

中国矿业大学计算机学院

2019 级本科生课程作业

课程名称	Linux 操作系统
作业次数	作业二
作业时间	2022 年 4 月 7 日
姓 名	胡钧耀
学 号	06192081
专 业	计算机科学与技术
任课教师	姜秀柱

目 录

一、 作业.....	1	1.4.3 df 命令.....	18
1.1 目录	1	1.4.4 du 命令.....	19
1.1.1 根目录	1	1.4.5 fdisk 命令	19
1.1.2 根目录的一级目录	1	1.5 文件系统.....	21
1.1.3 ls 命令	3	1.5.1 ext4 文件系统	21
1.1.4 mkdir 命令.....	5	1.5.2 mkfs 命令	21
1.1.5 cd 命令	5	1.5.3 fsck 命令	21
1.1.6 rmdir 命令.....	5	1.5.4 dumpe2fs 命令.....	22
1.1.7 cp 命令	6	1.6 挂载.....	23
1.1.8 mv 命令	6	1.6.1 定义与作用	23
1.1.9 rm 命令	7	1.6.2 mount 命令	23
1.1.10 pwd 命令.....	7	1.6.3 umount 命令	24
1.2 帮助	8	1.6.4 lsblk 命令	25
1.2.1 man 命令.....	8	1.6.5 findmnt 命令.....	25
1.2.2 --help 命令	9	1.7 用户、组与权限	26
1.2.3 help 命令.....	9	1.7.1 用户及其属性信息	26
1.2.4 info 命令.....	10	1.7.2 用户密码及其相关属性	26
1.2.5 whatis 命令	11	1.7.3 组及其属性信息	27
1.2.6 apropos 命令	12	1.7.4 组密码及其相关属性	27
1.3 文件	14	1.8 时间.....	28
1.3.1 文件类型	14	1.8.1 硬件时间和系统时间	28
1.3.2 文件属性	14	1.8.2 date 命令	28
1.3.3 chgrp 命令.....	15	1.8.3 tzselect 命令.....	28
1.3.4 chown 命令.....	15	1.8.4 hwclock 命令.....	29
1.3.5 chmod 命令.....	16	1.8.5 cal 命令.....	30
1.4 磁盘	17	二、 感悟	30
1.4.1 CHS、LBA 及其特点	17	参考文献.....	30
1.4.2 MBR、GPT 及其特点.....	17		

一、作业

1.1 目录

题目：给出 Ubuntu 安装后根目录下的一级目录的名称及存放的信息类别。常用目录命令有哪些（举例给出）？（10 分）

1.1.1 根目录

Linux 管理文件主要是通过目录管理，而不是 C 盘、D 盘等盘符进行管理，所以根目录十分重要，它是整个系统最重要的一个目录，一是因为所有的目录都是在根目录之中衍生创建出来的，每一个文件和目录都从这里开始，二是根目录也与系统的开机/还原/系统修复等动作有关系。只有 root 用户具有该目录下的写权限。

查看根目录下的一级目录的方法如下，代码如下所示，使用 `ls` 命令显示当前目录下的文件，`/` 表示根目录。

```
ls /
```

输入命令后，结果如下图所示。主要包括 `/bin`、`/boot`、`/dev`、`/etc`、`/home`、`/root`、`/run`、`/sbin`、`/tmp`、`/usr`、`/var` 等。

```
hujunyeo@HJYUbuntu:~$ ls /
bin    dev    lib    libx32  mnt    root   snap   sys    var
boot   etc    lib32  lost+found  opt    run    srv    tmp
cdrom  home   lib64  media    proc   sbin   swapfile  usr
```

图 1.1-1 查看根目录下的一级目录

1.1.2 根目录的一级目录

/bin 目录：即二进制（Binaries）的缩写。这里包含很重要的二进制可执行文件，存放着最经常使用的命令，例如 `cat`、`chmod`、`mkdir` 等。在 `/bin` 底下的指令可以被 root 与一般账号所使用。下图中的 `cat` 就是一个显示文件内容的命令，后面将详细介绍。

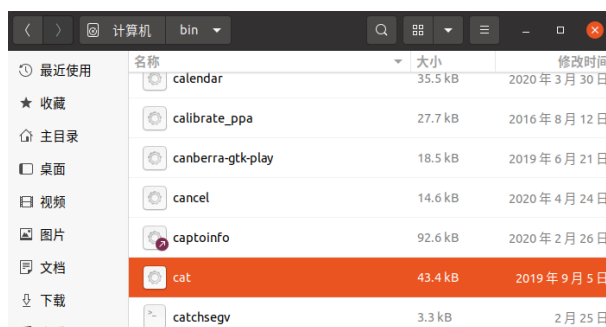


图 1.1-2 在 `/bin` 文件夹下存放的命令

/boot 目录：存放引导加载程序的相关文件，也就是开机会使用到的文件，包括 Linux 核心文件等等。

/dev 目录：即设备（Device）的缩写。Linux 系统中，任何装置与接口设备都是通过文件的形态存在于这个目录当中，这些包括终端设备、USB 或连接到系统的任何设备。

/etc 目录：包含了所有程序所需的配置文件，系统主要的配置文件几乎都放置在此。一般来说，这个目录下的各文件一般使用者具有阅读权限的，但是只有 root 有修改权限。

/home 目录：系统默认的用户家目录。所有用户用 home 目录来存储他们的个人档案。如当前情况下，只有 hujunyao 一个用户。

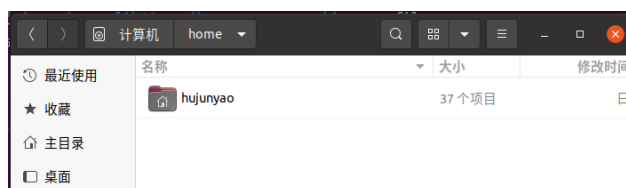


图 1.1-3 在/home 文件夹的不同用户

/lib 目录：包含支持位于/bin 和/sbin 下的二进制文件的库文件，库文件名为 ld*或 lib*.so.*。尤其重要的是/lib/modules/这个目录，该目录会放置核心相关的驱动程序。

/media 目录：即是媒体（Media）的英文，放置的就是可移除的装置，包括软盘、光盘、DVD 等等装置都暂时挂载于此，由于自己使用的虚拟机，虽然电脑是没有光盘驱动的，但是 VMware 是使用的 ISO 映像文件，所以可以在该目录下查看到当前 ubuntu 操作系统的.iso 文件。

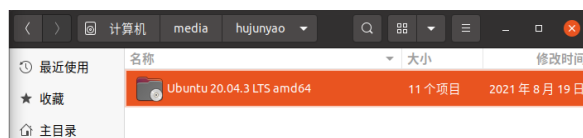


图 1.1-4 在/media 文件夹下的光盘映像文件

/mnt 目录：挂载（Mount）目录，是临时安装目录，系统管理员可以挂载文件系统。

/opt 目录：可选（Optional）目录，包含从个别厂商的附加应用程序。这些第三方的软件的应用程序应该安装在/opt 或者/opt 的子目录下。例如以前学习过 ROS，因此能在该文件夹下查看到 noetic 版本的 ROS 文件夹。

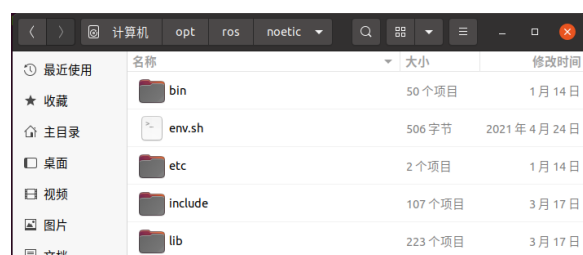


图 1.1-5 在/opt 文件夹下的 ROS 文件

/root 目录：系统管理员 root 的家目录。cd 是 Linux 内建的命令，无法直接使用 sudo cd 方法进入该文件夹，需要使用 sudo -i 从用户 hujunyao 变成 root 才可以进入。

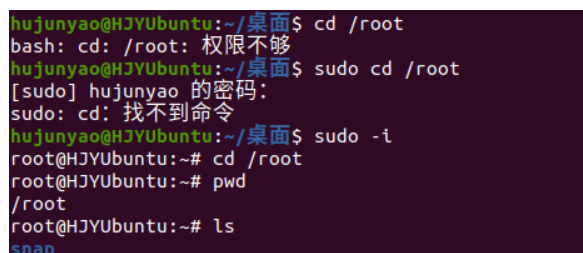


图 1.1-6 无法直接进入/root 文件夹

/run 目录: 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。

/sbin 目录: 即是超级用户的二进制文件 (Superuser Binaries) 的缩写, 这里存放的是系统管理员使用的系统管理程序。放在/sbin 下的为开机过程中所需要的一些指令, 里面包括了开机、修复、还原系统所需要的指令。常见指令包括 fdisk, ifconfig 等。

/tmp 目录: 即是临时 (Temporary) 的缩写, 用来存放一些临时文件。

/usr 目录: 即是共享资源 (Unix Shared Resources) 的缩写, 一定分清楚这个不是用户 User 的意思, 但是用户的很多应用程序和文件都放在这个目录下, 类似于 Windows 系统下的 Program Files 目录。

/var 目录: 即是变量 (Variable) 的缩写, 这个目录中存放着在不断扩充着的东西, 经常被修改的目录将被放在这个目录下, 这包括各种日志文件。

/lost+found 目录: 这个目录是使用标准的 ext2/ext3 文件系统格式才会产生的一个目录, 目的在于当文件系统发生错误时, 将一些遗失的片段放置到这个目录下。

/proc 目录: 进程 (Process) 目录, 是一个虚拟文件系统, 放置的数据都是在内存当中, 包含有关正在运行的进程的信息。因为这个目录下的数据都是在内存当中, 所以本身不占任何硬盘空间。

/sys 目录: 系统 (System) 目录, 也是一个虚拟的文件系统, 主要也是记录与核心相关的信息。包括目前已加载的核心模块与核心侦测到的硬件装置信息等等。这个目录同样不占硬盘容量。

/snap 目录: snap 是一种全新的软件包管理方式, 它类似一个容器拥有一个应用程序所有的文件和库, 各个应用程序之间完全独立, snap 软件包一般安装在/snap 目录下。

/srv 目录: 服务 (Service) 目录, 包含服务器特定服务相关的数据, 存放服务启动后需要提取的数据, 不用服务器就是空。

1.1.3 ls 命令

含义为 list, 作用是显示目录文件。

ls [-aAdhl1] 目录名称

选项与参数:

- a : 全部的文件, 连同隐藏档 (开头为 . 的文件) 一起列出来
- A : 全部的文件, 连同隐藏档, 但不包括 . 与 .. 这两个目录
- d : 仅列出目录本身, 而不是列出目录内的文件数据
- h : 将文件容量以人类较易读的方式 (例如 GB, KB 等等) 列出来
- l : 长数据串列出, 包含文件的属性与权限等等数据

下图实现了显示当前目录下非隐藏文件和目录。

```
hujunyao@HJYUbuntu:~$ ls
公共的      demo02
模板        Qt
视频        Qt6.2
图片        ROSProjectManager-0.5.0-Linux-x86_64
文档        ros_qtc_plugin-0.5.0
系统软件开发实践 ROS_TCP_Server
下载        ROS_TCP_Server.zip
音乐        server.py
桌面        snap
client.py   TCPdemo
demo01      VMwareTools-10.3.23-16594550
```

图 1.1-7

下图实现了显示当前目录隐藏文件在内的所有文件列表（-a）。

```
hujunyao@HJYUbuntu:~$ ls -a
.          .bashrc      Qt6.2
..         .cache       .ros
公共的     client.py    ROSProjectManager-0.5.0-Linux-x86_64
模板       .config     ros_qtc_plugin-0.5.0
视频       demo01      ROS_TCP_Server
图片       demo02     ROS_TCP_Server.zip
文档       .designer    server.py
系统软件开发实践 .gnupg      snap
下载       .local      .ssh
音乐       .mozilla    .sudo_as_admin_successful
桌面       .pki        TCPdemo
.bash_history .profile    VMwareTools-10.3.23-16594550
.bash_logout Qt          .vscode
```

图 1.1-8

下图实现了显示当前目录除了“.”和“..”的隐藏文件在内的所有文件列表（-A）。

```
hujunyao@HJYUbuntu:~$ ls -A
公共的     client.py    ROSProjectManager-0.5.0-Linux-x86_64
模板       .config     ros_qtc_plugin-0.5.0
视频       demo01      ROS_TCP_Server
图片       demo02     ROS_TCP_Server.zip
文档       .designer    server.py
系统软件开发实践 .gnupg      snap
下载       .local      .ssh
音乐       .mozilla    .sudo_as_admin_successful
桌面       .pki        TCPdemo
.bash_history .profile    VMwareTools-10.3.23-16594550
.bash_logout Qt          .vscode
.bashrc     Qt6.2
.cache     .ros
```

图 1.1-9

下图实现了显示当前目录本身的详细信息（-dl）。

```
hujunyao@HJYUbuntu:~$ ls -dl
drwxr-xr-x 31 hujunyao hujunyao 4096 3月 27 11:09 .
```

图 1.1-10

下图实现了列出人性化显示的文件和文件夹的详细信息（-lh）。

```
hujunyao@HJYUbuntu:~$ ls -lh
总用量 44M
drwxr-xr-x 2 hujunyao hujunyao 4.0K 1月 14 19:19 公共的
drwxr-xr-x 2 hujunyao hujunyao 4.0K 1月 14 19:19 模板
drwxr-xr-x 2 hujunyao hujunyao 4.0K 1月 14 19:19 视频
drwxr-xr-x 2 hujunyao hujunyao 4.0K 2月 22 18:41 图片
drwxr-xr-x 2 hujunyao hujunyao 4.0K 1月 15 14:17 文档
drwxr-xr-x 9 hujunyao hujunyao 4.0K 3月 15 15:30 系统软件开发实践
drwxr-xr-x 2 hujunyao hujunyao 4.0K 3月 18 10:07 下载
drwxr-xr-x 2 hujunyao hujunyao 4.0K 1月 14 19:19 音乐
drwxr-xr-x 3 hujunyao hujunyao 4.0K 1月 28 16:06 桌面
-rw-rw-r-- 1 hujunyao hujunyao 577 3月 25 13:10 client.py
drwxrwxr-x 7 hujunyao hujunyao 4.0K 3月 18 13:40 demo01
drwxrwxr-x 7 hujunyao hujunyao 4.0K 3月 25 11:09 demo02
drwxr-xr-x 6 hujunyao hujunyao 4.0K 1月 24 10:48 Qt
drwxrwxr-x 4 hujunyao hujunyao 4.0K 3月 18 12:26 Qt6.2
drwxrwxr-x 4 hujunyao hujunyao 4.0K 3月 17 19:45 ROSProjectManager-0.5.0-Linux-x86_64
drwxrwxr-x 7 hujunyao hujunyao 4.0K 3月 18 12:24 ros_qtc_plugin-0.5.0
drwxrwxr-x 6 hujunyao hujunyao 4.0K 3月 25 12:53 ROS_TCP_Server
-rw-rw-r-- 1 hujunyao hujunyao 44M 3月 27 11:09 ROS_TCP_Server.zip
-rw-rw-r-- 1 hujunyao hujunyao 2.3K 3月 25 18:31 server.py
drwx----- 4 hujunyao hujunyao 4.0K 1月 14 21:48 snap
drwxrwxrwx 4 hujunyao hujunyao 4.0K 3月 10 16:14 TCPdemo
drwxrwxr-x 3 hujunyao hujunyao 4.0K 1月 15 12:18 VMwareTools-10.3.23-16594550
```

图 1.1-11

1.1.4 mkdir 命令

含义为 make directories，作用是创建新目录。

mkdir [-p] 目录名称

选项与参数：

-p : 递归创建

下图实现了在当前文件夹创建新文件夹和递归创建的文件夹（-p）。

```
hujunyao@HJYUbuntu:~$ mkdir abc
hujunyao@HJYUbuntu:~$ mkdir abc/def
hujunyao@HJYUbuntu:~$ mkdir abc/def/efgh
mkdir: 无法创建目录 "abc/efgh": 没有那个文件或目录
hujunyao@HJYUbuntu:~$ mkdir -p abc/efgh
hujunyao@HJYUbuntu:~$ ls
公共的 桌面  ROSProjectManager-0.5.0-Linux-x86_64
模板   abc   ros_gtc_plugin-0.5.0
视频   abcd  ROS_TCP_Server
图片   client.py  ROS_TCP_Server.zip
文档   demo01  server.py
系统软件开发实践  demo02  Snap
下载   Qt      VMwareTools-10.3.23-16594550
音乐   Qt5.2
```

图 1.1-12

1.1.5 cd 命令

含义为 change directory，作用是切换目录。

cd 目录名称

选项与参数：

~ : home 目录

.. : 目前目录位置的上一层目录

下图实现了进入 home 目录和某一文件夹。

```
hujunyao@HJYUbuntu:~$ cd demo01
hujunyao@HJYUbuntu:~/demo01$ cd ~
hujunyao@HJYUbuntu:~$ pwd
/home/hujunyao
hujunyao@HJYUbuntu:~$ cd abc
hujunyao@HJYUbuntu:~/abc$ pwd
/home/hujunyao/abc
```

图 1.1-13

1.1.6 rmdir 命令

含义为 remove directories，作用是删除空目录（如果目录下存在文件则不能删除）。

rmdir [-p] 目录名称

选项与参数：

-p : 是当子目录被删除后使它也成为空目录的话，则顺便一并删除。

下图实现了删除 100（含有 101 和 102 两个文件夹）和 200（含有 201 一个文件夹）两个文件夹，100 文件夹不能使用递归删除 101，提示 100 非空（还有 102 文件夹）。

```
hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir -p 100/101
hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir -p 100/102
hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir -p 200/201
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 100
101 102
hujunyao@HJYUbuntu:~/LinuxCourse$ rmdir -p 100/101
rmdir: 删除目录 '100' 失败: 目录非空
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 100
102
hujunyao@HJYUbuntu:~/LinuxCourse$ rmdir -p 100/102
hujunyao@HJYUbuntu:~/LinuxCourse$ ls
200
hujunyao@HJYUbuntu:~/LinuxCourse$ rmdir -p 200/201
hujunyao@HJYUbuntu:~/LinuxCourse$ ls
```

图 1.1-14

1.1.7 cp 命令

含义为 copy，作用是复制文件或目录。

cp [-adfIprl] 源文件... 目标目录

选项与参数：

- a : 此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于 dpr 参数组合。
- d : 复制时保留链接。这里所说的链接相当于 Windows 系统中的快捷方式。
- f : 覆盖已经存在的目标文件而不给出提示。
- i : 与 -f 选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖，回答 y 时目标文件将被覆盖。
- p : 除复制文件的内容外，还把修改时间和访问权限也复制到新文件中。
- r : 若给出的源文件是一个目录文件，此时将复制该目录下所有的子目录和文件。
- l : 不复制文件，只是生成链接文件。

下图实现了将 100 文件夹中的 test.txt 复制到 200 文件夹中。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ mkdir 100
hujunyao@H3YUubuntu:~/LinuxCourse$ mkdir 200
hujunyao@H3YUubuntu:~/LinuxCourse$ gedit 100/test.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ ls 100
test.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ cp 100/test.txt 200
hujunyao@H3YUubuntu:~/LinuxCourse$ ls 100
test.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ ls 200
test.txt
```

图 1.1-15

下图实现了将 101 文件夹复制到 201 文件夹中，并且把 200/201/demo1.txt 覆盖为 100/101/demo1.txt。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ mkdir -p 100/101
hujunyao@H3YUubuntu:~/LinuxCourse$ mkdir -p 200/201
hujunyao@H3YUubuntu:~/LinuxCourse$ gedit 100/101/demo1.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ gedit 100/101/demo2.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ gedit 200/201/demo1.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ cp -a 100/101 200/201
hujunyao@H3YUubuntu:~/LinuxCourse$ ls 200/201
101 demo1.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ cp -i 100/101/demo1.txt 200/201
cp: 是否覆盖 '200/201/demo1.txt'? y
```

图 1.1-16

1.1.8 mv 命令

含义为 move，作用是剪切文件、改名。

mv [options] source dest

mv [options] source... directory

选项与参数：

- b : 当目标文件或目录存在时，在执行覆盖前，会为其创建一个备份。
- i : 如果指定移动的源目录或文件与目标的目录或文件同名，则会先询问是否覆盖旧文件，输入 y 表示直接覆盖，输入 n 表示取消该操作。
- f : 如果指定移动的源目录或文件与目标的目录或文件同名，不会询问，直接覆盖旧文件。
- n : 不要覆盖任何已存在的文件或目录。
- u : 当源文件比目标文件新或者目标文件不存在时，才执行移动操作。

下图实现了创建 100/demo.txt 和 200/demo1.txt,把 100 文件夹移动到 200 文件夹内,把 200 文件夹重命名为 300,再把 300/100/demo.txt 重命名为 300/100/demo2.txt,然后把 300/100/demo2.txt 移动到 300 文件夹内。

```

hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir 100
hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir 200
hujunyao@HJYUbuntu:~/LinuxCourse$ gedit 100/demo.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ gedit 200/demo1.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 200
demo1.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ mv 100 200
hujunyao@HJYUbuntu:~/LinuxCourse$ ls
200
hujunyao@HJYUbuntu:~/LinuxCourse$ mv 200 300
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 300/100
demo.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ mv 300/100/demo.txt 300/100/demo2.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 300/100
demo2.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ mv 300/100/demo2.txt 300
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 300
100 demo1.txt demo2.txt

```

图 1.1-17

1.1.9 rm 命令

含义为 remove,作用是删除文件或目录。

rm [options] name...

选项与参数:

- i : 删除前逐一询问确认。
- f : 即使原档案属性设为唯读,亦直接删除,无需逐一确认。
- r : 将目录及以下之档案亦逐一删除。

下图实现了在 100 文件夹中创建 3 个 txt 文件,然后删除其中一个,最后删除该 100 文件夹。

```

hujunyao@HJYUbuntu:~/LinuxCourse$ mkdir 100
hujunyao@HJYUbuntu:~/LinuxCourse$ gedit 1.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ gedit 2.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ gedit 3.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ mv 1.txt 2.txt 3.txt 100/
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 100
1.txt 2.txt 3.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ rm 100/1.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls 100
2.txt 3.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ rm 100
rm: 无法删除 '100': 是一个目录
hujunyao@HJYUbuntu:~/LinuxCourse$ rm -rf 100
hujunyao@HJYUbuntu:~/LinuxCourse$ ls

```

图 1.1-18

1.1.10 pwd 命令

pwd 命令: 含义为 print working directory,作用是显示当前目录。如下图所示。

pwd

选项与参数:

- L, --logical : 目录连接链接时,输出连接路径,不考虑软、硬连接路径
- P, --physical : 查看软、硬链接最终指向文件的全部路径

```

hujunyao@HJYUbuntu:~/abc$ pwd
/home/hujunyao/abc

```

图 1.1-19

1.2 帮助

题目：Linux 的命令帮助方式有哪些？各有什么特点？常用帮助命令有哪些（举例给出）？（15 分）

主要有 `man` 命令、`--help` 命令、`help` 命令、`info` 命令、`whatis` 命令和 `apropos` 命令，下面分开论述这几个命令的特点和使用。

1.2.1 man 命令

`man` 特点是适用于一般命令，非内置命令。它是包含了 Linux 中全部命令的手册（Manual），手册非常详细，这个手册分为八大章节，使用 `man` 时也可以指定不同的章节来阅读。八个章节是：标准命令（Standard commands）、系统调用（System call）、库函数（Library function）、设备文件（Special devices）、文件格式（File formats）、游戏和娱乐（Games and toys）、杂项、惯例与协定（Miscellaneous）、管理员命令（Administrative commands）。

```
man [man options] [[section] page ...] ...
```

```
man -k [apropos options] regexp ...
```

```
man -f [whatis 选项] 页 ...
```

选项与参数：

`-a, --all`：

默认情况下，`man` 在显示它找到的最符合的手册页后会退出。此选项强制 `man` 显示名字匹配搜索条件的所有的手册页。

`-f, --whatis`：

等同于 `whatis`。显示来自手册页的简短说明，每个手册页中都有一个概述。`whatis` 搜索手册页名称并显示任何与名称匹配的手册页描述。

`-k, --apropos`：

等同于 `apropos`。搜索关键词对应的手册概述并显示所有匹配结果。详见 `apropos`。

下图实现了查询 `man` 命令的所有的手册页，查找到在第一章节和第七章节都有说明，但是该命令最常用的是第一章节，所以不加 `-a` 参数时返回的结果等同于 `man.7`。

```
hejunyao@13YUbuntu:~/桌面$ man -a man
--Man-- 下一页: man(1) [ 查看 (return) | 跳过 (ctrl-D) | 退出 (ctrl-C) ]
return
--Man-- 下一页: man(7) [ 查看 (return) | 跳过 (ctrl-D) | 退出 (ctrl-C) ]
return

MAN(1)                                手册分页显示工具                                MAN(1)
名称
  man - an interface to the system reference manuals

概述
  man [man options] [[section] page ...] ...
  man -k [apropos options] regexp ...
  man -K [man options] [section] term ...
  man -f [whatis 选项] 页 ...
  man -l [man options] file ...
  man -w|-W [man options] page ...

MAN(7)                                Linux Programmer's Manual                                MAN(7)
NAME
  man - macros to format man pages

SYNOPSIS
  groff -Tascii -man file ...

  groff -Tps -man file ...

  man [section] title
```

图 1.2-1

下图实现了查询 `adjtime` 命令来自手册页的简短说明，可知有两个手册页被查询出来。每个手册页中都有一个概述。

```
hu junyao@HJYUbuntu:~/桌面$ man -f adjtime
adjtime_config (5) - information about hardware clock setting and drift factor
adjtime (3) - correct the time to synchronize the system clock
```

图 1.2-2

下图实现了查询和关键词 `time` 有关的手册概述并显示所有匹配结果，这些都是和一些和时间配置相关的命令。

```
hu junyao@HJYUbuntu:~/桌面$ man -k time
__ppc_get_timebase (3) - get the current value of the Time Base Register on P...
__ppc_get_timebase_freq (3) - get the current value of the Time Base Register...
adjtime (3) - correct the time to synchronize the system clock
adjtime_config (5) - information about hardware clock setting and drift factor
adjtimex (2) - tune kernel clock
alo_suspend (3) - wait for asynchronous I/O operation or timeout
asctime (3) - transform date and time to broken-down time or ASCII
asctime_r (3) - transform date and time to broken-down time or ASCII
bootparam (7) - introduction to boot time parameters of the Linux kernel
chrt (1) - manipulate the real-time attributes of a process
clock (3) - determine processor time
clock_getcpuclockid (3) - obtain ID of a process CPU-time clock
clock_getres (2) - clock and time functions
clock_getres (3) - clock and time functions
```

图 1.2-3

1.2.2 --help 命令

获取一些最基础的帮助。在通常输入错误命令的时候就会提示【请尝试执行“`xxx --help`”】来获取更多信息。适用于一般命令，非内置命令。它自己本身就是个参数。

命令名 `--help`

下图实现了查询 `ls` 命令的基本的使用方法和参数的介绍。

```
hu junyao@HJYUbuntu:~/桌面$ ls --help
用法: ls [选项]... [文件]...
列出给定文件（默认为当前目录）的信息。
如果不指定 -cftuvSUX 中任意一个或--sort 选项，则根据字母大小排序。

必选参数对长短选项同时适用。
-a, --all          不隐藏任何以 . 开始的项目
-A, --almost-all  列出除 . 及 .. 以外的任何项目
--author          与 -l 同时使用时，列出每个文件的作者
-b, --escape       以 C 风格的转义序列表示不可打印的字符
--block-size=大小  与 -l 同时使用时，将文件大小以此处给定的大小为
                  单位进行缩放；例如：“--block-size=M”；
                  请参考下文的大小格式说明
-B, --ignore-backups 不列出任何以 ~ 字符结束的项目
```

图 1.2-4

1.2.3 help 命令

该命令用于获取 `bash` 内置命令（如 `cd`、`echo`、`help`）帮助。

`help [-dms] [模式 ...]`

选项与参数：

- `-d` : 输出每个主题的简短描述
- `-m` : 以伪 `man` 手册的格式显示使用方法
- `-s` : 为每一个匹配 `PATTERN` 模式的主题仅显示一个用法简介
- `PATTERN` : 指定帮助主题的模式

对于某一些内部命令，使用该方法可以查询帮助，但是 `--help` 命令不可以，如 `echo` 命令，但是不是内建函数的肯定是不能查询到的，例如 `ls`，如下图所示。

```

hujunyao@H3YUbuntu:~/桌面$ help ls
bash: help: 没有与“ls”匹配的帮助主题。尝试使用“help help”、“man -k ls”或“info ls”
hujunyao@H3YUbuntu:~/桌面$ echo --help
--help
hujunyao@H3YUbuntu:~/桌面$ help echo
echo: echo [-neE] [参数 ...]
      将参数写到标准输出。
      在标准输出上，显示用空格分割的 ARG 参数后跟一个换行。
选项:
  -n      不要追加换行
  -e      启用下列反斜杠转义的解释
  -E      显式地抑制对于反斜杠转义的解释

```

图 1.2-5

下图实现了三种选项输出结果的比较。

```

hujunyao@H3YUbuntu:~/桌面$ help -s echo
echo: echo [-neE] [参数 ...]
hujunyao@H3YUbuntu:~/桌面$ help -d echo
echo - Write arguments to the standard output.
hujunyao@H3YUbuntu:~/桌面$ help -m echo
NAME
echo - 将参数写到标准输出。
SYNOPSIS
echo [-neE] [参数 ...]
DESCRIPTION
将参数写到标准输出。
      在标准输出上，显示用空格分割的 ARG 参数后跟一个换行。

```

图 1.2-6

1.2.4 info 命令

适用于一般命令，比 man 命令稍微友好一点。它不是像 info page 一口气输出一堆信息，info page 则是将文件数据拆成一个一个的段落，每个段落用自己的页面来撰写，是分页显示，用百分比表示。

info [OPTION]... [MENU-ITEM...]

选项与参数：

-k, --apropos=STRING

在所有手册的所有索引中查找 **STRING**

--index-search=STRING

转到由索引项 **STRING** 指向的节点

-o, --output=FILE

输出被选择的节点内容到指定的文件

-w, --where, --location

查看命令的 **info** 格式的帮助用户文档地址。

下图实现了输入 info 进入顶级目录。

```

file: dir,      Node: Top,      This is the top of the INFO tree.

This is the Info main menu (aka directory node).
A few useful Info commands:

'q' quits;
'H' lists all Info commands;
'h' starts the Info tutorial;
'nTextinfo RET' visits the Texinfo manual, etc.

* Menu:

```

图 1.2-7

下图实现了在手册的所有索引中查找与 time 有关的内容。

```

hujunyao@H3YUbuntu:~/桌面$ info -k time
(coreutils)What information is listed -- --full-time
(coreutils)Options for date -- --iso-8601[=TIMESPEC]
(coreutils)install invocation -- --preserve-timestamps
(coreutils)Options for date -- --rfc-3339=TIMESPEC
(coreutils)Sorting the output -- --time
(coreutils)Sorting the output -- --time <1>
(coreutils)touch invocation -- --time <2>
(coreutils)touch invocation -- --time <3>
(coreutils)du invocation -- --time <4>
(coreutils)du invocation -- --time <5>
(coreutils)du invocation -- --time <6>

```

图 1.2-8

下图实现了将 `echo` 命令的信息输出到 `out.txt` 文件中。

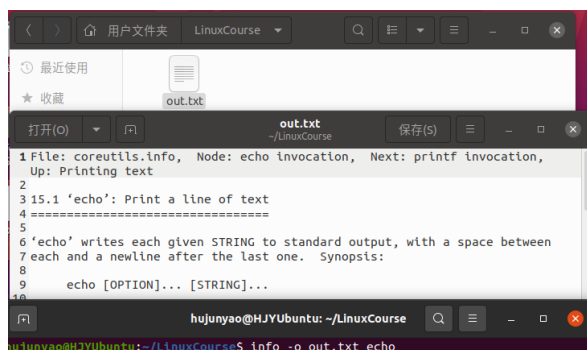


图 1.2-9

下图实现了在 `info` 命令的文档中查找和 `index` 有关的节点，输入 “,” 可以尝试寻找下一个序列节点是 `index` 的节点。

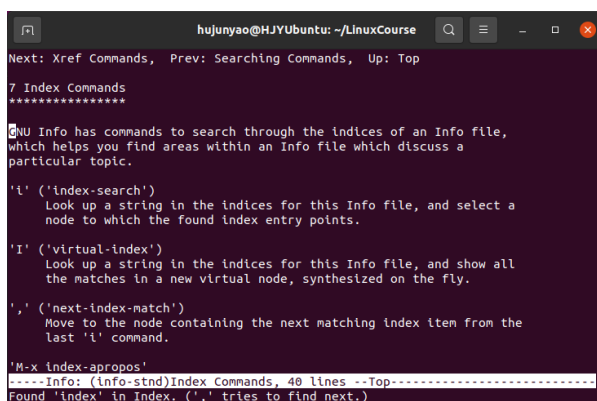


图 1.2-10

1.2.5 whatis 命令

相当于 `man -f` 命令。将会查找一个存储所有命令相关信息的数据库，根据命令名返回相关结果。

whatis [-dlv?V] [-r|-w] [-s 列表] 名称 ...

选项与参数：

-r, --regex

将每个名称解释为正则表达式。如果名称匹配页面名称的任何部分，将视为一个匹配。该选项会使 `whatis` 略微变慢，这是由数据库搜索的原理导致的。

-w, --wildcard

将每个名称解释为包含 `shell` 风格通配符的模式。要匹配，展开的名称要与整个页面名称匹配。该选项会使 `whatis` 略微变慢，这是由数据库搜索的原理导致的。

-l, --long

不按终端宽度截断输出。

--usage

打印简短的帮助消息并退出。

下图实现了使用正则表达式查找以 **reset** 结尾的命令。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ whatis -r 'reset$'
avc_reset (3)      - userspace SELinux AVC setup and teardown
glt-reset (1)      - Reset current HEAD to the specified state
reset (1)          - terminal initialization
systemd.preset (5) - Service enablement presets
tracker-reset (1)  - Reset the index and configuration
```

图 1.2-11

下图实现了使用通配符查找含有 **time** 的命令。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ whatis -w *time*
__ppc_get_timebase (3) - get the current value of the Time Base Register on P...
__ppc_get_timebase_freq (3) - get the current value of the Time Base Register...
adjtime (3)           - correct the time to synchronize the system clock
adjtime_config (5)    - information about hardware clock setting and drift factor
adjtimex (2)          - tune kernel clock
asctime (3)           - transform date and time to broken-down time or ASCII
asctime_r (3)         - transform date and time to broken-down time or ASCII
clock_gettime (2)     - clock and time functions
clock_gettime (3)     - clock and time functions
```

图 1.2-12

下图实现了检索的 **vcstime** 命令的概述完整输出。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ whatis vcstime
vcstime (8)          - Show time in upper right hand corner of the conso...
hujunyao@H3YUubuntu:~/LinuxCourse$ whatis -l vcstime
vcstime (8)          - Show time in upper right hand corner of the console
screen
```

图 1.2-13

下图实现了显示有关 **whatis** 命令的简短信息。它显示简短说明并退出。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ whatis --usage cd
用法: whatis [-dvrwl?V] [-C 文件] [-L 区域] [-m 系统] [-M 路径]
      [-s 列表] [--debug] [--verbose] [--regex] [--wildcard] [--long]
      [--config-file=文件] [--locale=区域] [--systems=系统]
      [--manpath=路径] [--sections=列表] [--section=列表] [--help]
      [-usage] [--version] 关键词...
hujunyao@H3YUubuntu:~/LinuxCourse$
```

图 1.2-14

1.2.6 apropos 命令

用来通过关键字查找定位手册页的名字和描述。它相当于使用带有 **-k** 选项的 **man** 命令。每个手册页里都有一个简短的描述，本命令会搜索包括关键词的描述。

apropos [-e|-w|-r] [-s 列表] 关键词 ...

选项与参数:

-r, --regex

将每个关键词解释为正则表达式。这是默认行为。每个关键词将与手册页名称和描述分别匹配。匹配不限于单词边界。

-w, --wildcard

将每个关键词解释为 **shell** 风格的通配符。关键词将与手册页名称和描述分别匹配。

-l, --long

不按终端宽度截断输出。

-s list, --sections=list, --section=list

只搜索指定的手册章节。列表是由冒号或逗号分隔的章节列表。如果列表中的指定的是简单的章节号，如“**3**”，则显示的描述列表将只包含“**3**”、“**3perl**”、“**3x**”等章节中的页面；如果列表中的章节号带扩展，如“**3perl**”，则显示的描述列表将只包含手册章节中对应部分的页面。

下图实现了使用正则表达式查找以 **reset** 结尾的命令。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ apropos -r 'reset$'
avc_reset (3) - userspace SELinux AVC setup and teardown
git-reset (1) - Reset current HEAD to the specified state
reset (1) - terminal initialization
systemd.preset (5) - Service enablement presets
tracker-reset (1) - Reset the index and configuration
```

图 1.2-15

下图实现了使用通配符查找含有 **time** 的命令。注意到这里 **bootparam** 命令不含 **time** 但是其简介中有 **time** 出现。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ apropos -w *time*
__ppc_get_timebase (3) - get the current value of the Time Base Register...
__ppc_get_timebase_freq (3) - get the current value of the Time Base Reg...
adjtime (3) - correct the time to synchronize the system clock
adjtime config (5) - information about hardware clock setting and drif...
adjtimex (2) - tune kernel clock
asctime (3) - transform date and time to broken-down time or ASCII
asctime_r (3) - transform date and time to broken-down time or ASCII
bootparam (7) - introduction to boot time parameters of the Linux...
chrt (1) - manipulate the real-time attributes of a process
clock_getcpuclockid (3) - obtain ID of a process CPU-time clock
clock_getres (2) - clock and time functions
clock_gettime (3) - clock and time functions
clock_gettime (2) - clock and time functions
clock_gettime (3) - clock and time functions
clock_settime (2) - clock and time functions
clock_settime (3) - clock and time functions
```

图 1.2-16

下图实现了检索的 **vcstime** 命令的概述完整输出。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ apropos vcstime
vcstime (8) - Show time in upper right hand corner of the conso...
hujunyao@H3YUubuntu:~/LinuxCourse$ apropos --long vcstime
vcstime (8) - Show time in upper right hand corner of the console
screen
```

图 1.2-17

下图实现了检索第二章和第五章中以 **time** 结尾的命令，并将概述完整输出。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ apropos -r 'time$' --sections=2,5 -l
adjtime_config (5) - information about hardware clock setting and drift f
actor
clock_gettime (2) - clock and time functions
clock_settime (2) - clock and time functions
security (2) - unimplemented system calls
gettimeofday (2) - get / set time
localtime (5) - Local timezone configuration file
modules (5) - kernel modules to load at boot time
settimeofday (2) - get / set time
stime (2) - set time
time (2) - get time in seconds
timer_gettime (2) - arm/disarm and fetch state of POSIX per-process time
r
timer_settime (2) - arm/disarm and fetch state of POSIX per-process time
r
timerfd_gettime (2) - timers that notify via file descriptors
timerfd_settime (2) - timers that notify via file descriptors
utime (2) - change file last access and modification times
```

图 1.2-18

1.3 文件

题目：Linux 的文件类型有哪些？一个文件的属性有哪些？修改文件的属性的命令有哪些（举例给出）？（10 分）

1.3.1 文件类型

Linux 的文件类型主要有：普通文件、目录、特殊文件（符号链接文件、包括字符设备文件、套接口文件、命名管道文件、块设备文件）。

表 1.3-1 Linux 文件类型及其符号

符号	含义
-	普通文件
d	目录文件（Directory）
l	链接文件（Link）
c	字符设备文件（Character）
s	套接字文件（Socket）
p	命名管道文件（Pipeline）
b	块文件（Block）

使用前面提到的 `ls -li` 命令可以具体查看某一文件夹内文件的类型和 inode 数据，文件大小同时人性化显示，如下图所示，在根目录下，查看最前面的十位字符串的第一位，可以了解到有普通文件（swapfile）、目录文件（boot 等）和链接文件（bin 等）。

```

total 924M
13 lrwxrwxrwx 1 root root 7 1月 14 16:28 bin -> usr/bin
786433 drwxr-xr-x 4 root root 4.0K 4月 1 08:28 boot
538301 drwxrwxr-x 2 root root 4.0K 1月 14 16:30 cdrom
1 drwxr-xr-x 20 root root 4.3K 3月 30 09:26 dev
1048577 drwxr-xr-x 142 root root 12K 3月 31 08:09 etc
393217 drwxr-xr-x 3 root root 4.0K 1月 14 16:35 home
14 lrwxrwxrwx 1 root root 7 1月 14 16:28 lib -> usr/lib
15 lrwxrwxrwx 1 root root 9 1月 14 16:28 lib32 -> usr/lib32
16 lrwxrwxrwx 1 root root 9 1月 14 16:28 lib64 -> usr/lib64
17 lrwxrwxrwx 1 root root 10 1月 14 16:28 libx32 -> usr/libx32
11 drwx----- 2 root root 16K 1月 14 16:25 lost+found
917505 drwxr-xr-x 3 root root 4.0K 1月 15 10:05 media
524289 drwxr-xr-x 3 root root 4.0K 1月 15 10:12 mnt
655361 drwxr-xr-x 3 root root 4.0K 1月 14 21:13 opt
1 dr-xr-xr-x 376 root root 0 3月 30 09:26 proc
524290 drwx----- 7 root root 4.0K 1月 15 12:13 root
1 drwxr-xr-x 35 root root 980 4月 1 15:09 run
18 lrwxrwxrwx 1 root root 0 1月 14 16:28/sbin -> usr/sbin
393218 drwxr-xr-x 13 root root 4.0K 1月 15 10:14 snap
655362 drwxr-xr-x 2 root root 4.0K 8月 19 2021 srv
12 -rw----- 1 root root 924M 1月 14 16:25 swapfile
1 dr-xr-xr-x 13 root root 0 3月 30 09:26 sys
524291 drwxrwxrwt 21 root root 20K 4月 1 17:37 tmp
131075 drwxr-xr-x 14 root root 4.0K 8月 19 2021 usr
262145 drwxr-xr-x 14 root root 4.0K 8月 19 2021 var
  
```

图 1.3-1 根目录下文件的不同类型

1.3.2 文件属性

观察上图可以知道文件的基本属性有八个，从左往右依次有：inode 索引节点、文件类型及权限、硬链接个数、文件所属用户、文件所属用户组、文件大小（目录通常是 4KB 大小）、文件修改时间、文件名。

inode 索引节点：Linux 系统的文件数据存储在块中，还需要一个地方存储文件的元信息（文件的创建者、创建日期等等），这种存储文件元信息的区域就叫做 inode 索引节点。因此，一个文件必须占用一个 inode，但至少占用一个 block。使用 inode 节点来产生的新的文件名，而不是产生新的文件，称为硬链接。再创建一个独立的文件，而这个文件会让数据的读取指向它连接的那个文件的文件名，inode 不一样，但文件的内容是另一文件的路径（有些类似于重定向，或者说是快捷方式），称为软链接。

文件类型及权限：文件类型已经在上述小节介绍，文件的权限主要是体现在十位标识的后面九位，三个为一组，都是 **rwX** 三个参数进行组合，**r** 表示可读（readable），**w** 表示可写（writable），**x** 表示可执行（executable），如果没有这一项权限，以 **-** 号代替该位，三组参数从左到右是属主权限、属组权限、其他用户权限，所以文件类型及权限一定是十位标识。

1.3.3 chgrp 命令

即改变群组（Change group），用于变更文件或目录的所属群组，它允许普通用户改变文件所属的组，只要该用户是该组的一员。

```
chgrp [-cfhRv][--help][--version][所属群组][文件或目录...]
```

选项与参数：

-c --changes

效果类似 **-v** 参数，但仅显示更改的部分。

-h 或 --no-dereference

只对符号连接的文件作修改，而不改动其他任何相关文件

-R --recursive

递归处理，将指定目录下的所有文件及子目录一并处理

-v --verbose

显示指令执行过程

下图实现了查看 LinuxCourse 文件夹下的 *out.txt* 文件的权限，并查看了可以修改为哪些属组（如 *adm*、*dip* 等），并将属组修改为 *cdrom*，可以查看到属组成功修改。

```
hujunyao@H3YUubuntu:~/LinuxCourse$ ls -l
总用量 4
-rw-rw-r-- 1 hujunyao hujunyao 2722 4月  1 17:08 out.txt
hujunyao@H3YUubuntu:~/LinuxCourse$ chgrp -v out.txt
adm      dip      lpadmin  plugdev  sudo
cdrom    hujunyao lxd      sambashare
hujunyao@H3YUubuntu:~/LinuxCourse$ chgrp -v cdrom out.txt
'out.txt' 的所属组已从 hujunyao 更改为 cdrom
hujunyao@H3YUubuntu:~/LinuxCourse$ ls -l
总用量 4
-rw-rw-r-- 1 hujunyao cdrom 2722 4月  1 17:08 out.txt
```

图 1.3-2

1.3.4 chown 命令

即改变所属用户和组（Change owner），用于设置文件所有者和文件关联组的命令。该命令将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户 ID，组可以是组名或者组 ID，文件是以空格分开的要改变权限的文件列表，支持通配符。该命令需要超级用户 *root* 的权限。

```
chown [-cfhvR] user[:group] file...
```

选项与参数：

user ：新的文件拥有者的使用者

group ：新的文件拥有者的使用者组

-c ：显示更改的部分的信息

-f ：忽略错误信息

-h ：修复符号链接

-v ：显示详细的处理信息

-R ：处理指定目录以及其子目录下的所有文件

下图实现了 `/opt/ros` 文件夹的权限从 `root` 到 `hujunyao` 进行互换，可以发现如果以 `hujunyao` 的权限尝试运行 `chown` 命令会提示不允许操作，必须切换到 `root` 权限才可以进行修改。

```
hujunyao@HJYUbuntu: /opt$ chown -v hujunyao ros
chown: 正在更改 'ros' 的所有者: 不允许的操作
无法将 'ros' 的所有者从 root 更改为 hujunyao
hujunyao@HJYUbuntu: /opt$ sudo -i
[sudo] hujunyao 的密码:
root@HJYUbuntu:~# cd /opt
root@HJYUbuntu: /opt# chown -v hujunyao ros
'ros' 的所有者已从 root 更改为 hujunyao
root@HJYUbuntu: /opt# ls -l
总用量 4
drwxr-xr-x 3 hujunyao root 4096 1月 14 21:13 ros
root@HJYUbuntu: /opt# chown -v root ros
'ros' 的所有者已从 hujunyao 更改为 root
root@HJYUbuntu: /opt# chown -R root ros
root@HJYUbuntu: /opt# ls -l
总用量 4
drwxr-xr-x 3 root root 4096 1月 14 21:13 ros
```

图 1.3-3

1.3.5 chmod 命令

Linux 文件属性有两种设置方法，一种是数字，一种是符号。Linux 文件的基本权限就有九个，分别是属主/属组/其他（owner/group/others）三种身份各有自己的可读、可写、可执行权限。r 对应 4，w 对应 2，x 对应 1。综上，`-rwxrwx---`就等价于 `770`。还有一个改变权限的方法是直接用文字 `rwX` 进行描述等于、新增、减少权限。综上，`-rwxrwx---`就等价于 `chmod u=rwx,g=rwx,o=-` 文件名

`chmod [-R] xyz 文件或目录`

`chmod u/g/o/a +/-/= r/w/x 文件或目录`

选项与参数：

xyz : 数字类型的权限属性，为 `rwX` 属性数值

-R : 进行递归(recursive)的持续变更，以及连同次目录下的所有文件都会变更

下图实现了对 `out.txt` 文件权限进行修改，首先设置为 `777`（属主、属组、其他用户均具有读写执行权限），然后清空其他用户的所有权限，然后减少属组的执行权限，增加其他用户的读写权限。

```
hujunyao@HJYUbuntu:~/LinuxCourse$ ls -l
总用量 4
-rw-rw-r-- 1 hujunyao hujunyao 2722 4月 1 17:08 out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ chmod 777 out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls -l
总用量 4
-rwxrwxrwx 1 hujunyao hujunyao 2722 4月 1 17:08 out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ chmod o= out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls -l
总用量 4
-rwxrwx--- 1 hujunyao hujunyao 2722 4月 1 17:08 out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ chmod g-x,o+rw out.txt
hujunyao@HJYUbuntu:~/LinuxCourse$ ls -l
总用量 4
-rwxrw-rw- 1 hujunyao hujunyao 2722 4月 1 17:08 out.txt
```

图 1.3-4

1.4 磁盘

题目：什么是 CHS、LBA？什么是 MBR、GPT？它们各自特点是什么？Linux 的常用磁盘管理命令有哪些（举例给出）？（15 分）

1.4.1 CHS、LBA 及其特点

CHS 和 LBA 都是指磁盘的寻址模式。CHS 和 LBA 类似于真实地址和虚拟地址的区别，二者可以使用公式相互映射，相互转换，但是查找地址的效率略有不同。

CHS: 是 **FDISK** 在分区期间所需的磁盘信息。这种寻址模式将硬盘划分为磁头 (Heads)、柱面(Cylinder)、扇区(Sector)，模式的名字即磁头、柱面、扇区的首字母缩写。相当于三级坐标，类似我国的“省、县、镇（乡）”三级行政区划，用第几磁头就可以表示数据在哪个磁面，用第几柱面可以表示数据在哪个磁道，用第几扇区就可以表示数据在某一磁道中的位置，一般来说，每个扇区的容量为 512B。知道磁头数、柱面数、扇区数，就可以确定数据在硬盘中的位置，也可以确定硬盘的容量，硬盘容量公式： $\text{硬盘容量} = \text{磁头数} \times \text{柱面数} \times \text{扇区数} \times 512\text{B}$ 。

LBA: 是指逻辑块寻址模式 (Logical Block Addressing)，是描述计算机存储设备上数据所在区块的通用机制。在 LBA 中，地址不再表示实际硬盘的实际物理地址 (CHS)，它将 CHS 这种三维寻址方式映射为一维的线性寻址，从 0 开始编号来定位区块，第一区块 LBA=0，第二区块 LBA=1，依此类推，效率大大提高，避免了烦琐的磁头/柱面/扇区的寻址方式。如果想访问真实物理地址，在访问硬盘时由硬盘控制器将这辑地址转换为实际的物理地址，只是多了一步转换。逻辑扇区号 LBA 的公式： $\text{LBA} = \text{磁头数} \times \text{每磁道扇区数} \times \text{当前所在柱面号} + \text{每磁道扇区数} \times \text{当前所在磁头号} + \text{当前所在扇区号} - 1$ 。

1.4.2 MBR、GPT 及其特点

MBR 和 GPT 都是指磁盘的分区技术。从技术改进和容量支持上看，相比于 MBR，GPT 的优势更为明显，也更合理一些。

MBR: 是指主分区引导记录 (Master Boot Record)。在 LBA 中，引导扇区是每个分区的第一扇区，而主引导扇区则是整个硬盘的第一扇区 (主分区的第一个扇区)。MBR 保存在主引导扇区中。在 MBR 中，分区表的大小是固定不变的，一共可容纳 4 个主分区信息。MBR 分区表中支持的硬盘最大分区容量上限大约为 2TB，对 2TB 以上容量的物理硬盘，不适合使用 MBR 分区方案。

GPT: 即全局唯一标识磁盘分区表 (Globally Unique Identifier Partition Table, GUID) 的缩写，这是实体硬盘的分区表的结构布局的标准。在 GTP 磁盘的第一个数据块中同样有一个与 MBR 类似的标记，叫做 PMBR。当不支持 GPT 的分区工具试图对硬盘进行操作时，它可以根据这份 PMBR 以传统方式启动，过程和 MBR+BIOS 完全一致，极大地提高了兼容性。而支持 GPT 的系统在检测 PMBR 后会直接跳到 GPT 表头读取分区表。GPT 的分区方案相对 MBR 来说更为先进，主要是 GPT 分区表头中可自定义分区数量的最大值，GPT 分区表的大小是可变的。相比 MBR，GPT 分区表则能够识别 2TB 以上的硬盘空间。除此之外，GPT 分区方案在硬盘的末端还有一个备份分区表，保证了分区信息不容易丢失。

下图展示了使用 MBR 分区和 GPT 分区的主要分区结构，可以知道虽然 GPT 结构更加复杂，但是支持的磁盘容量是要大很多的。

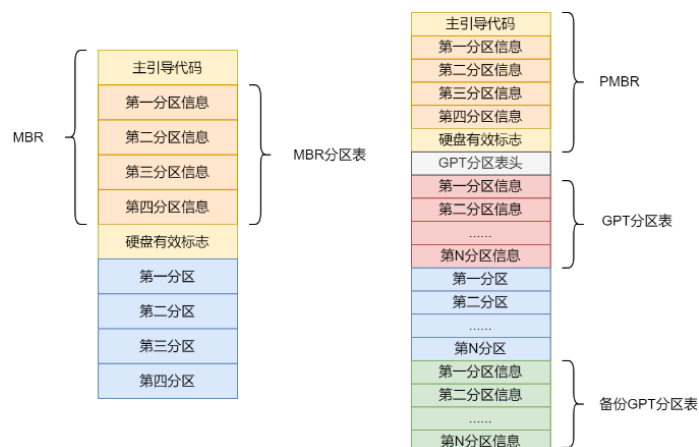


图 1.4-1 MBR 分区（左）与 GPT 分区（右）结构对比

1.4.3 df 命令

其作用是显示每个指定文件所在的文件系统的信息，默认是显示所有文件系统，也就是默认查看整个文件系统对于磁盘的使用情况（Disk full）。

df [选项]... [目录或文件]...

选项与参数：

- a：列出所有的文件系统，包含虚拟、重复和无法访问的文件系统
- h：以 GB、MB、KB 等格式显示大小
- T：显示文件系统类型
- i：不用硬盘容量，而以 inode 的数量来显示

下图实现了将系统内所有的文件系统列出，并显示文件系统类型。

```
hu@huynaoH3YUbuntu:~/桌面$ df -th
```

文件系统	类型	容量	已用	可用	已用%	挂载点
udev	devtmpfs	1.5G	0	1.5G	0%	/dev
tmpfs	tmpfs	295M	1.7M	294M	1%	/run
/dev/sda5	ext4	29G	19G	9.4G	67%	/
tmpfs	tmpfs	1.5G	0	1.5G	0%	/dev/shm
tmpfs	tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	tmpfs	1.5G	0	1.5G	0%	/sys/fs/cgroup
/dev/loop0	squashfs	128K	128K	0	100%	/snap/boot/5
/dev/loop1	squashfs	111M	111M	0	100%	/snap/core/12725
/dev/loop2	squashfs	220M	220M	0	100%	/snap/core/92
/dev/loop3	squashfs	56M	56M	0	100%	/snap/core18/2284
/dev/loop4	squashfs	210M	210M	0	100%	/snap/core/91
/dev/loop5	squashfs	62M	62M	0	100%	/snap/core20/1405
/dev/loop6	squashfs	56M	56M	0	100%	/snap/core18/2344
/dev/loop7	squashfs	111M	111M	0	100%	/snap/core/12821
/dev/loop8	squashfs	66M	66M	0	100%	/snap/gtk-common-themes/1515
/dev/loop9	squashfs	55M	55M	0	100%	/snap/snap-store/358
/dev/loop10	squashfs	219M	219M	0	100%	/snap/gnome-3-34-1804/77
/dev/loop11	squashfs	249M	249M	0	100%	/snap/gnome-3-38-2004/99
/dev/loop12	squashfs	44M	44M	0	100%	/snap/snapd/14978
/dev/loop13	squashfs	44M	44M	0	100%	/snap/snapd/15177
/dev/loop14	squashfs	219M	219M	0	100%	/snap/gnome-3-34-1804/72
/dev/loop15	squashfs	248M	248M	0	100%	/snap/gnome-3-38-2004/67
/dev/loop16	squashfs	66M	66M	0	100%	/snap/gtk-common-themes/1519
/dev/loop17	squashfs	62M	62M	0	100%	/snap/core20/1376
/dev/loop18	squashfs	51M	51M	0	100%	/snap/snap-store/547
/dev/sda1	vfat	511M	4.0K	511M	1%	/boot/efi
vmhgfs-fuse	fuse.vmhgfs-fuse	273G	243G	31G	89%	/mnt/hgfs
tmpfs	tmpfs	295M	36K	295M	1%	/run/user/1000
/dev/sr0	iso9660	2.9G	2.9G	0	100%	/media/hu@huynaoH3YUbuntu/Ubuntu 20.04.3 LTS amd64

图 1.4-2

下图实现了列出 opt 文件夹的文件系统的内存和节点使用情况。可以发现内存和节点的使用情况是不一样的，一般很少有这种情况，其实出现的根本原因是 inode 储存在一边，文件又存储在另一边，各自的内存大小的限制也不一样。

```
hu@huynaoH3YUbuntu:~$ df /opt -h
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/sda5	29G	19G	9.4G	67%	/

```
hu@huynaoH3YUbuntu:~$ df /opt -i
```

文件系统	Inodes	已用(I)	可用(I)	已用(I)%	挂载点
/dev/sda5	1933312	347862	1585450	18%	/

图 1.4-3

1.4.4 du 命令

其作用是，查看文件和目录磁盘使用的空间，即计算每个文件的磁盘用量，如果是目录则取总用量（Disk used），其实也就是查看该文件夹下文件还有文件夹的大小。

du [选项]... [目录或文件]...

选项与参数：

- a：输出所有文件的磁盘用量，不仅仅是目录
- c, --total：显示总计信息
- h：以 GB、MB、KB 等格式显示大小
- s：列出总量而已，而不列出每个各别的目录占用容量；
- S：不包括子目录的占用量
- k：以 KBytes 列出容量显示

下图实现了查询~/ROS_TCP_Server 文件夹的各文件夹内存大小（含子文件夹，不含文件，统计内存大小时加上子目录的占用量）、各文件夹内存大小（含子文件夹，统计内存大小时不加上子目录的占用量）、当前目录下各文件夹内存大小（不含子文件夹，加上子目录的占用量）、各文件夹和文件内存大小（含子文件夹，含文件）、各文件夹内存大小（含当前查询的文件夹的大小，即总用量）。

```
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ du -h /ROS_TCP_Server/src -h
8.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/scripts
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/src
28K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server
32K /home/hujunyang/ROS_TCP_Server/src
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ du -h /ROS_TCP_Server/src -hs
8.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/scripts
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/src
16K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server
4.0K /home/hujunyang/ROS_TCP_Server/src
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ du -h /ROS_TCP_Server/src -hs
32K /home/hujunyang/ROS_TCP_Server/src
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ du -ha /ROS_TCP_Server/src -ha
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/package.xml
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/scripts/hello.py
8.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/scripts
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/src
8.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/CMakeLists.txt
28K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server
0 /home/hujunyang/ROS_TCP_Server/src/CMakeLists.txt
32K /home/hujunyang/ROS_TCP_Server/src
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ du -hc /ROS_TCP_Server/src -hc
8.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/scripts
4.0K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server/src
28K /home/hujunyang/ROS_TCP_Server/src/ros_tcp_server
32K /home/hujunyang/ROS_TCP_Server/src
32K 总用量
```

图 1.4.4

1.4.5 fdisk 命令

用于显示或操作磁盘分区表。

fdisk [选项] <磁盘> 更改分区表
fdisk [选项] -l [<磁盘>] 列出分区表

下图实现了列出磁盘分区表的信息。

```
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ sudo fdisk /dev/sda -l
Disk /dev/sda: 30 GiB, 32212254720 字节, 62914560 个扇区
Disk model: VMware Virtual S
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: dos
磁盘标识符: 0x3f424b11

设备      启动    起点    末尾    扇区 大小 Id 类型
/dev/sda1 *      2048    1050623    1048576    512M  b W95 FAT32
/dev/sda2        1052670    62914559    61861890    29.5G  5 扩展
/dev/sda5        1052672    62912511    61859840    29.5G  83 Linux
hujunyang@HJYUbuntu:~/ROS_TCP_Server$ sudo fdisk /dev/sr0 -l
Disk /dev/sr0: 2.88 GiB, 3071934464 字节, 1499968 个扇区
Disk model: VMware SATA CD00
单元: 扇区 / 1 * 2048 = 2048 字节
扇区大小(逻辑/物理): 2048 字节 / 2048 字节
I/O 大小(最小/最佳): 2048 字节 / 2048 字节
磁盘标签类型: dos
磁盘标识符: 0x2cf4ba3e

设备      启动    起点    末尾    扇区 大小 Id 类型
/dev/sr0p1 *      0    5999871    5999872    11.5G  0 空
/dev/sr0p2      5271500    5279499      8000    15.0M  ef EFI (FAT-12/16/32)
```

图 1.4.5

下图实现了进入 fdisk 的交互界面，可根据提示完成获取磁盘信息、新增/删除一个分区等对磁盘的操作。对磁盘的操作首先位于内存中，只有通过 **w** 退出时才会将修改真正执行，相应的使用 **q** 退出则不会做修改。需要用户确定操作无误后再执行。

```
hujunyao@ubuntu:~$ sudo fdisk /dev/
/dev/loop0 /dev/loop12 /dev/loop16 /dev/loop3 /dev/loop7 /dev/sda1
/dev/loop1 /dev/loop13 /dev/loop17 /dev/loop4 /dev/loop8 /dev/sda2
/dev/loop10 /dev/loop14 /dev/loop18 /dev/loop5 /dev/loop9 /dev/sda5
/dev/loop11 /dev/loop15 /dev/loop2 /dev/loop6 /dev/sda /dev/sr0
hujunyao@ubuntu:~$ sudo fdisk /dev/sda
[sudo] hujunyao 的密码:
欢迎使用 fdisk (util-linux 2.42)
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 n 获取帮助): n

帮助:
DOS (MBR)
a  开启 可启动 标志
b  编辑磁盘的 dos 磁盘标签
c  开关 dos 兼容性标志

常规
d  删除分区
l  列出未分区的空闲区
t  列出已知分区类型
```

图 1.4-6

下图实现了显示 `/dev/sda` 的分区表，一共有三个分区，其中 `/dev/sda1` 是启动分区。

```
命令(输入 n 获取帮助): p
Disk /dev/sda: 30 GiB, 32212254720 字节, 62914560 个扇区
Disk model: VMware Virtual S
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: dos
磁盘标识符: 0x3f424b11

设备      启动   起点   末尾   扇区  大小  Id 类型
/dev/sda1 *    2048  1050623  1048576  512M  b  W95 FAT32
/dev/sda2      1052670 62914559 61861890 29.5G  5  扩展
/dev/sda5      1052672 62912511 61859840 29.5G  83  Linux
```

图 1.4-7

下图实现了删除 `/dev/sda5` 分区，再查看硬盘分区情况，发现已经删除，然后新建一个分区，从 1054718 开始，大小为 2000000，也就是末尾到 3054718，大小为 976.6M，类型为 Linux，分区为 `/dev/sda5`。以上步骤在内存中进行，然后不保存退出。

```
命令(输入 n 获取帮助): d
分区号 (1,2,5, 默认 5): 5
分区 5 已删除。

命令(输入 n 获取帮助): p
Disk /dev/sda: 30 GiB, 32212254720 字节, 62914560 个扇区
Disk model: VMware Virtual S
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: dos
磁盘标识符: 0x3f424b11

设备      启动   起点   末尾   扇区  大小  Id 类型
/dev/sda1 *    2048  1050623  1048576  512M  b  W95 FAT32
/dev/sda2      1052670 62914559 61861890 29.5G  5  扩展

命令(输入 n 获取帮助): n
所有主分区的空间都在使用中。
添加逻辑分区 5
第一个扇区 (1054718-62914559, 默认 1054718): 1054718
Last sector, +/-sectors or +/-size(K,M,G,T,P) (1054718-62914559, 默认 62914559): +2000000
创建了一个新分区 5, 类型为"Linux", 大小为 976.6 MiB。

命令(输入 n 获取帮助): p
Disk /dev/sda: 30 GiB, 32212254720 字节, 62914560 个扇区
Disk model: VMware Virtual S
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: dos
磁盘标识符: 0x3f424b11

设备      启动   起点   末尾   扇区  大小  Id 类型
/dev/sda1 *    2048  1050623  1048576  512M  b  W95 FAT32
/dev/sda2      1052670 62914559 61861890 29.5G  5  扩展
/dev/sda5      1054718 3054718 2000001 976.6M  83  Linux
```

图 1.4-8

1.5 文件系统

题目: 什么是 ext4 文件系统? Linux 的常用文件系统管理命令有哪些(举例给出)? (15 分)

1.5.1 ext4 文件系统

文件系统是操作系统用来合理组织存储不同分区的文件的方法,使用一种数据结构来对文件进行管理,而 ext4 文件系统指的是 Linux 第四代扩展文件系统(Fourth extended filesystem),这是一种 Linux 系统下的日志文件系统,也是 ext3 文件系统的后继版本,目前的大部分 Linux 文件系统也都默认采用了 ext4 文件系统。

1.5.2 mkfs 命令

格式化也就是做一个新的文件系统(Make file system),其作用是选择一个分区后,在其上面新的建立 Linux 文件系统。

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

选项:

-t, --type=<类型> : 文件系统类型;若不指定,将使用 ext2

下图实现了查看命令的版本和该命令支持的文件格式。

```
hujunyao@H3YUbuntu:~/桌面$ mkfs -V
mkfs, 来自 util-linux 2.34
hujunyao@H3YUbuntu:~/桌面$ mkfs
mkfs      mkfs.cramfs  mkfs.ext3  mkfs.fat   mkfs.msdos  mkfs.vfat
mkfs.bfs  mkfs.ext2  mkfs.ext4  mkfs.minix mkfs.ntfs
```

图 1.5-1

下图实现了即将格式化/dev/sda 分区,并指定为 ext4 类型。

```
hujunyao@H3YUbuntu:~/桌面$ sudo mkfs -t ext4 /dev/sda
[sudo] hujunyao 的密码:
mkfs: 1.45.5 (07-Jan-2020)
在 dos 中发现一个 /dev/sda 分区表
无论如何也要继续? (y,n)
```

图 1.5-2

1.5.3 fsck 命令

fsck 用于检测并修复 Linux 文件系统中的错误,一般会在系统启动时运行。

fsck [选项] -- [文件系统选项] [<文件系统> ...]

参数与选项:

- A : 检查所有文件系统
- s : 串行检查序列。如果有多个文件系统需要检查,那么按照顺序来操作。
- t <类型> : 指定要检查的文件系统类型
- C [<fd>] : 显示进度条;文件描述符 fd 用于图形用户界面
- M : 不检查已挂载的文件系统
- P : 并行检查文件系统,包括 root
- R : 跳过根文件系统;只在指定 '-A' 时有用
- r [<fd>] : 报告对每个已检查设备的统计;文件描述符用于 GUI
- V : 解释正在进行的操作

下图实现了对所有文件系统进行检测，检测到有文件系统已经挂载，系统提示可能出现的问题，于是继续排除已经挂载的文件系统，然后输出过程中产生的信息。

```
hujunyang@ubuntu:~/桌面$ fsck -A
fsck, 来自 util-linux 2.34
e2fsck 1.45.5 (07-Jan-2020)
/dev/sda5 已挂载。

警告!!! 该文件系统已被挂载。如果你继续操作将会
使文件系统遭受 *** 严重损坏 ***!

你真的要继续吗<n>? 已取消!
检查被中止。
hujunyang@ubuntu:~/桌面$ fsck -N -V
fsck, 来自 util-linux 2.34
正在检查所有文件系统。
UUID=33fce493-955c-42fa-aa24-e4766d3b6523 已经挂载
UUID=36fd-4c6d 已经挂载
```

图 1.5-3

1.5.4 dumpe2fs 命令

其作用是查看文件的详细信息，使用该命令会打印文件系统的超级块和块组信息，但是，当与挂载的文件系统一起使用时，打印的信息可能是旧的或不一致的。

dumpe2fs [选项] [设备名]

- b : 打印文件系统中预留的块信息
- h : 仅显示超级块信息
- i : 从指定的文件系统映像文件中读取文件系统信息
- x : 以 16 进制格式打印信息块成员

下图实现了对 `/dev/sda5` 分区实现显示预留的块信息、超级块信息。

```
hujunyang@ubuntu:~/桌面$ sudo dumpe2fs /dev/sda5 -b
dumpe2fs 1.45.5 (07-Jan-2020)
dumpe2fs: ext2 超级块中的幻数有错 尝试打开 /dev/sda5 时
找不到有效的文件系统超级块。
hujunyang@ubuntu:~/桌面$ sudo dumpe2fs /dev/sda5 -h
dumpe2fs 1.45.5 (07-Jan-2020)
Filesystem volume name: <none>
Last mounted on: /
Filesystem UUID: 33fce493-955c-42fa-aa24-e4766d3b6523
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype needs_recovery
extent 64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags: signed_directory_hash
Default mount options: user_xattr acl
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 193312
Block count: 7732480
```

图 1.5-4

下图实现了对 `/dev/sda5` 分区以 16 进制打印信息块成员，因信息较多只截取了部分。

```
组 0: (块 0x00000000-0x00007fff) 校验值 0xa586 [ITABLE_ZEROED]
主 超级块位于 0x00000000, 组描述符位于 0x00000001-0x00000004
保留的GDT位于 0x00000005-0x00000403
块位图位于 0x00000404 (+1028), 校验值 0x8b294024
Inode 位图位于 0x00000414 (+1044), 校验值 0xfed97e56
Inode表位于 0x00000424-0x00000623 (+1060)
3552 个可用 块, 7666 个可用Inode, 80 个目录, 5622个未使用的Inodes
可用块数: 0x00002c29-0x00002c34, 0x00002c38-0x00002c39, 0x00002c3c-0x00002c3d, 0x00002c40-0x00002c7b, 0x00002c80-0x00002d13, 0x00002d15-0x00002d7f, 0x00002d8d, 0x00002d90-0x00002d97, 0x00002d9a-0x00002d9d, 0x00002da5, 0x00002db2, 0x00002db6, 0x00002db9-0x00002dba, 0x00002dc0-0x00002ddf, 0x00002de1, 0x00002de3-0x00002de5, 0x00003000-0x000030c8, 0x00003caa-0x00003ce5, 0x00003030-0x0000412b, 0x0000428a-0x000042b7, 0x00005b78-0x00005fff, 0x00006d4b-0x00007058, 0x00007072-0x000079fa, 0x00007fc2-0x00007fff
可用Inode数: 0x00000016-0x000000bc, 0x000000be-0x000000cc, 0x000000ce-0x000000dd, 0x000000df, 0x000000e1-0x000000e3, 0x000000e7-0x000000fb, 0x000000fd-0x00000154, 0x00000156-0x0000015a, 0x0000015d-0x0000015e, 0x0000016a-0x00000176, 0x00000179-0x0000019a, 0x00000256, 0x00000283-0x0000028d, 0x00000a0b-0x00002000
组 1: (块 0x00000000-0x0000ffff) 校验值 0xa545 [ITABLE_ZEROED]
备份 超级块位于 0x00000000, 组描述符位于 0x00000001-0x00000004
保留的GDT位于 0x00000005-0x00000403
块位图位于 0x00000405 (bg #0 + 1029), 校验值 0xf4ac3955
Inode 位图位于 0x00000415 (bg #0 + 1045), 校验值 0xd1d51634
Inode表位于 0x00000424-0x00000623 (bg #0 + 1072)
1019 个可用 块, 8155 个可用Inode, 37 个目录, 8154个未使用的Inodes
可用块数: 0x00000404-0x00000412, 0x00000414-0x000007ff
可用Inode数: 0x00002001, 0x00002027-0x00004000
组 2: (块 0x00010000-0x0001ffff) 校验值 0x6651 [INODE_UNINIT, ITABLE_ZEROED]
块位图位于 0x00000406 (bg #0 + 1030), 校验值 0xfdd12cf2
Inode 位图位于 0x00000415 (bg #0 + 1046), 校验值 0x00000000
Inode表位于 0x00000824-0x00000a23 (bg #0 + 2084)
0 个可用 块, 8192 个可用Inode, 0 个目录, 8192个未使用的Inodes
```

图 1.5-5

1.6 挂载

题目：什么是 Linux 的挂载？挂载的作用是什么？常用的挂载命令有哪些（举例给出）？（10 分）

1.6.1 定义与作用

Linux 系统里面所有事物都是以文件的形式存在的，包括各种硬件设备也是如此，都有一个文件夹，这些所有文件都放置文件夹中，但是它们的文件系统是各是一套，文件目录结构不一样。如果 Linux 系统想使用这些硬件设备时，只有将 Linux 本身的文件目录与硬件设备的文件目录合二为一才能使用，分开是没有办法进行使用的。两套文件系统合并起来，访问这个目录就等同访问这个目录对应的设备文件，这个合二为一的过程就叫做挂载（Mount），而这个目录就叫做挂载点。如果不进行挂载，尽管通过 Linux 的 GUI 界面查看，找到硬件设备，但命令行的方式无法找到。

1.6.2 mount 命令

该命令用于挂载文件系统。

```
mount [-lhV]
mount -a [选项]
mount [选项] [--source] <源> | [--target] <目录>
mount [选项] <源> <目录>
mount <操作> <挂载点> [<目标>]
选项：
-a, --all                挂载 fstab 中的所有文件系统
-o, --options <列表>    挂载选项列表，以英文逗号分隔
-O, --test-opts <列表>  限制文件系统集合(和 -a 选项一起使用)
-r, --read-only          以只读方式挂载文件系统(同 -o ro)
-t, --types <列表>      限制文件系统类型集合
    --source <源>        指明源(路径、标签、uuid)
    --target <目标>      指明挂载点
-v, --verbose            打印当前进行的操作
-w, --rw, --read-write  以读写方式挂载文件系统(默认)
源：
-L, --label <标签> : 同 LABEL=<label>
-U, --uuid <uuid> : 同 UUID=<uuid>
LABEL=<标签> : 按文件系统标签指定设备
UUID=<uuid> : 按文件系统 UUID 指定设备
PARTLABEL=<标签> : 按分区标签指定设备
PARTUUID=<uuid> : 按分区 UUID 指定设备
<设备>      按设备路径指定设备
<目录>      绑定式挂载的挂载点(参阅 --bind/rbind)
<文件>      用于设置回环设备的常规文件
```

下图实现了直接输入命令，获取/etc/mtab 文件夹内的已挂载设备信息。

```
hujunyao@H3YUbuntu:~/桌面$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=1474948k,nr_inodes=368737,node=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,node=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=381684k,node=755,inode64)
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,node=755,inode64)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
```

图 1.6-1

下图实现了对于未挂载的新的磁盘设备/dev/sda6 挂载到/media/hujunyao/newdisk 文件夹下，同时使用了 df 命令对是否挂载上进行了检验。

```
hujunyao@H3YUbuntu:~/桌面$ mount /dev/sda /media/hujunyao/newdisk
hujunyao@H3YUbuntu:~/桌面$ mount /dev/sda6 /media/hujunyao/newdisk
mount: 只有 root 能执行该操作
hujunyao@H3YUbuntu:~/桌面$ sudo mount /dev/sda6 /media/hujunyao/newdisk
hujunyao@H3YUbuntu:~/桌面$ sudo mkdir /media/hujunyao/newdisk
hujunyao@H3YUbuntu:~/桌面$ sudo mount /dev/sda6 /media/hujunyao/newdisk
hujunyao@H3YUbuntu:~/桌面$ df
文件系统      1K-块      已用    可用  已用% 挂载点
udev          1474948      0   1474948     0% /dev
tmpfs         381684      1668    380016     1% /run
/dev/sda5     30313412 19231760  9623768    67% /
tmpfs        1508416      0   1508416     0% /dev/shm
tmpfs         5120         4     5116     1% /run/lock
tmpfs        1508416      0   1508416     0% /sys/fs/cgroup
/dev/loop0      128        128     0 100% /snap/bare/5
/dev/loop1    224512    224512     0 100% /snap/code/92
/dev/loop2     63488     63488     0 100% /snap/core20/1485
/dev/loop3    214144    214144     0 100% /snap/code/91
/dev/loop4     56960     56960     0 100% /snap/core18/2284
/dev/loop5     63488     63488     0 100% /snap/core20/1376
/dev/loop6     44800     44800     0 100% /snap/snapd/15177
/dev/loop7     56960     56960     0 100% /snap/core18/2344
/dev/loop8     44672     44672     0 100% /snap/snapd/14978
/dev/loop9    113408    113408     0 100% /snap/core/12821
/dev/loop10   254848    254848     0 100% /snap/gnome-3-38-2004/99
/dev/loop11   224256    224256     0 100% /snap/gnome-3-34-1804/72
/dev/loop12   224256    224256     0 100% /snap/gnome-3-34-1804/77
/dev/loop13   113536    113536     0 100% /snap/core/12725
/dev/loop14   253952    253952     0 100% /snap/gnome-3-38-2004/87
/dev/loop15     66688     66688     0 100% /snap/gtk-common-themes/1515
/dev/loop16   52224     52224     0 100% /snap/snap-store/547
/dev/loop17     66816     66816     0 100% /snap/gtk-common-themes/1519
/dev/loop18     55552     55552     0 100% /snap/snap-store/558
/dev/sda1       523244      4    523244     1% /boot/efi
vmhgfs-fuse   286072828 258852588 27220240    91% /mnt/hgfs
tmpfs         301680      32    301648     1% /run/user/1000
/dev/sr0       2999936    2999936     0 100% /media/hujunyao/Ubuntu 20.04.3 LTS amd64
/dev/sda6     5095040    20472   4796040     1% /media/hujunyao/newdisk
```

图 1.6-2

1.6.3 umount 命令

和 mount 相反，这个命令刚好是为了进行卸载文件系统，和挂载相反。

umount [-hV]

umount -a [选项]

umount [选项] <源> | <目录>

选项：

-a, --all	卸载所有文件系统
-A, --all-targets	卸载当前名字空间内指定设备对应的所有挂载点
-l, --lazy	立即断开文件系统，清理以后执行
-O, --test-opts <列表>	限制文件系统集合(和 -a 选项一起使用)
-R, --recursive	递归卸载目录及其子对象
-r, --read-only	若卸载失败，尝试以只读方式重新挂载
-t, --types <列表>	限制文件系统集合
-v, --verbose	打印当前进行的操作

下图实现了对于已经挂载到/media/hujunyao/newdisk 文件夹下的磁盘设备/dev/sda6 进行卸载，显示卸载过程，同时使用了 df 命令对是否卸载进行了检验。

```

hujunyao@HJYUbuntu:~/桌面$ sudo mount /dev/sda6 -v
mount: /media/hujunyao/newdisk (/dev/sda6) 已卸载
hujunyao@HJYUbuntu:~/桌面$ df
文件系统      1K-块      已用      可用 已用% 挂载点
udev          1474948         0   1474948    0% /dev
tmpfs         301684      1668    300016    1% /run
/dev/sda5     30313412  19231784   9623744   67% /
tmpfs        1508416         0   1508416    0% /dev/shm
tmpfs         5120         4     5116    1% /run/lock
tmpfs        1508416         0   1508416    0% /sys/fs/cgroup
/dev/loop0       128       128         0  100% /snap/bare/5
/dev/loop1     224512    224512         0  100% /snap/code/92
/dev/loop2     63488     63488         0  100% /snap/core20/1405
/dev/loop3     214144    214144         0  100% /snap/code/91
/dev/loop4      56960     56960         0  100% /snap/core18/2284
/dev/loop5     63488     63488         0  100% /snap/core20/1376
/dev/loop6     44800     44800         0  100% /snap/snapd/15177
/dev/loop7     56960     56960         0  100% /snap/core18/2344
/dev/loop8     44672     44672         0  100% /snap/snapd/14978
/dev/loop9     113408    113408         0  100% /snap/core/12821
/dev/loop10    254848    254848         0  100% /snap/gnome-3-38-2004/99
/dev/loop11    224256    224256         0  100% /snap/gnome-3-34-1804/72
/dev/loop12    224256    224256         0  100% /snap/gnome-3-34-1804/77
/dev/loop13    113536    113536         0  100% /snap/core/12725
/dev/loop14    253952    253952         0  100% /snap/gnome-3-38-2004/87
/dev/loop15     66688     66688         0  100% /snap/gtk-common-themes/1515
/dev/loop16     52224     52224         0  100% /snap/snap-store/547
/dev/loop17     66816     66816         0  100% /snap/gtk-common-themes/1519
/dev/loop18     55552     55552         0  100% /snap/snap-store/558
/dev/sda1      523248         4   523244    1% /boot/efi
vmhgfs-fuse   286072828 258852732 27220696   91% /mnt/hgfs
tmpfs         301680       32    301648    1% /run/user/1000
/dev/sr0      2999936   2999936         0  100% /media/hujunyao/Ubuntu 20.04.3 LTS amd64

```

图 1.6-3

1.6.4 lsblk 命令

主要是列出了块设备的信息，其中可以体现块设备的挂载点的位置。

下图实现了查看挂载和没有挂载的所有各种文件系统类型信息。

```

hujunyao@HJYUbuntu:~/桌面$ lsblk -l
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0    4K  1 loop /snap/bare/5
loop1       7:1      0   219.2M 1 loop /snap/code/92
loop2       7:2      0   61.9M 1 loop /snap/core20/1405
loop3       7:3      0   209.1M 1 loop /snap/code/91
loop4       7:4      0   55.5M 1 loop /snap/core18/2284
loop5       7:5      0   61.9M 1 loop /snap/core20/1376
loop6       7:6      0   43.6M 1 loop /snap/snapd/15177
loop7       7:7      0   55.5M 1 loop /snap/core18/2344
loop8       7:8      0   43.6M 1 loop /snap/snapd/14978
loop9       7:9      0   110.7M 1 loop /snap/core/12821
loop10      7:10     0   248.8M 1 loop /snap/gnome-3-38-2004/99
loop11      7:11     0    219M 1 loop /snap/gnome-3-34-1804/72
loop12      7:12     0    219M 1 loop /snap/gnome-3-34-1804/77
loop13      7:13     0   110.8M 1 loop /snap/core/12725
loop14      7:14     0   247.9M 1 loop /snap/gnome-3-38-2004/87
loop15      7:15     0    65.1M 1 loop /snap/gtk-common-themes/1515
loop16      7:16     0    51M 1 loop /snap/snap-store/547
loop17      7:17     0    65.2M 1 loop /snap/gtk-common-themes/1519
loop18      7:18     0    54.2M 1 loop /snap/snap-store/558
sda         8:0      0    35G  0 disk 
sda1        8:1      0   512M  0 part /boot/efi
sda2        8:2      0    1K  0 part 
sda5        8:5      0   29.5G  0 part /
sda6        8:6      0    5G  0 part 
sr0         11:0     1    2.9G  0 rom  /media/hujunyao/Ubuntu 20.04.3 LTS amd64

```

图 1.6-4

1.6.5 findmnt 命令

该命令用于列出 Linux 中已挂载的文件系统。该命令将在 `/etc/fstab`、`/etc/mtab`、`/proc/self/mountinfo` 中查找特定的文件系统。

下图实现了查看部分分区对应的挂载点。

```

hujunyao@HJYUbuntu:~/桌面$ findmnt /dev/sda1
TARGET SOURCE FSTYPE OPTIONS
/boot/efi /dev/sda1 vfat rw,relatime,fnask=0077,dnask=0077,codepage=437,iocharset=iso8859-1,
hujunyao@HJYUbuntu:~/桌面$ findmnt /dev/sda6
hujunyao@HJYUbuntu:~/桌面$ findmnt /dev/sda5
TARGET SOURCE FSTYPE OPTIONS
/ /dev/sda5 ext4 rw,relatime,errors=remount-ro

```

图 1.6-5

1.7 用户、组与权限

题目：Linux 用户、组及其权限的信息存放在什么文件里，这些文件存放在什么位置？，文件内的信息组织格式是什么？（15 分）

在 Linux 中，万物皆文件，所以用户与组，以及他们的权限，都是以配置文件的形式保存在系统当中，主要有四个文件，这四个文件都存放在根目录下的 *etc* 文件夹，以下为用户和组的主要配置文件和放置位置以及信息组织格式的介绍。

1.7.1 用户及其属性信息

存放的文件及位置：*/etc/passwd* 文件。

信息组织格式：下图实现了使用 `cat` 命令读取 */etc/passwd* 文件，可以查看用户及其属性信息的具体内容。

```
hujunyao@HJYUbuntu:~/桌面$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

图 1.7-1 1.7.1 用户及其属性信息文件

下面代码是读取的第一行内容，每一行内容类似。

```
root:x:0:0:root:/root:/bin/bash
```

每一项内容由冒号隔开，每一项分别是：登录名（*root*）、可选的加密后的密码（*x*）、数字用户 ID（*0*）、数字组 ID（*0*）、用户名和注释字段（*root*）、用户主目录（*/root*）、可选的用户命令解释器（*/bin/bash*）。

1.7.2 用户密码及其相关属性

存放的文件及位置：*/etc/shadow* 文件。

信息组织格式：使用 `cat` 命令读取 */etc/shadow* 文件，可以查看用户密码及其相关属性的具体内容。

```
hujunyao@HJYUbuntu:~/桌面$ sudo cat /etc/shadow
[sudo] hujunyao 的密码:
root!:19006:0:99999:7:::
daemon*:18858:0:99999:7:::
bin*:18858:0:99999:7:::
sys*:18858:0:99999:7:::
sync*:18858:0:99999:7:::
games*:18858:0:99999:7:::
man*:18858:0:99999:7:::
```

图 1.7-2 1.7.2 用户密码及其相关属性文件

下面代码是读取的第一行内容，每一行内容类似。

```
root!:19006:0:99999:7:::
```

每一项内容由冒号隔开，每一项分别是：登录名（*root*）、可选的加密后的密码（*!*）、最后一次更改密码的日期（最近一次更改密码的时间，表示从 1970 年 1 月 1 日开始的天数，19006）、密码的最小年龄（用户一次更改密码之后，要等多长时间才再次被允许更改密码，这里是 0）、最大密码年龄（这写天之后，用户必须更改密码，这里是 99999）、密码警告时间段（密码过期之前，提前警告用户的天数，这里是 7，也就是提前一周）、

密码禁用期（密码过期后，仍然接受此密码的天数，空字段表示没有强制密码过期）、账户过期日期（空字段表示账户永不过期）、保留字段。

1.7.3 组及其属性信息

存放的文件及位置：*/etc/group* 文件。

信息组织格式：使用 `cat` 命令读取*/etc/group* 文件，可以查看组及其属性信息的具体内容。

```
hujunyao@H2YUbuntu:~/桌面$ sudo cat /etc/group
cat: /etc/group: 没有那个文件或目录
hujunyao@H2YUbuntu:~/桌面$ sudo cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,hujunyao
tty:x:5:syslog
disk:x:6:
lp:x:7:
```

图 1.7-3 1.7.3 组及其属性信息文件

下面代码是读取的其中一行的内容，每一行内容类似。

```
sudo:x:27:hujunyao
```

每一项内容由冒号隔开，每一项分别是：属组名（`sudo`）、可选的加密后的密码（`x`）、数字组 ID（27）、用户列表（这里是这个组中的一系列用户的列表，使用逗号分隔，这里只有用户 `hujunyao`）。

1.7.4 组密码及其相关属性

存放的文件及位置：*/etc/gshadow* 文件。

信息组织格式：使用 `cat` 命令读取*/etc/gshadow* 文件，可以查看组密码及其相关属性信息的具体内容。

```
hujunyao@H2YUbuntu:~/桌面$ sudo cat /etc/gshadow
root:*:
daemon:*:
bin:*:
sys:*:
adm:*:syslog,hujunyao
tty:*:syslog
disk:*:
lp:*:
mail:*:
news:*:
```

图 1.7-4 1.7.4 组密码及其相关属性文件

下面代码是读取的其中一行的内容，每一行内容类似。

```
sudo:*:*:hujunyao
```

每一项内容由冒号隔开，每一项分别是：属组名（`sudo`）、可选的加密后的密码（`*`）、组管理员列表（这里没有）、用户列表（这里只有用户 `hujunyao`）。

1.8 时间

题目：Linux 的硬件时间和系统时间有什么区别？Linux 的日期、时间命令有哪些（举例给出）？（10 分）

1.8.1 硬件时间和系统时间

硬件时间：指的是主板上 BIOS 的时间，在主板上有一个给 BIOS 供电的电池，在系统开机的时候需要读取该时间，并根据它来设定系统时间，正是因为有这个单独的电池才能在持续断电、关机的情况下，电脑仍然可以进行计时，刚开机联网前的时间和实际时间基本一致的原因也在于此。

系统时间：这个时间就是执行 `date` 命令看到的时间，Linux 系统下所有的时间调用都基于这个时间，除了那些直接调用硬件时间的命令使用硬件时间，如 `hwclock` 等。

1.8.2 date 命令

以给定格式的字符串的形式显示当前时间，或者设置系统日期，也就是和系统时间有关的一系列操作在这完成。

```
date [选项]... [+格式]
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

选项与参数：

- d, --date=字符串：显示给定<字符串>描述的时间，而非“当前时间”
- %c：当前 locale 的日期和时间
- %X：显示 24 小时的格式
- R：查看时区

下图实现了首先查询了当前时间，然后将一个时间戳以本地日期格式进行输出，然后以管理员权限设置当前时间为凌晨零点五分，并输出当前系统时间，可知已被修改，经过一分钟左右之后，系统时间自动更新为正确的时间。

```
hu@junyao@H3YUubuntu:~/桌面$ date +%X
20时16分25秒
hu@junyao@H3YUubuntu:~/桌面$ date
2022年 04月 06日 星期三 20:16:37 CST
hu@junyao@H3YUubuntu:~/桌面$ date --date='@2147483647'
2038年 01月 19日 星期二 11:14:07 CST
hu@junyao@H3YUubuntu:~/桌面$ date -s '00:05:00'
date: 无法设置日期: 不允许的操作
2022年 04月 06日 星期三 00:05:00 CST
hu@junyao@H3YUubuntu:~/桌面$ sudo date -s '00:05:00'
2022年 04月 06日 星期三 00:05:00 CST
hu@junyao@H3YUubuntu:~/桌面$ date
2022年 04月 06日 星期三 00:05:01 CST
hu@junyao@H3YUubuntu:~/桌面$ # date +%c'
hu@junyao@H3YUubuntu:~/桌面$ date +%c'
2022年04月06日 星期三 00时05分55秒
hu@junyao@H3YUubuntu:~/桌面$ date +%c'
2022年04月06日 星期三 20时18分31秒
```

图 1.8-1

1.8.3 tzselect 命令

该命令用于时区的选择（Timezone Select），使用该命令选择自己所处国家或地区，使用本地的时区。

下图实现了时区的选择，选择中国上海作为本地时区的参考。可能还需要手动执行一步 `TZ='Asia/Shanghai'; export TZ` 之后才能更新时区。

```

hujunyao@HJYUbuntu:~/桌面$ tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
1) Africa
2) Americas
3) Antarctica
4) Asia
5) Atlantic Ocean
6) Australia
7) Europe
8) Indian Ocean
9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the timezone using the Postx TZ format.
#?

The following information has been given:

        China
        Beijing Time

Therefore TZ='Asia/Shanghai' will be used.
Selected time is now:  Wed Apr  6 20:23:34 CST 2022.
Universal Time is now:  Wed Apr  6 12:23:34 UTC 2022.
Is the above information OK?
1) Yes
2) No
#? 1

You can make this change permanent for yourself by appending the line
TZ='Asia/Shanghai'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Asia/Shanghai

```

图 1.8-2

1.8.4 hwclock 命令

该命令用于获取和设定硬件时钟（Hardware Clock）。

hwclock [function] [option...]

选项与参数：

- a：根据先前的记录来估算硬件时钟的偏差，并用来校正目前的硬件时钟。
- set --date=<日期与时间>：定硬件时钟
- show：显示硬件时钟的时间与日期
- w, --systohc：将硬件时钟调整为与目前的系统时钟一致
- s, --hctosys：将系统时钟调整为与目前的硬件时钟一致

下图实现了获取硬件时间，将硬件时钟调整为与目前的系统时钟一致，将系统时钟调整为与目前的硬件时钟一致（在系统时钟自动更新之前实现），以及根据先前的记录来估算硬件时钟的偏差。

```

hujunyao@HJYUbuntu:~/桌面$ sudo hwclock
[sudo] hujunyao 的密码:
2022-04-06 20:41:16.074582+08:00
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock -a
需要的调整不到一秒，将不设置时钟。
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock --show
2022-04-06 20:41:41.684174+08:00
hujunyao@HJYUbuntu:~/桌面$ sudo date -s '00:05:00'
2022年 04月 06日 星期三 00:05:00 CST
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock -w
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock
2022-04-06 00:05:11.887325+08:00
hujunyao@HJYUbuntu:~/桌面$ date
2022年 04月 06日 星期三 00:05:23 CST
hujunyao@HJYUbuntu:~/桌面$ date
2022年 04月 06日 星期三 20:43:37 CST
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock
[sudo] hujunyao 的密码:
2022-04-06 00:06:06.746651+08:00
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock -s
hujunyao@HJYUbuntu:~/桌面$ date
2022年 04月 06日 星期三 00:06:27 CST
hujunyao@HJYUbuntu:~/桌面$ date
2022年 04月 06日 星期三 20:44:33 CST
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock
2022-04-06 00:07:03.856033+08:00
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock -w
hujunyao@HJYUbuntu:~/桌面$ sudo hwclock
2022-04-06 20:44:47.543420+08:00

```

图 1.8-3

1.8.5 cal 命令

这个命令可以用来在命令行显示月历、年历。

cal [**参数**][**月份**][**年份**]

参数和选项：

-m : 显示指定的月份

-j : 显示在当年中的第几天

-3 : 上个月、这个月、下个月这三个月的月历

下图实现了显示今年 12 月月历、显示今天在当年是第几天第几周、显示当前时间前中后三个月。

```
hu junyao@H3YUbuntu:~/桌面$ cal -m 12
      十二月 2022
日 一 二 三 四 五 六
           1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

hu junyao@H3YUbuntu:~/桌面$ cal -j
      四月 2022
日 一 二 三 四 五 六
           91 92
93 94 95 96 97 98 99
100 101 102 103 104 105 106
107 108 109 110 111 112 113
114 115 116 117 118 119 120

hu junyao@H3YUbuntu:~/桌面$ cal -3
      三月 2022      四月 2022      五月 2022
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 6  7  8  9 10 11 12  3  4  5  6  7  8  9  8  9 10 11 12 13 14
13 14 15 16 17 18 19 10 11 12 13 14 15 16 15 16 17 18 19 20 21
20 21 22 23 24 25 26 17 18 19 20 21 22 23 22 23 24 25 26 27 28
27 28 29 30 31      24 25 26 27 28 29 30 29 30 31
```

图 1.8-4

二、感悟

Linux 操作系统有很多命令，课程的学习也是主要围绕命令展开，所以光在课上听理论肯定是不够的，要想熟练肯定得不断进行实践，往往实践出真知，特别是课上的 CentOS 和 Ubuntu 操作系统在执行的具体细节上还有一些差别，需要自行在网上查询更多的资料去了解，但是大体的东西都是一致的，这也体现了 Linux 操作系统的便利。

在学习 Linux 的过程中，需要理论和实践相结合，做到知行合一，才能学好，主要还是得自己多去尝试，以前只是使用该系统进行 ROS 的一些相关操作，只知道一些最基本的命令，通过这次作业，巩固了对各种命令的操作和作用的理解，也加深了对操作系统、计算机组成原理等其他课程的一些认识。

参考文献

- [1] 鸟哥. 鸟哥的 Linux 私房菜（第四版）[M]. 人民邮电出版社, 2019.
- [2] 刘遒. Linux 就该这么学[M]. 人民邮电出版社, 2017.
- [3] 菜鸟教程. Linux 教程[OL]. <https://www.runoob.com/linux/linux-tutorial.html>.