**netstat**

**[功能]**

netstat是一个监控TCP/IP网络的非常有用的工具.

**[描述]**

netstat命令是一个监控TCP/IP网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态、以及网络协议相关的信息.

netstat打印的信息类型，取决于第一个参数。如下：

(none) 默认,表示没有,netstat会显示打开的sockets列表.如果你没有指定地址类别，那么所有配置的地址类别的活动sockets将都会被打印出来。

-r 显示内核的路由表。

-g 显示IPV4,IPV6的多播组成员信息。(什么意思？)

-i 显示所有的网络接口表。

-M 显示一些“伪”链接。（什么意思？）

-s 显示每个协议的总体统计信息。

一些选项：

-v 显示一些详细的信息。

-n 显示数字地址而不是尝试显示主机符号端口或者用户名称。

--numeric-hosts 显示数字主机地址但是不会影响到端口和用户名的显示。

--numeric-ports 显示数字端口号但是不会影响主机和用户名的显示。

--numeric-users 显示数字用户ID但是不会影响主机和端口名的显示。

--protocol=family 为显示的链接指定地址类型(低层次协议)，这里family是用','分隔开的地址类型关键字列表，例如inet,unix,ipx,ax25,netrom,以及ddp.这样指定和直接用--inet, --unix (-x), --ipx, --ax25, --netrom和--ddp选项是一样的。

-c 这样导致netstat在每个连续的秒都打印选定的信息。

-e 显示额外的信息。使用这个选项两次可以显示更多的细节。

-o 包含和网络计时相关的信息。

-p 显示每个套接字属于的程序名称和PID.

-l 只显示侦听的套接字。(默认被忽略)

-a 显示侦听和没有侦听的套接字。如果有--interfaces(即前面的-i)会显示没有启动的interfaces.

-F 从FIB中打印路由的信息。（默认如此）

-C 打印来自路由缓存的路由信息。

-t 指明显示tcp端口。

-u 指明显示udp端口。

关于输出的各个字段含义：

1)对于Active Internet connections (TCP, UDP, raw):

Proto socket使用的协议（tcp,udp,raw）.

Recv-Q 链接到这个sockets的应用程序没有拷贝的字节数目。

Send-Q 没有被远程主机确认的字节数目。

Local Address 本地socket末端的地址和端口号。如果不使用-n那么socket地址会被解释成为名字(FQDN),端口号会被解释成为相应的服务名称。

Foreign Address 远程socket末端的地址和端口号。类似Local Address.

State 端口状态。由于raw模式中没有状态还有udp中不使用state，这一列会被设置为blank.可能值：

ESTABLISHED:socket有一个建立起来的链接。

SYN\_SENT:socket是活动的，尝试建立一个链接。

SYN\_RECV:从网络中接收到了一个链接请求。

FIN\_WAIT1:套接字关闭，链接断开。

FIN\_WAIT2:链接关闭，等待远程端的套接字断开。

TIME\_WAIT:套接字在关闭之后仍旧等待，以处理还在网络上面的包。

CLOSE:套接字不用了。

CLOSE\_WAIT:远程端关闭了，等待套接字关闭。

LAST\_ACK:远程端关闭了，套接字也关闭了，等待确认。

LISTEN:socket侦听到来的链接，这一列在使用-l或者-a选项的时候才有。

CLOSING:两端的套接字都关闭了，但是我们还没有把我们的数据都发送出去。

UNKNOWN:未知的状态。

User socket拥有者的用户名或者用户ID.

PID/Program name socket拥有者的进程ID和进程名，用斜线分割。使用-p会显示这个列。你需要有超级用户权限来查看其他的你不拥有的socket.这个信息对于IPX的sockets不适用。

Timer (这个需要再写)

2)对于Active UNIX domain Sockets

Proto 套接字使用的协议(unix协议).

RefCnt 引用次数（也就是附加到这个套接字上面的进程）.

Flags flags的显示可以是SO\_ACCEPTON(显示为ACC), SO\_WAITDATA(W)或者SO\_NOSPACE(N).

SO\_ACCECPTON用在一个没有被链接的套接字上面（如果他们相应的进程等待一个链接请求）。

其他的flags一般不会用到。

Type 有如下几种套接字访问类型：

SOCK\_DGRAM:socket在数据报模式使用。（非连接的）

SOCK\_STREAM:这是一个流套接字（连接）.

SOCK\_RAW:用于Raw套接字（什么意思？）.

SOCK\_RDM:用于可靠的发送消息。

SOCK\_SEQPACKET:这是一个有序的包套接字。

SOCK\_PACKET:Raw接口访问套接字（什么意思？）.

UNKNOWN:未知.

State 包含如下的关键字：

FREE:套接字没有被分配。

LISTENING:套接字等待一个连接请求。指定-l或者-a选项的时候才会输出.

CONNECTING:套接字准备建立一个连接。

CONNECTED:套接字已经连接。

DISCONNECTING:套接字正在断开连接。

(empty):套接字没有和其他者连接。

UNKNOWN:不会出现的状态。

PID/Program name 打开套接字的进程的进程号和进程名。参见Active Internet connections.

Path 相应的进程附加到套接字的路径名。

Active IPX sockets 这里需要由其他知道这的人来做。

Active NET/ROM sockets 这里需要由其他知道这的人来做。

Active AX.25 sockets 这里需要由其他知道这的人来做。

3）其他相关的文件：

/etc/services 用于转换服务的文件。

/proc/net/ 这里有许多文件需要用它们来获取信息。

**[举例]**

\*查看活动的网络连接状况：

$netstat

默认，netstat会显示活动的链接列表.如果你没有指定地址类别，那么所有配置的地址类别的活动sockets将都会被打印出来。显示的内容很多，包括两个部分:本地的Active UNIX domain Sockets连接和网络上的Active Internet connections (TCP, UDP, raw)连接。本地的那个连接一般都不看。

\*查看所有的网络连接状况：

$netstat -a

通过-a选项，可以查看所有活动和不活动的连接的情况。

\*查看网络状况，地址和端口用数字表示：

$netstat -n

这里如果不用-n那么地址和端口就会用主机名称或者服务名称来表示了。使用-n不进行DNS轮询(可以加速操作),否则可能会很慢才显示出来。

\*察看系统当前监听的端口:

$netstat -lnp

这里，显示的结果既包含本地的，又包含网络的。选项的含义如下：

-l : 仅显示监听套接字(所谓套接字就是使应用程序能够读写与收发通讯协议(protocol)与资料的程序)

-n : 不进行DNS轮询(可以加速操作)

-p : 显示进程标识符和程序名称，每一个套接字/端口都属于一个程序。

\*只显示关于网络的所有连接：

$netstat --inet

或$netstat --ip

这样,就只显示与网络相关的连接，不再显示本地的连接了。

\*显示tcp和udp的侦听端口并且显示相应的程序id和程序名：

$netstat -tulpn

这里,

-t : 指明显示TCP端口

-u : 指明显示UDP端口

\*显示路由表

$netstat -rn

输出如下：

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

192.168.100.0 \* 255.255.255.0 U 0 0 0 usb0

192.168.0.0 \* 255.255.255.0 U 0 0 0 eth3

link-local \* 255.255.0.0 U 0 0 0 eth3

default 192.168.0.1 0.0.0.0 UG 0 0 0 eth3

这里，在调用netstat时,-r标记将显示内核路由表，就像我们利用route命令一样。

- n 选项令netstat以点分四段式的形式输出IP地址，而不是象征性的主机名和网络名。如果想避免通过网络查找地址（比如避开DNS或NIS服务器），这一点是特别有用的。

netstat输出结果中，第二列展示的是路由条目所指的网关，如果没有使用网关，就会出现一个星号（\*）或者0.0.0.0；第三列展示路由的概述，在为具体的I P地址找出最恰当的路由时，内核将查看路由表内的所有条目，在对找到的路由与目标路由比较之前，将对I P地址和genmask进行按位“与”计算；第四列显示了不同的标记，这些标记的说明如下：

G 路由将采用网关。

U 准备使用的接口处于“活动”状态。

H 通过该路由，只能抵达一台主机。

D 如果路由表的条目是由ICMP重定向消息生成的，就会设置这个标记。

M 如果路由表条目已被ICMP重定向消息修改，就会设置这个标记。

netstat输出结果的Iface显示该连接所用的物理网卡，如eth0表示用第一张，eth1表示用第二张。

另外，对于这个输出，我的机器情况是：网卡是eth3,使用局域网，网关192.168.0.1,本地ip是192.168.0.118,机器上面用usb连接开发板，usb0配置是用ifconfig usb0 192.168.100.1.开发板ip自动是192.168.100.200.

\*显示网络的当前配置特性：

$netstat -i

输入之后，输出如下：

Kernel Interface table

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg

eth3 1500 0 215348 0 0 0 102444 0 0 0 BMRU

lo 16436 0 176 0 0 0 176 0 0 0 LRU

usb0 1500 0 3184 0 0 0 37555 0 0 0 BMRU

在这里，我的机器使用eth3来上网，同时用usb连接了一个开发板子设置了一个usb网址用来连接usb的开发板子。如果调用时还带上-a选项，它还将输出内核中所有接口,并不只是当前配置的接口。

MTU和Met字段表示的是接口的MTU和度量值值；RX和TX这两列表示的是已经准确无误地收发了多少数据包（ RX - OK / TX - OK）、产生了多少错误（ RX-ERR/TX-ERR）、丢弃了多少包（RX-DRP/TX-DRP），由于误差而遗失了多少包（RX-OVR/TX-OVR）；最后一列展示的是为这个接口设置的标记，在利用ifconfig显示接口配置时，这些标记都采用一个字母。它们的说明如下：

B 已经设置了一个广播地址。

L 该接口是一个回送设备。

M 接收所有数据包（混乱模式）。

N 避免跟踪。

O 在该接口上，禁用A R P。

P 这是一个点到点链接。

R 接口正在运行。

U 接口处于“活动”状态。

\*显示内核的所有接口:

$netstat -ia

输入之后，输出如下：

Kernel Interface table

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg

eth3 1500 0 291968 0 0 0 139979 0 0 0 BMRU

lo 16436 0 176 0 0 0 176 0 0 0 LRU

pan0 1500 0 0 0 0 0 0 0 0 0 BM

usb0 1500 0 3184 0 0 0 37555 0 0 0 BMRU

\*查看网络协议的统计信息：

$netstat -s

这里信息较多，不列举了，统计网络协议的总体信息。包括IP,ICMP,TCP,UDP等。通过这个命令的输出，可以确定一个接收到的包中的错误信息在哪，这样用户可以将软件或者网络本身的错误隔离开。

**[其他]**

这个命令，在不同的系统上实现有所不同。

但大多系统的netstat都提供了如下的信息：网络连接、网络接口信息、数据缓存信息、路由信息、网络协议统计信息等，通过不同的选项可以显示相应的信息。这里的系统没有关于数据缓存的信息，如果有，那么有例如'-m'之类的选项。