

# 项目1

## Linux基本配置与管理

### 学习目标

- 理解 Linux 的发展历史、Linux 的特性、Linux 基本命令及 Vi、Vim 编辑器的使用等相关理论知识。
- 掌握 VMware Workstation 的安装、虚拟主机 CentOS 7 的安装、系统克隆与快照管理、SecureCRT 与 SecureFX 配置管理等相关知识与技能。

### 1.1 项目概述

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 基本命令及 Vi、Vim 编辑器的使用等相关理论知识,项目实践部分讲解 VMware Workstation 的安装、虚拟主机 CentOS 7 的安装、系统克隆与快照管理、SecureCRT 与 SecureFX 配置管理等相关知识与技能。

### 1.2 必备知识

#### 1.2.1 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)工程师在美国贝尔实验室开发的一种操作系统。1972 年,他与丹尼斯·里奇(D. Ritchie)工程师一起用 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。

由于政策改变,在 UNIX v7 推出之后,美国电话电报公司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。



视频讲解

MINIX 最有名的学生用户是芬兰人李纳斯·托瓦兹(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。在设计哲学上,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 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 1.0 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代码必须公开,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。

### 1.2.2 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。它包含了 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,互相影响。

(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。



视频讲解

### 1.2.3 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,它属于目录操作命令,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缩写,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,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 都视其为一个空格。

## 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: 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 4.8.5
20150623 (Red Hat 4.8.5 - 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 1.1 所示。

表 1.1 cat 命令各选项及其功能说明

选 项	功 能 说 明
-A	等价于-vET
-b	对非空输出进行编号
-e	等价于-vE
-E	在每行结束处显示 \$
-n	从 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 - test01.txt      history.txt          mkfs.ext2           mkrfc2734 公共  图片  音乐
anaconda - ks.cfg   initial - setup - ks.cfg  mkfs.msos           mnt        模板  文档  桌面
font.map            mkfontdir            mkinitrd            user01     视频  下载
[root@localhost ~]# cat a1 - test01.txt //显示 a1 - test01.txt 文件的内容
aaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbb
cccccccccccccccc
[root@localhost ~]# cat -nE a1 - test01.txt //显示 a1 - test01.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 aaaaaaaaaaaaaaaaaa$
    2 bbbbbbbbbbbbbbbb$
    3 cccccccccccccccc$
[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 1.2 所示。

表 1.2 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 - test01.txt
cccccccccccccccc
bbbbbbbbbbbbbbbb
aaaaaaaaaaaaaaaa
[root@localhost ~]#
```

(3) head 命令: 查看文件的前  $n$  行内容。

head 命令用来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 1.3 所示。

表 1.3 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 $n$ 字节,如-c5,表示文件内容的前 5 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  $n$  行内容。

tail 命令用来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 1.4 所示。

表 1.4 tail 命令各选项及其功能说明

选 项	功 能 说 明
-c	显示文件的后 $n$ 字节,如-c5,表示文件内容后 5 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
-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 1.5 所示。

表 1.5 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 1.6 所示。

表 1.6 控制字符

控制字符	功能说明
\\	输出\本身
\a	输出警告声音
\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	Enter 键
\t	制表符,即 Tab 键
\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 命令中各选项的含义如表 1.7 所示。

表 1.7 shutdown 命令中各选项的含义

选 项	含 义
-k	并不真正关机,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	快速关机且不经过初始化程序
-c	取消一个已经运行的关机操作

需要特别说明的是,该命令只能由 root 用户使用。

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

```
[root@localhost ~]# halt now //立刻关闭系统
```

reboot 命令的工作过程与 halt 命令类似,其作用是重新启动系统,而 halt 命令是关机。其参数也与 halt 命令类似, 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 //5min后重启系统
[root@localhost ~]# shutdown -c //取消关机操作
```

退出终端窗口命令是 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/



视频讲解

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：路径不是从“/”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 1.8 所示。

表 1.8 特殊符号表示的目录

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

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 1.9 所示。

表 1.9 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 1.10 所示。



视频讲解

表 1.10 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 yyyy-mm-dd
-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 file01.txt //创建一个文件
[root@localhost mnt]# touch file0{2..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 file01.txt
-rw-r--r--. 1 root root 0 5月 10 19:35 file02.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 file04.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 file01.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file02.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 file04.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 1.11 所示。

表 1.11 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 file01.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file02.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 file04.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 file01.txt
-rw-r--r--. 1 root root 0 6月 26 2020 file02.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 file04.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` 命令是常用的命令,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
```

`rmdir` 命令各选项及其功能说明如表 1.12 所示。

表 1.12 `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 命令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 1.13 所示。

表 1.13 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 file01.txt
-rw-r--r--. 1 root root 0 6月 26 2022 file02.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 file04.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 1.14 所示。

表 1.14 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

续表

选项	功能说明
-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{1..3}.txt
[root@localhost mnt]# mkdir user0{1..3}
[root@localhost mnt]# dir
a01.txt a02.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 a01.txt
-rw-r--r--. 1 root root 0 5月 10 19:49 a02.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/a01.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 a01.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 1.15 所示。

表 1.15 mv 命令各选项及其功能说明

选项	功能说明
-f	覆盖前不询问
-i	覆盖前询问
-n	不覆盖已存在文件
-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 a01.txt
-rw-r--r--. 1 root root 0 6月 25 20:27 a02.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/a01.txt /mnt/test01.txt //将 a01.txt 重命名为 test01.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 a02.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 test01.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 中常用的备份命令之一。

tar 命令可以用于建立、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 1.16 所示。

表 1.16 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 来压缩/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，若无此选项，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

可以使用 tar 命令打包、归档文件或目录。

① 将/mnt 目录打包为一个文件 test01.tar，将其压缩为文件 test01.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{1..2}.txt //新建两个文件
[root@localhost ~]# mkdir /mnt/test0{1..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 a01.txt
-rw-r--r--. 1 root root 0 6月 25 22:32 a02.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/test01.tar /mnt //将/mnt下的所有文件归并为文件 test01.tar

tar: 从成员名中删除开头的"/"
/mnt/
/mnt/a01.txt
/mnt/a02.txt
/mnt/test01
/mnt/test02
[root@localhost ~]# ls /root/user01
test01.tar
[root@localhost ~]#

```

② 在目录/root/user01目录下生成压缩文件 test01.tar,使用 gzip 命令可对单个文件进行压缩,原归档文件 test01.tar 就没有了,并生成压缩文件 test01.tar.gz,执行以下命令。

```

[root@localhost ~]# gzip /root/user01/test01.tar
[root@localhost ~]# ls -l /root/user01
总用量 8
-rw-r--r--. 1 root root 190 6月 25 22:36 test01.tar.gz
[root@localhost ~]#

```

③ 在/root/user01目录下生成压缩文件 test01.tar.gz,可以一次完成归档和压缩,把两步合并为一步,执行以下命令。

```

[root@localhost ~]# tar -zcvf /root/user01/test01.tar.gz /mnt
tar: 从成员名中删除开头的"/"
/mnt/
/mnt/a01.txt
/mnt/a02.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 test01.tar.gz
[root@localhost ~]#

```

④ 对文件 test01.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 test01.tar.gz
[root@localhost user01]# gzip -d test01.tar.gz
[root@localhost user01]# tar -xf test01.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 test01.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 test01.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 a01.txt
-rw-r--r--. 1 root root 0 6月 25 23:12 a02.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 1.17 所示。

表 1.17 whereis 命令各选项及其功能说明

选 项	功 能 说 明
-b	只搜索二进制文件
-B<目录>	定义二进制文件查找路径
-m	只搜索 man 手册
-M<目录>	定义 man 手册查找路径
-s	只搜索源代码
-S<目录>	定义源代码查找路径
-f	终止<目录>参数列表

续表

选 项	功 能 说 明
-u	搜索不常见记录
-l	输出有效查找路径

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.1.gz
[root@localhost ~]#
```

(24) `locate` 命令: 查找文件或目录。

`locate` 命令用于查找文件或目录。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 1.18 所示。

表 1.18 `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 <code>DBPATH</code> 指定的数据库,而不是默认数据库/ <code>/var/lib/mlocate/mlocate.db</code>
-e	仅输出当前现有文件的条目
-L	当文件存在时,遵循尾随的符号链接(默认)
-h	显示帮助
-i	忽略字母大小写
-l	将输出(或计数)限制为 <code>LIMIT</code> 个条目
-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-68.11.0-1.el7.CentOS.x86_64.rpm
[root@localhost ~]#
```