

Linux 网络配置



Linux 网络配置



基础术语

1. static(静态): 手工配置 IP 地址及其他信息
2. DHCP(动态): 基于 IP/UDP 协议的协议, 通过相关的服务可以动态获取 IP 地址及其他信息。一般包括 DHCP Server 和 DHCP Client

Linux 网络配置

基础术语

3. BOOTP(动态): 基于 IP/UDP 协议的协议, 它可以让无盘站从一个中心服务器上获得 IP 地址, 为局域网中的无盘工作站分配动态 IP 地址, 并不需要每个用户去设置静态 IP 地址。使用 BOOTP 协议的时候, 一般包括 Bootstrap Protocol Server (自举协议服务端) 和 Bootstrap Protocol Client (自举协议客户端) 两部分。

Linux 网络配置

基础术语

- 4. DNS(Domain Name Sytem): 将 IP 地址解析为 FQDN(Fully Qualified Domain Name) 名称, 反之亦可。
- 5. 路由 (Route): 从一个节点到另一个节点的路径
- 6. 路由器 (Router): 寻找、计算最佳路径的设备

Linux 网络配置

基础术语

7. 网关 (Gateway): 本地网络到远程网络的出口地址

8. NetBIOS: 网络基本输入输出协议

9. NetBEUI: 网络基础输入输出增强型用户接口

Linux 网络配置

基础术语

10.WINS(Windows Inetnet Name Server):
NetBIOS 名称到 IP 地址的解析，反之亦可。

11.**hosts 文件**：在本地无 DNS 时可以在本地设备中通过配置，完成 **DNS** 名称到 IP 地址的解析

12.lmhosts 文件在本地无 WINS 时可以在本地设备中通过配置，完成 WINS 名称到 IP 地址的解析

Linux 网络配置

基础术语

Linux 网卡说明

en: 为 ethernet 类型

wl: 为 wlan 类型

ww: 为 wwan 类型



Linux 网络配置

基础术语

Linux 网卡说明

1. eno1

此为合并固件或者 BIOS 的名称为板载设备提供索引号的名称（嵌入设备或板载设备）。同时使用方案 2 作为备选

Linux 网络配置

基础术语

Linux 网卡说明

2. ens1

合并固件或者 BIOS 的名称提供 PCI 快捷热插拔插槽索引号的名称。如果固件信息适用且可用，则 systemd 默认情况下根据这个方案命名接口，同时使用方案 3 作为备选。

Linux 网络配置

基础术语

Linux 网卡说明

3. enp0s1

合并硬件连接器物理位置的名称。如果固件信息适用且可用，则 `systemd` 默认情况下根据这个方案命名接口，同时使用方案 5 作为备选。

Linux 网络配置

基础术语

Linux 网卡说明

4.enx78e7d1ea46da

合并接口 MAC 地址的名称。默认情况下 systemd 不会根据这个方案命名接口，但如果需要可以启用这个功能。

Linux 网络配置

基础术语

Linux 网卡说明

5.eth0

传统的不可预测内核属性 ethX 命名。如果其他所有方法都失败，systemd 就根据这个方案命名接口。

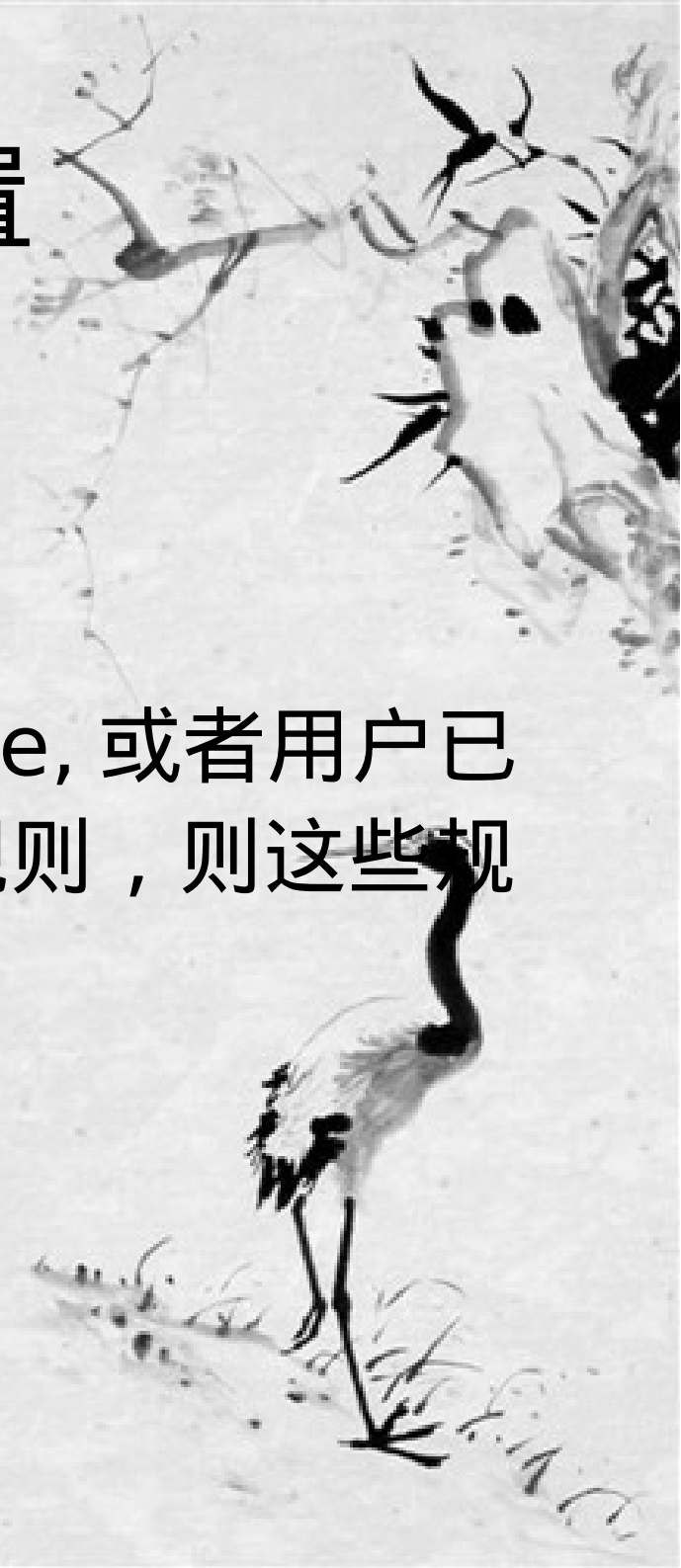


Linux 网络配置

基础术语

Linux 网卡说明

如果该系统启用了 biosdevname, 或者用户已添加更改内核设备名称的 udevd 规则, 则这些规则将替代默认的 systemd 策略。



Linux 网络配置

基础术语

Linux 网卡说明

命令 : `biosdevname`

功能 : 显示 bios 对设备的命名规范

语法格式 : `biosdevname [选项] [参数]`



Linux 网络配置

基础术语

Linux 网卡说明

biosdevname 对网卡的命名

Device	Old Name	New Name
Embedded network interface (LOM)	<code>eth[0123...]</code>	<code>em[1234...]</code> ^[a]
PCI card network interface	<code>eth[0123...]</code>	<code>p<slot>p<ethernet port></code> ^[b]
Virtual function	<code>eth[0123...]</code>	<code>p<slot>p<ethernet port>_<virtual interface></code> ^[c]
^[a] New enumeration starts at 1 . ^[b] For example: <code>p3p4</code> ^[c] For example: <code>p3p4_1</code>		

Linux 网络配置

基础术语

Linux 网卡说明

示例

1. 查看本机网卡

```
#biosdevname -i enp0s3  
P2p1
```

2. 关闭 biosdevname

```
#biosdevname=0
```



Linux 网络配置

基础术语

Linux 网卡说明

示例

3. 开启 biosdevname

`#biosdevname=1`

4. 将关闭 / 开启 biosdevname 传递给 kernel 参数

`/* 修改 grub, 将 biosdevname=0|1 添加至 kernel 字段`

Linux 网络配置

基础术语

Linux 网卡说明

udev

是 Linux kernel 2.6 系列的设备管理器。

它主要的功能是管理 /dev 目录底下的设备节点。

它同时也是用来接替 devfs 及 hotplug 的功能，这意味着它要在添加 / 删除硬件时处理 /dev 目录以及所有用户空间的行为，包括加载 firmware 时。

Linux 网络配置

基础术语

Linux 网卡说明

udev

udev 的最新版本依赖于升级后的 Linux kernel 2.6.13 的 uevent 接口的最新版本。使用新版本 udev 的系统不能在 2.6.13 以下版本启动，除非使用 noudev 参数来禁用 udev 并使用传统的 `/dev` 来进行设备读取。

Linux 网络配置

基础术语

Linux 网卡说明

udev

Linux 传统上使用静态设备创建方法，因此大量设备节点在 `/dev` 下创建（有时上千个），而不管相应的硬件设备是否真正存在。通常这由一个 `MAKEDEV` 脚本实现，这个脚本包含了许多通过世界上每一个可能存在的设备相关的主设备号和次设备号对 `mknod` 程序的调用。

Linux 网络配置

基础术语

Linux 网卡说明

udev

采用 udev 的方法，只有被内核检测到的设备才会获取为它们创建的设备节点。因为这些设备节点在每次系统启动时被创建，他们会被贮存在 ramfs(一个内存中的文件系统，不占用任何磁盘空间)。设备节点不需要大量磁盘空间，因此它使用的内存可以忽略。

Linux 网络配置

udev 配置

```
#ls -l /etc/udev/udev.conf
```

udev 规则

```
#cd /etc/udev/rules.d
```

```
#ls
```



Linux 网络配置

udev 规则说明

1. 操作符

“==”：比较键、值，若等于，则该条件满足；

“!=”：比较键、值，若不等于，则该条件满足；

“=”：对一个键赋值；



Linux 网络配置



udev 规则说明

1. 操作符

“+=”：为一个表示多个条目的键赋值。

“:=”：对一个键赋值，并拒绝之后所有对该键的改动。目的是防止后面的规则文件对该键赋值。



Linux 网络配置

udev 规则说明

2. 规则匹配建

ACTION: 事件 (uevent) 的行为, 例如 :
add(添加设备)、 remove(删除设备)。

KERNEL: 内核设备名称, 例如 : sda, cdrom。

DEVPATH: 设备的 devpath 路径。

Linux 网络配置

udev 规则说明

2. 规则匹配建

SUBSYSTEM: 设备的子系统名称，例如：sda 的子系统为 block。

BUS: 设备在 devpath 里的总线名称，例如：usb。

DRIVER: 设备在 devpath 里的设备驱动名称，例如：ide-cdrom。

Linux 网络配置

udev 规则说明

2. 规则匹配建

ID: 设备在 devpath 里的识别号。

SYSFS{filename}: 设备的 devpath 路径下设备的属性文件“filename”里的内容。

ENV{key}: 环境变量。在一条规则中，可以设定最多五条环境变量的 匹配键。

Linux 网络配置




udev 规则说明

2. 规则匹配建

PROGRAM: 调用外部命令。

RESULT: 外部命令 PROGRAM 的返回结果

SYMLINK: 为 /dev/ 下的设备文件产生符号链接。
由于 udev 只能为某个设备产生一个设备文件，
所以为了不覆盖系统默认的 udev 规则所产生的
文件，推荐使用符号链接。



Linux 网络配置

udev 规则说明

2. 规则匹配建

NAME: 在 `/dev` 下产生的设备文件名。只有第一次对某个设备的 NAME 的赋值行为生效，之后匹配的规则再对该设备的 NAME 赋值行为将被忽略。如果没有任何规则对设备的 NAME 赋值，udev 将使用内核设备名称来产生设备文件。

Linux 网络配置

udev 规则说明

2. 规则匹配建

OWNER, GROUP, MODE: 为设备设定权限。

ENV{key}: 导入一个环境变量。



Linux 网络配置

udev 规则说明

3. 值

Linux 用户可以随意地定制 udev 规则文件的值 (value)。如：

my_root_disk, my_printer

同时也可以引用下面的替换操作符



Linux 网络配置

udev 规则说明

3. 值

\$kernel, %k: 设备的内核设备名称
例如 :sda、cdrom。

\$number, %n: 设备的内核号码
例如 : sda3 的内核号码是 3。

\$devpath, %p: 设备的 devpath 路径。

\$id, %b: 设备在 devpath 里的 ID 号。



Linux 网络配置

udev 规则说明

3. 值

`$sysfs{file}, %s{file}` : 设备的 `sysfs` 里 `file` 的内容。其实就是设备的属性值。例如：

`$sysfs{size}` 表示该设备（ 磁盘 ）的大小。

`$env{key}, %E{key}`: 一个环境变量的值。

`$major, %M`: 设备的 `major` 号。

Linux 网络配置

udev 规则说明

3. 值

\$minor %m: 设备的 minor 号。

\$result, %c:PROGRAM 返回的结果。

\$parent, %P: 父设备的设备文件名。

\$root, %r:udev_root 的值，默认是 /dev/。

\$tempnode, %N: 临时设备名。

Linux 网络配置

udev 规则说明

3. 值

%% : 符号 % 本身。

\$\$: 符号 \$ 本身。



Linux 网络配置

基础术语

Linux 网卡说明

udev

udev 则是根据一定的规则在 `/dev/` 目录下生成设备文件的后台服务进程。

如当插入一个 U 盘，udev 服务就会自动生成 `/dev/sda` 这样的文件。



Linux 网络配置

网络基础配置

命令 :ifconfig

功能：显示 / 配置指定设备的网络地址等信息

选线 :ifconfig [设备文件] [选项] [地址 ...]



Linux 网络配置

网络基础配置

示例

1. 显示所有活动的网络设备信息

```
#ifconfig
```

2. 显示所有网络设备信息（包含未活动）

```
#ifconfig -a
```

3. 显示 enp0s3 的相关信息

```
#ifconfig enp0s3
```



Linux 网络配置

网络基础配置

示例

4. 配置 enp0s3 的 IP 地址

```
#ifconfig enp0s3 192.168.100.1
```

5. 配置 enp0s3 的 IP 地址及子网掩码

```
#ifconfig enp0s3 192.168.100.1/30
```

或

```
#ifconfig enp0s3 192.168.100.1 netmask  
255.255.255.248
```

Linux 网络配置

网络基础配置

示例

6. 修改配置 enp0s3 的 MAC 地址

```
#ifconfig enp0s3 hw ether  
00:01:02:03:04:05
```

7. 启用指定网络接口

```
#ifconfig enp0s3 up
```



Linux 网络配置

网络基础配置

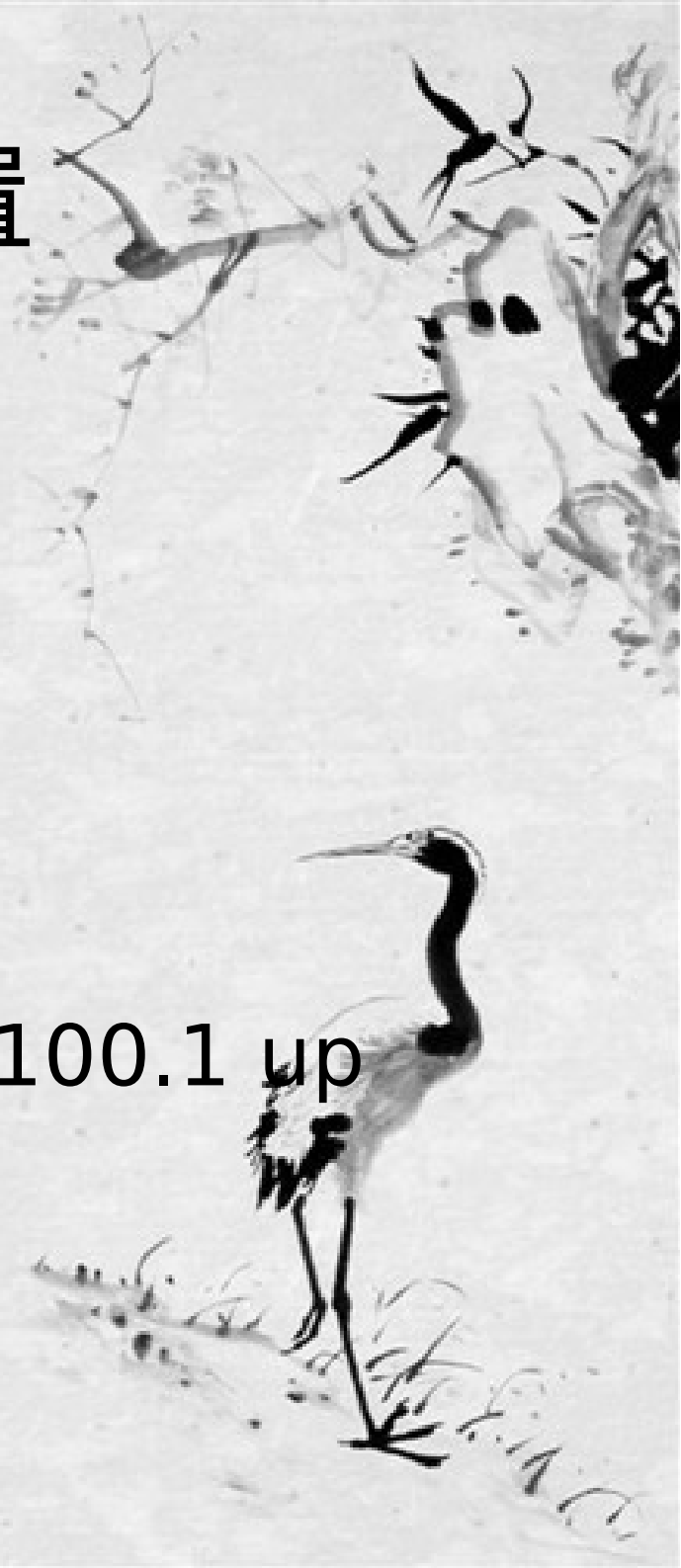
示例

8. 关闭指定网络接口

```
#ifconfig enp0s3 down
```

9. 启用子接口

```
#ifconfig enp0s3:0 192.168.100.1 up
```



Linux 网络配置

网络基础配置

命令 :mii-tool

功能：检查网络链路状态

语法格式 :mii-tool [选项] 网络设备名



Linux 网络配置

网络基础配置

示例

1. 显示 enp0s3 的链路状态

```
#mii-tool enp0s3
```



Linux 网络配置

网络基础配置

命令 :route

功能：显示 / 配置网关信息

语法格式 :route [选项] [网关地址] [设备]



Linux 网络配置

网络基础配置

示例

1. 显示当前设备路由表

```
#route -n
```

Destination: 目标的网络地址

Gateway: 通项这个网络段应该使用哪个网关

Flags: 标记 U: 正在使用状态 G: 网关

Metric: 跳计量 Ref: 为参考 Use: 为使用情况

Iface: 为网关基于在哪个设备上。

Linux 网络配置

网络基础配置

示例

2. 添加一条默认网关

```
#route --add default gw  
gateway_ip_address
```

如

```
#route --add default gw 192.168.100.254
```

3. 删除一条默认网关

```
#route --del default gw 192.168.100.254
```



Linux 网络配置

网络基础配置

示例



4. 添加一条定向目标网络的网关

```
#route add --net 192.168.200.0 netmask  
255.255.255.0 gw 192.168.100.254
```

5. 指定一条定向 host 的网关

```
#route add --host 192.168.200.213  
netmask 255.255.255.0 gw  
192.168.100.254
```



Linux 网络配置

网络基础配置

说明：

当路由表中有定向主机、定向网络、默认网络时，路由在寻址的时候的优先级为

Host > Network > Default



Linux 网络配置

网络基础配置

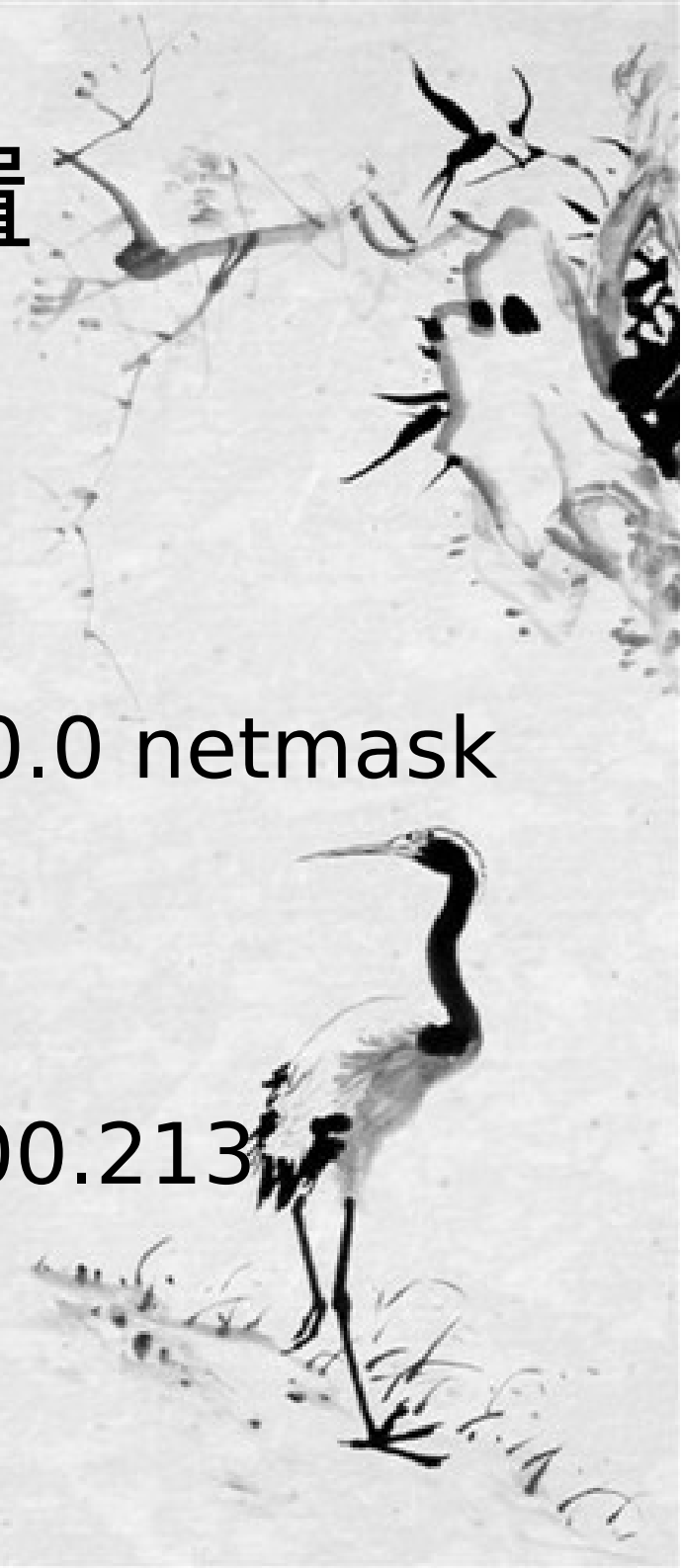
示例

6. 删除一条定向目标网络的网关

```
#route del --net 192.168.200.0 netmask  
255.255.255.0
```

7. 删除一条定向 host 的网关

```
#route del --host 192.168.200.213  
netmask 255.255.255.0
```



Linux 网络配置

网络基础配置

示例

8. 删除一条默认网关

```
#route del default gw 192.168.100.254
```

9. 指定默认路由从 enp1s2 出去

```
#route add default gw 192.168.1.1 dev  
enp1s2
```

Linux 网络配置

网络基础配置

命令 :ip

功能：显示 / 配置网络信息

语法格式 :ip [选项] 对象 [命令 | 帮助]



Linux 网络配置

网络基础配置

命令 :ip

三大选项

Link: 显示 / 设置链路层信息

Address: 显示 / 设置 IP 地址

Route: 显示 / 设置 Route 信息



Linux 网络配置

网络基础配置

示例：

1. 显示当前所有设备的 IP 地址及其他信息

```
#ip addr
```

或

```
#ip addr show
```

2. 显示指定设备的 IP 地址及其他信息

```
#ip addr show enp0s3
```



Linux 网络配置

网络基础配置

示例：

3. 显示当前所有设备的链路状态

```
#ip link
```

或

```
#ip link show
```

4. 显示指定设备的链路状态

```
#ip link show enp0s3
```



Linux 网络配置

网络基础配置

示例：

5. 显示路由的信息

```
#ip route
```

6. 显示所有的设备的网络统计数据

```
#ip -s link
```

7. 显示指定设备的网络统计数据

```
#ip -s link show enp0s3
```



Linux 网络配置

网络基础配置

示例：

8. 开启 / 关闭指定设备

```
#ip link set enp0s3 up
```

```
#ip link set enp0s3 down
```

9. 设置 MTU 值为 1500bytes

```
#ip link set enp0s3 mtu 1500
```



Linux 网络配置

网络基础配置

示例：

10. 为网络设备改名

```
#ip link set enp0s3 down  
#ip link set enp0s3 name lan  
#ip link set lan up
```

11. 恢复网络设备

```
#ip link set lan down  
#ip link set lan name enp0s3 up
```



Linux 网络配置

网络基础配置

示例：

12. 查看与本机联系的远程计算机网卡

```
#ip neigh
```



Linux 网络配置

网络基础配置

address 选项的参数

show: 单纯的显示出设备的 IP 信息

add|del: 进行相关参数的增加 (add) 或删除 (del) 设定, 主要有:

IP 参数: 主要就是网域的设定。

dev: 这个 IP 参数所要设定的设备,



Linux 网络配置

网络基础配置

dev 参数：

broadcast: 设定广播位址，如果设定值是 + 表示让系统自动计算；

label: 该设备的别名

scope：这个设备的领域，通常是以下几个类别



Linux 网络配置

网络基础配置

scope 参数：

global：允许来自所有来源的连线；

site：仅支持 IPv6，仅允许本主机的连接；

link：仅允许本设备自我连接；

host：仅允许本主机内部的连接；



Linux 网络配置

网络基础配置

示例：

1. 配置 IP

```
#ip addr add 192.168.100.123/24 brd +  
dev enp0s3
```

2. 配置一个子网卡

```
#ip addr add 192.168.1.123/24 brd + dev  
enp0s3 label enp0s3:0
```

Linux 网络配置

网络基础配置

示例：

3. 配置一个子网卡

```
#ip addr 192.168.1.123/24 brd + dev  
enp0s3 label enp0s3:niliu
```

4. 删除一个子网卡地址

```
#ip addr del 192.168.1.123/24 brd + dev  
enp0s3 label enp0s3:niliu
```



Linux 网络配置

网络基础配置

示例：

5. 删除一个 IP 地址

```
#ip addr del 192.168.100.123/24 dev  
enp0s3
```



Linux 网络配置

网络基础配置

route 选项

show: 显示路由表

add|del 增加或删除路由，其参数如下：

IP 或网域

via: 从哪个 gateway 路由数据

dev: 从哪个设备出去

mtu: 设定 MTU 值



Linux 网络配置

网络基础配置

route 选项

show: 显示路由表，其关键参数

proto: 此路由的路由协定，主要有 redirect, kernel, boot, static, ra 等，其中 kernel 指的是直接由 kernel 判断自动设定。

scope: 路由范围，主要是 link, 是与本设备有关的直链接



Linux 网络配置

网络基础配置

route 选项

add|del 增加或删除路由，其参数如下：

IP 或网域

via: 从哪个 gateway 路由数据

dev: 从哪个设备出去

mtu: 设定 MTU 值



Linux 网络配置

网络基础配置

1. 添加指定目标网络段

```
#ip route add 192.168.101.0/24 dev  
enp0s3
```

2. 删除指定的目标网络段

```
#ip route del 192.168.101.0/24
```



Linux 网络配置

网络基础配置

3. 对指定目标网络段，指定网关及指定的本地设备

```
#ip route add 192.168.101.0/24 via  
192.168.100.1 dev enp0s3
```

4. 设置默认路由

```
#ip route add default via 192.168.100.1  
dev enp0s3
```

Linux 网络配置

网络基础配置

5. 设置默认路由

```
#ip route add default via 192.168.100.1
```

6. 删除指定目标网络段，指定网关及指定的本地设备

```
#ip route del 192.168.101.0/24
```

7. 删除默认路由

```
#ip route del default
```



Linux 网络配置

网络基础配置

命令 :nmcli

功能：网络配置

语法格式 :nmcli [选项] [对象] [命令 | 帮助



Linux 网络配置

网络基础配置

nmcli 是 NetworkManager 服务的管理工具

NetworkManager 由一个管理系统网络连接、并且将其状态通过 D-BUS（是一个提供简单的应用程序互相通讯的途径的自由软件项目，它是做为 freedesktoporg 项目的一部分来开发的。）进行报告的后台服务，以及一个允许用户管理网络连接的客户端程序。

Linux 网络配置

网络基础配置

NetworkManager 的优点：简化网络连接的工作，让桌面本身和其他应用程序能感知网络。

NetworkManager 的后台服务：对于 GNOME 和 Xfce 那就是 nm-applet 工具，而对于 KDE 用户来说就是 KNetworkManager。

Linux 网络配置

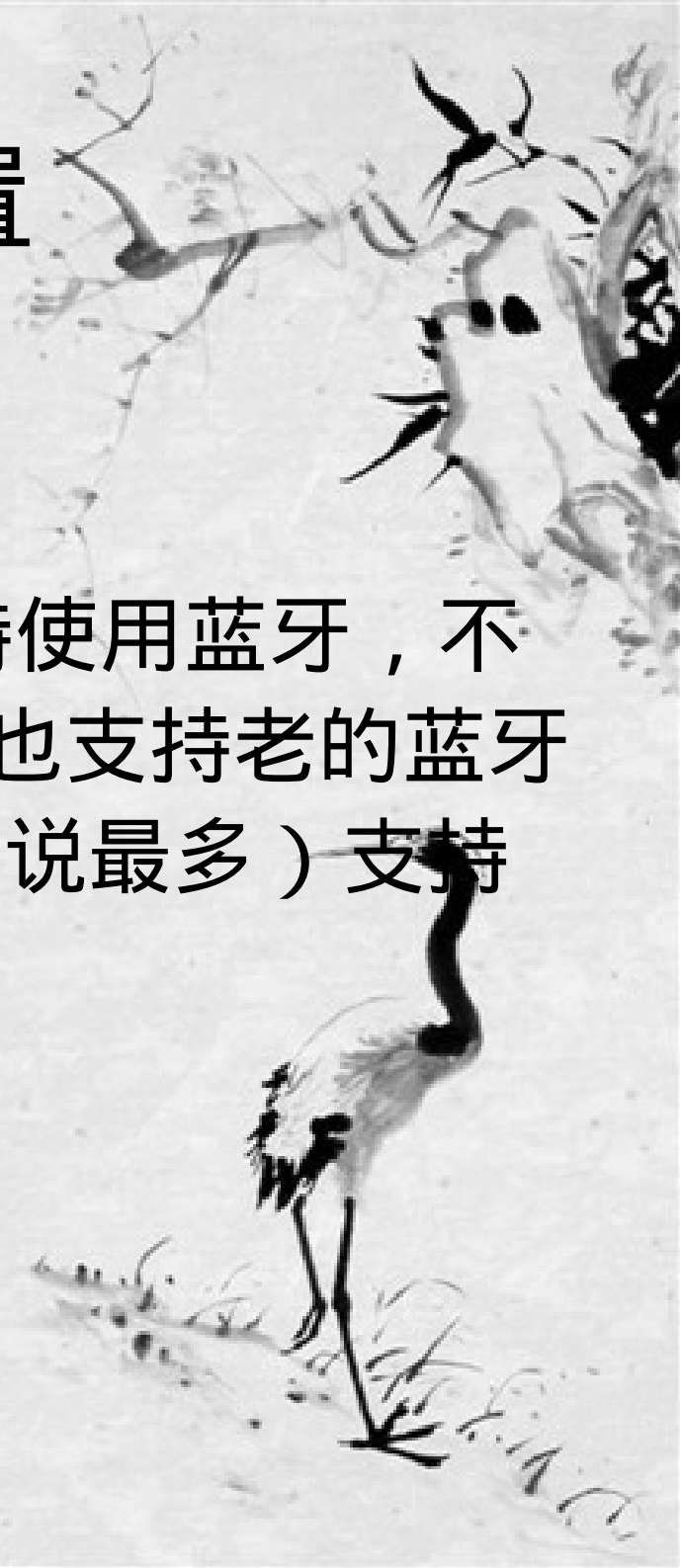
网络基础配置

NetworkManager 新版本支持更多的网络设备有关。从 0.8.1 开始增加了用于 GSM、UMTS 和 CDMA 卡等移动宽带设备 ModemManager 的支持。使用 ModemManager 也有新的功能，比如信号强度显示和选择 2G/3G 模式的功能。还可以关掉漫游功能以避免产生额外的连接费用；

Linux 网络配置

网络基础配置

NetworkManager 0.8.1 后还支持使用蓝牙，不仅支持个人区域网络 (PAN) 上网，也支持老的蓝牙拨号网络 (DUN) 协议。所有（或者说最多）支持的设备可以在 wiki 上找到；



Linux 网络配置



网络基础配置

NetworkManager 支持 IPv6 的自动配置和静态 IP ; NetworkManager 新版本还会带来 IPv6 的 DHCP 支持

NetworkManager 新版本增强设备以及协议的支持 , NetworkManager 最新版有一个全新的漂亮的客户端界面。

Linux 网络配置

网络基础配置

确认 NetworkManager 存在且运行

```
#systemctl status NetworkManager
```

网卡配置文件

```
#ls -l /etc/sysconfig/network-scripts/ifcfg-
```

*

Linux 网络配置

网络基础配置

nmcli 命令说明

命令	说明
nmcli dev status	列示所有设备
nmcli con show	列示所有连接的设备
nmcli con show --active	列示所有活动的设备 (工作状态)
nmcli con up "<ID>"	启用设备 ,ID 为设备的逻辑标识名 (此名称自定义)
nmcli con down "<ID>"	关闭设备 , 但如果服务重启将会再次激活此设备
nmcli dev dis <DEV>	关闭设备 , 如需激活设备需手工激活 (DEV 为物理设备名)
nmcli con add ...	添加一个新的链接
nmcli con mod ...	修改一个现有的链接
nmcli con del ..	删除一个现有的链接
nmcli net off	关闭所有的被管理的网络设备

Linux 网络配置

网络基础配置

nmcli 命令的结果：

每执行一次 nmcli 都直接对网络设备配置文件进行直接更改。

网络配置文件在 RHEL7 中位于：
/etc/sysconfig/network-scripts

配置文件均为 :ifcfg-xxx



Linux 网络配置

网络基础配置

nmcli 命令输入方法：

nmcli 支持缩写方式，如：

connection = con

disconnection = dis

delete = del

modify = mod

等。

同时，nmcli 还可以支持 <Tab> 键不起参数或选项



Linux 网络配置

网络基础配置

示例：

1. 列出活动的网络接口设备

```
#nmcli dev show
```

2. 查看当前区域内所有无线网络信息

```
#nmcli dev wifi
```

3. 显示所有的网络接口的信息

```
#nmcli con show
```



Linux 网络配置

网络基础配置

示例：

4. 显示所有活动（激活）的网络接口的信息

```
#nmcli con show --active
```

5. 查看当前的链接状态

```
#nmcli dev status
```

6. 指定网络接口设备的详细信息

```
#nmcli -p con show enp0s3
```



Linux 网络配置

网络基础配置

示例：

7. 查看设备的状态

```
#nmcli dev status
```

8. 查看指定的设备信息

```
#nmcli dev show enp0s3
```



Linux 网络配置

网络基础配置

示例：

9. 修改指定的设备名称

```
#nmcli con add con-name "default" type  
ethernet ifname enp0s3  
#nmcli con show default
```

10. 确认修改完设备名称后，出现设备名称的网络接口设备的配置文件

```
#ls -l /etc/sysconfig/network-scripts/ifcfg-  
default
```

Linux 网络配置

网络基础配置

示例：

11. 为指定网络设备添加 IP 地址 / 子网掩码

```
#nmcli con add con-name "default"  
ifname enp0s3 autoconnect no type  
ethernet ip4 192.168.100.1/24 gw4  
192.168.100.254
```

说明：

autoconnect: 不随 Linux 启动而激活接口

ip4:ipv4 地址 gw4:ipv4 的网关地址

Linux 网络配置

网络基础配置

示例：

11. 为指定网络设备添加 IP 地址 / 子网掩码

```
#nmcli con add ifname enp0s3  
autoconnect no type ethernet ip4  
192.168.100.1/24 gw4 192.168.100.254
```

12. 确认配置文件 (enp0s3 或 default)

Linux 网络配置

网络基础配置

示例：

13. 启动已经配置完成的网络设备

```
#nmcli con up "default"
```

或

```
#nmcli con up enp0s3
```

14. 关闭激活的设备

```
#nmcli con down "default"
```



Linux 网络配置

网络基础配置

示例：

15. 断开网络设备与本地系统的链接

```
#nmcli dev disconnect enp0s3
```

16. 将 default(enp0s3) 由自启动改为不自启动

```
#nmcli con mod "default"  
connection.autoconnect no
```

17. 增加 default(enp0s3) 的 DNS 地址

```
#nmcli con mod "default" ipv4.dns  
8.8.8.8
```


Linux 网络配置

网络基础配置

示例：

18. 增加 default(enp0s3) 的第 2 个 DNS 地址

```
#nmcli con mod "default" +ipv4.dns
```

8.8.8.8

```
#nmcli con show "default"
```

19. 修改 default 的 IP 地址及网关

```
#nmcli con mod "default"
```

```
ipv4.addresses "192.168.10.1/24  
192.168.10.254"
```



Linux 网络配置

网络基础配置

示例：

20. 增加 default 的 IP 地址

```
#nmcli con mod "default"  
+ipv4.addresses "192.168.10.1/24"
```

21. 删除现有一个链接文件

```
#nmcli con delete default  
#nmcli con show
```



Linux 网络配置

网络基础配置

示例：

22. 同时配置 2 个 DNS 地址

```
#nmcli con mod "default" ipv4.dns  
"202.106.148.1 202.106.0.20"
```



Linux 网络配置

网络基础配置

示例：

23. 添加 Wi-Fi 链接

1) 列示当前区域的 wifi

```
#nmcli dev wifi list
```

2) 对 wlan0 进行配置

```
#nmcli con add con-name myhome ifname wlan0  
type wifi ssid wifi2chuai ip4 192.168.1.1/24 gw4  
192.168.1.1
```

Linux 网络配置

网络基础配置

示例：

23. 添加 Wi-Fi 链接

3) 设定 wifi 的安全类型

```
#nmcli con modify myhome wifi-sec.key-  
mgmt wpa-psk
```

4) 设定 wifi 的密码 (密码为 wifioK)

```
#nmcli con modify myhome wifi-sec.psk  
wifioK
```

Linux 网络配置

网络基础配置

示例：

23. 添加 Wi-Fi 链接（方法二）

```
#nmcli dev wifi con 'wifi2chuai' password  
wifiok name 'myhome'
```

或

```
#nmcli dev wifi con 'wifi2chuai' password  
wifiok ifname wlan0
```



Linux 网络配置

网络基础配置

示例：

24. 指定 Wi-Fi 链接的 MTU

```
#nmcli con mod if 'myhome' 802-11-  
wireless.mtu 1350
```

25. 重新读取所有的网络设备配置文件

```
#nmcli con reload
```



Linux 网络配置

网络基础配置

示例：

26. 查看帮助

```
#man nm-settings
```

27. 查看 nmcli 的 add 帮助

```
#nmcli con add help
```

28. 查看模板

```
#man nmcli-examples
```



Linux 网络配置

网络基础配置

命令 :dhclient

功能：自动获取 IP 地址

前提要求：网络中存在 DHCP 服务器

语法格式 :dhclient [选项] [网络设备]



Linux 网络配置

网络基础配置

示例：

1. 使 enp0s3 动态获取 IP 地址

```
#dhclient enp0s3
```

2. 使 enp0s3 动态获取 IP 地址，并显示详细信息

```
#dhclient -v enp0s3
```



Linux 网络配置

网络基础配置

网络设备配置文件分析

```
#cd /etc/sysconfig/network-scripts/
```

```
#vim ifcfg-enp0s3
```



Linux 网络配置

网络基础配置

网络设备配置文件分析

//*

PEERDNS:yes|no

此为 peer2peerDNS 配置模式 ,DHCP 模式下必须匹配。如采用此配置,将直接修改 /etc/resolv.conf 文件里面的内容

PEERROUTES:yes|no

使用 peer2peer Router 模式

Linux 网络配置

ipv6 配置

```
#nmcli con add con-name enp0s8 type  
ethernet ifname enp0s8 \  
> ip6 2001:db8:0:1::c000:207/64 gw6 \  
> 2001:db8:0:1::1 ip4 192.0.2.7/24 \  
> gw4 192.0.2.1
```

Linux 网络配置

对比 nm 命令与 ifcfg 文件

nmcli con mod	ifcfg-*	说明
ipv6.method manual	IPV6_AUTOCONF=no	手工配置 ipv6
ipv6.method auto	IPV6_AUTOCONF=yes	使用 SLAAC(无状态配置) 通过发现协议自行配置
ipv6.method dhcp	IPV6_AUTOCONF=yes DHCPV6C=yes	使用 DHCPv6 分配地址, 但不使用 SLAAC
ipv6.addresses "2001:db8::a/64 2001:db8::1"	IPV6ADDR=2001:db8:a/ 64 IPv6_DEFAULTGW=200 1:db8::1	设置 IPV6 的地址及默认网关, 如果有多个 IPv6 地址, 则: IPV6ADDR_SECONDARIES="IPv6- IP-Address1 IPv6-IP-Address2"
ipv6.dns	DNS=0	设置 DNS
ipv6.dns-search niliu.edu	DOMAIN=niliu.edu	设置区域

Linux 网络配置

对比 nm 命令与 ifcfg 文件

nmcli con mod	ifcfg-*	说明
ipv4,ignore0auto0dn s true	IPV6_PEERDNS=no	忽略 DHCP 给予的 DNS 信息
connection.autoconn ect yes	ONBOOT=yes	随设备启动
connection.id enp0s3	NAME=enp0s3	定义设备名称
connection.interface- name enp0s3	DEVCE=enp0s3	指定网络设备
802-3-ethernet.mac- address	HWADDR=	定义网络设备的 MAC 地址

Linux 网络配置

ifconfig 配置 ipv6

1. 查看 ipv6

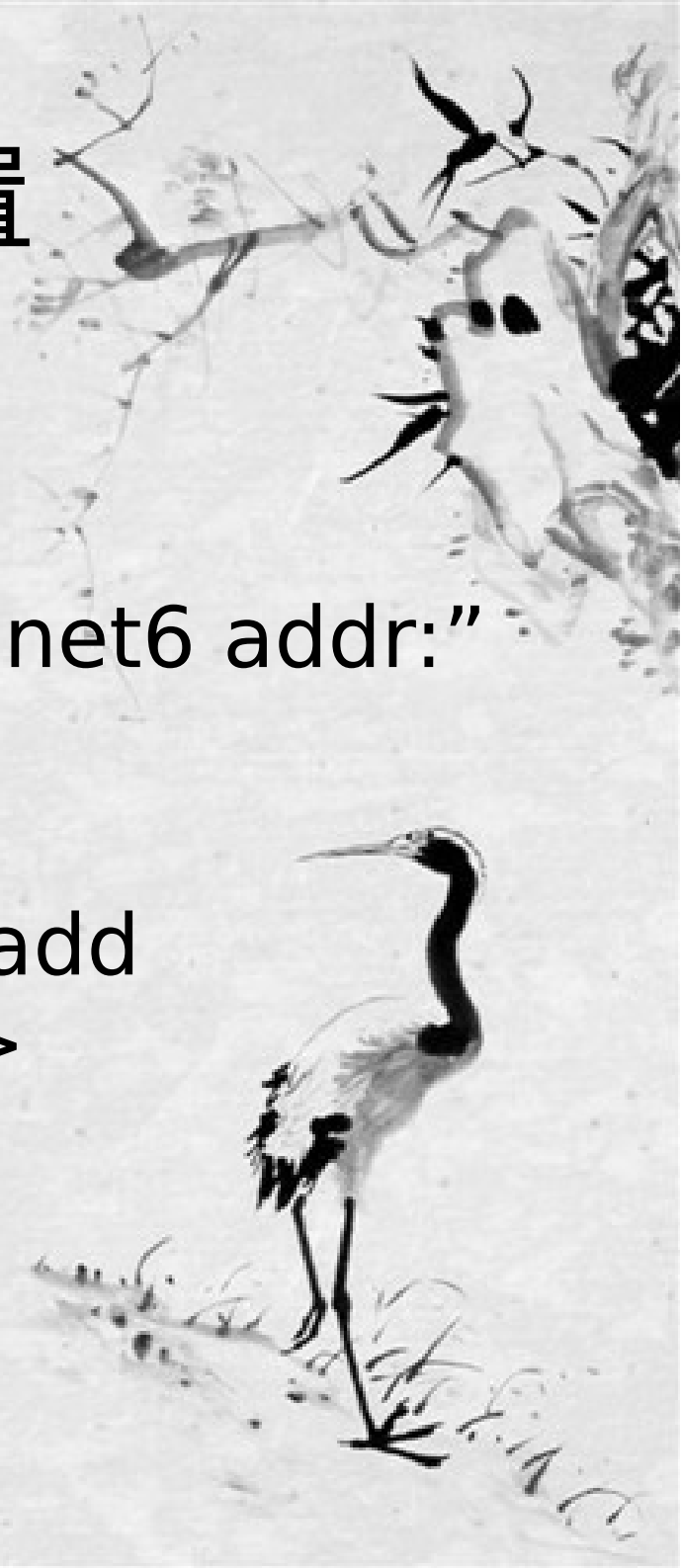
```
#ifconfig [interface] | grep "inet6 addr:"
```

2. 增加 ipv6

```
#ifconfig <interface> inet6 add  
<ipv6address>/<prefixlength>
```

如

```
#ifconfig enp0s3 inet6 add  
3ffe:ffff:0:f101::1/64
```



Linux 网络配置

ifconfig 配置 ipv6

3. 移除 IPv6

```
#ifconfig <interface> inet6 del  
<ipv6address>/<prefixlength>
```

如

```
#ifconfig enp0s3 inet6 del  
3ffe:ffff:0:f101::1/64
```



Linux 网络配置

ip 配置 ipv6

1. 查看 ipv6

```
#ip -6 addr show dev <interface>
```

如

```
#ip -6 addr show [dev] [enp0s3]
```



Linux 网络配置

ip 配置 ipv6

2. 添加 ipv6

```
#ip -6 addr add  
<ipv6address>/<prefixlength> dev  
<interface>
```

如

```
#ip -6 addr add 3ffe:ffff:0:f101::1/64 dev  
enp0s3
```



Linux 网络配置

ifconfig 配置 ipv6

3. 移除 IPv6

```
#ip -6 addr del  
<ipv6address>/<prefixlength> dev  
<interface>
```

如

```
#ip -6 addr del 3ffe:ffff:0:f101::1/64 dev  
enp0s3
```

Linux 网络配置

ipv6 配置

```
#nmcli con show static-eth0 | grep ipv6
```

设置 IPv6 地址 2001:db8:0:1::a00:1/64 和默认网关 2001:db8:0:1::1

```
#nmcli con mod static-eth0 ipv6.address  
"2001:db8:0:1::a00:1/64 2001:db8:0:1::1"
```

增加一个 IPv6 地址

```
#nmcli con mod static-eth0 +ipv6.dns  
2001:4860:4860::8888
```

Linux 网络配置

ipv6 配置 (2)

```
#nmcli con show
#nmcli con add con-name eno1 type
ethernet ifname eno1
#nmcli con show
#ip add show eno1
#nmcli con } show eno1 | grep ipv6
#nmcli con mod eno1 ipv6.addresses
'fddb:fe2a:ab1e::c0a8:1/64
fddb:fe2a:ab1e::c0a8:fe'
#nmcli con mod eno1 ipv6.method
manual
```

Linux 网络配置

ipv6 配置 (2)

```
#nmcli con down eno1
```

```
#nmcli con up eno1
```

```
#ip addr show dev eno1
```



Linux 网络配置

Linux 主机名

命令 :hostname

功能：显示 / 设置本地 Linux 的主机名称

语法格式 :hostname [参数] [主机名]

示例

1. 显示本地 Linux 主机名
#hostname



Linux 网络配置

Linux 主机名

示例

1. 显示本地 Linux 主机名

```
#hostname
```

2. 设置 Linux 主机名

```
#hostname snow.niliu.edu
```



Linux 网络配置

Linux 主机名

文件 : /etc/hostname

功能：显示 / 设置本地 Linux 的主机名称

示例

1. 查看或修改本地 Linux 主机名

```
#vim /etc/hostname
```



Linux 网络配置

Linux 主机名

命令 :hostnamectl

功能 : 管理 hostname

语法格式 :hostname [选项] { 命令 }



Linux 网络配置

Linux 主机名

示例：

1. 设置 Linux 主机名，并确认修改

```
#hostnamectl set-hostname  
snow.niliu.edu
```

```
#cat /etc/hostname
```

2. 查看 Linux 主机状态

```
#hostnamectl status
```



Linux 网络配置

网络基础配置 -DNS

1) 将本地计算机配置为 DNS 客户端

```
#vim /etc/resolv.conf
```

```
nameserver 8.8.8.8
```

```
nameserver 202.106.148.1
```

/* 如果 ifcfg-* 中配置 PEERDNS=yes 则需要配置在 ifcfg-* 文件中，否则此配置文件将会被覆盖 / 清空。

Linux 网络配置

网络基础配置 -DNS

1) 将本地计算机配置为 DNS 客户端

resolv.conf 中

#Generated by NetworkManager

指

此信息由 NetworkManager 进行配置（通过 ifcfg-enpx0x 的信息读取并写入）

Linux 网络配置

网络基础配置 -DNS

1) 将本地计算机配置为 DNS 客户端

resolv.conf 的关键字

nameserver: dns 服务器 IP 地址

domain: 如果本地计算机属于本域的管理者
(主服务器), 将使用此字段

search: 如果本地计算机不属于本域的管理者
(主服务器), 将使用此字段

Linux 网络配置

网络基础配置 -DNS

1) 将本地计算机配置为 DNS 客户端

增加 domain 或 search 字段，本地计算机在访问同一个域内的主机时，将不用在书写完整的 FQDN 来访问，可直接使用远程主机的主机名即可。

domain 与 search 字段相互冲突，默认情况下，哪个字段在下面，哪个字段生效

Linux 网络配置

网络基础配置 -DNS

1) 将本地计算机配置为 DNS 客户端

示例：

domain niliu.edu

或

search niliu.edu

或

domain niliu.edu chuai.edu

或

search niliu.edu chuai.edu



Linux 网络配置

网络基础配置 -DNS

/etc/resolv.conf 中的文件将会被 dhcp 或 NetworkManager 更改。因此每次更改此文件都无法保存。如果想取消这个特性，可以使用如下命令修改覆写机制

```
#nmcli con mod enp0s3 ipv4.ignore-auto-dns yes
```

将在 enp0s3 的配置文件中看到
PEERDNS=no

Linux 网络配置

网络基础配置 -DNS

2) 本地 DNS 数据库

```
#vim /etc/hosts
```

格式：

```
ip_address FQDN Alias_name
```

优点：直接通过本地查询 FQDN 所对应的 IP

缺点：分散式管理，难以维护



Linux 网络配置

网络基础配置 -DNS

3) 查看 hosts 数据库中的 localhost 字段

```
#getent hosts localhost
```



Linux 网络配置

网络通畅测试命令

1. ping: 测试网络通畅情况及相关信息
2. traceroute: 跟踪路由数据包到远程目的的路由
3. tracepath: 查看由本地到远程主机的各个 MTU

Linux 网络配置

示例：

1. ping 本地到远程是否通畅，并收集相关信息

```
#ping 192.168.100.100
```

2. 跟踪路由数据包所经历的路由

```
#traceroute www.baidu.com
```

3. 跟踪路由数据包所通过的各个 MTU 值

```
#tracepath www.baidu.com
```



Linux 网络配置

网络通畅测试命令

1. ping6: 测试网络通畅情况及相关信息
2. tarceroute6: 跟踪路由数据包到远程目的的路由
3. tarcepath6: 查看由本地到远程主机的各个 MTU

Linux 网络配置

示例：

1. 使用本地 enp0s3 ping(ipv6) 远程 ff02::1 地址 (Link-Local 由于不带任何子网信息，在 ping 时候必须指定其由哪个接口出去)

```
#ping6 fe02::1%enp0s3
```

2. ping

```
#ping6 2001::1
```



Linux 网络配置

示例：

3. tracepath6

```
#tracepath6 ff02::1
```

5. traceroute6

```
#traceroute6 ff02::1
```

4. 使用本地 enp0s3 链接远程 ssh

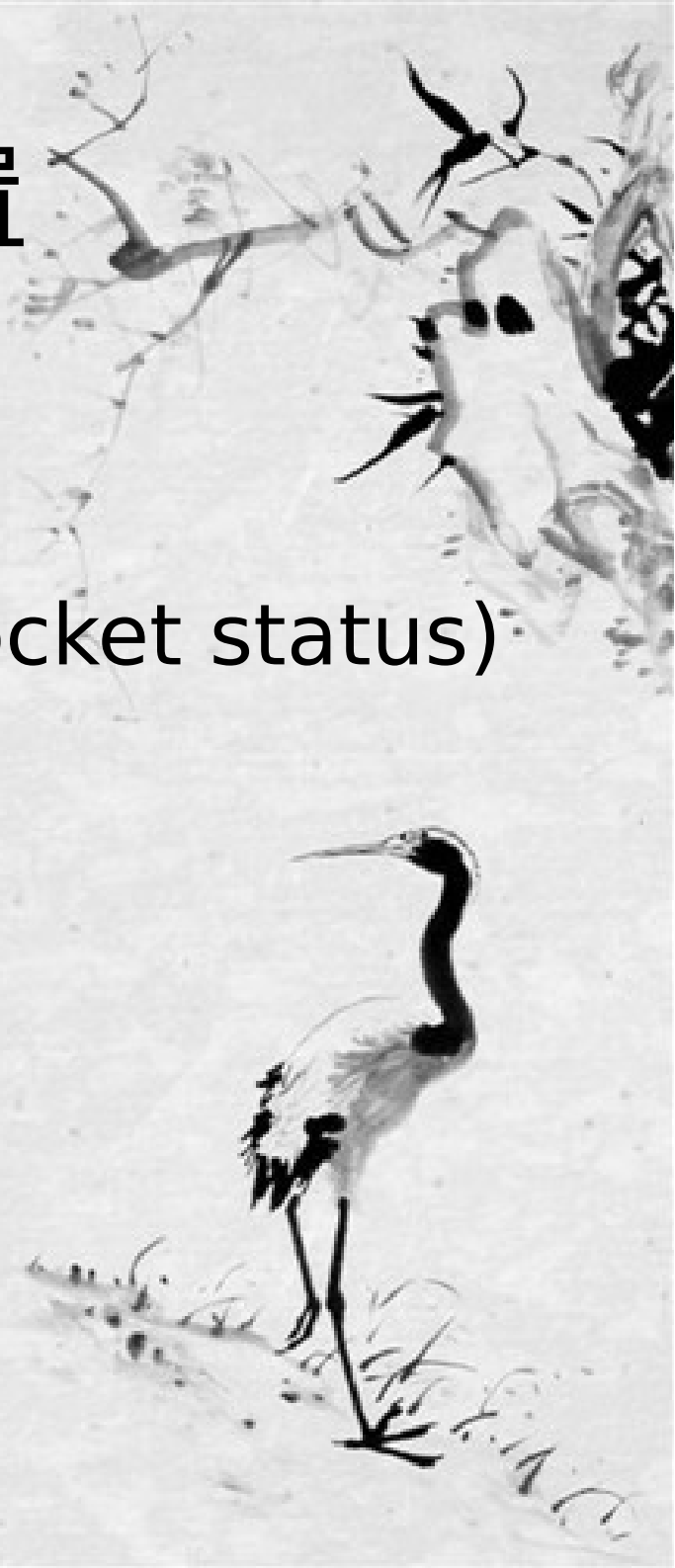
```
#ssh fe80::f482:dbff:fe25:6a9f%enp0s3
```

Linux 网络配置

命令 :netstat

功能 : 查看本地设备的服务状态 (socket status)

语法格式 :netstat [选项]



Linux 网络配置

参数：

- l: 显示仅监听的服务
- a: 显示所有的服务（含监听、已链接等）
- n: 以服务的端口号代表服务名称
- t: 显示 tcp 服务
- u: 显示 udp 服务



Linux 网络配置

参数：

-p: 显示进程所用的 socket

-A inet: 显示 ipv4 的信息

-A inet6: 显示 ipv6 信息

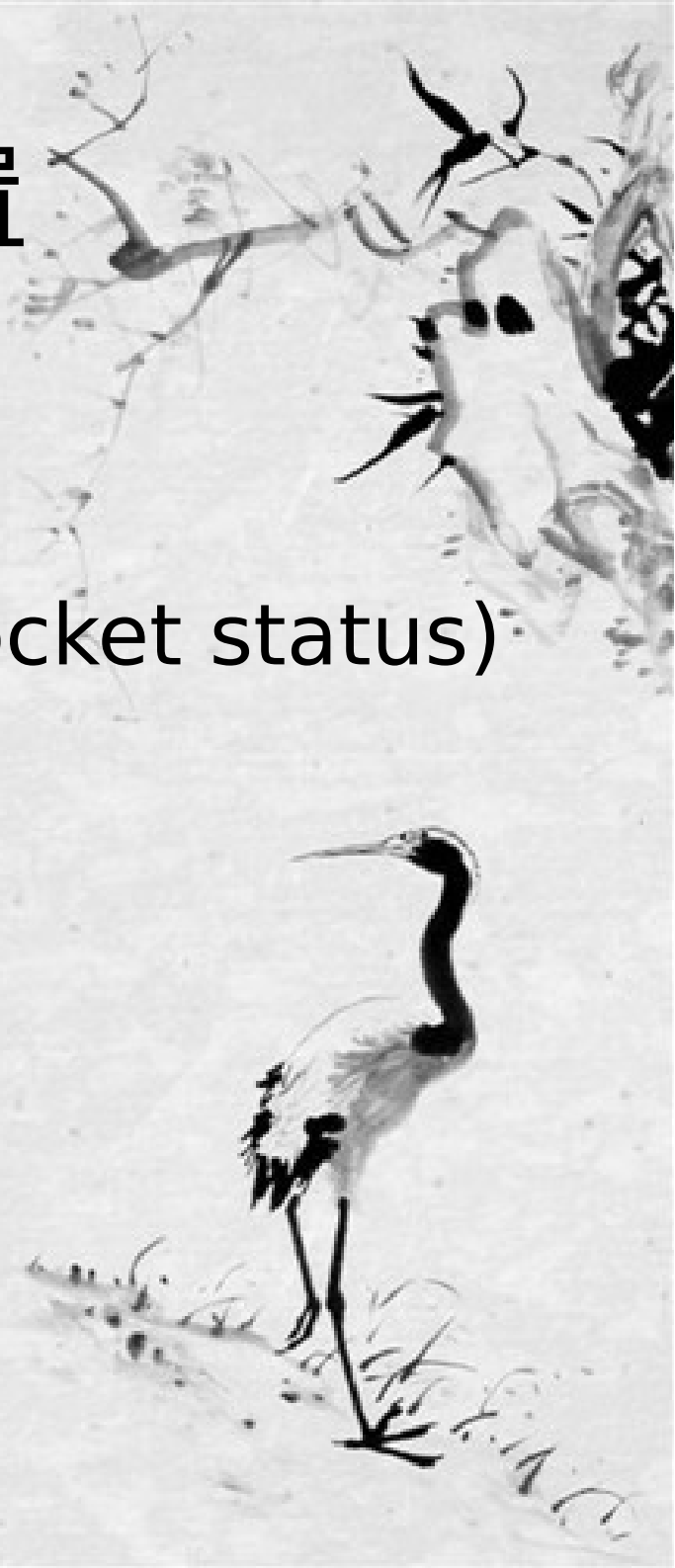


Linux 网络配置

命令 :ss

功能：查看本地设备的服务状态 (socket status)

语法格式 :ss [选项]



Linux 网络配置

参数 [与 netstat 相同]:

-l: 显示仅监听的服务

-a: 显示所有的服务 (含监听、已链接等)

-n: 以服务的端口号代表服务名称

-t: 显示 tcp 服务

-u: 显示 udp 服务



Linux 网络配置

命令 :lsof

功能 :lsof 命令可以列出被进程所打开的文件的信息

语法格式 :lsof [option]

举例 :
#lsof -i



Linux 网络配置

显示的各列信息

COMMAND : 进程的名称 PID : 进程标识符

USER : 进程所有者

FD : 文件描述符, 应用程序通过文件描述符识别该文件。如 cwd、txt 等 TYPE : 文件类型, 如 DIR、REG 等

DEVICE : 指定磁盘的名称

Linux 网络配置

显示的各列信息

SIZE : 文件的大小

NODE : 索引节点 (文件在磁盘上的标识)

NAME : 打开文件的确切名称



Linux 网络配置

示例：

1. 显示正在工作的 TCP 服务

```
#lsof -i tcp
```

2. 显示正在工作的 UDP 服务

```
#lsof -i udp
```

3. 显示所有正在工作的服务 (TCP 和 UDP)

```
#lsof -i
```



Linux 网络配置

示例：

4. 显示指定的 TCP 服务

```
#lsof -i tcp:80
```

5. 显示指定的 UDP 服务

```
#lsof -i udp:123
```

6. 显示哪个进程在使用指定端口 [TCP 及 UDP]

```
#lsof -i :123
```



Linux 网络配置

示例：

7. 列出某个用户的所有活跃的网络端口

```
#lsof -a -u snow -i
```

8. 列出所有的网络文件系统

```
#lsof -N
```

9. 显示所有打开的端口和 UNIX Domain 文件

```
#lsof -i -U
```



Linux 网络配置

示例：

10. 显示哪些进程打开了到 www.niliu.edu 的 UDP:123 端口的链接

```
#lsof -i udp@www.niliu.edu:123
```

11. 实时查看链接到 www.niliu.edu 的 ftp 情况 (默认为 15s 刷新一次)

```
#lsof -i tcp:www.niliu.edu:ftp -r
```

Linux 网络配置

示例：

12. 实时查看链接到 www.niliu.edu 的 ftp 情况
(-n 不将 IP 转换为 hostname)

```
#lsof -i tcp:www.niliu.edu:ftp -n
```



Linux 网络配置

还原 ethx

1. 在 grub 配置中向 kernel 传递参数
biosdevname=0 ← 关闭 biosdevname
2. 确认 /etc/udev/rules.d/ 下没有关于网络的规则文件
3. 进入 /etc/sysconfig/network-scripts/ 目录
4. 将 ifcfg-enp0s3 等文件改为 ifcfg-eth0 等名

Linux 网络配置

还原 ethx

5. 编辑 ifcfg-eth0

```
DEVICE="eth0"
```

```
BOOTPROTO="dhcp"
```

```
HWADDR="24:B6:FD:F8:D5:84"
```

```
NM_CONTROLLED="yes"
```

```
ONBOOT="yes"
```

6. 保存退出，重新启动相关服务 / 计算机

