linux常用命令

一、文件系统结构

操作系统中，用来管理和存储文件信息的软件机构称为文件管理系统，简称文件系统。具体来说，这部分系统就是负责为用户建立、读取、修改和转储文件，控制文件的存取，当用户不再使用时撤销文件等。

- FAT16

  MS-DOS6.X及以下版本使用。每个磁盘的分区最大只能达到2G，并且会浪费很多空间。在FAT16里有簇的概念，就相当于是图书馆里一格一格的书架，每个要存到磁盘的文件都必须配置足够数量的簇，才能存放到磁盘中，每个文件，无论大小，都至少要使用一个簇在保存。

  FAT16各分区与簇大小的关系如下表：

分区大小	**FAT16簇大小**
16MB-127MB	2KB
128MB-255MB	4KB
256MB-511MB	8KB
512MB-1023MB	16KB
1024MB-2047MB	32KB

  思考：如果一个1KB的文件，被保存到了一个1000M的分区，这个文件实际占用的空间是多少?

- FAT32

  Windows95以后的系统都支持。

  FAT32具有一个最大的优点：在一个不超过8GB 的分区中，FAT32分区格式的每个簇容量都固定为4KB，与FAT16相比，可以大大地减少磁盘的浪费，提高磁盘利用率。

突破了FAT16对每一个分区的容量只有2GB的限制，可以将一整个大硬盘定义成一个分区而不必分为几个分区使用，方便了对磁盘的管理。但是，FAT32里，无法存放大于4GB的单个文件，而且容易产生磁盘碎片，性能不佳。

另外，FAT不支持长文件名，只能支持8个字符，而且后缀名最多只支持3个字符。

- NTFS

  WindowsNT系列设计，用来取代FAT系统。每个簇的空间更小，磁盘的利用率更高，并且可以共享资源、文件夹以及对文件设置访问许可权限。

- RAW

  RAW文件系统是一种磁盘未经处理或者未格式化产生的文件系统。一般来说有这几种可能造成正常文件系统变成RAW文件系统：

  - 没有格式化
  - 格式化中途取消
  - 硬盘出现坏道
  - 硬盘出现不可预知的错误

- EXT

  EXT是扩展文件系统，目前最新的版本是5.0.

- HFS(+)

  苹果电脑上的文件系统。

Linux里的文件系统

不同于Windows系统有盘符的概念(C盘、D盘等)，Linux里只有一个盘符，是从 " / " 开始的，只有它是没有上级目录的。如果把整个应用目录系统看成一个树形结构，" / " 表示相当于这个树形结构的根，我们可以将 " / " 称之为根目录。

```
    sudo apt-get install tree
    tree --help  #查看帮助
    tree -L 1  #显示文件目录
```

```
root@ubuntu16 /# tree -L 1
.           #系统根目录，有且只有一个根目录
├── bin     #存放常见的命令
├── boot    #系统启动文件和核心文件都在这个目录
├── cdrom   #光驱
├── dev     #设备文件，其中许多都是在启动时或运行时生成的
```

```
├── etc     #系统配置文件都在这个目录下
├── home    #普通用户的家目录
├── lib     #系统链接库
├── lib64   #64位的链接库
├── lost+found   #系统自动生成的，如果文件系统出错，会在目录下产生文件，记录错误
├── media        #系统自动挂载的光驱、usb等
├── mnt          #mount简写 挂载其他文件系统
├── opt          #可在此安装第三方软件
├── proc         #虚拟目录，它是系统内存的映射，可以通过直接访问这个目录来获取系统信息。
├── root         #超管的目录
├── run          #进程运行数据
├── sbin         #管理员的命令，普通用户无法使用
├── snap         #Ubuntu自己的软件管理工具
├── srv          #服务信息
├── sys          #系统相关
├── tmp          #临时目录，所有用户都具有读写权限
├── usr          #unix software resource  用户的软件安装到这个目录
|    ├── bin     #应用程序的可执行文件
|    ├── sbin    #用户或超管的标准命令
|    ├── local   #管理员安装的应用程序目录
|    └── share   #共享文件目录
└── var          #存放不断扩充的文件。比如数据库文件、日志文件
     ├── log     #日志目录，各种应用的日志
     └── run     # /run的软连接
```

- 以 " . " 开头的文件是隐藏文件。
- " ./ " 表示的是当前目录； “ ../ ” 表示的上级目录
- “ ~ ” 表示的当前用户的家目录

二、目录管理

1. 绝对路径和相对路径

   linux的目录和windows不同，不区分盘符，只有一个根目录，根目录用/表示。

   - 绝对路径：从根目录到当前文件（目录）的路径，比如：/home/python
   - 相对路径：以当前目录为基准，表示上级目录或子目录
     - 用 . 表示当前目录

        - 用..表示上级目录
    - linux目录分隔符只能用正斜线（/）表示
    - 用 ~ 表示用户主目录，用 - 表示来源目录（你从哪个目录切换到当前目录的）

2. 目录切换

```
cd 目录名   #切换目录
 .  #当前目录
 ..  #代表上级目录
 /   #代表根目录
 ~   #用户家目录（宿主目录）   root用户的家目录/root    普通用户的家目录/home/用户名
 cd /etc/yum.repos.d
 cd / #切换到根目录
 cd -  #切换到来源目录
 cd ~ #返回用户的家目录
 cd   #返回用户的家目录

 pwd #显示当前目录的绝对路径名
```

3. 提示信息

```
[root@localhost ~]# cd /
[root@localhost /]$
root代表用户名
localhost 主机名
~ 用户的家目录
/  用户当前所处的目录（/ 根目录）
# 表示超级管理员在操作
$ 普通用户在操作
```

4. 创建目录

```
sudo mkdir  目录名
sudo mkdir -p  目录名    #递归创建目录
sudo mkdir -p  h1802/1/2

python@ubuntu:~/tmp/2$ sudo mkdir -p 3/{4,5}/{6,7,8}
python@ubuntu:~/tmp/2$ tree -L 3
.
└── 3
```

```
    ├── 4
    │   ├── 6
    │   ├── 7
    │   └── 8
    └── 5
        ├── 6
        ├── 7
        └── 8
```

5. 删除目录 rmdir

```
sudo rmdir [option]  目录名     #删除的时候目录必须为空
sudo rmdir -p  目录名    #递归删除空目录
sudo rmdir -p 1/2/3 #1,2,3目录都必须不能有文件
```

三、常见命令

1. ls

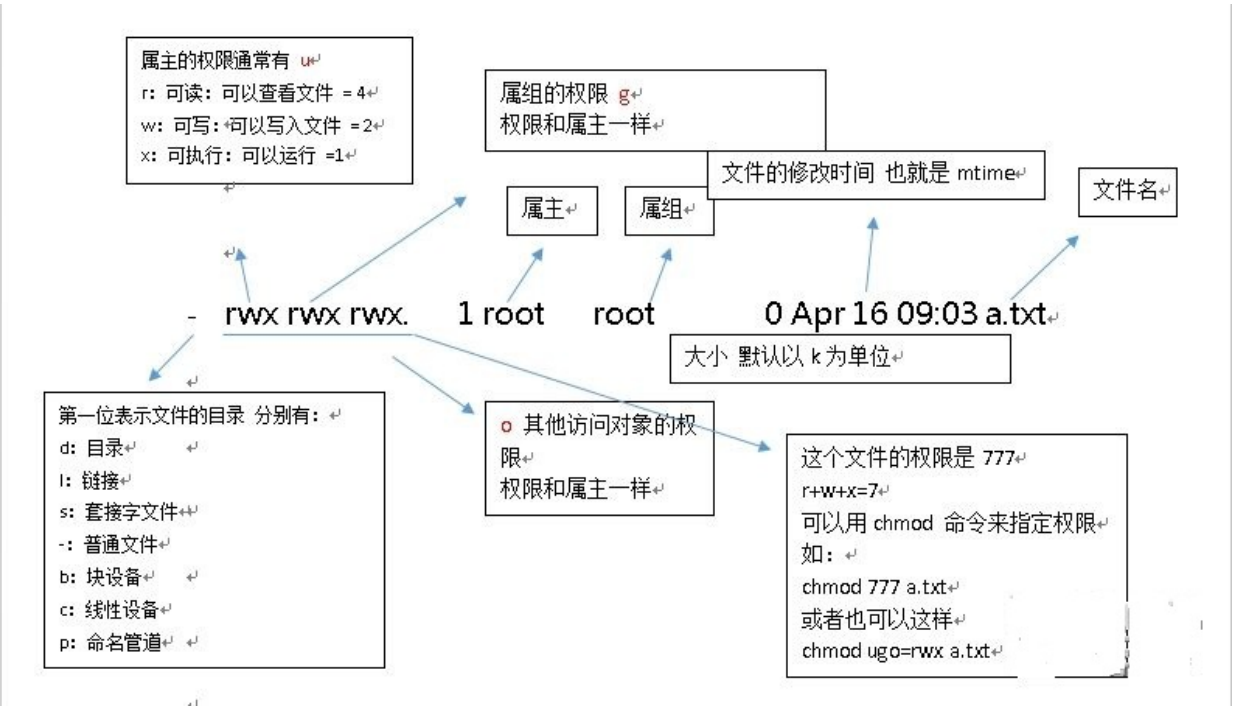
   列出目录下的文件或子目录

```
ls [参数]    #中括号表示可选
ls  -l    #以列表方式显示文件的详细信息
ls  -a    #显示隐藏文件，隐藏文件的文件名以.开头
ls  -A    #显示除.和..以外的任何文件
ls  -i    #查看文件的节点号
ls  --color #以不同颜色显示文件、目录、可执行文件
ls --help  #查找帮助

ls  t*  # 查找以t开头的文件
```

ls --help #查看命令参数

```
   类型及权限|连接数 | 用户   | 用户组 | 大小 |月 | 日| 年/时间 | 名称
   drwxr-xr-x  2     python  python   4096 3月  28  11:20
Templates
   drwxr-xr-x  2     python  python   4096 3月  28  11:20     Videos
```



第一部分：表示文件类型

符号	类型
-	普通文件。包括纯文本文件(ASCII)；二进制文件(binary)；数据格式的文件(data);各种压缩文件等。
d	目录文件。
l	链接文件。类似于Windows里的快捷方式。
c	字符设备文件。即串行端口的接口设备，例如键盘、鼠标等等。
b	块设备文件。就是存储数据以供系统存取的接口设备，简单而言就是硬盘
s	套接字文件。这类文件通常用在网络数据连接，最常在 /var/run目录中看到这种文件类型。
p	管道文件。它主要的目的是，解决多个程序同时存取一个文件所造成的错误。

第2部分，2-10列代表文件的权限：rwxr-xr-x. 第3部分：数字代表文件的硬链接数 第四部分：root代表文件的所有者 第5部分：root表示文件属于哪个用户组 第6部分：数字的表示文件大小，以字节为单位 第7部分：时间，表示文件的上次修改时间 第8部分：文件名

```
#文件权限
  drwxr-xr-x.  2 root root 4096 Nov 20 07:36 tmp
  r: read  可读
  w: write 可写
  x: excute  可执行
  -: 表示无权限

  权限（ugo）：
      2-4位   ower  文件的所有者
      5-7位   group: 用户组
      8-10位  other: 其他用户
```

2. ll

```
以列表方式显示，其实是ls -la的别名，这个配置在~/.bashrc中
白色代表普通文件
绿色代表可执行文件
蓝色代表目录
```

3. man命令

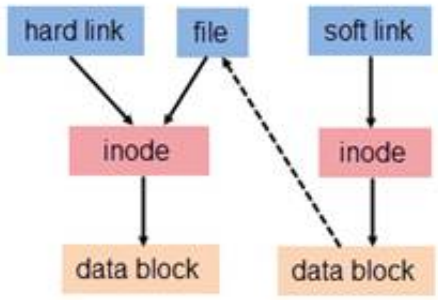
```
#命令的帮助文档
sudo apt  install man
#用法:
   man  命令名
常用的快捷键
空格 f      下翻页
b           上翻页
shift + g   到文件末尾
g          文件开头
q          退出
上下箭头    前翻和后翻
回车键      后翻
home       回到开始
```

4. history命令

   查看你敲过的命令

5. 硬链接和软连接

文件都有文件名与数据，这在 Linux 上被分成两个部分：用户数据 (user data) 与元数据 (metadata)。用户数据，即文件数据块 (data block)，数据块是记录文件真实内容的地方；而元数据则是文件的附加属性，如文件大小、创建时间、所有者等信息。在 Linux 中，元数据中的 inode 号（inode 是文件元数据的一部分但其并不包含文件名，inode 号即索引节点号）才是文件的唯一标识而非文件名。文件名仅是为了方便人们的记忆和使用，系统或程序通过 inode 号寻找正确的文件数据块。



为解决文件的共享使用，Linux 系统引入了两种链接：硬链接 (hard link) 与软链接（又称符号链接，即 soft link 或 symbolic link）。链接为 Linux 系统解决了文件的共享使用，还带来了隐藏文件路径、增加权限安全及节省存储等好处。

- 一个 inode 号对应多个文件名，则称这些文件名为**硬链接**

```
link 源文件名 新文件名
ln 源文件名 新文件名

stat 文件名 #查看文件信息
ls -i  #查看目录下文件的硬链接数
python@ubuntu:/csl$ ls -la
total 4
913923 lrwxrwxrwx. 1 root root    5 Mar 18 16:20 2.txt -> 1.txt
913926 -rw-r--r--. 2 root root    0 Mar 18 19:58 31.txt
913926 -rw-r--r--. 2 root root    0 Mar 18 19:58 32.txt
913925 -rwxr-xr-x. 1 root root    0 Mar 18 17:29 5.txt
913924 drwxr-xr-x. 2 root root 4096 Mar 18 17:29 test

# 硬链接的特点：
   1.只能对已存在的文件进行创建；
   2.不能对交叉文件系统进行硬链接的创建；
   3.不能对目录进行创建，只可对文件创建；
   4.删除一个硬链接文件并不影响其他有相同inode 号的文件。
```

- 软连接

软链接与硬链接不同，若文件用户数据块中存放的内容是另一文件的路径名，则该文件就是软连接。软链接就是一个普通文件，只是数据块内容有点特殊。软链接类似快捷方式

```
ln -s 源文件 目标文件。

软连接的特点：
    软链接有自己的文件属性及权限等；
    可对不存在的文件或目录创建软链接；
    软链接可对文件或目录创建；
    软链接可交叉文件系统；
    创建软链接时，链接计数 inode 不会增加；
    删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软连接被称为死链接
```

6. 常见快捷键

快捷键	说明
ctrl + c	终止正在正在执行的命令
ctrl + a	回到命令开始
ctrl + e	回到命名结尾
ctrl + u	清空命令行
ctrl + l 或者clear	清屏
tab	命令、文件名、目录名自动补齐

7. 常见命令错误

- 命令敲错了 -bash: kkkd: command not found
- 多个空格
- 这个命令还没有安装 command not found

四、文件操作

1. 文件创建

```
sudo touch  文件名 [文件名2] [文件名3]....   #创建多个空文件,如果文件存在，自动忽略，不会覆盖
echo 'hello world'  > 1.txt   #可以将显示内容输出到文件，但会覆盖原来的内容，文件不存在则创建
echo '世界，你好'    >> 1.txt  #将显示内容追加到文件末尾，文件不存在则创建
#输出重定向符号：  > 覆盖   >> 追加
```

输出重定向命令：> 将命令执行的结果输出到文件，如果文件存在，则覆盖其内容

>> 将命令执行的结果输出到文件，如果文件存在，则追加到文件末尾

2. 文件移动

```
sudo mv 源文件    目标文件    #销毁原件
sudo mv  1.txt  ./lpl/  #将1.txt移动到字目录lpl下，文件名不变
sudo mv ../3.txt ./31.txt   #移动到当前目录下，改名为31.txt
sudo mv  1.txt  2.txt   #如果在同一个目录就是文件重命名  将1.txt重命名为2.txt
rename 's/原文件名中需要替换的部分/替换后的部分/' 源文件名
rename  's/tx/txt/' 2.tx  #将2.tx替换为2.txt
```

3. 文件拷贝

```
sudo cp 源文件    目标文件
sudo cp -r  源目录    目标目录    #递归拷贝目录
sudo cp -r 4/8  5/7/
root@ubuntu:/home/python/tmp/2/3# tree -L 4
.
├── 4
│   ├── 1.txt
│   ├── 42.js
│   ├── 8
│   │   ├── 2.php
│   │   ├── 3.py
│   │   └── 5.hello
│   └── 9.php
└── 5
    ├── 6
    │   ├── 2.php
    │   ├── 3.py
```

```
│   └── 5.hello
├── 7
│   └── 8
│       ├── 2.php
│       ├── 3.py
│       └── 5.hello
└── 8
```

4. 文件删除

```
sudo rm  文件名
sudo rm -i 文件名  #删除前逐一确认
sudo rm  -f  文件名    #删除文件不带提示
sudo rm -rf  目录名    #递归删除目录，不管目录是否为空
```

5. 文件查看

```
cat  文件名        #输出文件内容，从前往后输出，
cat  文件1 文件2 >> 文件3  # 将两个文件合并指定文件

tac   文件名       #cat的反写，从后往前输出

head -n N  文件名   #显示文件的前几行，可以指定查看的行数，默认显示10行
head -N  文件名

tail -n N 文件名   #显示文件的最后几行，可以指定查看的行数(N)
tail -N  文件名    #
tail -f  cat 文件名   #实时显示文件内容

watch -d -n 秒数 cat 文件名  #实时显示文件内容 有高亮

sudo vim  文件名
more  文件名        #从前往后查看，可以翻屏 ,不能往前翻  回车一行行查看，空格翻屏（f翻屏）  q退出
less  文件名        #和more类似，可以前翻页，g首页 G尾页，b前翻页，空格和f后翻页，q退出
stat  文件名        #查看文件详细信息
#文件的三个时间：
   atime: accesstime 访问时间
   mtime：modifytime 修改时间
   ctime: changetime 修改状态时间（修改文件元数据）
```

6. 文件查找
    - find

```
# find 用于在系统内搜索指定文件
用法:
   find [路径] [参数] [文件名]
     -name  按文件名查找
     -iname 按文件名查找，不区分大小写
     -mtime +/-n  #-n表示n天以内修改的文件，+n表示修改超过n天的文件
     -user   #按文件属主查找
     -size [+/-]n[c/k/M/G] #查找文件长度为n块，+表示大于，-表示小于；
c是字节
     -perm 权限数值    #按照文件权限进行查找
     -maxdepth N     #查找的目录深度
     -ls             #以列表形式显示
     -type [f/d/l]   # 按类型查看
     -exec 执行shell命令，形式：  -exec command {} \;

find /  -name  "文件名"         #从根目录查找指定文件名的文件,如果不
指定目录，则从当前目录查找
find .  -name  "文件名"         #从当前目录查找指定文件名的文件
find    -name  "文件名"         #从当前目录查找指定文件名的文件

find /csl/sh1702 -name "2.txt"  #查找指定目录先的文件
find /tools -mtime -3           #查找tools目录下修改时间是3天以内的
文件
find /tools -mtime +3           #查找tools目录下修改时间是3天以上的
文件
find -mtime -3 -maxdepth 1  -ls

find /tools  -size 12c         #查找长度为12字节的文件
find /var  -size +10k -size -100k -name '*.log' #在/var目录下，查
找10-100k
find -size +10k -maxdepth 1  -ls

#按文件的所属用户查找
find  -maxdepth 1 -user root

#删除当前目录及其子目录下的所有后缀为txt的文件,注意{}和\中间有空格，最
后有一个;
find . -name '*.txt' -exec rm {} \;
```

```
#只查找当前目录下，权限是700的文件，并以列表形式显示
find -perm 700 -maxdepth 1 -ls
sudo find -maxdepth 2   -name '1.txt' -ls
```

- grep

```
#grep 是global search regular expression(RE) and print out the line的缩写，意思是全面搜索正则表达式并把行打印出来。是一种强大的文本搜索工具，它能使用正则表达式搜索文本内容，并把匹配的行打印出来。
用法:
   grep [options] 'pattern' filename
     -i  不区分大小写
     -c  只显示匹配行的数量
     -r  递归查找子目录
     -l  列出文件内容符合指定的范本样式的文件名称。
     -n  显示行号
     -w  只匹配单词，不是匹配单词一部分
     -E 按正则表达式搜索
     --color  以不同颜色显示匹配的关键字
     --include '*.py'  #仅搜索py文件
     --exclude '*.py'  #不搜索py文件
   # 在1.txt搜索this，要按照完整单词的模式进行匹配，并且显示匹配行
   grep -n -w 'this' 1.txt

   #在当前目录下所有文件中搜索this，并不同颜色显示关键字
   sudo grep  'this' * --color

   显示匹配行数
   sudo grep -c 'this' 1.txt
   搜索指定的1.txt，递归查找子目录
  grep -r -w -l   'this' --include '1.txt'

  # 显示当前目录下所有的文件，不显示目录
  ls -la | grep -E '^-'

  | 管道符，他可以将前一个命令的输出作为后一个命令的输入
```

7. which和whereis
   - which 命令名 #查找命令
   - whereis 文件名 #只能搜索命令、源文件、二进制文件

8. 文件内容统计（wc）

```
用法：
   wc [options] [文件列表]
      -l  统计有多少行
      -w  统计有多少单词
 $ wc -l  /etc/passwd  #统计passwd有多少用户
```

9. awk

   awk就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

```
语法：awk '{pattern + action}' {filenames}
cat /etc/passwd |awk  -F ':'  '{print $1}'   #$1显示第一列  -F 指定分割符为':'
#列数从左向右：1,2,3...
```

10. uniq

    报告或删除文件中重复的行。uniq只比较相邻行，所以它一般会和sort命令进行组合使用。

```
用法：uniq [选项]... [文件]
 -c            //在每行前加上表示相应行目出现次数的前缀编号
 -d            //只输出重复的行
 -D            //只输出重复的行，不过有几行输出几行
 -i            //忽略大小写
```

11. sort

    sort将文件/文本的每一行作为一个单位，相互比较，比较原则是从首字符向后，依次按ASCII码值进行比较，最后将他们按升序输出。

```
sort(选项)(参数)
选项:
   -u忽略相同行
   -k 按指定列排序
   -n 按数值排序
   -t 分割符
   -r 逆序
 #指定passwd文件按第三列 的数值比较，列之间的分隔符为:
 cat /etc/passwd | sort -n -t ':' -k 3

 #查找你最常用的10条命令
 history | awk '{print $2}' | sort | uniq -c | sort -r -n -k 1 |
head -n 10
```

五. 文件权限

1.文件权限修改

选项	说明
身份	
u	所有者(user)
g	所属组(group)
o	其他(other)
a	所有（all）
操作	
+	添加
-	去掉
=	设置
权限	
r	可读
w	可写
x	可执行
-	没有权限

```
    -rwxr-xr-x.  1 root root   24 Nov 21 20:26 1.sh
    -rw-r--r-x.  1 root root    0 Nov 20 07:37 1.txt
    -rw-r--r--.  1 root root    0 Nov 20 07:37 2.txt

    #1.数字表示
    r  4 100   读
    w  2 010   写
    x  1 001   执行
    -  0 000     没有权限
    rwx   7 可读可写可执行
    rw-   6 可读可写
    r-x   5 可读可执行
    r--   4 可读
    rwxrw-r-- 764  文件拥有者可读可写可执行  文件所属的组可读可写  其他人可
 读
```

```
#2.符号表示
u   表示文件的拥有者
g   文件所属的组
o   其他人
a   所有的人   all

u+/-/=   u=rwx   g+x   o-r   a+x == +x
+ 表示增加权限
-   削减权限
=   赋权限
chmod o-x 32.txt    #削减其他用户的可执行权限
chmod a=rwx 32.txt   #给所有人赋可读可写可执行权限
chmod o+x,g+w 32.txt
chmod o=x 32.txt
```

- chmod

```
用法:
    chmod   权限   文件名/目录
    chmod   -R   权限   目录   递归修改目录及其子目录的所有文件的权限

 #数字表示
 chmod 641   1.sh

 #符号表示
 chmod a=rwx 1.sh
 chmod g-w 1.sh
 chmod g-w,o+x 1.sh
 chmod -R   o+w tmp    #递归修改tmp及其子目录中所有文件的权限
```

- umask

  umask用于设置用户创建文件或者目录的默认权限，umask设置的是权限的“补码”，而我们常用chmod设置的是文件权限码。一般在/etc/profile,HOME/.bashprofile或者HOME/.profile中设置umask值。

  默认情况下的umask值是022(可以用umask命令查看），此时你建立的文件默认权限是644(6-0,6-2,6-2)，建立的目录的默认 权限是755(7-0,7-2,7-2)

  对于文件和目录来说， 最大的权限其实都是777，但是执行权限对于文件来说，很可怕，而对目录来说执行权限是个基本权限。所以默认目录的最大权限是777，而文件的默认最大权限就是666。

umask为002的文件默认权限计算方法

	所有者 r	所有者 w	所有者 x	所在组 r	所在组 w	所在组 x	其他 r	其他 w	其他 x
所有权限777	1	1	1	1	1	1	1	1	1
umask掩码002	0	0	0	0	1	0	0	1	0
计算后的值	1	1	1	1	0	1	1	0	1

umask为002的目录默认权限计算方法

	所有者 r	所有者 w	所有者 x	所在组 r	所在组 w	所在组 x	其他 r	其他 w	其他 x
所有权限666	1	1	0	1	1	0	1	1	0
umask掩码002	0	0	0	0	1	0	0	1	0
计算后的值	1	1	0	1	0	0	1	0	0

- 对于目录，直接使用777-umask即可，就得到了最终结果。
- 对于文件，先使用666-umask。

2. chown(change owner) 修改文件的所有者

```
要求：所有者必须在/etc/passwd文件中

    chown 用户名   文件名/目录名
    chown 用户名:组名   文件名/目录名
    chown :组名   文件名/目录名
    chown -R 用户名   文件名/目录名

    chown csl 1.sh
    chown csl:csl 1.sh  #修改用户和所属组
```

3. 修改用户组 chgrp(change group)

```
    组必须存在，组的信息在/etc/group文件里
    chgrp 组名   文件名/目录名
    chgrp -R 组名   目录
    chgrp -R csl tmp  #递归修改tmp及其子目录下文件所属组
```

4. lsattr/chattr 修改和查看文件只读属性

```
lsattr  文件名    查看文件的只读属性,使用ls无法查看
chattr +/-i 文件名   给文件增加或去除只读属性
chattr +/-a 文件名   只能追加数据，不能修改或删除

lsattr 3.py  #3.py有只读属性
----i--------e- 3.py

chattr -i 3.py  #去掉只读属性
chattr +i 3.py  #添加只读属性
```

六. 用户管理

1. 用户和组
    - 一个用户必须有一个主组
    - 一个用户可以有多个组
    - 一个组可以有多个用户
    - 用户账户的信息存放在/etc/passwd文件中；用户的密码存放到/etc/shadow，该文件只有root可以修改；组账户信息存放到/etc/group中

2. useradd 添加一个用户

```
用法:
   useradd [-gud] 用户名
     -g 指定主组名或组id，必须已经存在的组
     -u 指定用户的id
     -m 自动建立用户主目录
     -d 指定用户的家目录
     -s 指定用户登录后使用shell，默认是/bin/bash
   #创建一个用户没有指定组，则默认创建一个和用户名一样的组，作为用户的
主组

sudo useradd -u 1202 -g python -md /test2 -s /usr/sbin/nologin
test2
```

```
所有的用户都在/etc/passwd文件中
luoming:x:501:501::/home/luoming:/bin/bash
用户名  密码 用户id  用户所属组的id  用户的家目录   shell
              uid     gid
#Ubuntu 特别提供了一个adduser 命令以交互模式创建用户，
sudo adduser csl
```

3. 删除用户 userdel

```
userdel -r 用户名  删除用户同时删除家目录（家目录要和用户名一致才能删除）
#如果用户登录了无法删除，应该先切换用户，然后kill -9 用户进程号，然后在删除
```

4. 修改用户信息 usermod

```
usermod [option]  用户名
     -u  用户id
     -g  主组id
     -G  附属组名称
     -a  将用户添加到附属组，必须与-G配合使用
     -d  用户的家目录
     -l  用户登录名
   sudo usermod -u 1001 -g 999 -l lkz  liwenkai
   sudo usermod -a -G csl python  #将用户python添加到附属组csl中
   sudo usermod -l newusername  oldusername #修改用户名
```

5. 修改用户密码

   用法： passwd [-lu] 用户名 -l 锁定账户密码 -u 解锁账户密码 root 可以修改其他用户的密码 普通用户只能修改自己的密码

6. su和sudo

Ubuntu默认禁止使用root账户，在系统安装的时候，创建的第一个用户作为管理员（属于sudo组），其权限要低于root，但比普通用户高，普通用户只能处理自己创建的东西，管理员可以安装软件、修改日期、删除用户等。在Ubuntu中一般看到提示符是$，当执行需要root权限操作的时候需要提升权限，我们可以使用sudo暂时提升用户权限

我们也可以使用su切换用户身份，可以切换到root或管理员，完成工作后再切换回来

```
用法:
   sudo 命令 #需输入用户自己的密码

用法:
   su    账户名  #需要输入目标用户的密码
root切换到普通用户不用输入密码
普通用户切换，必须输入密码
因为Ubuntu默认不提供root密码，不能直接由su切换到root，可以先使用sudo来获取root权限
$ sudo su root  #临时切换到root

#启用root账户
$ sudo password root #根据提示为root输入密码

#设置sudo提升权限的时候不需要输入密码，需要修改/etc/sudoers文件

sudo vi /etc/sudoers
#也可以使用工具:sudo visudo编辑

# Allow members of group sudo to execute any command
#将sudo组添加NOPASSWD:
%sudo   ALL=(ALL:ALL) NOPASSWD:  ALL
#然后按esc
:wq!

#如果新添加的用户不属于sudo组，是不能使用sudo提升权限的，需要将用户添加到sudo组
```

```
#以属于sudo组的用户登录
sudo usermod -a -G sudo  用户名
```

7.其他命令

- id 查看用户的id和组信息
- groups查看用户的组
- whoami 查看当前的用户是谁

七. 组管理

```
#添加一个组
groupadd 组名

   1702:x:1001:
   组名 密码  gid
groupdel  组名  #删除组
groupmod  -n 新组名  旧组名
groups 显示用户的组

#所有的组信息都在/etc/group文件中记录
#用户密码在/etc/shadow
```