使用下面的 route 命令可以查看 **[Linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "https://www.cnblogs.com/baiduboy/p/_blank)** 内核路由表。

**[cpp]** [view plain](http://blog.csdn.net/vevenlcf/article/details/48026965" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/vevenlcf/article/details/48026965" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # route
2. Destination     Gateway         Genmask Flags Metric Ref    Use Iface
3. 192.168.0.0     \*               255.255.255.0   U     0      0        0 eth0
4. 169.254.0.0     \*               255.255.0.0     U     0      0        0 eth0
5. **default         192.168.0.1     0.0.0.0         UG    0      0        0 eth0**

route 命令的输出项说明

输出项 说明

|  |  |
| --- | --- |
| Destination | 目标网段或者主机 |
| Gateway | 网关地址，”\*” 表示目标是本主机所属的网络，不需要路由 |
| Genmask | 网络掩码 |
| Flags | 标记。一些可能的标记如下： |
|  | U — 路由是活动的 |
|  | H — 目标是一个主机 |
|  | G — 路由指向网关 |
|  | R — 恢复动态路由产生的表项 |
|  | D — 由路由的后台程序动态地安装 |
|  | M — 由路由的后台程序修改 |
|  | ! — 拒绝路由 |
| Metric | 路由距离，到达指定网络所需的中转数（linux 内核中没有使用） |
| Ref | 路由项引用次数（linux 内核中没有使用） |
| Use | 此路由项被路由软件查找的次数 |
| Iface | 该路由表项对应的输出接口 |

### **3 种路由类型**

#### **主机路由**

主机路由是路由选择表中指向单个IP地址或主机名的路由记录。主机路由的Flags字段为H。例如，在下面的示例中，本地主机通过IP地址192.168.1.1的路由器到达IP地址为10.0.0.10的主机。

Destination Gateway Genmask Flags Metric Ref Use Iface

----------- ------- ------- ----- ------ --- --- -----

10.0.0.10 192.168.1.1 255.255.255.255 UH 0 0 0 eth0

#### **网络路由**

网络路由是代表主机可以到达的网络。网络路由的Flags字段为N。例如，在下面的示例中，本地主机将发送到网络192.19.12的数据包转发到IP地址为192.168.1.1的路由器。

Destination Gateway Genmask Flags Metric Ref Use Iface

----------- ------- ------- ----- ----- --- --- -----

192.19.12 192.168.1.1 255.255.255.0 UN 0 0 0 eth0

#### **默认路由**

当主机不能在路由表中查找到目标主机的IP地址或网络路由时，数据包就被发送到默认路由（默认网关）上。默认路由的Flags字段为G。例如，在下面的示例中，默认路由是IP地址为192.168.1.1的路由器。

Destination Gateway Genmask Flags Metric Ref Use Iface

----------- ------- ------- ----- ------ --- --- -----

default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0

## **配置静态路由**

### **route 命令**

设置和查看路由表都可以用 route 命令，设置内核路由表的命令格式是：

# route [add|del] [-net|-host] target [netmask Nm] [gw Gw] [[dev] If]

其中：

* add : 添加一条路由规则
* del : 删除一条路由规则
* -net : 目的地址是一个网络
* -host : 目的地址是一个主机
* target : 目的网络或主机
* netmask : 目的地址的网络掩码
* gw : 路由数据包通过的网关
* dev : 为路由指定的网络接口

### **route 命令使用举例**

添加到主机的路由

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # route add -host 192.168.1.2 dev eth0
2. # route add -host 10.20.30.148 gw 10.20.30.40     #添加到10.20.30.148的网管

# route add -host 192.168.1.2 dev eth0

# route add -host 10.20.30.148 gw 10.20.30.40 #添加到10.20.30.148的网管

添加到网络的路由

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # route add -net 10.20.30.40 netmask 255.255.255.248 eth0   #添加10.20.30.40的网络
2. # route add -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41 #添加10.20.30.48的网络
3. # route add -net 192.168.1.0/24 eth1

# route add -net 10.20.30.40 netmask 255.255.255.248 eth0 #添加10.20.30.40的网络

# route add -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41 #添加10.20.30.48的网络

# route add -net 192.168.1.0/24 eth1

添加默认路由

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # route add default gw 192.168.1.1

# route add default gw 192.168.1.1

删除路由

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # route del -host 192.168.1.2 dev eth0:0
2. # route del -host 10.20.30.148 gw 10.20.30.40
3. # route del -net 10.20.30.40 netmask 255.255.255.248 eth0
4. # route del -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41
5. # route del -net 192.168.1.0/24 eth1
6. # route del default gw 192.168.1.1

# route del -host 192.168.1.2 dev eth0:0

# route del -host 10.20.30.148 gw 10.20.30.40

# route del -net 10.20.30.40 netmask 255.255.255.248 eth0

# route del -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41

# route del -net 192.168.1.0/24 eth1

# route del default gw 192.168.1.1

### **设置包转发**

在 CentOS 中默认的内核配置已经包含了路由功能，但默认并没有在系统启动时启用此功能。开启 **[linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "https://www.cnblogs.com/baiduboy/p/_blank)** 的路由功能可以通过调整内核的网络参数来实现。要配置和调整内核参数可以使用 sysctl 命令。例如：要开启 Linux 内核的数据包转发功能可以使用如下的命令。

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # sysctl -w net.ipv4.ip\_forward=1

# sysctl -w net.ipv4.ip\_forward=1

这样设置之后，当前系统就能实现包转发，但下次启动计算机时将失效。为了使在下次启动计算机时仍然有效，需要将下面的行写入配置文件/etc/sysctl.conf。

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # vi /etc/sysctl.conf
2. net.ipv4.ip\_forward = 1

# vi /etc/sysctl.conf

net.ipv4.ip\_forward = 1

用户还可以使用如下的命令查看当前系统是否支持包转发。

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. # sysctl net.ipv4.ip\_forward

# sysctl net.ipv4.ip\_forward

route 命令：

Linux系统的route命令用于显示和操作IP路由表（show / manipulate the IP routing table）。要实现两个不同的子网之间的通信，需要一台连接两个网络的路由器，或者同时位于两个网络的网关来实现。在Linux系统中，设置路由通常是为了解决以下问题：该Linux系统在一个局域网中，局域网中有一个网关，能够让机器访问Internet，那么就需要将这台机器的IP地址设置为Linux机器的默认路由。要注意的是，直接在命令行下执行route命令来添加路由，不会永久保存，当网卡重启或者机器重启之后，该路由就失效了；可以在/etc/rc.local中添加route命令来保证该路由设置永久有效。

**1．命令格式：**

route [-f] [-p] [Command [Destination] [mask Netmask] [Gateway] [metric Metric]] [if Interface]]

**2．命令功能：**

Route命令是用于操作基于内核ip路由表，它的主要作用是创建一个静态路由让指定一个主机或者一个网络通过一个网络接口，如eth0。当使用"add"或者"del"参数时，路由表被修改，如果没有参数，则显示路由表当前的内容。

**3．命令参数：**

-c 显示更多信息

-n 不解析名字

-v 显示详细的处理信息

-F 显示发送信息

-C 显示路由缓存

-f 清除所有网关入口的路由表。

-p 与 add 命令一起使用时使路由具有永久性。

add:添加一条新路由。

del:删除一条路由。

-net:目标地址是一个网络。

-host:目标地址是一个主机。

netmask:当添加一个网络路由时，需要使用网络掩码。

gw:路由数据包通过网关。注意，你指定的网关必须能够达到。

metric：设置路由跳数。

Command 指定您想运行的命令 (Add/Change/Delete/Print)。

Destination 指定该路由的网络目标。

mask Netmask 指定与网络目标相关的网络掩码（也被称作子网掩码）。

Gateway 指定网络目标定义的地址集和子网掩码可以到达的前进或下一跃点 IP 地址。

metric Metric 为路由指定一个整数成本值标（从 1 至 9999），当在路由表(与转发的数据包目标地址最匹配)的多个路由中进行选择时可以使用。

if Interface 为可以访问目标的接口指定接口索引。若要获得一个接口列表和它们相应的接口索引，使用 route print 命令的显示功能。可以使用十进制或十六进制值进行接口索引。

**4．使用实例：**

**实例1：显示当前路由**

**命令：**

route

route -n

**输出：**

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. [root@localhost ~]# route
2. Kernel IP routing table
3. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
4. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
5. e192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
6. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
7. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**
8. [root@localhost ~]# route -n
9. Kernel IP routing table
10. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
11. 192.168.120.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
12. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
13. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
14. 0.0.0.0         192.168.120.240 0.0.0.0         UG    0      0        0 eth0

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

e192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

0.0.0.0 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

**说明：**

第一行表示主机所在网络的地址为192.168.120.0，若数据传送目标是在本局域网内通信，则可直接通过eth0转发数据包;

第四行表示数据传送目的是访问Internet，则由接口eth0，将数据包发送到网关192.168.120.240

其中Flags为路由标志，标记当前网络节点的状态。

Flags标志说明：

U Up表示此路由当前为启动状态

H Host，表示此网关为一主机

G Gateway，表示此网关为一路由器

R Reinstate Route，使用动态路由重新初始化的路由

D Dynamically,此路由是动态性地写入

M Modified，此路由是由路由守护程序或导向器动态修改

! 表示此路由当前为关闭状态

备注：

route -n (-n 表示不解析名字,列出速度会比route 快)

**实例2：添加网关/设置网关**

**命令：**

route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0

**输出：**

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. [root@localhost ~]# route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0
2. [root@localhost ~]# route
3. Kernel IP routing table
4. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
5. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
6. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
7. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
8. 224.0.0.0       \*               240.0.0.0       U     0      0        0 eth0
9. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**

[root@localhost ~]# route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

224.0.0.0 \* 240.0.0.0 U 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]#

**说明：**

增加一条 到达244.0.0.0的路由

**实例3：屏蔽一条路由**

**命令：**

route add -net 224.0.0.0 netmask 240.0.0.0 reject

**输出：**

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. [root@localhost ~]# route add -net 224.0.0.0 netmask 240.0.0.0 reject
2. [root@localhost ~]# route
3. Kernel IP routing table
4. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
5. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
6. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
7. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
8. 224.0.0.0       -               240.0.0.0       !     0      -        0 -
9. 224.0.0.0       \*               240.0.0.0       U     0      0        0 eth0
10. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**

[root@localhost ~]# route add -net 224.0.0.0 netmask 240.0.0.0 reject

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

224.0.0.0 - 240.0.0.0 ! 0 - 0 -

224.0.0.0 \* 240.0.0.0 U 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

**说明：**

增加一条屏蔽的路由，目的地址为 224.x.x.x 将被拒绝

**实例4：删除路由记录**

**命令：**

route del -net 224.0.0.0 netmask 240.0.0.0

route del -net 224.0.0.0 netmask 240.0.0.0 reject

**输出：**

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. [root@localhost ~]# route
2. Kernel IP routing table
3. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
4. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
5. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
6. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
7. 224.0.0.0       -               240.0.0.0       !     0      -        0 -
8. 224.0.0.0       \*               240.0.0.0       U     0      0        0 eth0
9. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**
10. [root@localhost ~]# route del -net 224.0.0.0 netmask 240.0.0.0
11. [root@localhost ~]# route
12. Kernel IP routing table
13. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
14. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
15. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
16. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
17. 224.0.0.0       -               240.0.0.0       !     0      -        0 -
18. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**
19. [root@localhost ~]# route del -net 224.0.0.0 netmask 240.0.0.0 reject
20. [root@localhost ~]# route
21. Kernel IP routing table
22. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
23. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
24. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
25. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
26. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**
27. [root@localhost ~]#

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

224.0.0.0 - 240.0.0.0 ! 0 - 0 -

224.0.0.0 \* 240.0.0.0 U 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]# route del -net 224.0.0.0 netmask 240.0.0.0

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

224.0.0.0 - 240.0.0.0 ! 0 - 0 -

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]# route del -net 224.0.0.0 netmask 240.0.0.0 reject

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]#

**说明：**

**实例5：删除和添加设置默认网关**

**命令：**

route del default gw 192.168.120.240

route add default gw 192.168.120.240

**输出：**

**[cpp]** [view plain](http://blog.csdn.net/chenlycly/article/details/52141854" \o "view plain" \t "https://www.cnblogs.com/baiduboy/p/_blank) [copy](http://blog.csdn.net/chenlycly/article/details/52141854" \o "copy" \t "https://www.cnblogs.com/baiduboy/p/_blank) [print](http://blog.csdn.net/chenlycly/article/details/52141854" \o "print" \t "https://www.cnblogs.com/baiduboy/p/_blank)[?](http://blog.csdn.net/chenlycly/article/details/52141854" \o "?" \t "https://www.cnblogs.com/baiduboy/p/_blank)

1. [root@localhost ~]# route del **default gw 192.168.120.240**
2. [root@localhost ~]# route
3. Kernel IP routing table
4. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
5. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
6. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
7. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
8. [root@localhost ~]# route add **default gw 192.168.120.240**
9. [root@localhost ~]# route
10. Kernel IP routing table
11. Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
12. 192.168.120.0   \*               255.255.255.0   U     0      0        0 eth0
13. 192.168.0.0     192.168.120.1   255.255.0.0     UG    0      0        0 eth0
14. 10.0.0.0        192.168.120.1   255.0.0.0       UG    0      0        0 eth0
15. **default         192.168.120.240 0.0.0.0         UG    0      0        0 eth0**
16. [root@localhost ~]#

[root@localhost ~]# route del default gw 192.168.120.240

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

[root@localhost ~]# route add default gw 192.168.120.240

[root@localhost ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0

192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0

10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0

default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

[root@localhost ~]#