一、Linux 网络配置详解

来自：<http://hi.baidu.com/liulei0726/blog/item/833c5a5e2e2c8848fbf2c05d.html>

查看网路接口信息

　　1．了解linux中的网络接口设备

　　$/sbin/ifconfig 查看所有活动网络接口信息，其中包括一个lo环回端口。

　　2．查看指定网络接口的信息

　　$/sbin/ifconfig <网络接口名称> 查看指定接口信息。

　　3．查看系统中所有网络接口的信息

　　$/sbin/ifconfig –a 查看所有接口信息，包括非活动状态接口。

　　注：普通用户查看网络接口信息时，需要在命令前加“/sbin”，而管理员却不用。这是由于命令搜索路径的原因，可以通过“$echo $PATH”查看。

　　查看网关地址和路由信息

　　1、查看主机路由信息

　　#/route

　　显示当前linux主机中的路由表信息。

　　#/route ｜grep default

　　在route命令的显示结果中以“default”开始的行显示了，主机的默认网关地址，将route命令和grep命令组合，使用过滤route命令的显示结果只显示默认网关地址。

　　2.测试与其它主机的网络链接

　　$ping 目的主机地址

　　使用ping命令测试与其它主机的网络链接。ctrl+C终止。

　　$ping –c <测试数据包数量> <目的主机地址>

　　使用ping命令发送指定数量的数据包进行网络链接测试。

　　$使用ping命令进行网络链接测试的技巧。

　　注：当ping命令测试不成功时并不能判断当前主机到目标主机的连接是断开的，因为有很多因素，（如网络防火墙等）可以导致网络连接正常时主机之间无法ping通。

　　3、测试与其它主机的网络链接路径

　　$traceroute <目的主机地址>

　　traceroute命令显示当前主机与目的主机之间经过的所，有网络节点的地址，以及当前主机到每个中间结点的连接状态。

　　查看主机名称信息

　　1、查看当前主机名称

　　$Hostname 查看当前的主机名。

　　2、使用dns服务器查询域名

　　$Nslookup

　　交互模式：通常用于对DNS服务器进行测试

　　直接模式：用于查询某个域名对应的ip地址

　　使用网络命令进行网络设置

　　网络设置的方法

　　1、DHCP网络配置

　　$Dhclient 使用dhclient命令可以从DHCP服务器中申请新的网络配置应用到当前主机。

　　注：DHCP服务器通常提供给DHCP客户端最基本的网络配置信息，包括：接口ip地址，默认网关，DNS

　　服务器地址。

　　2、手工网络配置

　　网络接口ip地址和子网掩码

　　默认网关地址

　　当前主机名称

　　DNS服务器地址

　　Ip地址配置命令

　　Ifconfig <网络接口名称> network <子网掩码>

　　例：# ifconfig eth0 192.168.1.222 netmask 255.255.255.0

　　注：手工配置linux网络只限于临时使用，配置不保存。

　　扩展：# ifconfig eth0:1 10.0.0.1 netmask 255.0.0.0 一块网卡配置多个子ip地址。

　　# ifconfig eth0 hw ether MAC地址 修改网卡的mac地址，  
路由配置命令

　　1、删除默认网关路由

　　$Route del default gw <默认网关地址>

　　2、添加默认网关路由

　　$Route add default gw <默认网关地址>

　　例：#route add default gw 192.168.1.1

　　主机名称配置命令

　　# hostname 123

　　将主机暂时名称改为123。

　　DNS域名解析

　　Nslookup

　　>server

　　显示linux系统中使用的DNS服务器地址，也可临时设置生效。

　　修改配置文件进行网络设置

　　使用工具进行网络设置

　　# Netconfig 图形界面设置网络，重启network后生效。

　　界面中需要填写的内容：

　　主机网络接口的ip地址。

　　网络接口的子网掩码。

　　默认网关地址。

　　主DNS服务器的ip地址。

　　网络服务启动脚本

　　/etc/init.d/network 从新启动才能生效

　　扩展：# /etc/rc.d/init.d/network start 启动network

　　# /etc/rc.d/init.d/network stop 停止network

　　# /etc/rc.d/init.d/network restart 重启network

　　网络接口配置文件

　　/etc/sysconfig/network-scripts/ifconfg-\*

　　DEVICE=eth0 网卡编号

　　ONBOOT=yes 开机时是否启动网卡。

　　BOOTPROTO=static 静态ip或者DHCP动态获取。

　　IPADDR=192.168.1.163 ip

　　NETMASK=255.255.255.0 掩码

　　GATEWAY=192.168.1.1 网关

　　以下为动态获取方式：

　　DEVICE=eth0

　　ONBOOT=yes

　　BOOTPROTO=dhcp

　　停止和启动指定的网络接口

　　单个网络接口的配置进行修改后

　　Ifdown 网络接口名称 停止某个网络接口。

　　Ifup 网络接口名称 开启某个网络接口。

　　注：单个网络接口的配置文件进行修改后，不必每次都从新启动network服务使配置文件生效，可用以上命令。

　　扩展：对网络接口操作：

　　# ifconfig eth0 down 停止eth0接口

　　# ifconfig eth0 up 开启eth0接口

　　可以连续使用

　　Ifdown eth0 ; ifup eth0

　　主机名称配置文件

　　/etc/sysconfig/network

　　例：# vi /etc/sysconfig/network 使用vi编辑器打开主机名配置文件。

　　NETWORKING=yes 网络是否可用。

　　HOSTNAME=xxxx xxxx为新设置的主机名。

　　本地主机名称解析文件

　　/etc/hosts

　　Hosts 和 DNS具有类似的主机名称解析功能

　　域名服务器配置文件

　　/etc/resolv.conf

　　最多可以设置3行，前面的生效。

二、Linux网络配置与IP高级路由命令

来自：<http://hi.baidu.com/pomtch_sy/blog/item/ee4acf3f33730d0bbba1672f.html>

懂得网络配置命令是一般技术人员必备的技术,经过一段时间的研究和学习,总结了一些常用的命令和示例以便日后查阅.  
　　传统的在1--3点,ip高级路由命令在4--12点,两者部分可以通用,并达到同样的目的,但ip的功能更强大,可以实现更多的配置目的。  
　　首先,先了解传统的网络配置命令:

　　1. 使用ifconfig命令配置并查看网络接口情况  
　　示例1: 配置eth0的IP，同时激活设备:  
　　# ifconfig eth0 192.168.4.1 netmask 255.255.255.0 up  
　　示例2: 配置eth0别名设备 eth0:1 的IP，并添加路由  
　　# ifconfig eth0:1 192.168.4.2  
　　# route add –host 192.168.4.2 dev eth0:1  
　　示例3:激活（禁用）设备  
　　# ifconfig eth0:1 up(down)  
　　示例4:查看所有（指定）网络接口配置  
　　# ifconfig (eth0)

　　2. 使用route 命令配置路由表  
　　示例1:添加到主机路由  
　　# route add –host 192.168.4.2 dev eth0:1  
　　# route add –host 192.168.4.1 gw 192.168.4.250  
　　示例2:添加到网络的路由  
　　# route add –net IP netmask MASK eth0  
　　# route add –net IP netmask MASK gw IP  
　　# route add –net IP/24 eth1  
　　示例3:添加默认网关  
　　# route add default gw IP  
　　示例4:删除路由  
　　# route del –host 192.168.4.1 dev eth0:1  
　　示例5:查看路由信息  
　　# route 或 route -n (-n 表示不解析名字,列出速度会比route 快)

　　3.ARP 管理命令  
　　示例1:查看ARP缓存  
　　# arp  
　　示例2: 添加  
　　# arp –s IP MAC  
　　示例3: 删除  
   # arp –d IP

　　4. ip是iproute2软件包里面的一个强大的网络配置工具，它能够替代一些传统的网络管理工具。例如：ifconfig、route等,  
　　上面的示例完全可以用下面的ip命令实现,而且ip命令可以实现更多的功能.下面介绍一些示例:  
4.0 ip命令的语法  
　　ip [OPTIONS] OBJECT [COMMAND [ARGUMENTS]]  
4.1 ip link set--改变设备的属性. 缩写：set、s  
　　示例1：up/down 起动／关闭设备。  
　　# ip link set dev eth0 up  
　　这个等于传统的 # ifconfig eth0 up(down)  
　　示例2：改变设备传输队列的长度。  
　　参数:txqueuelen NUMBER或者txqlen NUMBER  
　　# ip link set dev eth0 txqueuelen 100  
　　示例3：改变网络设备MTU(最大传输单元)的值。  
　　# ip link set dev eth0 mtu 1500  
　　示例4： 修改网络设备的MAC地址。  
　　参数: address LLADDRESS  
　　# ip link set dev eth0 address 00:01:4f:00:15:f1  
4.2 ip link show--显示设备属性. 缩写：show、list、lst、sh、ls、l  
　　-s选项出现两次或者更多次，ip会输出更为详细的错误信息统计。  
　　示例:  
　　# ip -s -s link ls eth0　　  
　　eth0:　mtu 1500 qdisc cbq qlen 100  
　　link/ether 00:a0:cc:66:18:78 brd ff:ff:ff:ff:ff:ff  
　　RX: bytes　packets　errors　dropped overrun mcast  
　　2449949362 2786187 0 0 0 0  
　　RX errors: length crc frame fifo missed  
　　　 0 0 0 0 0  
　　TX: bytes　packets errors dropped carrier collsns  
　　178558497　1783946 332 0 332 35172  
　　TX errors: aborted fifo window heartbeat  
　　　 0 0 0 332  
　　这个命令等于传统的 ifconfig eth0

5.1 ip address add --添加一个新的协议地址. 缩写：add、a  
　　示例1：为每个地址设置一个字符串作为标签。为了和Linux-2.0的网络别名兼容，这个字符串必须以设备名开头，接着一个冒号，  
　　# ip addr add local 192.168.4.1/28 brd + label eth0:1 dev eth0  
　　示例2: 在以太网接口eth0上增加一个地址192.168.20.0，掩码长度为24位(155.155.155.0)，标准广播地址，标签为eth0:Alias：  
　　# ip addr add 192.168.4.2/24 brd + dev eth1 label eth1:1  
　　这个命令等于传统的: ifconfig eth1:1 192.168.4.2  
5.2 ip address delete--删除一个协议地址. 缩写：delete、del、d  
　　# ip addr del 192.168.4.1/24 brd + dev eth0 label eth0:Alias1  
5.3 ip address show--显示协议地址. 缩写：show、list、lst、sh、ls、l  
　　# ip addr ls eth0  
5.4.ip address flush--清除协议地址. 缩写：flush、f  
　　示例1 : 删除属于私网10.0.0.0/8的所有地址：  
　　# ip -s -s a f to 10/8  
　　示例2 : 取消所有以太网卡的IP地址  
　　# ip -4 addr flush label "eth0"

6. ip neighbour--neighbour/arp表管理命令  
　　缩写 neighbour、neighbor、neigh、n  
　　命令 add、change、replace、delete、fulsh、show(或者list)  
　6.1 ip neighbour add -- 添加一个新的邻接条目  
　　ip neighbour change--修改一个现有的条目  
　　ip neighbour replace--替换一个已有的条目  
　　缩写：add、a；change、chg；replace、repl  
　　示例1: 在设备eth0上，为地址10.0.0.3添加一个permanent ARP条目：  
　　# ip neigh add 10.0.0.3 lladdr 0:0:0:0:0:1 dev eth0 nud perm  
　　示例2:把状态改为reachable  
　　# ip neigh chg 10.0.0.3 dev eth0 nud reachable  
　6.2.ip neighbour delete--删除一个邻接条目  
　　示例1:删除设备eth0上的一个ARP条目10.0.0.3  
　　# ip neigh del 10.0.0.3 dev eth0  
　6.3.ip neighbour show--显示网络邻居的信息  
. 缩写：show、list、sh、ls  
　　示例1: # ip -s n ls 193.233.7.254  
　　193.233.7.254. dev eth0 lladdr 00:00:0c:76:3f:85 ref 5 used 12/13/20 nud reachable  
　6.4.ip neighbour flush--清除邻接条目. 缩写：flush、f  
　　示例1: (-s 可以显示详细信息)  
　　# ip -s -s n f 193.233.7.254

　　7. 路由表管理  
　7.1.缩写 route、ro、r  
　7.5.路由表  
　　从Linux-2.2开始，内核把路由归纳到许多路由表中，这些表都进行了编号，编号数字的范围是1到255。另外，  
　　为了方便，还可以在/etc/iproute2/rt\_tables中为路由表命名。  
　　默认情况下，所有的路由都会被插入到表main(编号254)中。在进行路由查询时，内核只使用路由表main。  
　7.6.ip route add -- 添加新路由  
　　ip route change -- 修改路由  
　　ip route replace -- 替换已有的路由  
　　缩写：add、a；change、chg；replace、repl  
　　示例1: 设置到网络10.0.0/24的路由经过网关193.233.7.65  
　　# ip route add 10.0.0/24 via 193.233.7.65  
　　示例2: 修改到网络10.0.0/24的直接路由，使其经过设备dummy  
　　# ip route chg 10.0.0/24 dev dummy  
　　示例3: 实现链路负载平衡.加入缺省多路径路由，让ppp0和ppp1分担负载(注意：scope值并非必需，它只不过是告诉内核，这个路由要经过网关而不是直连的。实际上，如果你知道远程端点的地址，使用via参数来设置就更好了)。  
                                  　　# ip route add default scope global nexthop dev ppp0 nexthop dev ppp1  
　　# ip route replace default scope global nexthop dev ppp0 nexthop dev ppp1  
　　示例4: 设置NAT路由。在转发来自192.203.80.144的数据包之前，先进行  
网络  
地址转换，把这个地址转换为193.233.7.83  
　　# ip route add nat 192.203.80.142 via 193.233.7.83  
　　示例5: 实现数据包级负载平衡,允许把数据包随机从多个路由发出。weight 可以设置权重.  
　　# ip route replace default equalize nexthop via 211.139.218.145  
dev eth0 weight 1 nexthop via 211.139.218.145 dev eth1 weight 1  
　7.7.ip route delete-- 删除路由  
　　缩写：delete、del、d  
　　示例1:删除上一节命令加入的多路径路由  
　　# ip route del default scope global nexthop dev ppp0 nexthop dev ppp1  
　7.8.ip route show -- 列出路由  
　　缩写：show、list、sh、ls、l  
　　示例1: 计算使用gated/bgp协议的路由个数  
　　# ip route ls proto gated/bgp 　wc  
　　1413 9891 79010  
　　示例2: 计算路由缓存里面的条数，由于被缓存路由的属性可能大于一行，以此需要使用-o选项  
　　# ip -o route ls cloned 　wc  
　　159 2543 18707  
　　示例3: 列出路由表TABLEID里面的路由。缺省设置是table main。TABLEID或者是一个真正的路由表ID或者是/etc/iproute2/rt\_tables文件定义的字符串，  
　　或者是以下的特殊值：  
　　all -- 列出所有表的路由；  
　　cache -- 列出路由缓存的内容。  
　　ip ro ls 193.233.7.82 tab cache  
　　示例4: 列出某个路由表的内容  
　　# ip route ls table fddi153  
　　示例5: 列出默认路由表的内容  
　　# ip route ls  
　　这个命令等于传统的: route  
　7.9.ip route flush -- 擦除路由表  
　　示例1: 删除路由表main中的所有网关路由（示例：在路由监控程序挂掉之后）：  
　　# ip -4 ro flush scope global type unicast  
　　示例2:清除所有被克隆出来的IPv6路由：  
　　# ip -6 -s -s ro flush cache  
　　示例3: 在gated程序挂掉之后，清除所有的BGP路由：  
　　# ip -s ro f proto gated/bgp  
　　示例4: 清除所有ipv4路由cache  
　　# ip route flush cache  
　　\*\*\* IPv4 routing cache is flushed.  
　7.10 ip route get -- 获得单个路由 .缩写：get、g  
　　使用这个命令可以获得到达目的地址的一个路由以及它的确切内容。  
　　ip route get命令和ip route show命令执行的操作是不同的。ip route show命令只是显示现有的路由，而ip route get命令在必要时会派生出新的路由。  
　　示例1: 搜索到193.233.7.82的路由  
　　# ip route get 193.233.7.82  
　　193.233.7.82 dev eth0 src 193.233.7.65 realms inr.ac cache mtu 1500 rtt 300  
　　示例2: 搜索目的地址是193.233.7.82，来自193.233.7.82，从eth0设备到达的路由（这条命令会产生一条非常有意思的路由，这是一条到193.233.7.82的回环路由）  
　　# ip r g 193.233.7.82 from 193.233.7.82 iif eth0  
　　193.233.7.82 from 193.233.7.82 dev eth0 src 193.233.7.65 realms inr.ac/inr.ac  
　　cache ; mtu 1500 rtt 300 iif eth0

　　8. ip route -- 路由策略数据库管理命令  
　　命令　　add、delete、show(或者list)  
　　注意：策略路由(policy routing)不等于路由策略(rouing policy)。  
　　在某些情况下，我们不只是需要通过数据包的目的地址决定路由，可能还需要通过其他一些域：源地址、IP协议、传输层端口甚至数据包的负载。  
　　这就叫做：策略路由(policy routing)。  
　　8.5. ip rule add -- 插入新的规则  
　　ip rule delete -- 删除规则  
　　缩写：add、a；delete、del、d  
　　示例1: 通过路由表inr.ruhep路由来自源地址为192.203.80/24的数据包  
　　ip ru add from 192.203.80/24 table inr.ruhep prio 220  
　　示例2:把源地址为193.233.7.83的数据报的源地址转换为192.203.80.144，并通过表1进行路由  
　　ip ru add from 193.233.7.83 nat 192.203.80.144 table 1 prio 320  
　　示例3:删除无用的缺省规则  
　　ip ru del prio 32767  
　　8.7. ip rule show -- 列出路由规则  
　　缩写：show、list、sh、ls、l  
　　示例1: # ip ru ls  
　　0: from all lookup local  
　　32762: from 192.168.4.89 lookup fddi153  
　　32764: from 192.168.4.88 lookup fddi153  
　　32766: from all lookup main  
　　32767: from all lookup 253  
　　  
9. ip maddress -- 多播地址管理  
　　缩写：show、list、sh、ls、l  
　　9.3.ip maddress show -- 列出多播地址  
　　示例1: # ip maddr ls dummy  
　　9.4. ip maddress add -- 加入多播地址  
　　ip maddress delete -- 删除多播地址  
　　缩写：add、a；delete、del、d  
　　使用这两个命令，我们可以添加／删除在  
网络  
接口上监听的链路层多播地址。这个命令只能管理链路层地址。  
　　示例1: 增加 # ip maddr add 33:33:00:00:00:01 dev dummy  
　　示例2: 查看 # ip -O maddr ls dummy  
　　2: dummy  
　　link 33:33:00:00:00:01 users 2 static  
　　link 01:00:5e:00:00:01  
　　示例3: 删除 # ip maddr del 33:33:00:00:00:01 dev dummy  
　　  
10.ip mroute -- 多播路由缓存管理  
　　10.4. ip mroute show -- 列出多播路由缓存条目  
　　缩写：show、list、sh、ls、l  
　　示例1:查看 # ip mroute ls  
　　(193.232.127.6, 224.0.1.39) Iif: unresolved  
　　(193.232.244.34, 224.0.1.40) Iif: unresolved  
　　(193.233.7.65, 224.66.66.66) Iif: eth0 Oifs: pimreg  
　　示例2:查看 # ip -s mr ls 224.66/16  
　　(193.233.7.65, 224.66.66.66) Iif: eth0 Oifs: pimreg  
　　9383 packets, 300256 bytes  
　　11. ip tunnel -- 通道配置  
　　缩写　tunnel、tunl  
　　11.4.ip tunnel add -- 添加新的通道  
　　ip tunnel change -- 修改现有的通道  
　　ip tunnel delete -- 删除一个通道  
　　缩写：add、a；change、chg；delete、del、d  
　　示例1:建立一个点对点通道，最大TTL是32  
　　# ip tunnel add Cisco mode sit remote 192.31.7.104 local 192.203.80.1 ttl 32  
　　11.4.ip tunnel show -- 列出现有的通道  
　　缩写：show、list、sh、ls、l  
　　示例1: # ip -s tunl ls Cisco  
　　12. ip monitor和rtmon -- 状态监视  
　　ip命令可以用于连续地监视设备、地址和路由的状态。这个命令选项的格式有点不同，命令选项的名字叫做monitor，接着是操作对象：  
　　ip monitor [ file FILE ] [ all 　 OBJECT-LIST ]  
　　示例1: # rtmon file /var/log/rtmon.log  
　　示例2: # ip monitor file /var/log/rtmon.log

三、Linux基本网络配置方法

来自：<http://hi.baidu.com/keepnet/blog/item/6c3f9620c6d1b0af4623e8b7.html>

网络信息查看

　　查看网路接口信息

　　1．了解linux中的网络接口设备

　　$/sbin/ifconfig 查看所有活动网络接口信息，其中包括一个lo环回端口。

　　2．查看指定网络接口的信息

　　$/sbin/ifconfig <网络接口名称> 查看指定接口信息。

　　3．查看系统中所有网络接口的信息

　　$/sbin/ifconfig –a 查看所有接口信息，包括非活动状态接口。

　　注：普通用户查看网络接口信息时，需要在命令前加“/sbin”，而管理员却不用。这是由于命令搜索路径的原因，可以通过“$echo $PATH”查看。

　　查看网关地址和路由信息

　　1、查看主机路由信息

　　#/route

　　显示当前linux主机中的路由表信息。

　　#/route ｜grep default

　　在route命令的显示结果中以“default”开始的行显示了，主机的默认网关地址，将route命令和grep命令组合，使用过滤route命令的显示结果只显示默认网关地址。

　　2.测试与其它主机的网络链接

　　$ping 目的主机地址

　　使用ping命令测试与其它主机的网络链接。ctrl+C终止。

　　$ping –c <测试数据包数量> <目的主机地址>

　　使用ping命令发送指定数量的数据包进行网络链接测试。

　　$使用ping命令进行网络链接测试的技巧。

　　注：当ping命令测试不成功时并不能判断当前主机到目标主机的连接是断开的，因为有很多因素，（如网络防火墙等）可以导致网络连接正常时主机之间无法ping通。

　　3、测试与其它主机的网络链接路径

　　$traceroute <目的主机地址>

　　traceroute命令显示当前主机与目的主机之间经过的所，有网络节点的地址，以及当前主机到每个中间结点的连接状态。

　　查看主机名称信息

　　1、查看当前主机名称

　　$Hostname 查看当前的主机名。

　　2、使用dns服务器查询域名

　　$Nslookup

　　交互模式：通常用于对DNS服务器进行测试

　　直接模式：用于查询某个域名对应的ip地址

　　使用网络命令进行网络设置

　　网络设置的方法

　　1、DHCP网络配置

　　$Dhclient 使用dhclient命令可以从DHCP服务器中申请新的网络配置应用到当前主机。

　　注：DHCP服务器通常提供给DHCP客户端最基本的网络配置信息，包括：接口ip地址，默认网关，DNS

　　服务器地址。

　　2、手工网络配置

　　网络接口ip地址和子网掩码

　　默认网关地址

　　当前主机名称

　　DNS服务器地址

　　Ip地址配置命令

　　Ifconfig <网络接口名称> network <子网掩码>

　　例：# ifconfig eth0 192.168.1.222 netmask 255.255.255.0

　　注：手工配置linux网络只限于临时使用，配置不保存。

　　扩展：# ifconfig eth0:1 10.0.0.1 netmask 255.0.0.0 一块网卡配置多个子ip地址。

　　# ifconfig eth0 hw ether MAC地址 修改网卡的mac地址，  
路由配置命令

　　1、删除默认网关路由

　　$Route del default gw <默认网关地址>

　　2、添加默认网关路由

　　$Route add default gw <默认网关地址>

　　例：#route add default gw 192.168.1.1

　　主机名称配置命令

　　# hostname 123

　　将主机暂时名称改为123。

　　DNS域名解析

　　Nslookup

　　>server

　　显示linux系统中使用的DNS服务器地址，也可临时设置生效。

　　修改配置文件进行网络设置

　　使用工具进行网络设置

　　# Netconfig 图形界面设置网络，重启network后生效。

　　界面中需要填写的内容：

　　主机网络接口的ip地址。

　　网络接口的子网掩码。

　　默认网关地址。

　　主DNS服务器的ip地址。

　　网络服务启动脚本

　　/etc/init.d/network 从新启动才能生效

　　扩展：# /etc/rc.d/init.d/network start 启动network

　　# /etc/rc.d/init.d/network stop 停止network

　　# /etc/rc.d/init.d/network restart 重启network

　　网络接口配置文件

　　/etc/sysconfig/network-scripts/ifconfg-\*

　　DEVICE=eth0 网卡编号

　　ONBOOT=yes 开机时是否启动网卡。

　　BOOTPROTO=static 静态ip或者DHCP动态获取。

　　IPADDR=192.168.1.163 ip

　　NETMASK=255.255.255.0 掩码

　　GATEWAY=192.168.1.1 网关

　　以下为动态获取方式：

　　DEVICE=eth0

　　ONBOOT=yes

　　BOOTPROTO=dhcp

　　停止和启动指定的网络接口

　　单个网络接口的配置进行修改后

　　Ifdown 网络接口名称 停止某个网络接口。

　　Ifup 网络接口名称 开启某个网络接口。

　　注：单个网络接口的配置文件进行修改后，不必每次都从新启动network服务使配置文件生效，可用以上命令。

　　扩展：对网络接口操作：

　　# ifconfig eth0 down 停止eth0接口

　　# ifconfig eth0 up 开启eth0接口

　　可以连续使用

　　Ifdown eth0 ; ifup eth0

　　主机名称配置文件

　　/etc/sysconfig/network

　　例：# vi /etc/sysconfig/network 使用vi编辑器打开主机名配置文件。

　　NETWORKING=yes 网络是否可用。

　　HOSTNAME=xxxx xxxx为新设置的主机名。

　　本地主机名称解析文件

　　/etc/hosts

　　Hosts 和 DNS具有类似的主机名称解析功能

　　域名服务器配置文件

　　/etc/resolv.conf

　　最多可以设置3行，前面的生效。  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

网络配置文件    LINUX 在 安装中 基本的配置 都已经完成了    都以文本方式 保存在/etc目录下  
.  
网络配置文件：/etc/sysconfig/network  
查看命令：   
[root@localhost ~]# cat /etc/sysconfig/network  
NETWORKING=yes  
NETWORKING\_IPV6=yes  
HOSTNAME=localhost.localdomain  
[root@localhost ~]#   
可能因为一些LINUX 版本 不同上面显示的信息也不同     这个文件 可以用 /netcfg 来维护 不过需要 X服务器支持  
主机名和IP 地址映射 ： /etc/hosts  
IP地址 的设计方便计算机的处理    但是记忆这些就比较枯燥了 为了 直接能访问主机 现在都采用域名方式 早期的域名就被保存在这个文件里  
查看命令：  
[root@localhost ~]# cat /etc/hosts  
# Do not remove the following line, or various programs  
# that require network functionality will fail.  
::1       localhost.localdomain     localhost  
[root@localhost ~]#   
这个文件和WINDOWS 系统中的 WINDOWS\system32\etc 下的 HOSTS 采用相同的格式和功能  
  
DNS 客户配置：/etc/resolv.conf  
这个文件包含了域名顺序和DNS 服务器的主机地址  
查看命令：   
[root@localhost ~]# cat /etc/resolv.conf  
  
nameserver 61.128.128.68          ；可使用多个DNS 服务器的地址  
nameserver 61.128.192.68  
[root@localhost ~]#   
  
主机名：/etc/hostname        ；有些版本的LINUX 主机名是在/etc/sysconfig/network 文件里    有些除外  
查看主机名：  
[root@localhost ~]# hostname  
localhost.localdomain  
[root@localhost ~]#   
网络服务：/etc/services  
这个文件跟 WINDOWS 下的 WINDOWS\system32\drivers\etc\services 文件一样 给出了端口 和 服务名对照    很多系统都会读取这个文件  
  
查看命令：   
[root@localhost ~]# tail /etc/services         ；这里如果使用 cat 命令查看    会有很多    
nimspooler        48001/udp                         # Nimbus Spooler  
nimhub            48002/tcp                         # Nimbus Hub  
nimhub            48002/udp                         # Nimbus Hub  
nimgtw            48003/tcp                         # Nimbus Gateway  
nimgtw            48003/udp                         # Nimbus Gateway  
com-bardac-dw     48556/tcp                         # com-bardac-dw  
com-bardac-dw     48556/udp                         # com-bardac-dw  
iqobject          48619/tcp                         # iqobject  
iqobject          48619/udp                         # iqobject  
# Local services  
[root@localhost ~]#   
  
网络协议文件：/etc/protocols  
这个文件列出了 用户系统支持    TCP／IP 各种协议名称  
查看命令：  
[root@localhost ~]# tail /etc/protocols           ；这里如果使用 cat 命令查看    会有很多   
sctp      132       SCTP              # Stream Control Transmission Protocol  
fc        133       FC                # Fibre Channel  
rsvp-e2e-ignore 134       RSVP-E2E-IGNORE  
#         135                       # Mobility Header  
udplite 136       UDPLite  
mpls-in-ip        137       MPLS-in-IP  
#     138-252 Unassigned                                         [IANA]  
#     253       Use for experimentation and testing             [RFC3692]  
#     254       Use for experimentation and testing             [RFC3692]  
#     255                   Reserved                               [IANA]  
[root@localhost ~]#   
  
禁止访问服务 和准许访问服务 ：/etc/hosts.deny 和 /etc/hosts.allow  
这个很简单     前者是    禁止访问的网络服务    后者正好相反   
  
网卡设备 ：cat /etc/sysconfig/network-scripts/ifcfg-eth0  
查看命令：   
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0   
# Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+  
DEVICE=eth0  
BOOTPROTO=dhcp  
HWADDR=00:E0:4C:B0:4F:59  
ONBOOT=yes  
TYPE=Ethernet  
[root@localhost ~]#   
  
注意如果配置文件    改变 需要重新启动服务才可以生效  
命令：   
[root@localhost ~]#    service network restart  
正在关闭接口 china：                                         [确定]  
正在关闭接口 eth0：                                          [确定]  
关闭环回接口：                                               [确定]  
弹出环回接口：                                               [确定]  
弹出界面 eth0：   
正在决定 eth0 的 IP 信息...  
  
网络端口配置 ifconfig  
查看命令：  
[root@localhost ~]# ifconfig  
eth0        Link encap:Ethernet    HWaddr 00:E0:4C:B0:4F:59    
inet6 addr: fe80::2e0:4cff:feb0:4f59/64 Scope:Link                       ；主机地址信息  
UP BROADCAST RUNNING MULTICAST    MTU:1500    Metric:1    ；MTU最大传输单元 metric 量度值  
RX packets:5719 errors:0 dropped:0 overruns:0 frame:0     ； RX    接收包   
TX packets:5838 errors:0 dropped:0 overruns:0 carrier:0      ；TX 传输包  
collisions:0 txqueuelen:1000   
RX bytes:3664215 (3.4 MiB)    TX bytes:671630 (655.8 KiB)  
Interrupt:153 Base address:0x6000   
  
lo          Link encap:Local Loopback    
inet addr:127.0.0.1    Mask:255.0.0.0  
inet6 addr: ::1/128 Scope:Host  
UP LOOPBACK RUNNING    MTU:16436    Metric:1  
RX packets:8835 errors:0 dropped:0 overruns:0 frame:0  
TX packets:8835 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0   
RX bytes:6032605 (5.7 MiB)    TX bytes:6032605 (5.7 MiB)  
  
ppp0        Link encap:Point-to-Point Protocol    
inet addr:219.152.75.238    P-t-P:219.152.72.1    Mask:255.255.255.255  
UP POINTOPOINT RUNNING NOARP MULTICAST    MTU:1492    Metric:1  
RX packets:25 errors:0 dropped:0 overruns:0 frame:0  
TX packets:25 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:3   
RX bytes:1197 (1.1 KiB)    TX bytes:1705 (1.6 KiB)  
  
[root@localhost ~]#   
；errors 错误    collisions 冲突     如果他们的数据 出现得太高 很可能是 硬件 或者网络介质损坏 如果看到接受和传输包的个数 异常    应该是 IP 出现混乱  
          Interrupt:153 Base address:0x6000    网卡的 中断号和 基地址  
eth0 是网卡    lo 是网络回送设备    eth0 对应MAC 地址    00:E0:4C:B0:4F:59       这个是由生产出来就有 不过可以通过 /etc/rc.d/init.d 中network 加入    ifconfig eth0 hw ether XX:XX:XX:XX:XX  
  
PPP0    点对点协议。。。 因为我是拨号连接。。。所以 有这个。。。  
  
启用和 禁用网络接口  
命令用法    
ifconfig interface [up    and down]  
这个命令要求 ROOT    用户。使用  
UP 是 开启     DOWN 是禁止  
例子：   
ifconfig eth0 down  
ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up