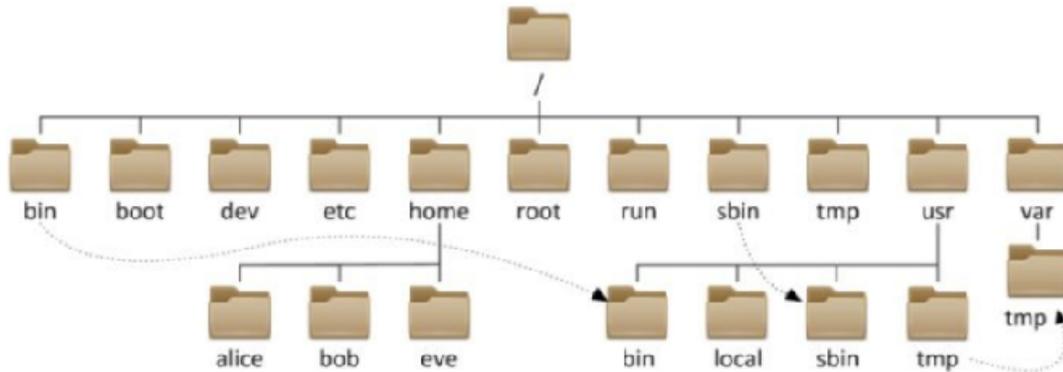


第二章 Linux基础命令

1、Linux目录结构

1) 目录结构



- /, 根目录, 最顶级的目录
- 路径描述的层次关系同样适用 / 来表示
- /home/test/a.txt, 表示根目录下的home文件夹内的test文件夹下有a.txt

2) 命令提示符结构说明

进入系统后, 任意空白处-》右键-》“打开终端”。终端类似于windows下的命令提示符



在终端中显示如下:

```
[ root@linuxvm1 ~] #
```

说明:

- @前的部分: 表示当前登录账户, 当前使用root账户登录
- @后的部分: 表示主机名, 当前主机名是linuxvm1
- 主机名后区域表示当前所处的目录, ~表示Home目录。
- 如果使用root登录则显示#, 如果使用普通用户则显示\$

2、Linux命令特点

1) Linux命令特点

- 在Linux系统中，命令区分大小写。
- 在命令行中，可以使用“Tab”键来自动补齐命令。
- 利用向上或向下的光标键，可以翻查曾经执行过的历史命令，并可以再次执行。
- 如果要在一个命令行上输入和执行多条命令，可以使用分号来分隔命令，如“cd /;ls”
- 断开一个长命令行，可以使用反斜杠“\”，可以将一个较长的命令分成多行表达。

2) 命令通用格式

```
命令名称 [选项] [参数]
```

说明：

- 命令名称 表示该命令可以做些什么
- 选项 表示该命令可以怎么做
- 参数 表示该命令对谁做

3、查看目录和文件的命令

查看目录类===

1) ls：用于列出目录下内容

语法：

```
ls [选项] [路径]
```

ls命令的常用选项如下

- a：显示所有文件，包括以“.”开头的隐藏文件。
- c：按文件的修改时间排序。
- l：以长格形式显示文件的详细信息。
- u：以文件上次被访问的时间排序

例：查看当前目录下所有文件

```
[root@localhost ~]#ls -al
```

当前目录“~”是用户的Home目录，也称为用户的家目录

每一个用户在Linux系统中都有自己的专属工作目录，称之为HOME目录。

- 普通用户的HOME目录，默认在： /home/用户名
- root用户的HOME目录，在： /root

例：查看/test目录下所有文件

```
[root@localhost ~]#ls -l /test
```

2) ll: 是ls -l的别名

3) pwd: 用于显示用户当前所处的目录

pwd: **p**rint **w**ork **d**irectory

例: 使用root登录后, 查看所在目录, 显示/root, 该目录是root的家目录

```
[ root@linuxvm1 ~]# pwd
/root
```

← root用户的家目录

4) cd: 用于在不同的目录中进行切换

cd: **c**hange **d**irectory

语法:

```
cd [linux路径]
```

- cd命令无需选项, 只有参数, 表示要切换到哪个目录下
- cd命令直接执行, 不写参数, 表示回到用户的HOME目录

例:

```
[root@localhost ~]# cd dir1 //改变目录位置至当前目录下的dir1子目录下
[root@localhost dir1]# cd ~ //改变目录位置至用户的家目录
[root@localhost ~]# cd .. //改变目录位置至当前目录的父目录
[root@localhost /]# cd //改变目录位置至用户的家目录
[root@localhost ~]# cd /dir1/subdir1 //利用绝对路径表示改变目录到/dir1/subdir1下
```

不带任何参数的“cd”命令相当于“cd ~”, 即将目录切换到用户的家目录。

注意:

cd tmp与**cd /tmp**的作用不同, **cd tmp**是进入当前目录下的tmp目录, **cd /tmp**是进行根目录下的tmp目录

5) 相对、绝对路径和特殊路径符

绝对路径: 以**根目录为起点**, 描述路径的一种写法, 路径描述**以/开头**

相对路径: 以**当前目录为起点**, 描述路径的一种写法, 路径描述**无需以/开头**

. 表示当前目录, 比如 `cd ./Desktop` 表示切换到当前目录下的Desktop目录内, 和 `cd Desktop` 效果一致

.. 表示上一级目录, 比如: `cd ..` 可切换到上一级目录

~ 表示HOME目录, 比如: `cd ~` 即可切换到HOME目录

查看文件类===

6) cat: 查看文件内容

cat会显示文件中所有内容

语法:

```
cat 文件路径
```

例: 查看刚才创建的info.txt文件的内容

```
[root@localhost ~]# cat info.txt //使用相对路径查看当前目录下的info.txt文件
```

cat可以连接多个文件一并输出

例:

```
[root@localhost ~]# cat -n info.txt stu.txt // 将当前目录下的2个文件连接输出
```

7) more: 查看文件内容, 可向下翻页

语法:

```
more 文件路径
```

例: 查看/etc/services文件

```
[root@localhost ~]# more /etc/services
```

在查看过程中, 空格翻页, 按q退出

8) less: 查看文件内容, 可上下翻页, 并搜索关键词

语法:

```
less 文件路径
```

- 在查看过程中, 空格向下翻页, b键向上翻页, 向上或向下箭头可单行移动, 按q键退出
- 在查看过程中, 输入'/关键字'可进入搜索模式, 高亮显示搜索到的词语
- 在查看过程中, 输入 '=' 可显示当前正在浏览的行数, 回车继续查看文件

10) head: 从头查看文件内容, 默认显示开头10行

语法:

```
head [-n 显示行数] 文件路径
```

例:

```
[root@localhost ~]# head -n 5 /etc/services // 指定从头显示文件, 只显示5行
```

11) tail: 从末尾查看文件内容, 默认显示

使用方式与head类似

目录操作类=====

12) mkdir: 创建新的目录

mkdir: **make directory**

语法:

```
mkdir [-p] linux路径
```

-p 选项表示创建多级目录, 可选

例:

```
[root@localhost ~]# mkdir test      在当前目录下创建test目录
[root@localhost ~]# mkdir /test     在根目录下创建test目录
[root@localhost ~]# mkdir -p /mydir/test  创建多级目录, 在根目录下创建mydir, 在mydir
下创建test
```

13) rmdir: 删除目录

使用rmdir删除的目录需为空目录

语法:

```
rmdir 目录路径
```

文件操作类===

14) touch: 创建文件

语法:

```
touch linux路径
```

例:

```
[root@localhost ~]# touch info.txt    在家目录创建info.txt文件
```

echo: 输出内容

语法:

```
echo [-n -e -E] 输出内容
```

选项说明:

-n: 输出内容后不换行 -e: 内容中如果有转义字符, 则使用转义字符的功能

特殊用法:

echo可使用">"将内容输出至文件中, 使用">>"将内容追加至文件中

例:

```
[root@localhost ~]# echo "hello world" > /test.txt    将内容输出至根目录下的test.txt
文件中, 如果文件不存在则创建文件
[root@localhost ~]# echo "I love you" >> /test.txt    将内容追加至根目录下的test.txt
文件中, 如果文件不存在则创建文件
```

15) mv: 移动目录/文件或更改名称

语法:

```
mv 原路径 新路径
```

例:

```
[root@localhost ~]# mkdir /test    在根目录下创建test目录
[root@localhost ~]# mv /test /tmp/  将根目录下test目录移动到/tmp下
[root@localhost ~]# mv /tmp/test /tmp/mytest  将/tmp/test更改名称为/tmp/mytest
[root@localhost ~]# touch jack.txt    在home目录下创建jack.txt
[root@localhost ~]# mv jack.txt rose.txt  将jack.txt 改名为rose.txt
[root@localhost ~]# mv rose.txt /      将rose.txt移动到根目录
```

第二个参数如果目录存在, 则是移动

第二个参数如果目录或文件名不存在, 则是更改名称。

16) cp: 复制目录或文件

语法:

```
cp [-r -a] 原路径 新路径
```

选项说明:

-r: 递归复制, 用于复制目录

-a: 表示复制文件的同时复制文件的特性, 一般复制配置文件时使用

例:

```
[root@localhost ~]# cp rose.txt /tmp    复制当前目录下的rose.txt到/tmp下
[root@localhost ~]# cp rose.txt jack.txt  复制当前目录下的rose.txt并更改为jack.txt
[root@localhost ~]# cp -r /test /test2   复制根目录下test目录并更名为test2,复制目
录需加-r选项
[root@localhost ~]# cp -a /etc/hosts hosts_2  复制文件及文件属性和权限
```

```
[root@linuxvm1 ~]# cp /etc/hosts hosts_1    不加-a复制文件, 文件的属性和权限
[root@linuxvm1 ~]# ll /etc/hosts hosts_1    会发生改变
-rw-r--r--. 1 root root 158 6月 7 2013 /etc/hosts
-rw-r--r--. 1 root root 158 2月 15 21:22 hosts_1
[root@linuxvm1 ~]# cp -a /etc/hosts hosts_2  加-a复制文件, 会将原文件的属性和权限
[root@linuxvm1 ~]# ll /etc/hosts hosts_2    一起复制
-rw-r--r--. 1 root root 158 6月 7 2013 /etc/hosts
-rw-r--r--. 1 root root 158 6月 7 2013 hosts_2
```

17) rm: 删除文件或目录

语法:

```
rm [-r -f] 路径1 [路径2...]
```

选项说明:

-r : 用于递归删除, 用于删除目录

-f : 强制删除, 不给提示信息

例:

```
[root@localhost ~]# rm -r /test      删除根目录下test文件夹, 注意删除文件夹时, 需要加-r选项
[root@localhost ~]# rm /tmp/stu.txt  删除根目录中tmp目录下的stu.txt, 删除文件不需要加-r
[root@localhost ~]# mkdir /test     在根目录中创建test文件夹
[root@localhost ~]# rm -rf /test    强制删除根目录下test文件夹, 不显示提示消息
```

练习:

- 1、在根目录下创建export目录
- 2、进入export目录, 创建data目录
- 3、复制data目录为servers目录, 再复制一次为software目录

结构如下:

```
/export/data/
/export/servers/
/export/software/
```

- 4、在/export/data/目录下创建ui.txt文件, 文件内容为 "192.168.121.130 hadoop1"
- 5、将ui.txt文件移动到/export/servers/目录下, 并更改名称为用户ip.txt
- 6、删除/export/servers/下的userip.txt文件
- 7、删除整个/export目录

文件查找类===

18) find: 指定目录下搜索文件

语法:

```
find [路径] [匹配表达式]
```

未指定路径时, 默认在当前目录下搜索文件

匹配表达式:

-name 文件名 : 查找指定名称的文件, 名称支持使用通配符

*匹配0-n个任意字符, ?匹配任一字符, []匹配给定的任意单个字符, [^]匹配不是给定的任意单个字符

-user 用户名：查找属于指定用户的文件

-group 组名：查找属于指定组的文件

-type 文件类型：查找指定类型的文件。b是块设备文件（硬盘/软盘），c是字符设备文件(键盘/鼠标)，d是目录，p是管道文件，l是链接文件，f是普通文件

-newer file：查找比指定文件的修改时间更近的文件

例：

```
[root@localhost ~]# find / -name '*game*'      在根目录下搜索名称中包含game的文件
[root@localhost ~]# find / -user user1        在根目录下搜索属于用户user1的文件
[root@localhost ~]# find / -type f           在根目录下搜索所有普通文件
[root@localhost ~]# find -user root -type b   在当前目录下搜索属于root用户的块设备
```

有时find在查找时，会占用时间，可以在命令末尾跟上'&'符号让其后台运行，不影响其他命令的输入和运行。

```
[root@localhost ~]# find -name '*game*&      在后台运行查找命令
```

```
[root@linuxvm1 ~]# find -name "*game*" &
[1] 17160
[1]+ 完成          find -name "*game*"
```

19) locate：文件查找

语法：

```
locate 文件路径
```

文件路径中支持通配符 *, ?, [], [^]

```
[root@localhost ~]# locate '*.txt'          查找磁盘上所有以txt结尾的文件
```

20) grep：查找文件中包含给定字符串的行

语法：

```
grep [-n -数值] 搜索词 文件路径
```

- 选项-n，可选，表示在结果中显示匹配的行的行号。
- 选项-数值，表示显示查找到的搜索词所在的行及前后几行。
- 搜索词，必填，表示要搜索的词语，带有空格或其它特殊符号，建议使用""引号将搜索词包围起来。
- 文件路径，必填，表示要过滤内容的文件路径，可作为内容输入端口。

例：

```
[root@localhost ~]# grep -n "name" /etc/services  在/etc/services中查找所有包含name的行，并显示行号
[root@localhost ~]# grep -2 "name" /etc/services  在/etc/services中查找所有包含name的行及前后2行内容
```

练习:

- 1、查找文件名以"ens33"结尾的文件, 并使用cat查看该文件
- 2、查找/var下所有类型为"b"的文件
- 3、在家目录下找到属于root用户的所有txt文件
- 4、查看/etc/passwd文件中包含有root字样的行

系统信息类===

21) dmesg: 显示系统硬件信息

22) free: 查看系统内存

23) date: 查看系统当前的日期和时间

24) cal: 显示指定月或年的日历

语法:

```
cal [月 年]
```

25) clock: 从计算机的硬件获得日期和时间

进程管理类===

26) ps: 查看系统的进程

语法:

```
ps [选项]
```

常用选项如下:

- a : 显示当前控制终端的进程
- u : 显示进程的用户名和启动时间等信息
- e : 显示所有的进程

```
[root@linuxvm1 ~]# ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1405  0.2  2.6 327548 50208 tty1    Ssl+ 16: 46   0: 32 /usr/bin/X : 0 - backgroun
root      15045  0.0  0.1 116296  2864 pts/0    Ss   20: 43   0: 00 bash
root      16166  0.0  0.0 151064  1820 pts/0    R+   21: 00   0: 00 ps -u
```

显示内容第一行意思分别是:

- USER: 进程所有者;

- PID: 进程ID号;
- %CPU: cpu占用率;
- %MEM: 内存占用率;
- VSZ: 虚拟内存使用量(KB);
- RSS: 占用的固定内存量(KB);
- TTY: 当前终端所使用号码, 再打开一个终端就是pts/1;
- STAT: 是当前终端该进程状态,**S**是睡眠,**R**是运行,**T**是停止,**D**不可中断,**Z**僵死;
- START: 进程被启动的时间;
- TIME: 实际使用CPU的时间;
- COMMAND: 命令名称与参数。

27) pidof: 查询某个服务进程的PID值

```
pidof [服务名称]
```

例:

```
[root@localhost ~]# pidof bash          查看终端bash的pid号
```

28) kill: 终止进程

语法:

```
kill [-9] 进程PID
```

-9 表示强制关闭进程。

例:

```
[root@localhost ~]# kill -9 15045      强制关闭终端bash进程(每台主机上bash进程id不一样)
```

29) killall: 终止个名称的服务所对应的全部进程

语法:

```
killall [-9] [进程名]
```

例:

```
[root@localhost ~]# killall -9 httpd
```

30) top: 实时监控进程的状态

屏幕每5秒刷新一次, 可使用 `top -d 20`, 使用屏幕每20秒刷新一次

31) at: 在特定时间运行Linux命令

语法:

```
at [时间日期]
```

例：在指定时间将内容写入到文件

```
[root@localhost ~]# at 10:00 AM
at> echo 'hello'>/t1.txt
at> CTRL+D
```

其他命令类===

32) clear: 清屏

33) uname: 显示系统信息

```
uname -a
```

34) man: 列出命令的帮助手册

```
man 命令名
```

例:

```
man ls
```

```
LS(1)                                                    General Commands Manual

NAME
  ls, dir, vdir - 列目录内容

提要
  ls [选项] [文件名...]

  POSIX 标准选项: [-CFRacdilqru1]

GNU 选项 (短格式):
  [-1abcfgiklmnopqrstuxABCDGFLNQRSUX] [-w cols] [-T cols] [-I pattern] [--full-time]
  [base, commas, across, vertical, single-column] [--sort={none,
  [--time={atime, access, use, ctime, status}] [--color[={none, auto, always}]] [--help] [--

描述 ( DESCRIPTION )
  程序ls先列出非目录的文件项，然后是每一个目录中的“可显示”文件。如果
  没有选项之外的参数【译注：即文件名部分为空】出现，缺省为“.”（当前目录）。
  “使得目录与非目录项同样对待。除非“-a”选项出现，文件名以“.”开始的文件不属“可显示”
```

说明:

- NAME: 命令名称
- 提要/SYNOPSIS: 名字的概要，简单说明命令的使用方法
- 描述/DESCRIPTION: 详细描述命令的使用，如各种参数选项的作用

补充: 查询某个命令的帮助信息也可以使用 `命令名 --help`。可显示当前命令的使用方法

35) shutdown: 在指定时间关闭系统

语法:

```
shutdown [选项] 时间 [警告信息]
```

常用选项:

-r: 关闭后重新启动

-h : 关闭系统

-c : 取消关闭命令

时间形式:

now: 表示立即

hh:mm: 指定绝对时间, hh表示小时, mm表示分钟

+m: 表示m分钟后

例:

```
[root@localhost ~]# shutdown -r now          立刻重启
[root@localhost ~]# shutdown -h 12:00       12点关机
[root@localhost ~]# shutdown -h +10         10分钟后关机
[root@localhost ~]# shutdown -c            取消关机任务
```

36) reboot: 重新启动系统

37) poweroff: 立即停止系统, 并关闭电源

38) history: 列出用户最近执行的命令

```
[root@myvm ~] # history
 1  pwd
 2  ls
 3  more /etc/services
 4  clear
 5  pwd
 6  pwd; ls
 7  ls
 8  pwd
 9  ls
10  ls -a
11  ls -l
12  ls -a -l
13  ls -al
```

所执行的历史命令前有编号, 只要在编号前加"!", 就可以重新运行该历史命令。如图:

```
[root@myvm ~] # !6
pwd; ls
/root
anaconda-ks.cfg      公共  视频  文档  音乐
initial-setup-ks.cfg 模板  图片  下载  桌面
```

39) wget: 在终端中下载网络文件

```
wget [参数] 下载地址
```

第三章 VIM编辑工具

1、vim简介

1) vim简介

- vi是visual interface的简称，它可以执行输出、删除、查找、替换、块操作等众多文本操作。
- Vim是从vi发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。
- vim不是一个排版程序，它只是一个文本编辑程序。vim是全屏幕文本编辑器，没有菜单，只有命令

2) 进入vim编辑器

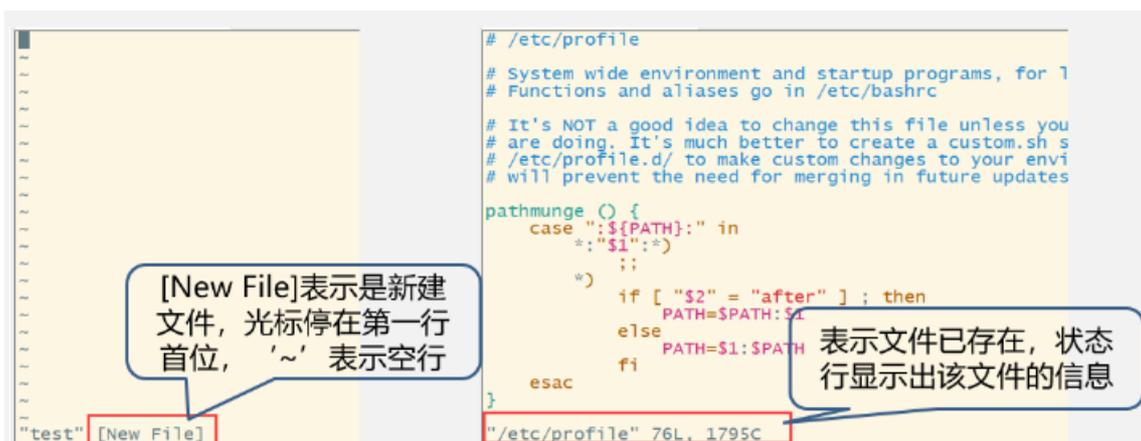
- 在系统提示符后输入vim和想要编辑（或建立）的文件名，便可进入vim
- 只输入vim，而不带文件名，也可以进入vim
- 初次进入vim不做任何操作是命令模式，键入:q（注意:号）就会退出vim。

2、Vim工作模式

vim有3种基本工作模式：命令模式、编辑模式和末行模式

1) 命令模式：

进入vim后默认是命令模式



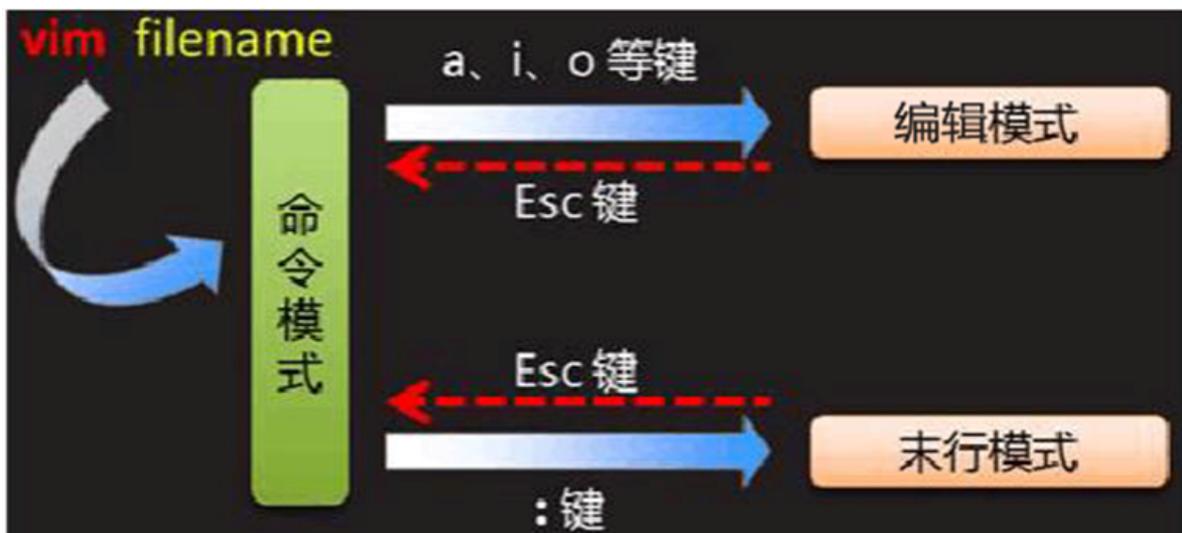
2) 编辑模式

- 在命令模式下按下相应的键可以进入编辑模式：插入命令i、附加命令a、打开命令o、修改命令c、取代命令r、替换命令s、
- 在编辑模式下，用户编辑的任何字符都被vim当作文件内容保存起来，并将其显示在屏幕上。在文本输入过程中（编辑模式下），若想回到命令模式下，按“Esc”键即可。

3) 末行模式

- 在命令模式下，用户按“:”键即可进入末行模式。
- 若在末行模式下输入命令的过程中改变了主意，可在用退格键将输入的命令全部删除之后，再按一下退格键，即可使vim回到命令模式。

4) 模式切换



3、vim常用命令

1) 命令模式命令说明

移动光标的命令	说明
Ctrl + f	屏幕向下移动一页，相当于“Page Down”键
Ctrl + b	屏幕向上移动一页，相当于“Page Up”键
G	移动到文件的最后一行
gg	移动到文件的第一行，相当于1G
n	n为数字。光标向下移动n行

查找命令	说明
/word	向下寻找名称为word的字符串
?word	向上寻找名称为word的字符串
n	这个n是英文按键。代表重复前一个查找的动作。
N	这个N是英文按键。与n刚好相反，为反向进行前一个查找动作
替换命令	说明
:n1,n2s/word1/word2/gc	n1,n2为数字，从n1~n2行寻找word1字符串，并将该字符串取代为word2。且在取代前显示提示字符给用户确认（confirm）是否需要取代

删除、复制与粘贴命令	说明
dd	删除光标所在的那一整行
ndd	n为数字。删除光标所在的向下n行，例如，5dd是删除5行
yy	复制光标所在的那一行
nyy	n为数字。复制光标处向下n行，例如20yy是复制20行
p, P	p为将已复制的数据在光标下一行粘贴上，P则为粘贴在光标上一行)
u	撤消前一个动作
Ctrl+r	重做上一个动作
.	重复前一个动作。如果想要重复删除、重复粘贴等动作，按下小数点就可以

2) 末行模式命令说明

常用命令	说明
:w	将数据写入硬盘文件中（常用）
:w!	若文件属性为只读时，强制写入该档案。
:q	退出vim
:q!	若曾修改过文件，又不想储存，则使用"!强制退出而不储存文件。
:wq	储存后离开，若为":wq!"，则为强制储存后离开
:w [filename]	将命令的数据储存成另一个文件（类似另存为新文件）
:set nu	显示行号，设定之后，会在每一行的前缀显示该行的行号
:nohl	取消高亮（查找字符串后，会高亮显示要查找的字符串）
:set nonu	与:set nu相反，为取消行号

练习：

1. 进入/tmp目录下，新建mydir目录；
2. 将/etc/hosts网络配置文件复制到/tmp/mydir下；
3. 使用vim编辑器打开复制的hosts文件，按i进入编辑模式，将本机IP地址和主机名(host1)映射；
4. 按下esc键回到命令模式，再按下:wq进入末行模式，保存并退出；
5. 复制/etc/profile文件至/tmp/mydir目录下，并重命名为myprofile文件；
6. 查看myprofile文件末尾20行内容；
7. 使用vim编辑器打开/tmp/mydir目录的myprofile文件；
8. 使用:set nu调出行号；
9. 查找myprofile中包含path字样的内容；
10. 跳转到53行，使用yy复制当前行；

11. 移动文件到最后一行;
12. 使用p在光标的下一行粘贴;
13. 使用dd删除光标所在行;
14. 编辑完后按下esc, 使用:wq进行保存并退出 (不保存仅退出使用:q!);
15. 删除/tmp/mydir目录下的hosts文件;
16. 删除/tmp下的mydir目录;

第四章 用户和组

1、用户配置文件

一个安全的系统，系统登陆密码是必要的。在登陆的时候，系统会验证用户名和密码，验证合格才能登陆，这样既起到了身份识别的作用，更能在一定程度上避免系统受到意外攻击。

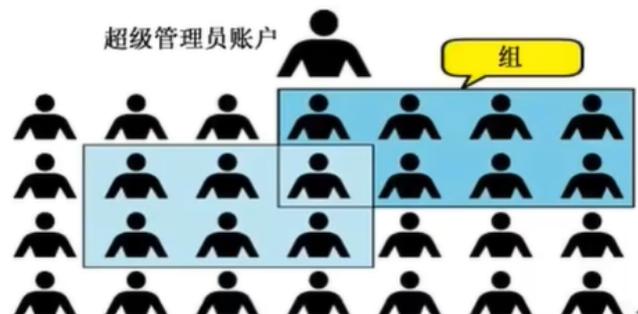
Linux操作系统是**多用户多任务**的操作系统。允许多个用户同时登录到系统。

Linux系统下的用户账户分为两种：**普通用户账户和超级用户账户**

用户、用户组

Linux系统中可以：

- 配置多个用户
- 配置多个用户组
- 用户可以加入多个用户组中



Linux中关于权限的管控级别有2个级别，分别是：

- 针对用户的权限控制
- 针对用户组的权限控制

比如，针对某文件，可以控制用户的权限，也可以控制用户组的权限。

1) 用户账号文件——passwd

- 文件“/etc/passwd”以行为单位，每行以冒号分隔为7个数据域，保存用户的账号信息。
- 由于**所有用户都对passwd有读取**权限，所以该文件中只定义用户账号，而不保存口令

文件格式如下 (`cat /etc/passwd`)：

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
user1:x:1002:1002::/home/user1:/bin/bash
```

说明：

文件中的每一行代表一个用户账户的资料，可以看到第一个用户是root。然后是一些标准账户，此类账户的shell为/sbin/nologin，代表无本地登录权限。最后一行是由系统管理员创建的普通账户

passwd文件的每一行用“:”分隔为7个域，各域的内容如下：

用户名:加密口令:UID:GID:用户的描述信息:HOME目录:执行终端(默认bash)

字段	说明
用户名	用户账号名称，用户登录时所使用的用户名
加密口令	用户口令，考虑系统的安全性，现在已经不使用该字段保存口令，而用字母“x”来填充该字段，真正的密码保存在shadow文件中
UID	用户号，唯一表示某用户的数字标识。root用户的UID为0，普通用户默认从1000开始顺序编号
GID	用户所属的私有组号，该数字对应group文件中的GID，普通组群的GID默认也从1000开始编号
用户描述信息	可选的 关于用户全名、用户电话等描述性信息
主目录	用户的宿主目录，用户成功登录后的默认目录
命令解释器	用户所使用的shell，默认为“/bin/bash”

2) 用户密码文件——shadow

由于所有用户对/etc/passwd文件均有读取权限，为了增强系统的安全性，用户经过加密之后的口令都存放在/etc/shadow文件中。/etc/shadow文件只对root用户可读，因而大大提高了系统的安全性

shadow文件的内容形式如下（`cat /etc/shadow`）：

```
root:$6$PQxz7W3s$Ra7Akw53/n7rntDgjpNwdCG66/5RZgjhoe1zT2F00ouf2iDM.AVvRIYoez10hGG7kBHEaah.oH5U1t6OQj2Rf.:17654:0:99999:7:::
bin:!:16925:0:99999:7:::
daemon:!:16925:0:99999:7:::
bobby:!!:17656:0:99999:7:::
user1:!!:17656:0:99999:7:::
```

每个用户的信息在shadow文件中占用一行，并且用“:”分隔为9个域，各域含义如表：

字段	说明
1	用户登录名
2	加密后的用户口令, *表示非登录用户, !! 表示没设置密码
3	从1970年1月1日起, 到用户最近一次口令被修改的天数
4	从1970年1月1日起, 到用户可以更改密码的天数, 即最短口令存活期
5	从1970年1月1日起, 到用户必须更改密码的天数, 即最长口令存活期
6	口令过期前几天提醒用户更改口令
7	口令过期后几天账户被禁用
8	口令被禁用的具体日期 (相对日期, 从1970年1月1日至禁用时的天数)
9	保留域, 用于功能扩展

2、用户管理

1) 增加用户

语法:

```
useradd [选项] 用户名
```

功能:

- (1) 在 /etc/passwd 文件中增添了一行记录;
- (2) 在 /home 目录下创建新用户的主目录。

常用选项:

- -g 组名

说明: 用于添加用户账号时指定该用户的主组群(私有组群), 需要组已经存在。如不指定“-g”参数, useradd命令将自动建立与用户账号同名的组作为该账号的主组群(私有组群)。

- -G 组名

指定用户的附属组群(标准组群)。一个用户账户可以同时是多个组群的成员。

- -d 目录

指定用户的HOME目录, HOME目录必须写绝对路径; 如果不指定-d则 默认在/home/用户名

- -c 用户说明

指定/etc/passwd文件中各用户信息中第5个字段的描述性内容

- -p passwd

说明: 为用户账号指定默认密码

- -u uid

说明: 为账户指定一个唯一的UID

- -s shell

指定用户的执行终端，默认是/bin/bash；如果是不可登录系统的账号，可设置为/bin/false或/dev/null；

- -r

说明：创建系统账户，系统账户的UID在1~999之间

注意：

新建的用户暂时还无法登录系统，因为还没有为该用户设置口令，需要再用 passwd 命令为其设置口令后，才能登录。

例：增加一个用户“student2”，然后查看其相关信息

```
[root@localhost ~]# useradd student2 //建立用户账号
[root@localhost ~]# tail /etc/passwd //查看password文件中添加的用户账号信息
[root@localhost ~]# tail /etc/shadow
[root@localhost ~]# ll /home //查看所建立账号的主目录
```

2) 用户口令管理

语法：

```
passwd [选项] [用户名]
```

常用选项：

选项：

- -l：锁定账号的口令。
- -u：解除锁定账号的口令。
- -d：删除指定账号的口令
- -S：查询指定用户账号的状态。

说明：

- 只有超级用户可以使用“passwd [用户名]”修改其他用户的口令；
- 只有超级用户可以使用选项；
- 普通用户只能用不带参数的passwd命令修改自己的口令。

例：修改user1的密码

```
[root@host1 ~]# passwd user1 ← 修改密码
更改用户 user1 的密码 。
新的 密码：
无效的密码： 密码少于 7 个字符
重新输入新的 密码：
passwd: 所有的身份验证令牌已经成功更新。
[root@host1 ~]# tail /etc/shadow
operator:*:17110:0:00000:7:...
```

3) 修改用户属性

语法：

```
usermod [选项] 用户名
```

功能：

修改用户帐号的各种属性。

常用选项：

- -l <新帐号名> 修改用户帐号的名称
- -L 锁定帐户
- -U 解锁帐户
- -g <群组> 修改用户所属的群组
- -d<登入目录> 修改用户登入时的目录。

例：修改用户账号名称，锁定/解锁账号

```
[root@localhost ~]# usermod -l std2 student2 //将账号名'student2'改为'std2'
[root@localhost ~]# usermod -L std2 //锁定“std2”用户，使其不能登录
[root@localhost ~]# usermod -U std2 //解锁“std2”用户账号，使其可以登录
```

4) 删除用户

语法：

userdel [-r] [用户账号]

功能：删除指定的用户账号

说明：

-r 选项可删除与用户账号相关的文件。

若不加选项，则仅删除用户账号，而不删除相关文件。

例：删除用户同时删除其工作主目录

```
[root@localhost ~]# tail /etc/passwd //查看用户账号std2是否存在
[root@localhost ~]# userdel std2 //删除std2账号
[root@localhost ~]# tail /etc/passwd //再次查看用户账号std2是否存在
[root@localhost ~]# ll -d /home //查看用户std2的主目录是否存在
[root@localhost ~]# userdel -r std2 //删除用户同时删除其工作主目录
```

5) 切换用户

语法：

su [选项] 用户名

选项说明：

-: 当前用户不仅切换为指定用户的身份，同时所用的工作环境也切换为此用户的环境（包括 PATH 变量、MAIL 变量等），使用 - 选项可省略用户名，默认会切换为 root 用户。

-l: 同 - 的使用类似，但后面需要添加欲切换的使用者账号。

su命令可以解决切换用户身份的需求，使得当前用户在不退出登录的情况下，顺畅地切换到其他用户，当从root管理员切换到普通用户时是不需要密码验证的，而从普通用户切换到root管理员就需要进行密码验证

例：

```
[root@localhost ~]# su user1
[user1@myvm ~]$ su -
```

从root管理员切换至普通用户
从普通用户切换回root，需要继续输入root密码

6) 显示用户ID

语法:

id [选项] [用户名称]

功能:

显示用户当前的UID, gid以及所属群组的组列表

常用选项:

- g: 显示用户所属群组的ID。
- G: 显示用户所属附加群组的ID。
- n: 显示用户所属群组或附加群组的名称。
- r: 显示实际ID。
- u: 显示用户ID。

7) 显示登录用户名

语法:

whoami

说明: whoami命令用于显示登录者自身的用户名称, 本指令相当于执行“id -un”指令

练习:

1. 创建一个新用户user01
2. 查看/etc/passwd文件的最后一行, 看看是如何记录的。
3. 给用户user01设置密码, 查看文件/etc/shadow文件的最后一行, 看看有什么变化。
4. 使用user01用户登录系统, 看能否登录成功。再次使用root账号登录系统
5. 锁定用户user01, user01不能使用密码登录。查看文件/etc/shadow文件的最后一行, 看看有什么变化。
6. 解除对用户user01的锁定, 查看/etc/shadow的文件变化
7. 更改用户user01的账户名为user02, 查看/etc/passwd文件的最后一行, 看看有什么变化。
8. 删除用户user02

3、组配置文件

1) 组账号文件——group

group文件位于“/etc”目录，用于存放用户的组账户信息，对于该文件的内容任何用户都可以读取。每个组群账户在group文件中占用一行，并且用“:”分隔为4个域。每一行各域的内容如下（使用 `cat /etc/group`）：

组群名称:组群口令（一般为空，用x占位）:GID:组群成员列表

```
user1:x:1000:user1
dept1:x:1001:jack,rose
jack:x:1002:
rose:x:1003:
```

用户的主组群并不把该用户作为成员列出，只有用户的附属群才会列出

2) 组密码文件——gshadow

/etc/gshadow文件用于存放组群的加密口令、组管理员等信息，该文件只有root用户可以读取。每个组群账户在gshadow文件中占用一行，并以“:”分隔为4个域。每一行中各域的内容如下：

组群名称:加密后的组群口令（没有就用!）:组群的管理员:组群成员列表

```
user1:!!:user1
dept1:!!:jack,rose
jack:!:
rose:!:
```

4、组管理

1) 添加/删除组

语法：

`groupadd/groupdel 组名`

例：添加组g1，查看group文件

```
[root@localhost ~]# groupadd g1
[root@localhost ~]# cat /etc/group
```

2) 修改组

语法：

`groupmod [选项] 组名`

常用选项

-g 更改组群的GID

-n <新组名> 更改组群的名称

3) 向组添加/删除用户

使用不带任何参数的useradd命令创建用户时，会同时创建一个和用户账户同名的组群，称为主组群。当一个组群中必须包含多个用户时，则需要使用附属组群。在附属组中增加、删除用户都用gpasswd命令。**注意：只有root用户和组管理员才能够使用这个命令**

语法：

```
gpasswd [选项] [用户] [组]
```

常用选项

-a 把用户加入组

-d 从组中删除用户

-r 取消组的密码

-A 给组指派管理员

例：新建testgroup组中,向组中添加用户user1，并指派user1为管理员

```
[root@localhost ~]# groupadd testgroup
[root@localhost ~]# gpasswd -a user1 testgroup
[root@localhost ~]# gpasswd -A user1 testgroup
```

练习

某公司有5个员工，分别在2个部门工作，每个人工作内容不同。需要在服务器上为每个人创建不同的账号，把相同部门的用户放在一个组中，每个用户都有自己的工作目录。

- 1、新建2个组，分别为renlibu和jishubu
- 2、查看组文件 /etc/group
- 3、新建5个普通用户账号，分别为zhangsan, lisi, wangwu, zhaoliu, qianqi。密码与账户名相同。
- 4、查看账户文件/etc/passwd
- 5、zhangsan与lisi属于renlibu; wangwu、zhaoliu与qianqi属于jishubu。
- 6、将账号加入组群后，查看组文件 /etc/group
- 7、renlibu的管理员是zhangsan; jishubu的管理员是wangwu。

补充：如果在使用su切换用户后，再修改用户属性时，会遇到进程占用的情况

注意：当某个用户登录过后，再更改该用户属性时，会提示该用户有进程正在占用，解决方式一：需要先使用exit命令或ctrl+d登出；解决方式二：使用kill 命令将使用的进程杀掉

```
[root@host1 ~]# su user01 ← 切换用户
[user01@host1 root]$ su root
密码:
[root@host1 ~]# usermod -l user02 user01 ← 再更改用户名会提示有进程占用
usermod: user user01 is currently used by process 1504
[user01@host1 root]$ exit ← 使用exit或ctrl+d 登出2次 再执行更改命
[root@host1 ~]# usermod -l user02 user01
```

```
[root@myvm ~]# usermod -l user02 user1
usermod: user user1 is currently used by process 5500
[root@myvm ~]# kill -9 5500
[root@myvm ~]# 已杀死
```

五、配置与管理Linux文件系统

操作系统利用文件系统保存和管理文件，包括文件建立、写入、读取、修改、转存与控制等。

在Linux中支持多种文件系统，而不同的文件系统会有不同的格式化方式和文件寻址方式。

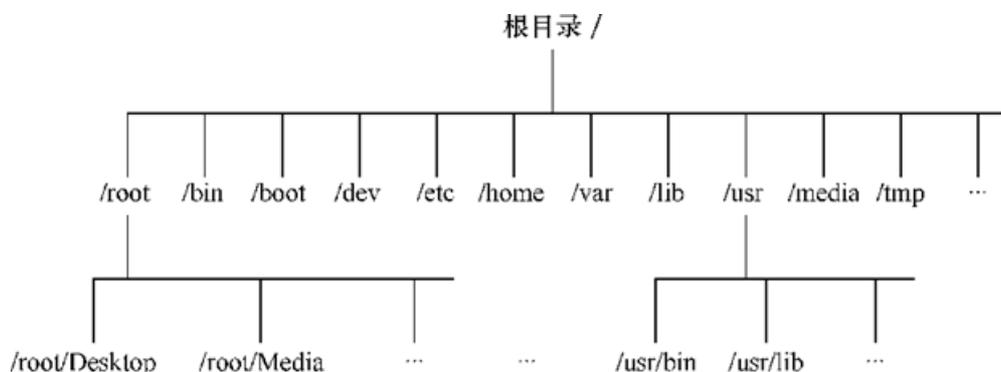
最常见的文件系统如下：

- Ext3：支持日志功能，能够帮助系统从非正常关机导致的异常中恢复
- Ext4：提供了很多新的特性，包括纳秒级时间戳、创建和使用巨型文件(16TB)、最大1EB的文件系统
- xfs：支持最大8EB的文件系统

1、文件系统结构

•Linux系统按照文件系统层次化标准采用树形结构存放文件。

•其顶层为根目录，用符号“/”表示，在根目录下是不同的子目录



常用目录及作用

目录名称	应放置文件的内容
/	Linux文件的最上层根目录
/boot	开机所需文件—内核、开机菜单以及所需配置文件等
/dev	以文件形式存放任何设备与接口
/usr	操作中不需要进行修改的命令程序文件、程序库、手册和其他文档等
/etc	配置文件
/home	用户家目录
/bin	Binary的缩写，存放用户的可运行程序，如ls、cp等，也包含其他shell，如bash和cs等
/lib	开机时用到的函数库，以及/bin与/sbin下面的命令要调用的函数
/sbin	开机过程中需要的命令
/media	用于挂载设备文件的目录
/opt	放置第三方的软件
/root	系统管理员的家目录
/srv	一些网络服务的数据文件目录
/tmp	任何人均可使用的“共享”临时目录
/proc	虚拟文件系统，如系统内核、进程、外部设备及网络状态等
/var	主要存放经常变化的文件，如日志
/mnt	临时文件系统的挂装目录，如U盘，光盘等都可以在这个目录下建立挂载点

2、文件类型

- 文件是操作系统用来存储信息的基本结构。Linux中，目录、字符设备、块设备、套接字、打印机等都被抽象成文件，即一切都是文件。
- 文件类型分7类：

符号	文件类型	符号	文件类型
Directory	目录文件	Character	字符设备文件
-	普通文件	Socket	套接字文件
Link	符号链接文件	Pipe	命名管道文件
Block	块设备文件		

3、访问权限介绍

Ls -l命令查看文件或目录的详细信息如下：

第2-10位表示访问权限



主要使用的权限符、权限及访问能力

权限符	权限	三位权限组的位置	对文件的访问权限	对目录的访问能力
r	可读	第一位置	查看文件内容	查看目录内容（显示子目录、文件列表）
w	可写	第二位置	修改文件内容	创建或删除目录中的任一文件或子目录
x	可执行	第三位置	执行文件（程序和脚本）	执行cd命令进入或退出该目录
-	无权限	任何位置	没有相应权限	没有相应权限

三位一组，分别表示文件所有者、同组用户、其他用户的访问权限

用户分类	英文表示	用户归属
文件所有者	user(u)	建立文件、目录的用户
同组用户	group(g)	属于同一组群的用户对属于该组群的文件有相同的访问权限
其他用户	others(o)	除了文件所有者、同组用户的其他用户。



4、访问权限表示

1. 字符表示法: 如rwxr-xr-x
2. 数字表示法: 授予读取 (r)、写入 (w) 和执行 (x) 分别以数字4、2、1来表示, 没有授权的部分就表示为0, 然后再把所授予的权限相加而成。几个示范的例子如下表

字符表示法	转换为数字	数字表示法 (八进制)
rwxrwxr-x	(421) (421) (401)	775
rwxr-xr-x	(421) (401) (401)	755
rw-rw-r--	(420) (420) (400)	664
rw-r--r--	(420) (400) (400)	644

练习:

根据下图中四个文件或目录的类型、各类用户的访问权限, 并用数字表示法表示他们的权限。将信息填写在下表中。

```
drwxr-xr-x.  40 root  root  1180 2月 27 20:10 run
lrwxrwxrwx.  1 root  root    8 2月  8 16:13 sbin -> usr/sbin
drwxr-xr-x.  2 root  root    6 2月 12 19:56 srv
dr-xr-xr-x. 13 root  root    0 2月 26 20:06 sys
-rw-r--r--.  1 user2 user2  0 2月 27 20:32 test.txt
```

文件/目录名称	文件/目录类型	文件所属用户	文件所属组	权限数字表示法	文字描述访问权限
run	目录	root	root	755	所属用户可读写与执行, 同组用户和其他用户可读可执行

5、文件和目录的权限管理

1) chmod命令

chmod命令: 修改指定文件或目录的访问权限。文件或目录的**所有者和root用户可使用该命令**。

chmod有两种格式, 分别适用于数字表示法、字符表示法。

语法:

```
chmod [-R] 新的权限 文件列表
```

说明:

- -R : 表示递归设置子目录或子文件
- 新权限可以数字或字符表示
- 文件列表可以有多个文件或目录, 用逗号分隔

a、chmod以数字形式改变权限的格式:

r 读权限数字为4

w 写权限数字为2

x 执行权限数字为1

- 无权限数字为0

例: 授以test.txt文件, 所有者和同组成员可读写, 其他成员只能读的权限

```
[root@localhost ~]# chmod 664 test.txt
```

所有者可读写 rw- 为6, 同组成员可读写 rw- 为6 ; 其他成员只读 r-- 为4

b、chmod以字符形式改变权限的格式:

<u>chmod</u>	u g o a	+ (加上) - (减去) = (设定)	r w x	文件或目录
--------------	------------------	----------------------------	-------------	-------

说明:

4种字母来表示不同的用户

- u : user, 表示所有者
- g : group, 表示属组
- o : others, 表示其他用户
- a : all, 表示以上3种用户

操作符号包括以下几种

- + : 添加某种权限
- - : 减去某种权限
- = : 赋予给定权限并取消原来的权限

使用下面3种字符的组合表示法设置操作权限

- r: read, 可读
- w: write, 写入
- x: execute, 执行

例: 使用字符形式授权test.txt文件, 所有者和同组成员可读写, 其他成员只能读的权限

```
[root@localhost ~]# chmod u=rw,g=rw,o=r test.txt
```

例: 新建test.sh文件, 将test.sh文件设置为所有者拥有全部权限, 其他人拥有执行权限。

两种方法:

1. 字符表示法: `chmod u=rwx,g=o,x test.sh`
2. 数字表示法: `chmod 711 test.sh`

例：test.sh文件 给同组成员在原权限基础上加上写的权限

```
chmod g+w test.sh
```

例：1.为新创建的文件aaa.txt设置组内用户可写和可执行的权限，为其他用户添加可写权限。

```
chmod g=wx,o=w aaa.txt
```

2.采用数字形式，将aaa.txt文件权限恢复为“rw-r--r--”。

```
chmod 644 aaa.txt
```

3.将目录/usr/src及其下面的所有子目录和文件的权限改为所有用户对其都有读、写权限。

```
chmod -R a=rw /usr/src
```

chmod练习

1. 在用户user1主目录下创建目录test，进入test目录创建空文件file1。使用ll命令显示文件信息，注意文件的权限和所属用户和组。
2. 对文件file1设置权限，使其他用户可以对此文件进行写操作。并查看设置结果。
3. 取消同组用户对此文件的读取权限。查看设置结果。
4. 用数字形式为文件file1设置权限，所有者可读、可写、可执行；其他用户和所属组用户只有读和执行的权限。设置完成后查看设置结果。
5. 用数字形式更改文件file1的权限，使所有者只能读取此文件，其他任何用户都没有权限。查看设置结果。
6. 为其他用户添加写权限。查看设置结果。
7. 回到上层目录，查看test的权限。
8. 为其他用户添加对此目录的写权限

2) chown命令

chown命令：修改指定文件或目录的所有者或属组

语法：

```
chown [-R] 新属主[: 新属组] 文件或目录
```

说明：

- -R：对目录及其子目录进行递归设置
- 同时设置属主、属组信息时，用：分隔；
- 可只设置属组，单独设置属组时，以：开头

例：将test.txt文件的属主修改为用户zhang

```
chown zhang test.txt
```

例：将aaa.txt文件的属主改为zhang3用户、属组改为li4。

```
chown zhang3:li4 aaa.txt
```

3) chgrp命令

chgrp命令：修改指定文件或目录的属组。

语法：

```
chgrp 新属组 文件或目录
```

例：将test.txt文件的属组修改为用户1

```
chgrp user1 test.txt
```

提示：chown和chgrp命令中所使用的用户和组必须存在于/etc/passwd和/etc/group，否则会报错。

chown练习

查看目录test及其中文件的所属用户组。

把目录test及其下的所有文件的所有者改成bin，所属组改成daemon。查看设置结果。

删除目录test及其下的文件

练习：

- 1、使用root账户登录后新建组：ranjian1, ranjian2；新建用户：lili、liuhu，加入ranjian1组；新建用户：chenhai、daijie，加入ranjian2组；
- 2、进入/tmp目录，新建目录dir1，新建文件test.txt；查看目录dir1，文件test.txt的权限，如何用数字法表示？daijie能否读取和修改test.txt的内容？为什么？请验证。
- 3、修改test.txt的所有者为daijie，所属组为ranjian1，此时，daijie能否读取text.txt的内容？为什么？lili、liuhu、chenhai谁能读取test.txt文件？为什么？
- 4、修改dir1的权限为700，daijie使用cd命令能否进入dir1？为什么？修改dir1的权限为701，daijie使用cd命令能否进入dir1？Daijie使用命令ls能否看到dir1中的内容？为什么？
- 5、修改dir1的权限为705, daijie进入dir1目录后，能否看到dir1中的内容？能否在该目录下新建、删除文件和子目录？
- 6、修改dir1的权限为707，daijie进入dir1目录后，能否在该目录下新建、删除文件和子目录，能否为文件改名？

六、配置与管理磁盘

1、查看磁盘信息

1) df 查看磁盘空间占用

语法:

```
df [参数选项]
```

常用选项:

-a: 显示所有文件系统磁盘使用情况

-T: 显示文件系统类型

-h: 格式化单位显示

物理设备的命名规则:

硬件设备	文件名称
SCSI/SATA/U盘	/dev/sd[a-p]
软驱	/dev/fd[0-1]
打印机	/dev/lp[0-15]
光驱	/dev/sr0
鼠标	/dev/mouse
磁带机	/dev/st0或dev/ht0

2) du显示文件或目录的空间使用情况

语法:

```
du [选项] [文件或目录名]
```

常用选项:

-a 递归显示指定目录下的子目录或子文件的占用情况

-s 只显示指定目录占用的总数

-h 格式化单位显示

2、磁盘管理

1) fdisk 磁盘分区

fdisk用于对磁盘进行管理

a、列出分区表

```
fdisk -l
```

b、磁盘分区

fdisk 磁盘

在command提示后面输入相应的命令来选择需要的操作，command支持的命令如下表：

命令	功能	命令	功能
a	调整硬盘启动分区	q	不保存更改，退出fdisk命令
d	删除硬盘分区	t	更改分区类型
l	列出所有支持的分区类型	u	切换所显示的分区大小的单位
m	列出所有命令	w	把修改写入硬盘分区表，然后退出
n	创建新分区	x	列出高级选项
p	列出硬盘分区表		

2) 主分区、扩展分区和逻辑分区

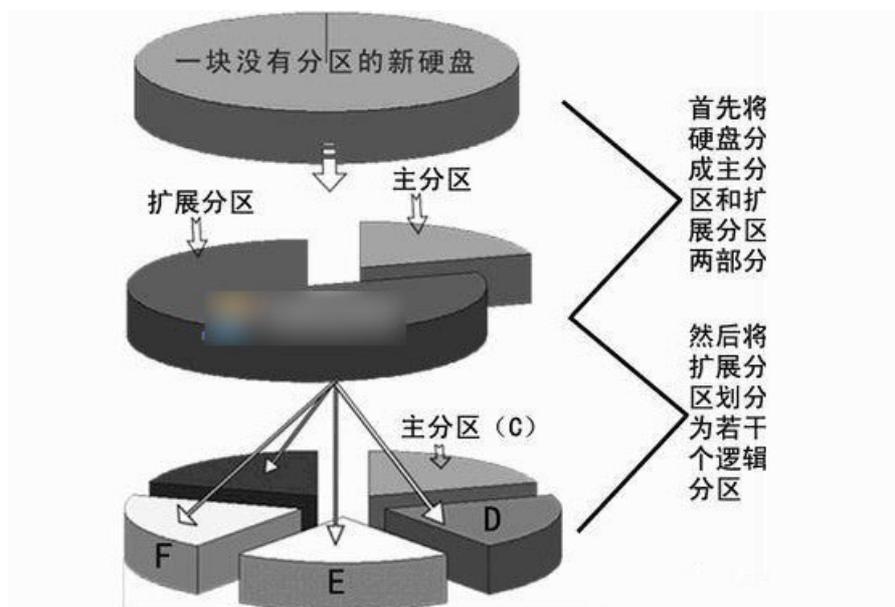
概念

主分区：也叫**引导分区**，**最多能创建4个**，**最少有1个**，当主分区创建了4个后，主分区和扩展分区就没有存在的必要了。

扩展分区：**最多一个**。严格意义上来讲它不是一个真正意义上的分区，它仅仅是一个指向下一个分区的指针，这种指针结构将形成一个单向链表。这样在主引导扇区中除了主分区外，仅需要存储一个被称为扩展分区的分区数据，通过这个扩展分区的数据可以找到下一个分区（逻辑磁盘）的起始位置，以此位置类推可以找到所有的分区。无论系统中建立多少个逻辑磁盘，在主引导扇区中通过一个扩展分区的参数就可以逐个找到每一个逻辑磁盘。

逻辑分区：**在扩展分区上面**，可以创建多个逻辑分区。

https://www.cnblogs.com/itmeatball/p/11469673.html?ivk_sa=1024320u



3) mkfs 建立文件系统

mkfs命令用于建立文件系统，类似于windows下对硬盘格式化。

语法:

```
mkfs [选项] 分区名
```

常用选项: :

-t <文件系统> : 指定要创建的文件系统类型

-c : 建立文件系统前首先检查坏块

-V: 输出建立文件系统详细信息

例: 指定分区sdb1的文件系统为xfs

```
[root@localhost ~]# mkfs -t xfs -c /dev/sdb1
```

4) mount 挂载

在磁盘上建立好文件系统之后, 还需要把建立的文件系统挂载到系统上才能使用。这个过程称为挂载。文件系统所挂载到的目录被称为挂载点。Linux系统中提供了/mnt和/media两个专门的挂载点。一般而言, 挂载点应该是一个空目录, 否则目录中原来的文件将被系统隐藏。

mount命令用于手动挂载文件系统。

语法:

```
mount [选项] 分区名 挂载点
```

常用选项:

-t <文件系统> : 指定要挂载的文件系统的类型

例: 把文件系统为xfs的磁盘分区/dev/sdb1挂载到/newFS目录下

```
[root@localhost ~]# mkdir /newFS          创建空目录
[root@localhost ~]# mount /dev/sdb1 /newFS 挂载到/newFS目录
```

5) umount 卸载

语法:

```
umount 分区名 挂载点
```

6) 自动挂载

如果要想实现每次开机自动挂载文件系统, 可以通过编辑/etc/fstab文件来实现。

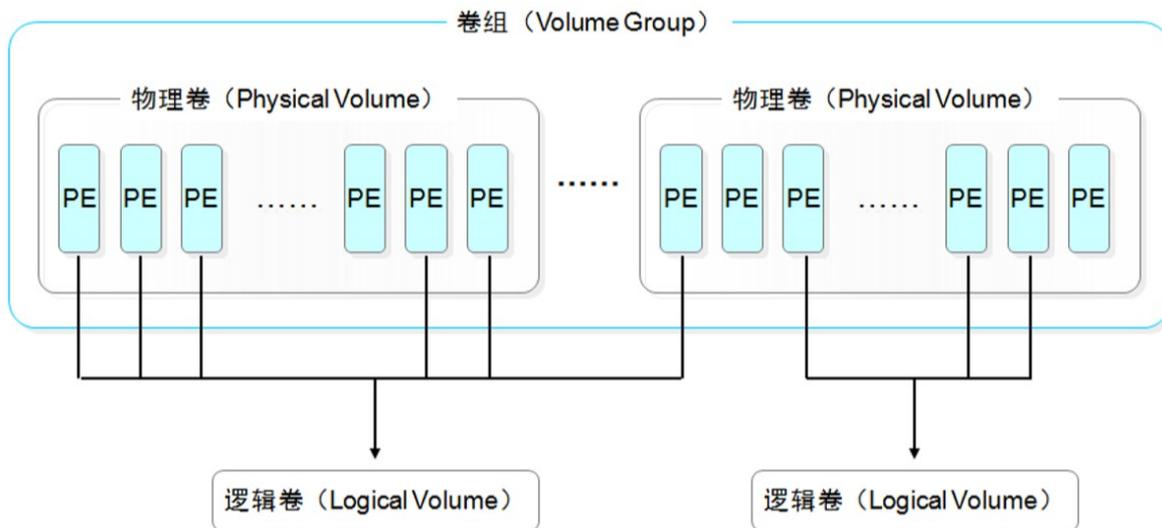
```
# 挂载的设备文件 挂载点 文件系统 挂载选项 是否备份 是否检查磁盘
/dev/mapper/centos-root / xfs defaults 0 0
UUID=5268686e-efd4-4fa9-8823-9c4275e0da11 /boot xfs defaults 0 0
/dev/mapper/centos-swap swap defaults 0 0
```

例: 将/dev/sdb1自动挂载到/newFS目录下

```
[root@localhost ~]# vim /etc/fstab
在fstab文件中输入
/dev/sdb1 /newFS xfs defaults 0 0
保存并退出
```

3、将原硬盘进行扩容

LVM :Logical Volume Manager 逻辑卷管理器，Linux环境下对磁盘分区进行管理的一种机制。LVM允许用户对硬盘资源进行动态调整



LVM技术是在硬盘分区和文件系统之间添加了一个逻辑层，它提供了一个抽象的卷组，可以把多块硬盘进行卷组合并。

思路：

1. 将新硬盘进行分区
2. 让新添加的硬盘设备支持LVM技术
3. 将新硬盘加入到原卷组(centos)中
4. 对根目录对应的分区 (/dev/centos/root) 进行扩容
5. 同步文件系统

使用lsblk、fdisk -l 命令我们发现，挂载点为/的逻辑分区是 /dev/centos/root

步骤如下：

- 1) 在关机状态下，添加新磁盘。将新磁盘进行分区，分区类型为'8e' (linux LVM)

```
[root@linuxvm1 ~]# fdisk /dev/sdb
```

- 2) 让/dev/sdb1分区支持LVM技术

```
[root@linuxvm1 ~]# pvcreate /dev/sdb1
```

- 3) 查看创建结果

```
[root@linuxvm1 ~]# pvdisplay
```

- 4) 将新加入的物理卷/dev/sdb1加入到卷组centos中

```
[root@linuxvm1 ~]# vgextend centos /dev/sdb1
```

```
[root@linuxvm1 ~]# vgextend centos /dev/sdb1
volume group "centos" successfully extended
```

5) 对逻辑卷 (/dev/centos/root) 进行扩容

```
[root@linuxvm1 ~]# lvextend -L +20G /dev/centos/root
```

```
[root@linuxvm1 ~]# lvextend -L +20G /dev/centos/root
Size of logical volume centos/root changed from 10.00 GiB (2560 extents) to 30.00 GiB (7680 extents)
Logical volume centos/root successfully resized.
```

6) 同步文件系统

```
[root@linuxvm1 ~]# xfs_growfs /dev/mapper/centos-root
```

```
[root@linuxvm1 ~]# xfs_growfs /dev/mapper/centos-root
meta-data=/dev/mapper/centos-root isize=512    agcount=4, agsize=655360 blks
         =                               sectsz=512   attr=2, projid32bit=1
         =                               crc=1      finobt=0 spinodes=0
data     =                               bsize=4096 blocks=2621440, imaxpct=25
         =                               sunit=0    swidth=0 blks
naming   =version 2                       bsize=4096 ascii-ci=0 ftype=1
log      =internal                        bsize=4096 blocks=2560, version=2
         =                               sectsz=512 sunit=0 blks, lazy-count=1
realtime =none                             extsz=4096 blocks=0, rtextents=0
data blocks changed from 2621440 to 7864320
```

7) 查看磁盘空间占用

```
[root@linuxvm1 ~]# df -hT
文件系统          类型          容量  已用  可用  已用% 挂载点
/dev/mapper/centos-root xfs          30G   4.6G   26G   16% /
```

如果遇到下列问题:

在给逻辑卷root进行扩容的时候, 我碰到了如下的报错

```
1  WARNING: Device for PV 67FxpU-S9ou-yyc0-21og-NoIc-W3I1-mcN1b5 not found or rejected
2  Couldn't find device with uuid 67FxpU-S9ou-yyc0-21og-NoIc-W3I1-mcN1b5.
3  Cannot change VG centos while PVs are missing.
4  Consider vgreduce --removemissing.
5  Cannot process volume group centos
6
```

可以使用以下命令进行解决

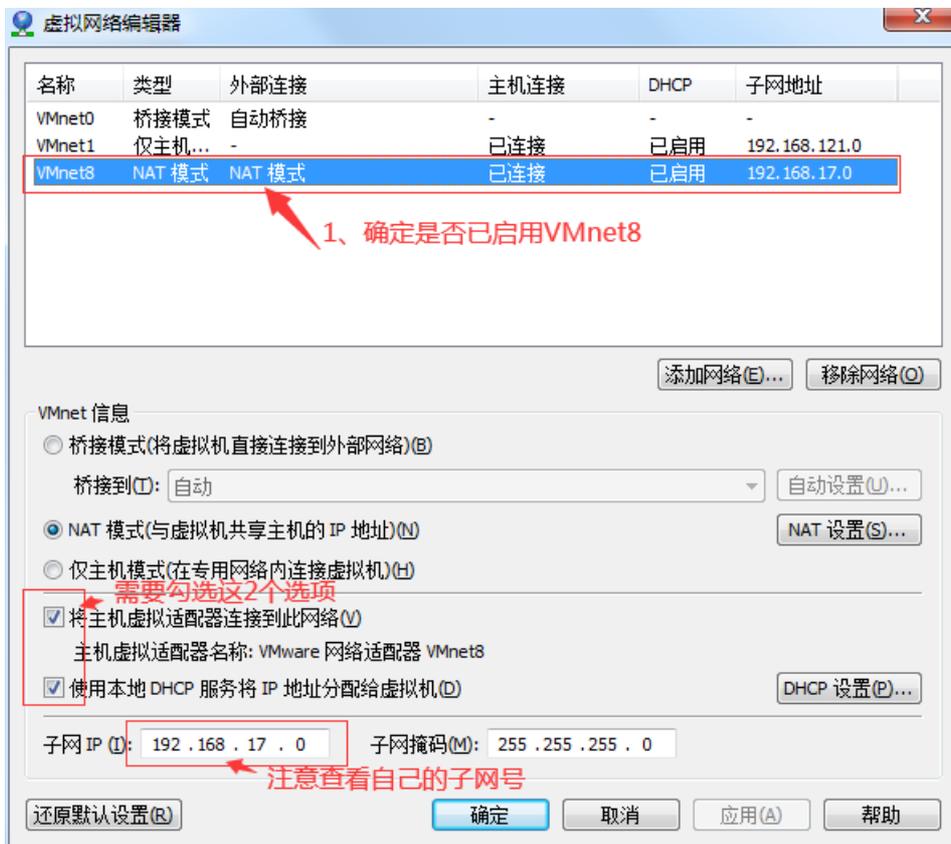
```
vgreduce --removemissing -v centos
```

执行后重新使用lvextend扩容

七、网络配置

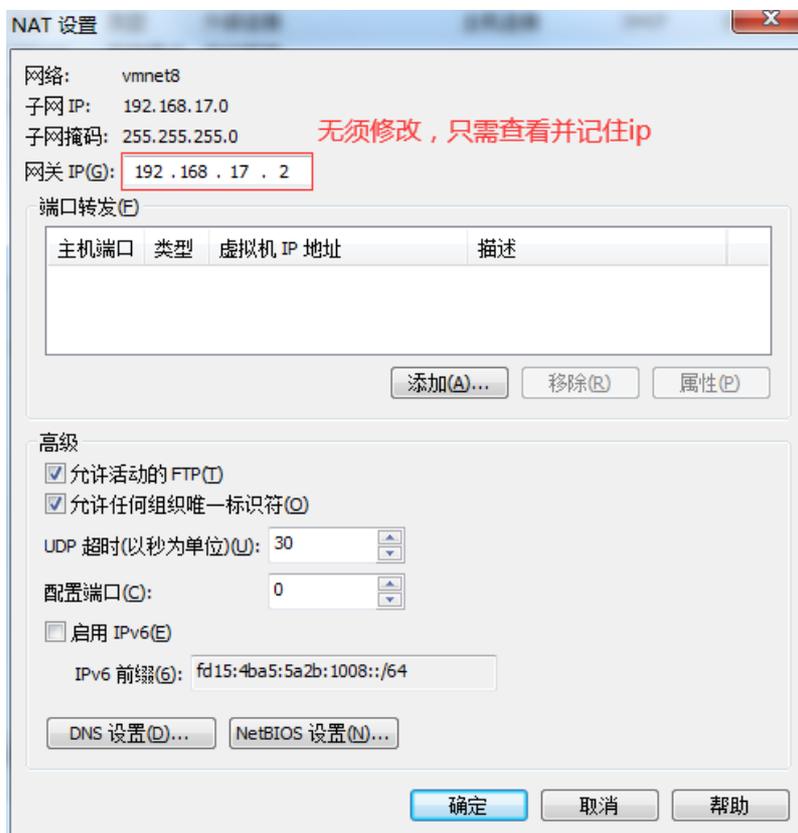
1、确定虚拟机连网方式

1) 打开VMware的菜单“编辑”=>“虚拟网络编辑器”。观察网络编辑器内容。子网IP号格式一般是:
192.168.xx.0。



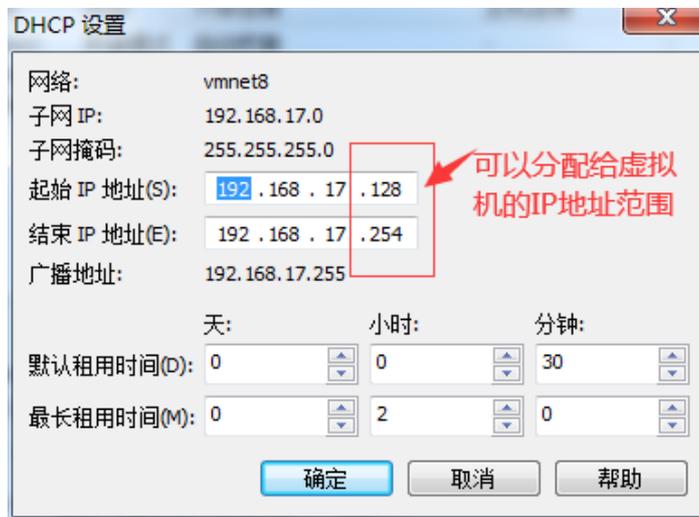
2) 查看网关IP地址

单击“虚拟网络编辑器”对话框中的“NAT设置”按钮，查看网关IP地址，网关IP地址格式一般是 192.168.xx.2



3) 查看虚拟机可分配的IP地址范围

单击“虚拟网络编辑器”对话框中的“DHCP设置”按钮，查看ip地址的范围。给虚拟机分配IP时，注意不要超出这个范围。



2、查看虚拟机网络配置情况

1) ifconfig命令

ifconfig用于显示和设置当前网络设备的状态

语法:

```
ifconfig [-a] [网络接口名称] [address]
```

```
[root@xiayan ~]#  
[root@xiayan ~]#  
[root@xiayan ~]# ifconfig ens33  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
  IP地址 inet 192.168.48.7 netmask 255.255.255.0 broadcast 192.168.48.255  
  inet6 fe80::3ab8:991b:a38a:e6bd prefixlen 64 scopeid 0x20<link>  
  ether 00:0c:29:4f:79:cd txqueuelen 1000 (Ethernet)  
  ipv6地址 RX packets 10309 bytes 709056 (692.4 KiB)  
            RX errors 0 dropped 0 overruns 0 frame 0  
            TX packets 991 bytes 122614 (119.7 KiB)  
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

选项说明:

- a: 输出所有活动和非活动网络接口的信息

网络接口名称:

- ens33: 第一块以太网卡的名称
- lo: local的简写, 一般指本地环回接口
- virbr0 是VMware默认创建的一个虚拟网桥, 其作用是为连接其上的虚拟机网卡提供 NAT 访问外网的功能。默认分配了一个IP 192.168.122.1, 并为其他虚拟网卡提供 DHCP 服务。

ens33接口说明:

- inet: 是虚拟机的ip地址
- netmask: 是子网掩码
- booadcast: 广播地址
- ether: 网卡MAC地址

2) ip命令

语法:

```
ip [动作] [命令]
```

常用ip命令:

```
ip link set ens33 up # 开启网卡
ip link set ens33 down # 关闭网卡
ip addr # 显示网卡IP信息
ip a # addr可简写成a
ip addr add 192.168.0.1/24 dev ens33 # 添加ens33网卡IP地址192.168.0.1
ip addr del 192.168.0.1/24 dev ens33 # 删除ens33网卡IP地址
ip route # 显示系统路由
ip route add default via 192.168.1.254 # 添加系统默认路由
```

```
[root@myvm ~]# ip route
default via 192.168.17.2 dev ens33 proto static metric 100
192.168.17.0/24 dev ens33 proto kernel scope link src 192.168.17.130 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
```

3、通过配置文件配置网络

在centos7中，网卡配置文件的前缀以ifcfg开始，加上网卡名称 共同组成了网卡配置文件的名字。如ifcfg-ens33。该文件完整路径是 /etc/sysconfig/network-scripts/ifcfg-ens33

文件内容说明如下:

```
TYPE=Ethernet # 设备类型
BOOTPROTO=dhcp # 地址分配模式，dhcp模式是动态分配ip地址，有时需要改为static固定ip
NAME=ens33 # 网卡名称
ONBOOT=yes # 是否启动
```

如果需要通过网络远程访问该系统，则虚拟机需要将ip固定。所以ifcfg-ens33文件需要更改为以下内容

```
TYPE=Ethernet # 设备类型
BOOTPROTO=static # 地址分配模式，需要改为static固定ip
NAME=ens33 # 网卡名称
ONBOOT=yes # 是否启动
IPADDR=192.168.xxx.xxx # ip地址,这里第一个xxx需要改为自己的网段，第二个xxx在128~254之间
NETMASK=255.255.255.0 # 子网掩码
GATEWAY=192.168.xxx.2 # 网关地址，xxx需要改为自己的网段
DNS1=8.8.8.8 # DNS地址
```

例：通过前面查看“虚拟网络编辑器”，子网号为192.168.17.0，更改配置文件如下

```
TYPE=Ethernet
BOOTPROTO=static
NAME=ens33
ONBOOT=yes
IPADDR=192.168.17.131 # 17是网段，130是在128~254之间
NETMASK=255.255.255.0
GATEWAY=192.168.17.2
DNS1=8.8.8.8
```

注意：修改完配置文件后，需重启网络服务，再使用ping命令测试网络能否联通

```
[root@localhost ~]# systemctl restart network
[root@localhost ~]# ping baidu.com
```

如果重启网络服务过程中，遇到如下提示

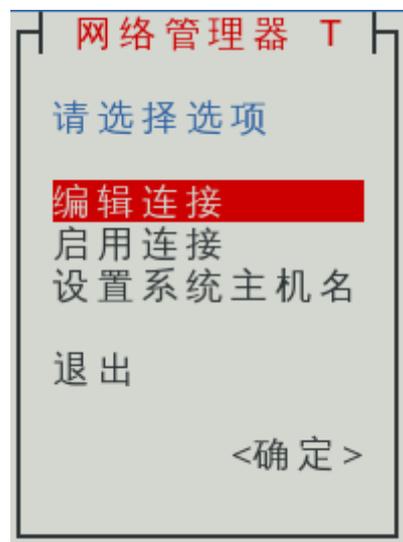
```
[root@myvm ~]# systemctl restart network
Job for network.service failed because the control process exited with error code. See "systemctl status network.service" and "journalctl -xe" for details.
```

有可能是重启网络服务时与网络设置的守护进程相冲突，可先关闭该进程服务，再执行重启命令。

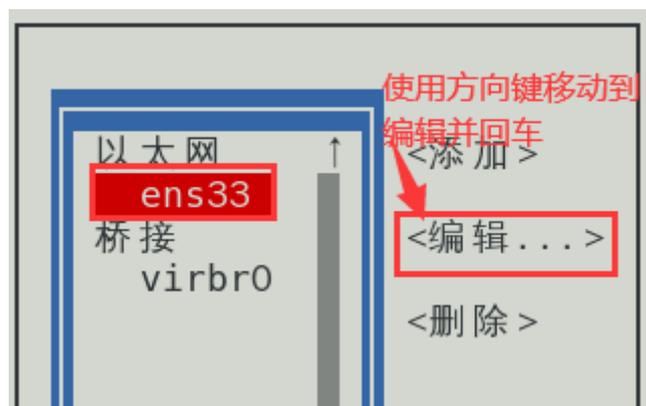
```
[root@localhost ~]# systemctl stop NetworkManager
```

4、通过nmtui命令配置网络

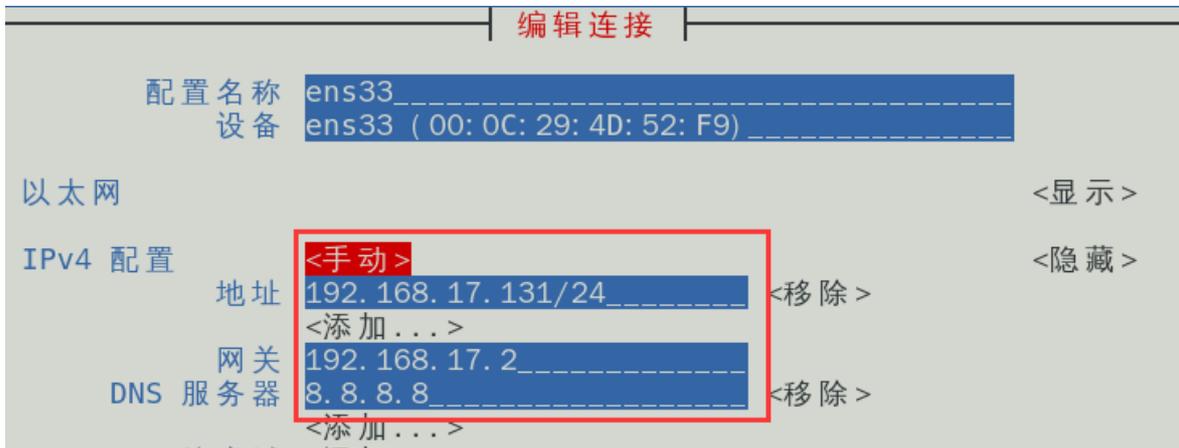
1) 在终端中输入 `nmtui` 后进入如下界面：



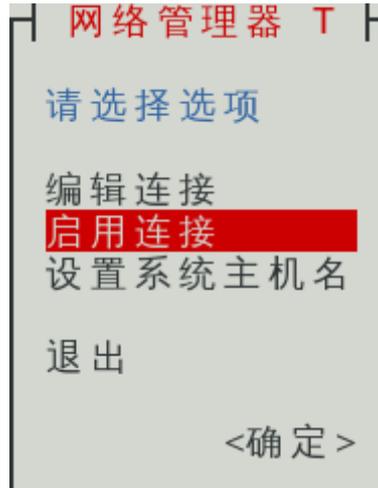
2) 选择编辑连接并回车，进入如下界面：



3) 使用方向键，移动到"<编辑...>"并回车。分别修改IPv4为手动，输入ip地址、网关和DNS服务器，编辑完成后，使用方向键移动到下面的"确定"并回车

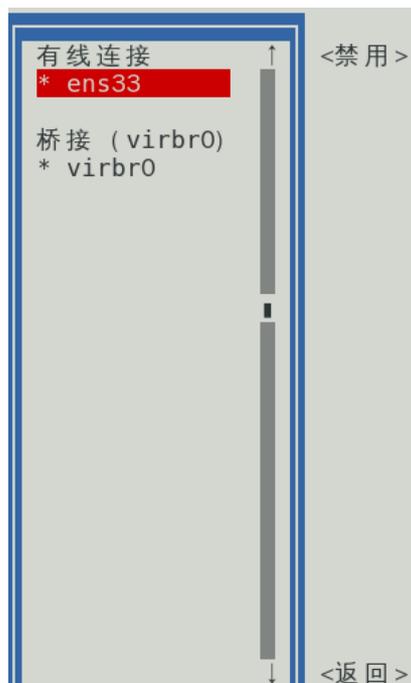


4) 回到网络管理器，选择"启用连接"并回车

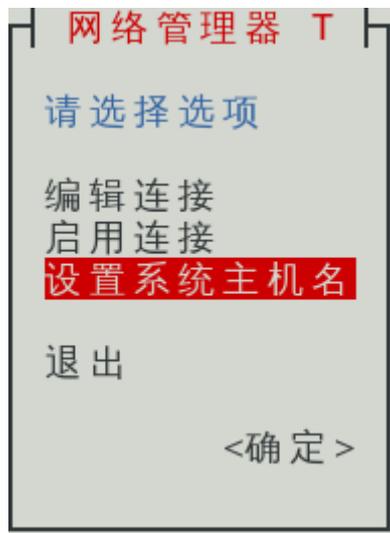


5) 选择有线连接下的ens33。按下2次回车，重启网络配置服务，注意：需保证ens33前有"*"号，有"*"号才是开始状态。

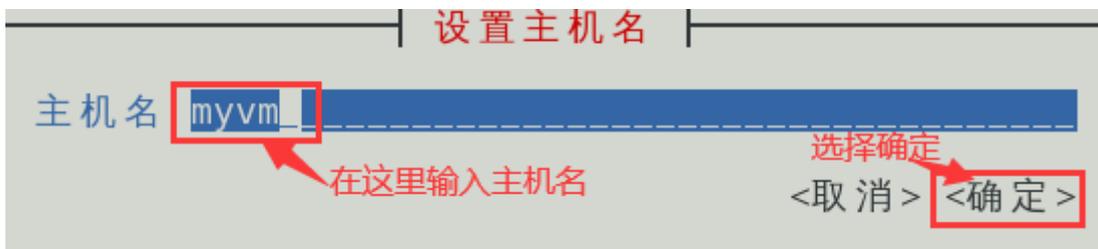
使用方向键移动到右侧的'返回'并回车



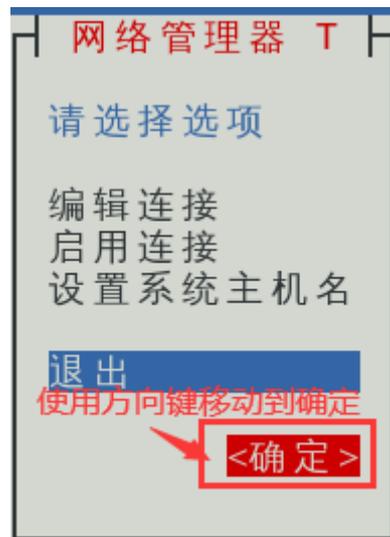
6) 在nmtui命令下也可以修改主机名称。回到网络管理器，选择"设置系统主机名"并回车



7) 修改主机名，并确定



8) 回到网络管理器，使用方向键移动到确定，并回车保存上述修改。



5、配置主机名与ip映射

IP地址实在是难以记忆，有没有什么办法可以通过主机名或替代的字符地址去代替数字化的IP地址呢？实际上，我们一直都是通过字符化的地址去访问服务器，很少指定IP地址。这一切，都是域名解析帮助我们解决的。

域名解析

访问www.baidu.com的流程如下:



即:

- 先查看本机的记录 (私人地址本)
 - Windows看: C:\Windows\System32\drivers\etc\hosts
 - Linux看: /etc/hosts
- 再联网去DNS服务器 (如114.114.114.114, 8.8.8.8等) 询问

修改虚拟主机上的/etc/hosts文件, 将ip地址与主机名相绑定, 即可达到目的。

1) 使用vim命令打开/etc/hosts文件

```
[root@localhost ~]# vim /etc/hosts
```

2) 在文件中添加ip地址与主机名

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.19.131 linuxvm1
192.168.19.132 linuxvm2
```

更改为自己的ip地址和主机名
这里是2个主机的ip和主机名

3) 使用ping命令, 验证是否绑定成功

```
[root@localhost ~]# ping linuxvm2
```

6、配置远程控制服务

1) 配置sshd服务

SSH(Secure shell) 是一种能够以安全的方式提供远程登录的协议。想要使用SSH协议来远程管理Linux系统, 则需要部署配置sshd服务程序。

sshd提供了两种安全验证的方法

基于口令的验证——用账户和密码来验证登录

基于密钥的验证——需要在本地生成密钥对, 然后把密钥对中的公钥上传到服务器, 并与服务器中的公钥进行比较。

在生产环境中使用密码进行口令存在着被暴力破解或嗅探截获的风险。如果正确配置了密钥验证方式, 那么sshd服务将更加安全。

密钥验证步骤:

a、生成密钥对

```
[root@localhost ~]# ssh-keygen
```

```
[root@linuxvm1 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:owqDk6/b+7HW2jAcfdzKBSRq3R15+2Qw5CdbzAO/Cm0 root@linuxvm1
The key's randomart image is:
+---[RSA 2048]---+
  . . . +.
  o + ..o*o
  o o o ..o*=
  . o o o . .==
  o S o.oE.=
  o o . o oo . .
  + + =. o .
  = ..*
  oo+++...
+---[SHA256]---+
```

b、将生成的公钥文件传送至远程主机

```
[root@localhost ~]# ssh-copy-id linuxvm2 // 这里的linuxvm2是第二台主机的
主机名
```

注意：操作在A主机上执行，则A主机在登录远程主机B时，无须再输入密码。如果B主机也想免密登录A主机，则B主机上也需进行上述操作。

c、通过ssh命令登录远程主机，**远程登录后，所做操作都在远程主机上。**

```
[root@localhost ~]# ssh linuxvm2
```

d、可通过scp命令在2个主机间传输文件

语法：

```
scp [参数] 文件路径 文件路径
```

用法一：将当前目录下test.txt文件复制到远程主机192.168.19.132的/tmp目录下

```
[root@localhost ~]# scp test.txt linuxvm2:/tmp
```

用法二：将/export/servers目录下的所有内容复制到远程主机/tmp目录下

```
[root@localhost ~]# scp -r /export/servers linuxvm2:/tmp
```

用法三：将远程主机/tmp/file1文件复制到本地的/home目录下

```
[root@localhost ~]# scp linuxvm2:/tmp/file1 /home/
```

八、配置FTP服务器

以HTTP为基础的www服务功能虽然强大，但对于文件传输来说却略显不足。一种专门用于文件传输的FTP服务应运而生。

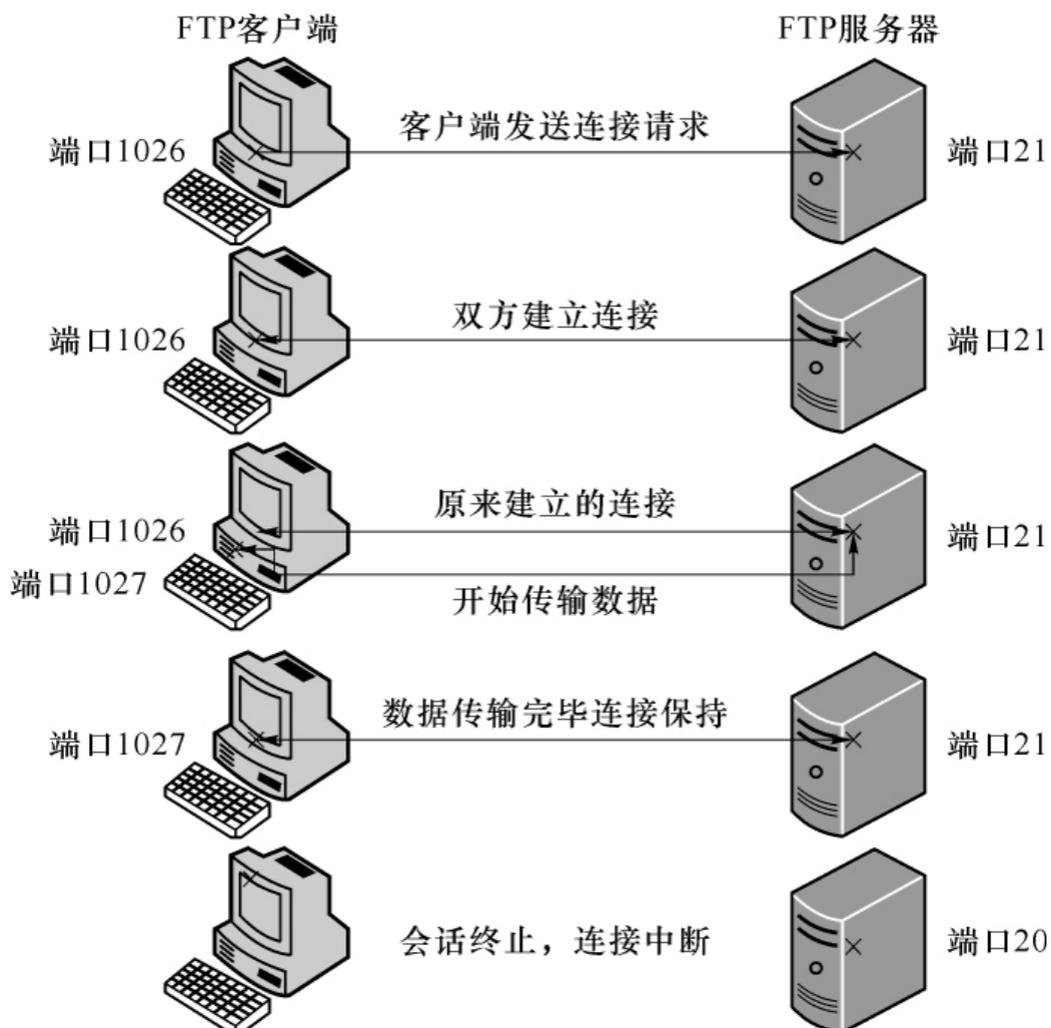
FTP服务就是文件传输服务，FTP的全称是File Transfer Protocol，文件传输协议，具备更强的文件传输可靠性和更高的效率。

1、ftp工作原理

ftp大大简化了文件传输的复杂性，它能够使文件通过网络从一台主机传递到另一台计算机上却不受计算机和操作系统类型的限制。只要双方都支持ftp协议，就可以方便可靠地进行文件的传送。

具体工作过程如下：

- 1) 客户端向服务器发出连接请求，同时客户端系统动态地打开一个大于1024的端口等候服务器连接（如1026端口）
- 2) 若ftp服务器在端口21侦听到该请求，则会在客户端1026端口和服务器的21端口之间建立起一个ftp会话连接。
- 3) 当需要传输数据时，ftp客户端再动态地打开一个大于1024的端口（如1027端口）连接到服务器的20端口，并在这两个端口之间进行数据的传输。当数据传输完毕，这两个端口会自动关闭。
- 4) 当ftp客户端断开与ftp服务器的连接时，客户端上动态分配的端口将自动释放。



ftp服务有两种工作模式：主动传输模式（Active FTP）和被动传输模式（Passive FTP）

ftp的匿名用户：

ftp服务不同于www，它首先要求登录到服务器上，然后再进行文件的传输。于是匿名用户访问就诞生了：通过使用一个共同的用户名anonymous，密码不限的管理策略让任何用户都可以很方便地从ftp服务器上下载软件。

2、安装和配置文件

安装步骤如下：

- 1) 利用rpm命令查询是否安装vsftpd服务

```
[root@localhost ~]# rpm -q vsftpd
```

- 2) 安装前先清除缓存

```
[root@localhost ~]# yum clean all
```

- 3) 安装服务

```
[root@localhost ~]# yum -y install vsftpd
```

- 4) 安装ftp软件包

```
[root@localhost ~]# yum -y install ftp
```

- 5) 检查安装组件是否成功

```
[root@localhost ~]# rpm -q vsftpd
```

```
[root@linuxvm1 ~]# rpm -qa vsftpd
vsftpd-3.0.2-29.el7_9.x86_64
```

- 6) 重新启动vsftpd服务，并且随系统启动

```
[root@localhost ~]# systemctl restart vsftpd // 重启服务
[root@localhost ~]# systemctl enable vsftpd // 服务随系统启动
```

- 7) 将ftp加入防火墙放行策略，并刷新

```
[root@localhost ~]# firewall-cmd --permanent --add-service=ftp
[root@localhost ~]# firewall-cmd --reload // 刷新防火墙
```

- 8) 开放SELinux

```
[root@localhost ~]# setenfore 0 setsebool -P ftpd_full_access=on
```

认识vsftpd的配置文件

- 1) 主配置文件

/etc/vsftpd/vsftpd.conf 内容长达127行，大部分是注释信息。可以使用grep命令添加-v参数，过滤并反选出没有包含#的内容行。注意：先将配置文件备份

```
[root@localhost ~]# mv /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.bak
[root@localhost ~]# grep -v '#' /etc/vsftpd/vsftpd.conf.bak >
/etc/vsftpd/vsftpd.conf
[root@localhost ~]# cat -n /etc/vsftpd/vsftpd.conf
```

```
[root@linuxvm1 ~]# cat -n /etc/vsftpd/vsftpd.conf
 1  anonymous_enable=YES
 2  local_enable=YES
 3  write_enable=YES
 4  local_umask=022
 5  dirmessage_enable=YES
 6  xferlog_enable=YES
 7  connect_from_port_20=YES
 8  xferlog_std_format=YES
 9  listen=NO
10  listen_ipv6=YES
11
12  pam_service_name=vsftpd
13  userlist_enable=YES
14  tcp_wrappers=YES
```

常用参数及作用如下：

参数	作用
download_enable = [YES NO]	是否允许下载文件
userlist_enable=[YES NO] userlist_deny=[YES NO]	设置用户列表为“允许”还是“禁止”操作
anonymous_enable=[YES NO]	是否允许匿名用户访问
anon_upload_enable=[YES NO]	是否允许匿名用户上传文件
anon_root=/var/ftp	匿名用户的FTP根目录
anon_mkdir_write_enable=[YES NO]	是否允许匿名用户创建目录
anon_other_write_enable=[YES NO]	是否开放匿名用户的写入权限（包括重命名、删除等操作权限）
local_enable=[YES NO]	是否允许本地用户登录FTP
local_root=/var/ftp	本地用户的FTP根目录
chroot_local_user=[YES NO]	是否将用户权限禁锢在FTP目录，以确保安全

2) /etc/pam.d/vsftpd

vsftpd的Pluggable Authentication Modules (PAM) 配置文件，主要用来加强vsftpd服务器的用户认证。

3) /etc/vsftpd/ftputers

所有位于此文件内的用户都不能访问vsftpd服务。当然，为了安全起见，这个文件中默认已经包括了root、bin和daemon等系统账号。

4) /etc/vsftpd/user_list

当userlist_deny=NO时, 仅允许文件列表中的用户访问FTP服务器。

当userlist_deny=YES 时, 这也是默认值, 拒绝文件列表中的用户访问FTP服务器

5) /var/ftp文件夹

vsftpd提供服务的文件集散地, 它包括一个pub子目录。在默认配置下, 所有的目录都是只读的, 不过只有root用户有写权限。

3、配置匿名用户FTP实例

a、匿名开放模式

允许匿名用户上传和下载文件, 匿名用户的访问目录设置为/var/ftp

1) 新建测试文件

```
[root@localhost ~]# echo 'hello ftp'>a.txt
```

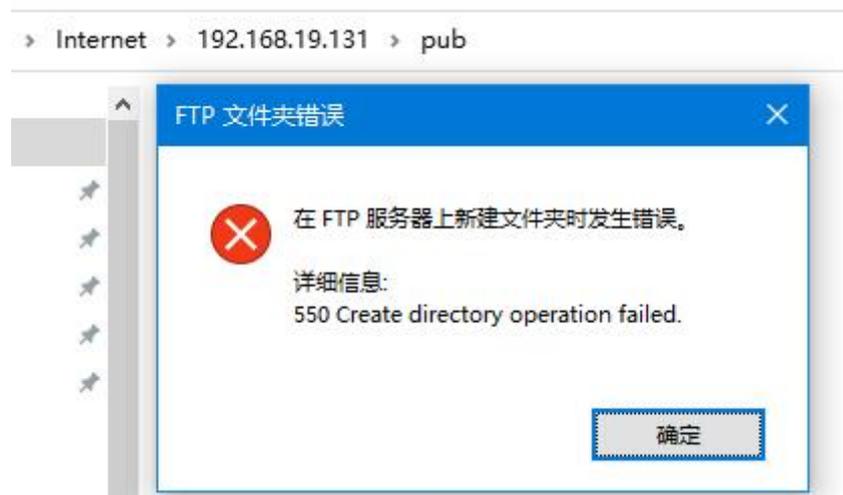
2) 编辑配置文件, 修改或添加代码

```
[root@localhost ~]# vim /etc/vsftpd/vsftpd.conf  
文件中修改或添加如下代码  
anonymous_enable=YES  
anon_root=/var/ftp  
anon_upload_enable=YES  
anon_mkdir_write_enable=YES  
anon_other_write_enable=YES
```

3) 允许SELinux, 让防火墙放行ftp, 重启vsftpd服务

```
[root@localhost ~]# setenforce 0  
[root@localhost ~]# systemctl restart vsftpd
```

4) 在windows客户端中的资源管理器中输入 ftp://192.168.19.131 (改为自己主机的ip地址), 打开pub目录, 可以看到a.txt, 这时可以从服务器下载该文件。但新建文件夹时, 结果出错了。



5) 设置本地系统权限, 将属主设为ftp, 或对pub目录赋予其他用户写的权限。并重启vsftpd服务

```
[root@localhost ~]# chown ftp /var/ftp/pub
[root@localhost ~]# systemctl restart vsftpd
```

4、配置本地用户FTP实例

需求分析:

为了增强安全性, 首先需要使用仅允许本地用户访问, 并禁止匿名用户登录。其次, 使用chroot功能将账户team1限制在/web/www/html目录下。其他用户不受限。

1) 建立FTP账号并禁止本地登录, 设置密码

```
[root@localhost ~]# useradd -s /sbin/nologin team1
[root@localhost ~]# useradd -s /sbin/nologin user2
[root@localhost ~]# passwd team1
[root@localhost ~]# passwd user2
```

2) 配置vsftpd.conf主配置文件并做相应修改

```
[root@localhost ~]# vim /etc/vsftpd/vsftpd.conf
# 文件中修改或添加如下代码:
anonymous_enable=NO
local_enable=YES
local_root=/web/www/html
chroot_local_user=NO    # 例外列表内用户受限制, 不可自由转换目录
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
allow_writeable_chroot=YES
```

3) 建立/etc/vsftpd/chroot_list文件, 添加受限制的账号

```
[root@localhost ~]# vim /etc/vsftpd/chroot_list
# 文件内添加受限制的账号:
team1
```

4) 设置SELinux允许, 重启vsftpd服务

```
[root@localhost ~]# setenforce 0
[root@localhost ~]# systemctl restart vsftpd
```

5) 创建测试目录和文件, 修改本地权限

```
[root@localhost ~]# mkdir -p /web/www/html
[root@localhost ~]# echo 'hello ftp'>/web/www/html/a.txt
[root@localhost ~]# ll -d /web/www/html
[root@localhost ~]# chmod -R o+w /web/www/html    // 其他用户可以写入
```

6) 另一主机做为客户端, 先安装ftp工具(如果已安装, 则这步可不做)

```
[root@localhost ~]# yum clean all
[root@localhost ~]# yum -y install ftp
```

7) 使用不同用户进行测试

- 使用team1用户不能更换目录，可以建立新文件夹，显示的目录是"/"，实际是/web/www/html

```
[root@linuxvm1 ~]# ftp linuxvm2
Connected to linuxvm2 (192.168.19.132).
220 (vsFTPd 3.0.2)
Name (linuxvm2:root): team1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,19,132,245,67).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 10 Mar 12 10:15 a.txt
226 Directory send OK.
ftp> pwd
257 "/"
ftp> mkdir test
257 "/test" created
ftp> ls
227 Entering Passive Mode (192,168,19,132,187,76).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 10 Mar 12 10:15 a.txt
drwxr-xr-x 2 1005 1007 6 Mar 12 10:47 test
226 Directory send OK.
ftp> cd /etc/
550 Failed to change directory
ftp> exit
221 Goodbye.
```

- 使用user2用户，能自己更换目录，可以将/etc/passwd文件下载到/root目录

```
[root@linuxvm1 ~]# ftp linuxvm2
Connected to linuxvm2 (192.168.19.132).
220 (vsFTPd 3.0.2)
Name (linuxvm2:root): user2
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,19,132,184,150).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 10 Mar 12 10:15 a.txt
drwxr-xr-x 2 1005 1007 6 Mar 12 10:47 test
226 Directory send OK.
ftp> pwd
257 "/web/www/html"
ftp> mkdir test2
257 "/web/www/html/test2" created
ftp> ls
227 Entering Passive Mode (192,168,19,132,217,9).
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 10 Mar 12 10:15 a.txt
drwxr-xr-x 2 1005 1007 6 Mar 12 10:47 test
drwxr-xr-x 2 1006 1008 6 Mar 12 10:56 test2
226 Directory send OK.
ftp> cd /etc/
250 Directory successfully changed.
ftp> pwd
257 "/etc"
ftp> get passwd
local: passwd remote: passwd
227 Entering Passive Mode (192,168,19,132,119,94).
150 Opening BINARY mode data connection for passwd (2365 bytes).
226 Transfer complete.
2365 bytes received in 0.000188 secs (12579.79 Kbytes/sec)
ftp> exit
```

说明：

chroot是靠例外列表来实现的，列表用户即是例外的用户。

如果配置文件中启用 `chroot_local_user=NO`，表示除列表内用户外，其他用户都可自由切换目录。即列表内用户受限，其他用户自由。

如果配置文件中启用 `chroot_local_user=YES`，表示除列表内用户外，其他用户都被限定在固定目录内。即列表内用户自由，其他用户受限。为了安全，建议使用这一种。

九、MySQL数据库的安装

1、安装MySQL

1) 安装YUM Repo

由于CentOS的yum源中没有mysql。需要到mysql的官网下载yum repo配置文件

```
[root@localhost ~]# wget https://dev.mysql.com/get/mysql57-community-release-e17-9.noarch.rpm
```

2) 进行repo的安装

```
[root@localhost ~]# rpm -ivh mysql57-community-release-e17-9.noarch.rpm
```

执行完成后会在/etc/yum.repos.d/目录下生成两个repo文件mysql-community.repo mysql-community-source.repo

3) 使用yum命令完成安装

注意：必须进入到 /etc/yum.repos.d/目录后再执行以下脚本

```
[root@localhost yum.repos.d]# yum install mysql-server
```

如果安装过程中提示下图，则安装失败：

```
mysql-community-libs-compat-5.7.37-1.e17.x86_64.rpm 的公钥尚未安装
```

```
失败的软件包是：mysql-community-libs-compat-5.7.37-1.e17.x86_64  
GPG 密钥配置为：file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

如果安装失败，则执行：

```
[root@localhost yum.repos.d]# vim mysql-community.repo
```

修改**对应安装版本**的gpgcheck=0即可，默认值为1

```
[mysql57-community]  
name=MySQL 5.7 Community Server  
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/e1/7/$basearch/  
enabled=1  
gpgcheck=0 # 修改这一行，注意mysql的版本  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

再次执行yum命令安装

```
[root@localhost yum.repos.d]# yum install mysql-server
```

安装成功会提示：

Complete!

4) 启动mysql服务

```
[root@localhost ~]# systemctl start mysqld
```

2、管理MySQL

1) 获取安装时的临时密码

```
[root@localhost ~]# grep 'temporary password' /var/log/mysqld.log
```

```
[root@Client1 yum.repos.d]# grep 'temporary password' /var/log/mysqld.log  
2021-06-22T08:42:33.886379Z 1 [Note] A temporary password is generated for root@localhost: Cj1fuB=Lx0qu
```

注意：密码不一定与框起来的内容相同

倘若没有获取临时密码，则先停止服务，删除原来安装过的mysql残留的数据，再启动mysql（该步骤不一定执行）

```
[root@localhost ~]# systemctl stop mysqld  
[root@localhost ~]# rm -rf /var/lib/mysql  
[root@localhost ~]# systemctl start mysqld
```

2) 使用密码登录

```
[root@localhost ~]# mysql -u root -p
```

输入上一步获取的密码

```
[root@linuxvm1 yum.repos.d]# mysql -u root -p  
Enter password:  
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)  
[root@linuxvm1 yum.repos.d]# mysql -u root -p  
Enter password:   
welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 6  
Server version: 5.7.41  
  
Copyright (c) 2000, 2023, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

3) 修改密码

a、修改密码语法

```
mysql> set password for root@localhost=password('123456');
```

会报以下错误：你的密码不满足当前的必要的条件

```
mysql> set password for root@localhost=password('123456');  
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

b、需要更改当前mysql的密码强度等级

```
mysql> set global validate_password_policy=LOW;
```

关于 validate_password_policy 的取值:

LOW: 只验证长度;

MEDIUM: 验证长度、数字、大小写、特殊字符;

STRONG: 验证长度、数字、大小写、特殊字符、字典文件;

c、更改密码长度

默认密码长度为8, 可以不用修改。如果设置6位密码, 则执行下列命令

```
mysql> set global validate_password_length=6;
```

更改强度等级和密码长度后, 再次执行修改密码命令。

4) 开启远程控制

MySQL默认是没有开启远程控制的, 必须添加远程访问的用户, 即默认是只能自己访问, 别的机器是访问不了的。

a、连接服务器:

```
[root@localhost ~]# mysql -u root -p  
输入密码
```

b、看当前所有数据库:

```
mysql> show databases;
```

c、进入mysql数据库:

```
mysql> use mysql;
```

d、查看mysql数据库中所有的表:

```
mysql> show tables;
```

e、查看user表中的数据:

```
mysql> select host,user,authentication_string from user;
```

f、修改user表中的Host:

```
mysql> update user set Host='%' where User='root';
```

g、刷新:

```
mysql> flush privileges;
```

5) 退出mysql

```
mysql> exit;
```

3、配置MySQL

1) 设置密码强度和长度

上面步骤通过修改变量的方式，降低密码强度和长度只是临时的，在重启服务后，会还原默认设置

```
mysql> show variables like '%validate%';
```

```
mysql> show variables like 'validate%';
+-----+-----+
| variable_name                | value |
+-----+-----+
| validate_password_check_user_name | OFF   |
| validate_password_dictionary_file |       |
| validate_password_length       | 8     |
| validate_password_mixed_case_count | 1     |
| validate_password_number_count  | 1     |
| validate_password_policy       | MEDIUM |
| validate_password_special_char_count | 1     |
+-----+-----+
7 rows in set (0.01 sec)
```

永久修改密码强度和密码长度，可以通过修改mysql配置文件的方式。

退出mysql客户端，修改mysql的配置文件：

```
[root@localhost ~]# vim /etc/my.cnf
```

文件中添加如下代码：

```
validate_password_policy=LOW          # 设置密码强度为低
validate_password_length=6           # 设置密码长度为6位
```

:wq 保存并退出

重启mysql服务

```
[root@localhost ~]# systemctl restart mysqld
```

2) 设置默认编码为utf8

查看字符编码

```
mysql> show variables like '%char%';
```

```
mysql> show variables like '%char%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	latin1
character_set_filesystem	binary
character_set_results	utf8
character_set_server	latin1
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/
validate_password_special_char_count	1

9 rows in set (0.00 sec)

退出mysql客户端，修改mysql的配置文件：

```
# vim /etc/my.cnf
```

文件中添加如下代码：

```
character_set_server=utf8  
init_connect='SET NAMES utf8'
```

:wq 保存并退出

重启mysql服务

再进入mysql客户端并查看

```
mysql> show variables like '%char%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/
validate_password_special_char_count	1

9 rows in set (0.00 sec)

4、Windows下连接MySQL

1) 关闭Centos7防火墙

```
[root@localhost ~]# systemctl stop firewalld
```

2) 在windows的cmd中访问mysql

```
C:\Users\ThinkPad>mysql -h 192.168.19.131 -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.41 MySQL Community Server (GPL)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

3) 在navicat中访问



5、数据库备份

1) 数据库备份

```
[root@localhost ~]# cd /var/lib/mysql          # 进入到MySQL库目录
[root@localhost mysql]# mysqldump -u root -p 数据库名>/指定目录/数据库文件.sql
```

2) 数据库还原

```
[root@localhost ~]# cd /var/lib/mysql          # 进入到MySQL库目录
[root@localhost mysql]# mysql -u root -p 数据库名</指定目录/数据库文件.sql
```

3) 其他配置

- 1、设置安全选项: mysql_secure_installation
- 2、关闭MySQL: systemctl stop mysqld
- 3、重启MySQL: systemctl restart mysqld
- 4、查看MySQL运行状态: systemctl status mysqld
- 5、设置开机启动: systemctl enable mysqld
- 6、关闭开机启动: systemctl disable mysqld
- 7、其他默认文件路径:
 - 配置文件: /etc/my.cnf
 - 日志文件: /var/log/mysqld.log
 - socket文件: /var/run/mysqld/mysqld.pid
 - 服务启动脚本: /usr/lib/systemd/system/mysqld.service
 - 数据库文件: /var/lib/mysql

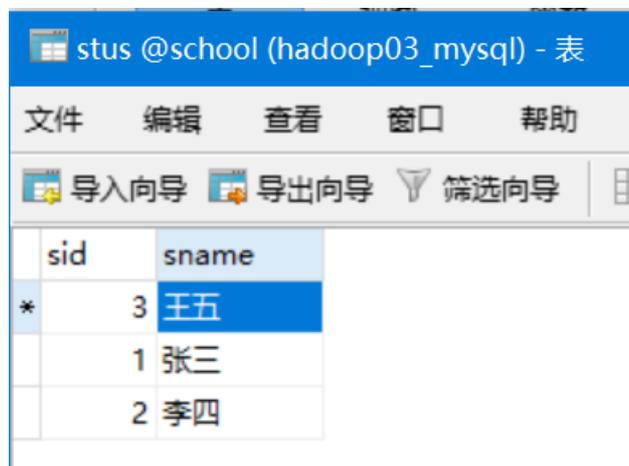
作业:

- 1、安装mysql, 启动服务, 修改密码
- 2、开启mysql**远程访问**, 设置**编码格式为utf8**
- 3、进入mysql, 创建**数据库school**和**表stus**, 插入一条测试数据

```
mysql> insert into stus values(null,'张三');
Query OK, 1 row affected (0.06 sec)

mysql> select * from stus;
+-----+-----+
| sid | sname |
+-----+-----+
| 1 | 张三 |
+-----+-----+
```

- 4、使用**windows下的navicat**连接**linux下的mysql**, 利用navicat向数据库插入测试数据



sid	sname
*	3 王五
	1 张三
	2 李四

在linux下的mysql中查询数据

```
mysql> select * from stus;
+-----+-----+
| sid | sname |
+-----+-----+
| 1   | 张三  |
| 2   | 李四  |
| 3   | 王五  |
+-----+-----+
3 rows in set (0.00 sec)
```

5、对数据库school进行备份后删除stus表

```
mysql> use school;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+-----+
| Tables_in_school |
+-----+-----+
| stus              |
+-----+-----+
1 row in set (0.00 sec)

mysql> drop table stus; ← 删除表
Query OK, 0 rows affected (0.30 sec)

mysql> show tables; ← 数据库中无表
Empty set (0.00 sec)

mysql> exit;
Bye
```

6、使用备份文件还原数据库

```
mysql> use school;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+-----+
| Tables_in_school |
+-----+-----+
| stus              |
+-----+-----+
1 row in set (0.00 sec)

mysql> select * from stus; ← 原删除的表已还原
+-----+-----+
| sid | sname |
+-----+-----+
| 1   | 张三  |
| 2   | 李四  |
| 3   | 王五  |
+-----+-----+
3 rows in set (0.00 sec)
```