

# Számítógépes alkalmazások Hardver–Szoftver–Hálózat

Soós Sándor

Nyugat-magyarországi Egyetem  
Simonyi Károly Műszaki, Faanyagtudományi és Művészeti Kar  
Informatikai és Gazdasági Intézet

E-mail: [soossandor@inf.nyme.hu](mailto:soossandor@inf.nyme.hu)

Sopron, 2015.



# Tartalomjegyzék

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnyei
- Hálózatok osztályozása
- Hálózatok működése



# Miről lesz szó a mai órán? I

- A hardver és szoftver rendszer modellje
- A szoftver összetevői, a szoftverfejlesztés fázisai
- A gépi kódú és az assembly programozás lényege
- Magas szintű programnyelvek csoportosítása
- A szintaktika és a szemantika fogalma
- Az operációs rendszer és rendszer-közelí szoftver
- Az ismert operációs rendszerek jellemzői
- Az operációs rendszer csatlakozási és kezelői felülete
- A felhasználók csoportosítása és jellemzőik
- Virtualizáció
- Számítógépek összekapcsolása
- A hálózatok előnyei, jellemző topológiák



# Miről lesz szó a mai órán? II

- Hálózatok osztályozása
- Hálózati topológiák
- Hálózatok működése, alapszolgáltatások
- A hálózatok ISO-OSI modellje
- Gyakori hálózati szolgáltatások
- Az intranet fogalma



# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnyei
- Hálózatok osztályozása
- Hálózatok működése



# A számítógép hardver-szoftver felépítése

Felhasználói Interfész
Alkalmazói Programrendszer
Alkalmazói Programinterfész
Adatbázis kezelő rendszer
Operációs rendszer interfész
Operációs rendszer
Hardver interfész
Hardver

- A múlt órán beszéltünk a hardverről
- Most következik a szoftver



# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- **A szoftver**
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnyei
- Hálózatok osztályozása
- Hálózatok működése



# A szoftver

- Mi a különbség a program és a szoftver között?
- Miből áll a szoftver?
  - programmodulok
  - rendszerdokumentáció, fejlesztői dokumentáció
  - konfigurációs adatok, és ezeket tároló fájlok
  - felhasználói dokumentáció
  - a szoftver bevezetését és használatát támogató eszközök: weblap, fórum, oktatóanyagok, demók, stb.
- Ezt mindig figyelembe kell venni, ha szoftvert készítünk, vagy vásárolunk





# A szoftver

- Mi a különbség a program és a szoftver között?
- Miből áll a szoftver?
  - programmodulok
  - rendszerdokumentáció, fejlesztői dokumentáció
  - konfigurációs adatok, és ezeket tároló fájlok
  - felhasználói dokumentáció
  - a szoftver bevezetését és használatát támogató eszközök: weblap, fórum, oktatóanyagok, demók, stb.
- Ezt mindig figyelembe kell venni, ha szoftvert készítünk, vagy vásárolunk



# A szoftver

- Mi a különbség a program és a szoftver között?
- Miből áll a szoftver?
  - programmodulok
  - rendszerdokumentáció, fejlesztői dokumentáció
  - konfigurációs adatok, és ezeket tároló fájlok
  - felhasználói dokumentáció
  - a szoftver bevezetését és használatát támogató eszközök: weblap, fórum, oktatóanyagok, demók, stb.
- Ezt mindig figyelembe kell venni, ha szoftvert készítünk, vagy vásárolunk



# A szoftverfejlesztés fázisai

- 1 szoftverspecifikáció (Mit csináljon a szoftver?)
- 2 szoftvertervezés (A rendszer megtervezése)
- 3 implementáció (A programkód előállítása)
- 4 tesztelés, validáció (A program helyességének ellenőrzése, bizonyítása)
- 5 szoftverevolúció (üzemeltetés, továbbfejlesztés)



# A programozás szintjei

- gépi kódú programozás (*bitek*)
- assembly programozás (szimbolikus kódok, *mnemonik*)
  - assembly nyelv: a programozási nyelv
  - assembler: az assembly nyelvű fordító program  
assembly nyelvű forráskód  $\Rightarrow$  gépi kód
- magasszintű programozási nyelvek („*mondatok*”)



# A programozás szintjei – Példa

gépi kód:	10011100	0000101000110000	0000101000110001
assembly:	ADD	adat1	adat2
Pascal:	adat1 := adat1 + adat2;		
C:	adat1 = adat1 + adat2;		
Basic:	LET adat1 = adat1 + adat2		
Java:	adat1 = adat1 + adat2;		



# Magasszintű programnyelvek I

- Imperatív (procedurális) programnyelvek
  - a programozónak le kell írnia a végrehajtandó algoritmust
    - FORTRAN – 1954
    - ALGOL 60 – 1960
    - COBOL – 1959
    - BASIC – 1964
    - PASCAL – 1968
    - C – 1974
- Funkcionális (logikai) programnyelvek
  - nem a megoldást írja le a programozó, hanem a feladat logikai/matematika modelljét, a kielégítendő szabályokat
  - ezek után a feltett kérdésekre automatikusan tud válaszolni a program
    - Prolog
    - LISP



# Magasszintű programnyelvek II

- Szimulációs nyelvek
  - a feladat statisztikai modelljéből kiindulva kísérletezi ki a megoldást
    - GPSS
    - Simula 67
    - TUTSIM
    - ASYST
- Grafikus programozást biztosító nyelvek
  - két- és háromdimenziós ábrák előállításához szükséges paraméterezhető rajzelemek az elemi utasítások
    - LabVIEW
    - VisSim



# Magasszintű programnyelvek III

- Adatbázis-kezelő nyelvek
  - az adatbázis-kezelő szerverek programozását lehetővé tevő nyelvek
    - SQL
    - Progress
- Objektum alapú és objektum-orientált nyelvek
  - a feladat modelljét objektumokkal, ezek kapcsolataival és a hozzájuk köthető eseményekkel írják le
    - Smalltalk
    - Eiffel
    - C++
    - Java
    - Delphi (Object Pascal)
    - C#





# A programozási nyelvek definiálása

- A számítógéppel való kommunikációhoz teljesen egyértelműen kell definiálni a szabályokat
- A programozási nyelveket két szinten definiáljuk:
  - ① Szintaktika, szintaxis
    - nyelvtani szabályok
    - megmondja, hogy melyek a helyes programok
    - a fordítóprogram el tudja dönteni, hogy melyek a szintaktikailag helyes programok, hiba esetén jelzi ezeket
  - ② Szemantika
    - jelentés, tartalom
    - meghatározza, hogy a szintaktikailag helyes programok mit jelentenek
    - szemantikai hiba esetén nem kapunk hibaüzenetet, de a program nem azt csinálja, amit várunk tőle



# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnyei
- Hálózatok osztályozása
- Hálózatok működése



# Operációs rendszer és rendszerközeli szoftverek

- **operációs rendszernek** (rendszer-szoftvernek) nevezzük azoknak a gyári programoknak az összességét, mely a számítógépes eszközök és a hálózat működését, az erőforrások elosztását és igénybevételét vezérlik (pl: DOS, UNIX, WINDOWS..., NOVELL NETWARE, Mac OS, stb.)
- **rendszerközeli szoftvernek** nevezzük azoknak a gyári programoknak összességét, melyek általános és gyakran szükséges szolgáltatásokkal az operációs rendszerhez tartozó programokat kiegészítik (pl. Total Commander)
- nincs éles határ a két kategória között
- Miért nevezzük operációs rendszernek?



# Operációs rendszerek

*A számítógép alapműködését, az erőforrásaival optimálisan történő gazdálkodást a felhasználás céljától függetlenül biztosító, általános programok együttese az operációs rendszer*

- **MS-DOS** (Disk Operating System, 1981-): egyfeladatos, egyfelhasználós (single user, single tasking), assemblyben írták, parancsnyelv bázisú, interpreteres (a kapott utasítást azonnal végrehajtja, ha nem hibás a parancs)
- **Windows** (1985-): eredetileg egyfelhasználós, többfeladatos (multitasking), újabb változatai (XP) már többfelhasználósak (multiuser) assembly-ben és C-ben írták, grafikus bázisú (ikonok, menü, egér, ...)
- **UNIX** (1973-): többfelhasználós, többfeladatos, C-ben írták, nyílt (nyilvános) forráskódú, parancsnyelv bázisú; a PC-ktől a nagyszámítógépekig használt operációs rendszer.



# Az operációs rendszer környezete

- Három csatlakozási felületet kell megvalósítani (OS = Operating System):
  - 1 OS – Felhasználók (kezelők)
  - 2 OS – Alkalmazások (programok)
  - 3 OS – Hardver
- Fő feladata virtuális gép(ek) megvalósítása és működtetése
- Mit nevezünk virtuális gépnek?



# OS – Felhasználók – Felhasználói felület

- Ember – gép kapcsolat
  - tájékoztatni kell a felhasználót a gép működéséről
  - lehetővé kell tenni a beavatkozást
- Kezeleni kell az ehhez szükséges eszközöket
  - képernyő, billentyűzet, egér, hangeszközök, egyéb be-kimeneti eszközök
- A felhasználók nem egyformák:
  - „egyszerű” felhasználók
  - alkalmazás fejlesztők, programozók
  - rendszermenedzserek, rendszergazdák



## Kezelői felület / Egyszerű felhasználók I

- elsősorban programokat futtatnak, a felhasználói programokkal kerülnek kapcsolatba, az operációs rendszerrel kevésbé
- a legfontosabb operációs rendszer szolgáltatás számukra a fájlrendszer
- bizonyos megbízhatóságot, biztonságot elvárnak, cserébe hajlandók bizonyos szabályokat követni, pl. bejelentkezés, jogosultságok, stb.
- számukra az operációs rendszer lehetőséget ad programok és adatok tárolására, használatára



## Kezelői felület / Egyszerű felhasználók II

- az egyszerű felhasználók számára a legfontosabb operációs rendszer szolgáltatások:
  - bejelentkezés
  - fájlok, adatok kezelése
  - programok áttekintése
  - programok futtatása
  - programok párhuzamos futtatása
  - programok leállítása (kilövése)





## Kezelői felület / Alkalmazásfejlesztők

- természetesen a programozók egyben „egyszerű” felhasználók is, de ezen felül...
- elsősorban programokat írnak, tesztelnek, elemeznek
- az operációs rendszer programozói felületét ismerik, bizonyos mértékig a belső működését is
- többnyire speciális fejlesztői környezeteket használnak
- szükségük van az operációs rendszer szolgáltatásaira ahhoz, hogy adatokat, statisztikákat kapjanak a rendszer állapotáról (memória, erőforrások kihasználtsága, stb.)
- az alkalmazásfejlesztők számára a legfontosabb operációs rendszer szolgáltatások:
  - API-k (Application Programming Interface) a különböző rendszer erőforrások programozásához
  - programok futtatása különböző módokon (debug)



## Kezelői felület / Rendszermenedzserek

- természetesen a rendszermenedzserek egyben „egyszerű” felhasználók is, de ezen felül...
- Feladataik:
  - az operációs rendszer telepítése, konfigurálása
  - Adminisztráció: felhasználók, jogosultságok, más erőforrások
  - Hangolás: a rendszer teljesítményét befolyásoló paraméterek figyelése és beállítása
  - Felügyelet: hibák észlelése, elhárítása, log fájlok ellenőrzése, kezelése
  - Biztonsági mentések rendszeres készítése
- részletesen ismerik a rendszert
- olyan eszközökre van szükségük, amelyekre a többi felhasználónak nincs szüksége és ők nem is érhetik el azokat



## Különböző felhasználói felületek

- A felhasználók különböző csoportokba tartozhatnak (egyszerű felhasználó, programozó, rendszergazda)
- Egy felhasználó több csoportba is tartozhat
- A felhasználók felkészültsége, tapasztalata is nagyon eltérő lehet
- Ugyanaz a felhasználó is többféle felületet igényel az eltérő feladatokhoz
- Akkor milyen a jó felhasználói felület?
  - sokféle
  - jól paraméterezhető
  - karakteres-grafikus
- Melyik operációs rendszer tudja mindezt?
- Biztos, hogy egyetlen operációs rendszert kell használnunk?
- Hogyan használhatunk több operációs rendszert egy számítógépen?



# Virtualizáció

- Napjainkban az informatika talán leggyakrabban emlegetett fogalma, technológiája
- Nagyon sokféle dolgot értünk alatta
- „A virtualizáció számítástechnikai kifejezés, számítógépi erőforrások különböző absztrakcióinak gyűjtőneve”
- Mit lehet virtualizálni egy számítógépben?
  - egyes hardvereszközöket:
    - virtuális memória
    - virtuális nyomtató
    - tároló virtualizáció, virtuális kötetek, fájlrendszerek
  - egy teljes számítógépet:
    - szervervirtualizáció - virtuális szerver
    - desktop virtualizáció - virtuális munkaállomás, operációs rendszer
- Cloud Computing (erről külön fogunk beszélni)



# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnyei
- Hálózatok osztályozása
- Hálózatok működése



# A hálózatok előnyei

- Erőforrások megosztása, közös használata
  - Milyen erőforrásokat oszthatunk meg?
    - nyomtató, plotter, szkennер, ...
    - háttértárak
    - processzor
    - hálózat
    - alkalmazások
    - adatok, adatbázisok
  - Ezáltal csökkentjük a rendszerben lévő redundanciát!
- Erőforrások többszörözése, biztonság
  - háttértárak, RAID
  - hálózat
  - adatok, adatbázisok
  - Ez növeli a redundanciát
- Kommunikáció, online szolgáltatások elérése



# A hálózatok előnyei

- Erőforrások megosztása, közös használata
  - Milyen erőforrásokat oszthatunk meg?
    - nyomtató, plotter, szkennер, ...
    - háttértárak
    - processzor
    - hálózat
    - alkalmazások
    - adatok, adatbázisok
  - Ezáltal csökkentjük a rendszerben lévő redundanciát!
- Erőforrások többszörözése, biztonság
  - háttértárak, RAID
  - hálózat
  - adatok, adatbázisok
  - Ez növeli a redundanciát
- Kommunikáció, online szolgáltatások elérése



# A hálózatok előnye

- Erőforrások megosztása, közös használata
  - Milyen erőforrásokat oszthatunk meg?
    - nyomtató, plotter, szkennel, ...
    - háttértárak
    - processzor
    - hálózat
    - alkalmazások
    - adatok, adatbázisok
  - Ezáltal csökkentjük a rendszerben lévő redundanciát!
- Erőforrások többszörözése, biztonság
  - háttértárak, RAID
  - hálózat
  - adatok, adatbázisok
  - Ez növeli a redundanciát
- Kommunikáció, online szolgáltatások elérése





# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnye
- **Hálózatok osztályozása**
- Hálózatok működése



# Hálózatok osztályozása méret szerint

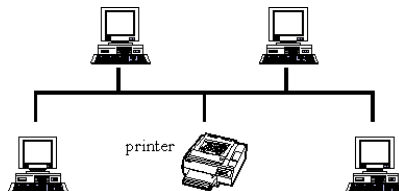
- LAN (Local Area Network / Helyi Hálózat)
  - kis kiterjedésű, épületen belül
  - nagy sebességű: 10-100 Mbps (megabit per mp)
- MAN (Metropolitan Area Network / Nagyvárosi Hálózat)
  - nagyobb kiterjedésű, két telephely között, egy városon belül
  - sebessége általában kisebb, mint a LAN esetében
  - tipikusan LAN hálózatokat kapcsol össze
- WAN (Wide Area Network / Nagy Kiterjedésű Hálózat)
  - nagy kiterjedésű, városok közötti, országokat, kontinenseket átívelő

Ez a hagyományos felosztás, ma már csak nagyon speciális esetekben van értelme saját MAN, vagy WAN hálózatot építeni az Internet használata helyett.



# Hálózati topológiák I

## 1 Bus (sín) topológia

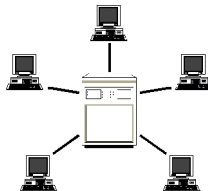


- minden eszköz egy közös bus-ra csatlakozik
- minden eszköznek van egy egyedi címe, a csomagot ezzel címezzük meg
- minden eszköz megkap minden csomagot, kiválasztja a neki szót
- speciális lezáró elem kell a kábel végére, hogy ne verődjenek vissza a csomagok
- ha bárhol megsérül a kábel, az egész hálózat leáll



# Hálózati topológiák II

## 2 Csillag topológia

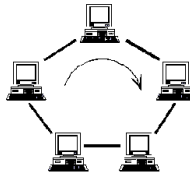


- minden eszköz közvetlenül a központban lévő szerverrel van összekapcsolva
- ha egy kábel megsérül, az csak egy munkaállomást érint
- a központ meghibásodásakor a teljes hálózat leáll



# Hálózati topológiák III

## 3 Gyűrű topológia

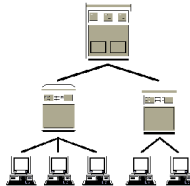


- minden eszköz két szomszédal áll kapcsolatban, a gyűrűnek nincsen végpontja
- a csomagok egy irányban haladnak körbe a gyűrűben, amíg el nem érik a címzettet
- ha egy helyen megsérül a kábel, akkor a teljes hálózat leáll



# Hálózati topológiák IV

## 4 Fa topológia



- az eszközök fastruktúrában vannak rendezve
- az egyes ágakat alhálózatoknak nevezzük
- aktív eszközök irányítják a csomagokat
- nagy hálózatok is kialakíthatók hatékonyan
- ha megsérül egy kábel, egy egész alhálózat elérhetetlenné válik



# Outline

## 1 Hardver-Szoftver

- A számítógép hardver-szoftver felépítése
- A szoftver
- Operációs rendszer

## 2 Hálózatok

- A hálózatok előnye
- Hálózatok osztályozása
- Hálózatok működése



# Hálózati címek, IP-címek

- A különböző hálózati rendszerek eltérő kommunikációs és azonosítási módokat használtak
- Mára szinte teljesen egyeduralkodóvá vált az Internet Protocol (IP) használata
- Minden Internet Protocol segítségével kommunikáló eszközt egy egyedi IP-cím azonosít
- Kétféle IP-cím szabvány:
  - IPv4
    - 32 bites egész szám, 4 darab (0-255 közötti) szám pontokkal elválasztva
    - 4 milliárd lehetséges IP-cím
  - IPv6
    - 128 bites egész szám, 8 darab 16 bites szám kettőspontokkal elválasztva, tizenhatos számrendszerben (hexa)
    - Hány lehetséges IP-cím van?  $2^{128} = 3,2 \times 10^{38}$





# IPv4 „régí” típusú IP-címek

- A 4 milliárd IP cím bőségesen elegendőnek tűnt, mára azonban elfogyóban vannak
- Nem adunk minden eszköznek önálló IP-címet:

Privát IP címtartományok	Tartomány kezdete	Tartomány vége	Címek száma
24 bites tömb (/8 prefix)	10.0.0.0	10.255.255.255	16 777 216
20 bites tömb (/12 prefix)	172.16.0.0	172.31.255.255	1 048 576
16 bites tömb (/16 prefix)	192.168.0.0	192.168.255.255	65 536



# IPv6 „új” típusú IP-címek

- 1994-ben vált szabvánnyá
- Gyakorlatilag korlátlan címtartomány
- Minden elképzelhető eszköz egyedi IP-címet kaphat
- A két rendszer párhuzamosan élhet egymás mellett
- Példák IPv6 címekre:
  - fe80:0000:0000:0000:0202:b3ff:fe1e:8329
  - fe80:0:0:0:202:b3ff:fe1e:8329, elhagyhatjuk a vezető nullákat
  - fe80::202:b3ff:fe1e:8329, elhagyhatjuk az egymás melletti 0 tagokat, de csak egyszer!
  - Most még szokatlanok ezek a számok, de meg fogunk barátkozni velük!



# Névszolgáltatás – DNS

- Az IP-címek emberi szem számára nehezen kezelhetők
- Megoldás: domain nevek és DNS – Domain Name Service
- Hierarchikus adatbázis
- Fa struktúrájú domain nevek
- Sok ezer szerver tárol (domain név – IP-cím) párokat
- Minden szerver a maga tartományának adatait tárolja és tartja karban
- Ha ismerünk egy domain nevet, akkor
  - megkérdezzük egy DNS szervert, hogy milyen IP-cím tartozik hozzá
  - vagy tudja, és visszaadja
  - vagy azt tudja, hogy melyik másik DNS szervertől kell megkérdezni, és továbbítja a kérést
  - így néhány lépés után megkapjuk a választ
  - *a valóságban kicsit bonyolultabb a dolog, de ez az alapelv*

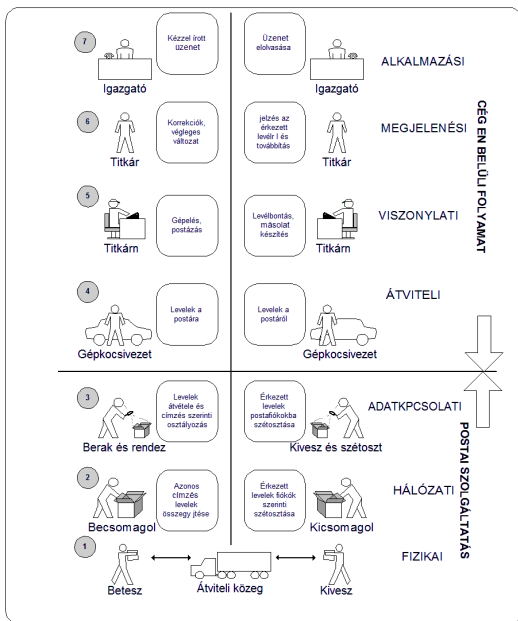


# A hálózati kommunikáció szabványos modellje

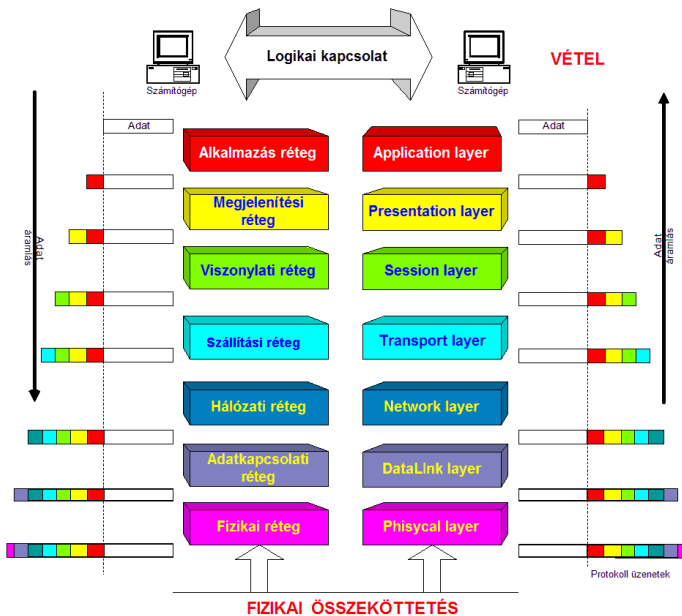
- ISO-OSI hétrétegű modellje
- Szabványokban definiálja két hálózati eszköz kommunikációjának módját
- ISO: International Organization for Standardization, Nemzetközi Szabványügyi Szervezet (iso görögül egyenlőt jelent)
- OSI: Open Systems Interconnection Reference Model, Nyílt rendszerek Összekapcsolása, referencia modell
  - az Internet működésének technikai alapja
  - minden eszköz összekapcsolható minden más eszközzel
  - Hogyan lehetséges ez?
    - 7 szintre osztja a hálózati kommunikációt
    - minden szint működését szabványok rögzítik
    - minden szint csak a közvetlenül alatta és felette lévő réteggel kommunikálhat, nem léphet át más szintekre



# ISO-OSI példa: Vállalati kommunikáció



# ISO-OSI hétrétegű modell



# Alkalmazások, protokollok és az egyes OSI rétegek

Réteg	Fontos példa	TCP/IP készlet	SS7	AppleTalk készlet	OSI készlet	IPX készlet	SNA	UMTS
7 - Alkalmazás	HL7, Modbus, SIP	HTTP, SMTP, SMPP, SNMP, FTP, Telnet, NFS, NTP	ISUP, INAP, MAP, TUP, TCAP	AFP, PAP	FTAM, X.400, X.500, DAP		APPC	
6 - Megjelenési	TDI, ASCII, EBCDIC, MIDI, MPEG	XDR, SSL, TLS		AFP, PAP	ISO 8823, X.226			
5 - Viszonylati	Named Pipes, NetBIOS, SAP, SDP	Viszonylat kiépítés TCP-vel		ASP, ADSP, ZIP	ISO 8327, X.225	NWLlink	DLC?	
4 - Átviteli	NetBEUI	TCP, UDP, RTP, SCTP		ATP, NBP, AEP, RTMP	TP0, TP1, TP2, TP3, TP4, OSPF	SPX, RIP		
3 - Hálózati	NetBEUI, Q.931	IP, ICMP, IPsec, ARP, RIP, BGP, OSPF	MTP-3, SCCP	DDP	X.25 (PLP), CLNP	IPX		RRC (Radio Resource Control)
2 - Adatkapcsolati	Ethernet, Token ring, FDDI, PPP, HDLC, Q.921, Frame Relay, ATM, Fibre Channel		MTP-2	LocalTalk, TokenTalk, EtherTalk, Apple Remote Access, PPP	X.25 (LAPB), Token Bus	IEEE 802.3 framing, Ethernet II framing	SDLC	MAC (Media Access Control)
1 - Fizikai	RS-232, V.35, V.34, Q.911, T1, E1, 10BASE-T, 100Base-TX, ISDN, SONET, DSL		MTP-1	LocalTalk árnyékolt, LocalTalk árnyékolás nélküli kábelben (PhoneNet)	X.25 (X.21bis), EIA/TIA-232, EIA/TIA-449, EIA-530, G.703		Twinax	PHY (Physical Layer)

# Miért jó, ha megismerünk egy ilyen rendszert

- Jobban átlátjuk, hogy mi történik a háttérben, amikor szörfölünk az Interneten
- Jobban értjük, amikor aláírjuk a szerződést az Internet szolgáltatóval
- Példa arra, hogy hogyan lehet megtervezni egy nagy rendszert
- Ezt az elvet felhasználhatjuk máshol is
- Példaként lásd a vállalati kommunikációról szóló példát!





# Hálózati szolgáltatások I

- Hagyományos szolgáltatások:
  - Elektronikus levelezés
  - FTP - File Transfer Protocol
  - News, newsgroups - Hírcsoportok
  - Gopher - A www előfutára, hierarchikus menürendszerben tette elérhetővé az internetes tartalmakat
  - WWW – World Wide Web
- Manapság minden hagyományos szolgáltatást magába olvasztott a World Wide Web
- Szinte minden szolgáltatást egyetlen böngésző programból érünk el
- Az e-mail és az FTP maradt használatban részben a www-tól függetlenül is, de ezeknek is megvan a webes megfelelőjük

# Hálózati szolgáltatások II

- Új szolgáltatások:
  - Azonnali üzenetküldés
  - Videó átvitel
  - Távkonferencia
  - Keresés
  - Szemantikus web
  - Web2 – Web 2.0
  - Cloud computing - Számítási felhő
  - Internet of Things (IoT) - Tárgyak Internete
  - ...



# Heterogén hálózatok

- Egy hagyományos vállalati rendszer 10 évvel ezelőtt:
  - belső vállalati hálózat:
    - megosztott szerverek, nyomtatók, stb.
    - belső kommunikáció, levelezés
    - elektronikus faliújság
    - céges telefonkönyv, a menza étlapja, hírek, megosztott fájlokban a szervereken
  - egyes gépeken, vagy minden gépen van Internet hozzáférés is:
    - webes keresés
    - kommunikáció, chat, Skype, stb.
    - a cég honlapja
    - partnerek honlapjai, egyéb internetes szolgáltatások
  - A cég informatikusai üzemeltetnek egy belső hálózatot a munkatársak számára, esetleg belépési lehetőséggel a partnercégek munkatársai számára
  - Ehhez kitalálnak, megvalósítanak és üzemeltetnek különböző egyedi megoldásokat



# Intranet

- Ötlet: Miért kell más megoldásokat használnunk a belső hálózaton, mint amit az Interneten használunk?
- Hagyomány: az Internet elterjedése előtt épültek ki a helyi hálózatok egyedi technikákkal, más protokollokat használva
- Intranet: olyan belső hálózat, ami ugyanazokat a módszereket, hardver és szoftver eszközöket használja, mint az Interneten
  - TCP-IP protokoll a hálózati eszközök közötti kommunikációhoz
  - HTTP protokoll a hipertext dokumentumok továbbításához
  - A munkaállomásokon böngésző program fut
  - A szervereken webszerver, levelezőszerver, FTP szerver fut
  - Kommunikációhoz használhatjuk az e-mailt, chat-et, Skype-ot, MSN-t stb.
- Mi választja el a külső és a belső hálózatot?
  - Tűzfal – Firewall



# Intranet

- Ötlet: Miért kell más megoldásokat használnunk a belső hálózaton, mint amit az Interneten használunk?
- Hagyomány: az Internet elterjedése előtt épültek ki a helyi hálózatok egyedi technikákkal, más protokollokat használva
- Intranet: olyan belső hálózat, ami ugyanazokat a módszereket, hardver és szoftver eszközöket használja, mint az Interneten
  - TCP-IP protokoll a hálózati eszközök közötti kommunikációhoz
  - HTTP protokoll a hipertext dokumentumok továbbításához
  - A munkaállomásokon böngésző program fut
  - A szervereken webszerver, levelezőszerver, FTP szerver fut
  - Kommunikációhoz használhatjuk az e-mailt, chat-et, Skype-ot, MSN-t stb.
- Mi választja el a külső és a belső hálózatot?
  - Tűzfal – Firewall



# Befejezés

Köszönöm a figyelmet!

