< Linux

Hálózat

- **Szerző:** Sallai András

- Web: http://szit.hu

Bevezetés

A Debian 9 megjelenése óta az elnevezési rendszer megváltozott. Megszűnt az olyan hálózati kártya név alapértelmezett használata, mint a eth0, eth1, stb.

A net-tools csomag Debian 9-ben alapértelmezetten már nincs telepítve, így az ifconfig és más parancsok, csak ezen csomag telepítésével használhatók. A net-tools csomagot az iproute2 csomag váltja, amely alapértelmezetten telepítve van. A parancsok összevetését lásd lejjebb.

Hálózati kártya

A hálózati kártyákról az ip utasítás ad információt. A hálózati kártyák elnevezése a következők szerint történik.

1. A hálózati kártya neve magába foglalhatja az alaplapi Firmware/BIOS indexét. (pl.: eno1)
2. Magába foglalhatja a PCI Express hutplug index számát. (pl.: ens1)
3. Az elnevezésben szerepelhet a kártya fizikai elhelyezkedése/geometriája (pl.: enp2s0)
4. Tartalmazhatja a MAC címet. (pl.: enx35c7e13422d)
5. Lehet a klasszikus kiszámíthatatlan ethX (pl.: eth0)

```
ip addr show
```

A kimenet ehhez hasonló lehet:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 1000
    link/ether 00:50:8d:7c:ab:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::250:8dff:fe7c:ab04/64 scope link
       valid_lft forever preferred_lft forever
```

A kimenetben egy lo és egy enp1s0 nevű hálózati eszköz. Ebből persze csak az enp1s0 valódi kártya. A lo nevű eszköz, az úgynevezett visszacsatoló eszköz, amely minden operációs rendszerben megtalálható és helyi hálózati funkciók megvalósítását teszi lehetővé.

Egy konkrét hálózati kártyát is lekérdezhetünk:

```
ip addr show enp1s0
```

Hálózati kártya

A hálózati kártyán keresztül kapcsolódunk a hálózathoz. Ehhez be kell legyen töltve a hálózati kártya meghajtója, a beállításoknak fel kell legyen konfigurálva.

Ellenőrizzük, hogy a meghajtó be van-e töltve:

```
dmesg | grep eth
```

Ehhez hasonló eredményt kell lássunk:

```
[    0.963501] r8169 0000:01:00.0 eth0: RTL8168evl/8111evl at
0xffffa161c0c69000, bc:5f:f4:7e:38:5d, XID 0c900800 IRQ 24
[    0.963503] r8169 0000:01:00.0 eth0: jumbo features [frames: 9200 bytes,
tx checksumming: ko]
[    1.004700] r8169 0000:01:00.0 enp1s0: renamed from eth0
```

A hálózati kártyák a többi hardvereszközhöz hasonlóan egy állományra hivatkozva elérhetők a /dev könyvtár alatt. A nevük például „enp”-vel kezdődik.

MAC cím

A MAC cím a hálózati kártya fizikai címe. A gyártók minden kártyához rendelnek egy ilyet, de mindegyikhez mást. A gyártóknak ki vannak osztva bizonyos tartományok, így elvileg akkor sem ütközhetnek a fizikai címek, ha más gyártótól származik két kártya.

A fizikai cím hat darab hexadecimális számból áll. Általában kettősponttal (:) vagy kötőjellel (-) tagoljuk őket. Például:

```
00:50:8d:7c:ab:05
```

A fizikai cím kiderítéséhez futtassuk az alábbi parancsot.

```
ip addr show enp1s0
```

```
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 1000
    link/ether 00:50:8d:7c:ab:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global eth0
```

```
    inet6 fe80::250:8dff:fe7c:ab04/64 scope link
       valid_lft forever preferred_lft forever
```

A példában van ip címe a fizikai kártyának, de ez attól függetlenül is működik.

IP cím

A hálózati kártyákhoz egy IP címet rendelünk TCP/IP alapú hálózatokban, így találják megy a csomagok a célgéphez vezető utat.

Fix IP cím beállítása:

```
ip address add 192.168.1.210/24 dev enp1s0
```

Dinamikus IP cím kérése:

```
dhclient enp1s0
```

Az így beállított IP címek újraindítás után azonban elvesznek.

Beállított IP cím megtekintése:

```
ip address show dev enp1s0
```

vagy:

```
ip address show enp1s0
```

Statisztika:

```
ip link show
```

IP cím törlése:

```
ip address del 192.168.1.100 dev enp1s0
```

Mivel az ip addr add parancsa nem törli a meglévő hálózati kártya IP cím beállítását. A hálózati kártyának két IP címe lesz.

```
ip addr add 192.168.1.210/24 dev enp1s0
```

Ezért kell törléskor megadni, hogy melyik címet szeretnénk törölni.

Így is fel lehet venni ha broadcast lehetőséget is beállítjuk:

```
ip address del 192.168.99.37/24 brd + dev enp1s0
```

Virtuális hálózati kártyát is létrehozhatunk:

```
ip address add 192.168.99.37/24 brd + dev enp1s0 label enp1s0:0
```

Az összes hálózati kártya beállításának törlése:

```
ip address flush dev enp1s0
```

A hálózati kártya bekapcsolása:

```
ip link set enp1s0 up
```

A hálózati eszköz kikapcsolása:

```
ip link set enp1s0 down
```

A cím cseréje:

```
ip route change default via 192.168.0.1
```

Hálózati kártya beállításainak rögzítése

A hálózati kártyákat beállíthatjuk, hogy mindig az adott fix IP címet használják, vagy kérjenek mindig IP címet egy DHCP szervertől.

A hálózati kártya beállításait az alábbi állományban véglegesítjük:

/etc/network/interfaces

```
allow-hotplug enp1s0
iface enp1s0 inet static
        address 192.168.5.1
        netmask 255.255.255.0
        broadcast 192.168.5.255
```

Ezek után a felhúzzuk a hálózati kártyát:

```
ifup enp1s0
```

Az ifup parancs a beállításokat a /etc/network/interface állományból veszi.

A hálózati kártya leállítása:

```
ifdown enp1s0
```

Dinamikus ip cím esetén:

```
allow-hotplug enp1s0
iface enp1s0 inet dhcp
```

Egy hálózati kártyán több IP cím

/etc/network/interfaces

```
auto enp1s0
allow-hotplug enp1s0
iface enp1s0 inet static
    address 192.168.1.42
    netmask 255.255.255.0
    gateway 192.168.1.1
    up   ip addr add 192.168.1.43/24 dev enp1s0 label enp1s0:0
    down ip addr del 192.168.1.43/24 dev enp1s0 label enp1s0:0
    up   ip addr add 192.168.1.44/24 dev enp1s0 label enp1s0:1
    down ip addr del 192.168.1.44/24 dev enp1s0 label enp1s0:1
```

A lehetséges beállításokról az „interfaces” kézikönyv szolgál több információval.

```
man interfaces
```

Ellenőrizzük szintaktikailag helyes-e:

```
ifup --no-act enp0s3
ifup --no-act --interfaces=/home/janos/interfaces.new enp0s3
```

A hálózat tesztelése

Először győződjünk meg arról, hogy a gépünkön működik a hálózati kártyánk. Elsőként futtassuk a ifconfig parancsot az IP cím kiderítésére:

```
ip address show
```

Az IP cím például a következő lett:

```
192.168.6.2
```

Ekkor küldjünk visszhangválasz kérelmet a hálózati kártyánk IP címének:

```
ping 192.168.6.2
```

A ping parancs Echo Request típusú ICMP csomagokat küld a célcímnek, folyamatosan. A folyam megszakítása a Ctrl + C billentyűkombinációval lehetséges.

Ha ez válaszol, akkor a hálózati kártyánk működik.

```
ping 192.168.5.4
```

```
PING 192.168.5.4 (192.168.5.4) 56(84) bytes of data.
```

```
64 bytes from 192.168.5.4: icmp_req=1 ttl=64 time=0.033 ms
64 bytes from 192.168.5.4: icmp_req=2 ttl=64 time=0.031 ms
64 bytes from 192.168.5.4: icmp_req=3 ttl=64 time=0.031 ms
64 bytes from 192.168.5.4: icmp_req=4 ttl=64 time=0.030 ms
^C
--- 192.168.5.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.030/0.031/0.033/0.004 ms
```

Ha ez válaszol a fentiekhez hasonlóan, akkor a hálózati kártyánk működik.

Ezek után fokozatosan távolabbi gépektől kérjünk visszhangot. Erre a célra elsőnek az azonos hálózatban lévő, melletünk lévő gépeket célszerű használni.

Tegyük fel, hogy van mellettünk egy másik gép a következő beállításokkal:

```
ip address add 192.168.6.17/24 dev enp1s0
```

Küldjünk visszhangkérelmet a géptől:

```
ping 192.168.6.17
```

Ha válaszol a hálózat a másik gép felé rendben van.

Névszerver megadása

Resolver

A névszerverek a következő fájlban vannak beállítva:

```
/etc/resolv.conf
```

```
nameserver 192.168.2.1
```

Több információt a resolv.conf kézkönyv 5-dik fejezete ad:

```
man resolv.conf
```

Hosts fájl

A névszerverek előtt hosts fájlokban tárolták az IP cím hostnév, illetve domain név összerendeléseket. Egy adott számítógépen magunk is beállíthatjuk néhány gép IP címét, és hostnevét, domain nevét. A rendszer telepítéskor saját hostnevét és tartománynevét eleve beírja az /etc/hosts fájlba, azokat tehát itt javíthatjuk, vagy újabb bejegyzéseket vehetünk fel bennük. Jól jöhet ez a beállítási lehetőség, ha nem szeretnénk DNS szervert beállítani, a webszerverünk, vagy a levelezőszerverünk beállításainak teszteléséhez.

A hosts fájl szintaxisa a következő:

```
IP_cím Elsődleges_host_név [Álnevek ...]
```

Legyen például egy „bem” nevű hostunk. A tartomány legyen „kek.and”. A géphez tartozó IP cím 192.168.5.2 Ekkor a /etc/hosts fájlban lesz egy ilyen sor:

```
192.168.5.2      bem.kek.and      bem
```

A rendszer ennél többet ír bele:

/etc/hosts

```
127.0.0.1   localhost
192.168.5.2 bem.kek.and       bem

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Beleírja a localhost-hoz tartozó 127.0.0.1-es IP címet például, amire mindig szükségünk lesz, tehát ehhez ne nyúljunk! Újabb tartományokat viszont felvehetünk. Legye például egy lila.and tartománynév aminek az IP elérhetőségét (192.168.5.12) szeretnénk itt felvenni, ekkor így javíthatjuk a hosts fájlt:

/etc/hosts

```
127.0.0.1   localhost
192.168.5.2 bem.kek.and         bem
192.168.5.12    lila.and

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

A harmadik sorba felvettük az IPv6-os beállítások előtt. Igazából mindegy, hogy hányadik sorba fogjuk beírni, minden sorban érvényes lesz.

Másik lehetséges probléma, szeretnénk saját gépünkön egy virtuális tartományt használni. Fentebb már említettük, hogy ez jól jöhet például web vagy levelezőszerver beállításainak tesztelésénél. Legyen a virtuális tartományunk a „piros.and”. Ekkor a hosts fájl így nézhet ki:

/etc/hosts

```
127.0.0.1   localhost
192.168.5.2 bem.kek.and       bem
192.168.5.12    lila.and
192.168.5.2     piros.and

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Teszteléshez pingessük meg a piros.and tartományt:

```
ping piros.and
```

Ügyeljünk arra, hogy ez csak akkor működik, ha az IP cím pingetés önmagában is működik:

```
ping 192.168.5.2
```

Ez utóbbi IP cím saját gépünk címe, amely beállításáról korábban gondoskodnunk kellett.

A fentiek mellett az IPv6-os alapbeállítások is szerepelnek az állományban telepítés után. Az IPv6-os „localhost” nevet tudjuk is tesztelni:

```
ping6 ip6-localhost
```

Vagy IPv6-os cím alapján:

```
ping6 ::1
```

További információkért nézd meg a hosts kézikönyvét:

```
man hosts
```

Átjáró beállítása

Olyan gépek eléréséhez, amely nincs az aktuális hálózatban, átjárót kell állítani. Ilyen gépek alatt az Internet gépeit értjük általában. Ha az Internetet szeretnénk elérni egy belső hálózatból, akkor alapértelmezett átjárót kell beállítanunk. Ha az alapértelmezett átjáró címe a 192.168.2.1, akkor a következő parancsot használhatjuk.

```
ip route add default via 192.168.5.6
```

Ellenőrzés:

```
ip route show
```

A beállítás véglegesítése:

/etc/network/interfaces

```
allow-hotplug enp1s0
iface enp1s0 inet static
        address 192.168.5.6
        netmask 255.255.255.0
        network 192.168.5.0
        broadcast 192.168.5.255
        gateway 192.168.5.1
```

Felvettünk egy „gateway” nevű sort, ahol megadjuk az alapértelmezett átjárót.

A fentiek helyett használhatjuk az ip parancsot is. A routing tábla megtekintése:

```
ip route show
```

Hálózat hozzáadása:

```
ip route add 192.168.87.0/24 via 172.16.0.10
```

A 192.168.87.0 hálózatot a 172.16.0.10 átjárón keresztül lehet elérni.

Törlés, például egy hálózat törlése:

```
ip route del 192.168.87.0/24
```

Eltávolítás nélkül, csere:

```
ip route change default via 192.168.0.1
```

```
ip route replace default via 192.168.0.1
```

Portok felderítése

Az iproute2 csomag tartalmaz egy ss nevű parancsot. Segítségével felderíthetők a használt hálózati portok.

A legegyszerűbb használat esetén egyszerűen beírjuk a ss parancsot:

```
# ss
```

A parancs megjeleníti az összes TCP, UDP és unix foglalatokat (socket).

Ez megjeleníti az összes kapcsolatot. Általában túlfut a képernyőn. Ezért tördeljük például a less paranccsal:

```
# ss | less
```

A kimenet szűrhető például TCP kapcsolatokra a -t kapcsolóval:

```
# ss -t
```

A -A kapcsolóval megadható a három betűvel is a TCP:

```
# ss -A tcp
```

Vagy így:

```
# ss --tcp --listening
```

A kimenetben az aktuális kapcsolatokat láthatjuk, de nem látjuk a hallgatózó (kapcsolatra váró) portokat.

A -a kapcsolóval megjeleníthetjük a létező kapcsolatok mellett a kapcsolatra váró portokat is:

```
# ss -ta
```

Az UDP kapcsolatok listázása a -u kapcsolóval lehetséges:

```
# ss -u
```

Itt is használhatjuk a -a kapcsolót, amely a kapcsolatra várakozó portokat is megjeleníti:

```
# ss -ua
```

Ugyanaz, egyértelműbben:

```
# ss -A udp --all
```

Ha csak a unix foglalatokat szeretnénk látni, akkor használjuk az -x kapcsolót:

```
# ss -x
```

Hosszú kapcsolóval:

```
# ss --unix
```

Az ss parancs alapértelmezetten megpróbálja feloldani a címeket és portokat nevekre. Ha ez sikerül a címek és portok helyett, azok nevei jelennek meg. Ez letiltható a -n kapcsolóval:

```
# ss -tn
```

Hosszú kapcsolókkal:

```
# ss -A tcp --numeric
```

Ha csak a hallgatózó portokat szeretnénk megtekinteni, például TCP esetén:

```
# ss -lt
```

Ugyanígy UDP esetén:

```
# ss -lu
```

Összefoglaló megjelenítése:

```
# ss -s
```

Időinformációk megjelenítése:

```
# ss -t -o
```

IPv4 kapcsolatok:

```
# ss -lt4
```

Az „l” és a „t” elhagyható.

IPv6 kapcsolatok:

```
# ss -lt6
```

Az „l” és a „t” természetesen itt is elhagyható.

A kapcsolat állapota alapján is szűrhetünk:

```
# ss -t state established
```

Ilyen állapotok lehetnek például TCP esetén:

- established
- syn-sent
- syn-recv
- fin-wait-{1,2}
- time-wait
- closed
- close-wait
- last-ack
- stb.

A syn-sent, syn-recv olyan gyorsan hajtódik végre, hogy azt a valóságban nem vagyunk képesek elkapni. Ezért használhatjuk a watch parancsot:

```
# watch -n 1 "ss -t state syn-sent"
```

A teszteléshez egy böngészőben nyissunk meg egy weboldalt.

Szűrhetünk konkrét portra is:

```
# ss -nt dst :80
```

Esetleg több portra:

```
# ss -nt dst :80 or dst :443
```

Esetleg így:

```
# ss -nt dport = :80
```

Még több szűrési lehetőség:

```
# ss -nt src 192.168.10.11 sport gt :1024
```

Operátorok:

- <= vagy le – kisebb vagy egyenlő
- >= vagy ge – nagyobb vagy egyenlő
- == vagy eq – egyenlő
- != vagy ne –nem egyenlő
- < vagy gt – kisebb
- > vagy lt – nagyobb

Több információért lásd az ss kézikönyvét:

```
$ man ss
```

Esetleg help:

```
$ ss -h
$ ss --help
```

Esetleg telepítsük a iproute2-doc csomagot:

```
# apt install iproute2-doc
```

Ekkor a dokumentációk a következő helyen találhatók:

- /usr/share/doc/iproute2-doc/

Hosztnév

hostnamectl

A hosztnév tulajdonképpen a számítógép neve.

A gépnév lekérdezése:

```
hostnamectl
```

A gépnév beállítása:

```
hostnamectl set-hostname gepnev
```

Ez a beállítás tartós, az /etc/hostname fájl tartalmát is cseréli.

Régi Debian 8 és korábbi Linuxon

A host nevének lekérdezése:

```
hostname
```

A host neve itt van tárolva a merevlemezen:

```
cat /etc/hostname
```

A hostnév aktualizálása a /etc/hostname fájl alapján:

```
/etc/init.d/hostname.sh start
```

Új név beállítása a kernelben, menet közben:

```
hostname iskolazo
```

A gép nevét „iskolazo” névre állítjuk.

Új név beállítása tartománynévvel együtt:

```
hostname iskolazo.zold.and
```

Véglegesítéshez tegyük az /etc/hostname fájlba:

```
mcedit /etc/hostname
```

Írjuk az állományba az új hosztnevet.

Érvényesítés (Debian9-en már nincs):

```
/etc/init.d/hostname.sh start
```

A gépnevek lekérdezése másként:

```
sysctl kernel.hostname
```

```
uname -n
cat /proc/sys/kernel/hostname
```

Ha a gépnév mellett a tartománynevet is szeretnénk beállítani, az /etc/hosts fájlban tegyük meg.

FQDN lekérdezése:

```
hostname -f
hostname --fqd
```

ARP tábla

Lekérdezés:

```
# ip neighbor show dev eth0
```

Ürítés:

```
# ip neighbor flush dev eth0
```

vagy:

```
ip neigh flush all
```

Egy bejegyzés törlése:

```
ip neighbor del 192.168.10.2 dev eth0
```

Egy bejegyzés hozzáadása:

```
ip neighbour add 192.168.10.3 lladdr 00:80:c3:29:d3:55 dev eth0
```

Net-tools és Iproute2

Az net-tools a régi Linux kernelekhez készült hálózati eszközgyűjtemény. Az újabb kernelek újabb lehetőségeihez az iproute2 eszközgyűjtemény használandó. A net-tools csomagot már régen nem fejlesztik. Most összevetjük a két csomag utasításait.

Egy egyszerűbb megfeleltetési táblázat:

Cél	Net-tools	Iproute2
cím és kapcsolat (link) beállítás	ifconfig	ip add, ip link
irányítótábla	route	ip route
szomszédok (neighbors)	arp	ip neigh
vlan	vconfig	ip link
alagút (tunnels)	iptunnel	ip tunnel

Cél	Net-tools	Iproute2
multicast	ipmaddr	ip maddr
statisztika	netstat	ss
segítésg	ifconfig --help	ip help
interfészek státusza	ifconfig -s netstat -i	ip -s link

Magyar nyelvű környezetben a magyar nyelvű kézikönyvet kapunk. Azonban az ifconfig kézikönyv magyar oldala sem tartalmazza például a -a kapcsolót. Eredeti angol kézikönyv megtekintése:

```
man -L en ifconfig
```

Nézünk néhány példát.

Minden kapcsolt megjelenítése a hálózati interfészeken:

```
ifconfig -a
ip link show
```

Az interfész aktiválása

```
ifconfig eth0 up
ip link set up eth0
```

Az interfész deaktiválása:

```
ifconfig eth0 down
ip link set down eth0
```

IPv4-es cím interfészhez rendelése:

```
ifconfig eth0 192.168.8.1/24
ip addr add 192.168.8.1/24 dev eth0
```

Több IP cím:

```
ifcofnig eth0:0 192.168.8.2 up
ip addr add 192.168.8.2/24 broadcast 192.168.8.255 dev eth0
```

```
ip addr add 192.168.8.2/24 dev eth0
```

Valójában nem a legjobb példa, mert az ifconfig esetén technikailag egy új interfész jön létre. Az ip parancs esetén ugyanazon interfészen új IP cím.

IPv4-es cím törlése interfészen:

```
ifconfig eth0 0
ip addr del 192.168.8.2/24 dev eth0
```

IPv4-es cím megjelenítése interfészen:

```
ifconfig eth0
ip addr show dev eth0
```

A hálózati interfész MAC cím cseréje:

```
ifconfig eth0 hw ether 05:12:ab:2c:37:ac
ip link set dev eth0 address 05:12:ab:2c:37:ac
```

Irányítótábla megjelenítése:

```
route -n ; route -rn
ip route show
```

```
ip route
ip route show
ip route ls
```

Alapértelmezett átjáró hozzáadása:

```
route add default gw 192.168.10.1 eth0
ip route add default via 192.168.10.1 dev eth0
```

```
route add default gw 192.168.10.1
ip route add defalt via 192.168.10.1
```

Alapértelmezett átjáró törlése:

```
route del default
ip route del default
```

Az ip parancs itt kivételes, mert cserét is lehetővé tesz:

```
ip route replace default via 192.168.10.1
```

Statikusan hálózat hozzáadása:

```
route add -net 192.168.20.0/24 gw 10.0.0.1 dev eth0
ip route add 192.168.20.0/24 via 10.0.0.1 dev eth0
```

Statikusan megadott hálózat törlése:

```
route del -net 192.168.20.0/24
ip route del 192.168.20.0/24
```

Foglalat/Sokcet statisztika megtekintése:

```
netstat -l
ss -l
```

ARP tábla megtekintése:

```
arp  -an
ip neigh
```

Statikus ARP bejegyezés hozzáadása:

```
arp -s 192.168.5.100 00:3c:54:29:ac:23
ip neigh add 192.168.5.100 lladdr 00:3c:54:29:ac:23 dev eth0
```

Statikus ARP bejegyzés törlése:

```
arp -d 192.168.5.100
ip neigh del 192.168.5.100 dev eth0
```

Promiscuous mód beállítása:

```
ifconfig eth0 promisc
ip link set dev eth0 proisc on
```

Promiscuous mód tiltása:

```
ifconfig eth0 -promisc
ip link set dev eth0 promisc off
```

IPv6-os cím megadása

```
ifconfig eth0 inet6 add 2005:0db5:0:f102::1/64
ip -6 addr add 2005:0db5:0:f102::1/64 dev eth0
```

IPv6-os cím megtekintése:

```
ifconfig eth0
ip -6 addr show dev eth0
```

IPv6-os cím törlése:

```
ifconfig eth0 inet6 del 2005:0db5:0:f102::1/64
ip -6 addr del 2005:0db5:0:f102::1/64 dev eth0
```

Multicast címek megjelenítése:

```
netstat -g
ip maddr list dev eth0
```

Multicat cím hozzáadása interfészhez:

```
ipmaddr add 33:44:00:00:00:01 dev eth0
ip maddr add 33:44:00:00:00:1 dev eth0
```

Multicat cím törlése:

```
ipmaddr del 33:44:00:00:00:1 dev eth0
ip maddr del 33:44:00:00:00:1 dev eth0
```

Források:

- http://xmodulo.com/linux-tcpip-networking-net-tools-iproute2.html
- http://baturin.org/docs/iproute2/
- https://github.com/techniq/wiki/wiki/net-tools-vs-iproute2
- http://linoxide.com/linux-command/use-ip-command-linux/

Kalkulátorok

IP címek, alhálózatok számításához sok segítséget nyújt az alábbi két. Hálózatok tervezéséhez nagyon nagy segítség.

```
apt-get install ipcalc
```

```
apt-get install subnetcalc
```

Tesztek

Hálózati eszközök

Milyen hálózati eszközök vannak:

```
ls /sys/class/net/
```

Az eth0 hardvercíme:

```
cat /sys/class/net/eth0/address
```

Nyitott portok

A nyitott portokat vizsgálhatjuk meg az nmap paranccsal. Az nmap paranccsal csak saját szerverünket teszteljük. Használata:

```
nmap localhost
```

További információért lásd az nmap kézikönyvét:

```
man nmap
```

Az nmap parancs alapértelmezetten nincs telepítve. A telepítendő csomag neve is nmap:

```
apt install nmap
```

Melyik folyamat mit csinál

```
netstat -anp
```

Milyen portok vannak használva, ki használja:

- -a minden kapcsolat
- -p folyamatazonosítók megjelenítése (más rendszereken -o)
- -n nevek helyett számok (IP és portszám)
- -t tcp kapcsolatok

Link ellenőrzése

Az ethtool a hálózat meghajtó és hardver ellenőrzésére való.

Mi most csak a linket ellenőrizzük:

```
apt-get install ethtool
```

```
ethtool eth0 | grep "Link detected"
```

A link valósidejű figyelése:

```
ip monitor
```

Függelék

Hálózati kártya cseréje

A Debian GNU/Linux 9 előtti rendszerek a hálózati kártya fizikai címét megjegyzik. Ha hálózati kártyát cserélünk, előfordulhat, hogy a kártya nem fog működni. A rendszer ugyanis az eth0 eszköznévhez hozzárendelte a fizikai címet. Ha kicseréljük a hálózati kártyát akkor az az eth1-en lesz ez után elérhető, erre kell IP címet kérni vagy beállítani. Újabb csere után már csak a eth2-öt tudjuk használni, stb.

Győződjünk meg arról, hogy milyen néven érhető el a hálózati kártyánk:

```
dmesg | grep eth
```

A kimenet ehhez hasonló lehet:

```
[     1.551528] e1000 eth0: e1000_probe: Intel(R) PRO/1000 Network
Connection
```

```
[     4.639882] udev[221]: renamed network interface eth0 to eth1
```

Ebből nekünk fontos a „renamed network interface eth0 to eth1” rész. Ebből tudjuk hogy a hálózati kártyánk eth1 néven érhető el.

Ekkor kérhetünk erre IP címet.

A másik megoldás lehet, ha átírjuk fizikai cím, kártyanév megfeleltetést.

Szerkesszük a következő állományt:

```
/etc/udev/rules.d/70-persistent-net.rules
```

Ehhez hasonló sort kell találnunk:

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="08:00:27:98:43:27", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

Itt egyszerűen írjuk át a fizikai címet, mostani kártyánk fizikai címét megadva.

Szótár: persistent [UK: pə'sɪstənt] Örök, állandó, tartós

Hálókártya nevének megadása

A hálózati kártyák fizikai címének és nevének összekötése:

```
cat >> /etc/udev/rules.d/10_netinterfaces.rules <<VEGE
KERNEL=="eth*", SYSFS{address}=="00:11:22:33:44:55", NAME="eth0"
KERNEL=="eth*", SYSFS{address}=="11:11:22:33:44:55", NAME="eth1"
VEGE
```

Persze a hálókártya nevét itt is átírhatjuk:

```
/etc/udev/rules.d/70-persistent-net.rules
```

Változtatás előtt olvassuk el a fejrész tartalmát.

Socket fájl

A socket egy olyan speciális fájl, amit a folyamatok egymás közötti kommunikációra használnak.

Ha az *ls -l* kimenetét megnézzük egy soket fájl esetén, a legelső karakter, ami az állomány típusát jelzi, egy „s” betű.

Egyes programok, mint amilyen a mysql is, képesek szimpla TCP socket helyett unix socketben kínálni kapcsolódást.

Ilyen socketeket találunk például a következő könyvtárba:

```
/var/run
```

A unix socketek a kapcsolattartásra a következő C nyelvi utasításokat használják:

- sendmsg()
- recvmsg()

A socket tulajdonképpen egy IPC mechanizmust valósít meg. Az IPC az Inter Process Communication rövidítése. Proceszek közötti kommunikációnak lehetne fordítani.

/etc/nsswitch.conf

Name Service Switch, azaz névkiszolgálás kapcsoló, röviden NSS. A GNU C programkönyvtár 2.0 verziótól tartalmazza. Az adatok elérésnek sorrendje tárolt helyük alapján.

Elérhető adatbformák

- aliases
  - A levelezőszerverek álnevei.
- ethers
  - Ethernet címek.
- group
  - /etc/group - getgrent()
- hosts
  - /etc/hosts - gethostbyname()
- netgroup
  - hálózati gép és felhasználói lista
- networks
  - Hálózati nevek és címek - getnetent()
- passwd
  - /etc/passwd
  - getpwent()
- protocols
  - /etc/protocols
  - getprotoent()
- rpc
  - távoli eljáráshívás
  - getrpcbyname()
- services
  - hálózati szolgáltatások
  - getservent()
- shadow
  - Árnyékjelszavak
  - getspnam()

Beállítási lehetőségek

- files
  - hozzáférés a fájlokhoz közvetlenül
- db

    - adatbázis elérése
- nis
    - adatok NIS kiszoláglón
- nisplus
    - adatok NIS+ kiszolgálón
- dns
    - host és a hálózati DNS használata
- compat
    - A group, passwd, shadow hasznáalta
- also
    - többféle válasz

A Debian GNU/Linux 6.x alapbeállításai:

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          files mdns4_minimal [NOTFOUND=return] dns mdns4
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Több információt nyerhetsz a nsswitch.conf kézikönyv segítségével:

```
man nsswitch.conf
```

Az ifconfig

Az ifconfig parancs azért került át a „Függelék” részbe, mert elavult, vagyis használata kerülendő. Csak a teljesség igénye miatt írtam róla.

A hálózati kártyákról az ifconfig utasítás is ad információt. De használata elavultnak számít, bár igen sokan használják.

```
ifconfig
```

A parancs megjeleníti az összes nem vezeték nélküli hálózati eszközt:

```
eth0      Link encap:Ethernet  HWaddr 00:51:8d:7c:ab:04
          inet addr:192.168.15.4  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe7c:ab04/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:47251 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35554 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:53081975 (50.6 MiB)  TX bytes:5162538 (4.9 MiB)
          Interrupt:23 Base address:0xa000

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:102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19283 (18.8 KiB)  TX bytes:19283 (18.8 KiB)
```

A kimenetben az alábbi táblázatból láthatunk, esetlegesen kulcsszavakat:

UP	a hálózati kártya aktiválva van
BROADCAST	képes broadcast üzeneteket küldeni
RUNNING	A szükséges erőforrások le vannak foglalva
MULTICAST	küldhet és fogadhat multicast üzeneteket
MTU	Az maximálisan átvihető csomagméret
ALLMULTICAST	az összes multicast üzenet elfogadása
PROMISC	az eszköz minden forgalmat elfogad

Az ifconfig parancs segítségével megtudhatjuk a hálózati kártya IP címét is.

```
ifconfig eth0
```

A kimenet ehhez hasonló lehet:

```
eth0      Link encap:Ethernet  HWaddr 00:50:8d:7c:ab:05
          inet addr:192.168.5.4  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe7c:ab04/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2682 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2890 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1832097 (1.7 MiB)  TX bytes:528899 (516.5 KiB)
          Interrupt:23 Base address:0xa000
```

Fix IP cím beállítása:

```
ifconfig eth0 192.168.5.8 netmask 255.255.255.0
```

Több hálózati kártya

Egy hálózati kártyára több IP címet is felhúzhatunk a következő módon:

```
ifconfig eth0:0 192.168.6.1 netmask 255.255.255.0
```

Az eredményt az alábbiakban láthatjuk:

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:8d:7c:ab:04
          inet addr:192.168.5.4  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe7c:ab04/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:218523 errors:0 dropped:0 overruns:0 frame:0
          TX packets:137980 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:305289972 (291.1 MiB)  TX bytes:13455215 (12.8 MiB)
          Interrupt:23 Base address:0x2000

eth0:0    Link encap:Ethernet  HWaddr 00:50:8d:7c:ab:04
          inet addr:192.168.6.1  Bcast:192.168.6.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:23 Base address:0x2000
```

Az IP cím állandósítása:

```
iface eth0:0 inet static
    address 192.168.6.1
    netmask 255.255.255.0
```

Újabb példa:

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.42
    netmask 255.255.255.0
    gateway 192.168.1.1

auto eth0:0
allow-hotplug eth0:0
iface eth0:0 inet static
    address 192.168.1.43
    netmask 255.255.255.0

auto eth0:1
allow-hotplug eth0:1
iface eth0:1 inet static
    address 192.168.1.44
    netmask 255.255.255.0
```

Hálózati kártya nevek

- lo - Loopback, visszacsatoló eszköz
- eth0 - Ethernet kártya
- pan0 - Personal Area Network - Bluetooth

- vboxnet0 - VirtualBox virtuális hálózati kártyája
- wlan0 - Wifi eszköz
- wmaster0 - Wifi eszköz
- ra0 - wireless
- ppp0 - Modem, pont-pont-kapcsolat

A kimenet például ilyen lehet:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 1000
    link/ether 00:50:8d:7c:ab:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::250:8dff:fe7c:ab04/64 scope link
       valid_lft forever preferred_lft forever
3: pan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether f2:b4:8f:e4:8e:77 brd ff:ff:ff:ff:ff:ff
4: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
```

Ubuntu tartós beállítás

```
nano /etc/netplan/50-cloud-init.yaml
```

```
network:
    ethernets:
        enp0s3:
           addresses: []
           dhcp: true
    version: 2
```

```
network:
    ethernets:
        enp0s3:
            addresses: [ip/maszk]
            gateway4: ip
            nameservers:
                addresses: [8.8.8.8,8.8.4.4]
            dhcp4: no
    version: 2
```

Végül alkalmazzuk a beállításokat:

```
netplan apply
```

Lásd még

```
apt install ifmetric
man ifmetric
```

Irodalom

Linux net

- http://linux-ip.net/
- http://baturin.org/docs/iproute2/
- http://sugo.ubuntu.hu/10.04/html/serverguide/hu/network-configuration.html
- http://www.tty1.net/blog/2010-04-21-ifconfig-ip-comparison_en.html
- https://www.binarytides.com/linux-ss-command/ (2018)

Debian net

- http://www.debian.org/doc/manuals/debian-reference/ch05.en.html
- http://wiki.debian.org/NetworkConfiguration

IPv4

- http://unixlinux.tmit.bme.hu/IP-alapok

IPv6

- http://tldp.org/HOWTO/Linux+IPv6-HOWTO/index.html
- http://ipv6int.net/
- http://www.kame.net/~suz/gen-ula.html
- http://tools.ietf.org/html/rfc4193
- http://en.wikipedia.org/wiki/Unique_local_address
- http://hu.wikipedia.org/wiki/IPv6
- http://hu.wikipedia.org/wiki/IPv6-c%C3%ADm

Map

Logikai nevek mapelése:

- http://www.linuxtopia.org/online_books/linux_system_administration/debian_linux_guides/debian_linux_reference_guide/ch-gateway.en_018.html
- http://pastebin.com/m5EhvK0f
- http://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/
- http://www.science.uva.nl/research/air/wiki/LogicalInterfaceNames
- https://github.com/systemd/systemd/blob/master/src/udev/udev-builtin-net_id.c

- https://major.io/2015/08/21/understanding-systemds-predictable-network-device-names/
- https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/