

深入解析文件系统原理

1. inode and block 概述

1.1 查看文件的inode信息： stat

1.2 目录文件结构

1.3 文件系统访问文件的步骤

1.4 查看文件的inode号码

1.5 删除指定inode号所对应的文件

1.6 查看文件系统的inode与block的信息

1.7 格式化时指定文件系统的inode个数与block大小

1.8 解决inode耗尽导致的磁盘故障

1.删除不使用的文件，配置磁盘配额

2.将文件备份，重新格式化此文件系统，指定较多的inode个数

2. Linux文件系统软硬件链接原理

2.1 硬链接 (hard link)

2.2 软链接 (soft link) ----别名：符号链接

网络参数管理

一、查看及测试网络

1、使用ifconfig命令查看网络接口

(1) 查看活动（激活）的网络接口

(2) 查看所有网络接口

(3) 查看指定的网络接口（不论该网络接口是否处于激活状态）

(4) ifconfig命令包含的信息：

2、使用ip、ethtool命令查看网络接口

(1) 查看网络接口的数据链路层信息

(2) 查看网络接口的网络层信息

(3) 查看指定网络接口速率、模式等信息

(4) 查看网卡链路是否已经接好

3、查看主机名、路由表

(1) hostname命令查看主机名称

(2) 查看路由表

4、查看网络连接情况 netstat命令

5、ping、tracert命令

二、设置网络地址参数

1、临时修改网络配置

2、固定修改网络配置--改文件

3、临时设置网卡子接口

4、永久设置网卡子接口

5、临时修改网卡的状态

6、重新加载网络配置文件

三、修改主机名

1、临时修改主机名

2、永久修改主机名

四、配置路由

1、临时配置路由

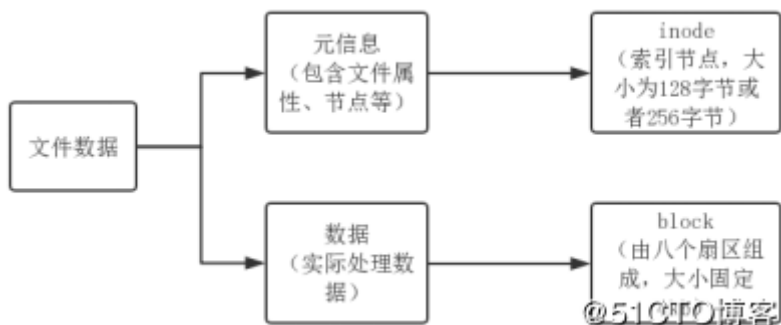
2、永久配置路由

五、设置DNS域名解析

1、设置DNS域名解析

深入解析文件系统原理

笔试题经常问----inode/block



1. inode and block 概述

- 扇区、柱面、磁道
- 硬盘最小存储单位“扇区” (sector) ,512字节~0.5KB,元信息（属性信息）, 储存元文件信息的区域叫做inode, 中文叫“索引节点”、“i节点”。
- block--块--是文件存取的最小单位=4KB, 连续8个扇区组成一个块, block存储文件数据
- inode和block是文件系统组成的基本核心概念, 文件系统是在分区格式化的时候形成的, 文件系统负责组织文件在分区上的存放。

注意: Inode不包含文件名, 文件名是储存在目录的目录项中
一个文件必须占用一个inode, 至少占用一个block。

1.1 查看文件的inode信息： stat

英文	别称	中文翻译	何时修改	查看命令
Access	Atime	最后一次访问文件或者目录的时间	读取	ls -lu
Modify	Mtime	最后一次修改文件或目录内容的时间	写入、修改	ls -l
Change/Create	Ctime	最后一次改变文件或属性的时间	修改文件名、写入、修改、改权限 (chmod)、做链接	ls -lc

面试

[Inode内容](#)

- * 文件的字节数
- * 文件拥有者的User ID
- * 文件的Group ID
- * 文件的读、写、执行权限
- * 文件的时间戳，共有三个：ctime指inode上一次变动的时间，mtime指文件内容上一次变动的时间，atime指文件上一次打开的时间。
- * 链接数，即有多少文件名指向这个inode
- * 文件数据block的位置

inode号码

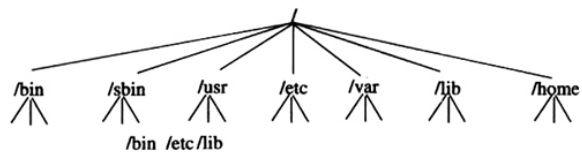
每个inode都有一个号码，操作系统用inode号码来识别不同的文件。

Unix/Linux系统内部不使用文件名，而使用inode号码来识别文件。对于系统来说，文件名只是inode号码便于识别的别称或者绰号。

表面上，用户通过文件名，打开文件。实际上，系统内部这个过程分成三步：
 首先，系统找到这个文件名对应的inode号码；
 其次，通过inode号码，获取inode信息；
 最后，根据inode信息，找到文件数据所在的block，读出数据。

1.2 目录文件结构

图1 Linux文件系统

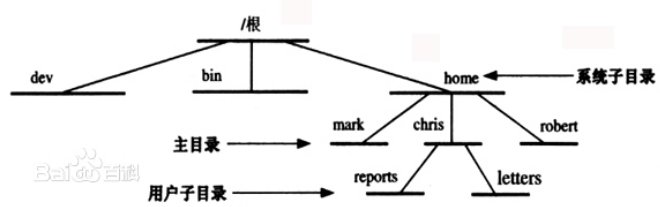


DOS也采用目录树的结构，但是与Linux的略有不同，如图2所示。

图2 DOS文件系统



图3 从根目录开始的Linux文件结构



目录也是一种文件

目录文件的结构

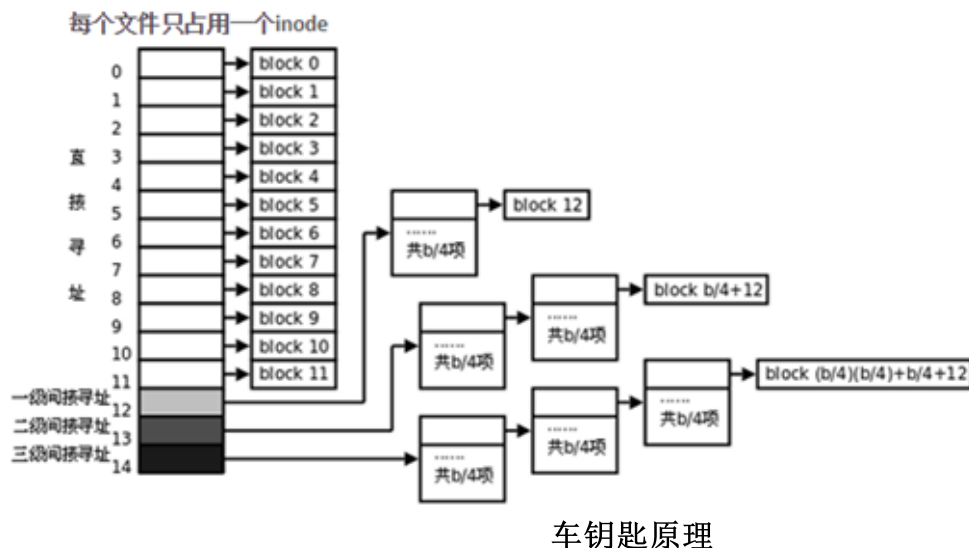
文件名1	inode 号码1
------	-----------

文件名2	inode 号码1
.....

- 每个inode都有一个号码，操作系统用inode号来识别不同的文件
- 文件系统内部不使用文件名，而是使用inode号码来识别文件。对于文件系统来说，文件名只是inode号码便于识别的别称，文件名是目录的数据。

1.3 文件系统访问文件的步骤

1. 用户在目录中看到要访问的文件名
2. 通过目录的数据找到这个文件名对应的inode号码
3. 通过inode号码，获取inode信息（文件的元信息）
4. 根据inode信息，找到文件数据所在的block，读出数据



inode一般为128字节或256字节，每个inode中都记录着文件所使用的的block号，每条记录着block号的信息占用4字节（4B）。inode中关于block号的记录一共包含12个直接、1个间接、1个双间接和1个三间接。

直接：12*4B=48B 12*4KB=48KB

间接：4B -> 4KB 4KB=1024B*4 / 4B =1024 1024*4KB=4096KB

双间接：4096KB = 1024B *4096 / 4B *4KB =1024*1024*4KB /1024KB =4096MB

三间接：1024*1024*4KB=1024*1024*4KB*1024B

=1024*1024*4*1024B / 4B * 4KB

=1024*1024*1024*4KB

=1024*1024*4MB

=1024*4GB

=4TB

在一个block大小默认为4KB的文件系统中，一个文件最大存储为：

48KB+4096KB+4069MB+4096GB ~ 4100GB

1.4 查看文件的inode号码

- `ls -i filename`
- `stat filename`

文件储存位置

文件名	目录项
元信息	inode
数据	block

Superblock里面存储的文件系统所有Inode、block的相关信息

访问过程：

当一个用户在Linux系统中试图访问一个文件时，系统会先根据文件名去查找它的inode，看该用户是否具有访问这个文件的权限。如果有，就指向相对应的数据block；如果没有，就返回Permission denied（拒绝访问）。

1.5 删除指定inode号所对应的文件

格式：`find ./ -inum inode号 -exec rm -rf {} \;`

当文件名中包含特殊符号时我们可以删除其inode号达到删除文件的目的

find命令本身是可以直接引用shell命令的 `-exec`

删前面的内容输出用`{}`表示

`\;` 过度结束符

注意：

```
[root@localhost ~]# find ./ -inum 100847539 | rm -rf
```

命令字 选项 参数

如果有默认参数就不能从前面接收，管道就失效了

```
[root@localhost ~]# find ./ -name "*.txt" | xargs ls -l
```

当文件名中有空格的时候，会被识别为多个文件，这时不能用 `xargs`，如果用管道会降低效率

面试题:

查找/root 目录中3天内修改过的文件，并移动到/tmp目录下

```
[root@localhost ~]# find /root -type f -a -mtime -3 -exec mv {} /tmp \;
```

1.6 查看文件系统的inode与block的信息

- **df -i 设备名**（文件系统已挂载时查询，查询inode总数与已用数量）
- **dumpe2fs -h 设备名**（文件系统无需挂载）

```
[root@localhost ~]# dumpe2fs -h /dev/sdb1 | grep "Inode count"
dumpe2fs 1.42.9 (28-Dec-2013)
```

```
Inode count:                65536
```

- **tune2fs -l 设备名**（文件系统无需挂载）

```
[root@localhost ~]# tune2fs -l /dev/sdb1 | grep "Inode count"
Inode count:                65536
```

1.7 格式化时指定文件系统的inode个数与block大小

格式: `mkfs.ext4 -N inode 数 -b 块大小（单位字节）设备名`

```
[root@localhost ~]# mkfs.ext4 -N 60000 -b 1024 /dev/sdb1
```

1.8 解决inode耗尽导致的磁盘故障

- 1.删除不使用的文件，配置磁盘配额
- 2.将文件备份，重新格式化此文件系统，指定较多的inode个数

2. Linux文件系统软硬件链接原理

2.1 硬链接 (hard link)

方法: `ln 源文件 目标文件`

特点: ---快捷方式

硬链接指向的文件inode号，新生成的硬链接文件的inode号与源文件的inode号相同，不可针对目录进行硬链接，必须在同一文件系统内。删除一个文件名，不影响另外一个的访问。

注意：（考点）

不能对目录、对分区做硬链接

1. [root@localhost ~]# ln /boot/ /tmp/

ln: "/boot/": 不允许将硬链接指向目录

2. [root@localhost ~]# ln bb.txt /boot/dd.txt

ln: 无法创建硬链接"/boot/dd.txt" => "bb.txt": 无效的跨设备连接

例如:

```
[root@localhost ~]# ll /bin/passwd
```

```
-rwsr-xr-x. 1 root root 27832 6月 10 2014 /bin/passwd
```

```
[root@localhost ~]# ll /usr/bin/passwd
```

```
-rwsr-xr-x. 1 root root 27832 6月 10 2014 /usr/bin/passwd
```

```
[root@localhost ~]# ls -li /bin/passwd
```

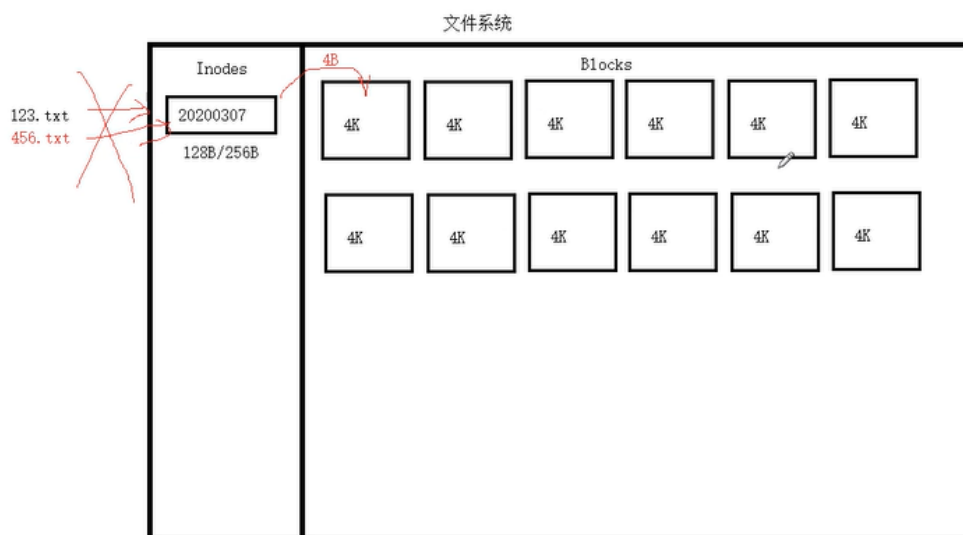
```
546399 /bin/passwd
```

```
[root@localhost ~]# ls -li /usr/bin/passwd
```

```
546399 /usr/bin/passwd
```

3. 区分于备份

4. 删除文件的原理：只是断开快捷方式与inode的链接方式，数据还在，但是再写入数据会覆盖掉，相当于删除了。



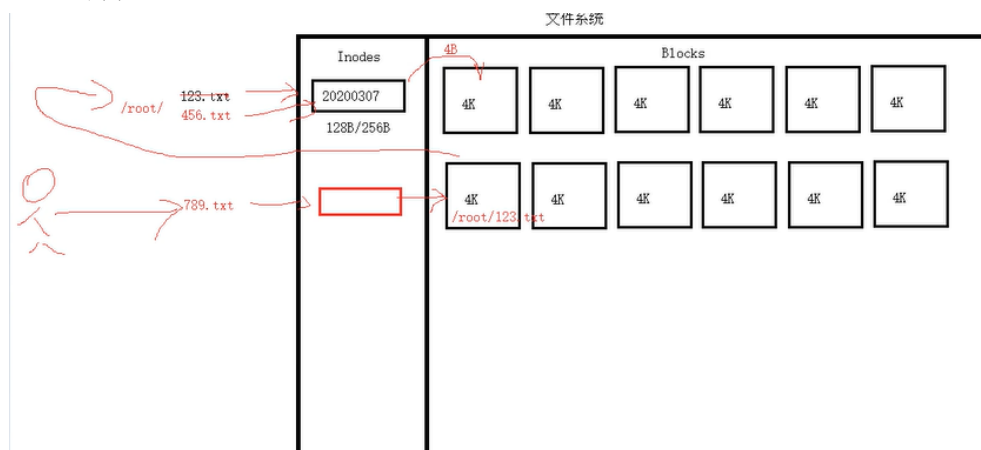
2.2 软链接 (soft link) -----别名：符号链接

方法: ln -s 源文件或目录 目标文件或目录

特点: 快捷方式

软链接指向的文件名，新生成的软链接文件的inode号与源文件不同，目录也可以生成软链接，软链接文件与源文件可以不在同一文件系统内，软链接文件的内容是源文件的

路径，读取时系统会自动导向源文件路径，根据源文件找到文件内容，但当源文件移动或重命名时，软链接将报错。



注意：源文件和目标文件尽量用绝对路径表示

案例：

```
[root@localhost ~]# ln -s /root/bb.txt /boot/cc.txt
[root@localhost ~]# cat bb.txt
hehe
[root@localhost ~]# cat /boot/cc.txt
hehe
[root@localhost ~]# ls -l /boot/cc.txt
lrwxrwxrwx. 1 root root 12 3月 10 02:39 /boot/cc.txt -> /root/bb.txt
[root@localhost ~]# ls -l bb.txt
-rw-r--r--. 1 root root 5 3月 10 01:51 bb.txt
[root@localhost ~]# rm -rf bb.txt
[root@localhost ~]# ls -l /boot/cc.txt
lrwxrwxrwx. 1 root root 12 3月 10 02:39 /boot/cc.txt -> /root/bb.txt
[root@localhost ~]# cat /boot/cc.txt
cat: /boot/cc.txt: 没有那个文件或目录
[root@localhost ~]# echo "hahahahaha">/root/bb.txt
[root@localhost ~]# cat /boot/cc.txt
hahahahaha
[root@localhost ~]# ls -l /boot/cc.txt
lrwxrwxrwx. 1 root root 12 3月 10 02:39 /boot/cc.txt -> /root/bb.txt
```

硬链接与软链接的对比

	硬链接	软链接
指向	inode号	文件名
inode号是否相同	相同	不同
是否可以针对目录	不可针对目录	可以针对目录
可否跨文件系统	不可跨文件系统	可跨文件系统
删除、重命名、移动源文件	不受影响	失效
创建命令	ln 源文件 目标文件	ln -s 源文件或目录 目标文件或目录

注意：

```
[root@localhost ~]# ll anaconda-ks.cfg
-rw-----. 1 root root 1806 1月 1 00:56 anaconda-ks.cfg
[root@localhost ~]# ln anaconda-ks.cfg an.cfg # 硬链接
[root@localhost ~]# ll anaconda-ks.cfg
-rw-----. 2 root root 1806 1月 1 00:56 anaconda-ks.cfg
[root@localhost ~]# ln -s /root/anaconda-ks.cfg /tmp/a.cfg # 软链接
[root@localhost ~]# ll anaconda-ks.cfg
-rw-----. 2 root root 1806 1月 1 00:56 anaconda-ks.cfg
```

硬链接会改变数，软链接不会改变

网络参数管理

一、查看及测试网络

1、使用ifconfig命令查看网络接口

(1) 查看活动（激活）的网络接口

```
ifconfig命令
```

(2) 查看所有网络接口

```
ifconfig -a命令
```

(3) 查看指定的网络接口（不论该网络接口是否处于激活状态）

```
ifconfig 设备名（服务器上的网卡名识别em1-4）
[root@localhost ~]# ifconfig ens32
```

```

ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.200.103 netmask 255.255.255.0 broadcast
192.168.200.255          #网关
    inet6 fe80::6f35:3c2c:e34b:1da0 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:57:13:4d txqueuelen 1000 (Ethernet)          #传
输速率
    RX packets 2368 bytes 171917 (167.8 KiB)
#接收数据包个数/大小
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 664 bytes 147962 (144.4 KiB)
#传输数据包个数/大小
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

(4) ifconfig命令包含的信息:

Link encap: Ethernet以太网 HWaddr: 物理地址 (MAC地址)

inet addr: IPv4地址 Bcast: 广播地址 Mask: 子网掩码

inet6 addr: IPv6地址

MTU: 最大传输单元 (Maximum Transmission Unit) Metric: 跳跃点

RX packets: 接收数据包 errors: 错误 dropped: 丢弃 overruns: 过
 载 frame: 帧数

TX packets: 发送数据包 errors: 错误 dropped: 丢弃 overruns: 过
 载 carrier: 载波

collisions: 冲撞 txqueuelen: 发送队列长度

RX bytes: 接收字节数 TX bytes: 发送字节数

2、使用ip、ethtool命令查看网络接口

(1) 查看网络接口的数据链路层信息

```

ip link命令
[root@localhost ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000 link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DEFAULT group default qlen 1000 link/ether
00:0c:29:5d:d7:80 brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
state DOWN mode DEFAULT group default qlen 1000 link/ether

```

```
52:54:00:76:88:ad brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master
virbr0 state DOWN mode D
EFAULT group default qlen 1000    link/ether 52:54:00:76:88:ad brd
ff:ff:ff:ff:ff:ff
```

(2) 查看网络接口的网络层信息

ip a (ip address) 命令 **常用**

```
[root@localhost ~]# ip a
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: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default
    qlen 1000    link/ether 00:0c:29:5d:d7:80 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.105/24 brd 192.168.200.255 scope global noprefixroute
ens32
    valid_lft forever preferred_lft forever
    inet6 fe80::62a7:4575:6d9b:6cb4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group defau
lt qlen 1000    link/ether 52:54:00:76:88:ad brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0
state DOWN grou
p default qlen 1000    link/ether 52:54:00:76:88:ad brd ff:ff:ff:ff:ff:ff
```

(3) 查看指定网络接口速率、模式等信息

ethtool命令

```
[root@localhost ~]# ethtool ens32
Settings for ens32:
Supported ports: [ TP ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 0
Transceiver: internal
Auto-negotiation: on
MDI-X: off (auto)
Supports Wake-on: d
Wake-on: d
Current message level: 0x00000007 (7)
                        drv probe link
Link detected: yes
```

(4) 查看网卡链路是否已经接好

```
[root@localhost ~]# mii-tool ens32
ens32: negotiated 1000baseT-FD flow-control, link ok
```

3、查看主机名、路由表

(1) hostname命令查看主机名称

```
[root@localhost ~]# hostname
```

localhost.localdomain

(2) 查看路由表

使用route命令查看路由表

- -n: 不执行DNS反向查找, 直接显示数字形式的IP地址 #主要目的是加速
- 当路由表中路由条目特别多的时候最好用 route -n

```
[root@localhost ~]# route
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
default	localhost	0.0.0.0	UG	100	0	0
ens32						
192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0
virbr0						
192.168.200.0	0.0.0.0	255.255.255.0	U	100	0	0
ens32						

```
[root@client ~]# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use
0.0.0.0	192.168.200.1	0.0.0.0	UG	100	0	0
ens32						
192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0
virbr0						
192.168.200.0	0.0.0.0	255.255.255.0	U	100	0	0
ens32						

使用netstat -r命令查看路由表

- -n: 不执行DNS反向查找, 直接显示数字形式的IP地址

```
[root@localhost ~]# netstat -rn
```

```
[root@client ~]# netstat -r
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt
default	localhost	0.0.0.0	UG	0	0	0
ens32						

```

192.168.122.0    0.0.0.0          255.255.255.0   U           0 0           0
virbr0
192.168.200.0   0.0.0.0          255.255.255.0   U           0 0           0
ens32
[root@client ~]# netstat -rn #加速作用
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt
Iface
0.0.0.0          192.168.200.1   0.0.0.0         UG        0 0         0
ens32
192.168.122.0    0.0.0.0          255.255.255.0   U         0 0         0
virbr0
192.168.200.0    0.0.0.0          255.255.255.0   U         0 0         0
ens32

```

4、查看网络连接情况 netstat命令

常用选项：

- -a: 显示当前主机中所有活动的网络连接信息
- -n: 以数字的形式显示相关信息
- -r: 显示路由表信息-----基本不用
- -t: 显示TCP协议相关的信息
- -u: 显示UDP协议相关的信息
- -p: 显示与网络连接相关的进程号、进程名称信息（需要root权限）
- -l: 查看监听状态的网络连接信息

```

[root@localhost ~]# netstat -anpt | head -5
[root@localhost ~]# netstat -anpu | head -5
[root@localhost ~]# netstat -anptu | head -5
[root@localhost ~]# netstat -lnpt | grep :22
常用: -anpt , -lnpt

```

类似netstat命令的有：

```

[root@server ~]# ss -lnpt | grep :80
LISTEN      0        128          :::80          :::*
users:((("h

```

```
ttpd",pid=7756,fd=4), ("httpd",pid=7755,fd=4), ("httpd",pid=7754,fd=4),  
("httpd",pid=7753,fd=4), ("httpd",pid=7752,fd=4), ("httpd",pid=7749,fd=4))
```

```
[root@localhost ~]# lsof -i :80
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
httpd	7749	root	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)
httpd	7752	apache	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)
httpd	7753	apache	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)
httpd	7754	apache	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)
httpd	7755	apache	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)
httpd	7756	apache	4u	IPv6	96623	0t0	TCP	*:http (LISTEN)

案例:

```
[root@server ~]# yum -y install httpd
```

```
[root@server ~]# systemctl start httpd
```

```
[root@server ~]# netstat -lnpt | grep :80
```

```
tcp6          0      0 ::::80          :::*             LISTEN  
7749/httpd
```

5、ping、traceroute命令

ping命令：测试网络连通性

traceroute命令：状态跟踪

常用选项：

- -c: 指定发送数据包的个数
- -i: 当ping通时，指定间隔多少秒发送下一个数据包
- -w: 当ping不通时，指定发送的每个数据包的超时时间，单位秒
- -s: 指定数据包大小

```
[root@localhost ~]# ping 192.168.200.2
```

【课外补充】

1、windows 中的 tracert命令

在dos中输入tracert -d (-d 不将地址解析成主机名)

```
C:\Users\lirui>tracert -d www.baidu.com
```

通过最多 30 个跃点跟踪

到 www.a.shifen.com [180.101.49.11] 的路由:

1	4 ms	1 ms	3 ms	192.168.2.1
2	4 ms	1 ms	1 ms	192.168.1.1
3	3 ms	4 ms	3 ms	101.95.91.6
4	8 ms	4 ms	8 ms	101.95.91.129
5	*	3 ms	5 ms	61.152.25.190
6	6 ms	6 ms	10 ms	202.97.74.158
7	8 ms	10 ms	8 ms	58.213.94.114
8	*	*	*	请求超时。
9	8 ms	12 ms	8 ms	58.213.96.54
10	*	*	*	请求超时。
11	*	*	*	请求超时。
12	*	*	*	请求超时。
13	10 ms	9 ms	8 ms	180.101.49.11

跟踪完成。

2、常见的TTL (Time To Live) 生存周期值:

windows: 128

linux: 64

unix: 255

cisco: 255

公司中常伪装ttl值, 经常看可以判断出来

比如; 让windows报80, linux报128

测试DNS域名解析 nslookup命令

www.taobao.com 叫域名, 就是给服务器起的名字, 但服务器真正工作还是通过IP地址, 可是IP地址不好记, 所以起了名字, 用nlookup命令, 看当前主机能否把域名的地址获取, 就是测试地址解析的, 看能不能解析成对应的IP。如果测不出来, 那么就是你指定的DNS地址有问题。

C:\Users\lirui>nslookup www.taobao.com

服务器: gjjline.bta.net.cn

Address: 202.106.0.20

DNS request timed out.

timeout was 2 seconds.

名称: www.taobao.com.danuoyi.tbcache.com

Addresses: 2408:871a:2800:2:3::3fa

2408:871a:1120:5002:3::3f9

2408:871a:5100:103:2::3d3

2408:871a:a900:2:3::3f9

2408:871a:5100:103:2::3d4

2408:871a:2800:4:3::3f9

Aliases: www.taobao.com

二、设置网络地址参数

1、临时修改网络配置

(1) 方法一: ifconfig 网络设备 IP地址[/短格式子网掩码]

(2) 方法二: ifconfig 网络设备 IP地址 [netmask 子网掩码]

```
[root@localhost ~]# ifconfig ens32 192.168.200.111          #子网掩码默认
255.255.255.0
```

```
[root@localhost ~]# ifconfig ens32 192.168.200.111/24
```

```
[root@localhost ~]# ifconfig ens32 192.168.200.111 netmask 255.255.255.0
```

#子网掩码不是默认需要指定

2、固定修改网络配置--改文件

(1) 配置文件: /etc/sysconfig/network-scripts/目录下的ifcfg-对应网络设备名称, 默认第一块网卡为ens32或者ens33

(2) 编辑配置文件: 前面带#表示非必要配置内容

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens32
```

```
TYPE=Ethernet          #类型(以太网)网卡, 默认
```

```
PROXY_METHOD=none
```

```
BROWSER_ONLY=no
```

```
DEFROUTE=yes
```

```
IPV4_FAILURE_FATAL=no
```

```
IPV6INIT=no
```

```
IPV6_AUTOCONF=no
```

```

IPV6_DEFROUTE=no
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens32
#设备名
UUID=b194e583-21c3-4109-a160-3d2a9876a30b          #UUID号
BOOTPROTO=static          #引导协议（dhcp：自动获取；static/none：
手动配置）
DEVICE=ens32
#设备名
ONBOOT=yes
#是否开机自动启用
IPADDR=192.168.200.111
#IPv4协议的IP地址
PREFIX=24          #短格式子网掩码 也可以写
NETMASK=255.255.255.0
GATEWAY=192.168.200.1          #网
关地址
DNS1=202.106.0.20
#DNS域名解析服务
HWADDR=00:0C:29:8F:D8:E0          #物理
地址（MAC地址）
IPV6_PRIVACY=no

```

3、临时设置网卡子接口

方法：ifconfig 网络设备:子接口名称 IP地址/短格式子网掩码

```

[root@localhost ~]# ifconfig ens32:0 192.168.1.1/24
[root@localhost ~]# ifconfig ens32:sec 192.168.2.1/24
[root@localhost ~]# ifconfig

```

4、永久设置网卡子接口

(1) 方法：需要在/etc/sysconfig/network-scripts/目录下手动添加配置文件

```

[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens32:0
TYPE=Ethernet

```

```
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=no
IPV6_DEFROUTE=no
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens32:0
DEVICE=ens32:0
ONBOOT=yes
IPADDR=192.168.200.11
PREFIX=24
GATEWAY=192.168.200.1
DNS1=202.106.0.20
IPV6_PRIVACY=no
[root@localhost ~]# systemctl restart network
```

配置子接口（永久）案例：

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost ~]# cp ifcfg-ens32 ifcfg-ens32:1
[root@localhost ~]# vim ifcfg-ens32:1
[root@localhost ~]# systemctl restart network
```

5、临时修改网卡的状态

(1) 方法：ifconfig 网络设备 up/down

```
[root@localhost ~]# ifconfig ens32 down && ifconfig ens32 up
```

down：禁

up：开启

6、重新加载网络配置文件

(1) 重启网络服务以实现重新读取配置文件的目的

方法：systemctl restart network = service network restart

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

(2) 修改某块网卡配置后，仅重启该网卡

格式：ifdown 网络设备;ifup 网络设备

```
[root@localhost ~]# ifdown ens32 ; ifup ens32
```

三、修改主机名

1、临时修改主机名

hostname 新的主机名

```
[root@localhost ~]# hostname crushlinux.com
```

```
[root@localhost ~]# bash
```

```
[root@crushlinux ~]# hostname
```

crushlinux.com

2、永久修改主机名

方法1：编辑/etc/sysconfig/network配置文件，重启系统后生效

```
[root@crushlinux ~]# vim /etc/sysconfig/network
```

```
# Created by anaconda
```

```
HOSTNAME=crushlinux.com
```

方法2：编辑/etc/hostname配置文件，重启后生效（推荐）

```
[root@crushlinux ~]# vim /etc/hostname
```

```
crushlinux.com
```

既不用服务器重启，又能现在就能变用户名的方法：

```
[root@server ~]# vim /etc/hostname
```

删掉原来的名字，添加你想改的名字

```
[root@server ~]# hostname abc.com
```

```
[root@server ~]# bash
```

```
[root@abc ~]#
```

四、配置路由

1、临时配置路由

(1) 临时添加、删除指定网段的路由记录

方法：route add -net 网段/短格式子网掩码 (/24,/16,/8) gw 网关地址

route del -net 网段/短格式子网掩码

```
[root@crushlinux ~]# route add -net 192.168.2.0/24 gw 192.168.200.1
[root@crushlinux ~]# route -n
[root@crushlinux ~]# route del -net 192.168.2.0/24
[root@crushlinux ~]# route -n
```

(2) 临时添加、删除默认网关记录

方法: route add default gw 网关地址

 route del default gw 网关地址

```
[root@crushlinux ~]# route del default gw 192.168.200.1
[root@crushlinux ~]# route -n
[root@crushlinux ~]# route add default gw 192.168.200.1
```

2、永久配置路由

(1) 方法一: 在/etc/rc.local中添加

软链接示例:

```
[root@abc ~]# ll /etc/rc.d/rc.local
-rw-r--r--. 1 root root 473 4月 11 2018 /etc/rc.d/rc.local
[root@abc ~]# ll /etc/rc.local
lrwxrwxrwx. 1 root root 13 1月 1 00:44 /etc/rc.local -> rc.d/rc.local
```

```
[root@crushlinux ~]# vim /etc/rc.local            #每次开机都会执行
route add -net 192.168.2.0/24 gw 192.168.200.1
route add -net 192.168.2.0/24 dev ens32
```

(2) 方法二: 在/etc/sysconfig/network中添加到末尾

注意: 网卡的配置文件中如果有设置了网关, 该文件优先级高于此配置文件

```
[root@crushlinux ~]# grep "GATEWAY" /etc/sysconfig/network-scripts/ifcfg-ens32

GATEWAY=192.168.200.1
```

(3) 方法三: 修改/etc/sysconfig/static-routes配置文件 (没有该文件手动建立)

#比较麻烦

```
[root@crushlinux ~]# vim /etc/sysconfig/static-routes
any net 192.168.1.0/24 gw 192.168.200.1
any net 192.168.2.0 netmask 255.255.255.0 gw 192.168.200.1
```

```
[root@crushlinux ~]# systemctl restart network
```

```
[root@crushlinux ~]# route -n
```

(4) 方法四：开启IP转发（路由功能）-----路由转发

1> echo "1" > /proc/sys/net/ipv4/ip_forward （临时开启）

2> 编辑/etc/sysctl.conf 文件将net.ipv4.ip_forward=0改为1（永久开启）

sysctl -p （使sysctl.conf文件立即生效）

临时开启

```
[root@crushlinux ~]# cat /proc/sys/net/ipv4/ip_forward
```

0

```
[root@crushlinux ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

永久开启

```
[root@crushlinux ~]# vim /etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1
```

```
[root@crushlinux ~]# sysctl -p
```

五、设置DNS域名解析

1、设置DNS域名解析

(1) 方法一：编辑/etc/sysconfig/network-scripts/（网卡配置文件）目录下网络设备的配置文件

```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens32
```

```
DNS1=202.106.0.20    #北京地区，根据地区走的，不同地区用不同的DNS
```

```
DNS2=8.8.8.8
```

```
DNS3=114.114.114.114
```

```
# 当第一个不能用的时候，用第二个，第二个不行第三个
```

(2) 方法二：vi编辑/etc/resolv.conf文件

```
[root@crushlinux~]# vim /etc/resolv.conf
```

```
# Generatedby NetworkManager
```

```
nameserver 02.106.0.20
```

```
nameserver 8.8.8.8
```

```
search localhost
```

2、域名解析本地主机映射文件

(1) 方法：编辑/etc/hosts文件, 系统查找优先于DNS，放常用的域名和IP地址，起到加速作用

(2) 说明：

/etc/hosts文件优先于DNS域名解析服务，也就是说，如果一个域名在hosts文件中已存在映射关系，将不再通过DNS服务器进行域名解析。hosts文件中一个ip地址可以对应多个域名或者别名。

Linux: /etc/hosts

Windows: C:\Windows\System32\drivers\etc\hosts

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

```
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhost4
::1          localhost localhost.localdomain localhost6
localhost6.localhost6
192.168.200.111 www.crushlinux.com
```

```
[root@crushlinux ~]# ping -c 2 www.crushlinux.com
```

```
PING www.crushlinux.com (192.168.200.111) 56(84) bytes of data.
64 bytes from www.crushlinux.com (192.168.200.111): icmp_seq=1 ttl=64
time=0.089 ms
64 bytes from www.crushlinux.com (192.168.200.111): icmp_seq=2 ttl=64
time=0.109 ms
--- www.crushlinux.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.089/0.099/0.109/0.010 ms
```