linux系统入门:

1.操作系统介绍

软件：驱动硬件

硬件：看得见，摸得着

操作系统：特殊的软件，控制硬件，内核+软件

操作系统：

unix(1969年诞生于贝尔实验室)

硬件-系统(hp-ux)-软件

BSD：freeBSD 黑莓

ios，mac os，苹果的 第一代 操作系统于1984年 发布

ms-dos 单任务

window 1.0 1985年发布 windows NT

开源 闭源

minix

linux：1991年 林纳斯.托瓦兹 linux内核 1.0/GPL

现在的操作系统：多任务，多用户

GNU：贡献了很多免费使用的软件

GPL 协议

UNIX和类Unix操作系统				
UNIX System V家族	A/UX	AIX	HP-UX	IRIX
	LynxOS	SCO OpenServer	Tru64	Xenix
	Solaris	OS/2		
BSD UNIX-386BSD家族	BSD/OS	FreeBSD	NetBSD	NEXTSTEP
	Mac OS X	iOS	OpenBSD	SUN OS
	OpenSolaris			
UNIX-Like	**GNU**	Linux	Android	Debian
	Ubuntu	Red Hat	Linux Mint	Minix
	QNX	GNU/Linux	GNU/Hurd	Debian GNU/Hurd
	GNU/kFreeBSD	StartOS		
其他	DOS	MS-DOS	Windows	React OS

gnu/linux:

红帽系： fedora redhat centos 最稳定企业服务器操作系统centos6 2.6.32 centos7 3.10 centos8 4.10

debian系： ubuntu kali 国产linux 内核 5.10 版本新 人工智能 自动驾驶 大数据

suse： opensuse suse企业版

2.centos系统的安装

ios镜像下载地址：

https://mirrors.tuna.tsinghua.edu.cn/centos-vault/7.8.2003/isos/x86_64/CentOS-7-x86_64-DVD-2003.iso

linux远程控制：

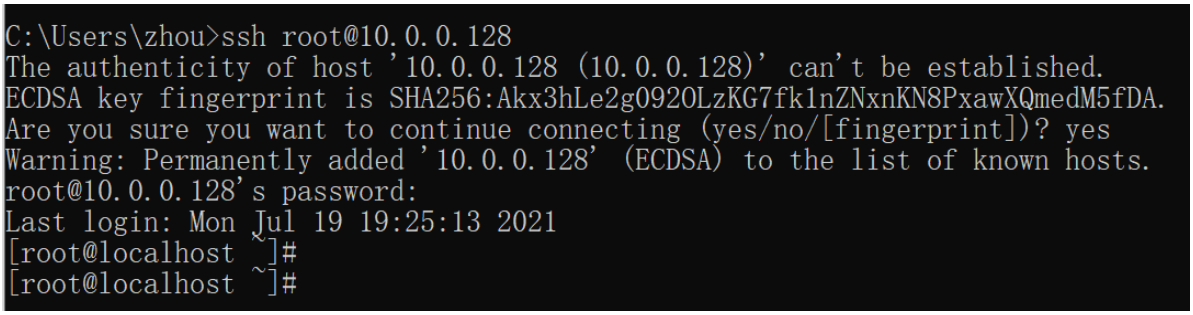
telnet协议： 明文

ssh协议： 加密

3.远程连接linux

第一步：查看ip地址 ip addr


```
#远程登录linux系统
ssh  root@192.168.202.136
#断开连接
exit
#重启系统
reboot
#立即关机
shutdown  -h  now
#十分钟之后关机
shutdown  -h  10
```



4.文件操作

命令1：

```
#切换目录
cd  #全称change directory
例子1:
cd  local    #切换到子目录local
cd  /usr/local  #切换到目录/usr/local
cd  ..       #切换到上一级目录
```

命令2:

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

命令3:

```
#新建文件
touch
例子1:  touch 1.txt  #创建单个文件
例子2:  touch test{01..10}.txt #批量创建文件
例子3:  touch /root/4.txt  #在指定的/root目录下，创建文件4.txt
```

命令4:

```
#查看目录下的文件
ls全称list
例子1：ls  test09.txt   #查看test09.txt是否存在
例子2：ls  *.txt        #查看以txt结尾的所有文件
例子3：ls  -1          #以一行一个文件的方式显示
例子4：ls  -a  al          #查看所有文件，包括隐藏文件
例子5：ls -a -1         #查看所有文件，以一行一个来显示
```

以.开头的文件是隐藏文件，默认不显示

命令5:

```
#重命名
mv全称move
例子1：mv .123.txt 123.txt   #将.123.txt文件重命名为123.txt
例子2：mv 123.txt /opt       #将当前目录下的123.txt移动到/opt目录下
```

命令6:

```
#复制
cp全称copy
例子1：cp test01.txt /opt/   #将当前目录下的test01.txt复制到/opt目录下
例子2：cp -a dev04 /opt/     #将目录dev04复制到/opt下
```

命令7:

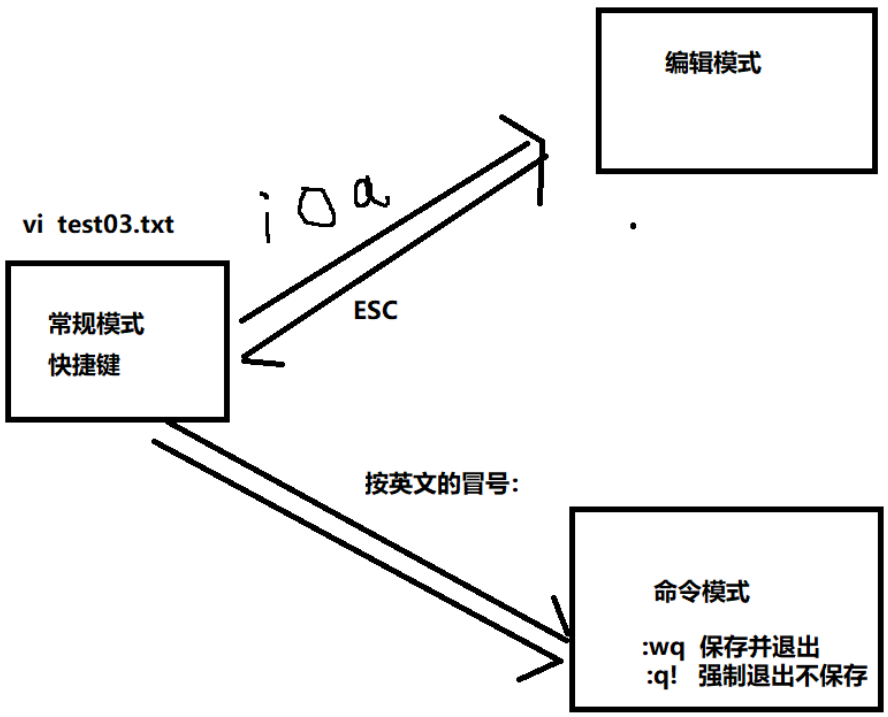
```
#删除
例子1:  rm /opt/123.txt  #将/opt目录下的123.txt文件删除，需要回复y确认删除
例子2:  rm -f /opt/test01.txt #将/opt目录下的test01.txt文件删除，不需要回复

#删除一个目录，linux的参数大部分没有先后顺序
[root@localhost ~]# rm -fr dev
[root@localhost ~]# rm -rf dev01
[root@localhost ~]# rm -f -r dev02
[root@localhost ~]# rm -r -f dev03
```

命令8:

```
#创建文件夹 创建目录directory
mkdir 全称make directory
例子1：mkdir dev   #创建一个dev目录
例子2：mkdir dev{01..10}     #批量创建多个目录
例子3：mkdir -p 1/2/3/4/5/6  #一次性创建多级子目录
```

命令9:

```
#vi编辑器
例子1:  vi  test03.txt   #编辑文件test03.txt
默认是常规模式，按ioa键进入编辑模式
在编辑模式中按esc回到常规模式
常规模式按:进入命令模式
命令模式按esc回到常规模式
```



命令8:

```
#从上往下顺序查看文本内容
cat
例子1：cat  test03.txt  #查看test03.txt的全部内容
#从下往上倒着查看文本内容
tac
例子1：tac  test03.txt  #倒着查看test03.txt的全部内容
```

命令9：

```
#查看文件头几行
head
例子1：  head  test03.txt       #查看文件的前十行，默认
例子2：  head  -n 5 test03.txt  #查看文件的前5行
例子3：  head  -5  test03.txt   #查看文件的前5行
```

命令10：

```
#查看文件倒数几行
tail
例子1：  tail  test03.txt       #查看文件的倒数十行，默认
例子2：  tail  -n 5 test03.txt  #查看文件的倒数5行
例子3：  tail  -5  test03.txt   #查看文件的倒数5行
```

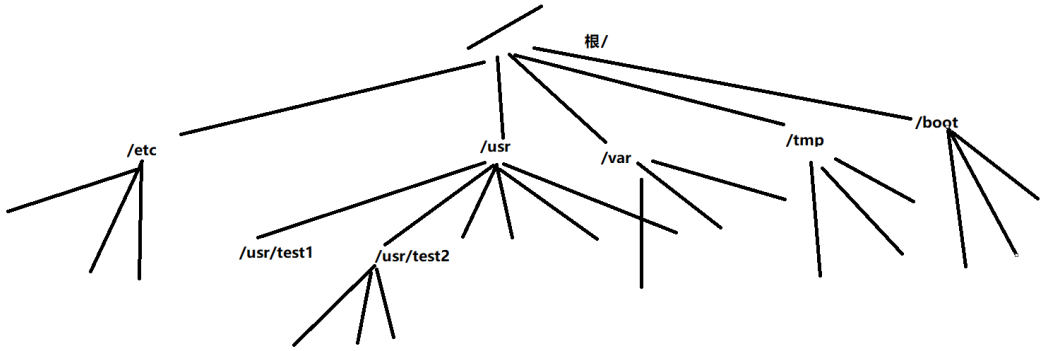
管道

```
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:83:e4:d9 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.128/24 brd 10.0.0.255 scope global noprefixroute dynamic ens33
       valid_lft 1253sec preferred_lft 1253sec
    inet6 fe80::ffe1:31ed:56dc:d9aa/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

[root@localhost ~]# ip addr|tail -4
    inet 10.0.0.128/24 brd 10.0.0.255 scope global noprefixroute dynamic ens33
       valid_lft 1224sec preferred_lft 1224sec
    inet6 fe80::ffe1:31ed:56dc:d9aa/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

[root@localhost ~]# ip addr|tail -4|head -1
    inet 10.0.0.128/24 brd 10.0.0.255 scope global noprefixroute dynamic ens33

[root@localhost ~]# ip addr|tail -4|head -1|cut -c 10-19
10.0.0.128
```

5.目录结构

相对路径：从所在目录开始

绝对路径：从根开始




```
/bin      #存放二进制的可执行文件，命令，特别重要，不能删除！
/boot     #开机启动需要的文件，  特别重要，不能删除！
/dev      #device设备文件，特别重要，不能删除！
/etc      #存放配置文件，特别重要，不能删除！
/home     #普通用户的家目录
/lib      #library 32位库，一般是so结尾，特别重要，不能删除！
/lib64    #library 64位库，一般是so结尾，特别重要，不能删除！
/media    #多媒体 (音乐 视频 文档)
/mnt      #mount挂载光盘，U盘
/opt      #部分软件安装存储目录
/proc     #process进程 ，  特别重要，不能删除！
/root     #root用户的家目录，特别重要，不能删除！
/run      #运行，程序运行的时候产生的文件
/sbin     #super bin超级用户才能使用的命令 ，特别重要，不能删除！
/srv      #源代码
/sys      #system系统目录 ，特别重要，不能删除！
/tmp      #用来存放临时文件的目录
/usr      #用户级的目录，特别重要，不能删除！
/var      #variable 变化的文件，特别重要，不能删除！
```

命令11

```
#统计
wc
wc -l按行统计，不会单独使用，需要接在管道后面
例子1：
[root@localhost ~]# cat test03.txt |wc -l
18
[root@localhost ~]# cat -n test03.txt
     1  sdsdsdsdsdsdsd是的是的
     2
     3  sdsd
     4  65656
     5  sdsd
     6  sdsd
     7  sdsd
     8  33333
     9  565656565
    10  33333
    11  565656565
    12  33333
```

```
    13  565656565
    14  33333
    15  565656565
    16
    17
    18  sdssds  ;;;;
```

命令12

```
#生成数字序列
seq
例子1：产生一个5到12的序列
[root@localhost ~]# seq 5 12
5
6
7
8
9
10
11
12

例子2：产生一个5到12等宽的序列
[root@localhost ~]# seq -w 5 12
05
06
07
08
09
10
11
12
```

命令13

```
#切换目录
cd 全称 change directory
例子1：cd  /opt
```

命令14

```
#打印当前目录
pwd == print work directory
例子1：
[root@localhost opt]# pwd
/opt
```

命令15

```
#按行过滤字符串
grep
例子1：  #普通过滤
[root@localhost ~]# grep '3' test03.txt
33333
33333
```

```
33333
33333
例子2：  #显示行号
[root@localhost ~]# grep -n '3' test03.txt
8:33333
10:33333
12:33333
14:33333
```

grep精准匹配-w

```
[root@localhost ~]# ip addr|grep 'inet'
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 10.0.0.128/24 brd 10.0.0.255 scope global noprefixroute dynamic ens33
    inet6 fe80::ffe1:31ed:56dc:d9aa/64 scope link noprefixroute
[root@localhost ~]#
[root@localhost ~]# ip addr|grep -w 'inet'
    inet 127.0.0.1/8 scope host lo
    inet 10.0.0.128/24 brd 10.0.0.255 scope global noprefixroute dynamic ens33
```

命令16

```
#按列过滤
awk
例子1：  取列,$1代表第一列，$2代表第二列，$NF代表最后一列
[root@localhost ~]# cat test01.txt
row 1, cell 1   row 1, cell 2
row 2, cell 1   row 2, cell 2
[root@localhost ~]# awk '{print $1}' test01.txt
row
row
[root@localhost ~]# awk '{print $2}' test01.txt
1,
2,
例子2：以逗号,做分隔符
[root@localhost ~]# cat test01.txt
row 1, cell 1   row 1, cell 2
row 2, cell 1   row 2, cell 2
[root@localhost ~]# awk -F ','   '{print $1}' test01.txt
row 1
row 2
[root@localhost ~]# awk -F ','   '{print $NF}' test01.txt
 cell 2
 cell 2
```

命令17

```
#排序
sort
例子1：
[root@localhost ~]# cat test02.txt
3
2
6
4
8
7
5
```

```
3
2
1
2
3
4
5
6
9
1
5
7
[root@localhost ~]# cat test02.txt|sort -n
1
1
2
2
2
3
3
3
4
4
5
5
5
6
6
7
7
8
9
```

命令18:

```
#统计去重
uniq
例子1:
[root@localhost ~]# cat test02.txt|sort -n
1
1
2
2
2
3
3
3
4
4
5
5
5
6
6
7
7
8
```

```
9
[root@localhost ~]# cat test02.txt|sort -n|uniq -c
      2 1
      3 2
      3 3
      2 4
      3 5
      2 6
      2 7
      1 8
      1 9
```

6.用户和用户组管理

命令19：

```
#创建用户
useradd
#创建一个用户
例子1：useradd  test1

#创建用户，并给用户指定用户组
例子2：
[root@localhost tmp]# useradd  -g test1  test3
[root@localhost tmp]# id test3
uid=1002(test3) gid=1000(test1) 组=1000(test1)
```

命令20：

```
#设置密码
passwd
例子1：passwd  test1
#用root用户给普通用户修改密码
[root@localhost ~]# passwd test1
更改用户 test1 的密码 。
新的 密码：123456
无效的密码： 密码是一个回文
重新输入新的 密码：123456
passwd：所有的身份验证令牌已经成功更新。

#普通用户自己修改密码
[test1@localhost ~]$ passwd
更改用户 test1 的密码 。
为 test1 更改 STRESS 密码。
（当前）UNIX 密码：
新的 密码：
无效的密码： 密码少于 8 个字符
新的 密码：
无效的密码： 密码少于 8 个字符
新的 密码：
无效的密码： 密码未通过字典检查 - 过于简单化/系统化
passwd: 已经超出服务重试的最多次数
123@qq.COM
1qaz@WSX

例子3：
```

```
#f交互修改密码
echo 123456|passwd --stdin test1
```

命令21：

```
#检查用户是否存在
id
例子1:
#用户存在，系统的返回结果
[root@localhost ~]# id test1
uid=1000(test1) gid=1000(test1) 组=1000(test1)

#用户不存在，系统的返回结果
[root@localhost ~]# id test2
id: test2: no such user
```

命令22：

```
#删除用户
userdel
例子1：
#被删除的用户还在登录状态
[root@localhost ~]# userdel test1
userdel: user test1 is currently used by process 2356
#被删除的用户，退出登录之后，可以正常删除
[root@localhost ~]# userdel test1
```

命令23：

```
#修改用户信息
usermode

#锁定用户
例子1：
[root@localhost ~]# usermod -L test1
[root@localhost ~]# lchage -l test1
帐号被锁。
至少：0
至多：99999
警告：7
不活跃：     从不
最后一次改变：  2021年07月20日
密码过期：    从不
密码不活跃：   从不
帐号过期：    从不

#禁止用户登录
[root@localhost ~]# usermod -s /sbin/nologin test2
[root@localhost ~]# grep -w 'test2' /etc/passwd
test2:x:1001:1001::/home/test2:/sbin/nologin
```

命令24：

```
#查看用户详细信息
lchage
例子1：
[root@localhost ~]# lchage -l test1
帐号没被锁。
至少：0
至多：99999
警告：7
不活跃：        从不
最后一次改变：  2021年07月20日
密码过期：      从不
密码不活跃：    从不
帐号过期：      从不
```

所有的用户信息存储在/etc/passwd文件中

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd
daemon:/dev/null:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
test1:x:1000:1000::/home/test1:/bin/bash
```

```
#passwd文件解释
root:x:0:0:root:/root:/bin/bash
test1:x:1000:1000::/home/test1:/bin/bash
test2:x:1001:1001::/home/test2:/sbin/nologin
test3:x:1002:1000::/home/test3:/sbin/nologin
haha:x:1004:1004:putong user:/home/haha:/bin/bash
第一列：用户名
第二列：x
第三列：uid
第四列：gid
第五列：注释，一般为空
第六列：家目录的位置
第七列：使用shell的名称，默认使用/bin/bash
```

所有的用户密码信息存储/etc/shadow

```
[root@localhost ~]# cat /etc/shadow
root:$6$QM3AHtFflOvGbCnt$2wTYZrnO8c66vycaxprE79G.I7hiy8EqXntG86FXxqlSawjtKoTjAnA
a9xFA3ad1QpFskJRPtOQeDPBnZZAdx0::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
operator:*:18353:0:99999:7:::
games:*:18353:0:99999:7:::
ftp:*:18353:0:99999:7:::
nobody:*:18353:0:99999:7:::
systemd-network:!!:18827::::::
dbus:!!:18827::::::
polkitd:!!:18827::::::
tss:!!:18827::::::
abrt:!!:18827::::::
sshd:!!:18827::::::
postfix:!!:18827::::::
chrony:!!:18827::::::
test1:$6$7PLnhlcr$4SCHINbWrameVSn6rxYgDOeUBwvYOHVbEOIudtNxWXIl5ULSt1Om8FvK6R3s9f
IOqNef1vq8NhVMNJwudmfkM1:18828:0:99999:7:::
```

7.权限管理

root用户权限最高

user普通用户

users用户组

文件权限

```
r 4 代表读权限 read
w 2 代表写权限 write
x 1 代表执行权限
- 0 空权限位

权限值表
0 ---
1 --x
2 -w-
3 -wx
4 r--
5 r-x
6 rw-
7 rwx

rw-   r--    r--
user  group  other
```

命令25

```
#改变文件的权限
chmod
例子1：
#修改权限之前
[test1@localhost tmp]$ ls -l
总用量 4
-rw-rw----. 1 test1 test1 8 7月  20 17:20 test1.txt
#修改权限
[test1@localhost tmp]$ chmod u+x test1.txt
#修改权限之后
[test1@localhost tmp]$ ls -l
总用量 4
-rwxrw----. 1 test1 test1 8 7月  20 17:20 test1.txt

例子2：
同时修改多个权限
[test1@localhost tmp]$ chmod u-x,g-x,o+x test1.txt
[test1@localhost tmp]$ ls -l
总用量 4
-rw-rw---x. 1 test1 test1 8 7月  20 17:20 test1.txt

例子3：
[test1@localhost tmp]$ chmod 777 test1.txt
[test1@localhost tmp]$ ls -l
总用量 4
-rwxrwxrwx. 1 test1 test1 8 7月  20 17:20 test1.txt
```

命令26

```
#修改文件的所属
chown
例子1：
[root@localhost tmp]# chown test2:test2 ls
[root@localhost tmp]# ls -l
总用量 404
-rwxr-xr-x. 1 test2 test2 159024 7月  20 17:43 grep
-rwxr-xr-x. 1 test2 test2 117608 7月  20 17:38 ls
-rwxr-xr-x. 1 test1 test1 130360 7月  20 17:43 mv
-rw-rw-rw-. 1 test1 test1     14 7月  20 17:38 test1.txt
[root@localhost tmp]# chown test1 ls
[root@localhost tmp]# ls -l
总用量 404
-rwxr-xr-x. 1 test2 test2 159024 7月  20 17:43 grep
-rwxr-xr-x. 1 test1 test2 117608 7月  20 17:38 ls
-rwxr-xr-x. 1 test1 test1 130360 7月  20 17:43 mv
-rw-rw-rw-. 1 test1 test1     14 7月  20 17:38 test1.txt

#使用uid和gid修改文件的所属用户和所属用户组  属主，属组
例子2：
[root@localhost tmp]# ls -l
总用量 404
-rwxr-xr-x. 1 test2 test2 159024 7月  20 17:43 grep
-rwxr-xr-x. 1 test1 test2 117608 7月  20 17:38 ls
-rwxr-xr-x. 1 test1 test1 130360 7月  20 17:43 mv
-rw-rw-rw-. 1 test1 test1     14 7月  20 17:38 test1.txt
[root@localhost tmp]# id test1
```

```
uid=1000(test1) gid=1000(test1) 组=1000(test1)
[root@localhost tmp]# id test2
uid=1001(test2) gid=1001(test2) 组=1001(test2)
[root@localhost tmp]# useradd  -g test1  test3
[root@localhost tmp]# id test3
uid=1002(test3) gid=1000(test1) 组=1000(test1)
[root@localhost tmp]# chown 1001:1001 test1.txt
[root@localhost tmp]# ls -l
总用量 404
-rwxr-xr-x. 1 test2 test2 159024 7月  20 17:43 grep
-rwxr-xr-x. 1 test1 test2 117608 7月  20 17:38 ls
-rwxr-xr-x. 1 test1 test1 130360 7月  20 17:43 mv
-rw-rw-rw-. 1 test2 test2     14 7月  20 17:38 test1.txt
```

文件属性

```
#文件属性
[root@localhost ~]# ls -l /tmp/123.txt
-rw-r--r--. 1 root root 0 7月  20 23:17 /tmp/123.txt
#第一段的第一个字符，表示文件类型 -文件 d目录 l软链接 b块设备
#第一段第2-4字符，表示该文件所属用户的权限
#第一段第5-7字符，表示该文件所属用户组的权限
#第一段第8-10字符，表示其他用户对该文件的权限
#第一段的第11个字符，表示开启selinux的状态下创建的
#第二段的数字，表示该文件的硬链接数量
#第三段的字符串，表示该文件所属用户
#第四段的字符串，表示该文件所属用户组
#第五段的数字，表示该文件的大小
#第六段到倒数第二段，都是该文件的修改时间
#最一段，该文件的名称
```

linux一切皆文件的系统

8.Linux Shell

优化ssh登录速度

```
#修改配置文件，先备份
[root@localhost ~]# cp /etc/ssh/sshd_config /tmp/
[root@localhost ~]# vi /etc/ssh/sshd_config
#直接输入:79回车
    79  GSSAPIAuthentication no
    115 UseDNS no
输入:wq保存退出

#重启sshd服务
[root@localhost ~]# systemctl restart sshd

#如果修改失败，还原配置文件
[root@localhost ~]# cp  /tmp/sshd_config  /etc/ssh/sshd_config
[root@localhost ~]# systemctl restart sshd
```

shell提示符

```
#root用户提示符
[root@localhost ~]#

#普通用户test1的提示符
[test1@localhost ~]$
[用户名@主机名 所在目录]#

#提示符格式定制
原格式
[root@test ~]# echo $PS1
[\u@\h \W]\$   # \u是用户，\h是主机名，\W是相对路径
修改后
[root@test ~]#cd /usr/local/bin/
[root@test bin]#export  PS1='[\u@\h \W]\$'
[root@test /usr/local/bin]#

#永久修改
[root@test 10:23:39 /usr/local/bin]#cd
[root@test 10:24:25 ~]#vi .bashrc
#找个空白的地方，插入一行
export  PS1='[\u@\h \t \W]\$'

#x可以参考https://www.cnblogs.com/Q--T/p/5394993.html
```

linux路径

```
相对路径  不完整路径，  例子../bin ./bin bin
绝对路径  也叫完整路径，例子/usr/local/bin/
```

命令hostname

```
#临时修改主机名
hostname
例子1：
[root@localhost ~]# hostname test
#需要重新登录生效
```

命令hostnamectl

```
hostnamectl

#查看主机的信息
[root@localhost ~]# hostnamectl
   Static hostname: localhost.localdomain
         Icon name: computer-vm
           Chassis: vm
        Machine ID: f8a89169114741a8ac6de82954c5fbcb
           Boot ID: dcf65386ccda42e29699d56101af8cf1
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1127.el7.x86_64
      Architecture: x86-64

#永久修改主机名
[root@localhost ~]# hostnamectl set-hostname test
```

```
#需要重新登录生效
```

命令reboot

```
#重启系统
reboot
例子1：
[root@test ~]# reboot
```

命令shutdown

```
#关闭或者重启linux
shutdown
例子1：
#立即关机
[root@localhost ~]# shutdown -h now

例子2：
#5分钟之后关机，可以使用shutdown -c取消
[root@localhost ~]# shutdown -h 5

例子3：
#5分钟之后重启系统，可以使用shutdown -c取消
[root@localhost ~]# shutdown -r 5
```

shell基础语法

```
命令    参数    目标
ls    -a -l    /opt
rm    -rf     /opt/test1

命令    目标1...    目标2
mv     源路径       目标路径
cp

命令
reboot

[root@test 11:12:02 /opt]#mv --help
用法：mv [选项]... [-T] 源文件 目标文件
  或：mv [选项]... 源文件... 目录
  或：mv [选项]... -t 目录 源文件...
#格式解释
[选项] 可选的参数
...     可以有多个
```

tab键补全

```
补全命令
#如果预选的特别多
[root@test 11:25:24 ~]#
Display all 1400 possibilities? (y or n)

#如果预选少
[root@test 11:25:24 ~]#cha
```

```
chacl   chage   chattr

补全路径
#如果预选的特别多
[root@test 11:25:24 ~]#cd /etc/
Display all 188 possibilities? (y or n)
#如果预选少
[root@test 11:25:24 ~]#cd /usr/src/
debug/   kernels/
```

快捷键

```
Ctrl + a    #光标跳转至正在输入的命令行的首部
Ctrl + e    #光标跳转至正在输入的命令行的尾部
Ctrl + c    #终止前台运行的程序
Ctrl + d    #在shell中，ctrl-d表示推出当前shell。
Ctrl + z    #将任务暂停，挂至后台，执行fg命令继续运行
Ctrl + l    #清屏，和clear命令等效。
Ctrl + k    #删除从光标到行末的所有字符
Ctrl + u    #删除从光标到行首的所有字符
Ctrl + r    #搜索历史命令，利用关键字
ctrl + w    #光标往前删除一个参数
esc + .     #上一条命令的最后一个参数，或者目标
```

history历史命令

```
#历史
history
例子1:
[root@test 14:32:10 ~]#history
    1  exit
    2  ls
    3  head -1 test03.txt
    4  head -1 test03.txt|cat
    5  head -1 test03.txt|tac
    6  head -2 test03.txt|tac
    7  head -2 test03.txt|cat
    8  ip addr
    9  ip addr|tail -4
   10  ip addr|tail -4|head -1
   ......

例子2:
#使用!调用历史命令
[root@test 14:32:10 ~]#history|head -5
    1  exit
    2  ls
    3  head -1 test03.txt
    4  head -1 test03.txt|cat
    5  head -1 test03.txt|tac
[root@test 14:32:26 ~]# !3
 head -1 test03.txt
head: 无法打开"test03.txt" 读取数据: 没有那个文件或目录

#使用!调用mv开头的命令
[root@test 14:42:17 ~]#history
```

```
    1  ls -a -l .bash_history
    2  history
    3  ls
    4  history
    5  mv aaaaa.txt /tmp/
    6  history
[root@test 14:42:19 ~]#!mv
mv aaaaa.txt /tmp/
mv: 无法获取"aaaaa.txt" 的文件状态(stat): 没有那个文件或目录


例子4:
#清除历史记录
history -c
删除主文件夹下面.bash_history
```

历史记录保存在家目录下的.bash_history文件中

命令别名(花名)

```
#别名
alias
例子1:
#查看别名
[root@test 15:23:17 ~]#alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-
tilde'

例子2:
#添加别名
[test1@test 15:24:23 ~]$alias rm='rm -i'
[test1@test 15:27:08 ~]$alias |grep rm
alias rm='rm -i'

例子3:
#取消别名
[test1@test 15:27:13 ~]$unalias rm
[test1@test 15:27:47 ~]$alias |grep rm

例子4:
#alias永久生效
[root@localhost ~]# vi .bashrc
#空白处，增加一行
alias rm='echo -bash: rm: 未找到命令'
```

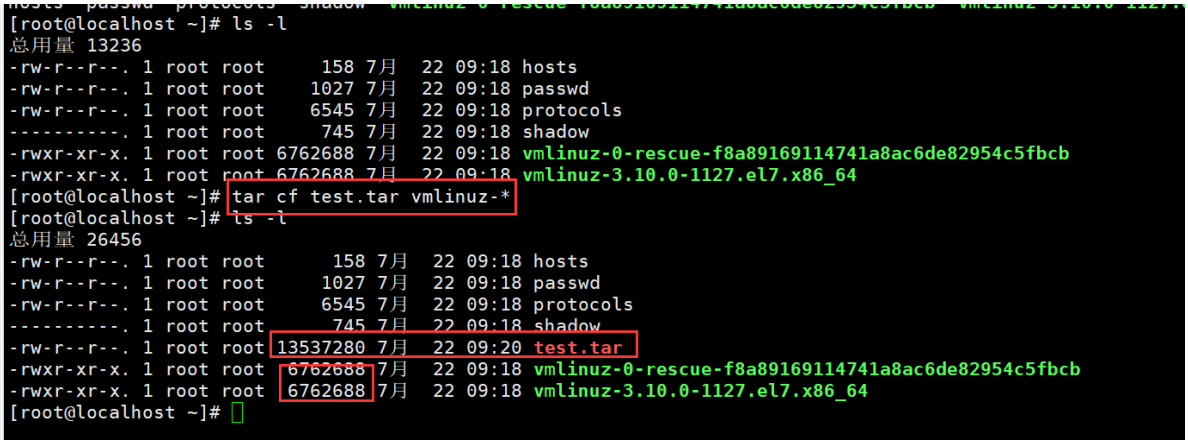
9.linux输入输出重定向

输入
< 输入重定向
<< 输入追加重定向
#标准输入0

输出
> 输出重定向， 将命令执行结果不输出到屏幕上，输出到文件里，会清空原文件
>> 输出追加重定向，不会清空原文件
#标准正确输出重定向1
#标准错误输出重定向2

10.linux压缩打包

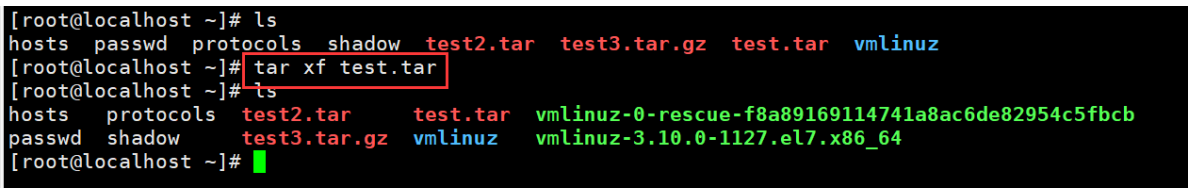
#打包和解压缩
tar
例子1：
#归档，不压缩
tar -cf test.tar vmlinuz-*



例子2：
#压缩并归档
tar -zcf test2.tar.gz vmlinuz-*

例子3：
#查看压缩包内容
[root@localhost ~]# tar -tf test.tar
vmlinuz-0-rescue-f8a89169114741a8ac6de82954c5fbcb
vmlinuz-3.10.0-1127.el7.x86_64

例子4：
#解压缩
tar -xf test.tar



```
#打包和压缩
gzip
#压缩单个文件
gzip protocols
#压缩多个文件，每一个文件产生一个单独的压缩包
gzip  hosts passwd shadow
#解压缩
gzip -d hosts.gz passwd.gz protocols.gz shadow.gz
```

```
#压缩
zip
例子1:
zip -r 123.zip day01

#解压
unzip
例子1:
[root@localhost file]# unzip 123.zip
```

```
#解压rar包
#需要安装软件
yum install epel-release -y
yum install unar -y
#再进行解压
unar -o /opt 456.rar
```

```
#下载文件
curl
例子1:
#下载文件
curl -o 123.zip http://192.168.18.10/day01.zip
```

```
#检查网络畅通
ping
例子1: ping 223.5.5.5
#如果网不通，重启网络服务
systemctl restart network
```

查看文件类型file

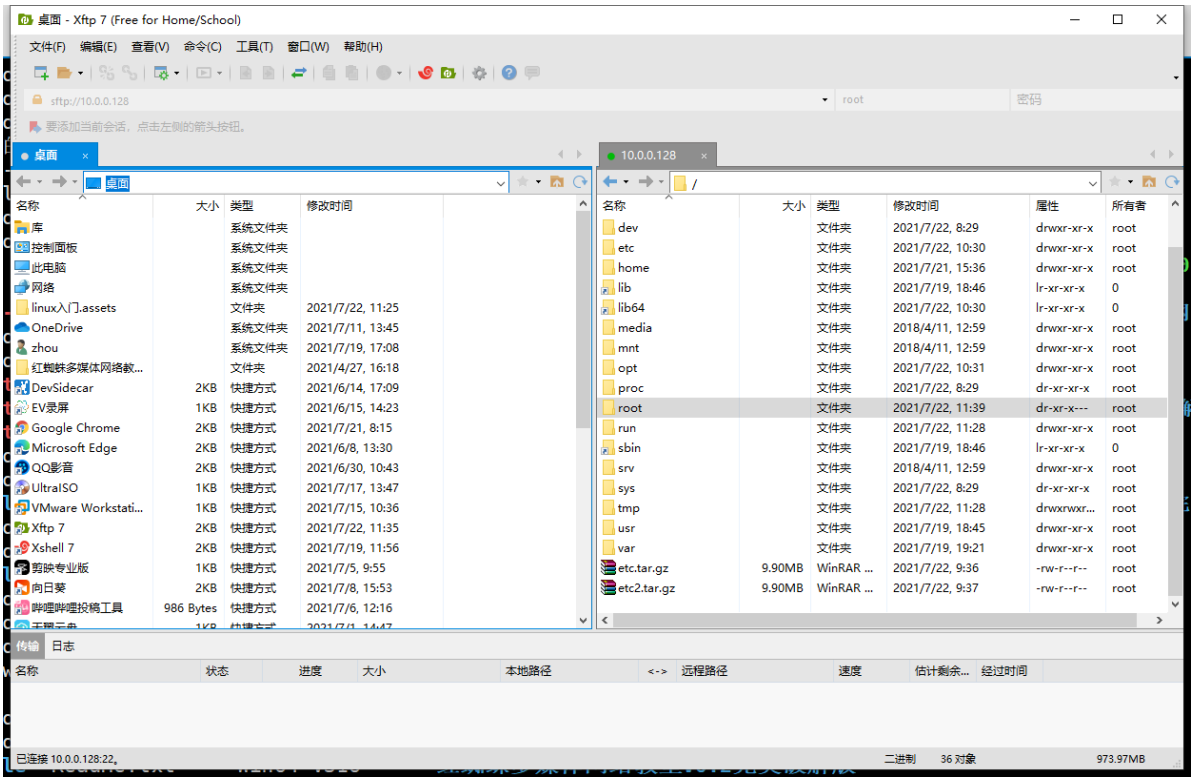
```
#查看文件类型
file
例子1:
file 123.zip
```

```
[root@localhost file]# file 1
1: ASCII text
[root@localhost file]# file 123.zip
123.zip: Zip archive data, at least v1.0 to extract
[root@localhost file]# file passwd.gz
passwd.gz: gzip compressed data, was "passwd", from Unix, last modified: Thu Jul 22 09:18:01 2021
[root@localhost file]#
gzip compressed data, from Unix, last modified: Tue Sep 10 05:24:25 2013, max compression
[root@localhost file]# file mysql.tar.gz
mysql.tar.gz: gzip compressed data, from Unix, last modified: Tue Sep 10 05:24:25 2013, max compression
```

11.文件传输

```
#上传和下载
rz  #上传
sz  #下载
#先安装lrzsz软件包
yum  install lrzsz  -y
#上传的例子
如果使用xshell，直接鼠标拖拽，或者执行rz -E选择要上传的文件
#下载的例子
sz /root/test3.tar.gz
```

xftp上传下载



scp传输

```
#主要用于linux和linux服务器之间传输文件
#把本地文件推送到远程服务端
scp typora-setup-x64.exe  root@10.0.0.128:/tmp
#把远端服务文件拉取到本地
scp root@10.0.0.128:/tmp/typora-setup-x64.exe  .
```

12.软件的安装

编译安装

```
1.下载源码包
cd /opt/
rm -fr *
curl -o nginx.tar.gz http://nginx.org/download/nginx-1.20.1.tar.gz

2.编译
tar xf nginx.tar.gz
cd nginx-1.20.1/
#编译参数
```

```
./configure --prefix=/usr/local/nginx --without-pcre --without-
http_rewrite_module --without-http_gzip_module
#编译
make
#安装
make install

3.运行
/usr/local/nginx/sbin/nginx
#使用浏览器访问http://<虚拟机的ip地址>
#关闭防火墙
systemctl stop firewalld
#取消防火墙的开机自启
systemctl disable firewalld
```

rpm安装

```
#redhat package manager包管理器
#安装wget
yum install wget -y
#使用wget下载rpm包
wget https://mirrors.tuna.tsinghua.edu.cn/centos/7/os/x86_64/Packages/tree-1.6.0-
10.el7.x86_64.rpm
#安装rpm包
rpm -ivh tree-1.6.0-10.el7.x86_64.rpm
#卸载
rpm -e tree
#升级
rpm -Uvh  xxx.rpm
#查看已安装的软件
rpm -qa|grep httpd
```

命令

```
#以树状的显示指定目录下的目录和文件的名称
tree
例子1:
tree /usr/local
```

yum安装

```
#自动解决rpm依赖
#yum安装扩展yum仓库
yum install epel-release -y
#yum安装nginx
yum install nginx -y
#yum移除nginx
yum remove nginx -y
#查看仓库rpm的数量
yum repolist
```

安装方法总结

编译安装：优点： 自由定制 痛点：难度高，步骤繁琐
rpm安装：优点：安装简单 痛点：需要自己解决依赖，不支持定制
yum安装：优点：自动解决依赖，默认安装最新版 痛点：不支持定制

13.find文件查找

#文件查找
例子1：普通查询
```
find   /etc    -maxdepth 1  -type f  -name "pa*"
命令   目录...   查找深度      类型      文件名称包含
```

```
[root@localhost ~]# find /etc -maxdepth 1   -type f  -name "pa*"
/etc/passwd
/etc/passwd-
[root@localhost ~]#
[root@localhost ~]# find /etc -type f  -name "pa*"
/etc/openldap/certs/password
/etc/passwd
/etc/passwd-
/etc/security/pam_env.conf
/etc/pam.d/passwd
/etc/pam.d/password-auth-ac
[root@localhost ~]#
```

例子2：按照文件大小查找(单位kMG)
查找大于100M的文件
```
find / -type f -size +100M
```
查找小于2k的文件
```
find /root/nginx-1.20.2 -type f -size -2k
```
查找大于50M同时小于100M的文件
```
find /  -type f   -size +50M  -and -size -100M
```

```
[root@linux01 nginx-1.20.2]# find /  -type f   -size +50M  -and -size -100M  2>/dev/null
/var/lib/rpm/Packages
/var/cache/yum/x86_64/7/updates/gen/primary_db.sqlite
/var/cache/yum/x86_64/7/updates/gen/filelists_db.sqlite
/boot/initramfs-0-rescue-356c7d6087214a578b92102ae49cb0b8.img
[root@linux01 nginx-1.20.2]# ls
```

例子3：
忽略大小写查询
```
find /etc -maxdepth 1    -iname "pa*"
```

```
[root@localhost ~]# find /etc -maxdepth 1    -name "pa*"
/etc/passwd
/etc/passwd-
/etc/pam.d
[root@localhost ~]# find /etc -maxdepth 1    -iname "pa*"
/etc/passwd
/etc/passwd-
/etc/pam.d
/etc/Passwd
```

例子4：
根据修改时间查找文件
时间参数：atime mtime ctime amin mmin cmin

```
#时间单位为天
find /opt -type f -mtime -1   #-1代表一天以内，+1一天以前
#时间单位为分钟
[root@localhost ~]# find /root -type f -mmin -20
/root/.bash_history
/root/ReadMe.txt
/root/.lesshst
```

例子5：
取反

```
[root@localhost ~]# find /root -type f -mmin -20  -name ".*"
/root/.bash_history
/root/.lesshst
[root@localhost ~]# find /root -type f -mmin -20  ! -name ".*"
/root/ReadMe.txt
[root@localhost ~]#
```

例子6：
对找出的文件进行处理

```
find /root -type f -mmin -30  ! -name ".*"  -exec rm {} \;
find /root  -maxdepth 1  -type d  -name "Apa*"   -mmin -30 -exec cp -a {} /tmp
\;
```

14.进程管理

进程：正在运行的程序

```
#查看进程
ps
参数1：ps -ef
```

```
[root@localhost ~]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 08:23 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --deserialize 2
root           2       0  0 08:23 ?        00:00:00 [kthreadd]
root           4       2  0 08:23 ?        00:00:00 [kworker/0:0H]
root           5       2  0 08:23 ?        00:00:00 [kworker/u256:0]
root           6       2  0 08:23 ?        00:00:00 [ksoftirqd/0]
root           7       2  0 08:23 ?        00:00:00 [migration/0]
root           8       2  0 08:23 ?        00:00:00 [rcu_bh]
root           9       2  0 08:23 ?        00:00:00 [rcu_sched]
root          10       2  0 08:23 ?        00:00:00 [lru-add-drain]
root          11       2  0 08:23 ?        00:00:00 [watchdog/0]
root          13       2  0 08:23 ?        00:00:00 [kdevtmpfs]
root          14       2  0 08:23 ?        00:00:00 [netns]
root          15       2  0 08:23 ?        00:00:00 [khungtaskd]
root          16       2  0 08:23 ?        00:00:00 [writeback]
root          17       2  0 08:23 ?        00:00:00 [kintegrityd]
root          18       2  0 08:23 ?        00:00:00 [bioset]
root          19       2  0 08:23 ?        00:00:00 [bioset]
root          20       2  0 08:23 ?        00:00:00 [bioset]
```

```
#关闭进程
kill
例子1：  kill  7851  #使用进程id号，来终止进程
         kill -9  pid号  #慎用！！！
pkill
例子1：  pkill sleep  #使用进程的命令名称，来终止进程
         pkill -9 sleep
```

```
#查看计算机的cpu，内存，进程等信息(windows任务管理器)
top

#查看内存命令
free -h

#查看硬盘命令
df -h

#查看cpu
lscpu
```

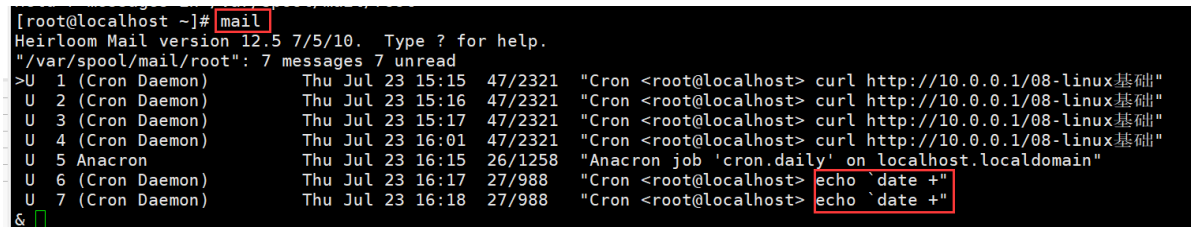
15.定时任务

定期执行任务（执行命令）

```
#时间命令
date
例子1:
#查看时间
[root@localhost ~]# date
2021年 07月 23日 星期五 14:38:19 CST
[root@localhost ~]# date +%F
2021-07-23
[root@localhost ~]# date +%T
14:35:47
[root@localhost tmp]# date +%F\ %T
2022-01-11 10:07:50

例子2:
#修改时间
[root@localhost ~]# date -s '20200723 14:40:00'
2020年 07月 23日 星期四 14:40:00 CST

#同步时间
systemctl restart chronyd
```

```
#定时任务的格式
*  *  * *  *    cmd
分 时 日 月 周    命令
分：0-59
时：0-23
日：0-31
月：1-12
周：1-7
#每5分钟执行一次
*/5  *  * *  *

#每1小时执行一次
01 */1 * * *

#每半个小时执行一次
00,30  */1 * * *

#每天晚上8:00执行一次
```

```
00 20 * * *

#每周1晚上8:00执行一次
00 20 * * 1,3,5

#查看定时任务，遇到特殊符号%,需要添加转义符号\;
[root@localhost ~]# crontab -l
* * * * *  echo `date +\%T` >>/tmp/time.txt

#编辑定时任务
[root@localhost ~]# crontab -e

如果定时任务的格式，或者内容有问题，系统都会发邮件提示
```



16.优化系统：

```
1.优化ssh
vi /etc/ssh/sshd_config
79行：GSSAPIAuthentication no
115行：UseDNS no
systemctl restart sshd

2.优化selinux
#修改配置文件，永久关闭
vi /etc/selinux/config
#第7行修改为
SELINUX=disabled
需要重启生效
#立即生效，临时的
setenforce 0
```

```
[root@localhost ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          disabled
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      31
[root@localhost ~]# setenforce 0
[root@localhost ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          disabled
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      31
[root@localhost ~]#
```

```
3.关闭firewalld
systemctl stop firewalld
systemctl disable firewalld

4.安装常用软件
yum install lrzsz vim tree wget net-tools screen tcpdump bash-completion -y
```

17.服务管理

服务管理

```
查看所有服务列表
systemctl list-unit-files
启动服务
systemctl  start  httpd  #start启动
停止服务
systemctl  stop   httpd  #stop停止
重启服务
systemctl  restart   httpd #restart重启
查看服务状态
systemctl  status   httpd  #查看服务状态
把服务设置为开机启动
systemctl enable httpd.service
取消服务的开机自启
systemctl disable httpd.service
```

软件和服务都是程序

软件一般是客户端curl

服务一般是给客户端提供某一个功能，7*24一直运行