 单选按钮,单击“下一步”按钮,进入选择磁盘界面,如图 5.6 所示。

(6) 选中“创建新虚拟磁盘”单选按钮,单击“下一步”按钮,进入指定磁盘容量界面,如图 5.7 所示。



图 5.4 “添加硬件向导”对话框



图 5.5 选择磁盘类型界面



图 5.6 选择磁盘界面

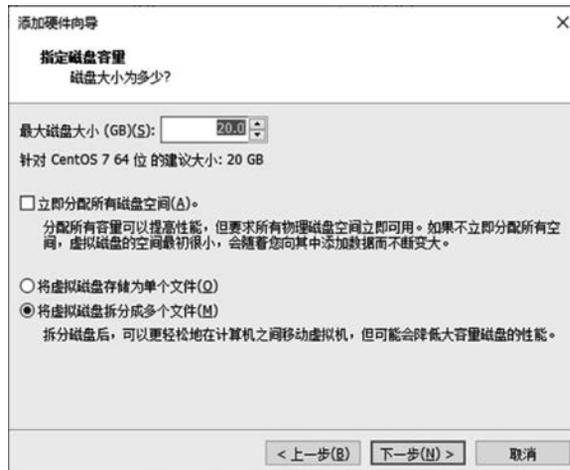


图 5.7 指定磁盘容量界面

(7) 设置最大磁盘大小,单击“下一步”按钮,进入指定磁盘文件界面,如图 5.8 所示。

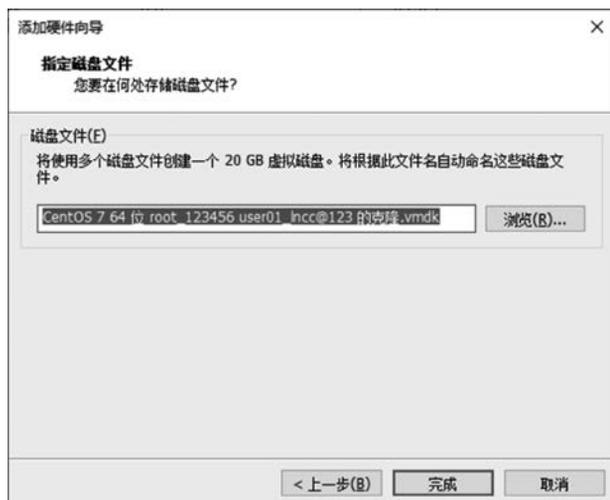


图 5.8 指定磁盘文件界面

(8) 单击“完成”按钮,完成在虚拟机中添加硬盘的工作,返回“虚拟机设置”对话框,可以看到刚刚添加的 20GB 的 SCSI 硬盘,如图 5.9 所示。



图 5.9 添加的 20GB 的 SCSI 硬盘

(9) 单击“确定”按钮,返回虚拟机主界面,重新启动 Linux 操作系统,再执行 `fdisk -l` 命令查看硬盘分区信息,如图 5.10 所示,可以看到新增加的硬盘/dev/sdb,系统识别到新的硬盘后,即可在该硬盘中建立新的分区。

```

192.168.100.100 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 窗口(W) 帮助(H)
输入主机 <Alt+R>
192.168.100.100 x
[root@localhost ~]#
[root@localhost ~]# fdisk -l

磁盘 /dev/sda: 42.9 GB, 42949672960 字节, 83886080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x00011a58

   设备 Boot      Start         End      Blocks   Id  System
/dev/sda1 *      2048         2099199   1048576   83  Linux
/dev/sda2                2099200     83886079   40893440   8e  Linux LVM

磁盘 /dev/sdb: 21.5 GB, 21474836480 字节, 41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘 /dev/mapper/centos-root: 37.7 GB, 37706792960 字节, 73646080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘 /dev/mapper/centos-swap: 4160 MB, 4160749568 字节, 8126464 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

[root@localhost ~]#

```

图 5.10 查看硬盘分区信息



## 5. 磁盘分区

在安装 Linux 操作系统时,其中有一个步骤是进行磁盘分区,在分区时可以采用 RAID 和 LVM 等方式,除此之外,Linux 操作系统中还提供了 fdisk、cfdisk、parted 等分区工具,这里主要介绍 fdisk 分区工具。

fdisk 磁盘分区工具在 DOS、Windows 和 Linux 操作系统中都有相应的应用程序,在 Linux 操作系统中,fdisk 是基于菜单的命令,对磁盘进行分区时,可以在 fdisk 命令后面直接加上要分区的磁盘作为参数。其命令格式如下。

```

fdisk [选项] <磁盘>      更改分区表
fdisk [选项] -l <磁盘>   列出分区表
fdisk -s <分区>         给出分区大小(块数)

```

fdisk 命令各选项及其功能说明如表 5.32 所示。

表 5.32 fdisk 命令各选项及其功能说明

选 项	功 能 说 明
-b <大小>	区大小(512、1024、2048 或 4096)
-c [= <模式>]	兼容模式: dos 或 nondos(默认)
-h	输出此帮助文本
-u [= <单位>]	显示单位: cylinders(柱面)或 sectors(扇区,默认)
-v	输出程序版本
-C <数字>	指定柱面数
-H <数字>	指定磁头数
-S <数字>	指定每个磁道的扇区数

在对新增加的第 2 块 SCSI 硬盘进行分区时,执行命令如下。

```
[root@localhost ~]# fdisk /dev/sdb
欢迎使用 fdisk (util - Linux 5.23.2)。
更改将停留在内存中,直到您决定将更改写入磁盘。
使用写入命令前请三思。
Device does not contain a recognized partition table
使用磁盘标识符 0xfb0b9128 创建新的 DOS 磁盘标签。
命令(输入 m 获取帮助):m
命令操作
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  g  create a new empty GPT partition table
  G  create an IRIX (SGI) partition table
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)
命令(输入 m 获取帮助):
```

在“命令(输入 m 获取帮助):”提示符后若输入“m”,则可以查看所有命令的帮助信息,输入相应的命令可选择需要的操作。fdisk 命令操作及其功能说明如表 5.33 所示。

表 5.33 fdisk 命令操作及其功能说明

命令操作	功能说明	命令操作	功能说明
a	设置可引导标签	o	建立空白 DOS 分区表
b	编辑 BSD 磁盘标签	p	显示分区列表
c	设置 DOS 操作系统兼容标记	q	不保存并退出
d	删除一个分区	s	新建空白 SUN 磁盘标签
g	新建一个空的 GPT 分区表	t	改变一个分区的系统 ID
G	新建一个 IRIX(SGI)分区表	u	改变显示记录单位
l	显示已知的文件系统类型,82 为 Linux Swap 分区,83 为 Linux 分区	v	验证分区表
m	显示帮助菜单	w	保存并退出
n	新建分区	x	附加功能(仅专家)

使用 fdisk 命令对新增加的 SCSI 硬盘/dev/sdb 进行分区操作,在此硬盘中创建两个主分区和一个扩展分区,在扩展分区中再创建两个逻辑分区。

(1) 执行 fdisk /dev/sdb 命令,进入交互的分区管理界面,在“命令(输入 m 获取帮助):”提示符后,用户可以输入特定的分区操作命令来完成各项分区管理任务,输入“n”可以进行创建分区

的操作,包括创建主分区、扩展分区和逻辑分区,根据提示继续输入“p”选择创建主分区,输入“e”选择创建扩展分区,之后依次选择分区序号、起始位置、结束位置或分区大小即可创建新分区。

选择分区号时,主分区和扩展分区的序号只能为1~4,分区的起始位置一般由fdisk命令默认识别,结束位置或分区大小可以使用类似于“+size{K,M,G}”的形式,如“+2G”表示将分区的容量设置为2GB。

下面先创建一个容量为5GB的主分区,主分区创建结束之后,输入“p”查看已创建好的分区/dev/sdb1,执行命令如下。

```
命令(输入 m 获取帮助):n
Partition type:
   p  primary (0 primary, 0 extended, 4 free)
   e  extended
Select (default p): p
分区号 (1 - 4, 默认 1):1
起始 扇区 (2048 - 41943039, 默认为 2048):
将使用默认值 2048
Last 扇区, + 扇区 or + size{K,M,G} (2048 - 41943039, 默认为 41943039): + 5G
分区 1 已设置为 Linux 类型,大小设为 5 GB

命令(输入 m 获取帮助):p

磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x0bcee221

   设备 Boot      Start       End          Blocks      Id System
/dev/sdb1                2048        10487807    5242880     83  Linux

命令(输入 m 获取帮助):
```

(2) 继续创建第2个容量为3GB的主分区,主分区创建结束之后,输入“p”查看已创建好的分区/dev/sdb1、/dev/sdb2,执行命令如下。

```
命令(输入 m 获取帮助):n
Partition type:
   p  primary (1 primary, 0 extended, 3 free)
   e  extended
Select (default p): p
分区号 (2 - 4, 默认 2):2
起始 扇区 (10487808 - 41943039, 默认为 10487808):
将使用默认值 10487808
Last 扇区, + 扇区 or + size{K,M,G} (10487808 - 41943039, 默认为 41943039): + 3G
分区 2 已设置为 Linux 类型,大小设为 3 GB

命令(输入 m 获取帮助):p

磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
```

```
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x0bcee221
```

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	10487807	5242880	83	Linux
/dev/sdb2		10487808	16779263	3145728	83	Linux

命令(输入 m 获取帮助):

(3) 继续创建扩展分区,需要特别注意的是,必须将所有的剩余磁盘空间都分配给扩展分区,输入“e”创建扩展分区,扩展分区创建结束之后,输入“p”查看已经创建好的主分区和扩展分区,执行命令如下。

```
命令(输入 m 获取帮助):n
Partition type:
  p primary (2 primary, 0 extended, 2 free)
  e extended
Select (default p): e
分区号 (3,4,默认 3):
起始 扇区 (16779264 - 41943039,默认为 16779264):
将使用默认值 16779264
Last 扇区, + 扇区 or + size{K,M,G} (16779264 - 41943039,默认为 41943039):
将使用默认值 41943039
分区 3 已设置为 Extended 类型,大小设为 12 GB
```

命令(输入 m 获取帮助):p

```
磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x0bcee221
```

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	10487807	5242880	83	Linux
/dev/sdb2		10487808	16779263	3145728	83	Linux
/dev/sdb3		16779264	41943039	12581888	5	Extended

命令(输入 m 获取帮助):

扩展分区的起始扇区和结束扇区使用默认值即可,可以把所有的剩余磁盘空间(共 12GB)全部分配给扩展分区,从以上操作可以看出,划分的两个主分区的容量分别为 5GB 和 3GB,扩展分区的容量为 12GB。

(4) 扩展分区创建完成后即可创建逻辑分区,在扩展分区中再创建两个逻辑分区,磁盘容量分别为 8GB 和 4GB,在创建逻辑分区的时候不需要指定分区编号,系统会自动从 5 开始顺序编号,执行命令如下。

```

命令(输入 m 获取帮助):n
Partition type:
  p primary (2 primary, 1 extended, 1 free)
  l logical (numbered from 5)
Select (default p): l
添加逻辑分区 5
起始 扇区 (16781312 - 41943039, 默认为 16781312):
将使用默认值 16781312
Last 扇区, + 扇区 or + size{K,M,G} (16781312 - 41943039, 默认为 41943039): + 8G
分区 5 已设置为 Linux 类型,大小设为 8 GB

命令(输入 m 获取帮助):n
Partition type:
  p primary (2 primary, 1 extended, 1 free)
  l logical (numbered from 5)
Select (default p): l
添加逻辑分区 6
起始 扇区 (33560576 - 41943039, 默认为 33560576):
将使用默认值 33560576
Last 扇区, + 扇区 or + size{K,M,G} (33560576 - 41943039, 默认为 41943039):
将使用默认值 41943039
分区 6 已设置为 Linux 类型,大小设为 4 GB

命令(输入 m 获取帮助):

```

(5) 再次输入“p”,查看分区创建情况,执行命令如下。

```

命令(输入 m 获取帮助):p

磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x0bcee221

   设备 Boot      Start       End    Blocks   Id  System
/dev/sdb1            2048    10487807    5242880   83  Linux
/dev/sdb2    10487808    16779263    3145728   83  Linux
/dev/sdb3    16779264    41943039   12581888    5  Extended
/dev/sdb5    16781312    33558527    8388608   83  Linux
/dev/sdb6    33560576    41943039    4191232   83  Linux

命令(输入 m 获取帮助):

```

(6) 完成对硬盘的分区以后,输入“w”保存并退出,或输入“q”不保存并退出 fdisk。硬盘分区完成以后,一般需要重启系统以使设置生效;如果不想重启系统,则可以使用 partprobe 命令使系统获取新的分区表的情况。这里可以使用 partprobe 命令重新查看/dev/sdb 硬盘中分区表的变化情况,执行命令如下。

```

命令(输入 m 获取帮助):w
The partition table has been altered!
Calling ioctl() to re-read partition table.
正在同步磁盘.
[root@localhost ~]# partprobe /dev/sdb
[root@localhost ~]# fdisk -l
磁盘 /dev/sda:45.9 GB, 42949672960 字节,83886080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x00011a58
  设备 Boot      Start          End      Blocks   Id  System
/dev/sda1  *            2048          2099199    1048576   83  Linux
/dev/sda2                2099200       83886079    40893440   8e  Linux          LVM
磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x0bcee221
  设备 Boot      Start          End      Blocks   Id  System
/dev/sdb1                2048          10487807    5242880   83  Linux
/dev/sdb2            10487808       16779263    3145728   83  Linux
/dev/sdb3            16779264       41943039    12581888   5   Extended
/dev/sdb5            16781312       33558527    8388608   83  Linux
/dev/sdb6            33560576       41943039    4191232   83  Linux
磁盘 /dev/mapper/CentOS-root:37.7 GB, 37706792960 字节,73646080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘 /dev/mapper/CentOS-swap:4160 MB, 4160749568 字节,8126464 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节

```

至此,已经完成了新增加硬盘的分区操作。

## 6. 磁盘格式化

完成分区创建之后,还不能直接使用磁盘,必须经过格式化才能使用,这是因为操作系统必须按照一定的方式来管理磁盘,并使系统识别出来,所以磁盘格式化的作用就是在分区中创建文件系统。Linux 操作系统专用的文件系统是 ext,包含 ext3、ext4 等诸多版本,CentOS 中默认使用 ext4 文件系统。

mkfs 命令的作用是在磁盘中创建 Linux 文件系统,mkfs 命令本身并不执行建立文件系统的工作,而是调用相关的程序来实现。其命令格式如下。

```
mkfs [选项] [-t <类型>] [文件系统选项] <设备> [<大小>]
```

mkfs 命令各选项及其功能说明,如表 5.34 所示。



表 5.34 mkfs 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-t	文件系统类型；若不指定,则使用 ext2	-v	显示版本信息并退出
-V	解释正在进行的操作	-h	显示帮助信息并退出

将新增加的 SCSI 硬盘分区/dev/sdb1 按 ext4 文件系统进行格式化,执行命令如下。

```
[root@localhost ~]# mkfs                               //输入完命令后连续按两次 Tab 键
mkfs      mkfs.cramfs  mkfs.ext3  mkfs.fat  mkfs.msdos  mkfs.xfs
mkfs.btrfs  mkfs.ext2  mkfs.ext4  mkfs.minix  mkfs.vfat
[root@localhost ~]# mkfs -t ext4 /dev/sdb1           //按 ext4 文件系统格式化
mke2fs 5.45.9 (28 - Dec - 2013)
文件系统标签 =
...
Writing superblocks and filesystem accounting information: 完成
```

使用同样的方法对/dev/sdb2、/dev/sdb5 和/dev/sdb6 进行格式化,需要注意的是,格式化时会清除分区中的所有数据,为了保证系统安全,要备份重要数据。



## 7. 磁盘挂载与卸载

挂载就是指定系统中的一个目录作为挂载点,用户通过访问这个目录来实现对硬盘分区数据的存取操作,作为挂载点的目录相当于一个访问硬盘分区的入口。例如,将/dev/sdb6 挂载到/mnt 目录中,当用户在/mnt 目录下执行相关数据的存储操作时,Linux 操作系统会到/dev/sdb6 上执行相关操作。磁盘挂载示意图,如图 5.11 所示。

在安装 Linux 操作系统的过程中,自动建立或识别的分区通常会由系统自动完成挂载工作,如/root 分区、/boot 分区等,新增加的硬盘分区、光盘、U 盘等设备,都必须由管理员手动挂载到系统目录中。

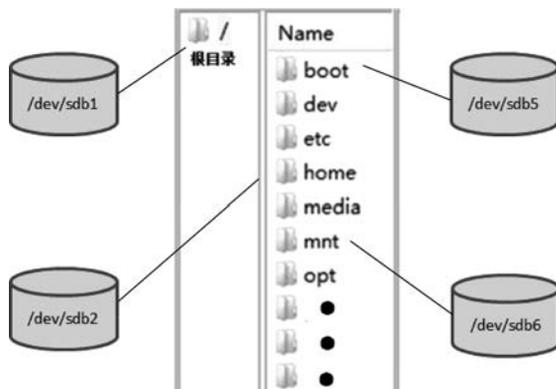


图 5.11 磁盘挂载示意图

Linux 操作系统中提供了两个默认的挂载目录: /media 和/mnt。

- ① /media 用作系统自动挂载点。
- ② /mnt 用作手动挂载点。

从理论上讲,Linux 操作系统中的任何一个目录都可以作为挂载点,但从系统的角度出发,以下几个目录是不能作为挂载点使用的: /bin、/sbin、/etc、/lib、/lib64。

(1) 手动挂载。

mount 命令的作用是将一个设备(通常是存储设备)挂载到一个已经存在的目录中,访问这个目录就是访问该存储设备。其命令格式如下。

```
mount [选项] [ -- source] <源> | [ -- target] <目录>
```

mount 命令各选项及其功能说明,如表 5.35 所示。

表 5.35 mount 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-a	挂载 fstab 中的所有文件系统	-n	不写/etc/mtab
-c	不对路径进行规范化	-o	挂载选项列表,以英文逗号分隔
-f	空运行;跳过 mount(2)系统调用	-r	以只读方式挂载文件系统(同-o ro)
-F	对每个设备禁用 fork(和-a 选项一起使用)	-t	限制文件系统类型集合
-T	/etc/fstab 的替代文件	-v	输出当前进行的操作
-h	显示帮助信息并退出	-V	显示版本信息并退出
-i	不调用 mount.<类型> 助手程序	-w	以读写方式挂载文件系统(默认)
-l	列出所有带有指定标签的挂载		

mount 命令-t<文件系统类型>与-o<选项>参数选项及其含义,如表 5.36 所示。

表 5.36 mount 命令-t<文件系统类型>与-o<选项>参数选项及其含义

-t <文件系统类型>		-o <选项>	
选 项	含 义	选 项	含 义
ext4/xfs	Linux 目前常用的文件系统	ro	以只读方式挂载
msdos	DOS 的文件系统,即 FAT16 文件系统	rw	以读写方式挂载
vfat	FAT32 文件系统	remount	重新挂载已经挂载的设备
iso9660	CD-ROM 文件系统	user	允许一般用户挂载设备
ntfs	NTFS 文件系统	nouser	不允许一般用户挂载设备
auto	自动检测文件系统	codepage=xxx	代码页
swap	交换分区的系统类型	iocharset=xxx	字符集

设备文件名对应分区的设备文件名,如/dev/sdb1;挂载点为用户指定的用于挂载点的目录,挂载点的目录需要满足以下几方面的要求。

- ① 目录事先存在,可使用 mkdir 命令新建目录。
- ② 挂载点目录不可被其他进程使用。
- ③ 挂载点的原有文件将被隐藏。

将新增加的 SCSI 硬盘分区/dev/sdb1、/dev/sdb2、/dev/sdb5 和/dev/sdb6 分别挂载到/mnt/data01、/mnt/data02、/mnt/data05 和/mnt/data06 目录中,执行命令如下。

```
[root@localhost cdrom]# cd /mnt
[root@localhost mnt]# mkdir data01 data02 data05 data06 //新建目录
[root@localhost mnt]# ls -l | grep '^d' //显示使用 grep 命令查找的以"d"开头的目录
drwxr-xr-x. 2 root root 6 8月 25 11:59 data01
drwxr-xr-x. 2 root root 6 8月 25 11:59 data02
drwxr-xr-x. 2 root root 6 8月 25 11:59 data05
drwxr-xr-x. 2 root root 6 8月 25 11:59 data06
```

```
dr--r--r--. 4 root root 82 8月 21 09:02 test
[root@localhost mnt]# mount /dev/sdb1 /mnt/data01 //挂载目录
[root@localhost mnt]# mount /dev/sdb2 /mnt/data02
[root@localhost mnt]# mount /dev/sdb5 /mnt/data05
[root@localhost mnt]# mount /dev/sdb6 /mnt/data06
```

完成挂载后,可以使用 df 命令查看挂载情况,df 命令主要用来查看系统中已经挂载的各个文件系统的磁盘使用情况,使用该命令可获取硬盘被占用的空间,以及目前剩余空间等信息。其命令格式如下。

```
df [选项] [文件]
```

df 命令各选项及其功能说明,如表 5.37 所示。

表 5.37 df 命令各选项及其功能说明

选 项	功 能 说 明
-a	显示所有文件系统的磁盘使用情况
-h	以人类易读的格式输出
-H	等于-h,但计算时,1K 表示 1000,而不是 1024
-T	输出所有已挂载文件系统的类型
-i	输出文件系统的 i-node 信息,如果 i-node 满了,则即使有空间也无法存储
-k	按块大小输出文件系统磁盘使用情况
-l	只显示本机的文件系统

使用 df 命令查看磁盘使用情况,执行命令如下。

```
[root@localhost mnt]# df -hT
文件系统          类型    容量    已用    可用    已用%    挂载点
/dev/mapper/CentOS - root  xfs    36G    5.2G    30G    15%     /
...
/dev/sdb1          ext4    5.8G    20M    5.6G    1%      /mnt/data01
/dev/sdb2          ext4    5.9G    9.0M    5.8G    1%      /mnt/data02
/dev/sdb5          ext4    7.8G    36M    7.3G    1%      /mnt/data05
/dev/sdb6          ext4    3.9G    16M    3.7G    1%      /mnt/data06
```

## (2) 光盘挂载。

Linux 将一切视为文件,光盘也不例外,识别出来的设备会存放在/dev 目录下,需要将它挂载在一个目录中,才能以文件形式查看或者使用光盘。

使用 mount 命令实现光盘挂载,执行命令如下。

```
[root@localhost ~]# mount /dev/cdrom /media
mount: /dev/sr0 写保护,将以只读方式挂载
```

也可以使用以下命令进行光盘挂载。

```
[root@localhost ~]# mount /dev/sr0 /media
mount: /dev/sr0 写保护,将以只读方式挂载
```



显示磁盘使用情况,执行命令如下。

```
[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      36G   5.2G   30G   15%    /
devtmpfs          devtmpfs  5.9G   0      5.9G   0%    /dev
tmpfs             tmpfs     5.9G   0      5.9G   0%    /dev/shm
...
/dev/sr0          iso9660   5.3G   5.3G   0      100%   /media
[root@localhost ~]#
```

显示磁盘挂载目录文件内容,执行命令如下。

```
[root@localhost ~]# ls -l /media
总用量 686
-rw-rw-r--. 1 root root 14 11月 26 2018 CentOS_BuildTag
...
-rw-rw-r--. 1 root root 1690 12月 10 2015 RPM-GPG-KEY-CentOS-Testing-7
-r--r--r--. 1 root root 2883 11月 26 2018 TRANS.TBL
[root@localhost ~]#
```

### (3) U 盘挂载。

Linux 将一切视为文件,U 盘也不例外,识别出来的设备会存放在/dev 目录下,需要将它挂载在一个目录中,才能以文件形式查看或者使用 U 盘。

使用 mount 命令实现 U 盘挂载,执行命令如下。

① 插入 U 盘,使用 fdisk -l 命令查看 U 盘是否被测试到,查看相关信息。

```
[root@localhost ~]# fdisk -l //查看U盘数据信息
磁盘 /dev/sdc:65.9 GB, 62930117632 字节,122910386 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x270b8f9b
 设备  Boot      Start         End      Blocks   Id System
/dev/sdc1  *        1060864     122910385     60924761    7  HPFS/NTFS/exFAT
```

② 进行 U 盘挂载,执行命令如下。

```
[root@localhost ~]# mkdir /mnt/u-disk
[root@localhost ~]# mount /dev/sdc1 /mnt/u-disk
mount: 未知的文件系统类型"NTFS"
```

从以上输出结果可以看出,无法进行 U 盘挂载,因为 U 盘文件系统类型为 NTFS,所以 Linux 操作系统默认情况下是无法识别的,需要安装支持 NTFS 格式的数据包,默认情况下,CentOS 7.6 安装光盘 ISO 镜像文件中包括 NTFS 格式的数据包,只是默认情况下不会自动安装,需要手动配置安装。

③ 挂载光盘,编辑库仓文件 local.repo,执行命令如下。

```
[root@localhost ~]# mount /dev/sr0 /media
mount: /dev/sr0 写保护,将以只读方式挂载
```

```
[root@localhost ~]# vim /etc/yum.repos.d/local.repo
[epel]
name = epel
baseurl = file:///media
gpgcheck = 0
enable = 1
"/etc/yum.repos.d/local.repo" 5L, 65C 已写入
```

④ 查看 NTFS 数据包,执行命令如下。

```
[root@localhost ~]# yum list | grep ntfs
ntfs-3g.x86_64                2:2017.3.23-15.el7          epel
ntfs-3g-devel.x86_64        2:2017.3.23-15.el7          epel
ntfsprogs.x86_64            2:2017.3.23-15.el7          epel
```

⑤ 安装 NTFS 数据包,执行命令如下。

```
[root@localhost ~]# yum install ntfs-3g - y
已加载插件:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
正在解决依赖关系
--> 正在检查事务
--> 软件包 ntfs-3g.x86_65.5.2017.3.23-15.el7 将被安装
--> 解决依赖关系完成
依赖关系解决
=====
Package      架构      版本          源          大小
=====
正在安装:
ntfs-3g      x86_64    2:2017.3.23-15.el7    epel        265 K
事务概要
=====
安装 1 软件包
总下载量:265 K
安装大小:612 K
...
正在安装 : 2:ntfs-3g-2017.3.23-15.el7.x86_64      1/1
验证中   : 2:ntfs-3g-2017.3.23-15.el7.x86_64      1/1
已安装:
ntfs-3g.x86_64 2:2017.3.23-15.el7
完毕!
```

⑥ 安装成功后即可进行 U 盘挂载,执行命令如下。

```
[root@localhost ~]# mount /dev/sdb1 /mnt/u-disk
The disk contains an unclean file system (0, 0).
The file system wasn't safely closed on Windows. Fixing.
```

⑦ U 盘挂载成功,查看 U 盘挂载情况,执行命令如下。

```
[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        36G    5.3G    31G     13%    /
...
/dev/sdb1          fuseblk    59G    5.7G    54G     9%     /mnt/u-disk
[root@localhost ~]#
```

⑧ 查看 U 盘数据信息,执行命令如下。

```
[root@localhost ~]# ls -l /mnt/u-disk
总用量 1
drwxrwxrwx. 1 root root 0 7月 8 17:12 GH0
...
drwxrwxrwx. 1 root root 0 8月 25 14:37 user02
```

(4) 自动挂载。

通过 mount 命令挂载的文件系统在 Linux 操作系统关机或重启时会被自动卸载,所以一般手动挂载磁盘之后要把挂载信息写入/etc/fstab 文件,系统在开机时会自动读取/etc/fstab 文件中的内容,根据文件中的配置挂载磁盘,这样就不需要每次开机启动之后都手动进行挂载了。/etc/fstab 文件称为系统数据表(File System Table),其会显示系统中已经存在的挂载信息。

使用 mount 命令实现 U 盘挂载,执行命令如下。

① 使用 cat /etc/fstab 命令查看文件内容。

```
[root@localhost ~]# cat /etc/fstab
# /etc/fstab
# Created by anaconda on Mon Jun 8 01:15:36 2020
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper /CentOS - root /                xfs    defaults    0 0
UUID = 6d58086e - 0a6b - 4399 - 93dc - c2016ea17fe0 /boot xfs    defaults    0 0
/dev/mapper /CentOS - swap swap    defaults    0 0
```

/etc/fstab 文件中的每一行对应一个自动挂载设备,每行包括 6 列。/etc/fstab 文件字段及其功能说明,如表 5.38 所示。

表 5.38 /etc/fstab 文件字段及其功能说明

字段	功能说明
第 1 列	需要挂载的设备文件名
第 2 列	挂载点,必须是一个目录名且必须使用绝对路径
第 3 列	文件系统类型,可以设置为 auto,即由系统自动检测
第 4 列	挂载参数,一般采用 defaults,还可以设置 rw,suid,dev,exec,auto 等参数
第 5 列	能否被 dump 备份。dump 是一个用来备份的命令,这个字段的取值通常为 0 或者 1(0 表示忽略,1 表示需要)
第 6 列	是否检验扇区。在开机的过程中,系统默认以 fsck 命令检验系统是否完整



② 编辑/etc/fstab 文件,在文件尾部添加一行命令,执行命令如下。

```
[root@localhost ~]# vim /etc/fstab
# /etc/fstab
...
/dev/mapper /CentOS - root /          xfs  defaults  0 0
UUID= 6d58086e - 0a6b - 4399 - 93dc - c2016ea17fe0 /boot  xfs  defaults  0 0
/dev/mapper /CentOS - swap swap      swap  defaults  0 0
/dev/sr0 /media auto defaults      0 0
~
"/etc/fstab" 12L, 515C 已写入
[root@localhost ~]# mount -a           //自动挂载系统中的所有文件系统
mount: /dev/sr0 写保护,将以只读方式挂载
[root@localhost ~]#
```

也可以使用以下命令修改文件的内容。

```
# echo "/dev/sr0 /media iso9660 defaults 0 0" >> /etc/fstab
# mount -a
```

③ 结果测试,重启系统,显示分区挂载情况,执行命令如下。

```
[root@localhost ~]# reboot
[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     36G   5.3G   31G   13%    /
...
/dev/sr0                 iso9660  5.3G   5.3G   0     100%   /media
...
```



(5) 卸载文件系统。

umount 命令用于卸载一个已经挂载的文件系统(分区),相当于 Windows 操作系统中的弹出设备。其命令格式如下。

```
umount [选项] <源> | <目录>
```

umount 命令各选项及其功能说明,如表 5.39 所示。

表 5.39 umount 命令各选项及其功能说明

选项	功能说明	选项	功能说明
-a	卸载所有文件系统	-l	立即断开文件系统
-A	卸载当前名称空间中指定设备对应的所有挂载点	-o	限制文件系统集合(和-a 选项一起使用)
-c	不对路径进行规范化	-R	递归卸载目录及其子对象
-d	若挂载了回环设备,则释放该回环设备	-r	若卸载失败,则尝试以只读方式重新挂载
-f	强制卸载(遇到不响应的 NFS 时)	-t	限制文件系统集合
-i	不调用 umount.<类型> 辅助程序	-v	输出当前进行的操作
-n	不写 /etc/mtab		

使用 umount 命令卸载文件系统,执行命令如下。

```
[root@localhost ~]# umount /mnt/u-disk
[root@localhost ~]# umount /media/cdrom
[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       36G   5.3G   31G   13%    /
devtmpfs                 devtmpfs  5.9G   0      5.9G   0%    /dev
...
```

在使用 `umount` 命令卸载文件系统时,必须保证此时的文件系统未处于 `busy` 状态,使文件系统处于 `busy` 状态的情况有:文件系统中打开的文件,某个进程的工作目录在此文件系统中,文件系统的缓存文件正在被使用等。

## 5.2.6 逻辑卷配置与管理

逻辑卷管理器(Logical Volume Manager,LVM)是建立在磁盘分区和文件系统之间的一个逻辑层,其设计目的是实现对磁盘的动态管理。管理员利用 LVM 不用重新分区磁盘即可动态调整文件系统的大小,而且,当服务器添加新磁盘后,管理员不必将已有的磁盘文件移动到校验磁盘中,通过 LVM 即可直接跨越磁盘扩展文件系统,提供了一种非常高效灵活的磁盘管理方式。

通过 LVM,用户可以在系统运行时动态调整文件系统的大小,把数据从一块硬盘重定位到另一块硬盘中,也可以提高 I/O 操作的性能,以及提供冗余保护,它的快照功能允许用户对逻辑卷进行实时的备份。

### 1. 逻辑卷简介

早期硬盘驱动器(Device Driver)呈现给操作系统的是一组连续的物理块,整个硬盘驱动器都分配给文件系统或者其他数据体,由操作系统或应用程序使用,这样做的缺点是缺乏灵活性:当一个硬盘驱动器的空间使用完时,很难扩展文件系统的大小;而当硬盘驱动器存储容量增加时,把整个硬盘驱动器分配给文件系统又会导致无法充分利用存储空间。

用户在安装 Linux 操作系统时遇到的一个常见问题是,如何正确评估分区的大小,以分配合适的硬盘空间?普通的磁盘分区管理方式在逻辑分区划分完成之后就无法改变其大小,当一个逻辑分区存放不下某个文件时,这个文件受上层文件系统的限制,无法跨越多个分区存放,所以也不能同时存放到其他磁盘上。当某个分区空间耗尽时,解决的方法通常是使用符号链接,或者使用调整分区大小的工具,但这并没有从根本上解决问题。随着逻辑卷管理功能的出现,该问题迎刃而解,用户可以在无须停机的情况下方便地调整各个分区的大小。

对一般用户而言,使用最多的是动态调整文件系统大小的功能。这样,在分区时就不必为如何设置分区的大小而烦恼,只要在硬盘中预留部分空间,并根据系统的使用情况动态调整分区大小即可。

LVM 是在磁盘分区和文件系统之间添加一个逻辑层,为文件系统屏蔽下层磁盘分区,通过它可以将若干个磁盘分区连接为一个整块的抽象卷组,在卷组中可以任意创建逻辑卷并在逻辑卷中建立文件系统,最终在系统中挂载使用的就是逻辑卷,逻辑卷的使用方法与管理方式和普通的磁盘分区是完全一样的。LVM 磁盘组织结构,如图 5.12 所示。

LVM 中主要涉及以下几个概念。

(1) 物理存储介质(Physical Storage Media):指系统的物理存储设备,如磁盘、`/dev/sda`、

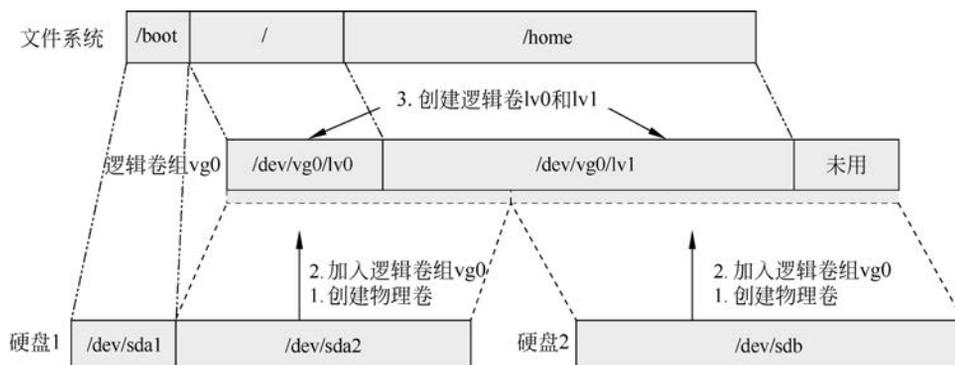


图 5.12 LVM 磁盘组织结构

/dev/had 等,是存储系统最底层的存储单元。

(2) 物理卷(Physical Volume, PV):指磁盘分区或逻辑上与磁盘分区具有同样功能的设备,是 LVM 最基本的存储逻辑块,但和基本的物理存储介质(如分区、磁盘)相比,其包含与 LVM 相关的管理参数。

(3) 卷组(Volume Group, VG):类似于非 LVM 系统中的物理磁盘,由一个或多个物理卷组成,可以在卷组中创建一个或多个逻辑卷。

(4) 逻辑卷:可以将卷组划分成若干个逻辑卷,相当于在逻辑硬盘上划分出几个逻辑分区,逻辑卷建立在卷组之上,每个逻辑分区上都可以创建具体的文件系统,如/home、/mnt 等。

(5) 物理块:每一个物理卷被划分成称为物理块的基本单元,具有唯一编号的物理块是可以被 LVM 寻址的最小单元,物理块的大小是可以配置的,默认为 4MB,物理卷由大小相同的基本单元——物理块组成。

在 CentOS 7.6 操作系统中,LVM 得到了重视,在安装系统的过程中,如果设置由系统自动进行分区,则系统除了创建一个/boot 引导分区之外,会对剩余的磁盘空间全部采用 LVM 进行管理,并在其中创建两个逻辑卷,分别挂载到/root 分区和/swap 分区中。

## 2. 配置逻辑卷

磁盘分区是实现 LVM 的前提和基础,在使用 LVM 时,需要先划分磁盘分区,再将磁盘分区的类型设置为 8e,最后才能将分区初始化为物理卷。

### (1) 创建磁盘分区。

这里使用前面安装的第二块硬盘的主分区/dev/sdb2 和逻辑分区/dev/sdb6 来进行演示,需要注意的是,要先将分区/dev/sdb2 和/dev/sdb6 卸载以便进行演示,并使用 fdisk 命令查看/dev/sdb 硬盘分区情况,执行命令如下。

```
[root@localhost ~]# fdisk -l /dev/sdb
磁盘 /dev/sdb:25.5 GB, 21474836480 字节,41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes

扇区大小(逻辑/物理):512 字节 / 512 字节
I/O 大小(最小/最佳):512 字节 / 512 字节
磁盘标签类型:dos
磁盘标识符:0x28cba55d
```

设备	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	10487807	5242880	83	Linux
/dev/sdb2		10487808	20973567	5242880	83	Linux
/dev/sdb3		20973568	41943039	10484736	5	Extended
/dev/sdb5		20975616	31461375	5242880	83	Linux
/dev/sdb6		31463424	41943039	5239808	83	Linux

在 fdisk 命令中,使用 t 选项可以更改分区的类型,如果不知道分区类型对应的 ID,则可以输入“L”来查看各分区类型对应的 ID,如图 5.13 所示。

```
命令(输入 m 获取帮助): t
分区号 (1-3,5,6, 默认 6):
Hex 代码(输入 L 列出所有代码): L

0 空                24  NEC DOS          81  Minix / 旧 Linu bf  Solaris
1  FAT12             27  隐藏的 NTFS win  82  Linux 交换 / so c1  DRDOS/sec (FAT-
2  XENIX root       39  Plan 9           83  Linux              c4  DRDOS/sec (FAT-
3  XENIX usr        3c  PartitionMagic  84  OS/2 隐藏的 C:  c6  DRDOS/sec (FAT-
4  FAT16 <32M      40  Verix 80286     85  Linux 扩展       c7  Syrix
5  扩展            41  PPC PreP Boot  86  NTFS 卷集       da  非文件系统数据
6  FAT16           42  SFS             87  NTFS 卷集       db  CP/M / CTOS / .
7  HPFS/NTFS/exFAT 4d  QNX4.x         88  Linux 纯文本    de  Dell 工具
8  AIX              4e  QNX4.x 第2部分  8e  Linux LVM        df  BootIT
9  AIX 可启动      4f  QNX4.x 第3部分  93  Amoeba          e1  DOS 访问
a  OS/2 启动管理器 50  OnTrack DM     94  Amoeba BBT      e3  DOS R/O
b  w95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  Speedstor
c  w95 FAT16 (LBA) 52  CP/M           a0  IBM Thinkpad 休 eb  BeOS fs
e  w95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         ee  GPT
f  w95 扩展 (LBA) 54  OnTrackDM6     a6  OpenBSD         ef  EFI (FAT-12/16/
10 OPUS           55  E2-Drive       a7  NextSTEP        f0  Linux/PA-RISC
11 隐藏的 FAT12    56  golden Bow    a8  Darwin UFS      f1  Speedstor
12 Compaq 诊断   5c  Priam Edisk    a9  NetBSD          f4  Speedstor
14 隐藏的 FAT16 <3 61  SpeedStor     ab  Darwin 启动    f2  DOS 次要
16 隐藏的 FAT16   63  GNU HURD or Sys af  HFS / HFS+     fb  VMware VMFS
17 隐藏的 HPFS/NTF 64  Novell Netware b7  BSDI fs         fc  VMware VMKORE
18 AST 智能睡眠   65  Novell Netware b8  BSDI swap       fd  Linux raid 自动
1b 隐藏的 w95 FAT3 70  DiskSecure 多  bb  Boot Wizard 隐 fe  LANstep
1c 隐藏的 w95 FAT3 75  PC/IX         be  Solaris 启动   ff  BBT
1e 隐藏的 w95 FAT1 80  旧 Minix
```

图 5.13 查看各分区类型对应的 ID

下面将 /dev/sdb2 和 /dev/sdb6 的分区类型更改为 Linux LVM,即将分区的 ID 修改为 8e,如图 5.14 所示。分区创建成功后要保存分区表,重启系统或使用 partprobe/dev/sdb 命令即可。

```
[root@localhost ~]# fdisk /dev/sdb
欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中,直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 m 获取帮助): t
分区号 (1-3,5,6, 默认 6): 2
Hex 代码(输入 L 列出所有代码): 8e
已将分区“Linux”的类型更改为“Linux LVM”

命令(输入 m 获取帮助): t
分区号 (1-3,5,6, 默认 6): 6
Hex 代码(输入 L 列出所有代码): 8e
已将分区“Linux”的类型更改为“Linux LVM”

命令(输入 m 获取帮助): p

磁盘 /dev/sdb: 21.5 GB, 21474836480 字节, 41943040 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x28cba55d

 设备 Boot      Start       End         Blocks  Id System
/dev/sdb1                2048     10487807     5242880  83  Linux
/dev/sdb2                10487808  20973567     5242880  8e  Linux LVM
/dev/sdb3                20973568  41943039    10484736  5   Extended
/dev/sdb5                20975616  31461375     5242880  83  Linux
/dev/sdb6                31463424  41943039     5239808  8e  Linux LVM

命令(输入 m 获取帮助): w
The partition table has been altered!

calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: 设备或资源忙。
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
正在同步磁盘。
[root@localhost ~]# partprobe /dev/sdb
```

图 5.14 更改分区类型

## (2) 创建物理卷。

pvcreate 命令用于将物理硬盘分区初始化为物理卷,以便 LVM 使用。其命令格式如下。

```
pvcreate [选项] [参数]
```

pvcreate 命令各选项及其功能说明,如表 5.40 所示。

表 5.40 pvcreate 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	强制创建物理卷,不需要用户确认	-y	所有问题都回答 yes
-u	指定设备的 UUID	-Z	是否利用前 4 个扇区

将/dev/sdb2 和/dev/sdb6 分区转换为物理卷,执行相关命令,如图 5.15 所示。

```
[root@localhost ~]# pvcreate /dev/sdb2 /dev/sdb6
WARNING: ext4 signature detected on /dev/sdb2 at offset 1080. wipe it? [y/n]: y
Wiping ext4 signature on /dev/sdb2.
WARNING: swap signature detected on /dev/sdb6 at offset 4086. wipe it? [y/n]: y
Wiping swap signature on /dev/sdb6.
Physical volume "/dev/sdb2" successfully created.
Physical volume "/dev/sdb6" successfully created.
[root@localhost ~]# pvcreate -y /dev/sdb2 /dev/sdb6
Physical volume "/dev/sdb2" successfully created.
Physical volume "/dev/sdb6" successfully created.
```

图 5.15 将分区转换为物理卷

pvscan 命令会扫描系统中连接的所有硬盘,并列出找到的物理卷列表。其命令格式如下。

```
pvscan [选项] [参数]
```

pvscan 命令各选项及其功能说明,如表 5.41 所示。

表 5.41 pvscan 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-d	调试模式	-u	显示 UUID
-n	仅显示不属于任何卷组的物理卷	-e	仅显示属于输出卷组的物理卷
-s	以短格式输出		

使用 pvscan 命令扫描系统中连接的所有硬盘,并列出找到的物理卷列表,执行命令如下。

```
[root@localhost ~]# pvscan -s
/dev/sda2
/dev/sdb6
/dev/sdb2
Total: 3 [48.99 GiB] / in use: 1 [<39.00 GiB] / in no VG: 2 [<10.00 GiB]
```

## (3) 创建卷组。

卷组设备文件在创建卷组时自动生成,位于/dev 目录下,与卷组同名,卷组中的所有逻辑设备文件都保存在该目录下,卷组中可以包含一个或多个物理卷。vgcreate 命令用于创建 LVM 卷组。其命令格式如下。

```
vgcreate [选项] 卷组名 物理卷名 [物理卷名 ...]
```

vgcreate 命令各选项及其功能说明,如表 5.42 所示。

表 5.42 vgcreate 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-l	卷组中允许创建的最大逻辑卷数	-s	卷组中的物理卷的大小,默认值为 4MB
-p	卷组中允许添加的最大物理卷数		

vgdisplay 命令用于显示 LVM 卷组的信息,如果不指定卷组参数,则分别显示所有卷组的属性。其命令格式如下。

```
vgdisplay [选项] [卷组名]
```

vgdisplay 命令各选项及其功能说明如表 5.43 所示。

表 5.43 vgdisplay 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-A	仅显示活动卷组的属性	-s	使用短格式输出信息

为物理卷/dev/sdb2 和/dev/sdb6 创建名为 vg-group01 的卷组并查看相关信息,如图 5.16 所示。

```
[root@localhost ~]# vgcreate vg-group01 /dev/sdb2 /dev/sdb6
Volume group "vg-group01" successfully created
[root@localhost ~]# vgdisplay vg-group01
--- Volume group ---
VG Name                vg-group01
System ID
Format                 lvm2
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                9.99 GiB
PE Size                4.00 MiB
Total PE              2558
Alloc PE / Size       0 / 0
Free PE / Size        2558 / 9.99 GiB
VG UUID                gAgSQj-bfy2-BFus-vDuk-fCCM-COSQ-OTufIr
```

图 5.16 创建卷组并查看相关信息

#### (4) 创建逻辑卷。

lvcreate 命令用于创建 LVM 逻辑卷,逻辑卷是创建在卷组之上的,逻辑卷对应的设备文件保存在卷组目录下。其命令格式如下。

```
lvcreate [选项] 逻辑卷名 卷组名
```

lvcreate 命令各选项及其功能说明,如表 5.44 所示。

表 5.44 lvcreate 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-L	指定逻辑卷的大小,单位为“kKmMgGtT”字节	-n	后接逻辑卷名
-l	指定逻辑卷的大小(LE 数)	-s	创建快照

lvdisplay 命令用于显示 LVM 逻辑卷空间大小、读写状态和快照信息等属性,如果省略逻辑卷参数,则 lvdisplay 命令显示所有的逻辑卷属性;否则仅显示指定的逻辑卷属性。其命令格式如下。

```
lvdisplay [选项] 逻辑卷名
```

lvdisplay 命令各选项及其功能说明如表 5.45 所示。

表 5.45 lvdisplay 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-C	以列的形式显示	-h	显示帮助信息

从 vg-group01 卷组中创建名为 databackup、容量为 8GB 的逻辑卷,并使用 lvdisplay 命令查看逻辑卷的详细信息,如图 5.17 所示。

```
[root@localhost ~]# lvcreate -L 8G -n databackup vg-group01
Logical volume "databackup" created.
[root@localhost ~]# lvdisplay /dev/vg-group01/databackup
--- Logical volume ---
LV Path                /dev/vg-group01/databackup
LV Name                databackup
VG Name                vg-group01
LV UUID                mBCred-8rMg-JqZn-f7ys-1aOI-gjGu-CH7c7b
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2020-08-26 18:33:20 +0800
LV Status              available
# open                 0
LV Size                8.00 GiB
Current LE             2048
Segments               2
Allocation              inherit
Read ahead sectors     auto
- currently set to    8192
Block device           253:2
```

图 5.17 创建逻辑卷并查看逻辑卷的详细信息

创建逻辑卷 databackup 后,查看 vg-group01 卷组的详细信息,如图 5.18 所示,可以看到 vg-group01 卷组还有 5.99GB 的空闲空间。

```
[root@localhost ~]# vgdisplay vg-group01
--- Volume group ---
VG Name                vg-group01
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No  2
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV                0
Max PV                 0
Cur PV                2
Act PV                2
VG Size                9.99 GiB
PE Size                4.00 MiB
Total PE               2558
Alloc PE / Size        2048 / 8.00 GiB
Free PE / Size         510 / 1.99 GiB
VG UUID                gAgSQj-bfY2-BFus-vDuk-fCCM-COSQ-OTufIr
```

图 5.18 查看 vg-group01 卷组的详细信息

(5) 创建并挂载文件系统。

逻辑卷相当于一个磁盘分区,使用逻辑卷需要进行格式化和挂载。

对逻辑卷/dev/vg-group01/databackup 进行格式化,如图 5.19 所示。

创建挂载点目录,对逻辑卷进行手动挂载或者修改/etc/fstab 文件进行自动挂载,挂载后即可使用,如图 5.20 所示。

```
[root@localhost ~]# mkfs.ext4 /dev/vg-group01/databackup
mke2fs 1.42.9 (28-Dec-2013)
文件系统标签=
OS type: Linux
块大小=4096 (log=2)
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2097152 blocks
104857 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: 完成
正在写入inode表: 完成
Creating journal (32768 blocks): 完成
writing superblocks and filesystem accounting information: 完成
```

图 5.19 对逻辑卷进行格式化

```
[root@localhost ~]# mkdir /mnt/backup-data
[root@localhost ~]# mount /dev/vg-group01/databackup /mnt/backup-data
[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	36G	14G	22G	39%	/
devtmpfs	devtmpfs	1.9G	0	1.9G	0%	/dev
tmpfs	tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	tmpfs	1.9G	13M	1.9G	1%	/run
tmpfs	tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/sr0	iso9660	4.3G	4.3G	0	100%	/media/cdrom
/dev/sda1	xfs	1014M	179M	836M	18%	/boot
tmpfs	tmpfs	378M	0	378M	0%	/run/user/0
tmpfs	tmpfs	378M	12K	378M	1%	/run/user/42
/dev/sdb5	ext4	4.8G	20M	4.6G	1%	/mnt/data05
/dev/sdb1	ext4	4.8G	20M	4.6G	1%	/mnt/data01
/dev/mapper/vg--group01-databackup	ext4	7.8G	36M	7.3G	1%	/mnt/backup-data

图 5.20 挂载并使用逻辑卷

### 3. 管理逻辑卷

逻辑卷创建完成以后,可以根据需要对其进行各种管理操作,如扩展、缩减和删除等。

(1) 增加新的物理卷到卷组中。

vgextend 命令用于动态扩展 LVM 卷组,它通过向卷组中添加物理卷来增加组的容量,LVM 卷组中的物理卷可以在使用 vgcreate 命令创建卷组时添加,也可以使用 vgextend 命令动态添加。其命令格式如下。

```
vgextend [选项] [卷组名] [物理卷路径]
```

vgextend 命令各选项及其功能说明如表 5.46 所示。

表 5.46 vgextend 命令各选项及其功能说明

选项	功能说明	选项	功能说明
-d	调试模式	-h	显示命令的帮助信息
-f	强制扩展卷组	-v	显示详细信息

(2) 从卷组中删除物理卷。

vgreduce 命令通过删除 LVM 卷组中的物理卷来减少卷组容量。其命令格式如下。

```
vgreduce [选项] [卷组名] [物理卷路径]
```

vgreduce 命令各选项及其功能说明如表 5.47 所示。

表 5.47 vgreduce 命令各选项及其功能说明

选 项	功 能 说 明
-a	如果没有指定要删除的物理卷,则删除所有空的物理卷
--removemissing	删除卷组中所有丢失的物理卷,使卷组恢复正常状态

(3) 减少逻辑卷空间。

lvreduce 命令用于减少 LVM 逻辑卷占用的空间。使用 lvreduce 命令收缩逻辑卷的空间时有可能会删除逻辑卷中已有的数据,所以在操作前必须进行确认。其命令格式如下。

```
lvreduce [选项] [参数]
```

lvreduce 命令各选项及其功能说明如表 5.48 所示。

表 5.48 lvreduce 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-L	指定逻辑卷的大小,单位为“kKmMgGtT”字节	-l	指定逻辑卷的大小(LE 数)

(4) 增加逻辑卷空间。

lvextend 命令用于动态地扩展逻辑卷的空间,而不中断应用程序对逻辑卷的访问。其命令格式如下。

```
lvextend [选项] [逻辑卷路径]
```

lvextend 命令各选项及其功能说明如表 5.49 所示。

表 5.49 lvextend 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	强制扩展逻辑卷空间	-l	指定逻辑卷的大小(LE 数)
-L	指定逻辑卷的大小,单位为“kKmMgGtT”字节	-r	重置文件系统使用的空间,单位为“kKmMgGtT”字节

(5) 更改卷组的属性。

vgchange 命令用于修改卷组的属性,可以设置卷组处于活动状态或非活动状态。其命令格式如下。

```
vgchange [选项] [卷组名]
```

vgchange 命令各选项及其功能说明如表 5.50 所示。

表 5.50 vgchange 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-a	设置卷组中的逻辑卷的可用性	-s	更改该卷组的物理卷大小
-L	更改现有不活动卷组的最大物理卷数量	-x	启用或禁用在此卷组中扩展/减少物理卷
-l	更改现有不活动卷组的最大逻辑卷数量		

(6) 删除逻辑卷。

lvremove 命令用于删除指定的逻辑卷。其命令格式如下。

```
lvremove [选项] [逻辑卷路径]
```

lvremove 命令各选项及其功能说明如表 5.51 所示。

表 5.51 lvremove 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	强制删除	-noudevsync	禁用 Udev 同步

(7) 创建卷组。

vgremove 命令用于删除指定的卷组。其命令格式如下。

```
vgremove [选项] [卷组名]
```

vgremove 命令各选项及其功能说明如表 5.52 所示。

表 5.52 vgremove 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	强制删除	-v	显示详细信息

(8) 删除物理卷。

pvremove 命令用于删除指定的物理卷。其命令格式如下。

```
pvremove [选项] [物理卷]
```

pvremove 命令各选项及其功能说明如表 5.53 所示。

表 5.53 pvremove 命令各选项及其功能说明

选 项	功 能 说 明	选 项	功 能 说 明
-f	强制删除	-y	所有问题都回答 yes

需要注意的是,当在现实生产环境中部署 LVM 时,要先创建物理卷、卷组、逻辑卷,再创建并挂载文件系统。当想重新部署 LVM 或不再需要使用 LVM 时,需要进行 LVM 的删除操作,其过程正好与创建 LVM 的过程相反,为此,需要提前备份好重要的数据信息,并依次卸载文件系统,删除逻辑卷、卷组、物理卷设备,这个顺序不可有误。

## 5.2.7 RAID 基础知识

独立磁盘冗余阵列(Redundant Arrays of Independent Disks, RAID)通常简称为磁盘阵列。简单地说,RAID 是由多个独立的高性能磁盘驱动器组成的磁盘子系统,提供了比单个磁盘更高的存储性能和数据冗余技术。

### 1. RAID 中的关键概念和技术

(1) 镜像。

镜像是一种冗余技术,为磁盘提供了保护功能,以防止磁盘发生故障而造成数据丢失。对于

RAID 而言,采用镜像技术将会同时在阵列中产生两个完全相同的数据副本,分布在两个不同的磁盘驱动器组中。镜像提供了完全的数据冗余能力,当一个数据副本失效不可用时,外部系统仍可正常访问另一个副本,不会对应用系统的运行和性能产生影响。此外,镜像不需要额外的计算和校验,用于修复故障非常快,直接复制即可。镜像技术可以从多个副本并发读取数据,提供了更高的读取性能,但不能并行写数据,写多个副本时会导致一定的 I/O 性能降低。

### (2) 数据条带。

磁盘存储的性能瓶颈在于磁头寻道定位,它是一种慢速机械运动,无法与高速的 CPU 匹配。再者,单个磁盘驱动器性能存在物理极限,I/O 性能非常有限。RAID 由多块磁盘组成,数据条带技术将数据以块的方式分布存储在多个磁盘中,从而可以对数据进行并发处理。这样写入和读取数据即可在多个磁盘中同时进行,并发产生非常高的聚合 I/O,有效地提高整体 I/O 性能,且具有良好的线性扩展性。这在对大容量数据进行处理时效果尤其显著,如果不分块,则数据只能先按顺序存储在磁盘阵列的磁盘中,需要时再按顺序读取。而通过条带技术,可获得数倍于顺序访问的性能提升。

### (3) 数据校验。

镜像具有安全性高、读取性能高的特点,但冗余开销太大。数据条带通过并发性大幅提高了性能,但未考虑数据安全性、可靠性。数据校验是一种冗余技术,它以校验数据提供数据的安全性,可以检测数据错误,并在能力允许的前提下进行数据重构。相对于镜像,数据校验大幅缩减了冗余开销,用较小的代价换取了极佳的数据完整性和可靠性。数据条带技术提供了性能,数据校验提供了数据安全性,不同等级的 RAID 往往同时结合使用这两种技术。

采用数据校验时,RAID 要在写入数据的同时进行校验计算,并将得到的校验数据存储在 RAID 成员磁盘中。校验数据可以集中保存在某个磁盘或分散存储在多个磁盘中,校验数据也可以分块,不同 RAID 等级的实现各不相同。当其中一部分数据出错时,就可以对剩余数据和校验数据进行反校验计算以重建丢失的数据。相对于镜像技术而言,校验技术节省了大量开销,但由于每次数据读写都要进行大量的校验运算,因此对计算机的运算速度要求很高,必须使用硬件 RAID 控制器。在数据重建恢复方面,校验技术比镜像技术复杂得多且速度慢得多。

## 2. 常见的 RAID 类型

### (1) RAID0。

RAID0 会把连续的数据分散到多个磁盘中进行存取,系统有数据请求时可以被多个磁盘并行执行,每个磁盘执行属于自己的那一部分数据请求。如果要做 RAID0,则一台服务器至少需要两块硬盘,其读写速度是一块硬盘的两倍。如果有  $N$  块硬盘,则其读写速度是一块硬盘的  $N$  倍。虽然 RAID0 的读写速度可以提高,但是由于没有数据备份功能,因此安全性会低很多。如图 5.21 所示为 RAID0 技术结构示意图。

RAID0 技术的优缺点分别如下。

优点:充分利用 I/O 总线性能,使其带宽翻倍,读写速度翻倍;充分利用磁盘空间,利用率为 100%。

缺点:不提供数据冗余;无数据校验,无法保证数据的正确性;存在单点故障。

应用场景:对数据完整性要求不高的场景,如日志存储、个人娱乐;对读写效率要求高,而对安全性能要求不高的场景,如图像工作站。

## (2) RAID1。

RAID1 会通过磁盘数据镜像实现数据冗余,在成对的独立磁盘中产生互为备份的数据。当原始数据繁忙时,可直接从镜像副本中读取数据。同样地,要做 RAID1 至少需要两块硬盘,当读取数据时,其中一块硬盘会被读取,另一块硬盘会被用作备份。其数据安全性较高,但是磁盘空间利用率较低,只有 50%。如图 5.22 所示为 RAID1 技术结构示意图。

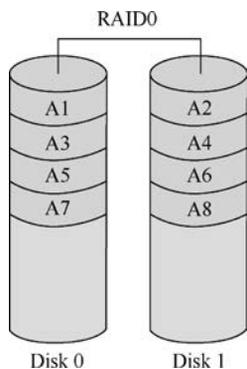


图 5.21 RAID0 技术结构示意图

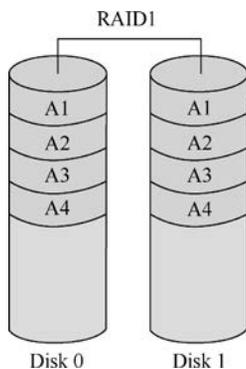


图 5.22 RAID1 技术结构示意图

RAID1 技术的优缺点如下。

优点：提供了数据冗余,数据双倍存储;提供了良好的读取性能。

缺点：无数据校验;磁盘利用率低,成本高。

应用场景：存放重要数据的场景,如数据存储领域。

## (3) RAID5。

RAID5 应该是目前最常见的 RAID 等级,具备很好的扩展性。当阵列磁盘数量增加时,并行操作的能力随之增加,可支持更多的磁盘,从而拥有更高的容量及更高的性能。RAID5 的磁盘可同时存储数据和校验数据,数据块和对应的校验信息保存在不同的磁盘中,当一个数据盘损坏时,系统可以根据同一条带的其他数据块和对应的校验数据来重建损坏的数据。与其他 RAID 等级一样,重建数据时,RAID5 的性能会受到较大的影响。

RAID5 兼顾了存储性能、数据安全和存储成本等各方面因素,基本上可以满足大部分的存储应用需求,数据中心大多采用它作为应用数据的保护方案。RAID0 大幅提升了设备的读写性能,但不具备容错能力;RAID1 虽然十分注重数据安全,但是磁盘利用率太低。RAID5 可以理解成 RAID0 和 RAID1 的折中方案,是目前综合性能最好的数据保护解决方案,一般而言,中小企业会采用 RAID5,大企业会采用 RAID10。如图 5.23 所示为 RAID5 技术结构示意图。

RAID5 技术的优缺点如下。

优点：读写性能高;有校验机制;磁盘空间利用率高。

缺点：磁盘越多,安全性能越差。

应用场景：对安全性能要求高的场景,如金融、数据库、存储等。

## (4) RAID01。

RAID01 是先做条带化再做镜像,本质是对物理磁盘实现镜像;而 RAID10 是先做镜像再做条带化,本质是对虚拟磁盘实现镜像。相同的配置下,RAID01 比 RAID10 具有更好的容错能力。

RAID01 的数据将同时写入到两个磁盘阵列中,如果其中一个阵列损坏,则其仍可继续工作,

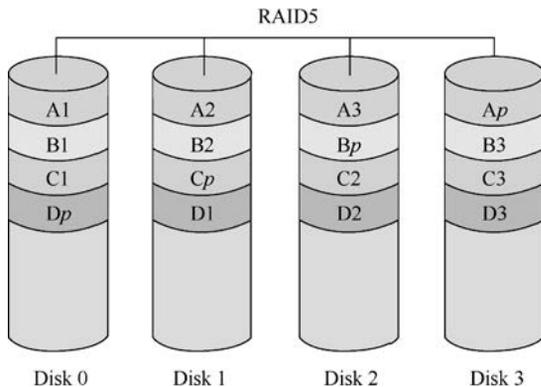


图 5.23 RAID5 技术结构示意图

在保证数据安全性的同时提高了性能。RAID01 和 RAID10 内部都含有 RAID1 模式,因此整体磁盘利用率仅为 50%。如图 5.24 所示为 RAID01 技术结构示意图。

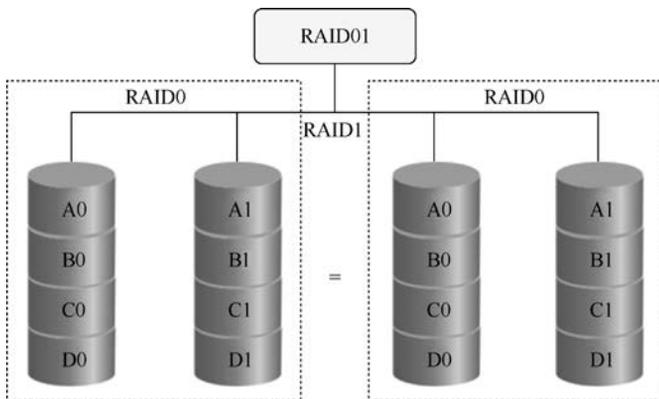


图 5.24 RAID01 技术结构示意图

RAID01 技术的优缺点如下。

优点: 提供了较高的 I/O 性能; 有数据冗余; 无单点故障。

缺点: 成本稍高; 安全性能比 RAID10 差。

应用场景: 特别适用于既有大量数据需要存取, 又对数据安全性要求严格的领域, 如银行、金融、商业超市、仓储库房、档案管理等。

(5) RAID10。

如图 5.25 所示为 RAID10 技术结构示意图。

RAID10 技术的优缺点如下。

优点: RAID10 的读取性能优于 RAID01; 提供了较高的 I/O 性能; 有数据冗余; 无单点故障; 安全性能高。

缺点: 成本稍高。

应用场景: 特别适用于既有大量数据需要存取, 又对数据安全性要求严格的领域, 如银行、金融、商业超市、仓储库房、档案管理等。

(6) RAID50。

RAID50 具有 RAID5 和 RAID0 的共同特性。它由两组 RAID5 磁盘组成(其中, 每组最少有 3

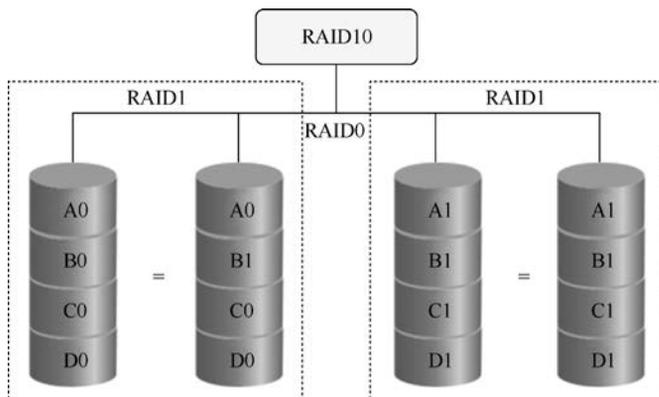


图 5.25 RAID10 技术结构示意图

个磁盘),每一组都使用了分布式奇偶位;而两组 RAID5 磁盘再组建成 RAID0,实现跨磁盘数据读取。RAID50 提供了可靠的数据存储和优秀的整体性能,并支持更大的卷尺寸。即使两个物理磁盘(每个阵列中的一个)发生故障,数据也可以顺利恢复。RAID50 最少需要 6 个磁盘,其适用于高可靠性存储、高读取速度、高数据传输性能的应用场景,包括事务处理和有许多用户存取小文件的办公应用程序。如图 5.26 所示为 RAID50 技术结构示意图。

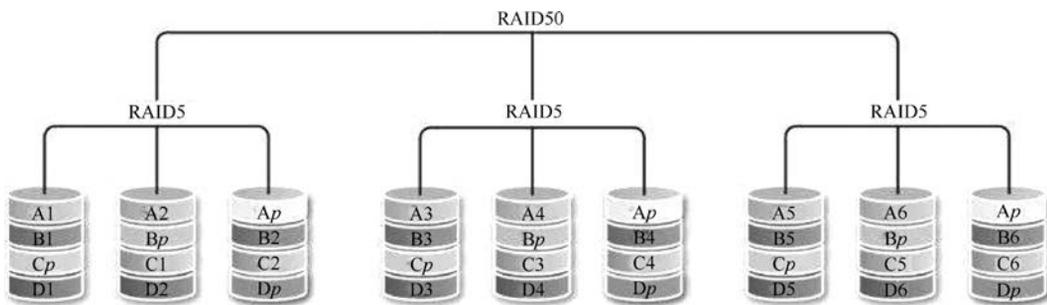


图 5.26 RAID50 技术结构示意图

## 5.2.8 网络配置管理

Linux 主机要想与网络中的其他主机进行通信,必须进行正确的网络配置,网络配置通常包括主机名、IP 地址、子网掩码、默认网关、DNS 服务器等的配置。

### 1. 网卡配置文件

网卡 IP 地址配置得是否正确决定了服务器能否相互通信,在 Linux 操作系统中,一切都是文件,因此配置网络服务其实就是编辑网卡配置文件。

在 CentOS 7 以前,网卡配置文件的前缀为 eth,第 1 块网卡为 eth0,第 2 块网卡为 eth1,以此类推;而在 CentOS 7.6 中,网卡配置文件的前缀是 ifcfg,其后为网卡接口名称,它们共同组成了网卡配置文件的名称,如 ifcfg-ens33。

CentOS 7.6 操作系统中的网卡配置文件为 /etc/sysconfig/network-scripts/ifcfg-<iface>,其中,iface 为网卡接口名称,本书中是 ens33,网卡配置文件的语法格式,如表 5.54 所示。



表 5.54 网卡配置文件的语法格式

选 项	功 能 说 明	默 认 值	可 选 值
TYPE	网络类型	Ethernet	Ethernet, Wireless, TeamPort, Team, VLAN
PROXY_METHOD	代理配置的方法	none	none, auto
BROWSER_ONLY	代理配置是否仅用于浏览器	no	no, yes
BOOTPROTO	网卡获取 IP 地址的方式	dhcp	none, dhcp, static, shared, ibft, autoip
DEFROUTE	即 default route, 是否将此设备设置为默认路由	yes	no, yes
IPV4_FAILURE_FATAL	如果 IPv4 配置失败, 是否禁用设备	no	no, yes
IPV6INIT	是否启用 IPv6 的接口	yes	no, yes
IPV6_AUTOCONF	如果 IPv6 配置失败, 则是否禁用设备	yes	no, yes
IPV6_DEFROUTE	如果 IPv6 配置失败, 则是否禁用设备	yes	no, yes
IPV6_FAILURE_FATAL	如果 IPv6 配置失败, 则是否禁用设备	no	no, yes
IPV6_ADDR_GEN_MODE	生成 IPv6 地址的方式	stable-privacy	eui64, stable-privacy
NAME	网络连接的名称		
UUID	用来标识网卡的唯一识别码		
DEVICE	网卡的名称	ens33	
ONBOOT	在开机或重启网卡时是否启动网卡	no	no, yes
HWADDR	硬件 MAC 地址		
IPADDR	IP 地址		
NETMASK	子网掩码		
PREFIX	网络前缀		
GATEWAY	网关		
DNS{1,2}	域名解析器		

配置网络 IP 地址, 并查看相关信息。

(1) 打开 Linux 操作系统终端窗口, 使用 `ifconfig` 或 `ip address` 命令, 可以查看本地 IP 地址, 如图 5.27 所示。

(2) 编辑网络配置文件 `/etc/sysconfig/network-scripts/ifcfg-ens33`, 如图 5.28 所示。

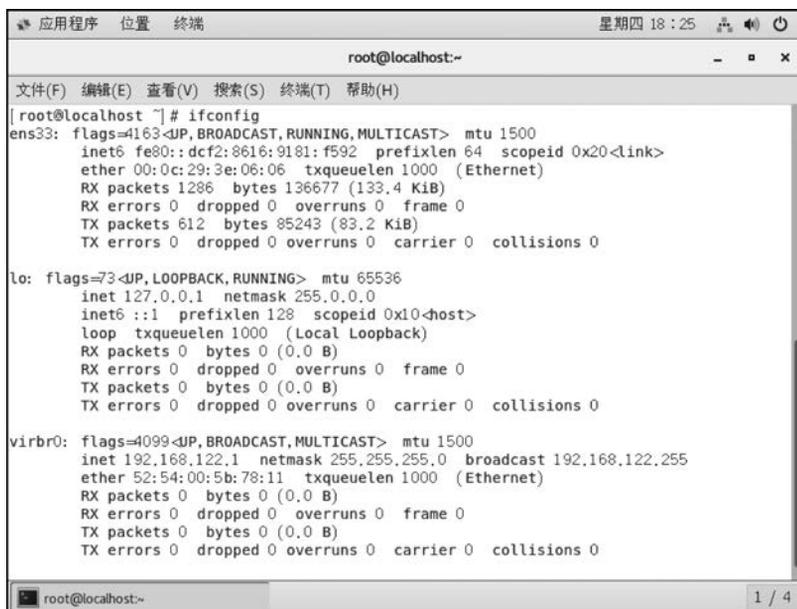
(3) 修改网卡配置文件的内容, 如图 5.29 所示。使用 Vim 编辑器进行配置内容的修改, 相关修改内容如下。

```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33
修改选项:
BOOTPROTO = dhcp ---> static           //将 DHCP 配置为静态
ONBOOT = no ---> yes                   //是否激活网卡, 配置为激活网卡
增加选项:
IPADDR = 195.168.100.100                //配置 IP 地址
```

```

PREFIX = 24 或 NETMASK = 255.255.255.0 //配置网络子网掩码
GATEWAY = 195.168.100.2 //配置网关
DNS1 = 8.8.8.8 //配置 DNS 地址
[root@localhost ~]# systemctl restart network //重启网络服务

```



```

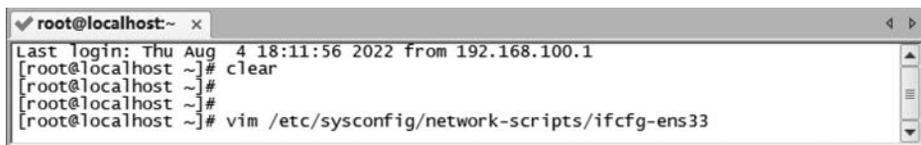
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::dcf2:8616:9181:f592 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3e:06:06 txqueuelen 1000 (Ethernet)
    RX packets 1286 bytes 136677 (133.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 612 bytes 85243 (83.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:5b:78:11 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

图 5.27 查看本地 IP 地址

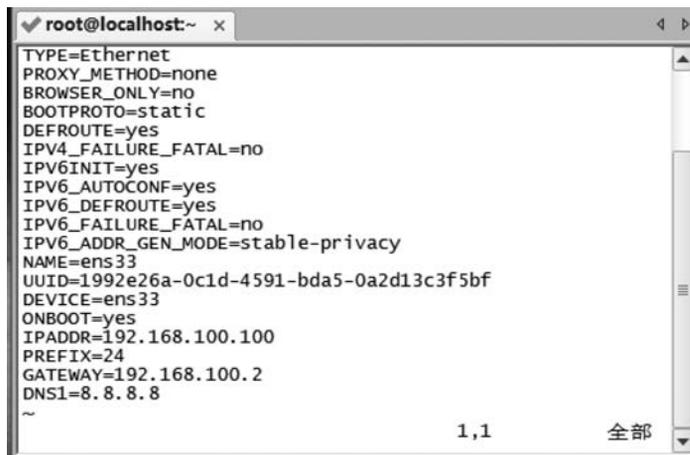


```

Last login: Thu Aug 4 18:11:56 2022 from 192.168.100.1
[root@localhost ~]# clear
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33

```

图 5.28 编辑网络配置文件



```

TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
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=ens33
UUID=1992e26a-0c1d-4591-bda5-0a2d13c3f5bf
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.100.100
PREFIX=24
GATEWAY=192.168.100.2
DNS1=8.8.8.8
~
1,1 全部

```

图 5.29 修改网卡配置文件的内容

(4) 使用 ifconfig 命令,查看网络配置结果,如图 5.30 所示。

```

[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.100 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::dcf2:8616:9181:f592 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3e:06:06 txqueuelen 1000 (Ethernet)
    RX packets 1049 bytes 116058 (113.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 461 bytes 69371 (67.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:5b:78:11 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]#

```

图 5.30 查看网络配置结果

## 2. 主机名配置文件与主机名解析配置文件

```

[root@localhost ~]# cat /etc/hostname
localhost.localdomain
[root@localhost ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@localhost ~]#

```

该文件只有一行,记录了本机的主机名,即用户在安装 CentOS 7.6 时指定的主机名,用户可以直接对其进行修改。

**注意:** 直接修改/etc/hostname 中的主机名时,应同时修改/etc/hosts 文件的内容。

## 3. 域名解析服务器配置文件

```

[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search csg.com //定义域名的搜索列表
nameserver 8.8.8.8 //定义 DNS 服务器的 IP 地址
[root@localhost ~]#

```

该文件的主要作用是定义 DNS 服务器,可根据网络的具体情况进行设置,它的格式很简单,每一行有一个关键字开关,后接配置参数,可以设置多个 DNS 服务器。

- (1) nameserver: 定义 DNS 服务器的 IP 地址。
- (2) domain: 定义本地域名。
- (3) search: 定义域名的搜索列表。
- (4) sortlist: 对返回的域名进行排序。

## 4. 网络常用管理命令

网络常用管理命令如下。



(1) `ifconfig` 命令管理网络接口。

`ifconfig` 命令是一个可以用来查看、配置、启用或禁用网络接口的命令。`ifconfig` 命令可以临时性地配置网卡的 IP 地址、子网掩码、网关等,使用 `ifconfig` 命令配置的网络相关信息,在主机重启后就不再存在,若需要使其永久有效,则可以将其保存在 `/etc/sysconfig/network-scripts/ifcfg-ens33` 文件中。其命令格式如下。

```
ifconfig [网络设备] [选项]
```

`ifconfig` 命令各选项及其功能说明如表 5.55 所示。

表 5.55 `ifconfig` 命令各选项及其功能说明

选 项	功 能 说 明
<code>up</code>	启动指定网络设备/网卡
<code>down</code>	关闭指定网络设备/网卡
<code>-arp</code>	设置指定网卡是否支持 ARP
<code>-promisc</code>	设置是否支持网卡的 promiscuous 模式,如果选择此参数,则网卡将接收网络中发送给它的所有数据包
<code>-allmulti</code>	设置是否支持多播模式,如果选择此参数,则网卡将接收网络中所有的多播数据包
<code>-a</code>	显示全部接口信息
<code>-s</code>	显示摘要信息
<code>add</code>	给指定网卡配置 IPv6 地址
<code>del</code>	删除给指定网卡配置的 IPv6 地址
<code>network &lt;子网掩码&gt;</code>	设置网卡的子网掩码
<code>tunnel &lt;地址&gt;</code>	建立 IPv4 与 IPv6 之间的隧道通信地址
<code>-broadcast &lt;地址&gt;</code>	为指定网卡设置广播协议
<code>-pointtopoint &lt;地址&gt;</code>	为网卡设置点对点通信协议

① 使用 `ifconfig` 命令配置网卡相关信息,显示 `ens33` 的网卡信息,执行命令如下。

```
[root@localhost ~]# ifconfig ens33
```

执行命令结果如图 5.31 所示。

```
[root@localhost ~]# ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.100 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::dcf2:8616:9181:f592 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3e:06:06 txqueuelen 1000 (Ethernet)
    RX packets 370 bytes 32078 (31.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 253 bytes 30517 (29.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

图 5.31 显示 `ens33` 的网卡信息

② 启动和关闭网卡,执行命令如下。

```
[root@localhost ~]# ifconfig ens33 down
[root@localhost ~]# ifconfig ens33 up
```

③ 配置网络接口相关信息,添加 IPv6 地址,进行相关测试,执行命令如下。

```
[root@localhost ~]# ifconfig ens33 add 2000::1/64 //添加 IPv6 地址
[root@localhost ~]# ping -6 2000::1 //测试网络连通性
```

```

PING 2000::1(2000::1) 56 data bytes
64 bytes from 2000::1: icmp_seq= 1 ttl= 64 time= 0.085 ms
64 bytes from 2000::1: icmp_seq= 2 ttl= 64 time= 0.103 ms
64 bytes from 2000::1: icmp_seq= 3 ttl= 64 time= 0.109 ms
^C                               //Ctrl + C 退出
--- 2000::1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6013ms
rtt min/avg/max/mdev = 0.085/0.103/0.109/0.014 ms
[root@localhost ~]# reboot          //重启操作系统
Last login: Sun Aug 30 19:00:20 2020 from 195.168.100.1
[root@localhost ~]# ping -6 2000::1 //测试网络连通性
connect: 网络不可达                //网络不可达
[root@localhost ~]#
    
```

④ 配置网络接口相关信息,添加 IPv4 地址,启动与关闭 ARP 功能,进行相关测试,执行命令如下。

```

[root@localhost ~]# ifconfig ens33 195.168.100.100 netmask 255.255.255.0 broadcast
195.168.100.255 //添加 IPv4 地址,加上子网掩码和一个广播地址
[root@localhost ~]# ifconfig ens33 arp //启动 ARP 功能
[root@localhost ~]# ifconfig ens33 -arp //关闭 ARP 功能
    
```

(2) hostnamectl 命令设置并查看主机名。

使用 hostnamectl 命令可以设置并查看主机名。其命令格式如下。

```
hostnamectl [选项] [主机名]
```

hostnamectl 命令各选项及其功能说明如表 5.56 所示。

表 5.56 hostnamectl 命令各选项及其功能说明

选 项	功 能 说 明
-h, --help	显示帮助信息
--version	显示安装包的版本信息
--static	修改静态主机名,也称为内核主机名,是系统在启动时从/etc/hostname 中自动初始化的主机名
--transient	修改瞬态主机名,瞬态主机名是在系统运行时临时分配的主机名,由内核管理,例如,通过 DHCP 或 DNS 服务器分配的 localhost 就是这种形式的主机名
--pretty	修改灵活主机名,灵活主机名是允许使用特殊字符的主机名,即使用 UTF-8 格式的自由主机名,以展示给终端用户
-P, --privileged	在执行之前获得的特权
--no-ask-password	输入的密码不提示
-H, --host=[USER@]HOST	操作远程主机
status	显示当前主机名状态
set-hostname NAME	设置当前主机名
set-icon-name NAME	设置系统主机名
set-chassis NAME	为主机设置 icon 名

使用 `hostnamectl` 命令设置并查看主机名,执行命令如下。

```
[root@localhost ~]# hostnamectl status
[root@localhost ~]# hostnamectl set --hostname lnc.csg.com
[root@localhost ~]# cat /etc/hostname
[root@localhost ~]# bash
```

① 查看当前主机名,如图 5.32 所示。

```
[root@localhost ~]# hostnamectl status
  Static hostname: localhost.localdomain
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 60561763fdde420a9ff2e80a9a9704ba
        Boot ID: 881236bdaalc49b8bacf18be3b3cd269
  Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-957.el7.x86_64
  Architecture: x86-64
```

图 5.32 查看当前主机名

② 设置主机名并查看相关信息,如图 5.33 所示。

```
[root@localhost ~]# hostnamectl set-hostname lnc.csg.com
[root@localhost ~]# cat /etc/hostname
lnc.csg.com
[root@localhost ~]# bash
[root@lnc ~]# hostnamectl status
  Static hostname: lnc.csg.com
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 60561763fdde420a9ff2e80a9a9704ba
        Boot ID: be2c500f65bd483bbb12ce0fb77c5675
  Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-957.el7.x86_64
  Architecture: x86-64
```

图 5.33 设置主机名并查看相关信息

(3) `route` 命令管理路由。

`route` 命令用来显示并设置 Linux 内核中的网络路由表,`route` 命令设置的主要是静态路由,要实现两个不同子网的通信,需要一台连接两个网络的路由器或者同时位于两个网络的网关。需要注意的是,直接在命令模式下使用 `route` 命令添加的路由信息不会永久保存,主机重启之后该路由就失效了,若需要使其永久有效,则可以在 `/etc/rc.local` 中添加 `route` 命令来保存设置。其命令格式如下。

`route` [选项]

`route` 命令各选项及其功能说明如表 5.57 所示。

表 5.57 `route` 命令各选项及其功能说明

选 项	功 能 说 明
-v	详细信息模式
-A	采用指定的地址类型
-n	以数字形式代替主机名形式来显示地址
-net	路由目标为网络
-host	路由目标为主机
-F	显示内核的 FIB 选路表

续表

选项	功能说明
-C	显示内核的路由缓存
add	添加一条路由
del	删除一条路由
target	指定目标网络或主机,可以是点分十进制的 IP 地址或主机/网络名
netmask	为添加的路由指定网络掩码
gw	为发往目标网络/主机的任何分组指定网关

使用 route 命令管理路由,执行操作如下。

① 显示当前路由信息,执行命令如下。

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

执行命令结果如图 5.34 所示。

```
[root@localhost ~]# route
kernel IP routing table
destination  Gateway      Genmask      Flags Metric Ref    use iface
default     gateway      0.0.0.0      UG    100    0     0 ens33
192.168.100.0  0.0.0.0      255.255.255.0  U    100    0     0 ens33
192.168.122.0  0.0.0.0      255.255.255.0  U    0      0     0 virbr0
```

图 5.34 显示当前路由信息

② 增加一条路由,执行命令如下。

```
[root@localhost ~]# route add -net 195.168.200.0 netmask 255.255.255.0 dev ens33
[root@localhost ~]# route
```

执行命令结果如图 5.35 所示。

```
[root@localhost ~]# route add -net 192.168.200.0 netmask 255.255.255.0 dev ens33
[root@localhost ~]# route
kernel IP routing table
destination  Gateway      Genmask      Flags Metric Ref    use iface
default     gateway      0.0.0.0      UG    100    0     0 ens33
192.168.100.0  0.0.0.0      255.255.255.0  U    100    0     0 ens33
192.168.122.0  0.0.0.0      255.255.255.0  U    0      0     0 virbr0
192.168.200.0  0.0.0.0      255.255.255.0  U    0      0     0 ens33
```

图 5.35 增加一条路由

③ 屏蔽一条路由,执行命令如下。

```
[root@localhost ~]# route add -net 195.168.200.0 netmask 255.255.255.0 reject
[root@localhost ~]# route
```

执行命令结果如图 5.36 所示。

```
[root@localhost ~]# route add -net 192.168.200.0 netmask 255.255.255.0 reject
[root@localhost ~]# route
kernel IP routing table
destination  Gateway      Genmask      Flags Metric Ref    use iface
default     gateway      0.0.0.0      UG    100    0     0 ens33
192.168.100.0  0.0.0.0      255.255.255.0  U    100    0     0 ens33
192.168.122.0  0.0.0.0      255.255.255.0  U    0      0     0 virbr0
192.168.200.0  -            255.255.255.0  !    0      -     0 -
192.168.200.0  0.0.0.0      255.255.255.0  U    0      0     0 ens33
```

图 5.36 屏蔽一条路由

④ 删除一条屏蔽路由,执行命令如下。

```
[root@localhost ~]# route del -net 195.168.200.0 netmask 255.255.255.0 reject
[root@localhost ~]# route
```

执行命令结果如图 5.37 所示。

```
[root@localhost ~]# route del --net 192.168.200.0 netmask 255.255.255.0 reject
[root@localhost ~]# route
kernel IP routing table
Destination Gateway Genmask Flags Metric Ref use Iface
default gateway 0.0.0.0 UG 100 0 0 ens33
192.168.100.0 0.0.0.0 255.255.255.0 U 100 0 0 ens33
192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0
192.168.200.0 0.0.0.0 255.255.255.0 U 0 0 0 ens33
```

图 5.37 删除一条屏蔽路由

(4) ping 命令检测网络连通性。

ping 命令是 Linux 操作系统中使用非常频繁的命令,用来测试主机之间网络的连通性。ping 命令使用的是 ICMP,它发送 ICMP 回送请求消息给目标主机,ICMP 规定,目标主机必须返回 ICMP 回送应答消息给源主机,如果源主机在一定时间内收到应答,则认为主机可达,否则不可达。其命令格式如下。

```
ping [选项] [目标网络]
```

ping 命令各选项及其功能说明如表 5.58 所示。

表 5.58 ping 命令各选项及其功能说明

选 项	功 能 说 明
-c <完成次数>	设置要求回应的次数
-f	极限检测
-i <时间间隔秒数>	指定收发信息的时间间隔
-l <网络界面>	使用指定的网络界面发送数据包
-n	只输出数值
-p <范本样式>	设置填满数据包的范本样式
-q	不显示指令执行过程,但开关和结尾的相关信息除外
-r	忽略普通的路由表,直接将数据包送到远端主机上
-R	记录路由过程
-s <数据包大小>	设置数据包的大小
-t <存活数值>	设置存活数值的大小
-v	显示指令的详细执行过程

使用 ping 命令检测网络连通性,执行操作如下。

① 在 Linux 操作系统中使用不带选项的 ping 命令后,系统会一直不断地发送检测包,直到按 Ctrl+C 组合键终止,执行命令如下。

```
[root@localhost ~]# ping www.163.com
```

执行命令结果如图 5.38 所示。

```
[root@localhost ~]# ping www.163.com
PING z163ipv6.v.lnyd.cdnycn (117.161.120.41) 56(84) bytes of data:
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=1 ttl=128 time=29.7 ms
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=2 ttl=128 time=30.0 ms
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=3 ttl=128 time=29.9 ms
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=4 ttl=128 time=30.1 ms
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=5 ttl=128 time=29.9 ms
64 bytes from 117.161.120.41 (117.161.120.41): icmp_seq=6 ttl=128 time=30.0 ms
^C
--- z163ipv6.v.lnyd.cdnycn ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5059ms
rtt min/avg/max/mdev = 29.739/29.976/30.116/0.152 ms
```

图 5.38 使用不带选项的 ping 命令

② 指定回应次数和时间间隔,设置回应次数为 4 次,时间间隔为 1s,执行命令如下。

```
[root@localhost ~]# ping -c 4 -i 1 www.163.com
```

执行命令结果如图 5.39 所示。

```
[root@localhost ~]# ping -c 4 -i 1 www.163.com
PING z163ip6.v.lnyd.cdnycn.cn (117.161.120.40) 56(84) bytes of data:
64 bytes from 117.161.120.40 (117.161.120.40): icmp_seq=1 ttl=128 time=45.5 ms
64 bytes from 117.161.120.40 (117.161.120.40): icmp_seq=2 ttl=128 time=47.6 ms
64 bytes from 117.161.120.40 (117.161.120.40): icmp_seq=3 ttl=128 time=50.1 ms
64 bytes from 117.161.120.40 (117.161.120.40): icmp_seq=4 ttl=128 time=36.7 ms

--- z163ip6.v.lnyd.cdnycn.cn ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3026ms
rtt min/avg/max/mdev = 36.731/44.997/50.125/5.044 ms
```

图 5.39 指定回应次数和时间间隔

(5) netstat 命令查看网络信息。

netstat 命令是一个综合的网络状态查看命令,可以从显示的 Linux 操作系统的网络状态信息中得知整个 Linux 操作系统的网络情况,包括网络连接、路由表、接口状态、网络链路和组播成员等。其命令格式如下。

```
netstat [选项]
```

netstat 命令各选项及其功能说明,如表 5.59 所示。

表 5.59 netstat 命令各选项及其功能说明

选 项	功 能 说 明
-a,--all	显示所有连接中的端口 Socket
-A <网络类型>,--<网络类型>	列出该网络类型连接中的相关地址
-c,--continuous	持续列出网络状态
-C,--cache	显示路由器配置的缓存信息
-e,--extend	显示网络其他相关信息
-F,--fib	显示 FIB
-g,--groups	显示组播成员名单
-h,--help	在线帮助
-i,--interfaces	显示网络界面信息表单
-l,--listening	显示监控中的服务器的 Socket
-M,--masquerade	显示伪装的网络连接
-n,--numeric	直接使用 IP 地址,而不通过域名服务器
-N,--netlink,--symbolic	显示网络硬件外围设备的符号连接名称
-o,--times	显示计时器
-p,--programs	显示正在使用的 Socket 的程序识别码和程序名称
-r,--route	显示路由表
-s,--statistics	显示网络工作信息统计表
-t,--tcp	显示 TCP 的连接状况
-u,--udp	显示 UDP 的连接状况
-v,--verbose	显示指令执行过程
-V,--version	显示版本信息

使用 netstat 命令查看网络状态相关信息,执行操作如下。

① 查看网络接口列表信息,执行命令如下。

```
[root@localhost ~]# netstat -i
```

执行命令结果如图 5.40 所示。

```
[root@localhost ~]# netstat -i
Kernel Interface table
Iface MTU RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
ens33 1500 357 0 0 0 177 0 0 0 0 BMRU
lo 65536 0 0 0 0 0 0 0 0 0 LRU
virbr0 1500 0 0 0 0 0 0 0 0 0 BMU
```

图 5.40 查看网络接口列表信息

② 查看网络所有连接端口的信息,执行命令如下。

```
[root@localhost ~]# netstat -an | more
```

执行命令结果如图 5.41 所示。

```
[root@localhost ~]# netstat -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:6000 0.0.0.0:* LISTEN
tcp 0 0 192.168.122.1:53 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:631 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN
tcp 0 52 192.168.100.100:22 192.168.100.1:54202 ESTABLISHED
tcp6 0 0 :::111 :::* LISTEN
tcp6 0 0 :::6000 :::* LISTEN
tcp6 0 0 :::22 :::* LISTEN
tcp6 0 0 :::1:631 :::* LISTEN
tcp6 0 0 :::1:25 :::* LISTEN
udp 0 0 0.0.0.0:5353 0.0.0.0:* LISTEN
udp 0 0 192.168.122.1:53 0.0.0.0:* LISTEN
udp 0 0 0.0.0.0:67 0.0.0.0:* LISTEN
udp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN
udp 0 0 0.0.0.0:53376 0.0.0.0:* LISTEN
udp 0 0 127.0.0.1:323 0.0.0.0:* LISTEN
udp 0 0 0.0.0.0:799 :::* LISTEN
udp6 0 0 :::111 :::* LISTEN
udp6 0 0 :::1:323 :::* LISTEN
udp6 0 0 :::799 :::* LISTEN
raw6 0 0 :::58 :::* LISTEN
Active UNIX domain sockets (servers and established)
Proto Refcnt Flags Type State I-Node Path
unix 2 ACC ] STREAM LISTENING 55143 @/tmp/dbus-ssjvjrE
unix 2 ACC ] STREAM LISTENING 25858 /run/lvm/lvmetad.socket
unix 2 ACC ] STREAM LISTENING 55834 @/tmp/dbus-Tu4cv0Zyw
unix 2 ACC ] STREAM LISTENING 45842 /run/gssproxy.sock
unix 2 ACC ] STREAM LISTENING 54422 @/tmp/.X11-unix/X0
unix 2 ACC ] STREAM LISTENING 56086 @/tmp/.ICE-unix/10260
unix 2 ACC ] STREAM LISTENING 53813 /var/run/libvirt/libvirt-sock
unix 2 ACC ] STREAM LISTENING 53815 /var/run/libvirt/libvirt-sock-ro
unix 2 ACC ] STREAM LISTENING 53817 /var/run/libvirt/libvirt-admin-sock
--More--
```

图 5.41 查看网络所有连接端口的信息

③ 查看网络所有 TCP 端口连接信息,执行命令如下。

```
[root@localhost ~]# netstat -at
```

执行命令结果如图 5.42 所示。

```
[root@localhost ~]# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:sunrpc 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:x11 0.0.0.0:* LISTEN
tcp 0 0 localhost:localdomain 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:ssh 0.0.0.0:* LISTEN
tcp 0 0 localhost:ipp 0.0.0.0:* LISTEN
tcp 0 0 localhost:smtp 0.0.0.0:* LISTEN
tcp 0 52 localhost.localdomain:ssh 192.168.100.1:57345 ESTABLISHED
tcp6 0 0 [::]:sunrpc [::]:* LISTEN
tcp6 0 0 [::]:x11 [::]:* LISTEN
tcp6 0 0 [::]:ssh [::]:* LISTEN
tcp6 0 0 localhost:ipp [::]:* LISTEN
tcp6 0 0 localhost:smtp [::]:* LISTEN
```

图 5.42 查看网络所有 TCP 端口连接信息

④ 查看网络组播成员名单信息,执行命令如下。

```
[root@localhost ~]# netstat -g
```

```
[root@localhost ~]# netstat -g
IPv6/IPv4 Group Memberships
Interface RefCnt Group
-----
lo 1 all-systems.mcast.net
ens33 1 224.0.0.251
ens33 1 all-systems.mcast.net
virbr0 1 224.0.0.251
virbr0 1 all-systems.mcast.net
lo 1 ff02::1
lo 1 ff01::1
ens33 1 ff02::1:ff81:f592
ens33 1 ff02::1
ens33 1 ff01::1
virbr0 1 ff02::1
virbr0 1 ff01::1
virbr0-nic 1 ff02::1
virbr0-nic 1 ff01::1
```

图 5.43 查看网络组播成员名单信息

执行命令结果如图 5.43 所示。

(6) nslookup 命令 DNS 解析。

nslookup 命令是常用域名查询命令,用于查询 DNS 信息,其有两种工作模式,即交互模式和非交互模式。在交互模式下,用户可以向 DNS 服务器查询各类主机、域名信息,或者输出域名中的主机列表;在非交互模式下,用户可以针对一个主机或域名获取特定的名称或所需信息。其命令格式如下。

#### nslookup 域名

使用 nslookup 命令进行域名查询,执行操作如下。

① 在交互模式下,使用 nslookup 命令查询域名相关信息,直到用户按 Ctrl+C 组合键退出查询模式,执行命令如下。

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

执行命令结果如图 5.44 所示。

```
[root@localhost ~]# nslookup
> www.lncc.edu.cn
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
Name:   www.lncc.edu.cn
Address: 202.199.187.105
> www.163.com
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
www.163.com canonical name = www.163.com.163jiasu.com.
www.163.com.163jiasu.com canonical name = www.163.com.bsgslb.cn.
www.163.com.bsgslb.cn canonical name = z163ipv6.v.bsgslb.cn.
z163ipv6.v.bsgslb.cn canonical name = z163ipv6.v.lnyd.cdneyuan.cn.
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.40
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.37
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.38
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.41
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.35
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.36
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.39
Name:   z163ipv6.v.lnyd.cdneyuan.cn
Address: 117.161.120.34
>
```

图 5.44 在交互模式下查询域名相关信息

② 在非交互模式下,使用 nslookup 命令查询域名相关信息,执行命令如下。

```
[root@localhost ~]# nslookup www.163.com
```

执行命令结果如图 5.45 所示。

(7) traceroute 命令跟踪路由。

traceroute 命令用于追踪网络数据包的路由途径,通过 traceroute 命令可以知道源计算机到达互联网另一端的主机的路径。其命令格式如下。

```
[root@localhost ~]# nslookup www.163.com
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
www.163.com canonical name = www.163.com.163jiasu.com.
www.163.com.163jiasu.com canonical name = www.163.com.bsgs1b.cn.
www.163.com.bsgs1b.cn canonical name = z163ipv6.v.bsgs1b.cn.
z163ipv6.v.bsgs1b.cn canonical name = z163ipv6.v.lnyd.cdnycn.cn.
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.36
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.38
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.35
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.40
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.41
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.37
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.39
Name:   z163ipv6.v.lnyd.cdnycn.cn
Address: 117.161.120.34
```

图 5.45 在非交互模式下查询域名相关信息

```
traceroute [选项] [目标主机或 IP 地址]
```

traceroute 命令各选项及其功能说明如表 5.60 所示。

表 5.60 traceroute 命令各选项及其功能说明

选 项	功 能 说 明
-d	使用 Socket 层级的排错功能
-f <存活数值>	设置第一个检测数据包的存活数值的大小
-g <网关>	设置来源路由网关,最多可设置 8 个
-i <网络界面>	使用指定的网络界面发送数据包
-l	使用 ICMP 回应取代 UDP 资料信息
-m <存活数值>	设置检测数据包的最大存活数值的大小
-n	直接使用 IP 地址而非主机名称
-p <通信端口>	设置 UDP 的通信端口
-q	发送数据包检测次数
-r	忽略普通的路由表,直接将数据包送到远端主机上
-s <来源地址>	设置本地主机发送数据包的 IP 地址
-t <服务类型>	设置检测数据包的 TOS 数值
-v	显示指令的详细执行过程

使用 traceroute 命令追踪网络数据包的路由途径,执行操作如下。

① 查看本地到网易(www.163.com)的路由访问情况,直到按 Ctrl+C 组合键终止,执行命令如下。

```
[root@localhost ~]# traceroute -q 4 www.163.com
```

执行命令结果如图 5.46 所示。

```
[root@localhost ~]# traceroute -q 4 www.163.com
traceroute to www.163.com (117.161.120.38), 30 hops max, 60 byte packets
 1 gateway (192.168.100.2) 0.165 ms 0.139 ms 0.156 ms 0.116 ms
 2 * * * *
 3 * * * *
 4 * * * *
 5 * * * *
 6 * * * *
 7 * * * *
 8 * * * *
 9 * * * *^C
```

图 5.46 查看本地到网易(www.163.com)的路由访问情况

**说明:** 记录按序号从 1 开始,每个记录就是一跳,一跳表示一个网关,可以看到每行有 4 个时间,单位都是 ms,这其实就是探测数据包向每个网关发送 4 个数据包,网关响应后返回的时间,有时会看到一些行是以“\*”符号表示的,出现这样的情况,可能是因为防火墙拦截了 ICMP 的返回信息,所以得不到相关的数据包返回数据。

② 将跳数设置为 5 后,查看本地到 www.163.com 的路由访问情况,执行命令如下。

```
[root@localhost ~]# traceroute -m 5 www.163.com
```

执行命令结果如图 5.47 所示。

```
[root@localhost ~]# traceroute -m 5 www.163.com
traceroute to www.163.com (117.161.120.37), 5 hops max, 60 byte packets
 1 gateway (192.168.100.2)  0.131 ms  0.151 ms  0.129 ms
 2 * * * *
 3 * * * *
 4 * * * *
 5 * * * *
[root@localhost ~]#
```

图 5.47 设置跳数后的路由访问情况

③ 查看路由访问情况,显示 IP 地址,不查看主机名,执行命令如下。

```
[root@localhost ~]# traceroute -n www.163.com
```

执行命令结果如图 5.48 所示。

```
[root@localhost ~]# traceroute -n www.163.com
traceroute to www.163.com (117.161.120.34), 30 hops max, 60 byte packets
 1 192.168.100.2  0.208 ms  0.147 ms  0.146 ms
 2 * * * *
 3 * * * *
 4 * * * *
 5 * * * *
 6 * * * *
 7 *AC
[root@localhost ~]#
```

图 5.48 显示 IP 地址,不查看主机名

(8) ip 命令网络配置。

ip 命令是 iproute2 软件包中的一个强大的网络配置命令,用来显示或操作路由、网络设备、策略路由和隧道等,它能够替代一些传统的网络管理命令,如 ifconfig、route 等。其命令格式如下。

```
ip [选项] [操作对象] [命令] [参数]
```

ip 命令各选项及其功能说明如表 5.61 所示。

表 5.61 ip 命令各选项及其功能说明

选 项	功 能 说 明
-V, -Version	输出 IP 的版本信息并退出
-s, -stats, -statistics	输出更为详尽的信息,如果这个选项出现两次或者多次,则输出的信息会更加详尽
-f, -family	后面接协议种类,包括 inet、inet6 或 link,用于强调使用的协议种类
-4	-family inet 的简写
-6	-family inet6 的简写
-o, oneline	对每行记录都使用单行输出,换行用字符代替,如果需要使用 wc、grep 等命令处理 IP 地址的输出,则会用到这个选项
-r, -resolve	查询域名解析系统,用获得的主机名代替主机 IP 地址

ip 命令各操作对象及其功能说明如表 5.62 所示。

表 5.62 ip 命令各操作对象及其功能说明

操作对象	功能说明	操作对象	功能说明
link	网络设备	rule	路由策略数据库中的规则
address	一个设备的协议地址(IPv4 或者 IPv6)	maddress	多播地址
neighbor	ARP 或者 NDISC 缓冲区条目	mroute	多播路由缓冲区条目
route	路由表条目	tunnel	IP 中的通道

iproute2 是 Linux 操作系统中管理控制 TCP/IP 网络和流量的新一代工具包,旨在替代工具链(net-tools),即人们比较熟悉的 ifconfig、arp、route、netstat 等命令。net-tools 和 iproute2 命令的对比,如表 5.63 所示。

表 5.63 net-tools 和 iproute2 命令的对比

net-tools 命令	iproute2 命令	net-tools 命令	iproute2 命令
arp -na	ip neigh	netstat -i	ip -s link
ifconfig	ip link	netstat -g	ip addr
ifconfig -a	ip addr show	netstat -l	is -l
ifconfig -help	ip help	netstat -r	ip route
ifconfig -s	ip -s link	route add	ip route add
ifconfig eth0 up	ip link set eth0 up	route del	ip route del
ipmaddr	ip maddr	route -n	ip route show
iptunnel	ip tunnel	vconfig	ip link
netstat	ss		

使用 ip 命令配置网络信息,执行操作如下。

① 使用 ip 命令查看网络地址配置情况,执行命令如下。

```
[root@localhost ~]# ip address show
```

执行命令结果如图 5.49 所示。

```
[root@localhost ~]# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:3e:06:06 brd ff:ff:ff:ff:ff:ff
   inet 192.168.100.100/24 brd 192.168.100.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
   inet6 fe80::dcf2:8616:9181:f592/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
   link/ether 52:54:00:5b:78:11 brd ff:ff:ff:ff:ff:ff
   inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
   link/ether 52:54:00:5b:78:11 brd ff:ff:ff:ff:ff:ff
```

图 5.49 使用 ip 命令查看网络地址配置情况

② 使用 ip 命令查看链路配置情况,执行命令如下。

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

执行命令结果如图 5.50 所示。

```
[root@localhost ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEF
AULT group default qlen 1000
    link/ether 00:0c:29:a7:60:fd brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode D
EFAULT group default qlen 1000
    link/ether 52:54:00:5b:78:11 brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN
mode DEFAULT group default qlen 1000
    link/ether 52:54:00:5b:78:11 brd ff:ff:ff:ff:ff:ff
[root@localhost ~]#
```

图 5.50 使用 ip 命令查看链路配置情况

③ 使用 ip 命令查看路由表信息,执行命令如下。

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

执行命令结果如图 5.51 所示。

```
[root@localhost ~]# ip route
default via 192.168.100.2 dev ens33 proto static metric 100
192.168.100.0/24 dev ens33 proto kernel scope link src 192.168.100.100 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
```

图 5.51 使用 ip 命令查看路由表信息

④ 使用 ip 命令查看链路信息,执行命令如下。

```
[root@localhost ~]# ip link show ens33
```

执行命令结果如图 5.52 所示。

```
[root@localhost ~]# ip link show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1000
    link/ether 00:0c:29:3e:06:06 brd ff:ff:ff:ff:ff:ff
[root@localhost ~]#
```

图 5.52 使用 ip 命令查看链路信息

⑤ 使用 ip 命令查看接口统计信息,执行命令如下。

```
[root@localhost ~]# ip -s link ls ens33
```

执行命令结果如图 5.53 所示。

```
[root@localhost ~]# ip -s link ls ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1000
    link/ether 00:0c:29:3e:06:06 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
    168032    1604    0      0      0      0
    TX: bytes  packets  errors  dropped  carrier  collsns
    109553    785     0      0      0      0
[root@localhost ~]#
```

图 5.53 使用 ip 命令查看接口统计信息

⑥ 使用 ip 命令查看 ARP 表信息,执行命令如下。

```
[root@localhost ~]# ip neigh show
```

执行命令结果如图 5.54 所示。

```
[root@localhost ~]# ip neigh show
192.168.100.2 dev ens33 lladdr 00:50:56:ff:1b:ec STALE
192.168.100.1 dev ens33 lladdr 00:50:56:c0:00:08 REACHABLE
```

图 5.54 使用 ip 命令查看 ARP 表信息

## 5.3 项目实施

### 5.3.1 RAID 配置基本命令

创建 4 个大小都为 2GB 的磁盘,并将其中 3 个创建为 RAID5 阵列磁盘,1 个创建为热备磁盘。

#### 1. 添加磁盘

添加 4 个大小都为 2GB 的磁盘,如图 5.55 所示。添加完成后,重新启动系统,使用 `fdisk -l | grep sd` 命令进行查看,可以看到 4 个磁盘已经被系统检测到,说明磁盘安装成功,如图 5.56 所示。



图 5.55 添加 4 个新磁盘

```
[root@localhost ~]# fdisk -l | grep sd
磁盘 /dev/sda: 42.9 GB, 42949672960 字节, 83886080 个扇区
/dev/sda1 *      2048    2099199    1048576    83  Linux
/dev/sda2      2099200    83886079    40893440    8e  Linux LVM
磁盘 /dev/sdb: 2147 MB, 2147483648 字节, 4194304 个扇区
磁盘 /dev/sdc: 2147 MB, 2147483648 字节, 4194304 个扇区
磁盘 /dev/sde: 2147 MB, 2147483648 字节, 4194304 个扇区
磁盘 /dev/sdd: 2147 MB, 2147483648 字节, 4194304 个扇区
```

图 5.56 磁盘安装成功

## 2. 对磁盘进行初始化

由于 RAID5 要用到整块磁盘,因此使用 fdisk 命令创建分区,此时,需要将整块磁盘创建成一个主分区,将分区类型改为 fd(Linux raid autodetect),如图 5.57 所示,设置完成后保存并退出,执行命令如下。

```
[root@localhost ~]# fdisk -l | grep sd
[root@localhost ~]# fdisk /dev/sdb
```

```
[root@localhost ~]# fdisk /dev/sdb
欢迎使用 fdisk (util-linux 2.23.2).

更改将停留在内存中,直到您决定将更改写入磁盘。
使用写入命令前请三思。

Device does not contain a recognized partition table
使用磁盘标识符 0x2839ca58 创建新的 DOS 磁盘标签。

命令(输入 m 获取帮助): n
Partition type:
  p  primary (0 primary, 0 extended, 4 free)
  e  extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-4194303, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-4194303, 默认为 4194303):
将使用默认值 4194303
分区 1 已设置为 Linux 类型, 大小设为 2 GiB

命令(输入 m 获取帮助): t
已选择分区 1
Hex 代码(输入 L 列出所有代码): fd
已将分区"Linux"的类型更改为"Linux raid autodetect"

命令(输入 m 获取帮助): w
The partition table has been altered!

calling ioctl() to re-read partition table.
正在同步磁盘。
[root@localhost ~]# fdisk -l | grep sdb
磁盘 /dev/sdb: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdb1      2048      4194303      2096128  fd  Linux raid autodetect
```

图 5.57 创建主分区并将分区类型改为 fd

使用同样的方法,设置另外 3 个硬盘,创建主分区并将分区类型改为 fd,使用 fdisk -l | grep sd [b-e]命令进行查看,磁盘初始化设置完成,执行命令如下。

```
[root@localhost ~]# fdisk -l | grep sd[b-e]
```

执行命令结果如图 5.58 所示。

```
[root@localhost ~]# fdisk -l | grep sd[b-e]
磁盘 /dev/sdb: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdb1      2048      4194303      2096128  fd  Linux raid autodetect
磁盘 /dev/sdc: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdc1      2048      4194303      2096128  fd  Linux raid autodetect
磁盘 /dev/sde: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sde1      2048      4194303      2096128  fd  Linux raid autodetect
磁盘 /dev/sdd: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdd1      2048      4194303      2096128  fd  Linux raid autodetect
```

图 5.58 磁盘初始化设置完成

## 3. 创建 RAID5 及其热备份

mdadm 是多磁盘和设备管理(Multiple Disk and Device Administration)的简称,是 Linux 操作系统中的一种标准的软件 RAID 管理工具。在 Linux 操作系统中,目前以虚拟块设备方式实现软件 RAID,利用多个底层的块设备虚拟出一个新的虚拟设备,并利用条带化技术将数据块均匀分布到多个磁盘中以提高虚拟设备的读写性能,利用不同的数据冗余算法来保护用户数据不会因为某个块设备发生故障而完全丢失,且能在设备被替换后将丢失的数据恢复到新的设备中。

目前,虚拟块设备支持 RAID0、RAID1、RAID4、RAID5、RAID6 和 RAID10 等不同的冗余级别和集成方式,也支持由多个 RAID 阵列的层叠构成的阵列。

mdadm 命令格式如下。

```
mdadm [模式] [选项]
```

mdadm 命令各模式及其功能说明、各选项及其功能说明分别如表 5.64 和表 5.65 所示。

表 5.64 mdadm 命令各模式及其功能说明

模 式	功 能 说 明
-A,--assemble	加入一个以前定义的阵列
-B,--build	创建一个逻辑阵列
-C,--create	创建一个新的阵列
-Q,--query	查看一个设备,判断它是一个虚拟块设备还是一个虚拟块设备阵列的一部分
-D,--detail	输出一个或多个虚拟块设备的详细信息
-E,--examine	输出设备中的虚拟块设备的超级块的内容
-F,--follow,--monitor	选择 Monitor 模式
-G,--grow	改变在用阵列的大小或形态

表 5.65 mdadm 命令各选项及其功能说明

选 项	功 能 说 明
-a,--auto{=no,yes,md,mdp,part,p}	自动创建对应的设备,yes 表示会自动在/dev下创建 RAID 设备
-l,--level=	指定要创建的 RAID 的级别(例如,-l 5 或--level=5 表示创建 RAID5)
-n,--raid-devices=	指定阵列中可用 device 数目(例如,-n 3 或--raid-devices=3 表示使用 3 块硬盘来创建 RAID)
-x,--spare-devices=	指定初始阵列的热备磁盘数量(例如,-x 1 或--spare-devices=1 表示热备磁盘只有 1 块)
-f,--fail	使一个 RAID 磁盘发生故障
-r,--remove	移除一个故障的 RAID 磁盘
--add	添加一个 RAID 磁盘
-s,--scan	扫描配置文件或/proc/mdstat 以搜寻丢失的信息
-S(大写)	停止 RAID 磁盘阵列
R,--run	阵列中的某一部分出现在其他阵列或文件系统中时,mdadm 会确认该阵列,使用此选项后将不做确认

使用 mdadm 命令直接将 4 个磁盘中的 3 个创建为 RAID5 阵列,1 个创建为热备磁盘,执行命令如下。

```
[root@localhost ~]# mdadm -- create /dev/md0 -- auto = yes -- level = 5 -- raid - devices = 3 -- spare - devices = 1 /dev/sd[b - e]1
[root@localhost ~]# mdadm - C /dev/md0 - a yes - l 5 - n 3 - x 1 /dev/sd[b - e]1
```

执行命令结果如图 5.59 所示。

对于初学者,建议使用如下完整命令。

```
mdadm -- create /dev/md0 -- auto = yes -- level = 5 -- raid - devices = 3 -- spare - devices = 1 /dev/sd[b - e]1
```

```
[root@localhost ~]# mdadm --create /dev/md0 --auto=yes --level=5 --raid-devices=3 --spare-devices=1 /dev/sd[b-e]1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.

[root@localhost ~]# mdadm -C /dev/md0 -a yes -l 5 -n 3 -x 1 /dev/sd[b-e]1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

图 5.59 创建 RAID5 阵列

创建 RAID5 阵列,执行命令如下。

```
[root@localhost ~]# mdadm -D /dev/md0
```

执行命令结果如图 5.60 所示。

```
[root@localhost ~]# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Thu Aug 27 22:28:04 2020
  Raid Level : raid5
  Array Size : 4188160 (3.99 GiB 4.29 GB)
  Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
  Raid Devices : 3
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Thu Aug 27 22:30:01 2020
  State : clean
  Active Devices : 3
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 1

  Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

  Name : localhost.localdomain:0 (local to host localhost.localdomain)
  UUID : 084c63a1:99c580c3:19a4a6e8:80668dcl
  Events : 19

   Number Major Minor RaidDevice State
    0         8      17        0 active sync  /dev/sdb1
    1         8      33        1 active sync  /dev/sdc1
    4         8      49        2 active sync  /dev/sdd1
    3         8      65        - spare      /dev/sde1
```

图 5.60 查看 RAID5 阵列状态

如果对命令比较熟悉,则可以使用简写命令 `mdadm -C /dev/md0 -a yes -l 5 -n 3 -x 1 /dev/sd[b-e]1` 创建 RAID5 阵列。这两条命令的功能完全一样,其中“`/dev/sd[b-e]1`”可以写成“`/dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1`”,也可以写成“`/dev/sd[b,c,d,e]1`”,这里通过“`[b-e]`”将重复的项目简化。

创建完成之后,使用 `mdadm -D /dev/md0` 命令查看 RAID5 阵列状态,如图 5.60 所示。从图 5.60 中可以看出,`/dev/sdb1`、`/dev/sdc1` 和 `/dev/sdd1` 组成了 RAID5 阵列,而 `/dev/sde1` 为热备磁盘,显示结果的主要字段的含义如下。

- (1) Version: 版本。
- (2) Creation Time: 创建时间。
- (3) Raid Level: RAID 的级别。
- (4) Array Size: 阵列容量。
- (5) Active Devices: 活动的磁盘数目。
- (6) Working Devices: 所有的磁盘数目。
- (7) Failed Devices: 出现故障的磁盘数目。
- (8) Spare Devices: 热备份的磁盘数目。

#### 4. 添加 RAID5 阵列

添加 RAID5 阵列到配置文件/etc/mdadm.conf 中,默认此文件是不存在的,执行命令如下。

```
[root@localhost ~]# echo 'DEVICE /dev/sd[b-e]1' >> /etc/mdadm.conf
[root@localhost ~]# mdadm -Ds >> /etc/mdadm.conf
[root@localhost ~]# cat /etc/mdadm.conf
```

执行命令结果如图 5.61 所示。

```
[root@localhost ~]# echo 'DEVICE /dev/sd[b-e]1' >> /etc/mdadm.conf
[root@localhost ~]# mdadm -Ds >> /etc/mdadm.conf
mdadm: Unknown keyword 'DEVICE'
[root@localhost ~]# cat /etc/mdadm.conf
'DEVICE /dev/sd[b-e]1'
ARRAY /dev/md0 metadata=1.2 spares=1 name=localhost.localdomain:0 UUID=a641a8af:adb107d0:c4414430:ba55ad59
```

图 5.61 添加 RAID5 阵列

#### 5. 格式化磁盘阵列

使用 mkfs.xfs /dev/md0 命令对磁盘阵列/dev/md0 进行格式化,执行命令如下。

```
[root@localhost ~]# mkfs
[root@localhost ~]# mkfs.xfs /dev/md0
```

执行命令结果如图 5.62 所示。

```
[root@localhost ~]# mkfs
mkfs      mkfs.btrfs  mkfs.cramfs  mkfs.ext2  mkfs.ext3  mkfs.ext4  mkfs.fat  mkfs.minix  mkfs.msdos  mkfs.vfat  mkfs.xfs
[root@localhost ~]# mkfs.xfs /dev/md0
meta-data=/dev/md0          isize=512    agcount=8, agsize=139944 blks
                             sectsz=512    attr=2, projid32bit=1
                             crc=1          finobt=0, sparse=0
data =                       bsize=4096 blocks=1047040, imaxpct=25
                             suitt=128     swidth=256 blks
naming =version 2           bsize=4096 ascii-ci=0 ftype=1
log =internal log         bsize=4096 blocks=2560, version=2
                             sectsz=512  suitt=8 blks, lazy-count=1
realtime=none              extsz=4096 blocks=0, rtextents=0
```

图 5.62 格式化硬盘阵列

#### 6. 挂载磁盘阵列

将磁盘阵列挂载后即可使用,也可以把挂载项写入/etc/fstab 文件中,此时可实现自动挂载,即使用 echo '/dev/md0 /mnt/raid5 xfs defaults 0 0' >> /etc/fstab 命令,这样下次系统重新启动后即可使用,执行命令如下。

```
[root@localhost ~]# mkdir /mnt/raid5
[root@localhost ~]# mount /dev/md0 /mnt/raid5
[root@localhost ~]# ls -l /mnt/raid5
[root@localhost ~]# echo '/dev/md0 /mnt/raid5 xfs defaults 0 0' >> /etc/fstab
```

执行命令结果如图 5.63 所示。

```
[root@localhost ~]# mkdir /mnt/raid5
[root@localhost ~]# mount /dev/md0 /mnt/raid5
[root@localhost ~]# ls -l /mnt/raid5
总用量 0
[root@localhost ~]# echo '/dev/md0 /mnt/raid5 xfs defaults 0 0' >> /etc/fstab
[root@localhost ~]# tail -3 /etc/fstab
UUID=6d58086e-0a6b-4399-93dc-c2016ea17fe0 /boot xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
/dev/md0 /mnt/raid5 xfs defaults 0 0
```

图 5.63 挂载磁盘阵列

查看磁盘挂载使用情况,执行命令如下。

```
[root@localhost ~]# df -hT
```

执行命令结果如图 5.64 所示。

```
[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 36G 4.2G 31G 12% /
devtmpfs  devtmpfs 1.9G 0 1.9G 0% /dev
tmpfs    tmpfs    1.9G 0 1.9G 0% /dev/shm
tmpfs    tmpfs    1.9G 13M 1.9G 1% /run
tmpfs    tmpfs    1.9G 0 1.9G 0% /sys/fs/cgroup
/dev/sda1 xfs 1014M 179M 836M 18% /boot
tmpfs    tmpfs    378M 12K 378M 1% /run/user/42
tmpfs    tmpfs    378M 0 378M 0% /run/user/0
/dev/md0  xfs 4.0G 33M 4.0G 1% /mnt/raid5
```

图 5.64 查看磁盘挂载使用情况

### 5.3.2 RAID5 阵列实例配置

测试以热备磁盘替换阵列中的磁盘并同步数据,移除损坏的磁盘,添加一个新磁盘作为热备磁盘,并删除 RAID 阵列。

#### 1. 写入测试文件

在 RAID5 阵列上写入一个大小为 10MB 的文件,将其命名为 10M\_file,以供数据恢复时测试使用,并显示该内容,执行命令如下。

```
[root@localhost ~]# cd /mnt/raid5
[root@localhost raid5]# dd if=/dev/zero of=10M_file count=1 bs=10M
[root@localhost raid5]# ls -l
```

执行命令结果如图 5.65 所示。

```
[root@localhost ~]# cd /mnt/raid5
[root@localhost raid5]# dd if=/dev/zero of=10M_file count=1 bs=10M
记录了1+0 的读入
记录了1+0 的写出
10485760字节(10 MB)已复制, 0.00785943 秒, 1.3 GB/秒
[root@localhost raid5]# ls -l
总用量 10240
-rw-r--r--. 1 root root 10485760 8月 28 05:49 10M_file
```

图 5.65 写入测试文件

#### 2. RAID 设备的数据恢复

如果 RAID 设备中的某个磁盘损坏,则系统会自动停止该磁盘的工作,使热备磁盘代替损坏的磁盘继续工作。例如,假设/dev/sdc1 损坏,更换损坏的 RAID 设备中成员的方法是先使用 mdadm /dev/md0 --fail /dev/sdc1 或 mdadm /dev/md0 -f /dev/sdc1 命令将损坏的 RAID 成员标记为失效,再使用 mdadm -D /dev/md0 命令查看 RAID 阵列信息,发现热备磁盘/dev/sde1 已经自动替换了损坏的/dev/sdc1,且文件没有损坏,执行命令如下。

```
[root@localhost ~]# mdadm /dev/md0 -f /dev/sdc1
[root@localhost ~]# mdadm -D /dev/md0
[root@localhost ~]# ls -l
```

执行命令结果如图 5.66 所示。

#### 3. 移除损坏的磁盘

使用 mdadm /dev/md0 -r /dev/sdc1 或 mdadm /dev/md0 --remove /dev/sdc1 命令移除损坏的磁盘/dev/sdc1,再次查看信息,可以看到 Failed Devices 字段数值变为 0,执行命令如下。

```
[root@localhost ~]# mdadm /dev/md0 -r /dev/sdc1
[root@localhost ~]# mdadm -D /dev/md0
```

```
[root@localhost raids]# mdadm /dev/md0 -f /dev/sdc1
mdadm: Unknown keyword 'DEVICE'
mdadm: set /dev/sdc1 faulty in /dev/md0
[root@localhost raids]# mdadm -D /dev/md0
mdadm: Unknown keyword 'DEVICE'
/dev/md0:
  Version : 1.2
  Creation Time : Fri Aug 28 05:06:13 2020
  Raid Level : raid5
  Array Size : 4188160 (3.99 GiB 4.29 GB)
  Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
  Raid Devices : 3
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Fri Aug 28 06:06:18 2020
  State : clean
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 1
  Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

   Name : localhost.localdomain:0 (local to host localhost.localdomain)
   UUID : a641a8af:adb107d0:c4414430:ba55ad59
   Events : 37

  Number Major Minor RaidDevice State
     0       8       17         0  active sync  /dev/sdb1
     3       8       65         1  active sync  /dev/sde1
     4       8       49         2  active sync  /dev/sdd1
     1       8       33         -  faulty     /dev/sdc1

[root@localhost raids]# ls -l
总用量 10240
-rw-r--r--. 1 root root 10485760 8月 28 05:49 10M_file
```

图 5.66 RAID5 设备的数据恢复

执行命令结果如图 5.67 所示。

```
[root@localhost ~]# mdadm /dev/md0 -r /dev/sdc1
mdadm: hot removed /dev/sdc1 from /dev/md0
[root@localhost ~]# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Aug 28 18:18:39 2020
  Raid Level : raid5
  Array Size : 4188160 (3.99 GiB 4.29 GB)
  Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Fri Aug 28 18:21:25 2020
  State : clean
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

   Name : localhost.localdomain:0 (local to host localhost.localdomain)
   UUID : ad8d67fb:d6ac0c3f:e27ac3f2:207124a7
   Events : 38

  Number Major Minor RaidDevice State
     0       8       17         0  active sync  /dev/sdb1
     3       8       65         1  active sync  /dev/sde1
     4       8       49         2  active sync  /dev/sdd1
```

图 5.67 移除损坏的磁盘

#### 4. 添加新的磁盘作为热备磁盘

添加新的磁盘,可以看到新增的磁盘/dev/sdf,执行命令如下。

```
[root@localhost ~]# fdisk -l grep sd
[root@localhost ~]# fdisk /dev/sdf
[root@localhost ~]# mkfs.xfs /dev/sdf1
```

执行命令结果如图 5.68 所示。

使用 `mdadm /dev/md0 --add /dev/sdf1` 或 `mdadm /dev/md0 --a /dev/sdf1` 命令,在阵列中添加

```
[root@localhost ~]# fdisk -l | grep sd
磁盘 /dev/sda: 42.9 GB, 42949672960 字节, 83886080 个扇区
/dev/sda1: 2048 2099199 1048576 83 Linux
/dev/sda2: 2099200 83886079 40893440 8e Linux LVM
磁盘 /dev/sdb: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdb1: 2048 4194303 2096128 fd Linux raid autodetect
磁盘 /dev/sdc: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdc1: 2048 4194303 2096128 fd Linux raid autodetect
磁盘 /dev/sdf: 2147 MB, 2147483648 字节, 4194304 个扇区
磁盘 /dev/sde: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sde1: 2048 4194303 2096128 fd Linux raid autodetect
磁盘 /dev/sdd: 2147 MB, 2147483648 字节, 4194304 个扇区
/dev/sdd1: 2048 4194303 2096128 fd Linux raid autodetect
[root@localhost ~]# fdisk /dev/sdf
欢迎使用 fdisk (util-linux 2.23.2).

更改将停留在内存中, 直到您决定将更改写入磁盘。
使用写入命令前请三思。

Device does not contain a recognized partition table
使用磁盘标识符 0x052457d2 创建新的 DOS 磁盘标签。

命令(输入 m 获取帮助): p

磁盘 /dev/sdf: 2147 MB, 2147483648 字节, 4194304 个扇区
units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x052457d2

   设备 Boot      Start         End      Blocks   Id  System
   -----
命令(输入 m 获取帮助): n
Partition type:
  p  primary (0 primary, 0 extended, 4 free)
  e  extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-4194303, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-4194303, 默认为 4194303):
将使用默认值 4194303
分区 1 已设置为 Linux 类型, 大小设为 2 GiB

命令(输入 m 获取帮助): t
已选择分区 1
Hex 代码(输入 L 列出所有代码): fd
已将分区“Linux”的类型更改为“Linux raid autodetect”

命令(输入 m 获取帮助): w
The partition table has been altered!
calling ioctl() to re-read partition table.
正在同步磁盘。
[root@localhost ~]# mkfs.xfs /dev/sdf1
```

图 5.68 新增的磁盘/dev/sdf

一块新的磁盘/dev/sdf1, 添加之后其会自动变为热备磁盘, 查看相关信息, 执行命令如下。

```
[root@localhost ~]# mdadm /dev/md0 --add /dev/sdf1
[root@localhost ~]# mdadm -D /dev/md0
```

执行命令结果如图 5.69 所示。

```
[root@localhost ~]# mdadm /dev/md0 --add /dev/sdf1
mdadm: added /dev/sdf1
[root@localhost ~]# mdadm -D /dev/md0
/dev/md0:
Version : 1.2
Creation Time : Fri Aug 28 18:18:39 2020
Raid Level : raid5
Array Size : 4188160 (3.99 GiB 4.29 GB)
Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
Raid Devices : 3
Total Devices : 4
Persistence : Superblock is persistent

Update Time : Fri Aug 28 18:59:17 2020
State : clean
Active Devices : 3
Working Devices : 4
Failed Devices : 0
Spare Devices : 1

Layout : left-symmetric
Chunk Size : 512K
Consistency Policy : resync

Name : localhost.localdomain:0 (local to host localhost.localdomain)
UUID : ad8d67fb:d6ac0c3f:e27ac3f2:207124a7
Events : 41

Number Major Minor RaidDevice State
0 8 17 0 active sync /dev/sdb1
3 8 65 1 active sync /dev/sda1
4 8 49 2 active sync /dev/sdd1
5 8 81 - spare /dev/sdf1
```

图 5.69 查看相关信息

## 5. 删除 RAID 阵列

RAID 阵列的删除一定要慎重, 操作不当可能会导致系统无法启动, 操作步骤如下。

(1) 如果系统中配置了自动挂载功能,则应该使用 Vim 编辑器删除/etc/fstab 文件中的 RAID 的相关启动信息,即删除信息“/dev/md0 /mnt/raid5 xfs defaults 0 0”。

(2) 卸载 RAID 磁盘挂载(使用 umount /mnt/raid5 命令)。

(3) 停止 RAID 磁盘工作(使用 mdadm -S /dev/md0 命令)。

(4) 删除 RAID 中的相关磁盘(使用 mdadm --misc --zero-superblock /dev/sd[b,d-f]1 命令)。

(5) 删除 RAID 相关配置文件(使用 rm -f /etc/mdadm.conf 命令)。

(6) 使用 mdadm -D /dev/md0 命令查看 RAID5 阵列相关情况,可以看出已经删除了 RAID5 阵列,执行命令如下。

```
[root@localhost ~]# vim /etc/fstab
[root@localhost ~]# tail -3 /etc/fstab
[root@localhost ~]# umount /mnt/raid5
[root@localhost ~]# mdadm -S /dev/md0
[root@localhost ~]# mdadm --misc --zero-superblock /dev/sd[b,d-f]1
[root@localhost ~]# rm -f /etc/mdadm.conf
[root@localhost ~]# mdadm -D /dev/md0
```

执行命令结果如图 5.70 所示。

```
[root@localhost ~]# vim /etc/fstab
#
# /etc/fstab
# Created by anaconda on Mon Jun 8 01:15:36 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults xfs 0 0
/dev/mapper/centos-swap swap defaults xfs 0 0
/dev/md0 /mnt/raid5 xfs defaults 0 0
~

[root@localhost ~]# tail -3 /etc/fstab
/dev/mapper/centos-root / xfs defaults xfs 0 0
/dev/mapper/centos-swap swap defaults xfs 0 0
~
[root@localhost ~]# umount /mnt/raid5
[root@localhost ~]# mdadm -S /dev/md0
mdadm: stopped /dev/md0
[root@localhost ~]# mdadm --misc --zero-superblock /dev/sd[b,d-f]1
[root@localhost ~]# rm -f /etc/mdadm.conf
[root@localhost ~]# mdadm -D /dev/md0
mdadm: cannot open /dev/md0: No such file or directory
```

图 5.70 删除 RAID5 阵列

### 5.3.3 磁盘扩容配置

当主机磁盘空间不足时,需要对主机磁盘进行扩容,操作过程如下。

(1) 本案例在虚拟机中添加 20GB 磁盘进行操作演示,如图 5.71 所示。

(2) 添加磁盘完成后,重启虚拟机,使用命令查看主机磁盘情况,执行命令如下。

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

执行命令结果如图 5.72 所示。

(3) 使用命令对磁盘进行分区,执行命令如下。

```
[root@localhost ~]# fdisk /dev/sdb
```



图 5.71 添加磁盘

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda         8:0    0  40G  0 disk
├─sda1      8:1    0   1G  0 part /boot
├─sda2      8:2    0  39G  0 part
│   └─centos-root 253:0  0  35G  0 lvm /
│       └─centos-swap 253:1  0   4G  0 lvm [SWAP]
└─sdb       8:16   0  20G  0 disk
sr0         11:0   1  4.3G  0 rom
[root@localhost ~]#
```

图 5.72 主机磁盘情况

执行命令结果如图 5.73 所示。

```
[root@localhost ~]# fdisk /dev/sdb
欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

Device does not contain a recognized partition table
使用磁盘标识符 0xffe9c397 创建新的 DOS 磁盘标签。

命令(输入 m 获取帮助): n
Partition type:
  p  primary (0 primary, 0 extended, 4 free)
  e  extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-41943039, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-41943039, 默认为 41943039):
将使用默认值 41943039
分区 1 已设置为 Linux 类型, 大小设为 20 GiB

命令(输入 m 获取帮助): w
The partition table has been altered!

calling ioctl() to re-read partition table.
正在同步磁盘。
[root@localhost ~]#
```

图 5.73 主机磁盘分区

(4) 使用命令对磁盘进行格式化,执行命令如下。

```
[root@localhost ~]# lsblk
[root@localhost ~]# mkfs.xfs /dev/sdb1
```

执行命令结果如图 5.74 所示。

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda         8:0    0   40G  0 disk
├─sda1     8:1    0    1G  0 part /boot
├─sda2     8:2    0   39G  0 part
│   ├─centos-root 253:0  0   35G  0 lvm /
│   └─centos-swap 253:1  0    4G  0 lvm [SWAP]
└─sdb      8:16   0   20G  0 disk
   └─sdb1   8:17   0   20G  0 part
sr0        11:0    1   4.3G  0 rom
[root@localhost ~]#
[root@localhost ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1          isize=512    agcount=4, agsize=1310656 blks
                =           sectsz=512   attr=2, projid32bit=1
                =           crc=1        finobt=0, sparse=0
data        =           bsize=4096   blocks=524264, imaxpct=25
                =           sunit=0     swidth=0 blks
naming      =version 2     bsize=4096   ascii-ci=0 ftype=1
log         =internal log bsize=4096   blocks=2560, version=2
                =           sectsz=512   sunit=0 blks, lazy-count=1
realtime    =none         extsz=4096   blocks=0, rtextents=0
[root@localhost ~]#
```

图 5.74 云主机磁盘格式化

(5) 添加新 LVM 到已有的 LVM 组中,实现扩容,加载生效,执行命令如下。

```
[root@localhost ~]# pvcreate /dev/sdb1
[root@localhost ~]# vgextend centos /dev/sdb1
[root@localhost ~]# lvextend -L 500G /dev/mapper/centos - root
[root@localhost ~]# xfs_growfs /dev/centos/root
[root@localhost ~]# fsadm resize /dev/mapper/centos - root
[root@localhost ~]# df -hT
```

执行命令结果如图 5.75 所示。

```
[root@localhost ~]# pvcreate /dev/sdb1
WARNING: xfs signature detected on /dev/sdb1 at offset 0. wipe it? [y/n]: y
Wiping xfs signature on /dev/sdb1.
Physical volume "/dev/sdb1" successfully created.
[root@localhost ~]#
[root@localhost ~]# vgextend centos /dev/sdb1
Volume group "centos" successfully extended
[root@localhost ~]#
[root@localhost ~]# lvextend -L 500G /dev/mapper/centos-root
Insufficient free space: 119041 extents needed, but only 5119 available
[root@localhost ~]#
[root@localhost ~]# xfs_growfs /dev/centos/root
meta-data=/dev/mapper/centos-root  isize=512    agcount=4, agsize=2293504 blks
                =           sectsz=512   attr=2, projid32bit=1
                =           crc=1        finobt=0 spinodes=0
data        =           bsize=4096   blocks=9174016, imaxpct=25
                =           sunit=0     swidth=0 blks
naming      =version 2     bsize=4096   ascii-ci=0 ftype=1
log         =internal     bsize=4096   blocks=4479, version=2
                =           sectsz=512   sunit=0 blks, lazy-count=1
realtime    =none         extsz=4096   blocks=0, rtextents=0
[root@localhost ~]#
[root@localhost ~]# fsadm resize /dev/mapper/centos-root
[root@localhost ~]#
[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        35G   5.7G   30G    17% /
devtmpfs      devtmpfs   3.8G   0   3.8G    0% /dev
tmpfs         tmpfs      3.9G   0   3.9G    0% /dev/shm
tmpfs         tmpfs      3.9G  13M   3.8G    1% /run
tmpfs         tmpfs      3.9G   0   3.9G    0% /sys/fs/cgroup
/dev/sda1     xfs       1014M  179M  836M   18% /boot
tmpfs         tmpfs      781M  4.0K  781M    1% /run/user/42
tmpfs         tmpfs      781M  24K   781M    1% /run/user/0
/dev/sr0      iso9660    4.3G  4.3G   0  100% /run/media/root/Centos 7 x86_64
[root@localhost ~]#
```

图 5.75 主机磁盘扩容

## 课后习题

### 1. 选择题

- (1) 下列中不是 Linux 操作系统的特点的是( )。
- A. 多用户                      B. 单任务                      C. 开放性                      D. 设备独立性

