

目录：

1.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**Ubuntu**安装后根目录下的一级目录的名称及存放的信息类别

1.2Linux常用目录命令：

cd命令

ls命令

pwd命令

mkdir命令

rmdir命令

2.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）？

2.1命令帮助方式及其特点

2.2举例

3.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）

4.什么是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）

4.1CHS与LBA是两种磁盘寻址方式。

4.2MBR与GPT则是两种磁盘分区方式。

4.3常用磁盘管理命令

5.什么是ext4文件系统？linux的常用文件系统管理命令有哪些（举例给出）

5.1ext4文件系统介绍

5.2文件系统管理命令

6.什么是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）

6.1挂载及其作用

6.2常用命令：

7.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）

7.1存放各类信息的文件位置

7.2各文件的作用以及信息组织格式

/etc/passwd文件

/etc/shadow文件

/etc/group文件

/etc/sudoers

8.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.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）

1.1Ubuntu安装后根目录下的一级目录的名称及存放的信息类别

当在一台Linux系统的机器上执行ls / 命令时，终端会向我们展示根目录下的全部文件信息。如下图所示（租用的一台阿里云服务器，安装了Ubuntu18.04版本）：

```
root@iZ8vb48stipso6rv9jbbqeZ:/# ls
bin    etc    initrd.img.old  lost+found  opt    root  srv    tmp    vmlinuz
boot   home   lib             media       patch  run   swapfile  usr    vmlinuz.old
dev    initrd.img  lib64          mnt         proc   sbin  sys     var     www
```

除了个别Ubuntu独有的目录外，这里挑选了一些全部Linux系统都存在的一级目录，其名称与存放的信息类别如下：

- **/bin:** bin是Binaries的缩写。该目录下主要存放着一些系统启动时所需要的普通程序和系统程序。此外，还存放着常用的命令对应的可执行程序如cat、cp、rm、su等。
- **/boot:** 该目录下存放了与系统激活的核心文件，包括initrd.img、vmlinuz等，因此本目录不可随意删除。其中，initrd.img为系统激活时最先加载的文件，vmlinuz为kernel的image文件。此外，该目录下还存放了一些连接文件和镜像文件。
- **/dev:** dev是Device的缩写。该目录下主要存放了device file。每个外部设备在/dev目录下均有一个相对应的项目；

除此之外，/dev目录下还存放有一些虚拟的装置，不对应到任何实体装置，例如空设备的/dev/null，任何写入该设备的请求均会被执行，但被写入的资料会如同进入空设备般消失。

- **/etc:** etc是Etcetera的缩写，该目录下存放着许多系统所需的重要配置和管理文件，有一些是纯文档，有一些是.conf文件，还有一些自成单一目录。
- **/home:** 该目录是用户的主目录，在Linux中，每个用户都有一个一般以自己的账号命名的目录，存放在/home下。例如，当用户jywang登录时，其所在的默认目录为/home/jywang
- **lib:** lib是library的缩写。该目录存放许多与系统激活时所需要的重要的动态链接库文件，包含最重要的GUN C library在内。几乎所有的应用程序都要用到这些动态链接库文件。

- **lost+found:** 这个目录一般是空的。当设备非法关闭时，该目录下会出现一些文件。
- **/media:** 可移动介质的安装点。Linux系统会自动识别一些设备如U盘、光驱等，识别后，Linux会把识别后的设备挂载到这个目录下。
- **/mnt:** mnt是mount point即默认挂载点的缩写。Linux允许用户临时挂载其他的文件系统，进入该目录可以查看挂载的其他文件系统的内容。
- **/opt:** opt是optional的缩写，该目录用于存放用户自定义安装的环境或软件。
- **/proc** proc是Processes的缩写。该目录是一个虚拟的文件系统，即不占用任何硬盘空间，该目录下的文件均被放置于内存中。/proc会动态的记录系统正在运行的进程、硬件状态、内存使用的多少等信息。
- **/root:** Linux下管理员默认为root用户，该目录是root用户的主目录。
- **/sbin:** s是super的缩写，即该目录存放了超级管理员所使用的一些程序的可执行文件。
- **/srv:** 该目录存放一些服务系统启动之后所需要提取的数据
- **/sys:** 在Linux2.6内核以后，出现了/sys目录。该目录下安装了2.6内核中新出现的一个文件系统sysfs。
- **/tmp:** tep是temporary的缩写。该目录存放系统启动时产生的临时文件。有时某些应用程序执行中产生的临时文件也会暂放至此目录。
- **/usr:** usr是unix shared resources的缩写。用户的很多应用程序和文件都放在这个目录。其子目录/usr/bin中存放一些应用程序所需的动态链接库。
- **/var:** 该目录中存放着被系统修改过的数据。我们习惯将被经常修改的文件放在这个目录下，如各种日志文件等。

1.2Linux常用目录命令:

这里仍然以我的一台阿里云服务器为例，解释常用的目录命令。

cd命令

cd命令可以进入到目标路径下。我们执行`cd /opt/javaProjects/JavaWebBegin`，即可以进入`/opt/javaProjects/JavaWebBegin`目录下。

```
root@iZ8vb48stipso6rv9jbbqeZ:/# cd /opt/javaProjects/JavaWebBegin/
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin#
```

ls命令

ls命令可以列出目录的内容。类似DOS下的dir命令

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# ls
pom.xml  servlet-01  servlet02  servlet03
```

ls命令还可以增加一些参数，较为常用的-l参数，可以显示所有文件的详细信息。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# ls -l
total 16
-rw-r--r-- 1 root root 2011 Mar 25 16:17 pom.xml
drwxr-xr-x 4 root root 4096 Mar 25 16:17 servlet-01
drwxr-xr-x 3 root root 4096 Mar 25 16:17 servlet02
drwxr-xr-x 4 root root 4096 Mar 25 16:18 servlet03
```

pwd命令

pwd命令用于显示当前工作目录的路径。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# pwd
/opt/javaProjects/JavaWebBegin
```

mkdir命令

即make directory，用于创建一个新的目录。我们在当前工作目录下新建了一个LinuxHomework2目录。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# mkdir LinuxHomework2
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# ls
LinuxHomework2 pom.xml  servlet-01  servlet02  servlet03
```

rmdir命令

与mkdir相对，即remove directory，用于删除一个新的目录。可以看出，刚刚建立的LinuxHomework2目录已经被删除。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# rmdir LinuxHomework2/
root@iZ8vb48stipso6rv9jbbqeZ:/opt/javaProjects/JavaWebBegin# ls
pom.xml  servlet-01  servlet02  servlet03
```

2.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）？

2.1命令帮助方式及其特点

Linux获取命令帮助主要通过三种方式——`help`命令、特定命令的`help`参数以及`man`命令。

`man`命令显示的内容是手册页，包含指令帮助、配置文件帮助和编程帮助等信息。一般`man`命令显示的帮助信息要更加丰富，它比命令的`help`选项多了命令的用法示例、命令的描述等内容。

特别的，`man`命令显示的内容来自于磁盘上的`manual`手册页文件，手册共有以下几个章节：

- 1：标准用户命令
- 2：系统调用
- 3：库调用
- 4：特殊文件的访问入口
- 5：文件格式
- 6：游戏
- 7：杂项
- 8：管理命令
- 9：跟内核有关的文件。

2.2举例

可以通过在终端输入`pwd --help`或`help pwd`，获得`pwd`命令的帮助方式。

```
root@iZ8vb48stipso6rv9jbbqeZ:~# help pwd
pwd: pwd [-LP]
    Print the name of the current working directory.

Options:
  -L      print the value of $PWD if it names the current working
          directory
  -P      print the physical directory, without any symbolic links

By default, `pwd' behaves as if `-L' were specified.

Exit Status:
Returns 0 unless an invalid option is given or the current directory
cannot be read.
root@iZ8vb48stipso6rv9jbbqeZ:~#
```

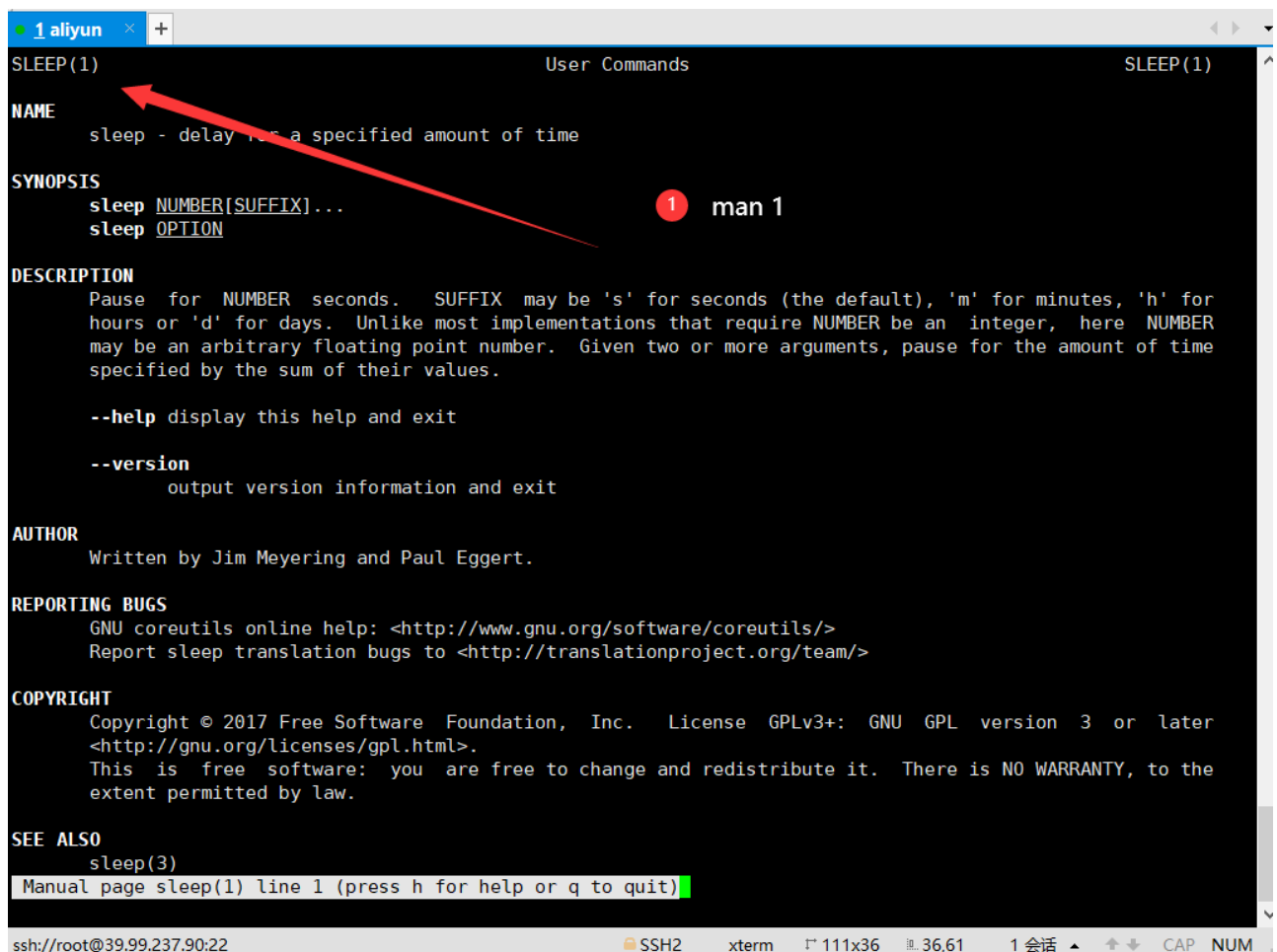
```
root@iZ8vb48stipso6rv9jbbqeZ:~# pwd --help
pwd: pwd [-LP]
    Print the name of the current working directory.

Options:
  -L      print the value of $PWD if it names the current working
          directory
  -P      print the physical directory, without any symbolic links

By default, `pwd' behaves as if `-L' were specified.

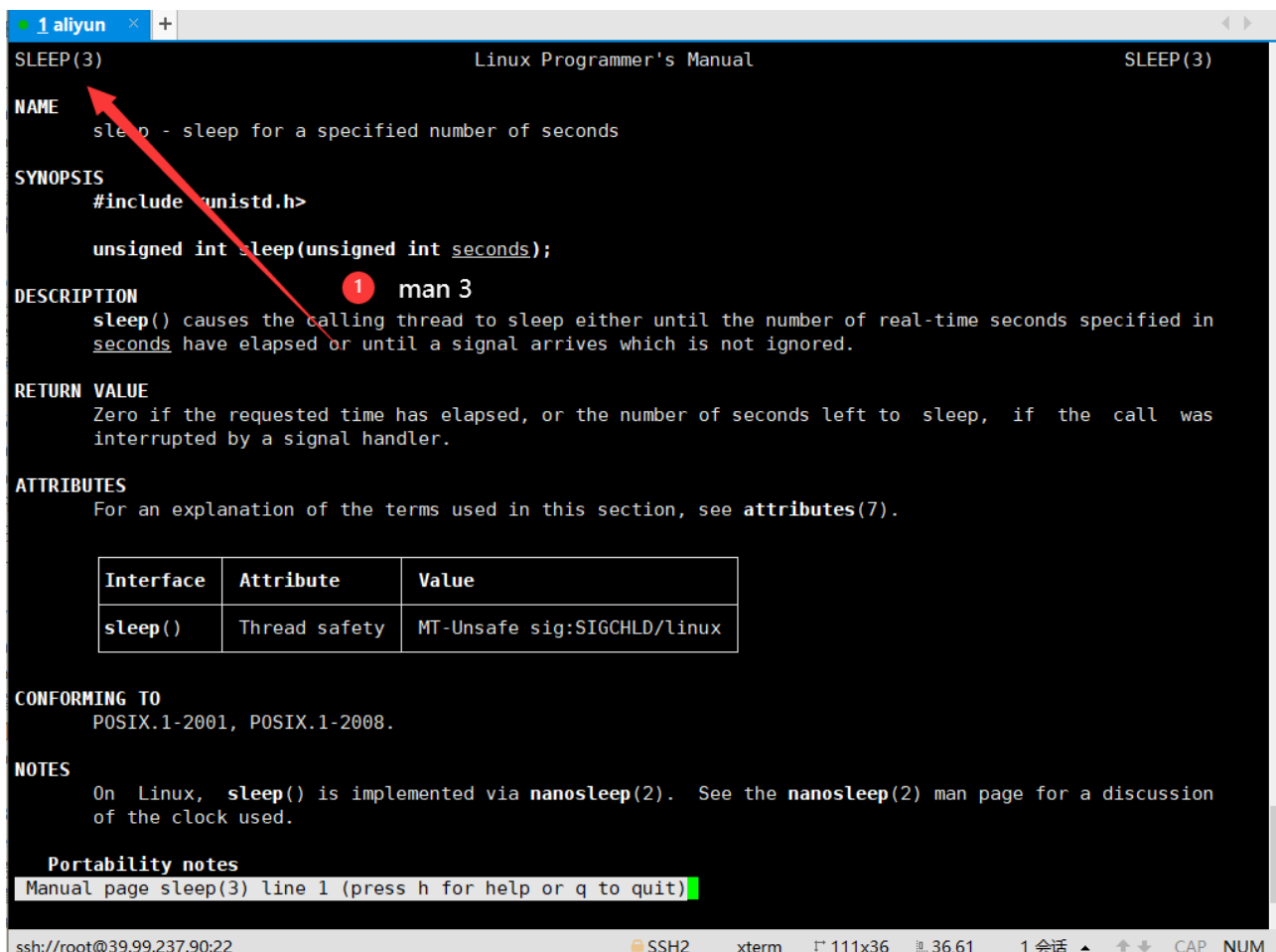
Exit Status:
Returns 0 unless an invalid option is given or the current directory
cannot be read.
root@iZ8vb48stipso6rv9jbbqeZ:~#
```

还可以通过man 1 sleep与man 3 sleep分别查看用户命令sleep与C库中的sleep命令。



```
1 aliyun xterm 111x36 36,61 1 会话 CAP NUM
SLEEP(1) User Commands SLEEP(1)
NAME
    sleep - delay for a specified amount of time
SYNOPSIS
    sleep NUMBER[SUFFIX]...
    sleep OPTION
DESCRIPTION
    Pause for NUMBER seconds. SUFFIX may be 's' for seconds (the default), 'm' for minutes, 'h' for
    hours or 'd' for days. Unlike most implementations that require NUMBER be an integer, here
    NUMBER may be an arbitrary floating point number. Given two or more arguments, pause for the
    amount of time specified by the sum of their values.

    --help display this help and exit
    --version
        output version information and exit
AUTHOR
    Written by Jim Meyering and Paul Eggert.
REPORTING BUGS
    GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
    Report sleep translation bugs to <http://translationproject.org/team/>
COPYRIGHT
    Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later
    <http://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the
    extent permitted by law.
SEE ALSO
    sleep(3)
Manual page sleep(1) line 1 (press h for help or q to quit)
ssh://root@39.99.237.90:22
```



```
SLEEP(3) Linux Programmer's Manual SLEEP(3)
NAME
sleep - sleep for a specified number of seconds
SYNOPSIS
#include <unistd.h>

unsigned int sleep(unsigned int seconds);
DESCRIPTION
1 man 3
sleep() causes the calling thread to sleep either until the number of real-time seconds specified in
seconds have elapsed or until a signal arrives which is not ignored.
RETURN VALUE
Zero if the requested time has elapsed, or the number of seconds left to sleep, if the call was
interrupted by a signal handler.
ATTRIBUTES
For an explanation of the terms used in this section, see attributes(7).



| Interface | Attribute     | Value                       |
|-----------|---------------|-----------------------------|
| sleep()   | Thread safety | MT-Unsafe sig:SIGCHLD/linux |


CONFORMING TO
POSIX.1-2001, POSIX.1-2008.
NOTES
On Linux, sleep() is implemented via nanosleep(2). See the nanosleep(2) man page for a discussion
of the clock used.

Portability notes
Manual page sleep(3) line 1 (press h for help or q to quit)
```

3.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）

Linux的文件类型有7种。当使用ls -l命令显示某一文件的详细信息时，根据该文件第一个属性的不同取值，将Linux的文件划分为7种类型：

- 普通文件'-'：普通文件就是用户通常访问的文件。或者说，不属于其余六种文件类型的文件是普通文件。
- 目录文件'd'：目录文件就是一个目录，相当于windows下的文件夹。
- 链接文件'l'：符号链接文件与windows下的快捷方式类似，分为符号链接与硬链接。
- 块设备文件'b'：用于存储数据以供系统存取的接口设备。
- 字符设备文件'c'：即串行端口的接口设备，如鼠标键盘等。
- 套接字文件's'：套接字文件用于网络数据连接。

- 管道文件'p': 管道文件主要用来解决多个程序同时访问一个文件所造成的错误, 以及进程之间的通信问题。

Linux系统中, 一个文件的基本属性有三种: 读、写、执行。不同的用户对于文件也有不同的读写和执行权限。

- 读权限: 表示具有读取目录结构的权限, 可以查看和阅读文件。
- 写权限: 可以新建、删除、重命名、移动目录后文件。但写权限受父目录权限控制。
- 执行权限: 文件必须拥有执行权限才可以运行。比如二进制文件和脚本文件。目录文件要有执行权限才可以进入。

修改文件属性或是权限的常用命令如下:

- **chmod**: 改变文件或目录的访问权限。如, 通过`chmod u-w cat.txt`命令, 将`cat.txt`文件的写权限取消。
- **chgrp**: 改变文件或目录的所属组。如, 可以通过`chgrp -R root file.txt`命令, 将`file.txt`的所属组设置为`root`。
- **chown**: 将文件或目录的所有者改变为指定用户。如通过`chown jywang cat.txt`命令, 将`cat.txt`的所有者更改为用户`jywang`。

4.什么是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)

4.1CHS与LBA是两种磁盘寻址方式。

- **CHS**是一种比较古老的磁盘寻址方式, 即Cylinder/Head/Sector参数。其中, 磁头数(Heads)表示硬盘总共有几个磁头, 也就是有几面盘片, 最大为255(用8个二进制位存储); 柱面数(Cylinders)表示硬盘每一面盘片上有几条磁道, 最大为1023(用10个二进制位存储); 扇区数(Sectors)表示每一条磁道上有几个扇区, 最大为63(用6个二进制位存储)。总共24位二进制数存储。

在磁盘容量还不是很大的时候, CHS寻址方式适用。但由于其最大寻址空间为8G左右, 随着技术的发展, 已经无法用于更大容量磁盘的寻址了。

- **LBA**是人们为了解决上述问题，提出的一种新的磁盘寻址方式。**LBA**编址方式将**CHS**这种三维寻址方式转变为一维的线性寻址，它把硬盘所有的物理扇区的**C/H/S**编号通过一定的规则转变为一线性的编号，系统效率得到大大提高，避免了烦琐的磁头/柱面/扇区的寻址方式。在访问硬盘时，由硬盘控制器再这种逻辑地址转换为实际硬盘的**CHS**格式物理地址。

LBA寻址方式的最大寻址空间是**128PB**。

4.2 MBR与GPT则是两种磁盘分区方式。

- **MBR**: **Master Boot Record**，即主分区引导记录。它为每个扇区/区块都被分配了一个逻辑块地址，即**LBA**，而引导扇区则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**就保存在主引导扇区中。另外，这个扇区里还包含了硬盘分区表**DPT**（**Disk Partition Table**），和结束标志字（**Magic number**）。扇区总计**512**字节，**MBR**占**446**字节（**0000H - 01BDH**），**DPT**占据**64**个字节（**01BEH - 01FDH**），最后的**magic number**占**2**字节（**01FEH - 01FFH**）。
- **GPT**: **GPT**全称为**Globally Unique Identifier Partition Table**，也叫做**GUID**分区表，它是**UEFI**规范的一部分。由于硬盘容量的急速增长，**MBR**的**2.2T**容量难以满足要求，而**UEFI BIOS**的推广也为**GPT**的实现打下了坚实的技术基础，**GPT**应运而生，它和**MBR**分区的区别就是引导方式和分区结构的不同；支持硬盘分区的容量不同，**MBR**分区结构最高支持到**2T**硬盘，**2T**及以上分区就需要**GPT**分区结构；**MBR**分区直接存储数据和引导记录，**GPT**分区的数据和引导记录是分开存储的，并且理论上支持无限量的分区；**GPT**在**win10**系统中可以支持**128**个分区，最高可以管理**18EB**的磁盘。

4.3 常用磁盘管理命令

- **df**命令：**df**命令会收集和整理当前已经挂载的全部文件系统的一些重要统计数据。

这里仍然以自己的阿里云服务器为例，在服务器的终端输入命令**df**，得到如下信息：

```
root@iZ8vb48stipso6rv9jbbqeZ:~# df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             996652         0     996652   0% /dev
tmpfs            204060       3004     201056   2% /run
/dev/vda1       41152812 10159200    29089956  26% /
tmpfs            1020300         8     1020292   1% /dev/shm
tmpfs             5120          0         5120   0% /run/lock
tmpfs            1020300         0     1020300   0% /sys/fs/cgroup
tmpfs            204060         0     204060   0% /run/user/0
root@iZ8vb48stipso6rv9jbbqeZ:~#
```

输出信息共有六列，分别是文件系统、1K-块、已用、可用、已用%以及挂载点。

上面的列表显示信息为例，udev被挂载到目录/dev下，其容量为1GB，其中0GB已用，占总容量的0%，剩余空间为1GB。

我们可以通过df -t参数来指定特定的文件系统，如执行命令df -t ext4，则会显示所有已挂载的ext4文件系统。

```
root@iZ8vb48stipso6rv9jbbqeZ:~# df -t ext4
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/vda1        41152812 10159256   29089900   26% /
root@iZ8vb48stipso6rv9jbbqeZ:~#
```

- **du命令**：在Ubuntu系统中，可以使用du命令查看文件和目录大小。

比如，我们使用命令du -sh /opt/tomcat9，即可查看tomcat9服务器所占用的磁盘大小。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/tomcat9# du -sh /opt/tomcat9/
45M      /opt/tomcat9/
```

我们可以通过-a参数，来详细的显示目标文件夹中每一个文件所占用的磁盘大小。输入命令du -a /opt/javaProjects/JavaWebBegin/servlet03/src，从而查看在《系统软件开发》课程中，完成的一个基于flex&bison&servlet的javaweb项目中java源文件所占用的磁盘情况。

```
root@iZ8vb48stipso6rv9jbbqeZ:/opt/tomcat9# du /opt/javaProjects/JavaWebBegin/servlet03/src/
4      /opt/javaProjects/JavaWebBegin/servlet03/src/test/java
8      /opt/javaProjects/JavaWebBegin/servlet03/src/test
148    /opt/javaProjects/JavaWebBegin/servlet03/src/main/java/com/chen/calculator
20     /opt/javaProjects/JavaWebBegin/servlet03/src/main/java/com/chen/web
172    /opt/javaProjects/JavaWebBegin/servlet03/src/main/java/com/chen
176    /opt/javaProjects/JavaWebBegin/servlet03/src/main/java/com
180    /opt/javaProjects/JavaWebBegin/servlet03/src/main/java
4      /opt/javaProjects/JavaWebBegin/servlet03/src/main/resources
8      /opt/javaProjects/JavaWebBegin/servlet03/src/main/webapp/WEB-INF
236    /opt/javaProjects/JavaWebBegin/servlet03/src/main/webapp
424    /opt/javaProjects/JavaWebBegin/servlet03/src/main
436    /opt/javaProjects/JavaWebBegin/servlet03/src/
```

- **其他命令**：除了df、du命令外，我们可以使用fdisk命令对磁盘分区；使用mkfs命令对分区格式化；使用mount命令挂载磁盘分区；使用umount命令卸载磁盘分区等等。

5.什么是ext4文件系统？linux的常用文件系统管理命令有哪些（举例给出）

5.1ext4文件系统介绍

Linux最早的文件系统是Minix，但是专门为Linux设计的文件系统——拓展文件系统第二版（EXT2）被设计出来并添加到Linux中，这对Linux产生了巨大的影响。EXT2文件系统功能强大，易扩充，性能上也进行了全面优化，从而成为了所有Linux发行版的标准文件系统类型。但ext2 文件系统与 20 世纪 90 年代的大多数文件系统一样，如果在将数据写入到磁盘

的时候，系统发生崩溃或断电，则容易发生灾难性的数据损坏。随着时间的推移，由于碎片（单个文件存储在多个位置，物理上其分散在旋转的磁盘上），它们也遭受了严重的性能损失。

而后诞生的ext3解决了这个问题。ext3 和 20 世纪 90 年代后期的其它文件系统，如微软的 NTFS，使用日志来解决这个问题。日志是磁盘上的一种特殊的分配区域，其写入被存储在事务中；如果该事务完成磁盘写入，则日志中的数据将提交给文件系统自身。如果系统在该操作提交前崩溃，则重新启动的系统识别其为未完成的事务而将其进行回滚，就像从未发生过一样。这意味着正在处理的文件可能依然会丢失，但文件系统本身保持一致，且其它所有数据都是安全的。

ext4 在功能上与 ext3 在功能上非常相似，但支持大文件系统，提高了对碎片的抵抗力，有更高的性能以及更好的时间戳。ext4和之前的ext3等文件系统有着一些非常明显的差别：

- 向后兼容
- 大文件系统：从ext3的32位寻址变为了ext4的48位寻址
- 分配方式改进：ext4在将存储块写入磁盘之前的存储块的分配方式进行了大量改进，还可以显著提高读写性能。
- 无限制的子目录：ext3仅限于32000个子目录，而ext4允许无限数量的子目录。
- 日志校验：ext3没有对日志进行校验，这给处于内核直接控制之外的磁盘或自带缓存的控制器设备带来了很多问题。ext4系统对日志进行校验和允许文件系统崩溃后第一次挂载时意识到其某些条目是无效或无序的。因此，这避免了回滚部分条目或无序日志条目的错误，并进一步损坏的文件系统——即使部分存储设备假做或不遵守写入障碍。
- 改进的时间戳：ext3 提供粒度为一秒的时间戳。虽然足以满足大多数用途，但任务关键型应用程序经常需要更严格的时间控制。ext4 通过提供纳秒级的时间戳，使其可用于那些企业、科学以及任务关键型的应用程序。

5.2 文件系统管理命令

这一部分与文件有关的命令是Linux中最常用的命令，较为简单。总结如下：

- 显示文件内容命令：cat、more、echo
- 显示目录内容以及更改目录命令：ls、pwd、cd
- 建立、删除文件命令：touch、rm
- 建立、删除目录命令：mkdir、rmdir
- 复制、移动命令：cp、mv
- 压缩备份命令：tar、gzip、gunzip
- 权限管理命令：chmod、chown、chgrp
- 文件搜索命令：whereis、find、locate、which

6.什么是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）

6.1挂载及其作用

Linux系统中所有的设备都被当作文件，所有的设备都被映射为一个特殊文件，这个文件被称作“设备文件”。对于上层应用程序而言，所有对这个设备的操作都是通过读写这个文件实现的。通过文件来操作硬件，是Linux的一大特色。Linux把所有的设备文件都放在/dev目录下：

```
root@iZ8vb48stipso6rv9jbbqeZ:~# cd /dev
root@iZ8vb48stipso6rv9jbbqeZ:/dev# ls
autofs      initctl      network_throughput  tty11  tty30  tty5       ttyS1  ttyS29  vcsa
block       input        null                tty12  tty31  tty50      ttyS10 ttyS3   vcsa1
btrfs-control kmsg         port                tty13  tty32  tty51      ttyS11 ttyS30  vcsa2
bus         lightnvm     ppp                 tty14  tty33  tty52      ttyS12 ttyS31  vcsa3
char        log          psaux               tty15  tty34  tty53      ttyS13 ttyS4   vcsa4
console     loop0        ptmx                tty16  tty35  tty54      ttyS14 ttyS5   vcsa5
core        loop1        pts                 tty17  tty36  tty55      ttyS15 ttyS6   vcsa6
cpu_dma_latency loop2        random              tty18  tty37  tty56      ttyS16 ttyS7   vda
cuse        loop3        rfkill              tty19  tty38  tty57      ttyS17 ttyS8   vda1
disk        loop4        rtc                 tty2   tty39  tty58      ttyS18 ttyS9   vfio
dri          loop5        rtc0                tty20  tty4   tty59      ttyS19 uhid    vga_arbiter
ecryptfs    loop6        shm                 tty21  tty40  tty6        ttyS2  uinput  vhci
fb0          loop7        snapshot            tty22  tty41  tty60      ttyS20 urandom vhost-net
fd           loop-control snd                 tty23  tty42  tty61      ttyS21 userio  vhost-vsock
full         mapper       stderr              tty24  tty43  tty62      ttyS22 vcs     virtio-ports
fuse         mcelog       stdin               tty25  tty44  tty63      ttyS23 vcs1    vportlp1
hidraw0      mem          stdout              tty26  tty45  tty7        ttyS24 vcs2    zero
hpet         memory_bandwidth tty                  tty27  tty46  tty8        ttyS25 vcs3
hugepages    mqueue       tty0                tty28  tty47  tty9        ttyS26 vcs4
hwrng        net          tty1                tty29  tty48  ttyprintk   ttyS27 vcs5
i2c-0        network_latency tty10               tty3   tty49  tty50      ttyS28 vcs6
```

这些文件中大部分是块设备文件和字符设备文件。用户不能直接通过设备文件访问存储设备，所有的存储设备（包括硬盘、光盘等）在使用前必须首先被挂载到一个目录下，然后就可以像操作目录一样使用这个存储设备了。

换句话说，Linux 系统中“一切皆文件”，所有文件都放置在以根目录为树根的树形目录结构中。在 Linux 看来，任何硬件设备也都是文件，它们各有自己的一套文件系统（文件目录结构）。因此产生的问题是，当在 Linux 系统中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，硬件设备才能为我们所用。合二为一的过程称为“挂载”。

即根文件系统之外的其他文件系统要想能够被访问，都必须通过“关联”至根文件系统上的某个目录来实现，此关联操作即为“挂载”，此目录即为挂载点；

或者说，挂载，指的就是将设备文件中的顶级目录连接到 Linux 根目录下的某一目录（最好是空目录），访问此目录就等同于访问设备文件。

6.2常用命令：

mount命令：mount [选项] 设备名 挂载点

其中，常用的选项有：

- -a 加载文件/etc/fstab中设置的所有设备。
- -f 不实际加载设备。
- -F 需要-a参数同时使用
- -h 显示在线帮助信息
- -n 不将加载信息记录在/etc/mtab文件中
- -o 指定加载文件系统时的选项
- -t 指定设备的文件系统类型。

需要注意的是，在挂载分区前，需要新建挂载点，在/mnt目录下新建两个目录，作为分区的挂载点。

umount 命令：umount 设备名。

umount命令用于卸载文件系统。一个可能会用到的参数是-r，这个参数指导umount在无法卸载文件系统的情况下，尝试以只读方式重新载入。

7.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）

7.1存放各类信息的文件位置

用户组的相关信息存放在/etc/group文件中。

用户的基本信息存放在/etc/passwd文件中，而用户的登陆密码信息存放在/etc/shadow文件中。

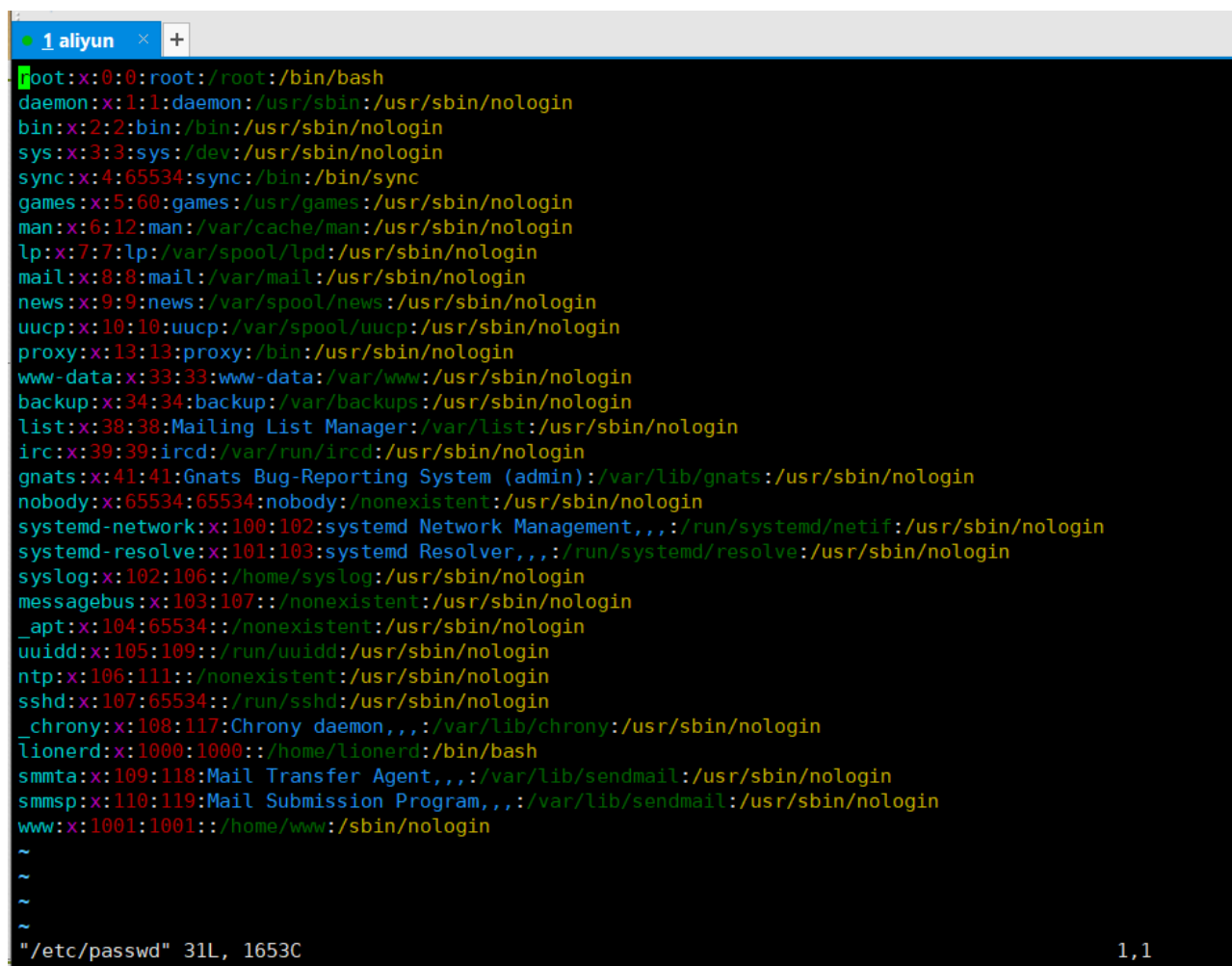
用户的权限信息存放在/etc/sudoers文件中。

7.2 各文件的作用以及信息组织格式

/etc/passwd文件

当我们登录Linux主机时，输入的是我们的账号。但Linux并不认识账号，仅认识ID。ID与账号的对应关系就在/etc/passwd中。该文件每一行都代表一个账号，每一行（即每个账号）由七个字段组成，字段间使用冒号分割。

我们使用vim /etc/passwd来查看文件内容（使用vim而非cat的原因在于，vim显示的内容区分度较高...）：



```
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
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/:home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/:nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/:nonexistent:/usr/sbin/nologin
uidd:x:105:109:/:run/uidd:/usr/sbin/nologin
ntp:x:106:111:/:nonexistent:/usr/sbin/nologin
sshd:x:107:65534:/:run/sshd:/usr/sbin/nologin
chrony:x:108:117:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
lionerd:x:1000:1000:/:home/lionerd:/bin/bash
smmta:x:109:118:Mail Transfer Agent,,,:/var/lib/sendmail:/usr/sbin/nologin
smmsp:x:110:119:Mail Submission Program,,,:/var/lib/sendmail:/usr/sbin/nologin
www:x:1001:1001:/:home/www:/sbin/nologin
~
~
~
~
"/etc/passwd" 31L, 1653C
```

七个字段的含意如下：

- 登录名
- 口令占位符
- 用户ID号（UID）
- 默认组ID号（GID）
- 用户的私人信息：包括全名、办公室、工作电话等
- 用户的主目录
- 登录Shell

/etc/shadow文件

可以看出，/etc/passwd中并没有用户的登陆密码。Linux将用户的登陆密码保存到/etc/shadow文件中。该文件每一行同样代表一个用户，并以冒号分隔每一个字段。使用 `vim /etc/shadow` 查看文件，如下图所示。可以看出，自己的阿里云服务器上，除了自身的root账户外，还有为同学创建的lionerd账户设置了密码。（图中路线标记的两个账户）（考虑到安全性，对登陆密码打了马赛克）。

```
root:$6$3My$dx3fz9EQMC377u12210p125f:sdIthK0:19073:0:99999:7:::
daemon:*:18113:0:99999:7:::
bin:*:18113:0:99999:7:::
sys:*:18113:0:99999:7:::
sync:*:18113:0:99999:7:::
games:*:18113:0:99999:7:::
man:*:18113:0:99999:7:::
lp:*:18113:0:99999:7:::
mail:*:18113:0:99999:7:::
news:*:18113:0:99999:7:::
uucp:*:18113:0:99999:7:::
proxy:*:18113:0:99999:7:::
www-data:*:18113:0:99999:7:::
backup:*:18113:0:99999:7:::
list:*:18113:0:99999:7:::
irc:*:18113:0:99999:7:::
gnats:*:18113:0:99999:7:::
nobody:*:18113:0:99999:7:::
systemd-network:*:18113:0:99999:7:::
systemd-resolve:*:18113:0:99999:7:::
syslog:*:18113:0:99999:7:::
messagebus:*:18113:0:99999:7:::
_apt:*:18113:0:99999:7:::
uidd:*:19031:0:99999:7:::
ntp:*:19031:0:99999:7:::
sshd:*:19031:0:99999:7:::
chrony:*:19031:0:99999:7:::
lionerd:$6$3tC0333IX38GIxjC.I33Vf33mxu25LC/Rl:19073:0:99999:7:::
smmta:*:19073:0:99999:7:::
smmsp:*:19073:0:99999:7:::
www:*:19073:0:99999:7:::
```

各个字段的含意如下；

- 登录名
- 加密后的口令
- 上次修改口令的日期
- 两次修改口令之间的天数（最少）
- 两次修改口令之间的天数（最多）
- 提前多少天提醒用户修改口令
- 在口令过期多少天后禁用该账号
- 账号过期的日期
- 保留，目前为空。

/etc/group文件

/etc/group文件中保存有系统中所有组的名称，以及每个组的成员列表。文件中的每一行表示一个组，由四个冒号分割的字段组成。

使用 `vim /etc/group` 命令查看服务器中的组信息：


```
root@iZ8vb48stipso6rv9jbbqeZ:/dev# vim /etc/group
```

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:
audio:x:29:
dip:x:30:
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
```

4个字段的含意如下：

- 组名
- 组口令占位符
- 组ID（GID）号
- 成员列表，用逗号分开

和passwd文件一样，如果口令字段为一个x的话，就表示还有一个/etc/gshadow文件用于存放组口令。但一般来说，组口令很少会用到，因此不必太在意这个字段。即时这个字段为空，也不需要担心安全问题。

/etc/sudoers

查看/etc/sudoers文件内容：

```
root@iZ8vb48stipso6rv9jbbqeZ:/dev# vim /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
~
~
```

从注释信息中可以发现：这个文件必须使用visudo进行编辑。在该文件中写入规则，可以为用户/用户组设置不同的权限。

8.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）

Linux中存在两种时间——硬件时间和系统时间。（不止Linux，操作系统均存在这两种时间）。

硬件时间是主板上由电池供电的时钟，这个硬件时钟可以在BIOS中进行设置。（也即真实时间 (Real Time Clock, RTC) 或 CMOS 时钟）一般只存储时间值，直至 2016 年后 UEFI 硬件支持对时区和夏令时的存储。

系统时间 (也即软件时间) 是指当前Linux Kernel中的时钟，可以追踪时间值、时区以及可能存在的夏令时。系统时间由 Linux 内核计算，时间值为 UTC+0 下自 1970 年 1 月 1 日午夜至今的秒数——可以在 [time.is 网站](https://time.is) 查询得到。

尤其需要注意的是：操作系统完全启动后，系统时间与硬件时间是独立的。

Linux中的所有命令（包括函数）都是采用的系统时钟设置。在Linux中，用于时钟查看和设置的命令主要有date、hwclock和clock。其中，clock和hwclock用法相近，只用一个就行，只不过clock命令除了支持x86硬件体系外，还支持Alpha硬件体系。

使用date命令查看系统时间：

```
root@iZ8vb48stipso6rv9jbbqeZ:/dev# date
Thu Apr  7 15:30:14 CST 2022
```

使用hwclock --show命令，查看硬件时间：

```
root@iZ8vb48stipso6rv9jbbqeZ:/dev# hwclock --show
2022-04-07 15:30:41.521210+0800
```

使用命令hwclock --set --date="yy/mm/dd hh:mm:ss"来设置硬件时间。

使用命令 hwclock --hctosys同步硬件时钟与系统时钟。

使用命令hwclock --systohc同步系统时钟与硬件时钟。