

# Hálózati architektúrák és Protokollok

## GI – 7

Kocsis Gergely  
2017.05.08.

# Knoppix alapok

---

Virtuális gép létrehozása VirtualBox-ban (hálózatelérés: bridge módban)

- Rendszerindítás DVD-ről vagy ISO állományból
- Billentyűzetkiosztás beállítása
  - Rendszerindításkor (boot prompt)
  - Parancssorból (setxkbmap)
- Parancsablak (terminal window), root shell (su)
- Automatikus IP konfiguráció ellenőrzése (ifconfig)
- System log elérése (Alt+F12)



# Knoppix első lépések

---

Parancssori manuál:

**\$ man {parancsnév}**

Pl. \$ man ifconfig

Hálózati beállításokat kizárólag rendszergazdai jogosultsággal módosíthatunk!

Rendszergazdai jogosultsági szint (root shell) terminálablakban:

**\$ su -**

Billentyűzetkiosztás átváltása terminálablakban:

**\$ setxkbmap {hu|us}**

Pl. **\$ setxkbmap hu**



# Hálózati interfész paraméterei

---

## L1-es (fizikai) paraméterek

- Ethernet hálózati interfész kapcsolati sebességének és irányítottságának lekérdezése:

```
$ mii-tool eth0
```

- Ethernet interfész fizikai (link szintű) paramétereinek lekérdezése és módosítása:

```
$ ethtool eth0
```

## L2-es paraméterek

- Kérdezzük le az eth0 azonosítójú hálózati interfész MAC címét:

```
# ifconfig eth0
```

```
# ifconfig -a
```

Az IEEE OUI adatbázisából kérdezzük le a MAC címünk gyártóazonosítóját (OUI) – pl: <http://aruljohn.com/mac.pl>



# Ethernet kapcsolat ellenőrzése

---

## Hálózati kapcsolat ellenőrzése: L1/L2

- Vizuális ellenőrzés: a két Ethernet végpont összekapcsolása után az interfész link státusz-t jelző LED-je világít.
- Szoftveres ellenőrzés:
  - mii-tool  
**eth0: negotiated 1000baseT-FD flow-control, link ok**
  - ethtool  
**Speed: 1000Mb/s**  
**Duplex: Full**  
**Link detected: yes**
  - ifconfig  
**UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1**  
(UP: aktív, -: passzív státuszú interfész)  
(MTU - Maximum Transmission Unit)



# Hálózati interfész státusza

---

Interfész státuszának módosítása

```
$ ifconfig eth0 {up|down}
```

Pl. Helyezzük az interfészt passzív, majd ismét aktív státuszba:

```
$ ifconfig eth0 down
```

```
$ ifconfig eth0 up
```

Mindkét parancs kiadása után ellenőrizzük az interfész beállításait az `ifconfig eth0` paranccsal. Hogyan állapítható meg az `ifconfig` kimenet alapján az interfész státusza?



# ARP tábla

---

ARP tábla lekérdezése

```
$ arp [-a]
```

- Statikus és dinamikus bejegyzések felismerése!
- Meddig él egy dinamikus bejegyzés?
- Mi történik az ARP tábla bejegyzéseivel a hálózati interfész IP címének megváltoztatásakor?

Statikus bejegyzés létrehozása a táblában

```
$ arp -s IP_cím MAC_cím
```

```
pl. $ arp -s 10.0.0.1 00:01:02:03:04:05
```

Statikus bejegyzés törlése

```
$ arp -d IP_cím
```

```
pl. $ arp -d 10.0.0.1
```



# IP cím konfiguráció

---

## Feladat:

Konfiguráljuk fel a 10.0.1.x/24 IP címet számítógépünk eth0-s interfészen!

```
$ ifconfig eth0 10.0.1.x netmask 255.255.255.0 broadcast 10.0.1.255 up
```

Kompakt forma:

```
$ ifconfig eth0 10.0.1.x/24 broadcast 10.0.1.255 up
```

Töröljük az interfész IP címét:

```
$ ifconfig eth0 0.0.0.0
```

Dinamikus IP cím konfiguráció:

```
$ dhclient eth0
```



# IP kommunikáció ellenőrzése (ping)

A ping parancs az IP összeköttetés ellenőrzésére használható. ICMP (Internet Control Message Protocol) ECHO REQUEST csomagokat küld ki, melyre a megszólított csomópont ICMP ECHO REPLY üzenettel válaszol.

```
ping [opciók] {ip_cím|domain_név}
ping 193.6.128.5
ping -c 4 www.c3.hu
ping -s 1000 193.6.128.25
ping -w 3 10.0.1.1
```

## Opciók:

- -b: broadcast cím pingelése
- -c {szám}: adott számú ECHO REQUEST kérés küldése után leáll a program
- -f : elárasztásos (folyamatos) ping
- -s {méret}: a kiküldött ICMP ECHO REQUEST csomagok mérete bájtban
- -w {időtartam}: időtúllépés megadása másodpercben



# A route parancs

Statikus útvonal bejegyzése:

```
$ route add -net 10.0.1.0/24 dev eth0
```

```
[root@erlang ~]# route add -net 10.0.1.0/24 dev eth0
[root@erlang ~]# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.0.1.0         0.0.0.0         255.255.255.0  U         0      0      0 eth0
193.6.135.0     0.0.0.0         255.255.255.0  U         0      0      0 eth0
169.254.0.0     0.0.0.0         255.255.0.0    U        1002   0      0 eth0
0.0.0.0         193.6.135.1    0.0.0.0        UG         0      0      0 eth0
```

Statikus útvonal bejegyzése:

```
$ route add -net 10.0.2.0/24 gw 10.0.0.1 dev eth0
```

```
[root@erlang ~]# ifconfig eth0:0 10.0.0.2/24
[root@erlang ~]# route add -net 10.0.2.0/24 gw 10.0.0.1 dev eth0
[root@erlang ~]# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.0.0.0         0.0.0.0         255.255.255.0  U         0      0      0 eth0
10.0.1.0         0.0.0.0         255.255.255.0  U         0      0      0 eth0
10.0.2.0         10.0.0.1        255.255.255.0  UG         0      0      0 eth0
193.6.135.0     0.0.0.0         255.255.255.0  U         0      0      0 eth0
169.254.0.0     0.0.0.0         255.255.0.0    U        1002   0      0 eth0
0.0.0.0         193.6.135.1    0.0.0.0        UG         0      0      0 eth0
[root@erlang ~]# █
```



# A route parancs

Default útvonal bejegyzése:

```
$ route add default gw 10.0.0.1
```

## Feladat:

- A tanteremben alakítsunk ki 4 db **/26**-os alhálózatot a **10.0.1.0/24** kiinduló hálózatból.

A hallgatói gépek oszloponként egy IP hálózatba tartoznak, mindenki konfiguráljon fel egy szabad címet az eth0 interfészén a megfelelő alhálózatból.

- Pingeljünk meg az oszlopunkban (saját hálózatunkban) egy másik csomópontot.
- Pingeljünk meg egy másik oszlopbeli csomópontot. Kapunk választ? Magyarázzuk meg a jelenséget!
- Módosítsuk úgy a routing táblánkat, hogy a terem minden alhálózatát el tudjuk érni!

**Ip datagram továbbítás beállítása**

```
sysctl -w net.ipv4.ip_forward=1
```

