

Linux服务器构建与运维管理

第02章：Linux基本命令

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

<http://linux.xg.hactcm.edu.cn>
<http://www.51xueweb.cn>

河南中医药大学信息管理与信息系统教研室
信息技术学院网络与信息系统科研工作室

2021.3

提纲

- 系统基本配置 `hostnamectl timedatectl localctl`
- 查看系统信息 `who id uname date ip addr dmidecode lspci`

系统配置命令

- 用户权限管理 `groupadd groupmod groupdel useradd passwd usermod userdel`
`chmod chown chgrp umask`
- 文件目录操作 `cd mkdir touch file mv cp rm rmdir find whereis which ln`
- 文本处理操作 `ls cat more less head tail grep sort uniq sed vi wc diff awk paste`
- 磁盘存储管理 `zip gzip bzip2 tar fdisk df du mkfs mdadm mount umount`
- 网络接口管理 `nmcli nmtui bond`

系统管理命令

- 其他操作命令 `man help echo export history clear`

系统辅助命令



什么是命令?

命令就是Shell命令!

- **内置命令:**
 - Shell 自带的命令。
 - 在 Shell 内部可以通过函数来实现，当 Shell 启动后，这些命令所对应的代码（函数体代码）也被加载到内存中，所以使用内置命令是非常快速的。
- **外部命令:**
 - 外部命令是应用程序，一个命令就对应一个应用程序。
 - 运行外部命令要开启一个新进程，效率上比内置命令差很多。



什么是命令？

命令就是Shell命令！

- **用户输入一个命令后：**
 - Shell检测命令是不是内置命令，如果是就执行，如果不是继续。
 - Shell检测命令有对应的外部程序，转而执行外部程序，执行结束后回到 Shell。
 - Shell检测命令没有对应的外部程序，就提示用户该命令不存在。
- **如果需要执行一个命令，但是系统提示不存在：**
 - 检查是否输入错误，也许就是写错命令了。
 - 安装该命令对应的应用程序，安装成功后就有该命令了。
 - 操作系统发行版会预先安装一些程序，但是不能够满足全部需求。
 - 可以使用C语言等写个程序安装到Linux，创建一个满足需求的命令。



什么是命令？

命令就是Shell命令！

命令三要素

语法

选项

参数

基本格式：command [选项] [参数]

[]表示可选的，也就是可有可无。

有些命令不写选项和参数也能执行，有些命令在必要的时候可以附带选项和参数。

- 选项的作用是调整命令功能。
 - 没有选项，命令只能执行最基本的功能；
 - 增加了选项，则能执行更多功能，或者显示更加丰富的数据。
- 选项分为两种：短格式选项和长格式选项
 - 短格式选项是长格式选项简写，用一个减号-和一个字母表示，例如ls -l。
 - 长格式选项是完整英文单词，用两个减号--和一个单词表示，例如ls --all。
 - 通常情况：短格式选项是长格式选项的缩写，短格式有对应长格式选项。
 - 也有例外：比如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还需要通过帮助手册来查询。



什么是命令？

命令就是Shell命令！

命令三要素

语法

选项

参数

基本格式：command [选项] [参数]

[]表示可选的，也就是可有可无。

有些命令不写选项和参数也能执行，有些命令在必要的时候可以附带选项和参数。

- 选项的作用是调整命令功能。
- 参数是命令的操作对象，一般情况下，文件、目录、用户和进程等都可以作为参数被命令操作。
 - 命令一般都需要参数，用于指定命令操作的对象是谁。
 - 命令如果省略参数，则该命令有默认参数，就按照默认参数执行。
 - 命令可以同时附带选项和参数，例如：ls -l /etc/
 - 有些命令的选项后面也可以附带参数，用来补全选项，或者调整选项的功能细节。



越讲越糊涂 / 越听越迷糊

迷 茫
M I M A N G

会用就行了 / 多用就懂了



什么是Shell?

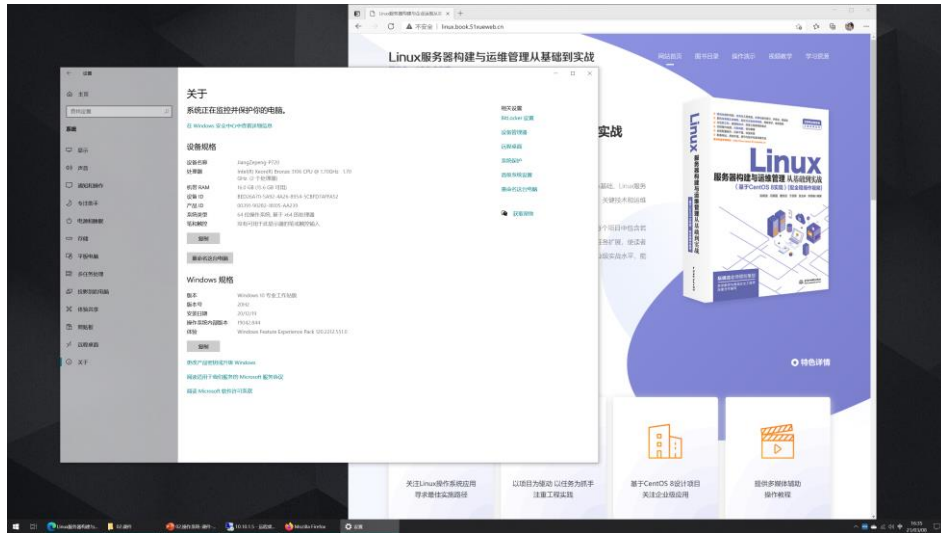
- **Shell是在Linux操作系统中运行的一种特殊程序:**
 - Shell指一种应用程序，这个应用程序提供了一个界面，用户通过这个界面访问操作系统内核的服务。
 - Shell位于操作系统内核与用户之间。
 - Shell负责接收用户输入的命令并进行解释，将执行的操作传递给系统内核执行。
 - Shell在用户和内核之间充当“翻译官”的角色。
- **Shell也是一种命令语言，还是一种程序设计语言。**
 - Shell Script即Shell脚本，是一种为Shell编写的脚本程序。
 - 当说Shell编程时，说的就是Shell脚本编程。
 - 编写Shell脚本同时是为了进行自动化或者半自动化的操作系统维护管理。
 - Shell和Shell Script是两个不同的概念。



如何管理Linux操作系统

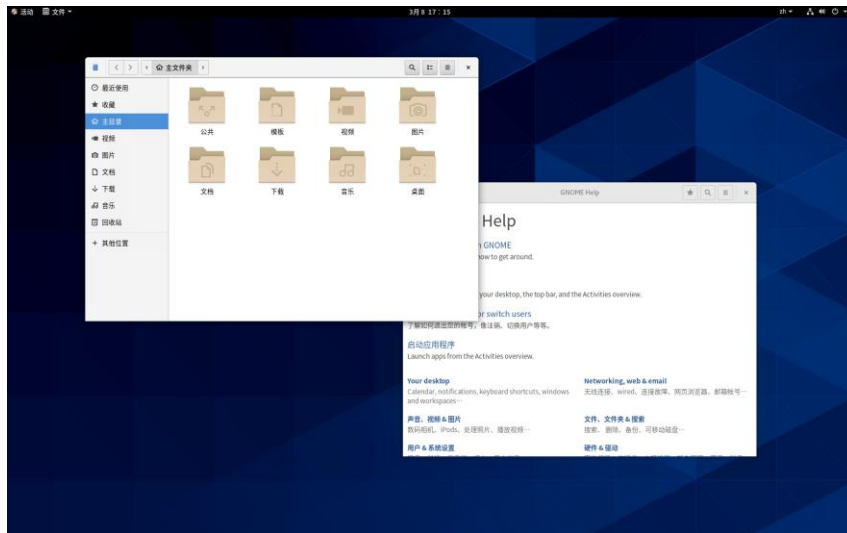
- **Linux操作系统的交互界面主要分为三类：**
 - GUI：图形用户界面，Graphical User Interface。
 - CLI：命令行界面，Command Line Interface shell。
 - TUI：终端用户界面，基于文本的用户界面。
- **Linux操作系统的管理方式主要分为两类：**
 - 本地管理：直接在操作系统主机上操作，通过键盘鼠标显示器。
 - **远程管理**：网络方式远程管理控制，通过远程终端软件。
 - 远程管理客户端软件：远程桌面、Termius、Putty
 - 远程管理服务端软件：RDP Server、OpenSSH
 - 远程管理通信协议：RDP、SSH、Telnet、VNC

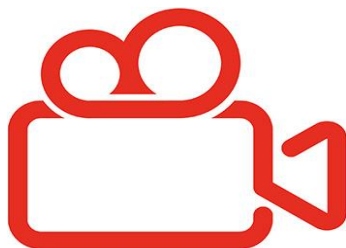




```
CentOS Linux 8
Kernel 4.18.8-240.15.1.el8_3.x86_64 on an x86_64

Project-02-Task-01 login: root
Password:
Last login: Mon Mar  8 16:17:45 from 10.10.0.1
root@Project-02-Task-01 ~# ip addr
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: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:56:9a:f6:33 brd ff:ff:ff:ff:ff:ff
    inet 10.10.2.103/24 brd 10.10.2.255 scope global noprefixroute ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::258:56ff:fe9a:f633:64 scope link
        valid_lft forever preferred_lft forever
root@Project-02-Task-01 ~# ping www.baidu.com
PING www.a.shifen.com (182.61.208.6) 56(84) bytes of data.
64 bytes from 182.61.208.6 (182.61.208.6): icmp_seq=1 ttl=49 time=12.6 ms
64 bytes from 182.61.208.6 (182.61.208.6): icmp_seq=2 ttl=49 time=12.4 ms
64 bytes from 182.61.208.6 (182.61.208.6): icmp_seq=3 ttl=49 time=12.3 ms
64 bytes from 182.61.208.6 (182.61.208.6): icmp_seq=4 ttl=49 time=12.3 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.279/12.405/12.617/0.137 ms
root@Project-02-Task-01 ~# who am i
root      tty1      Mar  8 16:22
root@Project-02-Task-01 ~#
root@Project-02-Task-01 ~#
```





✓ 创建虚拟机并完成CentOS的安装

- 在VirtualBox中创建虚拟机，完成CentOS操作系统安装。
- 完成CentOS系统升级，并配置远程访问。
- 通过Termuis和SSH进行远程访问开展后续学习

虚拟机配置	操作系统配置
虚拟机名称: VM-Project-02-Task-01-10.10.2.103 内存: 1024MB CPU: 1颗 1核心 虚拟磁盘: 10GB 网卡: 1块, 桥接	主机名: Project-02-Task-01 IP地址: 10.10.2.103 子网掩码: 255.255.255.0 网关: 10.10.2.1 DNS: 8.8.8.8



1.系统基本配置

1.1 hostnamectl

- 通过hostnamectl命令可查询和更改主机名。

命令详解:

【语法】

hostnamectl [选项] [参数]

【选项】

--transient	设置临时主机名
--static	设置静态主机名
--pretty	设置自定义主机名 (可包含特殊字符)
status	显示当前主机名设置
set-hostname	设置系统主机名
set-icon-name	设置主机的图标名称
set-chassis	设置主机的机箱类型
set-deployment	设置主机的部署环境

【参数】

名称 指定要设置的名称

操作命令+配置文件+脚本程序+结束



```

Last login: Mon Mar  8 11:39:18 2021
[root@Project-02-Task-01 ~]# hostname
Project-02-Task-01
[root@Project-02-Task-01 ~]# hostnamectl status
  Static hostname: Project-02-Task-01
        Icon name: computer-vm
      Chassis: vm
    Machine ID: d244317154e049468dcd4e1370367d32
      Boot ID: f2f1d391939844afb9a575554203349d
  Virtualization: vmware
  Operating System: CentOS Linux 8
    CPE OS Name: cpe:/o:centos:centos:8
      Kernel: Linux 4.18.0-240.10.1.el8_3.x86_64
  Architecture: x86_64

```

```

[root@Project-02-Task-01 ~]# hostnamectl --help
hostnamectl [OPTIONS...] COMMAND ...

Query or change system hostname.

-h --help          Show this help
--version          Show package version
--no-ask-password Do not prompt for password
-H --host=[USER@]HOST Operate on remote host
-M --machine=CONTAINER Operate on local container
--transient        Only set transient hostname
--static           Only set static hostname
--pretty           Only set pretty hostname

Commands:
  status           Show current hostname settings
  set-hostname NAME Set system hostname
  set-icon-name NAME Set icon name for host
  set-chassis NAME Set chassis type for host
  set-deployment NAME Set deployment environment for host
  set-location NAME Set location for host

```

```

[root@Project-02-Task-01 ~]# hostnamectl set-hostname vm-hostname
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# hostnamectl status
  Static hostname: vm-hostname
        Icon name: computer-vm
      Chassis: vm
    Machine ID: d244317154e049468dcd4e1370367d32
      Boot ID: f2f1d391939844afb9a575554203349d
  Virtualization: vmware
  Operating System: CentOS Linux 8
    CPE OS Name: cpe:/o:centos:centos:8
      Kernel: Linux 4.18.0-240.10.1.el8_3.x86_64
  Architecture: x86_64
[root@Project-02-Task-01 ~]#

```

小贴士:

1. hostnamectl status 查看的内容项介绍如下:

- Static hostname: 静态主机名为 Project-02-Task-01
- Icon name: 图标名称为 computer-vm
- Chassis: 底层硬件环境为 vm
- Machine ID: 虚拟机 ID 为 1a02991cf169414da531350b6d888cc3
- Boot ID: 引导程序 ID 为 4cd6124402ab4ee9bf83ded3075528d2
- Virtualization: 虚拟化类型为 oracle
- Operating System: 操作系统为 CentOS Linux 8 (Core)
- CPE OS Name: CPE 的操作系统名称为 cpe:/o:centos:centos:8
- Kernel: 内核为 Linux 4.18.0-147.5.1.el8_1.x86_64
- Architecture: 系统架构为 x86-64

1.系统基本配置

1.2 timedatectl

- 通过timedatectl命令可查看和设置系统的时区与时间，也可实现与远程NTP服务器系统时钟的自动同步。

命令详解:

【语法】

timedatectl [选项] [参数]

【选项】

-p --property = NAME	仅显示此名称的属性
-a --all	显示所有属性，包括空属性
--value	显示属性时，仅打印值
status	显示当前时间设置
show	显示 systemd-timedated 的属性
set-time TIME	设置系统时间
set-timezone ZONE	设置系统时区
list-timezones	显示系统中可设置的时区

【参数】

时间日期格式	指定设置日期时间
时区	指定设置的时区名称



```
Last login: Mon Mar  8 16:22:25 2021
```

```
[root@Project-02-Task-01 ~]# timedatectl
Local time:  2021-03-09 18:11:23 CST
Universal time: 2021-03-09 10:11:23 UTC
RTC time:    2021-03-09 10:11:18
Time zone:   Asia/Shanghai (CST, +0800)
System clock synchronized: no
NTP service: n/a
RTC in local TZ: no
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# timedatectl --help
timedatectl [OPTIONS...] COMMAND ...
```

Query or change system time and date settings.

```
-h --help           Show this help message
--version          Show package version
--no-pager         Do not pipe output into a pager
--no-ask-password Do not prompt for password
-H --host=[USER@]HOST Operate on remote host
-M --machine=CONTAINER Operate on local container
--adjust-system-clock Adjust system clock when changing local RTC mode
--monitor          Monitor status of systemd-timesyncd
-p --property=NAME Show only properties by this name
-a --all           Show all properties, including empty ones
--value           When showing properties, only print the value
```

Commands:

```
status             Show current time settings
show               Show properties of systemd-timedated
set-time TIME      Set system time
set-timezone ZONE Set system time zone
list-timezones     Show known time zones
set-local-rtc BOOL Control whether RTC is in local time
set-ntp BOOL       Enable or disable network time synchronization
```

systemd-timesyncd Commands:

```
timesync-status   Show status of systemd-timesyncd
show-timesync     Show properties of systemd-timesyncd
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# timedatectl list-timezones
```

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Bamako
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Brazzaville
Africa/Bujumbura
Africa/Cairo
Africa/Casablanca
Africa/Ceuta
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
```

```
[root@Project-02-Task-01 ~]# timedatectl list-timezones | grep "Asia/S"
Asia/Sakhalin
Asia/Samarkand
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Srednekolymsk
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# timedatectl set-timezone Asia/Shanghai
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# timedatectl
          Local time: 二 2021-03-09 18:17:16 CST
          Universal time: 二 2021-03-09 10:17:16 UTC
          RTC time: 二 2021-03-09 10:17:10
          Time zone: Asia/Shanghai (CST, +0800)
System clock synchronized: no
          NTP service: n/a
          RTC in local TZ: no
```

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# timedatectl list-timezones | grep "Asia/C"
Asia/Chita
Asia/Choibalsan
Asia/Colombo
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# timedatectl set-timezone Asia/Chita
[root@Project-02-Task-01 ~]# timedatectl
          Local time: 二 2021-03-09 19:18:16 +09
          Universal time: 二 2021-03-09 10:18:16 UTC
          RTC time: 二 2021-03-09 10:18:10
          Time zone: Asia/Chita (+09, +0900)
System clock synchronized: no
          NTP service: n/a
          RTC in local TZ: no
[root@Project-02-Task-01 ~]# timedatectl set-timezone Asia/Shanghai
[root@Project-02-Task-01 ~]# timedatectl
          Local time: 二 2021-03-09 18:18:24 CST
          Universal time: 二 2021-03-09 10:18:24 UTC
          RTC time: 二 2021-03-09 10:18:19
          Time zone: Asia/Shanghai (CST, +0800)
System clock synchronized: no
          NTP service: n/a
          RTC in local TZ: no
```

```
[root@Project-02-Task-01 ~]# timedatectl set-time '2021-04-09 20:20:20'
[root@Project-02-Task-01 ~]# timedatectl
          Local time: 五 2021-04-09 20:20:22 CST
          Universal time: 五 2021-04-09 12:20:22 UTC
          RTC time: 五 2021-04-09 12:20:23
          Time zone: Asia/Shanghai (CST, +0800)
System clock synchronized: no
          NTP service: n/a
          RTC in local TZ: no
[root@Project-02-Task-01 ~]# date
2021年 04月 09日 星期五 20:20:32 CST
[root@Project-02-Task-01 ~]# timedatectl set-time '2021-03-09 18:19:20'
[root@Project-02-Task-01 ~]# date
2021年 03月 09日 星期二 18:19:22 CST
[root@Project-02-Task-01 ~]#
```


1.系统基本配置

1.3 localectl

- 通过localectl命令可查看与设置程序运行的语言环境。

命令详解:

【语法】

localectl [选项] [参数]

【选项】

status	显示当前区域设置
set-locale LOCALE	设置系统语言环境
list-locales	显示已知的语言环境
set-keymap MAP [MAP]	设置控制台和 X11 键盘映射
list-keymaps	显示已知的虚拟控制台键盘映射

【参数】

语言格式 指定设置语言格式

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# localectl
System Locale: LANG=zh_CN.UTF-8
VC Keymap: cn
X11 Layout: cn
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# localectl --help
localectl [OPTIONS...] COMMAND ...
```

Query or change system locale and keyboard settings.

```
-h --help           Show this help
--version          Show package version
--no-pager         Do not pipe output into a pager
--no-ask-password Do not prompt for password
-H --host=[USER@]HOST Operate on remote host
-M --machine=CONTAINER Operate on local container
--no-convert       Don't convert keyboard mappings
```

Commands:

```
status             Show current locale settings
set-locale LOCALE... Set system locale
list-locales       Show known locales
set-keymap MAP [MAP] Set console and X11 keyboard mappings
list-keymaps       Show known virtual console keyboard mappings
set-x11-keymap LAYOUT [MODEL [VARIANT [OPTIONS]]]
                  Set X11 and console keyboard mappings
list-x11-keymap-models Show known X11 keyboard mapping models
list-x11-keymap-layouts Show known X11 keyboard mapping layouts
list-x11-keymap-variants [LAYOUT]
                  Show known X11 keyboard mapping variants
list-x11-keymap-options Show known X11 keyboard mapping options
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# localectl list-locales
C.utf8
zh_CN
zh_CN.gb18030
zh_CN.gbk
zh_CN.utf8
zh_HK
zh_HK.utf8
zh_SG
zh_SG.gbk
zh_SG.utf8
zh_TW
zh_TW.euctw
zh_TW.utf8
```

```
[root@Project-02-Task-01 ~]# localectl status
```

```
System Locale: LANG=zh_CN.UTF-8
VC Keymap: cn
X11 Layout: cn
```

```
[root@Project-02-Task-01 ~]#
```

2.查看系统信息

2.1 who

- 通过who命令可查看当前登录到系统中用户的信息。
 - who命令只显示直接登录到系统中的用户，不显示通过su命令的切换用户的登录者。

命令详解:

【语法】

who [选项] [参数]

【选项】

-a	显示全部信息
-b	显示系统最近启动时间
-d	显示已杀死的进程
-H	显示各字段的标题信息
-l	查看系统登录进程
-p, --process	显示由 init 进程衍生的活动进程
-q, --count	列出所有已登录用户的登录名与用户数量
-t	显示系统上次锁定时间
-u	显示已登录用户列表

【参数】

文件 指定要查询的文件

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# who
root pts/0 2021-03-09 18:10 (10.10.0.1)
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# who --help
用法: who [选项]... [文件 | 参数1 参数2 ]
显示当前已登录的用户信息。
```

```
-a, --all 等于-b -d --login -p -r -t -T -u 选项的组合
-b, --boot 上次系统启动时间
-d, --dead 显示已死的进程
-H, --heading 输出头部的标题列
-l, --login 显示系统登录进程
--lookup 尝试通过 DNS 查验主机名
-m 只面对和标准输入有直接交互的主机和用户
-p, --process 显示由 init 进程衍生的活动进程
-q, --count 列出所有已登录用户的登录名与用户数量
-r, --runlevel 显示当前的运行级别
-s, --short 只显示名称、线路和时间(默认)
-T, -w, --mesg 用+, - 或 ? 标注用户消息状态
-u, --users 列出已登录的用户
--message 等于-T
--writable 等于-T
--help 显示此帮助信息并退出
--version 显示版本信息并退出
```

If FILE is not specified, use /var/run/utmp. /var/log/wtmp as FILE is common.
If ARG1 ARG2 given, -m presumed: 'am i' or 'mom likes' are usual.

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>
请向 <http://translationproject.org/team/zh_CN.html> 报告 who 的翻译错误
完整文档请见: <<https://www.gnu.org/software/coreutils/who>>
或者在本地使用: info '(coreutils) who invocation'

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# who -b
系统引导 2021-03-08 11:47
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# who -q
root
# 用户数=1
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# who --users
root pts/0 2021-03-09 18:10 . 6100 (10.10.0.1)
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# who -u
root pts/0 2021-03-09 18:10 . 6100 (10.10.0.1)
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

2.查看系统信息

2.2 id

- 通过id命令可查看用户的ID，以及所属群组的ID。
 - 显示用户以及所属群组的实际与有效ID，若有效ID等于实际ID，则仅显示实际ID。

命令详解：

【语法】

id [选项] [参数]

【选项】

-g	显示用户所属群组 ID
-G	显示用户所属附加群组 ID
-n	显示用户、所属群组或附加群组的名称
-r	显示实际 ID
-u	显示用户 ID

【参数】

用户名 指定要查看的用户名

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# id
uid=0(root) gid=0(root) 组=0(root) 环境=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# id --help
Usage: id [OPTION]... [USER]
Print user and group information for the specified USER,
or (when USER omitted) for the current user.
```

```
-a          ignore, for compatibility with other versions
-Z, --context print only the security context of the process
-g, --group  print only the effective group ID
-G, --groups print all group IDs
-n, --name  print a name instead of a number, for -ugG
-r, --real  print the real ID instead of the effective ID, with -ugG
-u, --user  print only the effective user ID
-z, --zero  delimit entries with NUL characters, not whitespace;
            not permitted in default format
--help     显示此帮助信息并退出
--version  显示版本信息并退出
```

如果不附带任何选项，程序会显示一些可供识别用户身份的有用信息。

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>

请向 <http://translationproject.org/team/zh_CN.html> 报告 id 的翻译错误

完整文档请见: <<https://www.gnu.org/software/coreutils/id>>

或者在本地使用: info '(coreutils) id invocation'

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# id -g
```

```
0
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# id -G
```

```
0
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# id -Z
```

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# █
```

2.查看系统信息

2.3 uname

- 通过uname命令可查看系统相关信息。
 - 如内核版本号、硬件架构、主机名称和操作系统类型等。
 - 相似功能命令hostname，其可用于查看系统的主机名和临时修改系统主机名。

命令详解:

【语法】

uname [选项]

【选项】

-a	显示所有系统的相关信息
-m	显示主机硬件名
-n	显示主机在网络节点上的名称
-r	显示 Linux 操作系统内核版本号
-s	显示 Linux 内核名称
-p	显示处理器类型或 unknown
-i	显示硬件平台类型或 unknown
-o	显示操作系统名

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# uname
Linux
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# uname --help
```

用法: `uname [选项]...`

输出一组系统信息。如果不跟随选项, 则视为只附加 `-s` 选项。

```
-a, --all          以如下次序输出所有信息。其中若 -p 和
                  -i 的探测结果不可知则被省略:
-s, --kernel-name  输出内核名称
-n, --nodename     输出网络节点上的主机名
-r, --kernel-release 输出内核发行号
-v, --kernel-version 输出内核版本
-m, --machine      输出主机的硬件架构名称
-p, --processor    输出处理器类型 (不可移植)
-i, --hardware-platform 输出硬件平台或 (不可移植)
-o, --operating-system 输出操作系统名称
--help            显示此帮助信息并退出
--version         显示版本信息并退出
```

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>

请向 <http://translationproject.org/team/zh_CN.html> 报告 `uname` 的翻译错误

完整文档请见: <<https://www.gnu.org/software/coreutils/uname>>

或者在本地使用: `info '(coreutils) uname invocation'`

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -a
Linux Project-02-Task-01 4.18.0-240.15.1.el8_3.x86_64 #1 SMP Mon Mar 1 17:16:16 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -s
```

```
Linux
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -r
```

```
4.18.0-240.15.1.el8_3.x86_64
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -m
```

```
x86_64
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -p
```

```
x86_64
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -i
```

```
x86_64
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uname -o
```

```
GNU/Linux
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# █
```


2.查看系统信息

2.4 date

- 通过date命令可查看或设置系统时间与日期。
 - 用户可根据需求设置显示的时间格式。

命令详解:

【语法】

date [选项] [参数]

【选项】

-d<字符串>	显示字符串所指的日期与时间
-r	显示文件或目录的最后修改时间
-s	设置时间, 由 STRING 描述
-u	显示或设置世界标准时间 (UTC)

【参数】

+时间日期格式 指定显示的日期时间格式

操作命令+配置文件+脚本程序+结束



```
Last login: Tue Mar  9 18:10:51 2021 from 10.10.0.1
```

```
[root@Project-02-Task-01 ~]# date
```

```
2021年 03月 09日 星期二 19:01:51 CST
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# date +%Y-%m-%d"
```

```
2021-03-09
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# date +%c"
```

```
2021年03月09日 星期二 19时02分12秒
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# date +%Y%m%d%H%M%S"
```

```
202103091902261615287746
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# date +%Y%m%d%H%M%S".bak.tar.gz
```

```
202103091902401615287760.bak.tar.gz
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# maridb-date-$(date +%Y%m%d%H%M%S").bak.tar.gz
```

```
-bash: maridb-date-202103091903051615287785.bak.tar.gz: 未找到命令
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# echo "maridb-date-$(date +%Y%m%d%H%M%S").bak.tar.gz
```

```
maridb-date-202103091903271615287807.bak.tar.gz
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]#
```

【时间格式】

%H	小时，24小时制 (00~23)
%I	小时，12小时制 (01~12)
%k	小时，24小时制 (0~23)
%l	小时，12小时制 (1~12)
%M	分钟 (00~59)
%p	显示出 AM 或 PM
%r	显示 12 小时制时间 (hh:mm:ss)
%s	从 1970 年 1 月 1 日 00:00:00 到目前经历的秒数
%S	显示秒 (00~59)
%T	显示 24 小时制时间 (hh:mm:ss)
%X	显示时间的格式 (%H:%M:%S)
%Z	显示时区，日期域 (CST)
%a	星期的简称 (Sun~Sat)
%A	星期的全称 (Sunday~Saturday)
%h, %b	月的简称 (Jan~Dec)
%B	月的全称 (January~December)
%c	日期和时间 (格式如 Tue Nov 20 14:12:58 2020)
%d	当前月的第几天 (01~31)
%x, %D	日期 (mm/dd/yy)
%j	一年的第几天 (001~366)
%m	月份 (01~12)
%w	当前星期的第几天 (0 代表星期天)
%W	当前年的第几个星期 (00~53, 星期一为第一天)
%y	年的最后两个数字 (如 1970 则是 70)

2.查看系统信息

- 通过ip命令可查看或操作Linux主机的路由、网络设备和隧道等。
 - ip是Linux下较新且功能强大的网络配置工具， ip addr用于查看IP地址。

命令详解:

【语法】

ip [选项] [参数]

【选项】

-s	输出更详细的信息
-f	强制使用指定的协议族
-4	指定使用的网络层协议是 IPv4 协议
-6	指定使用的网络层协议是 IPv6 协议
-0	输出信息每条记录输出一行，不换行显示
-r	显示主机时，不使用 IP 地址，而使用主机的域名
link	网络设备
address	一个设备的协议 (IP 或者 IPV6) 地址，可简写为 addr
neighbour	ARP 或者 NDISC 缓冲区条目
route	路由表条目
rule	路由策略数据库中的规则
maddress	多播地址
mroute	多播路由缓冲区条目
tunnel	IP 上的通道

操作命令+配置文件+脚本程序+结束



```

[root@Project-02-Task-01 ~]# ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [-force] [-batch filename]
where OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
               tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
               netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
               vrf | sr | nexthop | mptcp }
       OPTIONS := {-V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                  -h[uman-readable] | -i[ec] | -j[son] | -p[retty] |
                  -f[amily] { inet | inet6 | mpls | bridge | link } |
                  -4 | -6 | -I | -D | -M | -B | -0 |
                  -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                  -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                  -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                  -c[olor]}

[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ip addr
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: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:9a:f6:33 brd ff:ff:ff:ff:ff:ff
    inet 10.10.2.103/24 brd 10.10.2.255 scope global noprefixroute ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9a:f633/64 scope link
        valid_lft forever preferred_lft forever

[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ip route
default via 10.10.2.1 dev ens192 proto static metric 100
10.10.2.0/24 dev ens192 proto kernel scope link src 10.10.2.103 metric 100

[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ip maddr
1:
    lo
        inet 224.0.0.1
        inet6 ff02::1
        inet6 ff01::1
2:
    ens192
        link 01:00:5e:00:00:01
        link 33:33:00:00:00:01
        link 33:33:ff:9a:f6:33
        inet 224.0.0.1
        inet6 ff02::1:ff9a:f633
        inet6 ff02::1
        inet6 ff01::1

[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#

```

1、ip addr 命令查看网络信息的字段说明

- | | |
|---------------------------|-----------------|
| ● lo | 环回网卡 |
| ● enp0s3 | 网卡名称 |
| ● LOOPBACK | 环回接口 |
| ● BROADCAST | 网卡有广播地址，可以发送广播包 |
| ● UP | 网卡为开启状态 |
| ● LOWER_UP | 网线连接正常 |
| ● mtu:1500 | 最大传输单元为 1500 字节 |
| ● qdisc fq_codel | 队列规则为流队列控制延迟 |
| ● state UP | 状态开启 |
| ● group default qlen 1000 | 组默认为 1000 |
| ● link/ether | MAC 地址 |
| ● inet | IPv4 地址信息 |
| ● scope global | 该网卡可接受全域的包 |
| ● brd | 广播地址 |
| ● noprefixroute enp0s3 | 无前缀路由 |
| ● valid_lft | 正常使用时长 |
| ● preferred_lft | 优先使用时长 |
| ● inet6 | IPv6 地址信息 |

2.查看系统信息

2.6 dmidecode

- 通过dmidecode命令可查看主机的DMI
 - (Desktop Management Interface, 桌面管理接口) 信息,
 - 其输出的信息包括BIOS、系统、主板、处理器、内存、缓存等。

命令详解:

【语法】

dmidecode [选项] [参数]

【选项】

-d	从设备文件读取信息(默认值为/dev/mem)
-s	只显示指定的 DMI 字符串信息
-t	只显示指定条目的信息
-u	显示未解码的原始条目内容
--dump-bin file	将 DMI 信息转储到一个二进制文件中
--from-dump FILE	从一个二进制文件读取 DMI 信息

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# dmidecode --help
```

```
Usage: dmidecode [OPTIONS]
```

```
Options are:
```

```
-d, --dev-mem FILE    Read memory from device FILE (default: /dev/mem)
-h, --help            Display this help text and exit
-q, --quiet          Less verbose output
-s, --string KEYWORD Only display the value of the given DMI string
-t, --type TYPE       Only display the entries of given type
-H, --handle HANDLE  Only display the entry of given handle
-u, --dump           Do not decode the entries
  --dump-bin FILE    Dump the DMI data to a binary file
  --from-dump FILE   Read the DMI data from a binary file
  --no-sysfs         Do not attempt to read DMI data from sysfs files
  --oem-string N     Only display the value of the given OEM string
-V, --version        Display the version and exit
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# dmidecode -t system
```

```
dmidecode: option requires an argument -- 't'
```

```
Type number or keyword expected
```

```
Valid type keywords are:
```

```
bios
system
baseboard
chassis
processor
memory
cache
connector
slot
```

```
[root@Project-02-Task-01 ~]# dmidecode -t system
```

```
# dmidecode 3.2
```

```
Getting SMBIOS data from sysfs.
```

```
SMBIOS 2.7 present.
```

```
Handle 0x0001, DMI type 1, 27 bytes
```

```
System Information
```

```
Manufacturer: VMware, Inc.
Product Name: VMware7,1
Version: None
Serial Number: VMware-42 1a 2b e9 da 14 6a e4-e8 3a 6d b1 85 fa f7 64
UUID: e92b1a42-14da-e46a-e83a-6db185faf764
Wake-up Type: Power Switch
SKU Number: Not Specified
Family: Not Specified
```

```
Handle 0x0097, DMI type 23, 13 bytes
```

```
System Reset
```

```
Status: Enabled
Watchdog Timer: Present
Boot Option: Do Not Reboot
Boot Option On Limit: Do Not Reboot
Reset Count: Unknown
Reset Limit: Unknown
Timer Interval: Unknown
Timeout: Unknown
```

```
Handle 0x009A, DMI type 32, 11 bytes
```

```
System Boot Information
```

```
Status: No errors detected
```

2.查看系统信息

2.7 lspci

- 通过lspci命令可查看主机的PCI总线设备或连接到总线上的设备。
 - 其显示的信息来自于/proc/bus/pci。

命令详解:

【语法】

lspci [选项]

【选项】

-v	以冗余模式显示所有设备的详细信息
-n	以数字形式显示 PCI 生产厂商和设备号
-x	以十六进制显示 PCI 配置空间的前 64 个字节
-b	以总线为中心的视图，实现 PCI 卡上所有的终端号和地址
-t	以树形方式显示包含所有总线、桥、设备和连接图表
-f	当没有任何操作需要完成时，不显示任何信息
-D	测试模式，所作修改不写入配置文件
-d	仅显示给定厂商和设备的信息
-s	仅显示指定总线、插槽上的设备和设备上的功能块信息

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# lspci -v | tail -n 48
02:01.0 SATA controller: VMware SATA AHCI controller (prog-if 01 [AHCI 1.0])
    DeviceName: sata0
    Subsystem: VMware SATA AHCI controller
    Physical Slot: 33
    Flags: bus master, 66MHz, fast devsel, latency 64, IRQ 59
    Memory at fc000000 (32-bit, non-prefetchable) [size=4K]
    Expansion ROM at fc010000 [disabled] [size=64K]
    Capabilities: [40] Power Management version 3
    Capabilities: [48] MSI: Enable+ Count=1/1 Maskable- 64bit+
    Capabilities: [60] SATA HBA v1.0
    Capabilities: [70] PCI Advanced Features
    Kernel driver in use: ahci
    Kernel modules: ahci

03:00.0 Serial Attached SCSI controller: VMware PVSCSI SCSI Controller (rev 02)
    DeviceName: SCSI0
    Subsystem: VMware PVSCSI SCSI Controller
    Physical Slot: 160
    Flags: bus master, fast devsel, latency 0, IRQ 18
    I/O ports at 3000 [size=8]
    Memory at fe900000 (64-bit, non-prefetchable) [size=32K]
    Expansion ROM at fe910000 [disabled] [size=64K]
    Capabilities: [40] Express Endpoint, MSI 00
    Capabilities: [7c] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [94] Power Management version 3
    Capabilities: [9c] MSI-X: Enable+ Count=24 Masked-
    Capabilities: [100] Device Serial Number da-14-6a-e0-50-05-05-69
    Kernel driver in use: vmw_pvscsi
    Kernel modules: vmw_pvscsi

0b:00.0 Ethernet controller: VMware VMXNET3 Ethernet Controller (rev 01)
    DeviceName: Ethernet0
    Subsystem: VMware VMXNET3 Ethernet Controller
    Physical Slot: 192
    Flags: bus master, fast devsel, latency 0, IRQ 19
    Memory at fe103000 (32-bit, non-prefetchable) [size=4K]
    Memory at fe102000 (32-bit, non-prefetchable) [size=4K]
    Memory at fe100000 (32-bit, non-prefetchable) [size=8K]
    I/O ports at 2000 [size=16]
    Expansion ROM at fe110000 [disabled] [size=64K]
    Capabilities: [40] Power Management version 3
    Capabilities: [48] Express Endpoint, MSI 00
    Capabilities: [84] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [9c] MSI-X: Enable+ Count=25 Masked-
    Capabilities: [100] Device Serial Number 00-50-56-ff-9a-f6-33
    Kernel driver in use: vmxnet3
    Kernel modules: vmxnet3

[root@Project-02-Task-01 ~]#
```


3.用户权限管理

3.1 用户与用户组

- Linux系统的用户分为三种：超级用户、普通用户、系统用户（也称伪用户、程序用户）。
 - 超级用户：
 - 即root用户，是Linux系统中默认的超级用户账号，对本主机拥有最高及完整的权限。
 - 只有当进行系统管理、维护任务时，才建议使用root用户登录系统，日常系统操作建议使用普通用户账号。
 - root用户对应的UID为0。



3.用户权限管理

3.1 用户与用户组

- Linux系统的用户分为三种：超级用户、普通用户、系统用户（也称伪用户、程序用户）。
 - 普通用户：
 - 普通用户账号需要由超级用户创建，拥有的权限受到一定限制，一般只在用户的家目录（个人目录）中有完全权限。
 - 普通用户对应的UID范围为1000-65535。
 - 系统用户：
 - 在安装Linux系统及部分应用程序时，会添加一些特定的低权限用户账号，这些用户一般不允许登录到系统，而仅用于维持系统或某个程序的正常运行。
 - 例如：bin、daemon、ftp、mail等，对应的UID范围为1~200（系统分配给进程使用）、201-999（运行服务的用户，动态分配）。



配置文件路径: /etc/login.defs

```
Last login: Tue Mar  9 19:01:46 2021 from 10.10.0.1
[root@Project-02-Task-01 ~]# cat /etc/login.defs
#
# Please note that the parameters in this configuration file control the
# behavior of the tools from the shadow-utils component. None of these
# tools uses the PAM mechanism, and the utilities that use PAM (such as the
# passwd command) should therefore be configured elsewhere. Refer to
# /etc/pam.d/system-auth for more information.
#
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, relative to the
# home directory.  If you _do_ define both, MAIL_DIR takes precedence.
# QMAIL_DIR is for Qmail
#
#QMAIL_DIR      Maildir
MAIL_DIR        /var/spool/mail
#MAIL_FILE      .mail

# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.
UMASK           022

# HOME_MODE is used by useradd(8) and newusers(8) to set the mode for new
# home directories.
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
HOME_MODE       0700

# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_MIN_LEN     Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN         1000
UID_MAX         60000
# System accounts
SYS_UID_MIN     201
SYS_UID_MAX     999

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN         1000
GID_MAX         60000
# System accounts
```

3.用户权限管理

3.2 权限

- Linux是多用户多任务操作系统，为了保护系统和用户的数据安全，Linux系统对用户访问文件或目录的权限规则的定义如下：
 - 将文件的访问权限划分为3种：可读（r）、可写（w）、可执行（x）；
 - 将文件的访问者划分为3类：所有者（u）、同群组的用户（g）、其他组用户（o）；
 - 用户对文件可以独立设置权限。



3.用户权限管理

3.2 权限

表 2-0-1 权限表示方法

八进制	二进制	文件目录权限	权限描述
0	000	---	无权限
1	001	--x	执行
2	010	-w-	写入
3	011	-wx	写入执行
4	100	r--	读取
5	101	r-x	读取执行
6	110	rw-	读取写入
7	111	rwx	读取写入执行

表 2-0-2 权限规则定义的对象

选项	说明
u	user, 文件或目录的所有者
g	group, 文件或目录的所属群组
o	other, 除了文件或目录所有者或所属群组之外的用户
a	all, 即全部的用户
r	读权限
w	写权限
x	执行或切换权限, 数字代号为“1”
-	无任何权限, 数字代号为“0”
s	特殊功能说明: 变更文件或目录的权限



```
[root@Project-02-Task-01 ~]# ls -l /var/log/
总用量 1460
drwxr-xr-x. 2 root root 4096 2月 26 15:56 anaconda
drwx----- 2 root root 23 2月 26 15:59 audit
-rw----- 1 root root 0 3月 9 03:15 boot.log
-rw----- 1 root root 13809 3月 8 10:43 boot.log-20210308
-rw----- 1 root root 13527 3月 9 03:15 boot.log-20210309
-rw----- 1 root utmp 0 2月 26 15:49 btmp
-rw-rw---- 1 root utmp 0 2月 26 15:49 btmp-20210308
-rw-rw---- 1 root root 11879 3月 15 17:01 cron
-rw-rw---- 1 root root 3611 3月 8 10:01 cron-20210308
-rw-rw---- 1 root root 44086 3月 14 03:01 cron-20210314
-rw-rw---- 1 root root 19676 3月 15 17:02 dnf.librepo.log
-rw-rw---- 1 root root 23412 3月 8 09:59 dnf.librepo.log-20210308
-rw-rw---- 1 root root 79366 3月 14 03:35 dnf.librepo.log-20210314
-rw-rw---- 1 root root 265023 3月 15 17:02 dnf.log
-rw-rw---- 1 root root 19709 3月 15 17:02 dnf.rpm.log
-rw-rw---- 1 root root 1116 3月 8 11:47 firewallld
-rw-rw---- 1 root root 624 3月 15 16:02 hawkey.log
-rw-rw---- 1 root root 308 3月 8 09:59 hawkey.log-20210308
-rw-rw---- 1 root root 2545 3月 14 03:35 hawkey.log-20210314
-rw-rw---- 1 root utmp 291708 3月 15 17:20 lastlog
-rw-rw---- 1 root root 0 3月 14 03:45 maillog
-rw-rw---- 1 root root 0 2月 26 15:50 maillog-20210308
-rw-rw---- 1 root root 0 3月 8 10:43 maillog-20210314
-rw-rw---- 1 root root 17731 3月 15 17:20 messages
-rw-rw---- 1 root root 398374 3月 8 10:04 messages-20210308
-rw-rw---- 1 root root 452468 3月 14 03:35 messages-20210314
drwx----- 2 root root 6 2月 26 15:49 private
-rw-rw---- 1 root root 338 3月 15 17:20 secure
-rw-rw---- 1 root root 1467 3月 8 09:49 secure-20210308
-rw-rw---- 1 root root 4442 3月 9 19:24 secure-20210314
-rw-rw---- 1 root root 0 3月 14 03:45 spooler
-rw-rw---- 1 root root 0 2月 26 15:50 spooler-20210308
-rw-rw---- 1 root root 0 3月 8 10:43 spooler-20210314
drwxr-x--- 2 sssd sssd 270 3月 14 03:45 sssd
drwxr-xr-x. 2 root root 23 1月 5 00:24 tuned
-rw-rw---- 1 root root 775 2月 26 15:59 vmware-network.1.log
-rw-rw---- 1 root root 775 3月 8 09:49 vmware-network.log
-rw-rw---- 1 root root 6450 3月 8 11:47 vmware-vgauthsvc.log.0
-rw-rw---- 1 root root 11515 3月 9 18:19 vmware-vmstvc-root.log
-rw-rw---- 1 root root 696 3月 8 11:47 vmware-vmtoolsd-root.log
-rw-rw---- 1 root utmp 15360 3月 15 17:20 wtmp
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# ls -l /var/log/
总用量 1460
drwxr-xr-x. 2 root root 4096 2月 26 15:56 anaconda
drwx----- 2 root root 23 2月 26 15:59 audit
-rw----- 1 root root 0 3月 9 03:15 boot.log
-rw----- 1 root root 13809 3月 8 10:43 boot.log-20210308
-rw----- 1 root root 13527 3月 9 03:15 boot.log-20210309
```

解释 “drwxr-xr-x” 由两部分组成。

第一列：“d” 表示文件类型为目录（目录或文件夹）。

其他列：“rwxr-xr-x” 表示文件权限。

权限分为三段：

2-4位

“rwx”：表示文件所有者的权限。

5-7位

“r-x”：文件所属组的权限。

8-10位

“r-x”：其他用户对文件的权限。

3.用户权限管理

3.3 用户组管理

- 通过groupadd命令可创建用户组。
 - 使用该命令创建的用户信息保存在/etc/group文件中。

命令详解:

【语法】

groupadd [选项] [参数]

【选项】

-g	指定创建用户组 ID
-r	创建系统用户组，系统用户组的 ID 小于 500 (该值由系统配置文件决定)
-o	允许用户组的 ID 不唯一

【参数】

用户组名	指定创建的用户组名
------	-----------

操作命令+配置文件+脚本程序+结束



3.用户权限管理

3.3 用户组管理

- 通过groupmod命令可修改用户组的名称或ID。

命令详解:

【语法】

groupmod [选项] [参数]

【选项】

-g<用户组 ID>	设置要使用的用户组 ID
-o	重复使用用户组 ID
-n<新用户组名称>	设置欲使用的用户组名称

【参数】

用户组名	指定要修改的用户组名
------	------------

操作命令+配置文件+脚本程序+结束



3.用户权限管理

3.3 用户组管理

- 通过groupdel命令可删除指定的用户组。
 - 如果该组下有用户，则必需先删除用户，才可删除该组。

命令详解:

【语法】

groupdel [参数]

【参数】

用户组名

指定要删除的用户组名

操作命令+配置文件+脚本程序+结束



配置文件路径: /etc/group

```
[root@Project-02-Task-01 ~]# cat /etc/group
```

```
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:
cdrom:x:11:
mail:x:12:
man:x:15:
dialout:x:18:
floppy:x:19:
games:x:20:
tape:x:33:
video:x:39:
ftp:x:50:
lock:x:54:
audio:x:63:
users:x:100:
nobody:x:65534:
dbus:x:81:
utmp:x:22:
utempter:x:35:
input:x:999:
kvm:x:36:
render:x:998:
systemd-journal:x:190:
systemd-coredump:x:997:
systemd-resolve:x:193:
tss:x:59:
polkitd:x:996:
ssh_keys:x:995:
unbound:x:994:
sssd:x:993:
sshd:x:74:
rngd:x:992:
```

```
[root@Project-02-Task-01 ~]# groupadd demogroup01
[root@Project-02-Task-01 ~]# groupadd -g 500 demogrpup02
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
demogroup01:x:1000:
demogrpup02:x:500:
[root@Project-02-Task-01 ~]#
```

配置文件路径: /etc/group

```
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
demogroup01:x:1000:
demogrup02:x:500:
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# groupmod -g 501 demogroup01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
demogroup01:x:501:
demogrup02:x:500:
[root@Project-02-Task-01 ~]# groupmod -g 502 -n demogroup03 demogroup01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
demogrup02:x:500:
demogroup03:x:502:
[root@Project-02-Task-01 ~]# groupdel demogroup01 demogroup02
用法: groupdel [选项] 组

选项:
  -h, --help          显示此帮助信息并推出
  -R, --root CHROOT_DIR  chroot 到的目录
  -P, --prefix PREFIX_DIR  prefix directory where are located the /etc/* files
  -f, --force          delete group even if it is the primary group of a user

[root@Project-02-Task-01 ~]# groupdel demogroup02
groupdel: "demogroup02"组不存在
[root@Project-02-Task-01 ~]# groupdel demogroup03
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
demogrup02:x:500:
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# groupdel demo*
groupdel: "demo*"组不存在
[root@Project-02-Task-01 ~]# groupdel demogrup02
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/group | grep demo
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

3.用户权限管理

3.4 用户管理

- 通过useradd命令可创建系统用户。
 - 使用该命令所创建的用户信息保存在/etc/passwd文件中。

命令详解:

【语法】

useradd [选项] [参数]

【选项】

-c <备注>	加上备注文字, 备注文字会保存在 passwd 的备注栏位中
-d <登入目录>	指定用户登入时的启始目录
-D	变更预设值
-e <有效期限>	指定帐号的有效期限
-f <缓冲天数>	指定在密码过期后多少天即禁用该帐号
-g <群组>	指定用户所属群组
-G <群组>	指定用户所属附加群组
-m	自动创建用户的登入目录
-M	不要自动创建用户的登入目录
-n	取消创建以用户名称为名的群组
-r	创建系统帐号
-s<Shell>	指定用户登入后所使用的 Shell
-u<uid>	指定用户 uid

【参数】

用户名 指定要创建的用户名



3.用户权限管理

3.4 用户管理

- 通过passwd命令可设置用户的认证信息，包括用户密码及过期时间等。
 - 具有超级用户权限的用户可通过该命令管理其他用户的密码。

命令详解:

【语法】

passwd [选项] [参数]

【选项】

-d	删除密码，拥有超级用户权限才能使用
-f	强制执行
-k	设置只有在密码过期失效后，才能更新
-l	锁定密码
-s	列出密码的相关信息，拥有超级用户权限才能使用
-u	解除密码锁定

【参数】

用户名 指定要修改密码的用户名

操作命令+配置文件+脚本程序+结束



3.用户权限管理

3.4 用户管理

- 通过usermod命令可修改用户的基本信息。
 - 该命令不允许修改当前登录用户，如果用户有执行的程序，则无法修改用户UID。

命令详解：

【语法】

usermod [选项] [参数]

【选项】

-c <备注>	修改用户名的备注信息
-d <登入目录>	修改用户登入目录
-e <有效期限>	修改用户名的有效期限
-f <缓冲天数>	修改在密码过期后多少天禁用该用户
-g <群组>	修改用户所属群组
-G <群组>	修改用户所属附加群组
-l <用户名>	修改用户名
-L	锁定用户密码
-s <Shell>	修改用户登入后所使用的 Shell
-u <uid>	修改用户 UID
-U	解除密码锁定

【参数】

用户名 指定要修改信息的用户名

操作命令+配置文件+脚本程序+结束



3.用户权限管理

3.4 用户管理

- 通过userdel命令可删除用户，以及与用户相关的文件及目录。
 - 如果不加选项，则仅删除用户帐号，而不删除相关文件及目录。

命令详解：

【语法】

userdel [选项] [参数]

【选项】

-f	强制删除用户帐号
-r	删除用户同时删除其主目录

【参数】

用户名	指定要修改信息的用户名
-----	-------------

操作命令+配置文件+脚本程序+结束



配置文件路径: /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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rngd:x:995:992:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# useradd demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# useradd -r demouser01
useradd: 用户“demouser01”已存在
[root@Project-02-Task-01 ~]# useradd -r demouser02
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
drwx-----. 2 demouser01 demouser01 62 3月 15 19:28 demouser01
[root@Project-02-Task-01 ~]# cat /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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rngd:x:995:992:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
demouser01:x:1000:1000:./home/demouser01:/bin/bash
demouser02:x:994:991:./home/demouser02:/bin/bash

```


配置文件路径: /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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel OverFlow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rngd:x:995:992:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# useradd demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# useradd -r demouser01
useradd: 用户“demouser01”已存在
[root@Project-02-Task-01 ~]# useradd -r demouser02
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
drwx-----. 2 demouser01 demouser01 62 3月 15 19:28 demouser01
[root@Project-02-Task-01 ~]# cat /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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel OverFlow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rngd:x:995:992:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
demouser01:x:1000:1000:./home/demouser01:/bin/bash
demouser02:x:994:991:./home/demouser02:/bin/bash

```

/etc/passwd文件中的信息由“:”隔开，共分为7段，例如“root:x:0:0:root:/root:/bin/bash”。

- root 用户名称
- x 密码占位符，存放账户的口令，用x表示，密码保存在/etc/shadow中
- 0 用户的UID
- 0 用户基本组GID
- root 用户的详细信息
- /root root用户的家目录，普通用户的家目录在/home下
- /bin/bash 用户登录的Shell

```
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
drwx----- . 2 demouser01 demouser01 62 3月 15 19:28 demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/passwd | grep demo
demouser01:x:1000:1000::/home/demouser01:/bin/bash
demouser02:x:994:991::/home/demouser02:/bin/bash
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# cat /etc/shadow | grep demo
demouser01:!:18701:0:99999:7:::
demouser02:!:18701:0:0:0:0:
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# passwd demouser01
更改用户 demouser01 的密码。
新的 密码:
无效的密码: 密码少于 8 个字符
重新输入新的 密码:
抱歉, 密码不匹配。
新的 密码:
无效的密码: 密码未通过字典检查 - 太简单或太有规律
重新输入新的 密码:
抱歉, 密码不匹配。
新的 密码:
重新输入新的 密码:
passwd: 所有的身份验证令牌已经成功更新。
[root@Project-02-Task-01 ~]# cat /etc/shadow | grep demo
demouser01:$6$S0y0NaaswJmHm.S$9qX32s2kAa0jKbUqKY1MmY4ysI2cGY0eASrU3jRngofU56Uh465RzdTLsn6oTBO3nh1.HDtwRm0iuCUFt.YfU1:18701:0:99999:7:::
demouser02:!:18701:0:0:0:0:
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# userdel demouser02
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# userdel demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
drwx----- . 2 1000 1000 62 3月 15 19:28 demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# userdel -r demouser01
userdel: 用户“demouser01”不存在
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /home
总用量 0
drwx----- . 2 1000 1000 62 3月 15 19:28 demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# rm -rf /home/demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# █
```

3.用户权限管理

- 通过chmod命令可更改文件或目录的访问权限。该命令有两种操作方式：
 - 一种是包含字母和操作符表达式的文字设定法
 - 另一种是包含数字的数字设定法

命令详解:

【语法】

chmod [选项] [参数]

【选项】

-c	显示命令执行过程, 仅显示更改部分
-f	强制执行, 不显示错误信息
-R	递归处理, 将指定目录下的所有文件及子目录一并处理
-v	显示命令执行过程
<权限范围>+<权限设置>	增加权限范围的文件或目录的权限设置
<权限范围>-<权限设置>	取消权限范围的文件或目录的权限设置
<权限范围>=<权限设置>	指定权限范围的文件或目录的权限设置

【参数】

权限模式	指定文件的权限模式
文件	指定要改变权限的文件



3.用户权限管理

- 通过chown命令可修改文件或目录的属主或属组。

命令详解:

【语法】

chown [选项] [参数]

【选项】

- v 显示属主修改的详细信息
- f 强制执行，不显示错误信息
- h 改变符号链接文件的属主时，不影响该链接所指向的目标文件
- c 若该文件或目录属主已更改，才显示其更改动作
- R 以递归方式修改当前目录下的所有文件与子目录

【参数】

- 用户属主 指定属主和属组。当省略“:组”，则仅改变文件所有者
- 文件 指定要改变属主和属组的文件列表。支持多个文件和目录，支持 Shell 通配符

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# mkdir demodir
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# touch demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-rw-r--r--. 1 root root 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chmod 0644 demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-rw-r--r--. 1 root root 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chmod ug+x, o-r demodir/file1.txt
chmod: 无效模式: "ug+x,"
请尝试执行 "chmod --help" 来获取更多信息。
[root@Project-02-Task-01 ~]# chmod ug+x,o-r demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-rwxr-x---. 1 root root 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chmod =r demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-r--r--r--. 1 root root 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# useradd -r demouser01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chown demouser01 demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-r--r--r--. 1 demouser01 root 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chown demouser01:demouser01 demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-r--r--r--. 1 demouser01 demouser01 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# groupadd demogroup01
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# chgrp demogroup01 demodir/file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-r--r--r--. 1 demouser01 demogroup01 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# █
```

3.用户权限管理

3.5 权限管理

- 通过umask命令可设置新建文件或目录的权限。

命令详解:

【语法】

umask [选项] [参数]

【选项】

- p 输出的权限掩码可直接作为指令来执行
- s 以符号方式输出权限掩码

【参数】

- 权限掩码 指定权限掩码

操作命令+配置文件+脚本程序+结束



```

[root@Project-02-Task-01 ~]# ls -l demodir
总用量 0
-r--r--r--. 1 demouser01 demogroup01 0 3月 15 19:44 file1.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# touch demodir demofile02.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# mkdir demodir02
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l
总用量 8
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# umask
0022
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# umask 000
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# touch demofile03.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# mkdir demodir03
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l
总用量 8
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
drwxrwxrwx. 2 root root 6 3月 15 19:57 demodir03
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
-rw-rw-rw-. 1 root root 0 3月 15 19:57 demofile03.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#

```

umask值与权限

umask值	文件	目录
0	6	7
1	6	6
2	4	5
3	4	4
4	2	3
5	2	2
6	0	1
7	0	0

常用的umask及对应的文件与目录权限

umask值	文件	目录
022	644	755
027	640	750
002	664	775
006	660	771
007	660	770

4.文件目录操作

4.1 创建目录

- 通过mkdir命令可创建目录。
 - 在创建目录时，应保证创建的目录与它所在目录下的目录没有重名。

命令详解：

【语法】

mkdir [选项] [参数]

【选项】

-p	递归创建多级目录
-m	创建目录的同时设置目录的权限
-v	显示目录的创建过程

【参数】

目录名 指定要创建的目录列表，多个目录之间用空格隔开

操作命令+配置文件+脚本程序+结束



4.文件目录操作

4.2 创建文件

- 通过touch命令可创建文件，该命令还可改变文件的访问时间和修改时间。

命令详解:

【语法】

touch [选项] [参数]

【选项】

-a	改变文件的访问时间为系统当前时间
-c	如果文件不存在，就不创建也不提示
-m	改变文件的修改时间为系统当前时间
-d, -i	使用指定的日期时间
-r<参考文件或目录>	把指定文件或目录的日期时间设置为参考文件或目录的日期时间

【参数】

文件名 指定要设置时间属性的文件列表

操作命令+配置文件+脚本程序+结束



4.文件目录操作

4.3 文件目录操作

- 通过mv命令可将文件移至一个目标位置，或将一组文件移至一个目标目录。

命令详解:

【语法】

mv [选项] [参数]

【选项】

-b, --backup	当文件存在时，执行覆盖前为其创建备份
-f	若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录
-i	交互式操作，覆盖前先行询问用户
-S<后缀>	为备份文件指定后缀，而不使用默认的后缀
--target-directory=<目录>	指定源文件要移动到目标目录
-u	当源文件的修改日期在目标文件之后或者目标文件不存在时，执行移动操作

【参数】

源文件或目录	源文件列表
目标文件	在移动文件的同时，将其改名为目标文件
目标目录	将源文件移动到目标目录下

操作命令+配置文件+脚本程序+结束



4.文件目录操作

4.3 文件目录操作

- 通过cp命令可复制文件或目录到目标位置。复制目录时，需使用-R选项。

命令详解：

【语法】

cp [选项] [参数]

【选项】

-a	此参数的效果和同时指定"-dpR"参数相同
-d	当复制符号链接时，把目标文件或目录也创建为符号链接，并指向与源文件或目录链接的原始文件或目录
-f	强制复制文件或目录
-i	交互式操作，覆盖前先行询问用户
-l	对源文件创建硬连接，而非复制文件
-p	保留源文件或目录的属性
-R/r	以递归方式将指定目录下的所有文件与子目录一并处理
-s	对源文件创建符号链接
-b	当文件存在时，执行覆盖前为其创建备份

【参数】

源文件或目录	指定源文件列表
目标文件或目录	指定目标文件。当“源文件”为多个文件时，要求“目标文件”为指定的目录



4.文件目录操作

4.3 文件目录操作

- 通过rm命令可删除文件或目录。
 - 该命令可删除一个目录中的一个或多个的文件或目录。链接文件则只断开链接，源文件保持不变。

命令详解:

【语法】

rm [选项] [参数]

【选项】

-f	强制删除文件或目录
-i	删除已有文件或目录之前先询问用户
-r 或-R	递归处理，将指定目录下的所有文件与子目录一并处理
--preserve-root	不对根目录进行递归操作
-v	显示命令的详细执行过程

【参数】

文件 指定被删除的文件列表，如果参数中含有目录，则必须加上-r 或-R 选项

操作命令+配置文件+脚本程序+结束



4.文件目录操作

4.3 文件目录操作

- rmdir命令用来删除目录，可在一个目录中删除一个或多个空的子目录；
 - 删除目录时，必须具有对其父目录的写权限，并且其子目录被删除之前应该是空目录；
 - 当前工作目录必须在被删除目录之上，不能是被删除目录本身，也不能是被删除目录的子目录。



4.文件目录操作

4.4 查看文件信息

- 通过file命令可查看文件类型，也可辨别一些文件的编码格式。
 - 通过查看文件头部信息来获取文件类型。通过ls -l命令查看文件和目录的权限信息。

命令详解：

【语法】

file [选项] [参数]

【选项】

-b	列出文件辨识结果时，不显示文件名称
-c	详细显示命令执行过程，便于排错或分析程序执行的情形
-f	列出文件中文件名的文件类型
-F	使用指定分隔符号替换输出文件名后的默认的“:”分隔符
-L	查看对应软链接对应文件的文件类型
-z	解读压缩文件的内容
-i	显示文件的 MIME 类别

【参数】

文件	指定要查看类型的文件列表，多个文件之间使用空格分开，可以使用 Shell 通配符匹配多个文件
目录	指定要查看类型的目录名称

操作命令+配置文件+脚本程序+结束



```

[root@Project-02-Task-01 ~]# cd /opt
[root@Project-02-Task-01 opt]# ls -l
总用量 0
[root@Project-02-Task-01 opt]# mkdir demodir01
[root@Project-02-Task-01 opt]# mkdir demodir02
[root@Project-02-Task-01 opt]# mkdir -p demodir03/subdir03-01
[root@Project-02-Task-01 opt]# mkdir -v demodir04
mkdir: 已创建目录 'demodir04'
[root@Project-02-Task-01 opt]# mkdir -pv demodir05/{subdir05-01,subdir05-02}
mkdir: 已创建目录 'demodir05'
mkdir: 已创建目录 'demodir05/subdir05-01'
mkdir: 已创建目录 'demodir05/subdir05-02'
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# touch demofile01.txt
[root@Project-02-Task-01 opt]# touch -c -t 05012021 demofile01.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# ls -l
总用量 0
drwxrwxrwx. 2 root root 6 3月 15 20:17 demodir01
drwxrwxrwx. 2 root root 6 3月 15 20:17 demodir02
drwxrwxrwx. 3 root root 25 3月 15 20:18 demodir03
drwxrwxrwx. 2 root root 6 3月 15 20:18 demodir04
drwxrwxrwx. 4 root root 44 3月 15 20:19 demodir05
-rw-rw-rw-. 1 root root 0 5月 1 2021 demofile01.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# mv demofile01.txt demodir01
[root@Project-02-Task-01 opt]# cp demodir01/demofile01.txt demodir02/demofile02.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# rm demodir01/demofile01.txt
rm: 是否删除普通空文件 'demodir01/demofile01.txt'? y
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# rm -f demodir02/demofile02.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# rm -f demodir02/demofile02.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# rm demodir02/demofile02.txt
rm: 无法删除'demodir02/demofile02.txt': 没有那个文件或目录
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# touch demodir01/demofile01.txt
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# file demodir01/demofile01.txt
demodir01/demofile01.txt: empty
[root@Project-02-Task-01 opt]# file /var/log/
anaconda/          dnf.librepo.log-20210308 maillog-20210314      sssd/
audit/             dnf.librepo.log-20210314 messages             tuned/
boot.log           dnf.log                messages-20210308    vmware-network.1.log
boot.log-20210308 dnf.rpm.log            messages-20210314    vmware-network.log
boot.log-20210309 firewallld              private/             vmware-vgauthsvc.log.0
btm                hawkey.log             secure               vmware-vmvc-root.log
btm-20210308       hawkey.log-20210308   secure-20210308     vmware-vmtoolsd-root.log
cron               hawkey.log-20210314   secure-20210314     wtmp
cron-20210308     lastlog                 spooler
cron-20210314     maillog                 spooler-20210308
dnf.librepo.log    maillog-20210308      spooler-20210314
[root@Project-02-Task-01 opt]# file /var/log/messages
/var/log/messages: UTF-8 Unicode text
[root@Project-02-Task-01 opt]# █

```

```

[root@Project-02-Task-01 opt]# touch demofile.txt
[root@Project-02-Task-01 opt]# touch demofile.mp3
[root@Project-02-Task-01 opt]# touch demofile.docx
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# ls -l
总用量 0
drwxrwxrwx. 2 root root 28 3月 15 20:23 demof1r01
drwxrwxrwx. 2 root root 6 3月 15 20:22 demof1r02
drwxrwxrwx. 3 root root 25 3月 15 20:18 demof1r03
drwxrwxrwx. 2 root root 6 3月 15 20:18 demof1r04
drwxrwxrwx. 4 root root 44 3月 15 20:19 demof1r05
-rw-rw-rw-. 1 root root 0 3月 15 20:31 demofile.docx
-rw-rw-rw-. 1 root root 0 3月 15 20:31 demofile.mp3
-rw-rw-rw-. 1 root root 0 3月 15 20:31 demofile.txt
-rw-rw-rw-. 1 root root 0 3月 15 20:25 welcome
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# file demofile.docx
demofile.docx: empty
[root@Project-02-Task-01 opt]# file demofile.mp3
demofile.mp3: empty
[root@Project-02-Task-01 opt]# file demofile.txt
demofile.txt: empty
[root@Project-02-Task-01 opt]# echo 'welcome'>demofile.txt
[root@Project-02-Task-01 opt]# file demofile.txt
demofile.txt: ASCII text
[root@Project-02-Task-01 opt]# wget http://web.book.51xueweb.cn/anli2/06/medias/DaiWoDaoShanDing.mp3
--2021-03-15 20:32:15-- http://web.book.51xueweb.cn/anli2/06/medias/DaiWoDaoShanDing.mp3
正在解析主机 web.book.51xueweb.cn (web.book.51xueweb.cn)... 101.200.49.211
正在连接 web.book.51xueweb.cn (web.book.51xueweb.cn)[101.200.49.211]:80... 已连接。
已发出 HTTP 请求，正在等待响应... 200 OK
长度：4580009 (4.4M) [audio/mpeg]
正在保存至：“DaiWoDaoShanDing.mp3”

```

```

DaiWoDaoShanDing.mp3 100%[=====] 4.37M 129KB/s 用时 36s

```

```

2021-03-15 20:32:51 (125 KB/s) - 已保存 “DaiWoDaoShanDing.mp3” [4580009/4580009]

```

```

[root@Project-02-Task-01 opt]# file DaiWoDaoShanDing.mp3
DaiWoDaoShanDing.mp3: Audio file with ID3 version 2.3.0, contains:MPEG ADTS, layer III, v1, 192 kbps, 44.1 kHz, JntStereo
[root@Project-02-Task-01 opt]#
[root@Project-02-Task-01 opt]# ls -l
总用量 4480
-rw-rw-rw-. 1 root root 4580009 10月 26 2018 DaiWoDaoShanDing.mp3
drwxrwxrwx. 2 root root 28 3月 15 20:23 demof1r01
drwxrwxrwx. 2 root root 6 3月 15 20:22 demof1r02
drwxrwxrwx. 3 root root 25 3月 15 20:18 demof1r03
drwxrwxrwx. 2 root root 6 3月 15 20:18 demof1r04
drwxrwxrwx. 4 root root 44 3月 15 20:19 demof1r05
-rw-rw-rw-. 1 root root 0 3月 15 20:31 demofile.docx
-rw-rw-rw-. 1 root root 0 3月 15 20:31 demofile.mp3
-rw-rw-rw-. 1 root root 8 3月 15 20:32 demofile.txt
-rw-rw-rw-. 1 root root 0 3月 15 20:25 welcome
[root@Project-02-Task-01 opt]#

```


5. 文本处理操作

5.1 ls

- 通过ls命令可查看目录列表以及查看文件或目录的权限信息等详细信息，ls命令的输出信息可进行彩色加亮显示，以区分不同类型的文件或目录。

命令详解:

【语法】

ls [选项] [参数]

【选项】

-a	显示所有文件，包括隐藏文件（以“.”或“.”开头的文件）
-A	显示除隐藏文件以外的所有文件
-b	将文件中的不可输出字符以反斜线“\”加字符编码的方式输出
-C	多列显示输出结果
-d	显示目录名，而不显示目录下的内容列表； 显示符号链接文件本身，而不显示其所指向的目录列表
-F	在每个输出项后追加文件的类型标识符
-k	以 KB（千字节）为单位显示文件大小
-l	以长格式显示目录下的内容列表的详细信息
-r	以文件名反序排列并输出目录内容列表
-R	以递归指定目录下的所有文件及子目录一并处理
-s	显示文件和目录的大小，以块为单位
-S	以文件的大小进行排序
-t	用文件和目录的更改时间排序
-X	根据扩展名排序
-1	每行只列出一个文件

【参数】

目录 指定要显示列表的目录，也可以是具体的文件



```
Last login: Mon Mar 15 17:20:55 2021 from 10.10.0.1
[root@Project-02-Task-01 ~]# ls
anaconda-ks.cfg date demodir demodir02 demodir03 demofile02.txt demofile03.txt
[root@Project-02-Task-01 ~]# ls -l
总用量 8
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
drwxrwxrwx. 2 root root 6 3月 15 19:57 demodir03
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
-rw-rw-rw-. 1 root root 0 3月 15 19:57 demofile03.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l /var/log/se*
-rw-----. 1 root root 4789 3月 16 19:52 /var/log/secure
-rw-----. 1 root root 1467 3月 8 09:49 /var/log/secure-20210308
-rw-----. 1 root root 4442 3月 9 19:24 /var/log/secure-20210314
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -lt /var/log/se*
-rw-----. 1 root root 4789 3月 16 19:52 /var/log/secure
-rw-----. 1 root root 4442 3月 9 19:24 /var/log/secure-20210314
-rw-----. 1 root root 1467 3月 8 09:49 /var/log/secure-20210308
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -ltk /var/log/se*
-rw-----. 1 root root 4789 3月 16 19:52 /var/log/secure
-rw-----. 1 root root 4442 3月 9 19:24 /var/log/secure-20210314
-rw-----. 1 root root 1467 3月 8 09:49 /var/log/secure-20210308
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -lsk /var/log/se*
-rw-----. 1 root root 4789 3月 16 19:52 /var/log/secure
-rw-----. 1 root root 4442 3月 9 19:24 /var/log/secure-20210314
-rw-----. 1 root root 1467 3月 8 09:49 /var/log/secure-20210308
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

5. 文本处理操作

5.2 文本信息查看

- 通过cat命令可用于查看纯文本内容，通常使用cat查看一屏即显示完的短文本。

命令详解：

【语法】

cat [选项] [参数]

【选项】

-b	在每一行行首显示行号，跳过空白行
-E	在每行的结束处显示\$
-n	在每一行行首显示行号，包括空白行
-T	将 TAB 符显示为^I
-v	显示不可打印字符

【参数】

文件列表 指定要连接的文件列表

【快捷键】

Ctrl+S	停止滚屏
Ctrl+Q	恢复滚屏
Ctrl+C	终止该命令的执行，并返回 Shell 提示符状态

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- 通过more命令可分页查看较长内容的文本，同时支持关键字定位查看。

命令详解:

【语法】

more [选项] [参数]

【选项】

+<数字>	从指定数字行开始显示
-<数字>	指定每屏显示的行数
+/pattern	从 pattern 前两行开始显示
-c	不进行滚屏操作，每次刷新整个屏幕
-l	忽略 Ctrl+l (换页) 字符
-s	将连续的多个空行压缩成一行显示
-u	把文件内容中的下划线去掉

【参数】

文件	指定分页显示内容的文件
----	-------------

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- 通过more命令可分页查看较长内容的文本，同时支持关键字定位查看。

命令详解:

【语法】

more [选项] [参数]

【快捷键】

Space	显示文本的下一屏内容
Enter	只显示文本的下一行内容
h	显示帮助屏，该屏上有相关的帮助信息
b	显示上一屏内容
q	退出 more 命令

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- less的作用与more十分相似，不同点为less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。

命令详解：

【语法】

less [选项] [参数]

【选项】

-e	文件内容显示完毕后，自动退出
-f	强制显示文件
-g	不加亮显示搜索到的所有关键词
-l	搜索时忽略大小写
-N	在每一行行首显示行号
-s	把连续多个空白行合并成一个空白行
-S	在单行显示较长的内容，而不换行显示

【参数】

文件 指定要分屏显示内容的文件

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- less的作用与more十分相似，不同点为less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。

命令详解：

【语法】

less [选项] [参数]

【命令内部键】

b	向后翻一页
d	向后翻半页
h	显示帮助界面
u	向前滚动半页
Space	向后滚动一页
Enter	向后滚动一行
y	向前滚动一行
j	向前移动一行
k	向后移动一行
G	移动到最后一行
g	移动到第一行

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- less的作用与more十分相似，不同点为less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。

命令详解：

【语法】

less [选项] [参数]

【命令内部键】

/	使用一个模式进行搜索，并定位到下一个匹配的文本
n	向前查找下一个匹配的文本
N	向后查找前一个匹配的文本
?	使用模式进行搜索，并定位到前一个匹配的文本
q/Shift+zz	退出 less 命令
v	进入编辑模式，使用配置的编辑器编辑当前文件
ma	使用 a 标记文本的当前位置
'a	导航到标记 a 处

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- less的作用与more十分相似，不同点为less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。

命令详解:

【语法】

less [选项] [参数]

【快捷键】

Ctrl +	向前移动一屏
Ctrl + B	向后移动一屏
Ctrl + D	向前移动半屏
Ctrl + U	向后移动半屏

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- 通过head命令可查看文件的开头内容，默认显示头部10行内容。

命令详解:

【语法】

head [选项] [参数]

【选项】

- n <数字> 指定显示头部内容的行数
- c <字符数> 指定显示头部内容的字符数
- v 显示文件名的头信息
- q 不显示文件名的头信息

【参数】

- 文件列表 指定显示头部内容的文件列表

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.2 文本信息查看

- 通过tail命令可查看文件的尾部内容，默认显示尾部10行内容。

命令详解:

【语法】

tail [选项] [参数]

【选项】

- c <数字> 指定显示尾部内容的行数
- f 显示文件最新追加的内容
- n <数字> 指定显示头部内容的字符数
- v 当有多个文件参数时，仅输出各文件名
- q 当有多个文件参数时，不输出各个文件名

【参数】

- 文件列表 指定显示尾部内容的文件列表

操作命令+配置文件+脚本程序+结束




```
Mar 14 03:45:01 Project-02-Task-01 rsyslogd[1213]: [origin software="rsyslogd" swVersion="8.1911.0-6.el8" x-pid="1213" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 14 04:36:03 Project-02-Task-01 dnf[8963]: 元数据缓存近期已刷新。
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 14 05:37:03 Project-02-Task-01 dnf[8984]: 元数据缓存近期已刷新。
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
Mar 14 06:38:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
```

```
--更多-- (2%)
```

```
[root@Project-02-Task-01 ~]# head /var/log/messages
Mar 14 03:45:01 Project-02-Task-01 rsyslogd[1213]: [origin software="rsyslogd" swVersion="8.1911.0-6.el8" x-pid="1213" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 14 04:36:03 Project-02-Task-01 dnf[8963]: 元数据缓存近期已刷新。
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 14 05:37:03 Project-02-Task-01 dnf[8984]: 元数据缓存近期已刷新。
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 14 05:37:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
Mar 14 06:38:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# head -v -n 2 /var/log/messages
==> /var/log/messages <==
Mar 14 03:45:01 Project-02-Task-01 rsyslogd[1213]: [origin software="rsyslogd" swVersion="8.1911.0-6.el8" x-pid="1213" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Mar 14 04:36:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# tail /var/log/messages
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Listening on D-Bus User Message Bus Socket.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Sockets.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Basic System.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Default.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Startup finished in 69ms.
Mar 16 19:52:11 Project-02-Task-01 systemd[1]: Started User Manager for UID 0.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 20:23:03 Project-02-Task-01 dnf[19104]: 元数据缓存近期已刷新。
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# tail -v -n 2 /var/log/messages
=> /var/log/messages <
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

tail -f /var/log/secure

5. 文本处理操作

5.3 信息检索与排序

- 通过grep命令可按照设置的匹配规则（或者匹配模式）搜索指定的文件，并显示符合匹配条件的行。

命令详解：

【语法】

grep [选项] [参数]

【选项】

-i	搜索时忽略大小写
-c	仅显示匹配行的数量
-l	仅显示符合匹配的文件名，不显示具体的匹配行
-n	显示所有的匹配行，并显示行号
-h	查询多文件时不显示文件名
-w	匹配整个词
-x	匹配整行
-r	递归搜索
-b	显示匹配行距文件头部的偏移量，以字节为单位
-o	与 b 结合使用，显示匹配词距文件头部的偏移量，以字节为单位

【参数】

匹配模式	指定进行搜索的匹配模式
文件	指定要搜索的文件

操作命令+配置文件+脚本程序+结束



5. 文本处理操作

5.3 信息检索与排序

- 通过sort命令可将文件的每行作为一个单位相互比较，比较原则是从首字符向后，依次按ASCII码值进行，最后按升序输出。

命令详解：

【语法】

sort [选项] [参数]

【选项】

-b	忽略行首的空格字符
-c	检查文件是否已经按照顺序排序
-d	排序时，处理英文字母、数字及空格字符外，忽略其他的字符
-f	排序时，将小写字母视为大写字母
-i	排序时，除了 040 至 176 之间的 ASCII 字符外，忽略其他的字符
-m	将几个排序号的文件进行合并
-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<输出文件>	将排序后的结果存入制定的文件
-r	以相反的顺序进行排序
-t<分隔字符>	指定排序时所用的栏位分隔字符
-k	指定需要排序的栏位

【参数】

文件 指定的待排序的文件列表



5. 文本处理操作

- 通过uniq命令可移除或发现文件中相邻重复行。

命令详解:

【语法】

uniq [选项] [参数]

【选项】

- c 在每列左边显示该行重复出现的次数
- d 仅显示重复出现的行
- f<栏位> 忽略比较指定的栏位
- s<字符> 忽略比较指定的字符
- u 仅显示未重复的行的内容
- w<字符> 指定要比较的字符

【参数】

- 输入文件 指定要去除的重复行文件
- 输出文件 指定要去除的重复行后的内容的写入文件

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# grep '^root' /etc/passwd
root:x:0:0:root:/root:/bin/bash
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# grep -c root /etc/passwd
2
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# tail /var/log/messages
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Listening on D-Bus User Message Bus Socket.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Sockets.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Basic System.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Default.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Startup finished in 69ms.
Mar 16 19:52:11 Project-02-Task-01 systemd[1]: Started User Manager for UID 0.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 20:23:03 Project-02-Task-01 dnf[19104]: 元数据缓存近期已刷新。
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# grep -c dnf /var/log/messages
319
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# grep -c dnf /var/log/
grep: /var/log/: 不是一个目录
0
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# grep -cr dnf /var/log/
/var/log/tastlog.0:0
/var/log/wtmp:0
/var/log/sss/sss_implicit_files.log-20210308.gz:0
/var/log/sss/sss.log-20210309.gz:0
/var/log/sss/sss_nss.log-20210308.gz:0
/var/log/sss/sss_implicit_files.log-20210314:0
/var/log/sss/sss_implicit_files.log:0
/var/log/sss/sss.log-20210314:0
/var/log/sss/sss.log:0
/var/log/sss/sss_nss.log-20210314:0
/var/log/sss/sss_nss.log:0
/var/log/tuned/tuned.log:0
/var/log/audit/audit.log:405
/var/log/anaconda/anaconda.log:0
/var/log/anaconda/syslog:0
/var/log/anaconda/x.log:0
/var/log/anaconda/program.log:0
/var/log/anaconda/packaging.log:107
/var/log/anaconda/storage.log:0
/var/log/anaconda/lvm.log:0
/var/log/anaconda/dnf.librepo.log:0
/var/log/anaconda/hawkey.log:2
/var/log/anaconda/dbus.log:0
/var/log/anaconda/ks-script-_m76uw4q.log:0
/var/log/anaconda/ks-script-kjp94pqa.log:0
/var/log/anaconda/ks-script-y48b6f1.log:0
/var/log/anaconda/journal.log:72
/var/log/boot.log:0
/var/log/vmware-vgauthsvc.log:0:0
/var/log/vmware-vmtoolsd-root.log:0
```

```
[root@Project-02-Task-01 ~]# cat demosort.txt
```

```
A:10:6.1
```

```
C:30:4.3
```

```
D:40:3.4
```

```
B:20:5.2
```

```
F:60:1.6
```

```
F:60:1.6
```

```
E:50:2.5
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# sort demosort.txt
```

```
A:10:6.1
```

```
B:20:5.2
```

```
C:30:4.3
```

```
D:40:3.4
```

```
E:50:2.5
```

```
F:60:1.6
```

```
F:60:1.6
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# sort -n -k 3 -t: demosort.txt
```

```
F:60:1.6
```

```
F:60:1.6
```

```
E:50:2.5
```

```
D:40:3.4
```

```
C:30:4.3
```

```
B:20:5.2
```

```
A:10:6.1
```

```
[root@Project-02-Task-01 ~]# sort -nr -k 3 -t: demosort.txt
```

```
A:10:6.1
```

```
B:20:5.2
```

```
C:30:4.3
```

```
D:40:3.4
```

```
E:50:2.5
```

```
F:60:1.6
```

```
F:60:1.6
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# cat demouniq.txt
```

```
A
```

```
A
```

```
C
```

```
C
```

```
C
```

```
B
```

```
B
```

```
D
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# uniq -c demouniq.txt
```

```
2 A
```

```
3 C
```

```
2 B
```

```
1 D
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# sort demouniq.txt | uniq
```

```
A
```

```
B
```

```
C
```

```
D
```

```
[root@Project-02-Task-01 ~]# sort demouniq.txt | uniq -c
```

```
2 A
```

```
2 B
```

```
3 C
```

```
1 D
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# █
```

5.文本处理操作

5.3 信息检索与排序

- 通过sed命令可自动编辑一个或多个文件、简化对文件的反复操作、编写转换程序等。
 - sed拥有两个数据缓冲区，一个活动的模式空间和一个辅助的暂存空间。
 - sed编辑器工作原理是首先将文本文件的一行内容存储在模式空间中，然后使用内部命令对该行进行处理，处理完成后，将模式空间中的文本显示到标准输出设备上（显示终端），然后处理下一行文本内容，重复此过程，直到文本结束。

<https://www.gnu.org/software/sed/manual/sed.htm>



5. 文本处理操作

5.3 信息检索与排序

- 通过sed命令可自动编辑一个或多个文件、简化对文件的反复操作、编写转换程序等。

命令详解:

【语法】

sed [选项] [参数]

【选项】

-e	直接在命令行模式上进行 sed 动作编辑
-f	将 sed 的动作写在一个文件内
-i	直接修改读取的内容, 而不是输出终端
-n	只打印模式匹配的行
-r	支持扩展表达式

【参数】

文件	指定待处理的文本文件列表
----	--------------

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# cat demosed.txt
Linux - Sysadmin.
Database - Oracle, MySQL etc.
Cool - Websites.
Storage - NetAPP, ENC etc.
Security - Firewall, Network, Online etc.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed '1i *The Start*' demosed.txt
*The Start*
Linux - Sysadmin.
Database - Oracle, MySQL etc.
Cool - Websites.
Storage - NetAPP, ENC etc.
Security - Firewall, Network, Online etc.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed '$a *The End*' demosed.txt
Linux - Sysadmin.
Database - Oracle, MySQL etc.
Cool - Websites.
Storage - NetAPP, ENC etc.
Security - Firewall, Network, Online etc.
*The End*
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed '2a *Hello World*' demosed.txt
Linux - Sysadmin.
Database - Oracle, MySQL etc.
*Hello World*
Cool - Websites.
Storage - NetAPP, ENC etc.
Security - Firewall, Network, Online etc.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed '1i *The Start*' demosed.txt | sed '$a *The End*'
*The Start*
Linux - Sysadmin.
Database - Oracle, MySQL etc.
Cool - Websites.
Storage - NetAPP, ENC etc.
Security - Firewall, Network, Online etc.
*The End*
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed '2, 4c *Hello World*' demosed.txt
Linux - Sysadmin.
*Hello World*
Security - Firewall, Network, Online etc.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# tail -f /var/log/messages
Mar 16 19:52:11 Project-02-Task-01 systemd[9014]: Listening on D-Bus User Message Bus Socket.
Mar 16 19:52:11 Project-02-Task-01 systemd[9014]: Reached target Sockets.
Mar 16 19:52:11 Project-02-Task-01 systemd[9014]: Reached target Basic System.
Mar 16 19:52:11 Project-02-Task-01 systemd[9014]: Reached target Default.
Mar 16 19:52:11 Project-02-Task-01 systemd[9014]: Startup finished in 69ms.
Mar 16 19:52:11 Project-02-Task-01 systemd[1]: Started User Manager for UID 0.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 20:23:03 Project-02-Task-01 dnf[1918]: 元数据缓存近期已刷新。
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

**统计/var/log/messages日志信息
哪些命令产生?
频次如何?**


```
[root@Project-02-Task-01 ~]# tail /var/log/messages
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Listening on D-Bus User Message Bus Socket.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Sockets.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Basic System.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Reached target Default.
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Startup finished in 69ms.
Mar 16 19:52:11 Project-02-Task-01 systemd[1]: Started User Manager for UID 0.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 20:23:03 Project-02-Task-01 dnf[19104]: 元数据缓存近期已刷新。
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed 's/\[.*$/' /var/log/messages | tail
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 dnf
Mar 16 20:23:03 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 systemd
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# tail /var/log/messages
Mar 16 19:52:11 Project-02-Task-01 systemd[19014]: Startup finished in 69ms.
Mar 16 19:52:11 Project-02-Task-01 systemd[1]: Started User Manager for UID 0.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 20:23:03 Project-02-Task-01 dnf[19104]: 元数据缓存近期已刷新。
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 20:23:03 Project-02-Task-01 systemd[1]: Started dnf makecache.
Mar 16 21:23:53 Project-02-Task-01 systemd[1]: Starting dnf makecache...
Mar 16 21:23:53 Project-02-Task-01 dnf[19247]: 元数据缓存近期已刷新。
Mar 16 21:23:53 Project-02-Task-01 systemd[1]: dnf-makecache.service: Succeeded.
Mar 16 21:23:53 Project-02-Task-01 systemd[1]: Started dnf makecache.
```

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed 's/\[.*$/ /' /var/log/messages | tail
```

```
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 19:52:11 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 dnf
Mar 16 20:23:03 Project-02-Task-01 systemd
Mar 16 20:23:03 Project-02-Task-01 systemd
Mar 16 21:23:53 Project-02-Task-01 systemd
Mar 16 21:23:53 Project-02-Task-01 dnf
Mar 16 21:23:53 Project-02-Task-01 systemd
Mar 16 21:23:53 Project-02-Task-01 systemd
```

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed 's/\[.*$/ /' /var/log/messages | sed 's/\{35\}/' | tail
```

```
systemd
systemd
systemd
dnf
systemd
systemd
systemd
dnf
systemd
systemd
```

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# sed 's/\[.*$/ /' /var/log/messages | sed 's/\{35\}/' | sort | uniq -c
```

```
128 dnf
  1 nss
  1 rsyslogd
261 systemd
  4 systemd-logind
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# █
```

5.文本处理操作

5.4 vim

- vi是Linux下标准的文本编辑工具，是Linux系统中内置的编辑器，熟练地使用vi工具可以高效的编辑代码，配置系统文件等，是程序员和运维人员必备的技能。
- vi编辑器有三种基本的工作模式，分别是命令模式、文本编辑模式和末行模式。
 - 命令模式
 - 命令模式是vi命令的默认工作模式，并可转换为文本编辑模式和末行模式。在命令模式下，从键盘上输入的任何字符都被当作命令来解释，而不会在屏幕上显示。
 - 如果输入的字符是合法的vi子命令，则vi就会完成相应的操作。
 - 文本编辑模式
 - 文本编辑模式用于字符编辑。在命令模式下输入i（插入命令）、a（附加命令）等命令后进入文本编辑模式。
 - 按Esc键可从文本编辑模式返回到命令模式。
 - 末行模式
 - 末行模式也称ex转义模式。在命令模式下，按“:”键进入末行模式，此时vi会在屏幕的底部显示“:”符号作为末行模式的提示符，等待用户输入相关命令。
 - 命令执行完毕后，vi自动回到命令模式。



5. 文本处理操作

5.4 vim

- vi是Linux下标准的文本编辑工具，是Linux系统中内置的编辑器。

命令详解：

【语法】

vi [选项] [参数]

【选项】

- +<行号> 从指定行号的行开始显示文本内容
- b 以二进制模式打开文件，用于编辑二进制文件和可执行文件
- c<命令> 在完成对第一个文件编辑任务后，执行给出的命令
- d 以 diff 模式打开文件，当多个文件编辑时，显示文件差异部分
- M 关闭修改功能
- n 不使用缓存功能，将不产生“.swap”文件
- o<文件数量> 同时打开指定数量的文件
- R 以只读方式打开文件
- s 安静模式，不显示命令的任何错误信息

【参数】

- 文件列表 指定要编辑的文件列表，多个文件之间使用空格隔开

操作命令+配置文件+脚本程序+结束



5.文本处理操作

5.4 vim

- vi是Linux下标准的文本编辑工具，是Linux系统中内置的编辑器。

命令详解:

【子命令】

i	在当前字符前插入文本
I	在行首插入文本
a	光标后插入
A	在当前行尾插入
o	在当前行后插入一个空行
O	在当前行前插入一个空行

操作命令+配置文件+脚本程序+结束



5.文本处理操作

5.4 vim

- vi是Linux下标准的文本编辑工具，是Linux系统中内置的编辑器。

命令详解:

【子命令】

dd	删除光标所在行内容
ndd	n 为数字，删除光标所在的向下 n 行，如 20dd 则是删除光标下 20 行的内容
d1G	删除光标所在行至第一行的内容
dG	删除光标所在行到最后一行的内容
d\$	删除光标所在位置至该行的最后一个字符的内容
Ctrl+u	删除输入方式下所输入的文本

操作命令+配置文件+脚本程序+结束



5.文本处理操作

5.4 vim

- vi是Linux下标准的文本编辑工具，是Linux系统中内置的编辑器。

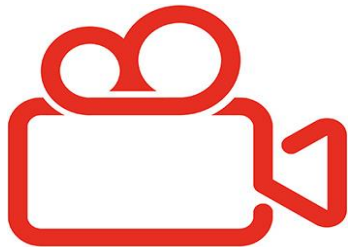
命令详解:

【子命令】

yy	复制当前行
nyy	复制当前行以下的 n 行
dd	剪切当前行
ndd	剪切当前行以下的 n 行
p/P	粘贴在当前光标所在行下(p) 或行上(P)

操作命令+配置文件+脚本程序+结束





- ✓ 使用vim进行文本编辑
 - 使用vi编辑器
 - 命令模式下操作
 - 文本编辑模式下操作
 - 保存退出



5.文本处理操作

5.5 其他的文本处理命令

wc

diff

awk

paste



```
[root@Project-02-Task-01 ~]# wc /var/log/messages
395 3604 33705 /var/log/messages
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# wc --help
```

```
用法: wc [选项]... [文件]...
```

```
或: wc [选项]... --files0-from=F
```

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. A word is a non-zero-length sequence of characters delimited by white space.

如果没有指定文件，或者文件为"-", 则从标准输入读取。

The options below may be used to select which counts are printed, always in the following order: newline, word, character, byte, maximum line length.

-c, --bytes	print the byte counts
-m, --chars	print the character counts
-l, --lines	print the newline counts
--files0-from=F	read input from the files specified by NUL-terminated names in file F; If F is - then read names from standard input
-L, --max-line-length	print the maximum display width
-w, --words	print the word counts
--help	显示此帮助信息并退出
--version	显示版本信息并退出

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>

请向 <http://translationproject.org/team/zh_CN.html> 报告 wc 的翻译错误

完整文档请见: <<https://www.gnu.org/software/coreutils/wc>>

或者在本地使用: info '(coreutils) wc invocation'

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# diff demosed.txt demosort.txt
1,5c1,7
< Linux - Sysadmin.
< Database - Oracle, MySQL etc.
< Cool - Websites.
< Storage - NetAPP, ENC etc.
< Security - Firewall, Network, Online etc.
---
> A:10:6.1
> C:30:4.3
> D:40:3.4
> B:20:5.2
> F:60:1.6
> F:60:1.6
> E:50:2.5
```

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# diff --help
```

```
用法: diff [选项]... 文件们
逐行比较<文件们>。
```

长选项的必需参数也是相应短选项的必需参数。

```
--normal          以正常的 diff 方式输出 (默认)
-q, --brief       只有在文件不同时报告
-s, --report-identical-files 当两个一样时仍然显示结果
-c, -C NUM, --context[=NUM] output NUM (default 3) lines of copied context
-u, -U 数量, --unified[=数量] 输出 <数量> (默认为 3) 行一致化上下文
-e, --ed          以 ed script 方式输出
-n, --rcs        以 RCS diff 格式输出
-y, --side-by-side output in two columns
-W, --width=数量 每行显示最多 <数量> (默认 130) 个字符
--left-column    当有两行相同时只显示左边栏的一行
--suppress-common-lines 当有两行相同时不显示

-p, --show-c-function show which C function each change is in
-F, --show-function-line=RE show the most recent line matching RE
--label LABEL    use LABEL instead of file name and timestamp
                 (can be repeated)

-t, --expand-tabs 将输出中的 tab 转换成空格
-T, --initial-tab 每行先加上 tab 字符, 使 tab 字符可以对齐
--tabsize=数字   TAB 格的宽度, 默认为 8 个打印列宽
--suppress-blank-empty suppress space or tab before empty output lines
-l, --paginate   将输出送至 "pr" 指令来分页

-r, --recursive 连同所有子目录一起比较
--no-dereference don't follow symbolic links
-N, --new-file   不存在的文件以空文件方式处理
--unidirectional-new-file 若第一文件不存在, 以空文件处理
--ignore-file-name-case 忽略文件名大小写的区别
--no-ignore-file-name-case 不忽略文件名大小写的区别
```

https://www.ruanyifeng.com/blog/2012/08/how_to_read_diff.html

```
--to-file=文件2 将操作数中的所有文件/目录和<文件2>作比较;
                 <文件2>可以是目录
```

```
-i, --ignore-case 忽略文件内容大小写的区别
-E, --ignore-tab-expansion 忽略由制表符宽度造成的差异
```

```
[root@Project-02-Task-01 ~]# sed 's/[.]*$//' /var/log/messages | sed 's/.\{35\}//' | sort | uniq -c
128 dnf
  1 nss
  1 rsyslogd
261 systemd
  4 systemd-logind
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# awk -F ' ' '{print $5}' /var/log/messages | sed 's/[.]*$//' | sort | uniq -c
128 dnf
  1 nss
  1 rsyslogd
261 systemd
  4 systemd-logind
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# tail /etc/passwd
dbus:x:81:81:System message bus:/:sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/:sbin/nologin
polkitd:x:998:996:User for polkitd:/:sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/:sbin/nologin
sssd:x:996:993:User for sssd:/:sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/var/lib/ssh:/sbin/nologin
rngd:x:995:992:Random Number Generator Daemon:/var/lib/rngd:/:sbin/nologin
demouser01:x:994:991:./home/demouser01:/bin/bash
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# awk -F ':' '{print $1}' /etc/passwd
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
operator
games
ftp
nobody
dbus
systemd-coredump
systemd-resolve
tss
polkitd
unbound
sssd
sshd
rngd
demouser01
[ro
[ro
```

<https://www.ruanyifeng.com/blog/2018/11/awk.html>

```
[root@Project-02-Task-01 ~]# paste --help
```

```
用法: paste [选项]... [文件]...
```

```
Write lines consisting of the sequentially corresponding lines from
each FILE, separated by TABs, to standard output.
```

如果没有指定文件，或者文件为"-", 则从标准输入读取。

必选参数对长短选项同时适用。

```
-d, --delimiters=列表 改用指定列表里的字符替代制表分隔符
-s, --serial          不使用平行的行目输出模式，而是每个文件占用一行
-z, --zero-terminated 以 NUL 字符而非换行符作为行尾分隔符
--help              显示此帮助信息并退出
--version           显示版本信息并退出
```

GNU coreutils 在线帮助: <<https://www.gnu.org/software/coreutils/>>

请向 <http://translationproject.org/team/zh_CN.html> 报告 paste 的翻译错误

完整文档请见: <<https://www.gnu.org/software/coreutils/paste>>

或者在本地使用: info '(coreutils) paste invocation'

```
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# cat demopaste1.txt
```

```
zhangsan
```

```
lisi
```

```
wangwu
```

```
zhaoliu
```

```
[root@Project-02-Task-01 ~]# cat demopaste2.txt
```

```
T
```

```
F
```

```
T
```

```
T
```

```
[root@Project-02-Task-01 ~]# cat demopaste3.txt
```

```
luoyang
```

```
wuhan
```

```
nanyang
```

```
[root@Project-02-Task-01 ~]# paste demopaste1.txt demopaste2.txt demopaste3.txt
```

```
zhangsan      T      luoyang
```

```
lisi F      wuhan
```

```
wangwu T
```

```
zhaoliu T      nanyang
```

```
[root@Project-02-Task-01 ~]# paste -s demopaste1.txt demopaste2.txt demopaste3.txt
```

```
zhangsan      lisi      wangwu      zhaoliu
```

```
T      F      T      T
```

```
luoyang wuhan      nanyang
```

```
[root@Project-02-Task-01 ~]# paste -z demopaste1.txt demopaste2.txt demopaste3.txt
```

```
zhangsan
```

```
lisi
```

```
wangwu
```

```
zhaoliu
```

```
T
```

```
F
```

```
T
```

```
T
```

```
luoyang
```

```
wuhan
```

```
nanyang
```

```
[root@Project-02-Task-01 ~]#
```

6.磁盘存储管理

6.1 打包压缩

- tar是Linux操作系统下经常用到的归档工具。
 - tar命令用于把多个文件和目录打包成一个文件并归档。
 - 在Linux中很多压缩程序都只能针对一个文件进行压缩，因此要压缩一个目录或多个文件时，需要将其先打包为一个文件，然后再进行压缩。
 - tar通常和压缩工具结合使用。

- 打包与压缩的不同
 - 打包：将多个文件或者目录变为一个文件。
 - 压缩：将一个文件通过压缩算法变为一个更小的文件，占用更少的存储空间。



6. 磁盘存储管理

6.1 打包压缩

□ tar是Linux操作系统下经常用到的归档工具。

命令详解:

【语法】

tar [选项] [参数]

【选项】

-A	新增文件到以存在的备份文件
-B	设置区块大小
-c	建立新的备份文件
-C <目录>	选项用在解压缩, 在特定目录解压缩
-d	记录文件的差别
-x	从备份文件中还原文件
-t	列出备份文件的内容
-z	通过 gzip 指令处理备份文件
-j	支持 bzip2 解压文件
-Z	通过 compress 指令处理备份文件
-f<备份文件>	指定备份文件
-v	显示指令执行过程
-r	添加文件到已经压缩的文件
-u	添加改变了和现有的文件到已经存在的压缩文件
-k	保留原有文件不覆盖
-m	保留文件不被覆盖
-w	确认压缩文件的正确性

【参数】

文件或目录 指定要打包的文件或目录列表



```
[root@Project-02-Task-01 ~]# ls -l
总用量 20
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
drwxrwxrwx. 2 root root 6 3月 15 19:57 demodir03
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
-rw-rw-rw-. 1 root root 0 3月 15 19:57 demofile03.txt
-rw-r--r--. 1 root root 138 3月 16 21:30 demosed.txt
-rw-r--r--. 1 root root 63 3月 16 20:39 demosort.txt
-rw-r--r--. 1 root root 16 3月 16 20:42 demouniq.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# tar -cvf demobak.tar demo*
demodir/
demodir/file1.txt
demodir02/
demodir03/
demofile02.txt
demofile03.txt
demosed.txt
demosort.txt
demouniq.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l
总用量 32
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
-rw-r--r--. 1 root root 10240 3月 16 22:58 demobak.tar
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
drwxrwxrwx. 2 root root 6 3月 15 19:57 demodir03
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
-rw-rw-rw-. 1 root root 0 3月 15 19:57 demofile03.txt
-rw-r--r--. 1 root root 138 3月 16 21:30 demosed.txt
-rw-r--r--. 1 root root 63 3月 16 20:39 demosort.txt
-rw-r--r--. 1 root root 16 3月 16 20:42 demouniq.txt
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```


6.磁盘存储管理

6.1 打包压缩

zip
unzip

gzip
ungzip

bzip
unbzip



```
[root@Project-02-Task-01 ~]# gzip --help
Usage: gzip [OPTION]... [FILE]...
Compress or uncompress FILES (by default, compress FILES in-place).
```

Mandatory arguments to long options are mandatory for short options too.

```
-c, --stdout      write on standard output, keep original files unchanged
-d, --decompress  decompress
-f, --force       force overwrite of output file and compress links
-h, --help        give this help
-k, --keep        keep (don't delete) input files
-l, --list        list compressed file contents
-L, --license     display software license
-n, --no-name     do not save or restore the original name and timestamp
-N, --name        save or restore the original name and timestamp
-q, --quiet       suppress all warnings
-r, --recursive  operate recursively on directories
  --rsyncable    make rsync-friendly archive
-S, --suffix=SUF use suffix SUF on compressed files
--synchronous    synchronous output (safer if system crashes, but slower)
-t, --test       test compressed file integrity
-v, --verbose    verbose mode
-V, --version    display version number
-l, --fast       compress faster
-9, --best       compress better
```

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.

```
[root@Project-02-Task-01 ~]# gzip demobak.tar
[root@Project-02-Task-01 ~]# ls -l | grep demobak.tar.gz
-rw-r--r--. 1 root root 515 3月 16 22:58 demobak.tar.gz
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# gunzip demobak.tar.gz
[root@Project-02-Task-01 ~]# ls -l | grep demobak.tar
-rw-r--r--. 1 root root 10240 3月 16 22:58 demobak.tar
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

```
[root@Project-02-Task-01 ~]# tar -czvf linux-log-bak.$(date +%Y%m%d%H%M%S)s".tar.gz /var/log/m*
tar: 从成员名中删除开头的“/”
/var/log/maillog
tar: 从硬连接目标中删除开头的“/”
/var/log/maillog-20210308
/var/log/maillog-20210314
/var/log/messages
/var/log/messages-20210308
/var/log/messages-20210314
[root@Project-02-Task-01 ~]# tar -czvf linux-log-bak.$(date +%Y%m%d%H%M%S)s".tar.gz /var/log/m*
tar: 从成员名中删除开头的“/”
/var/log/maillog
tar: 从硬连接目标中删除开头的“/”
/var/log/maillog-20210308
/var/log/maillog-20210314
/var/log/messages
/var/log/messages-20210308
/var/log/messages-20210314
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# ls -l
总用量 248
-rw-----. 1 root root 1177 2月 26 15:56 anaconda-ks.cfg
-rw-r--r--. 1 root root 40 3月 9 18:57 date
-rw-r--r--. 1 root root 10240 3月 16 22:58 demobak.tar
drwxr-xr-x. 2 root root 23 3月 15 19:55 demodir
drwxr-xr-x. 2 root root 6 3月 15 19:56 demodir02
drwxrwxrwx. 2 root root 6 3月 15 19:57 demofile
-rw-r--r--. 1 root root 0 3月 15 19:55 demofile02.txt
-rw-rw-rw-. 1 root root 0 3月 15 19:57 demofile03.txt
-rw-r--r--. 1 root root 138 3月 16 21:30 demosed.txt
-rw-r--r--. 1 root root 63 3月 16 20:39 demosort.txt
-rw-r--r--. 1 root root 16 3月 16 20:42 demouniq.txt
-rw-r--r--. 1 root root 107855 3月 16 23:25 linux-log-bak.202103162325231615908323.tar.gz
-rw-r--r--. 1 root root 107855 3月 16 23:25 linux-log-bak.202103162325581615908358.tar.gz
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]#
```

6. 磁盘存储管理

6.2 磁盘管理

- 通过fdisk命令可查看磁盘使用情况，并对磁盘进行分区和格式化。
 - fdisk采用问答式界面进行操作。

命令详解：

【语法】

fdisk [选项] [参数]

【选项】

-b<分区大小>	指定每个分区的大小
-l	列出指定的设备的分区表状况
-s<分区编号>	将指定的分区大小输出到标准输出上，单位为区块
-u	搭配-l参数列表，会用分区数目取代柱面数目，来表示每个分区的起始地址
-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

【参数】

设备文件 指定要进行分区或者显示分区的磁盘设备

操作命令+配置文件+脚本程序+结束



6. 磁盘存储管理

6.2 磁盘管理

- 通过df命令可查看文件系统的磁盘使用情况。

命令详解:

【语法】

df [选项] [参数]

【选项】

-a	显示全部文件系统列表
-h	以合适的单位来显示, 提高可读性
-H	等于“-h”, 但是计算式, 1K=1000, 而不是 1K=1024
-i	用索引节点信息替代磁盘信息
-k	指定区块的大小为 1024 字节
-l	只显示本地文件系统
-m	指定区块大小为 1048576 字节
--sync	在取得磁盘信息前, 先执行 sync 命令
-T	文件系统类型
--no-sync	获取磁盘空间使用情况前不执行磁盘的同步操作
--block-size=<区块大小>	指定区块大小
-t<文件系统类型>	只显示选定文件系统的磁盘信息
-x<文件系统类型>	不显示选定文件系统的磁盘信息

【参数】

文件 指定文件系统上的文件



6.磁盘存储管理

6.2 磁盘管理

- 通过du命令可查看文件和目录的磁盘使用空间。

命令详解:

【语法】

du [选项] [参数]

【选项】

-a	显示目录中个别文件的大小
-b	以 Byte 为单位显示目录或文件大小
-c	除显示个别目录或文件大小外，也显示所有目录或文件的总和
-h	以 KB、MB、GB 为单位，提高信息可读性
-k	以 KB 为单位
-m	以 MB 为单位
-l	重复计算硬件链接的文件

【参数】

文件或目录 指定文件或目录

操作命令+配置文件+脚本程序+结束



6. 磁盘存储管理

6.2 磁盘管理

- 通过mdadm命令可在Linux进行RAID管理，该命令能够诊断、监控和收集详细的磁盘阵列信息。

命令详解：

【语法】

mdadm [选项]

【选项】

-A, --assemble	组装已存在的阵列
-C, --create	创建阵列
-F, --follow, --monitor	选择监控模式
-G, --grow	修改在用阵列的大小或形态
-I, --incremental	在阵列中添加/删除单个磁盘，并可能启动该阵列
-l, --level	指定 RAID 级别
-p, --layout	指定 RAID5、6、10 的奇偶校验规则
-n, --raid-devices	指定阵列中可用 device 数目
-x, --spare-devices	指定初始阵列的富余 device 数目
-z, --size	指定区块大小
-a, --auto{ (=no,yes,md,mdp,part,p)}	是否自动创建 RAID

操作命令+配置文件+脚本程序+结束



6.磁盘存储管理

6.2 磁盘管理

- 文件系统需能够链接到目录才能使用，挂载就是将文件系统与目录结合的操作。
 - 通过mount命令可加载文件系统到指定的挂载点。
 - 通过unmount命令可以卸载指定的挂载点。

命令详解:

【语法】

mount [选项] [参数]

【选项】

-a	挂载安装在fstab中提到的所有文件系统
-f	测试挂载设备。可与-v等参数同时使用以查看mount的执行过程
-l	显示已挂载的文件系统列表
-n	禁止将挂载信息记录在/etc/mtab文件
-r	将文件系统加载为只读模式
-t	指定挂载文件系统类型

【参数】

设备文件名	指定要加载的文件系统对应的设备名
加载点	指定加载点目录

操作命令+配置文件+脚本程序+结束




```
[root@Project-02-Task-01 ~]# fdisk -l
Disk /dev/sda: 16 GiB, 17179869184 字节, 33554432 个扇区
单元: 扇区 / 1 * 512 = 512 字节
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: gpt
磁盘标识符: DBD5D1BC-EC42-47C1-9132-732537C00E78
```

设备	起点	末尾	扇区	大小	类型
/dev/sda1	2048	1230847	1228800	600M	EFI 系统
/dev/sda2	1230848	3327999	2097152	1G	Linux 文件系统
/dev/sda3	3328000	33552383	30224384	14.4G	Linux LVM

```
Disk /dev/mapper/cl-root: 12.8 GiB, 13753122816 字节, 26861568 个扇区
```

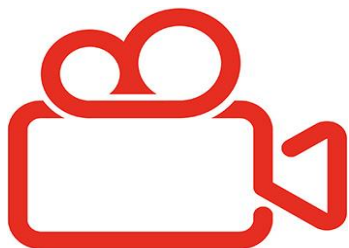
```
单元: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
```

```
Disk /dev/mapper/cl-swap: 1.6 GiB, 1719664640 字节, 3358720 个扇区
```

```
单元: 扇区 / 1 * 512 = 512 字节
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@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# df /
文件系统          1K-块    已用    可用  已用% 挂载点
/dev/mapper/cl-root 13420544 1752568 11667976   14% /
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# df
文件系统          1K-块    已用    可用  已用% 挂载点
devtmpfs          912124      0   912124    0% /dev
tmpfs             930960      0   930960    0% /dev/shm
tmpfs             930960  25144   905816    3% /run
tmpfs             930960      0   930960    0% /sys/fs/cgroup
/dev/mapper/cl-root 13420544 1752568 11667976   14% /
/dev/sda2         1038336  315172   723164   31% /boot
/dev/sda1         613184   7032   606152    2% /boot/efi
tmpfs             186192      0   186192    0% /run/user/0
```

```
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# du /var/log/
0    /var/log/private
28   /var/log/sss
8    /var/log/tuned
360  /var/log/audit
4060 /var/log/anaconda
6016 /var/log/
[root@Project-02-Task-01 ~]# du -h /var/log/
0    /var/log/private
28K  /var/log/sss
8.0K /var/log/tuned
360K /var/log/audit
4.0M /var/log/anaconda
5.9M /var/log/
[root@Project-02-Task-01 ~]#
```



✓ 磁盘管理

- 为虚拟机增加新的磁盘
- 对新的磁盘进行分区和格式化操作
- 对分区进行挂载使用
- 通过Linux实现Raid 1



7.其他操作命令

屏幕
显示

手册
帮助

操作
历史



7.其他操作命令

7.1 手册帮助

手册
帮助

联机文档：man

```
[root@Project-02-Task-01 ~]# cd /usr/share/man
[root@Project-02-Task-01 man]# ls
ca da es hu it ko man1 man1x man2x man3p man4 man5 man6 man7 man8 man9 mann pl ru sv uk zh_TW
cs de fr id ja man0p man1p man2 man3 man3x man4x man5x man6x man7x man8x man9x nl pt_BR sk tr zh_CN
[root@Project-02-Task-01 man]#
```

使用帮助：help

内置命令：help命令

外部命令：--help选项



```
[root@Project-02-Task-01 ~]# compgen --help
```

```
compgen: compgen [-abdefgjkusv] [-o 选项] [-A 动作] [-G 全局模式] [-W 词语列表] [-F 函数] [-C 命令] [-X 过滤模式] [-P 前缀] [-S 后缀] [词语]  
依据选项显示可能的补全。
```

意图在能产生可能的补全的 `shell` 函数内部使用。
如果提供了可选的 `WORD` 参数, 则产生按照 `WORD`
进行的匹配。

退出状态:

除非使用了无效选项或者错误发生, 否则返回成功。

```
[root@Project-02-Task-01 ~]# compgen -a
```

```
cp  
egrep  
fgrep  
grep  
l.  
ll  
ls  
mv  
rm  
which  
xzegrep  
xzfgrep  
xzgrep  
zegrep  
zfgrep  
zgrep  
[root@Project-02-Task-01 ~]# compgen -b  
.  
:  
[  
alias  
bg  
bind  
break  
builtin  
caller  
cd  
command  
compgen  
complete  
compt  
continue  
declare  
dirs  
disown  
echo  
enable  
eval  
exec  
exit  
export  
false  
fc  
fg  
getopts  
hash  
help  
history  
jobs
```

`compgen -a`: 列出可用别名

`compgen -b`: 列出内置命令

`compgen -c`: 列出所有命令

```
[root@Project-02-Task-01 ~]# man
您需要什么手册页?
[root@Project-02-Task-01 ~]# man ls
[root@Project-02-Task-01 ~]#
[root@Project-02-Task-01 ~]# help
GNU bash, 版本 4.4.19(1)-release (x86_64-redhat-linux-gnu)
这些 shell 命令是内部定义的, 请输入 'help' 以获取一个列表。
输入 'help 名称' 以得到有关函数'名称'的更多信息。
使用 'info bash' 来获得关于 shell 的更多一般性信息。
使用 'man -k' 或 'info' 来获取不在列表中的命令的更多信息。
```

名称旁边的星号(*)表示该命令被禁用。

```
job_spec [&]
(( 表达式 ))
. 文件名 [参数]
:
[ 参数... ]
[[ 表达式 ]]
alias [-p] [名称 [=值] ... ]
bg [任务声明 ...]
bind [-lpsvsvx] [-m 键映射] [-f 文件名] [-q 名称] [-u 名称] [-r 键序列] [-x 键序列:shell-命令] [键序列:readli>
break [n]
builtin [shell 内建 [参数 ...]]
caller [表达式]
case 词 in [模式 [ | 模式]...] 命令 ;;)... esac
cd [-L|[-P [-e]] [-@]] [目录]
command [-pVv] 命令 [参数 ...]
compgen [-abcdfgjkusv] [-o 选项] [-A 动作] [-G 全局模式] [-W 词语列表] [-F 函数] [-C 命令] [-X 过滤模式] [->
complete [-abcdfgjkusv] [-pr] [-DE] [-o 选项] [-A 动作] [-G 全局模式] [-W 词语列表] [-F 函数] [-C 命令] [-X>
comptopt [-o|+o 选项] [-DE] [名称 ...]
continue [n]
coproc [名称] 命令 [重定向]
declare [-aAffgIlNrtux] [-p] [名称 [=值] ...]
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec ... | pid ...]
echo [-neE] [参数 ...]
enable [-a] [-dnps] [-f 文件名] [名称 ...]
eval [参数 ...]
exec [-cl] [-a 名称] [命令 [参数 ...]] [重定向 ...]
exit [n]
export [-fn] [名称 [=值] ...] 或 export -p
false
fc [-e 编辑器名] [-lnr] [起始] [终结] 或 fc -s [模式=替换串] [命令]
fg [任务声明]
for 名称 [in 词语 ... ]; do 命令; done
for (( 表达式1; 表达式2; 表达式3 )); do 命令; done
function 名称 { 命令 ; } 或 name () { 命令 ; }
getopts 选项字符串 名称 [参数]
hash [-lr] [-p 路径名] [-dt] [名称 ...]
help [-dms] [模式 ...]
```

```
[root@Project-02-Task-01 ~]# help cd
```

```
cd: cd [-L|[-P [-e]] [-@]] [目录]
    改变 shell 工作目录。
```

改变当前目录至 DIR 目录。默认的 DIR 目录是 shell 变量 HOME 的值。

变量 CDPATH 定义了含有 DIR 的目录的搜索路径, 其中不同的目录名称由冒号 (:)分隔。一个空的目录名称表示当前目录。如果要切换到 DIR 由斜杠 (/) 开头, 则 CDPATH

```
history [-c] [-d 偏移量] [n] 或 history -anrw [文件名] 或 history -ps 参数 [参数...]
if 命令; then 命令; [ elif 命令; then 命令; ]... [ else 命令; ] fi
jobs [-lnprs] [任务声明 ...] 或 jobs -x 命令 [参数]
kill [-s 信号声明 | -n 信号编号 | -信号声明] 进程号 | 任务声明 ... 或 kill -l [信号声明]
let 参数 [参数 ...]
local [option] 名称 [=值] ...
logout [n]
mcpfile [-d 分隔符] [-n 计数] [-O 起始序号] [-s 计数] [-t] [-u fd] [-C 回调] [-c 量子] [数组]
popd [-n] [+N | -N]
printf [-v var] 格式 [参数]
pushd [-n] [+N | -N | 目录]
pwd [-LP]
read [-ers] [-a 数组] [-d 分隔符] [-i 缓冲区文字] [-n 读取字符数] [-N 读取字符数] [-p 提示符] [-t 超时] [-u>
readarray [-n 计数] [-O 起始序号] [-s 计数] [-t] [-u fd] [-C 回调] [-c 量子] [数组]
readonly [-aAf] [名称 [=值] ...] 或 readonly -p
return [n]
select NAME [in 词语 ... ]; do 命令; done
set [--abefghkmnpstuvxBCHP] [-o 选项名] [--] [参数 ...]
shift [n]
shopt [-pqsu] [-o] [选项名 ...]
source 文件名 [参数]
suspend [-f]
test [表达式]
time [-p] 管道
times
trap [-lp] [[参数] 信号声明 ...]
true
type [-afptP] 名称 [名称 ...]
typeset [-aAffgIlNrtux] [-p] 名称 [=值] ...
ulimit [-SHabcdfiklmnpqrstuvxPT] [限制]
umask [-p] [-S] [模式]
unalias [-a] 名称 [名称 ...]
unset [-f] [-v] [-n] [名称 ...]
until 命令; do 命令; done
variables - 一些 shell 变量的名称和含义
wait [-n] [编号 ...]
while 命令; do 命令; done
{ 命令 ; }
```

7.其他操作命令

7.2 输出

- 使用echo可以输出一段文字，起到提示作用。
 - echo会将输入的字符串送往标准输出，可以输出shell变量的值。

命令详解：

【语法】

echo [选项] [参数]

【选项】

-n	不尾随换行符
-e	激活转义字符
-	金庸解释反斜杠的转义功能

【参数】

变量或字符	指定输出的内容
-------	---------

操作命令+配置文件+脚本程序+结束



7.其他操作命令

7.3 输出

- 使用export可以输出shell变量为环境变量，也可以管理环境变量。
 - export定义的环境变量仅对本次登录有效。

命令详解:

【语法】

export [选项] [参数]

【选项】

-f	代表[变量名称]中为函数名称
-n	删除指定的变量
-p	列出所有 shell 定义的环境变量

【参数】

变量	指定要输出或者删除的环境变量
----	----------------

操作命令+配置文件+脚本程序+结束




```
[root@Project-02-Task-01 ~]# export --help
export: export [-fn] [名称 [=值] ...] 或 export -p
为 shell 变量设定导出属性。
```

标记每个 **NAME** 名称为自动导出到后续命令执行的环境，如果提供了 **VALUE** 则导出前将 **VALUE** 作为赋值。

选项：

- f 指 shell 函数
- n 删除每个 **NAME** 名称的导出属性
- p 显示所有导出的变量和函数的列表

'--' 的参数禁用进一步的选项处理。

退出状态：

返回成功，除非使用了无效的选项或者 **NAME** 名称。

```
[root@Project-02-Task-01 ~]# export -p
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/0/bus"
declare -x HISTCONTROL="ignoredups"
declare -x HISTSIZE="1000"
declare -x HOME="/root"
declare -x HOSTNAME="Project-02-Task-01"
declare -x LANG="zh_CN.UTF-8"
declare -x LESSOPEN="||/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="root"
declare -x LS_COLORS="rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*tar=01;31:*tgz=01;31:*arc=01;31:*arj=01;31:*taz=01;31:*lha=01;31:*lzh=01;31:*lz4=01;31:*lzh=01;31:*lzma=01;31:*tlz=01;31:*txz=01;31:*tzo=01;31:*t7z=01;31:*zip=01;31:*z=01;31:*dz=01;31:*gz=01;31:*lrz=01;31:*lz=01;31:*lzo=01;31:*xz=01;31:*zst=01;31:*tst=01;31:*bz2=01;31:*bz=01;31:*tbz=01;31:*tbz2=01;31:*tz=01;31:*deb=01;31:*rpm=01;31:*jar=01;31:*war=01;31:*ear=01;31:*sar=01;31:*rar=01;31:*alz=01;31:*ace=01;31:*zoo=01;31:*cpio=01;31:*7z=01;31:*rz=01;31:*cab=01;31:*wim=01;31:*swm=01;31:*dwm=01;31:*esd=01;31:*jpg=01;35:*jpeg=01;35:*mjpg=01;35:*mjpeg=01;35:*gif=01;35:*bmp=01;35:*pbm=01;35:*pgm=01;35:*ppm=01;35:*tga=01;35:*xbm=01;35:*xpm=01;35:*tif=01;35:*tiff=01;35:*png=01;35:*svg=01;35:*svgz=01;35:*mng=01;35:*pcx=01;35:*mov=01;35:*mpg=01;35:*mpeg=01;35:*m2v=01;35:*mkv=01;35:*webm=01;35:*ogm=01;35:*mp4=01;35:*m4v=01;35:*mp4v=01;35:*vob=01;35:*qt=01;35:*nuv=01;35:*wmv=01;35:*asf=01;35:*rm=01;35:*rmvb=01;35:*flc=01;35:*avi=01;35:*fli=01;35:*flv=01;35:*gl=01;35:*dl=01;35:*xcf=01;35:*xwd=01;35:*yuv=01;35:*cgm=01;35:*emf=01;35:*ogv=01;35:*ogx=01;35:*aac=01;36:*au=01;36:*flac=01;36:*m4a=01;36:*mid=01;36:*midi=01;36:*mka=01;36:*mp3=01;36:*mpc=01;36:*ogg=01;36:*ra=01;36:*wav=01;36:*oga=01;36:*opus=01;36:*spx=01;36:*xspf=01;36:"
declare -x MAIL="/var/spool/mail/root"
declare -x OLDPWD="/usr/share/man"
declare -x PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin"
declare -x PWD="/root"
declare -x SELINUX_LEVEL_REQUESTED=""
declare -x SELINUX_ROLE_REQUESTED=""
declare -x SELINUX_USE_CURRENT_RANGE=""
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="10.10.0.1 64986 22"
declare -x SSH_CONNECTION="10.10.0.1 64986 10.10.2.103 22"
declare -x SSH_TTY="/dev/pts/0"
declare -x TERM="xterm"
declare -x USER="root"
declare -x XDG_RUNTIME_DIR="/run/user/0"
declare -x XDG_SESSION_ID="12"
[root@Project-02-Task-01 ~]#
```

7.其他操作命令

7.4 操作历史

- 使用history。
 - export定义的环境变量仅对本次登录有效。

命令详解:

【语法】

history [选项] [参数]

【选项】

-n	列出指定的 n 条命令
-c	清除所有的历史操作记录
-a	将新增的指令加入 histfiles 文件
-r	将 histfiles 文件内容输出到 shell 的历史中
-w	将当前历史命令缓冲区的命令写入到 histfiles 文件

【参数】

n	指定要输出的历史记录条数
---	--------------

操作命令+配置文件+脚本程序+结束



```
[root@Project-02-Task-01 ~]# history 20
```

```
666 clear
667 export --help
668 export -p
669 export -f ls
670 export -f min
671 export -f
672 export -f user
673 export -f HOSTNAME
674 export -f expr
675 export -f RANDOM
676 clear
677 history -n 30
678* histor
679 clear
680 history 20
681 clear
682 history 40
683 history --help
684 clear
685 history 20
```

```
[root@Project-02-Task-01 ~]# history --help
```

```
history: history [-c] [-d 偏移量] [n] 或 history -anrw [文件名] 或 history -ps 参数 [参数...]
    Display or manipulate the history list.
```

Display the history list with line numbers, prefixing each modified entry with a `*'. An argument of N lists only the last N entries.

Options:

- c clear the history list by deleting all of the entries
- d offset delete the history entry at position OFFSET.

- a append history lines from this session to the history file
- n read all history lines not already read from the history file and append them to the history list
- r read the history file and append the contents to the history list
- w write the current history to the history file

- p perform history expansion on each ARG and display the result without storing it in the history list
- s append the ARGs to the history list as a single entry

If FILENAME is given, it is used as the history file. Otherwise, if HISTFILE has a value, that is used, else ~/.bash_history.

If the HISTTIMEFORMAT variable is set and not null, its value is used as a format string for strftime(3) to print the time stamp associated with each displayed history entry. No time stamps are printed otherwise.

Exit Status:

Returns success unless an invalid option is given or an error occurs.

```
[root@Project-02-Task-01 ~]# █
```

```
[root@Project-02-Task-01 ~]# history
```

```
1 clear
2 ls
3 cd /
4 cd ~
5 mkdir demo1
6 mkdir demo2
7 touch file1
8 touch file2
9 ping www.baidu.com
10 mtr
11 history
12 top
13 vi file1
14 vi file2
15 ping www.baidu.com
16 cd /
17 ls
18 ls -l
19 cd ~
20 history
21 clear
22 history
[root@Project-02-Task-01 ~]#
```

哪些命令用的最多？频次如何？

```
[root@Project-02-Task-01 ~]# history
```

```
1 clear
2 ls
3 cd /
4 cd ~
5 mkdir demo1
6 mkdir demo2
7 touch file1
8 touch file2
9 ping www.baidu.com
10 mtr
11 history
12 top
13 vi file1
14 vi file2
15 ping www.baidu.com
16 cd /
17 ls
18 ls -l
19 cd ~
20 history
21 clear
22 history
```

```
[root@Project-02-Task-01 ~]# history | sed 's#^s+[0-9]\s\+#g' | awk -F ' ' '{print $1}' | sort | uniq -c | sort -k1,1nr -k2
```

```
4 cd
4 history
3 ls
2 clear
2 mkdir
2 ping
2 touch
2 vi
1 mtr
1 top
```

```
[root@Project-02-Task-01 ~]# history | awk -F ' ' '{print $2}' | sort | uniq -c | sort -k1,1nr -k2
```

```
5 history
4 cd
3 ls
2 clear
2 mkdir
2 ping
2 touch
2 vi
1 mtr
1 top
```

```
[root@Project-02-Task-01 ~]#
```

