

- 一、常用指令

- 1、ls
- 2、pwd
- 3、cd
- 4、mkdir
- 5、touch
- 6、cp
- 7、mv
- 8、rm
- 9、vim
- 10、> >>
- 11、cat
- 12、df
- 13、free
- 14、head
- 15、tail
- 16、less
- 17、wc
- 18、date
- 19、cal
- 20、clear / ctrl + l
- 21、|.
- 22、hostname
- 23、id
- 24、whoami
- 25、ps
- 26、top
- 27、du
- 28、find
- 29、service
- 30、kill / killall
- 31、ifconfig
- 32、reboot
- 33、shutdown
- 34、uptime
- 35、uname
- 36、netstat
- 37、man

- 二、Vim

- 1、打开文件
- 2、进入编辑模式
- 2、退出文件
- 3、光标的移动
- 4、修改操作
- 6、搜索/查找
- 7、替换
- 8、其他

- 三、自有服务

- [1、运行模式](#)
- [2、用户和用户组](#)
- [3、网络设置](#)
- [4、ssh服务](#)
- [5、修改主机名](#)
- [6、chkconfig](#)
- [7、ntp服务](#)
- [8、防火墙](#)
- [9、rpm](#)
- [10、cron](#)
- [四、权限管理](#)
 - [1、chown](#)
 - [2、chgrp](#)
 - [3、sudo](#)
- [五、网络](#)
 - [1、ping](#)
 - [2、netstat](#)
 - [3、traceroute](#)
 - [4、arp](#)
 - [5、tcpdump](#)
- [六、shell](#)

一、常用指令

1、ls

```
# list 列出当前工作目录下的所有文件和文件夹的名称
ls
ls -l [path]      #以详细列表的形式显示
ls -a [path]      #显示所有文件
ls -la [path]     #以详细列表的形式显示所有文件
ls -lh [path]     #以更加易读的方式显示出文件大小
```

2、pwd

```
# print working directory 打印当前工作目录
pwd
```

3、cd

```
# change directory 切换当前工作目录
cd [path]
cd .
cd ..            #返回到父目录
cd              #切换到用户主目录
cd ~            #切换到用户主目录
cd /            #切换到根目录
```

4、mkdir

```
# make directory 创建目录
mkdir [path]
mkdir -p [path] #实现多层一次性创建
```

5、touch

```
# 创建文件
touch [file]
```

6、cp

```
# copy 复制文件或文件夹
cp [src] [dest]
cp -r [src] [dest] #递归处理，将指定目录下的所有文件与子目录一并处理
```

7、mv

```
# move 移动、剪切
move [src] [dest]
```

8、rm

```
# remove 移除、删除
rm [opt] [file]
rm -f [file] #无需提示 force
rm -r [file] #删除文件夹下的所有文件，需要一一确认
rm -rf [file] #删除文件夹下的所有文件，无需提示
```

9、vim

```
# 内置的文本编辑器
vim [file]
```

10、> >>

```
# 输出重定向
[command] > [file] #覆盖原先文件内容
[command] >> [file] #原先文件内容末尾追加
```

11、cat

```
# 查看文件内容
cat [file]

# 对文件进行合并
cat [file2] [file2] > [newfile]
```

12、df

```
# 查看磁盘空间
df
df -h    以更加易读的方式显示存储空间
```

```
tutou_li@lsz:~$ df
文件系统      1k-块    已用    可用  已用% 挂载点
udev          986340      0   986340    0% /dev
tmpfs         201748    3304   198444    2% /run
/dev/sda1     20509264 7388764 12055644   38% /
tmpfs         1008740    6364   1002376    1% /dev/shm
tmpfs          5120      4     5116    1% /run/lock
tmpfs         1008740      0   1008740    0% /sys/fs/cgroup
/dev/loop0     14848    14848      0  100% /snap/gnome-logs/37
/dev/loop2     13312    13312      0  100% /snap/gnome-characters/103
/dev/loop6     2432     2432      0  100% /snap/gnome-calculator/180
/dev/loop4     3840     3840      0  100% /snap/gnome-system-monitor/51
/dev/loop1     144384   144384      0  100% /snap/gnome-3-26-1604/70
/dev/loop3     35584   35584      0  100% /snap/gtk-common-themes/319
/dev/loop5     89088   89088      0  100% /snap/core/4917
tmpfs         201748     16   201732    1% /run/user/121
tmpfs         201748     56   201692    1% /run/user/1000
tmpfs         201748      0   201748    0% /run/user/0
tmpfs         201748      0   201748    0% /run/user/1001
/dev/sr0       57218   57218      0  100% /mnt/cdrom
```

13、free

```
# 查看内存情况
free
free -m 以MB为单位
free -g 以GB为单位
```

```
tutou_li@lsz:~$ free -m
              总计      已用      空闲      共享    缓冲/缓存    可用
内存:         1970      1418        92        21        459      376
交换:          947       465      482
tutou_li@lsz:~$ free -g
              总计      已用      空闲      共享    缓冲/缓存    可用
内存:           1         1         0         0         0         0
交换:           0         0         0
tutou_li@lsz:~$ free
              总计      已用      空闲      共享    缓冲/缓存    可用
内存:    2017480    1452568    94072    22104    470840    385464
交换:    969960     476160    493800
```

14、head

```
# 查看文件的前n行 默认10
head -n [file]
```

15、tail

```
# 查看文件的后n行 默认10
tail -n [file]
tail -f [file]  查看一个文件的动态变化，一般用于查看系统日志
```

16、less

```
# 查看文件，以较少的内容输出，可以使用辅助功能键
less a  #空格键向下翻页 B键向上翻页 上下键移动一行，回车向下移动一行
```

17、wc

```
# 统计文件内容信息
wc -l [file]    #行数
wc -w [file]    #单词数
wc -c [file]    #字节数
```

```
tutou_li@lsz:~$ wc -w test.txt
2 test.txt
tutou_li@lsz:~$ wc -c test.txt
12 test.txt
tutou_li@lsz:~$ wc -l test.txt
1 test.txt
```

18、date

```
# 表示操作时间和日期
date                #2020年 04月 28日 星期二 17:31:25 CST(当地时间)
date +%F            #2020-04-28
date "+%Y-%m-%d"    #2020-04-28
date "+%F %T"       #2020-04-28 17:40:20
date "+%Y-%m-%d %H:%M:%S" #2020-04-28 17:40:20
date -d "-1 day" "+%F %T" #2020-04-28 17:40:20
```

```
tutou_li@lsz:~$ date
2020年 04月 29日 星期三 14:27:02 CST
```

19、cal

```
# 用来操作日历
cal      #查看当前月份的日历(第一天是周日)
cal -3   #查看上个月+本月+下一个月的日历
cal -y   #查看本年的日历
cal -m   #查看当前月份的日历(第一天是周一)
```

```
tutou_li@lsz:~$ cal
      四月 2020
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

tutou_li@lsz:~$ cal -y
          2020
      一月          二月          三月
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           1  1  2  3  4  5  6  7
 5  6  7  8  9 10 11  2  3  4  5  6  7  8  8  9 10 11 12 13 14
12 13 14 15 16 17 18  9 10 11 12 13 14 15 15 16 17 18 19 20 21
19 20 21 22 23 24 25 16 17 18 19 20 21 22 22 23 24 25 26 27 28
26 27 28 29 30 31  23 24 25 26 27 28 29 29 30 31

          四月          五月          六月
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           1  2           1  2  3  4  5  6
 5  6  7  8  9 10 11  3  4  5  6  7  8  9  7  8  9 10 11 12 13
12 13 14 15 16 17 18 10 11 12 13 14 15 16 14 15 16 17 18 19 20
19 20 21 22 23 24 25 17 18 19 20 21 22 23 21 22 23 24 25 26 27
26 27 28 29 30  24 25 26 27 28 29 30 28 29 30
```

20、clear / ctrl + l

```
# 清空终端显示内容
clear
ctrl + l
```

21、|

```
# 管道一般可用于“过滤”，“特殊”，“扩展处理”
ls | grep a      #过滤出所有包含a字符的内容
```

```
tutou_li@lsz:~$ ls |grep a
aa
examples.desktop_
```

22、hostname

```
# 操作服务器主机名
hostname      #输出完整的主机名
hostname -f   #输出当前主机名中的FQDN(Fully Qualified Domain Name全限定域名)
```

```
tutou_li@lsz:~$ hostname
lsz
```

23、id

```
# 查看用户的基本信息(用户id, 用户组id, 附加id...)
id      #显示当前用户信息
id [username]  #显示指定用户信息
```

```
****
验证用户信息      /etc/passwd
验证用户组信息    /etc/group
****
```

```
tutou_li@lsz:~$ id
uid=1000(tutou_li) gid=1000(tutou_li) 组=1000(tutou_li),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
```

24、whoami

显示当前登录的用户名，一般用户shell脚本，一般用户获取当前操作的用户名，用于记录日志
whoami

```
tutou_li@lsz:~$ whoami
tutou_li
```

25、ps

用户查看当前服务器的进程信息
ps
ps -e #列出全部进程
ps -f #显示全部的列
ps -ef | grep [processname] #搜索指定进程

```
tutou_li@lsz:~$ ps
  PID TTY          TIME CMD
 20162 pts/3        00:00:01 bash
 23040 pts/3        00:00:01 ps
```

26、top

查看服务器进程占的资源
top #动态显示
M #将结果按照内存使用从大到小的顺序排列
P #将结果按照CPU使用从大到小的顺序排列

```
tutou_li@lsz:~$ top

top - 15:05:04 up 19:33, 2 users, load average: 1.10, 0.96, 0.56
任务: 286 total, 1 running, 220 sleeping, 2 stopped, 0 zombie
%Cpu(s): 15.6 us, 8.0 sy, 0.0 ni, 75.6 id, 0.7 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2017480 total, 73232 free, 1469512 used, 474736 buff/cache
KiB Swap: 969960 total, 484328 free, 485632 used. 367988 avail Mem

 进程 USER      PR  NI   VIRT   RES   SHR  %CPU  %MEM    TIME+  COMMAND
   734 root        20   0  692356  23600  6720   15.6   1.2   8:26.17 snapd
   292 root        20   0     0     0     0    7.0   0.0   0:55.82 jbd2/sda1+
  1692 tutou_li    20   0  426748  35044 18560    2.0   1.7   2:39.65 Xorg
  1823 tutou_li    20   0 3058228 349516 32600    1.3  17.3   9:50.98 gnome-she+
 15596 tutou_li    20   0  848108  32216 18184    1.0   1.6   0:32.68 gnome-ter+
  1267 gdm        20   0  260804   3524  3088    0.3   0.2   4:04.35 Xwayland
 23041 tutou_li    20   0   51364   4096  3340    0.3   0.2   0:00.77 top
     1 root        20   0  225700   6120  3804    0.0   0.3   0:19.11 systemd
     2 root        20   0     0     0     0    0.0   0.0   0:00.01 kthreadd
     4 root         0 -20     0     0     0    0.0   0.0   0:00.00 kworker/0+
     6 root         0 -20     0     0     0    0.0   0.0   0:00.00 mm_percpu+
     7 root        20   0     0     0     0    0.0   0.0   1:09.37 ksoftirqd+
     8 root        20   0     0     0     0    0.0   0.0   0:05.09 rcu_sched
     9 root        20   0     0     0     0    0.0   0.0   0:00.00 rcu_bh
    10 root        rt    0     0     0     0    0.0   0.0   0:00.00 migration+
```

27、du

查看目录的大小
du
du -s #summary 显示汇总的大小
du -h #以更可读的方式显示大小

```
tutou_li@lsz:~$ du
12      ./local/share/gnome-shell
4       ./local/share/applications
4       ./local/share/ibus-table
4       ./local/share/gnome-mines/scores
8       ./local/share/gnome-mines
76      ./local/share/gvfs-metadata
4       ./local/share/nautilus/scripts
8       ./local/share/nautilus
8       ./local/share/rhythmbox
1132    ./local/share/app-info/xmls
1136    ./local/share/app-info
4       ./local/share/evolution/tasks/trash
8       ./local/share/evolution/tasks
4       ./local/share/evolution/memos/trash
8       ./local/share/evolution/memos
4       ./local/share/evolution/addressbook/system/photos
92      ./local/share/evolution/addressbook/system
4       ./local/share/evolution/addressbook/trash
100     ./local/share/evolution/addressbook
8       ./local/share/evolution/calendar/system
4       ./local/share/evolution/calendar/trash
16      ./local/share/evolution/calendar
4       ./local/share/evolution/mail/trash
8       ./local/share/evolution/mail
144     ./local/share/evolution
```

28、find

```
# 查找文件
find [path] [opt] [value]
find -name [file]    #按照文件名搜索
find -type f        #搜索文件
find -type d        #搜索文件夹
```

29、service

```
# 用于控制一些软件的服务，启动/停止/重启
service [服务名] start/stop/restart
```

30、kill / killall

```
# 杀死进程
kill [PID]
killall [processname]
```

31、ifconfig

```
# 用于操作网卡的相关指令
ifconfig
```



```
tutou_li@lsz:~$ ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::356b:a1e2:3f07:4074 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:4a:89:9e txqueuelen 1000 (以太网)
    RX packets 366675 bytes 506404097 (506.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 184162 bytes 15074823 (15.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 12028 bytes 1201674 (1.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12028 bytes 1201674 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

32、reboot

```
# 重新启动计算机
reboot          #立即重启
reboot -w       #模拟重启，将重启写入日志
```

33、shutdown

```
# 关机
shutdown -h now      #立即关机
shutdown -h [time]   #定时关机
shutdown -c          #取消关机
```

34、uptime

```
# 输出计算机的持续在线时间
uptime
```

```
tutou_li@lsz:~$ uptime
14:38:48 up 19:07, 2 users, load average: 0.03, 0.05, 0.19
```

35、uname

```
# 获取操作系统名称
uname
uname -a      #获取所有信息
```

```
tutou_li@lsz:~$ uname -a
Linux lsz 4.15.0-96-generic #97-Ubuntu SMP Wed Apr 1 03:25:46 UTC 2020 x86_64 x
86_64 x86_64 GNU/Linux
```

36、netstat

```
# 查看网络连接状态
netstat
netstat -t #只列出tcp协议的连接
netstat -n #列出ip地址+端口
netstat -l #过滤出state列中值为Listen的连接
netstat -p #显示出PID值
```

```
tutou_li@lsz:~$ netstat
激活Internet连接 (w/o 服务器)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.115:34610     dashboard.snapcra:https ESTABLISHED
tcp        0      0 localhost:37846         localhost:ssh            ESTABLISHED
tcp        0      0 lsz:43358              ec2-34-213-76-134:https ESTABLISHED
tcp        0      0 localhost:ssh          localhost:37846         ESTABLISHED
活跃的UNIX域套接字 (w/o 服务器)
Proto RefCnt Flags   Type       State      I-Node   路径
unix    2      [ ]     数据报     State      74014    /run/user/1001/systemd/
notify
unix    2      [ ]     数据报     State      47884    /run/user/0/systemd/not
ify
unix    2      [ ]     数据报     State      34839    /run/user/1000/systemd/
notify
unix    2      [ ]     数据报     State      27930    /run/user/121/systemd/n
otify
unix    2      [ ]     数据报     State      121191   /var/lib/samba/private/
msg.sock/18109
unix    2      [ ]     数据报     State      121197   /var/lib/samba/private/
msg.sock/18112
unix    2      [ ]     数据报     State      121198   /var/lib/samba/private/
msg.sock/18111
unix    2      [ ]     数据报     State      121211   /var/lib/samba/private/
msg.sock/18114
unix    2      [ ]     数据报     State      121587   /var/lib/samba/private/
msg.sock/18162
```

37、man

```
# manual 帮助手册
man [command] #查看命令帮助，按q退出
```

```
tutou_li@lsz:~$ man apt
[2]+ 已停止                  man apt
```

二、Vim

1、打开文件

```
# 打开文件
vim [filepath]
# 打开指定的文件，并把光标移动到指定行
vim [+number] [filename]
# 打开指定的文件，并且高亮显示关键词
vim [+/keywords] [filepath]
# 同时打开多个文件
vim [filepath1] [filepath2] [filepath3]...
```

2、进入编辑模式

```
# 在光标所在字符前开始插入
i
# 在光标所在字符后开始插入
a
# 在光标所在行的下面另起一新行插入
O
# 在光标所在行的行首还是插入，行首有空格则在空格之后插入
```

```
I
# 在光标所在行的行尾开始插入
A
# 在光标所在行的上面另起一行插入
O
# 删除光标所在行并开始插入
S
```

2、退出文件

```
# 进入末行模式
:
# 退出末行模式
Esc
# 退出
:q(quit)
# 保存
:w(write)
# 另存为
:w [path]
# 强制退出
:q!
# 保存退出(若无修改,文件修改时间不会被修改)
:x
# 保存退出(文件修改时间会被更改)
:wq
```

3、光标的移动

```
# 移动到行首
shift + 6 或者 ^
# 将光标移动到行尾
shift + 4 或者 $
# 光标移动到首行
gg
# 光标移动到末行
shift + g 或者 G
# 向上翻页
ctrl + b(back) 或者 PgUp
# 向下翻页
ctrl + f(forward) 或者 PgDn
# 移动指定行/列数
[number] + 方向键
# 移动到指定行
:[number]
```

4、修改操作

```
# 复制number行
[number]yy
```

```
# 可视化复制
ctrl + v    --> 方向键移动选中需要复制区块    --> yy
# 粘贴
p(paste)

# 剪切/删除光标所在行（删除后下一行上移）
dd
# 剪切/删除所在行光标之后的内容
shift + d    或者 D
# 剪切/删除光标所在行及下一行
dd 方向下键
# 剪切/删除光标所在行及上一行
dd 方向上键

# 撤销
u(undo)
#恢复
ctrl + r
```

6、搜索/查找

```
# 进入末行模式
/[keywords] 按N/n切换上/查找的内容
# 取消高亮
:nohl(nohighlight)
```

7、替换

```
# 替换光标所在行搜索到的第一个内容
:s/[old]/[new]
# 替换光标所在行搜索到的全部内容
# 替换每一行搜索到的第一个内容
:%s/[old]/[new]
# 替换每一行搜索到的全部内容
:%s/[old]/[new]/g
```

8、其他

```
# 显示行号
:set nu
# 取消显示
:set nonu
# 查看当前打开的所有文件名称 %a表示当前打开的文件 #表示上一个打开的文件
:files
# 切换文件
:open [filename]
# 切换到下一个文件
:bn
# 切换到上一个文件
:bp
```

```
# 代码着色（默认开启）
:syntax on
# 取消代码着色
:syntax off
# 使用计算器
编辑模式 --> ctrl + R --> =
# 文件加密
:X
```

三、自有服务

1、运行模式

```
# 修改运行模式为命令行模式
vim etc/inittab --> 3
```

2、用户和用户组

```
# 存储用户的关键信息
/etc/passwd
# 存储用户组的关键信息
/etc/group
# 存储用户的密码信息
/etc/shadow

# 添加用户，同时创建同名的home目录和用户组
useradd [option] [username]
    -g 指定用户的用户主组，可以是ID或组名
    -G 指定用户的用户附加组，可以是ID或组名
    -u 自定义uid，系统默认从500之后开始分配uid

# 修改用户
usermod [option] [username]
    -g 指定用户的用户主组，可以是ID或组名
    -G 指定用户的用户附加组，可以是ID或组名
    -u 自定义uid，系统默认从500之后开始分配uid
    -l 修改用户名

# 设置密码
passwd [username]

# 删除用户
userdel [username]
    -r 删除用户的同时删除家目录

# 切换用户
su [username]

# 添加用户组
groupadd [option] [groupname]
    -g 选择自己设置一个自定义的用户组ID数字，默认500开始

# 用户组编辑
```

```
groupmod [option] [groupname]
    -g 选择自己设置一个自定义的用户组ID数字
    -n 设置新的用户组名称

# 用户组删除
groupdel [groupname]
```

3、网络设置

```
# 网卡配置路径
/etc/sysconfig/network-scripts

# 各种快捷方式路径
/etc/init.d

# 创建快捷方式
ln -s [originpath] [shortcutpath]

# 重启所有网卡
service network restart
/etc/init.d/network restart

# 关闭某个网卡
ifdown [networkname]

# 开启某个网卡
ifup [networkname]
```

4、ssh服务

```
# 配置文件所在路径
/etc/ssh/ssh_config
/etc/init.d/sshd

# 启动/停止/重启
service sshd start/stop/restart

# 软件
Xshell、SecureCRT、PuTTY、FileZilla、PSCP
```

5、修改主机名

```
# 查看主机名
hostname
hostname -f(FQDN、全限定域名)

# 临时修改主机名
hostname [newname]
# 永久设置主机名（需要重启）
/etc/sysconfig/network
# 修改linux服务器的hosts文件，设置FQDN
/etc/hosts
```

6、chkconfig

```
# 开机启动服务查询(0-6对应开机级别)
chkconfig --list

# 删除服务
chkconfig --del [servicename]

# 添加服务
chkconfig --add [servicenameche]

#设置服务在某个级别下开启或关闭
chkconfig --level 35 [servicenameche] on/off
```

7、ntp服务

```
# 主要是用于对计算机的时间同步管理操作
# 手动同步
ntpdate [ip]

# 启动ntpd服务自动同步
service ntpd start
/etc/init.d/ntpd start
chkconfig --level 35 ntpd on
```

8、防火墙

```
# 开启/关闭/重启防火墙
service iptables start/stop/restart
/etc/init.d/iptables start/stop/restart

# 查看防火墙状态
service iptables status

# 查看规则
iptables -L -n

# 置防火墙
iptables -A INPUT -p tcp --dport 80 -j ACCEPT #允许访问80端口
/etc/init.d/iptables save #将防火墙规则保存到/etc/init.d/iptables
```

9、rpm

```
# 查询所有安装的软件
rpm -qa

# 卸载软件（存在依赖关系时，不能被卸载）
rpm -e [software]

# 强制卸载软件
rpm -e [software] --npdeps

# 安装软件
rpm -ivh [path]

# 查看块状设备的信息
lsblk

# 解挂操作
umount [path]

# 挂载光盘(原始路径均在 /dev/ 下)
mount [device origin path] [mount path]
```

10、cron

```
# 计划任务
crontab [option]
-l #list, 列出指定用户的计划任务列表
-e #edit, 编辑指定用户的计划任务列表
#分(0-59) 时(0-23) 日(1-31) 月(1-12) 周(0-6) 命令
#*(所有数字) -(做区间) /(每/个) ,(多个取值)
0 0 * * * * reboot
* * * * * date+"%F %T" >> /root/rt.txt
-u #user, 指定的用户名, 不指定则表示当前用户
-r #remove, 删除指定用户的计划任务列表

# 黑名单配置文件
/etc/cron.deny
```



```
# 白名单配置文件（优先级高于黑名单）  
/etc/cron.allow
```

四、权限管理

```
# owner、group、others  
# read、write、execute  
读权限：  
    对于文件夹：读权限影响用户是否能够列出目录结构  
    对于文件：读权限影响用户是否可以查看文件内容  
写权限：  
    对于文件夹：写权限影响用户是否能够创建/删除/复制到/移动到文档  
    对于文件：写权限影响用户是否可以编辑文件  
执行权限：  
    对于文件：执行权限影响用户是否可以执行文件  
  
# 查看权限  
ls -l [path]  
ll  
  
# 设置权限（仅root或owner可用）  
chmod [option] [mode] [document]  
    -R：递归设置权限（选项）  
    -u：所有者身份  
    -g：给所有者同组用户设置  
    -o：给其他用户设置权限  
    -a：给所有人设置权限  
    -+：增加权限  
    --：减少权限  
    -=：确定权限  
    -r：可读  
    -w：可写  
    -x：执行  
chmod u+x, g+rx, o+r [filename]  
chmod u=x, g=rx, o=r [filename]  
chmod a=--- [filename]  
  
# 数字权限 r=4, w=2, x=1, 尽量不要出现2和3  
chmod 777 [filename]  
  
# 使用root用户创建文件夹，权限默认为 drwxr-xr-x  
# 在Linux中，如果要删除一个文件，不是看文件有没有对应的权限，而是看文件所在目录有没有写权限
```

1、chown

```
属主：所属的用户  
属组：所属的用户组  
# 更改文档的所属用户  
chown -R [username] [document]  
# 更改文档的所属用户和所属用户组  
chown -R [username:groupname] [document]
```

2、chgrp

```
# 更改文档的所属用户组
chgrp -R [groupname] [path]
```

3、sudo

```
# 配置
visudo

# 给某个用户增加命令权限
test ALL=(ALL) [commandPath]

# 查询特殊权限
sudo -l
```

五、网络

1、ping

```
# 检测当前主机与目标主机之间的连通性
ping www.baidu.com
```

2、netstat

```
# 查看网络连接信息
netstat
netstat -tnlp
netstat -an
```

3、tracert

```
# 查找当前主机与目标主机之间所有的网关
tracert www.baidu.com
```

4、arp

```
# 地址解析协议（Address Resolution Protocol）根据IP地址获取MAC地址的协议
# 查询地址
arp -a
# 删除缓存地址
arp -d
```

5、tcpdump

```
# 用来抓包
tcpdump [protocol] [port]
tcpdump [protocol] [port] [host]
tcpdump -i [networ]
```

六、shell

```
# Linux中shell的位置, bash(Bourne shell)
/bin/bash
```

```
# 修改shell解释器, 用户不能登录
usermod -s /sbin/nologin [username]
```

代码规范:

```
#!/bin/bash      指定告知系统当前这个脚本要使用的shell解释器
```

文件命名规范:

文件名.sh

输出的内容包含字母和符号, 则需要用引号包括起来。纯数字可包可不包

```
#-----tese.sh-----#
#!/bin/bash
echo "hello world"
#-----#
```

执行:

```
chmod +x test.sh
./test.sh
或者
chmod +x test.sh
/bin/bash test2.sh
```

```
#变量定义, 变量名等号左右不能有空格, 双引号能够识别变量, 单引号不能
class_name="white"
```

```
#变量使用, 变量的使用必须在变量名前面加上$符号
echo $class_name
```

```
#只读变量
```

```
readonly [variable name]
```

```
#获取输入
```

```
read -p [tip message] [variable name]
```

```
#删除变量
```

```
unset [variable name]
```

```
#条件语句
```

```
if[condition];then command; fi
```

```
if condition
```

```
then
    command1
    command2
    ...
fi

if condition
then
    command1
    command2
    ...
else
    command
fi

if condition
then
    command1
elif condition
then
    command2
else
    command...
fi

#算术运算符(借助expr工具)
val=`expr $a + $b`

#关系运算符
$a -eq $b
$a -ne $b
$a -gt $b
$a -lt $b
$a -ge $b
$a -le $b

#逻辑运算符
!false
$a -o $b
$a -a $b

#字符串运算符
$a = $b
$a != $b
-z $a    #判断字符串长度是否为0, 为0返回true
-n $a    #判断字符串长度是否为0, 不为0返回true

#文件测试运算符
-b $file    #是否为块设备文件, 是返回true
-c $file    #是否为字符设备文件, 是返回true
-d $file    #是否为目录, 是返回true
-f $file    #是否为普通文件
-p $file    #是否为有名管道
-r $file    #是否为可读
-w $file    #是否为可写
-x $file    #是否为可执行
-s $file    #检测文件是否为空
-e $file    #检测文件(目录)是否存在
```

#附带选项

\$1 \$2 #第一个输入参数 第二个输入参数