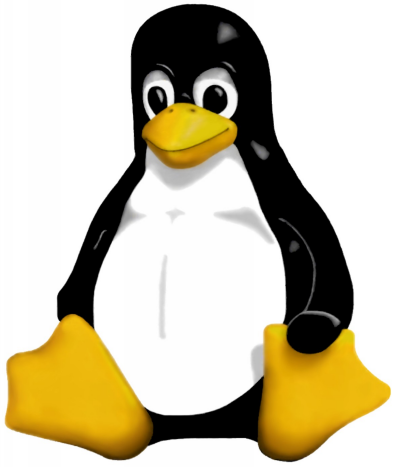


GNU/Linux

IV. kiadás

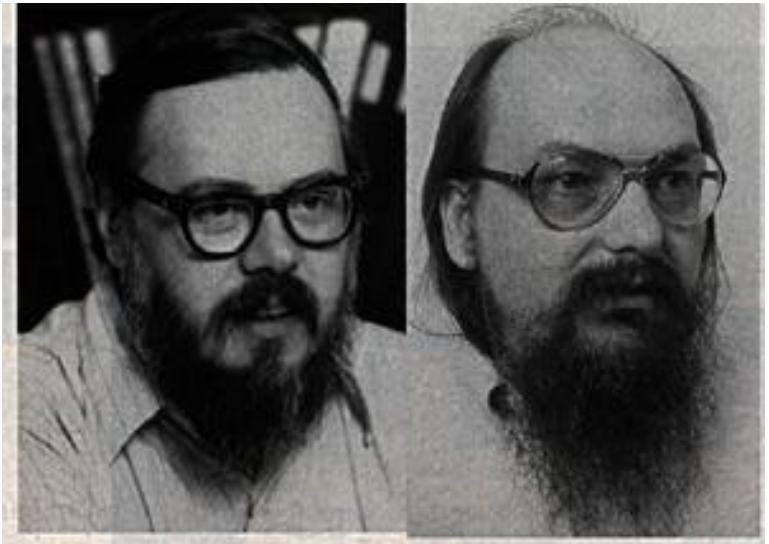
Készítette: Fuszenecker Róbert <hg8lhs@gmail.com> 2008-ban.

Történeti áttekintés

1969-től napjainkig

- 1964, MIT intézet: Multics projekt GE-645 gépen
  - többfelhasználós, időosztásos operációs rendszer
  - B2 biztonsági minősítés (1985), ACL, SMP, relációs adatbázisok kezelése
- 1969, Ken Tompson: a „Space travel” nevű játék PDP–7-en
  - nem multiuser, nem multitaszk, de megy
  - UNICS (később UNIX)
- 1971 (?): PDP–11
  - portolni szerették volna a UNICS-ot, de az assembly kód nem hordozható, ezért...
  - megszületik a C nyelv: Dennis M. Ritchie, 1973

PDP-7



Dennis Ritchie and Kenneth Thompson: they set the style for software development – and for software developers

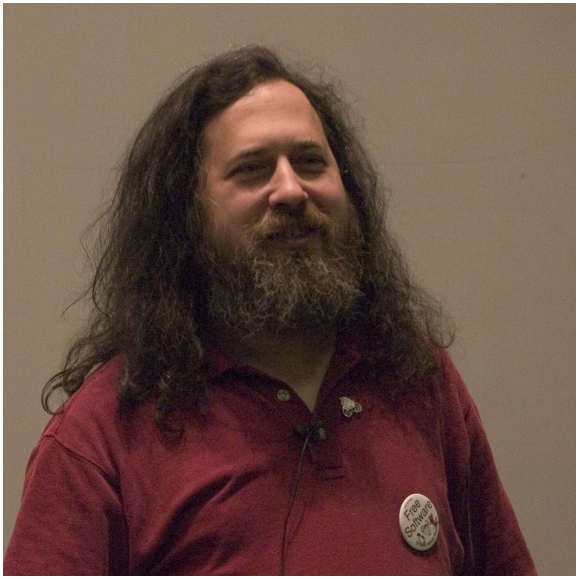
Dennis Ritchie és Ken Thompson

- 1974: a Berkeley egyetem oktatási célra másolatot kap
  - 1977: kiadja a saját disztibúcióját **BSD** néven, ami
    - Berkeley Softver Distribution (régebben)
    - Berkeley Software Design (manapság)
  - Ma is létező „kiadások”:
    - FreeBSD: modern technológiák, új fejlesztések
    - OpenBSD: nyíltság, szabványosság, biztonság
    - NetBSD: hordozhatóság (53 architektúra)
    - DesktopBSD, PC-BSD, DragonflyBSD: asztali felhasználásra
- A UNIX **de facto szabvány** lett, „ipari” implementációk
  - Apple: A/UX
  - IBM: AIX

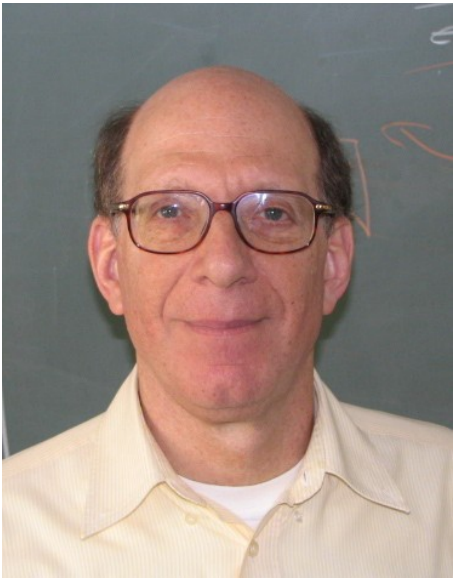
- Berkeley Egyetem: BSD/OS
- **Szovjetúnió: DEMOS (ДЕМОС)**
- Hewlett-Packard (HP): HP-UX
- Silicon Graphics, Inc. (SGI): IRIX
- Apple: MacOS X
- Waterloo Egyetem: QNX
- Sun Microsystems: Solaris, SunOS
- **Microsoft: XENIX**
- SCO Group: SCO UNIX
- Novell: UnixWare
- Siemens: SINIX
- Andrew S. Tanenbaum: MINIX
- Linus Torvalds és a többiek: Linux

- **1985: POSIX szabvány** – IEEE 1003 / ISO 9945

  POSIX = **P**ortable **O**perating **S**ystem **I**nterface
- 1985, **Richard M. Stallman**: „Szabad szoftver kiáltvány”: készüljön egy teljesen szabadon használható operációs rendszert, ami olyan, mint a UNIX, de nem UNIX ➜ a GNU nem UNIX („**G**NU is **n**ot **U**NIX”)
- 1987, **Andrew S. Tannenbaum**, Vrije Universiteit, Amsterdam: megszületik a MINIX (**min**imal UN**IX**)
- 1990-es évek
  - **Linus Torvalds**, egy finn egyetemista kísérletezik az i386-os processzorral és a Minix-szel
  - Eredményeit közzéteszi az interneten: akkoriban még a comp.os.minix hírcsoportban (newsgroup)
  - A GNU projekt és a Linux (Linus' UNIX) egymásra talál ➜ megszületik a (POSIX kompatibilis) **GNU/Linux**



Richard M. Stallman



Andrew S. Tannenbaum prof.

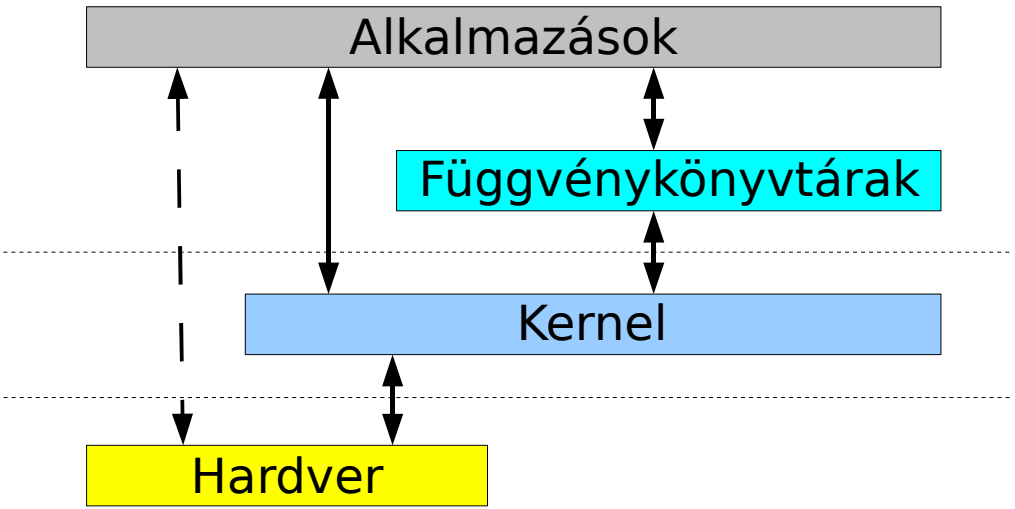
Linus Torvalds

- Az **internetnek** köszönhetően hihetetlenül gyorsan fejlődik és terjed. A linux-terjesztések (disztribúciók) száma közel 600 (nagy részüket már senki sem használja, pl. egyfloppys terjesztések)
- Ma a UNIX jogok az **Open Group**nál vannak
- a **POSIX szabvány** – IEEE 1003 / ISO 9945 – megvásárolható, letölthető

A UNIX alapfilozófiája

- Legyen **minden erőforrás fájl** (ID-vel, azonosítóval megcímezhető adatfolyam (stream))
  - eszközök (pl. hangkártya, nyomtató, merevlemez)
  - hálózati kapcsolat (socketek)
  - rendezett adathalmazok (képek, hangok, szöveges fájlok), stb.
- egységesen kezelhetők legyenek az erőforrások (open(2), read(2), write(2), close(2), ioctl(2), seek(2), mmap(2))
- az operációs rendszer és a programok hordozhatósága (POSIX szabványnak (is) köszönhetően)
- „villamosmérnökök írták saját maguknak”

Az operációs rendszer felépítése



- **a hardver**: a számítógép fizikailag megérinthető elemei
- **kernel**: az operációs rendszer magja; feladatai (a teljesség igénye nélkül):
  - elrejteni a hardver sajátosságait, kezelni az eszközöket (definiált interfészeken keresztül)
  - processzor- és memóriakezelés
  - folyamatok irányítása: processzek indítása, ütemezése, leállítása, kivételek kezelése
  - inter-process (folyamatok közötti) kommunikáció
  - közös erőforrások kezelése (kölcsönös kizárás)
  - „extra” szolgáltatások:
    - fájlrendszerek
    - hálózat
    - felhasználók, csoportok, jogosultságok, stb.

- **függvénykönyvtárak**: magasabb szintű, összetett műveletek, melyeket a felhasználói programok gyakran használnak, például:
  - magas szintű fájlkezelés:

    printf(3) vs. ***write(2)***, ***fscanf(3)*** vs. ***read(2)***
  - sztringek, asszociatív tömbök
  - matematikai műveletek, speciális algoritmusok (pl. FFT)
  - speciális fájlformátumok (képek, hangok, tömörítési eljárások, stb.)

  a függvénykönyvtárak előnye, hogy
  - lehetővé teszik a kód-újrahasznosítást: amit más megírt, azt nem kell megírnom a programomban

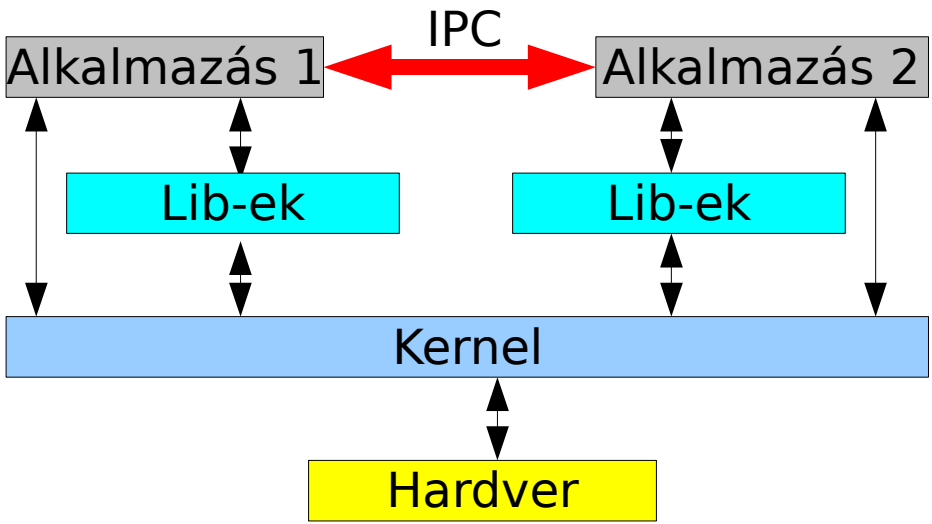
- memóriatakarékosság: ha egy függvénykönyvtár már be van töltve a memóriába, nem kell egy másik folyamat számára újra betölteni annak indításakor

- **alkalmazások**:

  definíció: folyamat (process): a lemezen tárolt program egy elindított változata (PID azonosítóval)

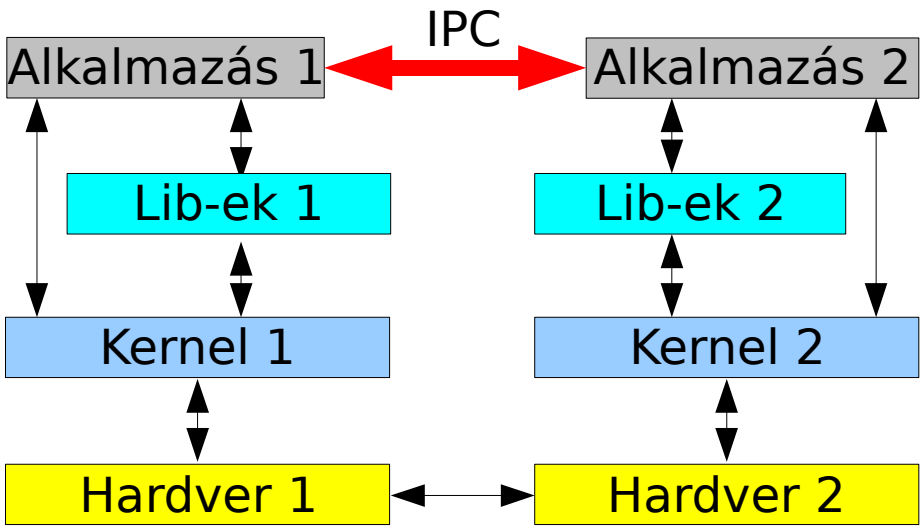
  - **előtérben futó folyamat**: általában használják a standard inputot, illetve outputot
  - **háttérben futó folyamat**: nem használják a standard inputot, illetve outputot
  - **démon**: olyan háttér folyamat, amely valamilyen erőforrás kezelésért (multiplexelés) felel
  - **zombi**: olyan folyamat, ami valami miatt „megdöglött”, időnként az operációs rendszer kitakarítja a memóriából

Inter-processz kommunikáció (IPC)



- **feladata**: a folyamatok közötti adatcsere megvalósítása; olyankor alkalmazzuk, ha egy erőforráshoz több folyamat szeretne hozzáférni ➜ multiplexelés; például:
  - nyomtatás, adatbázis-kezelés, hang, grafikus felhasználói felület
- **megoldások**:
  - **osztott memória, szemaforok/mutexek**
    előnye: gyors
    tipikusan a grafikus felület (X szerver) használja
  - **fifo-k** (pipe-ok és UNIX domain socketek)
    előnye: jól illeszkedik a UNIX filozófiához ➜ minden dolog fájlnak tekintendő; kicsit lassabb, mint az osztott memória
  - **hálózat** (jellemzően TCP/IP)
    előnye: a processzeknek nem kell ugyanazon a gépen futniuk ➜ **kliens-szerver modell** alakítható ki

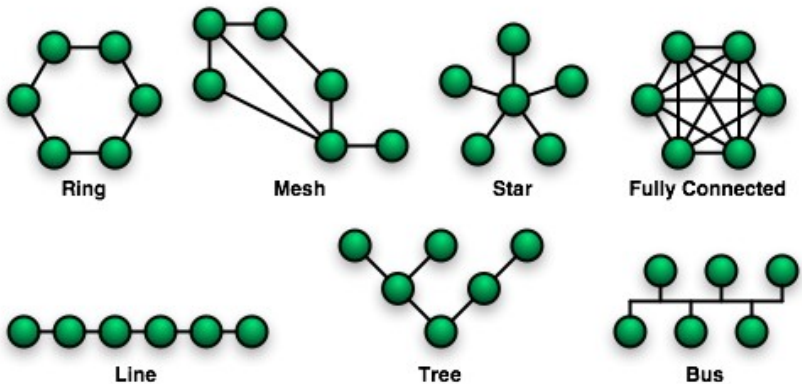
IPC megvalósítása hálózaton



Hálózatok

- Számítógépek összekapcsolása információcsere céljából
- Fogalmak (előző félévekből ismerősek):
  - hálózati topológiák
  - csomópont (node)
  - ismétlő (repeater)
  - hub
  - switch
  - híd (bridge)
  - átjáró (gateway)
  - útvonalválasztó (router)

- Hálózati topológiák (rövid összefoglalás, ismétlés, példák):
  - busz (pl. token-bus, 10Mbps ethernet)
  - gyűrű (pl. token-ring)
  - csillag (pl. 100/1000 Mbps Ethernet)
  - ad-hoc (pl. wifi)



Hálózati protokollok

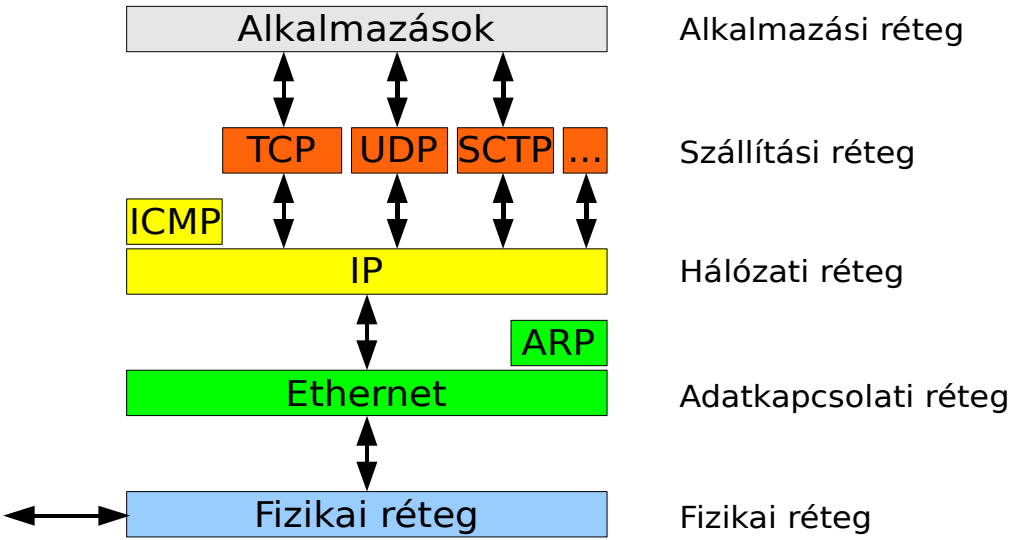
„Protokoll: az OSI modell **azonos szinten** elhelyezkedő rétegei közötti kommunikáció szabályrendszerét írja le.”

(Sándor Tamás)

Ebben a részben a következő protokollokkal ismerkedünk meg:

- Ethernet (IEEE 802.3)
- ARP (RFC 826)
- IP v4 és v6 (RFC 791 és 2460)
- ICMP (RFC 792)
- TCP (RFC 793)
- UDP (RFC 768)

Hálózati rétegek



Hálózati rétegek feladatai

- **Fizikai réteg**: megteremti a „bitszintű” kapcsolatot a hálózat elemei (csomópontok) között
- **Adatkapcsolati réteg** (jelen esetben Ethernet):
  - keretszintű kommunikációt tesz lehetővé, a node-okat fizikai címmel (MAC-címmel) azonosítja
  - az ARP (Address Resolution Protocol) segítségével az IP-címeket MAC-címeké konvertálja
- **Hálózati réteg**: a csomagok egyik csomóponttól a másik csomópontig juttatását irányítják
  - az IP csomag tartalmazza a forrás gép és a cél gép IP-címét, ami alapján az útvonalválasztók eljuttatják a csomagot a megfelelő hálótatba

- az ICMP (Internet Control Message Protocol) üzenetek a hálózat állapotáról adnak információt
  - ICMP echo request ➔ ping
  - csomópont nem található (host not found)
  - hálózat nem található (network not found), stb.

- **Szállítási réteg**:
  - **TCP (Transmission Control Protocol)**: az adatfolyam daraboktra tördelését végzi, majd a célgépen (sorrendhelyesen és hibamentesen) visszaállítja az adatfolyamot, így egyfajta „soros bájtátvitelt” tesz lehetővé (stream)

    megbízható, sorrendhelyes, nyugtázott átvitel

    a **TCP portcím** segítségével megkülönböztethetők,

megcímezhetők azon folyamatok (gyakran démonok, rendszerfolyamatok), melyek „regisztraálták” magukat, vagyis megnyitottak egy TCP portot

- **UDP (User Datagram Protocol)**: lehetővé teszi, hogy a folyamatok üzeneteket küldjenek egymásnak; az üzenetek épsége és sorrendje nem garantált, mert nincs nyugtázás

  az UDP is használ portcímeket, melyek függetlenek a TCP portoktól

  tipikusan real-time alkalmazásokban (Voice-over-IP)

- **SCTP (Stream Control Transport Protocol)**: multimédia tartalmak blokkos átvitelét (SCTP üzenetek) teszik lehetővé; egy SCTP üzenet több megabájt méretű is lehet

  előnye a TCP-hez képest, hogy torlódás esetén eldobja a régóta várakozó üzeneteket, így utat enged

az újabbaknak („legyen inkább kis hiba a hangban/képben, de ne késsen, és élvezhető maradjon”)

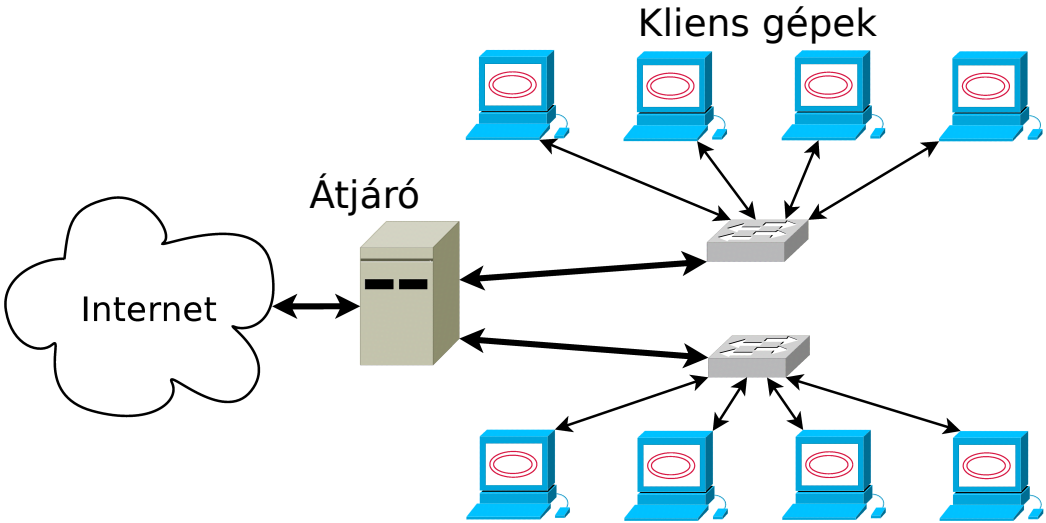
üzenetalapú, megbízható, sorrendhelyes, nyugtázott

- **Egyéb**: tipikusan routing protokollok, melyek a hálózat útvonalválasztóinak működését befolyásolják

- **Alkalmazási réteg**: az alkalmazások egymás között megvalósuló protokolljai, pl.:
  - FTP (File Transfer Protocol): fájlok átvitelére
  - TFTP (Trivial File Transfer Protocol): boot-olás során, fájlok átvitelére (egyszerűsített ftp)
  - POP (Post Office Protocol): levelek letöltése a szerverről
  - stb.

Hálózatok kialakítása

A következő ábra példát mutat egy helyi hálózat kialakítására:



A fenti hálózat a következő összetevőket tartalmazza:

- kliens számítógépek, két alhálózatba szervezve (a hálózati árjáró más-más interfészére „csatlakoznak”)
- switch-ek: ezek biztosítják a fizikai kapcsolatot a hálózati csomópontok között
- átjáró (gateway)
- külső, publikus hálózat, „Internet”

A hálózat elemeinek paraméterei

- **MAC-cím**: 48 bit hosszú, a hálózati interfész egyedi azonosítója, elvileg nem változtatható meg, a gyártó programozza a hardverbe

  Létezik broadcast (mindenkit megszólító) MAC-cím is: FF:FF:FF:FF:FF:FF

  A hálózati kártyák „egymás között” Ethernet protokollt (Layer 2) használnak, ezért az IP-címeket MAC címmé kell konvertálni. Erre szolgál az ARP protokoll.

- **IP-cím**: a hálózati interfész Layer-3 címe, ez alapján történik az IP keretek célba juttatása két végpont (pl. számítógép, intelligens nyomtató, router) között

- IPv4 esetén 32 bit hosszú:

  4×8 bites szakaszok decimális formátumban, ponttal („.”) elválasztva, pl.: 192.168.0.34

- IPv6 esetén 128 bit: 8×16 bites szakaszok hexadecimális formátumban, kettősponttal („:”) elválasztva

  pl.

  fe80:0000:0000:0000:0000:0000:0000:0001

  IPv6 címek tömörebb írásmódja:

  - a bevezető „0”-kat nem kell kiírni:

    pl: fe80::0012 → fe80::12

  - azon szakaszok, melyek csak 0-t tartalmaznak, rövidíthetők „::”-tal:

pl. fe80:0000:0000:0000:0000:0000:0000:0001
→ fe80:0:0:0:0:0:0:1 → fe80::1

- **hálózati maszk** (IPv4) vagy **prefix-hossz** (IPv6):

  azt határozza meg, hogy az IP-cím elején hány bit azonosítja a hálózatot, és az IP-cím végén hány bit azonosítja a hálózati eszközt

  - Példa #1:

Maszk:	255.255.255.0 (első 24 bit 1-es értékű)
Hálózat címe:	10.0.3.0
Node 1:	10.0.3.1
Node 2:	10.0.3.2
…	
Node 254:	10.0.3.254

Megjegyzés: ha a hálózati maszk első 24 bitje „1” értékű, akkor a maradék 8 biten találhatók a hálózati node-ok címe, tehát elvileg 256 darad node lehetséges, gyakorlatilag a **0. node a hálózatot azonosítja**, a 255. node pedig a broadcast cím (lásd később).

- Példa #2:

Prefix-hossz:	48 bit
Maszk:	ffff:ffff:ffff:ffff:ffff:ffff:0:0
Hálózat címe:	fe80::0
Node 1:	fe80::1 (azaz fe80:0:0:0:0:0:0:1)
Node 2:	fe80::2
…	…
Node N:	fe80::ffff:fffe

Megjegyzés: ha a hálózati maszk első 48 bitje „1” értékű, akkor a maradék 16 biten találhatók a hálózati node-ok címe, tehát elvileg 65536 darad node lehetséges, gyakorlatilag a **0. node a hálózatot azonosítja**, a 65535. node pedig a broadcast cím (lásd később).

- **hálózat címe**: ha az IP-cím node része 0, akkor a cím a hálózatot azonosítja (bármely node címe és a maszk közötti bitenkénti „ÉS” kapcsolattal számítható ki).

  A hálózati címet az átjárók és útvonalválasztók használják.

  Példák:

  - IPv4: 10.0.3.255
  - IPv6: fe80:: (azaz fe80:0:0:0:0:0:0:0)

- **broadcast cím**: ha az IP-cím node részének minden bitje „1” értékű, az IP-cím a hálózat broadcast címe. Ha egy IP-csomag „cél” mezejében a broadcast cím található, akkor azt a csomagot a hálózat minden node-ja fogadja

  Példák:

  - IPv4: 10.0.3.255
  - IPv6: fe80::ffff:ffff (azaz fe80:0:0:0:0:0:ffff:ffff)

Speciális IP-tartományok

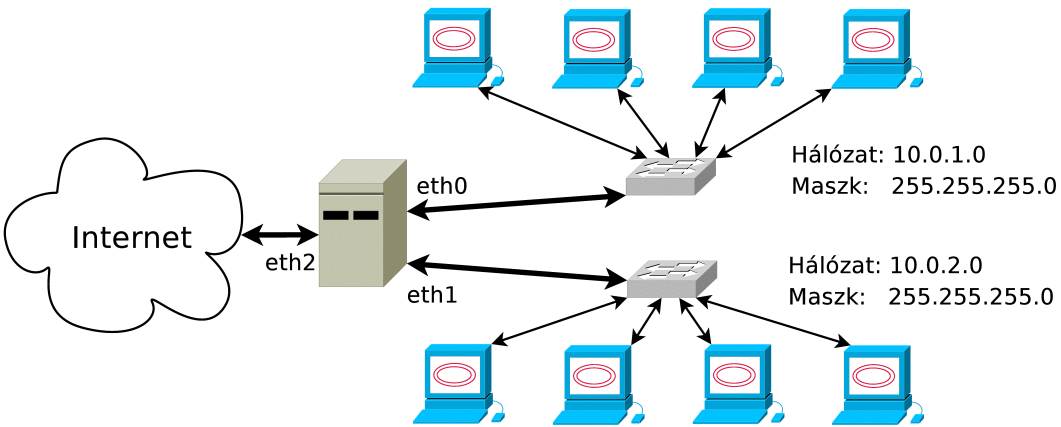
Bizonyos IP-tartományokat az IETF (Internet Engeneering Task Force) speciális célokra tart fenn:

- loopback (localhost)
  - 127.0.0.0/8
  - ::1/128
- link local IP-k (IP autokonfigurációra):
  - 169.254.0.0/16
  - fe80::/10
- site local (csak az adott szervezeten belül route-olják)
  - fc00::/7

- multicast IP-címek:
  - 224.0.0.0/4
  - FF00::/8
- privát hálózatok, szabadon felhasználhatók (RFC 1918):
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

Útvonalválasztás

Az alábbi hálózat arra mutat példát, hogy hogyan kell egy átjáró útvonalválasztását (routing) beállítani.



A **kliens node-ok útvonalválasztása** egyszerű, mert minden, nem nekik szóló IP-csomagot az egyetlen hálózati interfészük felé kell továbbítaniuk.

Az **átjáró** beállításának szabályai:

- ha az árjáróba/úrvonalválasztóba érkező csomag cél mezeje a 10.0.1.0/255.255.255.0 hálózatot címzi, akkor a csomagot az „eth0” interfészre kell juttatni (az „eth0” interfészen keresztül eljut a „felső” hálózatba, ahol – szerencsés esetben – a cél node található)

  Például:

  a 10.0.1.23 című node-nak szóló csomagot az „eth**0**” felé kell routolni, mert

  10.0.1.23 & 255.255.255.0 = 10.0.**1**.0 → eth**0**

- ha az árjáróba/úrvonalválasztóba érkező csomag cél mezeje a 10.0.**2**.0/255.255.255.0 hálózatot címzi, akkor a csomagot az „eth**1**” interfészre kell juttatni
- **minden mást** az „eth**2**” interfészen keresztül kell elküldeni (ez a default gateway, azaz az alapértelmezett átjáró az Internet felé)

A route-olási szabályokat a routing tábla tartalmazza (példa):

```
hg8lhs@santacitta:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.64.64.64     0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
10.0.3.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         0.0.0.0         0.0.0.0         U     0      0        0 ppp0
```

A fenti routing tábla jelentése:

- a `10.64.64.64` című node-nak (netmask 255.255.255.255, vagyis 1 db IP van az adott hálózatban → 1 db node) küldendő csomagokat a „ppp0” interfészen keresztül kell elküldeni
- a **10.0.3**.0 című hálózat az „eth0” hálózati kártyára „vannak kötve”
- minden más cél-című csomagot a „ppp0” eszközön át kell küldeni

Hálózati címfordítás – NAT

A NAT a „Network Address Translation” kifejezés rövidítése.

Típusai:

- **SNAT**: Source Network Address Translation: az árjáró a belső (privát) hálózatból érkező IP csomag **forrás IP-címét** kicseréli a saját, külső (publikus) IP-címével, a válaszként érkező csomagban pedig „visszacseréli” az IP-címeket.

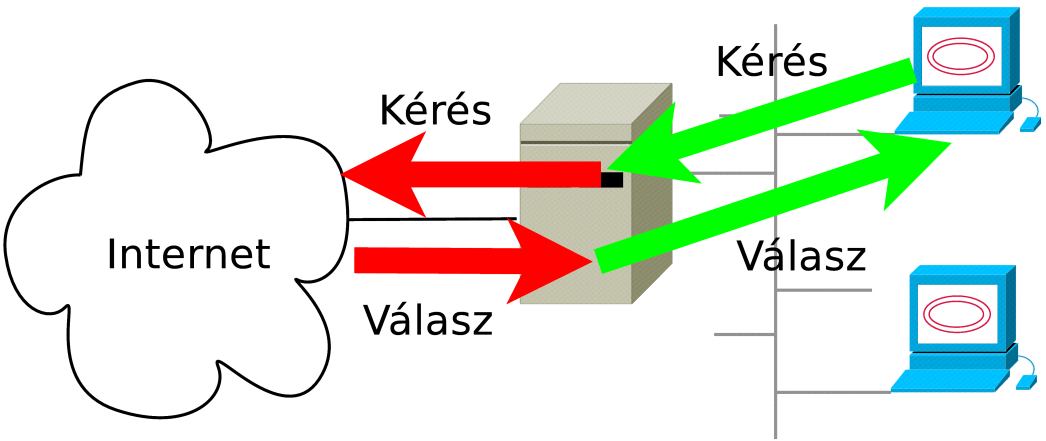
  Tehát a privát hálózatban levő node úgy látja, mintha közvetlenül a cél node-dal kommunikálna, a külső (cél) node pedig azt hiszi, hogy az átjáróval kommunikál. Ezzel a módszerrel elrejthetők a privát hálózat node-jai, növelhető a biztonság.

  Gyakran az IP-címek cseréje a TCP ill. UDP portok cseréjével is jár.

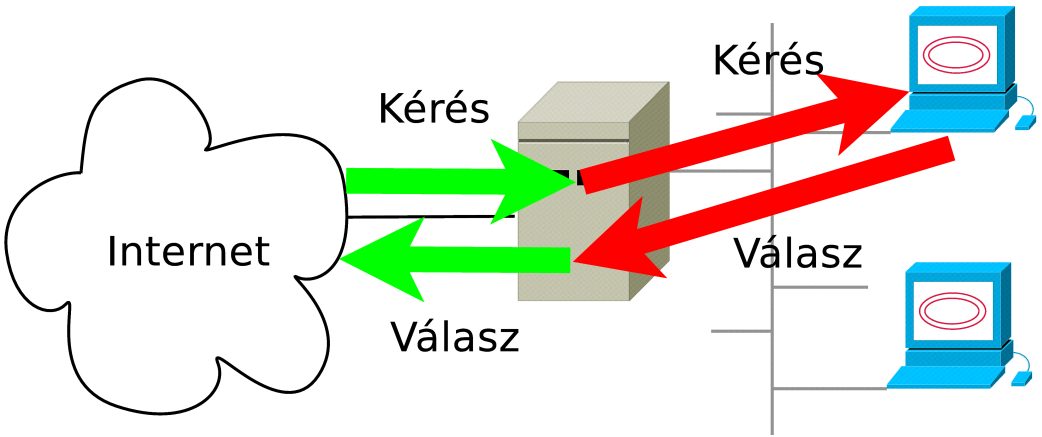
- **DNAT**: Destination Network Address Translation: az átjáró a külső hálózatból (pl. Internet) érkező IP-csomagban a **cél IP-címet** (= átjáró publikus IP-címe) kicseréli a privát hálózat egyik gépének IP-címével, majd az privát hálózat gépétől érkező válasz csomagban visszacseréli a címeket.

  Ezzel azt érhetjük el, hogy egy szolgáltatást több szerver láthat el: a „kívülről” érkező kéréseket a privát hálózat gépeihez továbbítjuk, azok pedig párhuzamosan tudják elvégezni a szerver-funkciókat. A megfelelő IP-cím kiválasztásának algoritmusa többnyire egyszerű körülforgó (round-robin, pl. NetBSD-ben) vagy terhelésfüggő (egy túlterhelt szerverre nem küld újabb kérést)
- **Port forwarding**: a DNAT speciális esete: nem csak IP-cím alapján szelektál, hanem TCP/UDP port alapján is. Ezzel elérhető, hogy bizonyos szolgáltatásokat (HTTP, FTP, SMTP) más-más szerver szolgáljon ki.

SNAT – Source NAT



DNAT – Destination NAT



Hálózati kiszolgálók

- Definíció: a számítógép-hálózat azon elemei, amelyek meghatározott szolgáltatást nyújtanak a többiek számára
- A szolgáltatások a szerver gép rögzített TCP és/vagy UDP portjain érhetők el. Ezeket a szolgáltatás-portcím párokat részben szabványosították (), részben „kialakultak”, de facto szabványok lettek
- A szolgáltatás-portcím hozzárendelések a `/etc/services` fájlban találhatók. Például:
  - echo – 7/tcp – 7/udp: válaszként visszaküldi azt, amit a kliens üzenetként küldött
  - daytime – 13/tcp – 13/udp: pontos időt szolgáltat
  - ftp – 21/tcp: fájlok feltöltése, letöltése (nem biztonságos)

- smtp – 25/tcp: levelek küldése
- domain – 53/tcp – 53/udp: gépnév→IP-cím konverzió
- www – 80/tcp: World Wired Web, világháló
- pop3 – 110/tcp: Post Office Protocol, levelek letöltése
- ntp – 123/tcp – 123/udp: Network Time Protocol, pontos idő lekérdezése
- netbios-ns – 137/tcp – 137/udp: Windows fájl megosztás :: név feloldás
- netbios-ssn – 139/tcp – 139/udp: Windows fájl megosztás, session layer (viszony réteg)

- Az én `/etc/services` fájlom 557 összerendelést tartalmaz.

Az NGW100

Az NGW100 név a „Network GateWay” kifejezésből ered. Képes hálózati átjáróként működni, mert a gyártó cég (ATMEL) két Ethernet csatlakozóval szerelte fel.

Főbb tulajdonságai:

- AP7000 (AVR32B rev. 1) processzor, 130 MHz-es órajellel
- 32 Mbájt SDRAM
- 8 Mbájt NAND FLASH (BOOT memória), JFFS2 fájlrendszerrel
- 8 Mbájt soros FLASH, JFFS2 fájlrendszerrel
- 2 db Ethernet MAC
- USB csatlakozó
- MMC/SD kártya foglalat

- 2 db SPI csatlakozó
- számos GPIO kivezetés
- soros konzol, JTAG csatlakozó debugolás céljából
- u-BOOT, előtelepített Linux (kernel, függvénykönyvtárak és felhasználói programok)

Szerver funkciói:

- WEB szerver
- Fájl szerver (Windows megosztás)
- ssh és telnet (hálózati bejelentkezés)

<EOF>

Köszönöm a figyelmet!

Ez a dokumentum a http://hg8lhs.ham.hu/tdk oldalról tölthető le.

Felhasznált szoftverek

- Ubuntu Linux 7.10 (kernel: 2.6.22-14_amd64)
- OpenOffice Writer 2.3.0
- The GIMP 2.4.2
- Dia 0.96.1
- Acrobat Reader for Linux 8.1.1