



PÁZMÁNY PÉTER
KATOLIKUS EGYETEM

BEVEZETÉS A SZÁMÍTÁSTECHNIKÁBA

PPKE, ITK, 2016. őszi félév
Papp Áron

1. ÓRA
BEMUTATKOZÁS,
BEVEZETÉS

BEMUTATKOZÁS

- A tantárgy felelőse: Dr. Siklósi Borbála
Email: siklosi.borbala@itk.ppke.hu
<http://users.itk.ppke.hu/~sikbo>
Szoba: 314
- Gyakorlatvezetők:
 - Nika Zsolt
Email: nika.zsolt@itk.ppke.hu
<http://users.itk.ppke.hu/~nikzs>
 - Papp Áron
Email: papp.aron@itk.ppke.hu
<http://users.itk.ppke.hu/~papar>

LABORREND

Csoport	Idő / hely	Gyakorlatvezető
06	2016/17/1 K:14:15-16:00(ITK 422 PC labor)	Nika Zsolt
05	2016/17/1 K:12:15-14:00(ITK 422 PC labor)	Papp Áron
07	2016/17/1 K:10:15-12:00(ITK 422 PC labor)	Papp Áron
01-02MB	2016/17/1 K:12:15-14:00(ITK 219 PC labor)	Dr. Siklósi Borbála
03-04MB	2016/17/1 K:10:15-12:00(ITK 219 PC labor)	Dr. Siklósi Borbála
DB	2016/17/1 H:10:15-12:00(ITK 219 PC labor)	Dr. Siklósi Borbála
08	2016/17/1 K:10:15-12:00(ITK 302 PC labor)	Nika Zsolt

SZÁMONKÉRÉS

- ZH-k
 - 2 db lesz, mindkettő max. 100 pont
 - Az első az őszi szünet környékére fog esni
- RöpszH-k
 - Minden óra elején, “az előző rész tartalmából”
 - Pontozás: -1 / 0 / +1
- Aláírás feltétele
 - Max. 3 hiányzás
 - RöpszH-k pontszámainak összege pozitív legyen
 - Mindkét ZH-t min. 50%-ra kell megírni
(az egyiket lehet pótolni)

SZÁMONKÉRÉS

- Értékelés
 - 51 - 60 : 2
 - 61 - 70 : 3
 - 71 - 85 : 4
 - 86 - : 5
- Programozási feladat
 - Választható
 - A feltételek teljesülése esetén a végső pontszámhoz max. 10 ponttal járul hozzá
 - bash nyelven kell elkészíteni
- Órai aktivitás
 - Plusz ponttal jutalmazzuk :)

A FÉLÉV RENDJE

1. Bemutakozás, bevezetés, ITK-s alapok
2. Számrendszerek, számábrázolás, karakterkódolás
3. Számítógép generációk, operációs rendszerek
4. Hardver alapismeretek
5. Adattárolás és fájlrendszerek
6. Linux alapok
7. Linux alapok - folyt.
8. LaTeX 1.
9. LaTeX 2.
10. LaTeX 3. MB
11. LaTeX 3. DB
12. Szöveget tartalmazó fileok
13. Hálózatok

GÉPTERMI HÁZIREND

- A gépteremben csak a kurzusra feliratkozott hallgatók tartózkodhatnak
- Enni, inni tilos!
- Gyakorlatilag minden tilos, csak a gépek rendeltetés szerű használata engedélyezett
- Szándékos rongálás (hardware / software) esetén fegyelmi eljárás kezdeményezhető
- Bejelentkezéshez mindenki az egyetemi azonosítóját használja

SECURE SHELL

- Mi az SSH?
 - A Secure Shell egy szabványcsalád, és egyben egy protokoll is, amit egy helyi és egy távoli számítógép közötti biztonságos csatorna kiépítésére fejlesztettek ki
 - Nyilvános kulcsú titkosítást használ a távoli számítógép hitelesítésére
 - Opcionálisan a távoli számítógép is hitelesítheti a felhasználót
- Putty
 - Terminál emulátor
 - Különböző protokollokat támogat, mi SSH-n fogjuk elérni a szervert
- WinSCP
 - Secure copy
 - Fájlvitel SSH-n keresztül

GNU/LINUX



- A GNU egy komplett operációs rendszer
- A Linux egy UNIX-szerű, nyílt forráskódú rendszermag
- A GNU projectet Richard Matthew Stallman 1983-ban indította
- A Linux-ot Linus Torvalds kezdte fejleszteni 1991-ben
- Különböző elektronikus eszközökben megtalálható
 - Hálózati eszközök (routerek, switchek, ...)
 - Hordozható eszközök (mobiltelefonok, okostelefonok, PDA-k, tabletek, órák, navigációs készülékek, ...)
 - Háztartási gépek, szórakoztató elektronikai berendezések, konzolok, set-top boxok, ...)
 - Szerverek, személyi számítógépek
 - Szuperszámítógépek

2. ÓRA

SZÁMRENDSZEREK,
SZÁMÁBRÁZOLÁS,
KARAKTERKÓDOLÁS

A SZÁMRENDSZER

- A számrendszer [numeral system] matematikai fogalom: egy módszer, melynek célja az írott formában történő megjelenítés
- Helyiértéken (pozíción) alapuló számrendszereket tárgyalunk (a római számok például sorrendiségen alapulnak)
- A számrendszer alapja [base, radix] meghatározza az egyes pozíciókba írható számjegyek maximumát
- A számjegyek [digit] a számok írására használt karakterek
- A számítástechnikában gyakran használt számrendszerek:
2, 8, 10, 16

ALAKI- ÉS HELYIÉRTÉK

- Számjegy értéke = alakí értéke \cdot helyiérték
- Alaki értéke: számjegyhez tartozó értéke
- Helyiérték: a számrendszer alapjának pozíció szerinti hatványa

EGÉSZ ÉS TÖRT SZÁMOK

- Egész számok felírása

$$a_n \ a_{n-1} \ a_{n-2} \ \dots \ a_1 \ a_0$$

- Egész számok értéke

$$(a_n \cdot A^n) + (a_{n-1} \cdot A^{n-1}) + \dots + (a_1 \cdot A^1) + (a_0 \cdot A^0)$$

- Tört számok felírása

$$a_n \ a_{n-1} \ \dots \ a_1 \ a_0 \ a_{-1} \ \dots \ a_{-k}$$

- Tört számok értéke

$$a_n \cdot A^n + a_{n-1} \cdot A^{n-1} + \dots + a_1 \cdot A^1 + a_0 \cdot A^0 + a_{-1} \cdot A^{-1} + \dots + a_{-k} \cdot A^{-k}$$

ÁTVÁLTÁS 10 \rightarrow 2 SZÁMRENDSZERBE

177		1
88		0
44		0
22		0
11		1
5		1
2		0
1		1
0		

$$177_{(10)} = 1011\ 0001_{(2)}$$

PONTOSSÁG

- Nem egész számok nem mindig írhatóak fel véges számjeggyel
- Nem egész számok pontosság függ a számrendszer alapjától
- Pl.:

$$1/3_{(10)} = 0.333..._{(10)} = 0.1_{(3)}$$

- Keressünk további példákat!

AZ ADATMENNYISÉG MÉRTÉKEGYSÉGEI

- Alapegység: bit (true/false)
- 1 Byte = 8 bit

SI		Bináris	
Prefix	Szorzó	Prefix	Szorzó
K (kilo)	1000	Ki (kibi)	1024
M (mega)	1000 ²	Mi (mebi)	1024 ²
G (giga)	1000 ³	Gi (gibi)	1024 ³
T (tera)	1000 ⁴	Ti (tebi)	1024 ⁴
P (peta)	1000 ⁵	Pi (pebi)	1024 ⁵

GÉPI SZÁMÁBRÁZOLÁS

- Nem negatív egész számok: megegyezik a bináris számok leírásával, tehát 2-es számrendszerben tároljuk
- Összeadás művelete nemnegatív egész bináris számokon

```
  1001 1011
+ 0110 1100
-----
1 0000 0111
```

- Hogyan dönthetjük el, hogy melyik bináris szám nagyobb?

GÉPI SZÁMÁBRÁZOLÁS

- Negatív számok előjelbites ábrázolása
 - A legnagyobb helyiértéken a szám előjelét tároljuk
 - A maradék helyiértékeken a számot
 - Pl. 8 bites változóméret esetén: $-32_{(10)} = 1010\ 0000_{(2)}$
- 2-es komplement ábrázolás
 - A szám abszolút értékének a számjegyeit invertáljuk
 - Majd hozzáadunk 1-et a számhoz
- A 2-es komplement előnye: nincs szükség kivonás műveletre
 - Pl. : $19_{(10)} - 9_{(10)} \qquad 0001\ 0011_{(2)} - 0000\ 1001_{(2)}$

Kivonandó:	0000 1001	0001 0011	A túlcsordulást levágva az
1-es kompl.:	1111 0110	+1111 0111	eredmény: $1010_{(2)}$ vagyis $10_{(10)}$
2-es kompl.:	1111 0111	-----	

1 0000 1010

MSB/LSB

- MSB: Most Significant Bit
- LSB: Least Significant Bit
- Architektúránként eltér
- Azt jelzi, hogy a fizikai tárolás vagy hálózati továbbítás során a változóhoz tartozó adott méretű memóriaterület címéhez tartozó Byte a számnak a legértékesebb vagy a legkevésbé értékes helyiértékét jelöli
- Endianness: big-endian / little-endian
(-> Swift: Gulliver utazásai)

TÚLCSORDULÁS

- Egész számok ábrázolása esetén meghatározott számú biten korlátozott az ábrázolható számok nagysága
- Lehetséges alul- vagy túlcsordulás
- A megvalósítástól függően eredményezhet
 - levágást: a nem ábrázolható rész levágjuk, nem használjuk fel
 - szaturációt: a legnagyobb vagy legkisebb ábrázolható értéket tároljuk

LEBEGŐPONTOS SZÁMÁBRÁZOLÁS

- Normalizált alak: a szám felbontása egy kéttagú szorzatra, ahol a második tag a számrendszer alapjának valamely hatványa, melyet, ha az első számmal beszorzunk az eredeti számot kapjuk vissza. A tizedes pont előtt csak 1 számjegy lehet.

$$\text{Pl. : } 380_{(10)} = 3.8 * 10^2$$

- Az így kapott eredményt bináris formában tároljuk el, és a legértékesebb helyiértéken jelöljük a szám előjelét
- IEEE 754: {előjel} {exponens} {mantissza}
- Bináris szám esetén a tizedespont előtt csak 1-es lehet
- NaN: Not a Number

KARAKTEREK ÉS KARAKTERKÉSZLETEK

- A karakter a számhoz hasonlóan fogalom, a karakterkészlet egy kódtáblázat, ahol az egyes karakterekhez kódszámot rendelnek
- ASCII kódtábla
 - American Standard Code for Information Interchange (1960-as évek)
 - Alapesetben 7-bites, az extended változat 8-bites
- Unicode
 - A cél egy kódtáblában tárolni a világ összes betűjelét
 - Az Unicode egy kódolási szabvány, nem kódtábla
- UTF-8
 - Változó 1-6 Byte hosszú tárolás
 - ASCII kompatibilis és önszinkronizáló (nem kell a string elejéről kezdeni az olvasást, hogy elkülönüljenek a karakterek)

3. ÓRA
OPERÁCIÓS
RENDSZEREK

OPERÁCIÓS RENDSZER FOGALMA

- Ajánlott irodalom: Tanenbaum - Operációs rendszerek
- A számítógépet közvetlenül gépi nyelv szintjén programozhatjuk, ez azonban kényelmetlen
- Minden alrendszer (utasításkészlet, memóriaszervezés, I/O rendszer, sínstruktúra) a programozónak kellene lekezelnie a programjában, de nem akarja
- Az operációs rendszer elrejtí előlünk a hardvert: absztrakciós rétegeket hoz létre, hogy a hardver elérése a felhasználói programokból egyszerű legyen
- Kiterjesztett vagy virtuális gépet biztosít a felhasználónak

OPERÁCIÓS RENDSZER MINT ERŐFORRÁS-KEZELŐ

- “felülről lefelé” nézőpont: kényelmes csatlakoztatási felület a felhasználók számára
- “alulról felfelé” nézőpont: az operációs rendszer célja, hogy az összetett rendszer minden részét kezelje
- Erőforrások megosztása:
 - időalapú: az erőforrások felváltott használata
 - téralapú: az erőforrás részekre osztása
- A hacker támadások egy része azt használja ki, hogy az adott program átlépi a saját hatáskörét, és hozzáfér egyéb folyamatokhoz rendelt erőforrásokhoz

OPERÁCIÓS RENDSZEREK TÖRTÉNELME I.

- Számítógép generációkon keresztül tekintjük at az op.rsz-ek fejlődését
- I. generáció (1945–1955): Vákumcsövek és kapcsolótáblák
- Howard Aiken, Neumann János, J. Presper Eckert, John William Mauchley és Konrad Suse számítógépek építésében értek el sikereket
- Először relék, majd vákumcsövek alkották a gépeket
- Programozás gépi nyelven történt, nem voltak programozási nyelvek sem
- Eleinte kapcsolótáblákat cserélgették, majd az 50-es években megjelentek a lyukkártyák

OPERÁCIÓS RENDSZEREK TÖRTÉNELME II.

- II. generáció (1955–1965): tranzisztorok és kötegeltek rendszerek
- Ma ezeket a gépeket nevezzük mainframe-eknek
- Assembly, FORTRAN nyelvek használata
- Lyukkártyán, később szalagon vitték be a programokat
- Az eredmény a nyomtatóra ment
- Tudományos és mérnöki számításokra használták, pl. partiális differenciálegyenletek numerikus megoldására
- Op.rsz-ek: Fortran Monitor System és IBSYS (IBM rendszere a 7094-re)

OPERÁCIÓS RENDSZEREK TÖRTÉNELME III.

- III. generáció (1965–1980): integrált áramkörök
- A korábbi szó-orientált gépek mellett megjelentek a karakter-orientált gépek (pl. 1401), melyeket bankmok és biztosítók használtak szalagrendezésre, nyomtatásra
- A két irányvonalat az IBM egy közös rendszerrel a System/360-nal egyesítette
- Később megjelent a 370, 4300, 3080, 3090, és a mai is kapható System Z
- Multiprogramozás: amíg a processzor egy művelet eredményét várt, azalatt a memória egy másik részében számítást végzett



OPERÁCIÓS RENDSZEREK TÖRTÉNELME III.

- Spooling (Simultaneous Peripheral Operation On Line): a kártyákról a feladatokat lemezre másolták, így amikor egy feladat befejeződött a háttértárról gyorsan be tudta olvasni a következőt
- Időosztás (Time sharing): a processzor órajelén egyszerre több felhasználó/alkalmazás osztozik
- LAN (Local area network): helyi hálózat tette lehetővé a fájlkiszolgáló szerverek működését
- Közvetítő réteg (Middleware): a lokális felhasználókat kötötte össze a távoli erőforrásokkal

OPERÁCIÓS RENDSZEREK TÖRTÉNELME III.

- Mindeközben megjelentek a miniszámítógépek is, a DEC PDP sorozat jegyében, melyeket a korábbi nagygépek árának töredékéért lehetett megvásárolni
- Ken Thompson a Bell Labs munkatársa egy PDP-7-en kezdett el programozni egy egyfelhasználós rendszert, ami a Unix operációs rendszer t torkollott
- A Unix forrása nyílt volt, mindenki saját, inkompatibilis változatot kezdett fejleszteni
- Két fő változat: System V és a BSD
- Az IEEE POSIX nevű szabványát a legtöbb mai Unix betartja
- “Unix: Live free or die”

OPERÁCIÓS RENDSZEREK TÖRTÉNELME IV.

- IV. generáció (1980-): személyi számítógépek
- Az LSI (Large Scale Integration) áramkörök fejlődésével megérkezett a mikroprocesszor alapú személyi számítógépek kora: bárkinek lehetett saját gépe
- 8 bitesek: Intel 8080 (1974-ben jelent meg), Zilog Z80 (CP/M rendszert futtatott), Motorola 6800, MOS 6502
- 16 bitesek: Intel 8086, Intel 8088 (IBM PC)
- A Microsoft felvásárolt a DOS rendszert és MS-DOS néven adta ki, ami gyorsan elterjedt az IBM PC-ken
- Az MS-DOS tartalmazott a BASIC nyelv támogatást

OPERÁCIÓS RENDSZEREK TÖRTÉNELME IV.

- Doug Engelbart vetette fel pár évvel a DOS előtt a grafikus felhasználói felület ötletét (GUI), és Steve Jobs látta meg ebben a lehetőséget: 1984-ben jelent meg az Apple Macintosh
- Az első Mac a Motorola 68000 processzorát tartalmazta
- Később átváltottak az IBM 32, majd 64 bites RISC processzraira (PowerPC) és megjelent a Berkely Unix-ra épülő Mac OS X
- 2005-ben bejelentette az Apple, hogy átáll az Intel CPU-kra
- A Microsoft piacra dobta a Windows-t, hogy versenyben maradjon az Apple-el szemben

OPERÁCIÓS RENDSZEREK TÖRTÉNELME IV.

- Amikor a Unix forrását lezárta az AT&T a hallgatók előtt Prof. Tanenbaum hozzáfogott a Minix fejlesztéséhez
- A Minixet kisméretűnek tartották meg, hogy a hallgatók is tudják futtatni gépeiken, így tett az egyik hallgató Linus Torvalds is
- Torvaldsnak hiányzott pár funkció a Minixből, amikre programot írt, majd egy másfajta terminálmeghajtót is készített, később egy lemezmeghajtót és fájlrendszert
- Az eredményeket a USENET-en comp.os.minix csoportban közzétette, és segítőkre lelt
- 1994. március 13-án megjelent a Linux 1.0 :)

FOGALMAK: PROCESSZUS

- Végrehajtás alatt lévő program
- Hozzá tartozik egy címtartomány (a memória egy szelete), amin belül a processzus írhat/olvashat
- Van egy regiszterkészlete is (utasításszámláló, veremmutató, ...)
- Egy adott processzushoz tartozó információkat az op.rsz. a processzustáblázatban tárol, és ezeket az adatokat használja fel, amikor az időosztás során az adott processzusnak újból ad egy CPU időszeletet
- A processzusok fa-strukturát alkotnak (szülő és gyerek processzusok)

FOGALMAK: FÁJLOK

- Az op.rsz. feladata, hogy az I/O műveletek felett egy fájlrendszer absztrakciót biztosítson
- Rendszerhívásokkal lehet fájlokat létrehozni, törölni, olvasni és írni
- A fájlok könyvtárakba vannak szervezve és POSIX rendszereken valamennyi fájl az útvonalja segítségével elérhető az ún. gyökérkönyvtárból
- Jogkezelés: owner, group, other / read, write, execute
- Specifikus fájlok: block- és karakterspecifikus lehet
- Adatcső: az egyik processzus kimenetét a másik bemenetére irányítja

FOGALMAK: PARANCSÉRTELMEZŐ

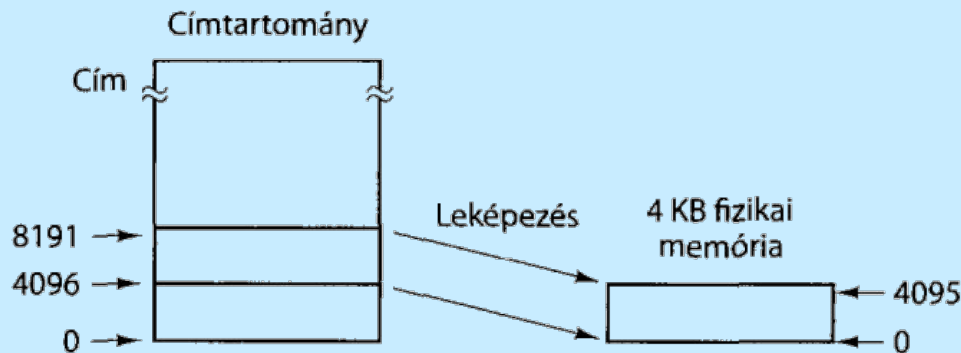
- Kapcsolódási felület a felhasználó és az operációs rendszer magja között
- POSIX rendszereken shell-nek hívják
- Többféle változatok: sh, csh, zsh, ksh, bash, ...
- A prompt jelzi, hogy az értelmező várja az utasítást
- A grafikus felhasználói felületek is gyakorlatilag parancsértelmezők

FOGALMAK: RENDSZERHÍVÁSOK

- A rendszerhívásokkal a felhasználói programok jelzik a rendszermag számára, hogy feladatot kell végrehajtson
- Gyakorlatilag olyan eljáráshívások, amik a magba más privilegizált op.rsz. komponensbe tudnak belépni
- Csoportjai:
 - Processzuskezelő rendszerhívások
 - Szignálkezelő rendszerhívások
 - Fájlkezelő rendszerhívások
 - Könyvtárkezelő rendszerhívások
 - A védelem rendszerhívásai
 - Az időkezelés rendszerhívásai

FOGALMAK: VIRTUÁLIS MEMÓRIA

- Az 1950-es években a programozóknak muszáj volt akkora részekre bontani a programot, hogy elférjen a memóriában – ezt nevezzük átfedésnek (overlays)
- A módszer mai napig megmaradt, csak automatizálva lett – 1961-ben Fotheringham által (virtuális memória, lapozás)
- Ezáltal megtörtént a címtartomány és a memóriarekeszek fogalmának különválasztása



FOGALMAK: VIRTUÁLIS MEMÓRIA

- A lapozás lépései:

- A memória tartalmának lemezre mentése.
- A 8192 és 12287 közti szavak megkeresése a lemezen.
- A 8192 és 12287 közti szavak betöltése a memóriába.
- A memóriatérkép megváltoztatása; a 8192 és a 12287 közti címek leképezése a 0 és 4095 közti memóriarekeszekre.

- Fogalmak:

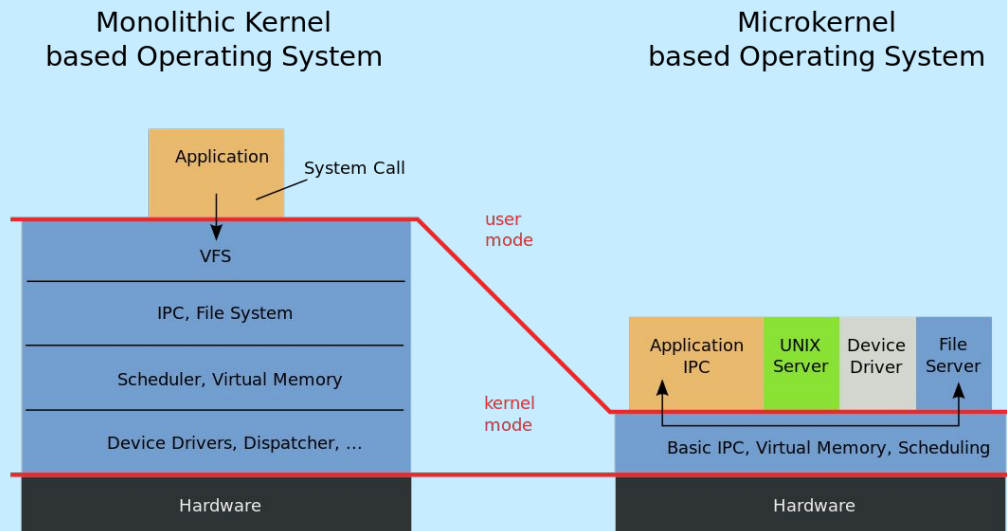
- A program a virtuális címtartományra hivatkozhat
- A memóriarekeszeket a fizikai címtartomány címzi meg
- A memóriatérkép az egyes virtuális címeknek megfelelő fizikai címeket határozza meg

STRUKTÚRÁK: MONOLÍTIKUS RENDSZEREK

- “Struktúrája a struktúrálatlanság”
- Az operációs rendszer eljárások gyűjteménye, bármelyik hívhatja a másikat korlátozás nélkül
- A paraméterek és a visszaadott érték alapján minden eljárásnak jól definiált felülete van, ha a programozó úgy gondolja, hogy eljárásában egy másik eljárás valami hasznosat nyújthat, akkor azt szabadon hívhatja

STRUKTÚRÁK: MICROKERNEL

- Szemben a monolítikus rendszerekkel a mag méretét minimalizálják, és külső forrásból éri el a kernel azokat a komponenseket, amiket nem tartalmaz



STRUKTÚRÁK: RÉTEGELT RENDSZEREK

- A rendszert rétegekből álló hierarchia jellemzi, minden réteget az alatta lévőre építenek fel
- Első: THE (E. W. Dijkstra)
- Rétegek:
 - 5: Gépkezelő
 - 4: Felhasználói programok
 - 3: Bement/kimenet kezelése
 - 2: Gépkezelő processzus kommunikáció
 - 1: Memória- és dobkezelés
 - 0: Processzor-hozzárendelés és multiprogramozás

STRUKTÚRÁK: VIRTUÁLIS GÉPEK

- Egy adott hardver/szoftver architektúra emulálását jelenti
- Első: VM/370 (1979): különválasztották a multiprogramozást és a hardver eléréséhez használt kiterjesztett gépet
- Példák:
 - MS-DOS rendszerek futtatása Windowson emulált környezetben
 - VMWare, VirtualBox
 - Xen, LXD, VMWare, KVM
 - Java Virtual Machine, Python runtime
 - Linux chroot
 - Solaris containers
 - Android runtime

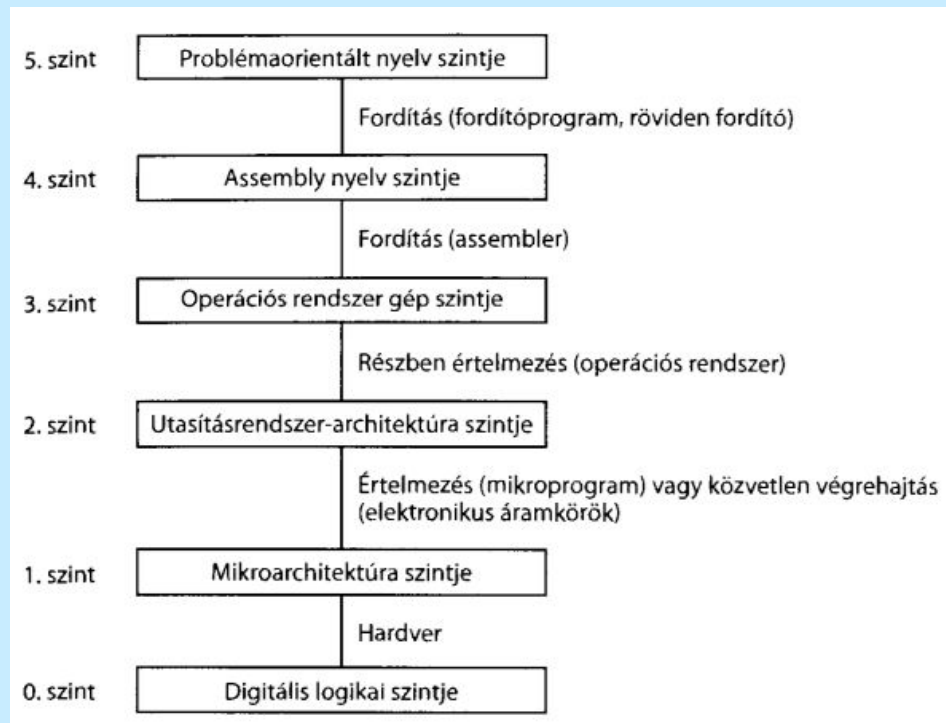
4. ÓRA
ARCHITEKTÚRA

SZÁMÍTÓGÉP ARCHITEKTÚRÁK

- Ajánlott irodalom: Tanenbaum – Számítógép architektúrák
- Digitális számítógép: problémák megoldása utasítások révén
- Program: utasítások sorozata
- Az elektronikus áramkörök utasítások egy szűk halmazát képesek felismerni, programjainkat konvertálni kell
- Fő utasítások:
 - Adj össze két számot
 - Ellenőrizd, hogy a szám nulla-e?
 - Egy számot másolj a memória egyik címéről egy másikra
- A gépi nyelvek kényelmetlenek az ember számára, ezért strukturálták a működést absztrakciók sorozatára, innen ered a strukturált számítógép-felépítés

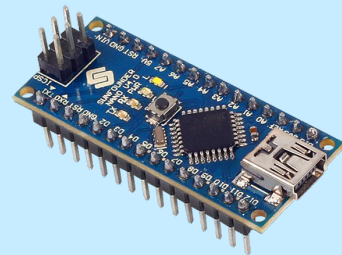
TÖBBSZINTŰ SZÁMÍTÓGÉPEK

- 5: Magas szintű nyelvek
- 4: Szimbólikus nyelv
- 3: Bővített utasítások
- 2: ISA
- 1: ALU
- 0: kapuk
- -1: elektronika

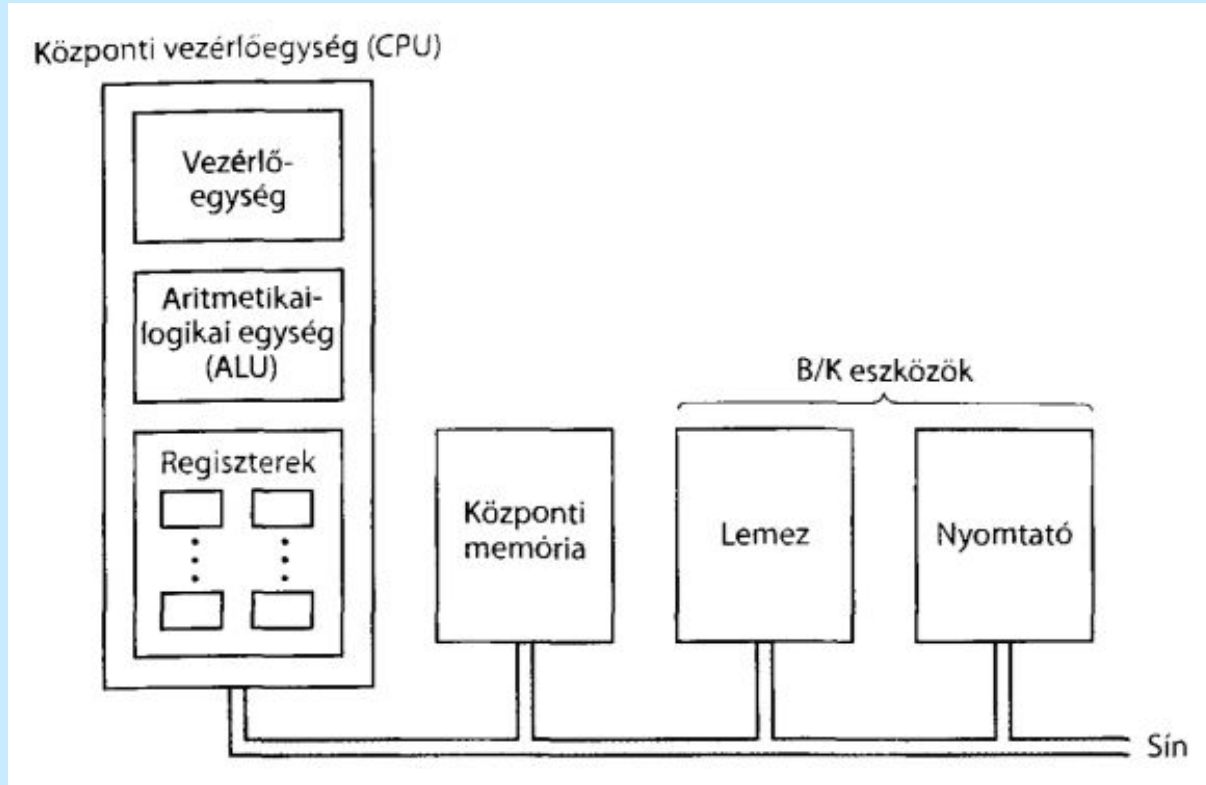


SZÁMÍTÓGÉPEK TERMÉKSKÁLÁJA

- Eldobható számítógép
- Mikrovezérlő
- Személyi számítógép
- Szerver
- Elosztott rendszer (klaszterek)
- Nagyszámítógép



PROCESSZOR: FELÉPÍTÉS

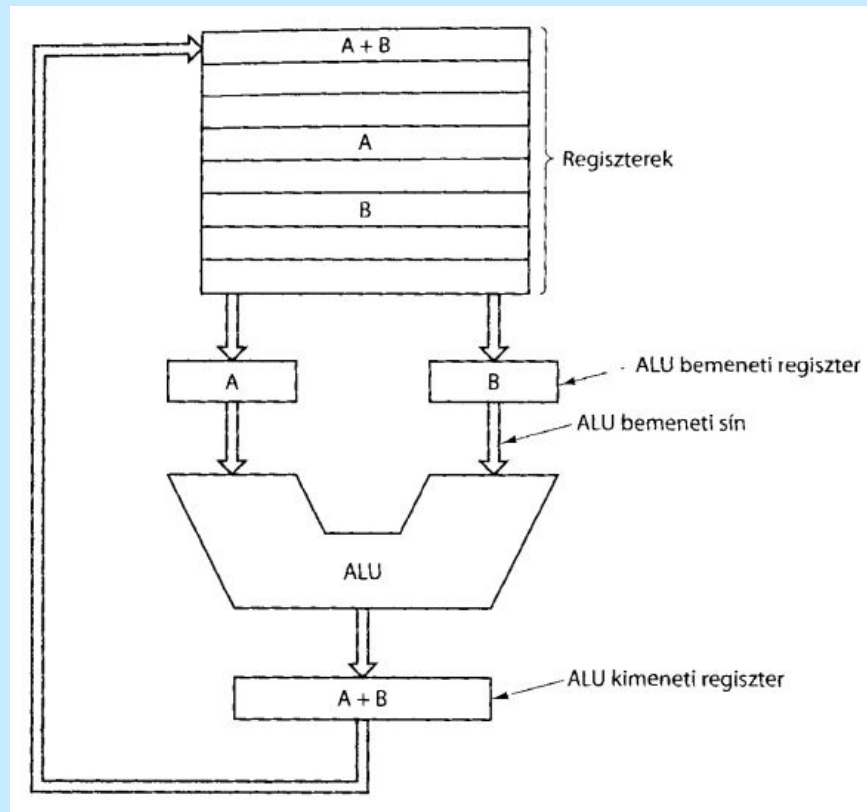


PROCESSZOR: FELÉPÍTÉS

- Sín (Bus): összeköti a részegységeket, adatok és vezérlőjelek továbbítására szolgál
- CPU (Central Processing Unit): feladata a központi memóriában tárolt programok végrehajtása
- Regiszterek: kis méretű, gyors memória
- PC (Program Counter): az egyik regiszter, a következő program memória beli címét tartalmazza
- IR (Instruction Register): az utasításregiszter, a végrehajtás alatt lévő utasítást tartalmazza

PROCESSZOR: FELÉPÍTÉS

- Adatút (Data path):
 - regiszterek
 - ALU (Arithmetical Logical Unit)
 - Sínek
- Az ALU bementi a regiszterekből olvas be és az eredményt a kimeneti regiszterekbe írja
- Fontosabb ALU műveletek
 - összeadás
 - kivonás
 - összehasonlítás nullával



PROCESSZOR: UTASÍTÁS VÉGREHAJTÁS

- A végrehajtás lépései (betöltő-dekódoló-végrehajtó ciklus):
 - A soron következő utasítás beolvasása a memóriából az utasításregiszterbe
 - Az utasításslámláló beállítása a következő utasítás címére
 - A beolvasott utasítás típusának meghatározása.
 - Ha az utasítás memóriabeli szót használ, a szó helyének megállapítása.
 - Ha szükséges, a szó beolvasása a CPU egy regiszterébe.
 - Az utasítás végrehajtása.
 - Vissza az 1. pontra, a következő utasítás végrehajtásának megkezdése.

PROCESSZOR: CISC ÉS RISC

- CISC: Complex instruction set computing
- RISC: Reduced instruction set computing
- #1: Program végrehajtására képes processzoráramkör építése
- #2: Programot értelmezni tudó interpreter írása
- Azt állították, hogy a számítógépek tervezésének legjobb módja, ha kevés egyszerű utasításunk van, amelyek adatútjának egyszeri bejárásával végrehajthatók.
- Ha egy CISC-utasítás helyettesítéséhez 4-5 RISC-utasítás kell, még mindig a RISC a gyorsabb, mert a RISC utasítások 10-szer gyorsabbak egy CISC-nél (mivel nem interpretáltak)

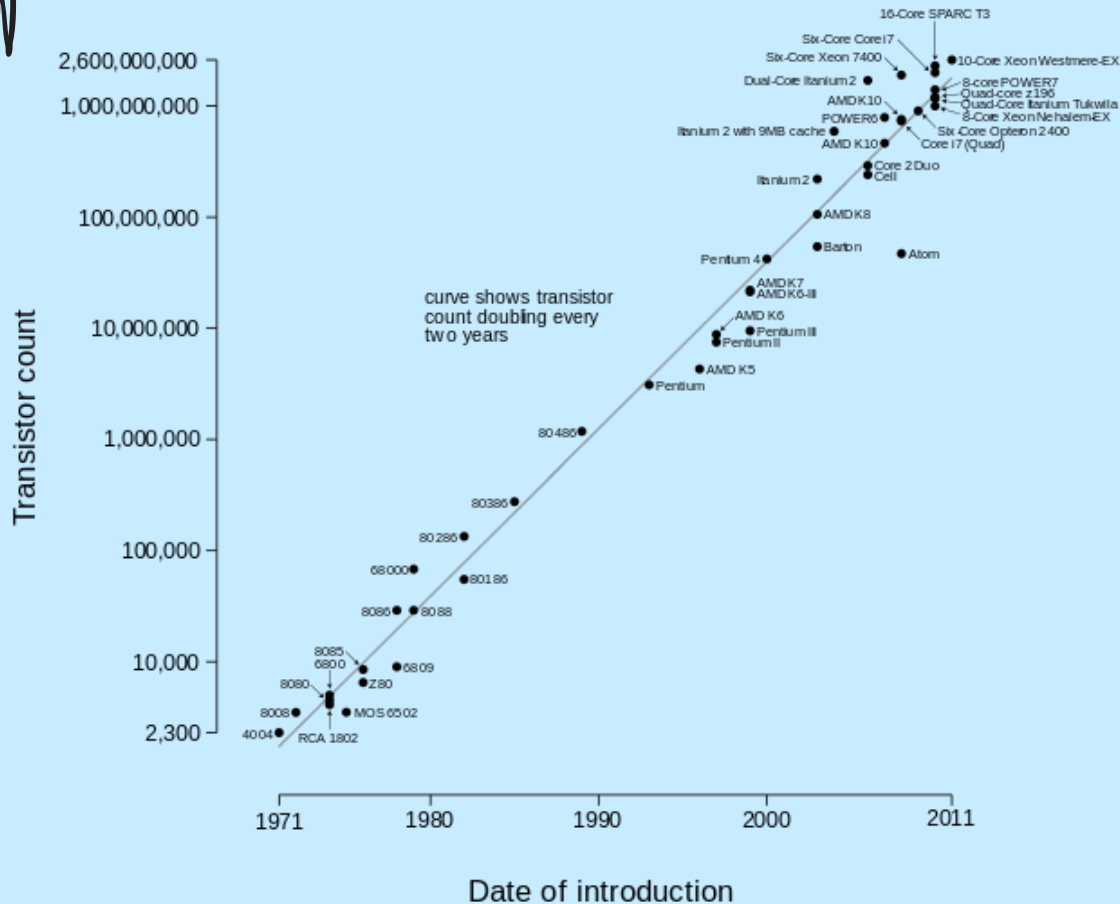
PROCESSZOR: CISC ÉS RISC

- Ugyan logikusak az érvek a RISC mellett, mégsem szorította ki a piacról a CISC-et az alábbiak miatt:
 - Visszafelé kompatibilitás
 - Dollármilliárdok, amiket a CISC rendszerek fejlesztésére költöttek
 - 486-tól kezdődően az Intel egy RISC magot is épít a CPU-kba az egyszerű utasítások számára, a bonyolultabbakat CISC módon hajtja végre
- A hibrid megközelítés nem olyan gyors, mint a tisztán RISC módszer, de versenyképes, és megmarad a kompatibilitás

PROCESSOR: MOORE'S LAW

- Gordon E. Moore az Intel egyik alapítója
- “Az integrált áramkörök összetettsége 18 hónaponként megduplázódik”

Microprocessor Transistor Counts 1971-2011 & Moore's Law



MEMÓRIA: BITEK

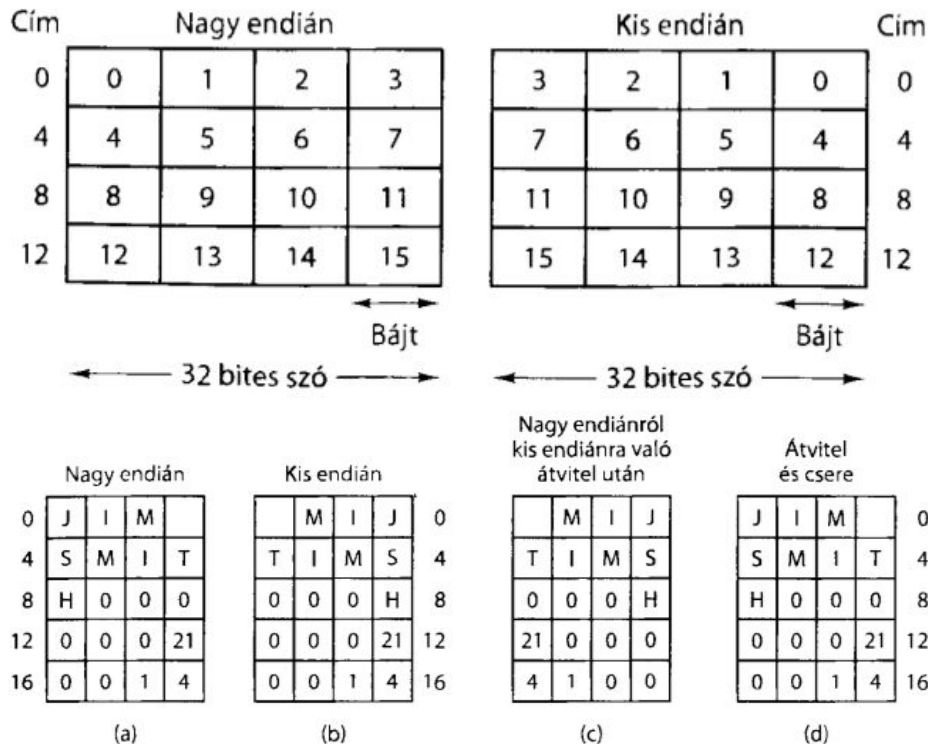
- A memória alapegysége a bit (0 vagy 1)
- Minél több feszültség szintet kell megkülönböztetnünk, annál bonyoltultabb feladat, ezért “hatékony” 2-es számrendszert használni
- BCD (Binary coded decimal): 4-biten tárol egy 10-es számrendszer beli számot
- 1944
 - decimális: 0001 1001 0100 0100
 - bináris: 0000 0111 1001 1000

MEMÓRIA: CÍMZÉS

- A memória egyforma, k méretű cellákba van rendezve
- 0-tól n -ig címezhetőek a cellák - $n \cdot 2^k$ bit
- A 8-bites cella-méret vált általánossá
- Egy 64 bites rendszernek 64 bitesek a regiszterei, így 64 biten tudja megcímezni a memóriát is (maximális memória méret: $2^{64} \cdot 1 \text{ Byte}$)

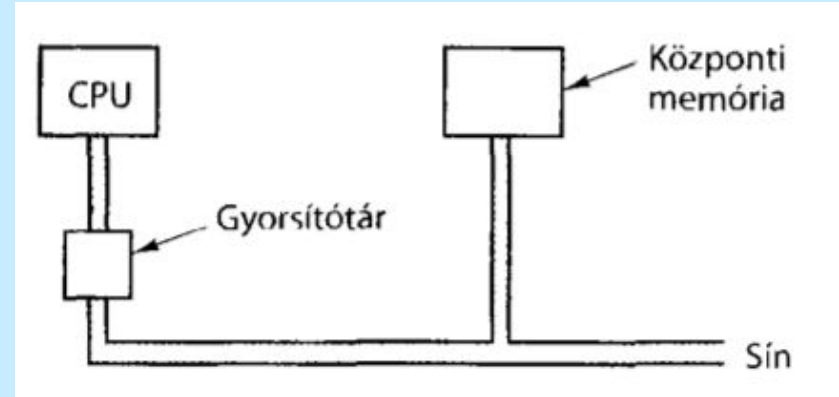
MEMÓRIA: BÁJTTSORREND

- Mindkét reprezentáció teljesen jó, és önmagában konzisztens.
- A problémák akkor kezdődnek, amikor az egyik gép adatokat akar küldeni a másiknak hálózaton keresztül.



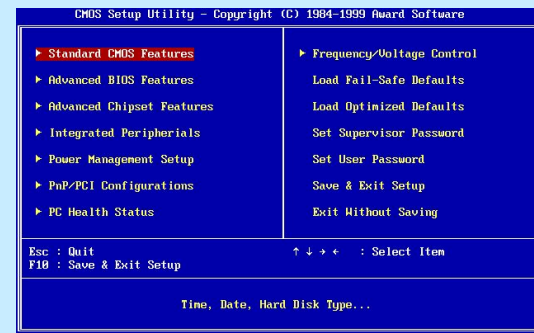
MEMÓRIA: GYORSÍTÓTÁR

- CPU gyártók célja sebesség növelése, míg a memória gyártók a kapacitást növelik
- Gyors memóriát a CPU lapkán kell elhelyezni, ez azonban drága lenne, a sínen kapcsolt memória jóval olcsóbb
- Hibrid megoldás: kevés, gyors memória (cache) a CPU lapkán és sok, de lassú memória a sínen keresztül elérve
- Lokalitási elv: soron következő utasítások gyakran használják a korábbi memóriaterület szomszédságát
- A cache-be mindig egy területet másol, így esélyes, hogy egy következő utasítást csupán cacheből ki lehet szolgálni



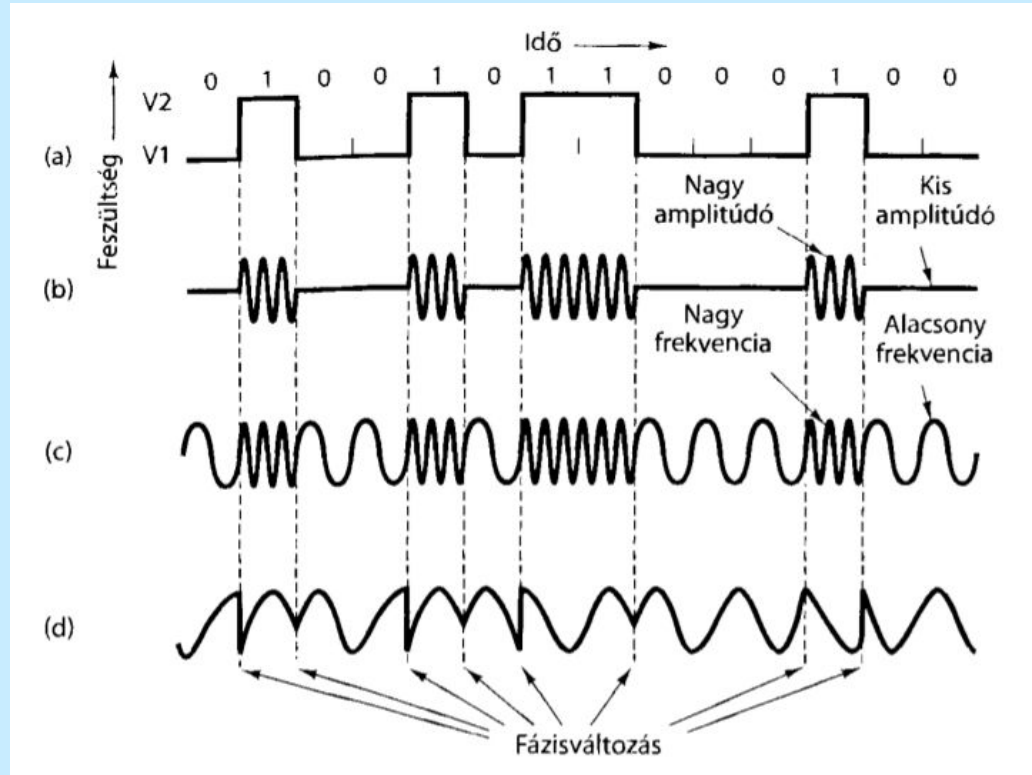
BASIC INPUT OUTPUT SYSTEM (BIOS)

- Az alaplapon túl a bővítőkártyák is saját BIOS-al rendelkezhetnek
- Korábban hasonló funkciót töltöttek be a firmware-ek, 1975-ben vezették be a BIOS-t, mint (könnyen) módosítható változatot
- Hardver és a szoftver közötti kapcsolat
- Feladatai:
 - Hardver ellenőrzése
 - Hardvervezérlők betöltése
 - Operációs rendszer betöltése
 - Interfész biztosítása az operációs rendszer számára a harverelemek eléréséhez



FIZIKAI JELÁTVITELI MÓDSZEREK

- a) Kétszintű jel
- b) Amplitúdómoduláció
- c) Frekvenciamoduláció
- d) Fázismoduláció



5. ÓRA
ADATTÁROLÁS

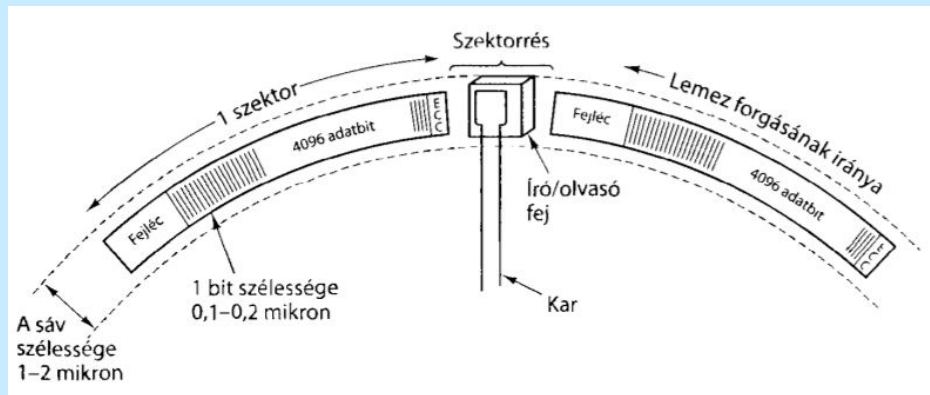
HÁTTÉRMEMÓRIA: HIEARCHIA

- Egy bájt tárolásának költsége fentről lefelé csökken
- A sebesség is fentről lefelé csökken
- Jellemző elérési idők
 - Regiszterek: 1-5 ns
 - Memória: 10-50 ns
 - SSD: 0.1-0.3 ms
 - Mágneslemez: 3-12 ms

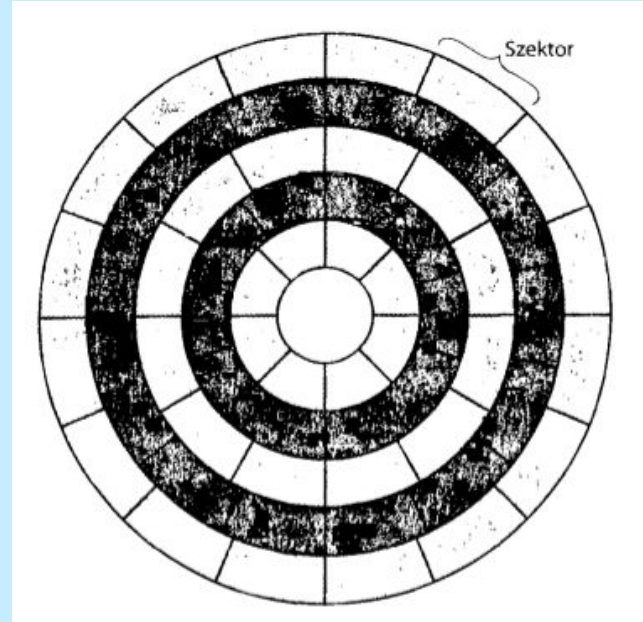
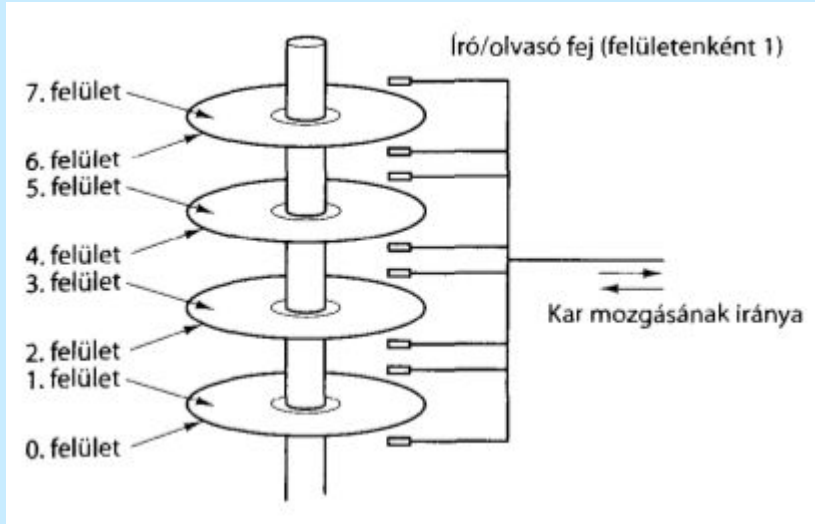


HÁTTÉRMEMÓRIA: MEREVLEMEZ

- Alumínium lemez, mágnesezhető bevonattal ellátva
- Indukciós tekercset tartalmazó fej a lemez felszíne felett légpárnán lebeg
- Egy teljes körülfordulás alatt felírt bitsorozat a sáv
- A sávok szektorokra vannak bontva



HÁTTÉRMEMÓRIA: MEREVLEMEZ



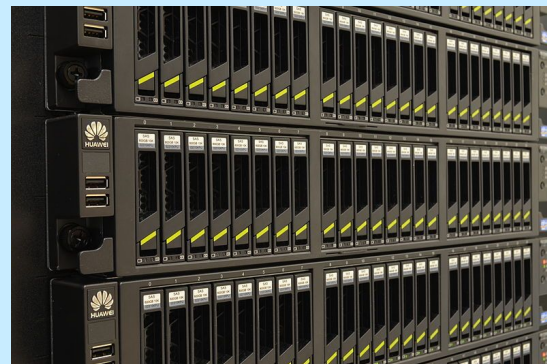
HÁTTÉRMEMÓRIA: SOLID STATE DRIVE



- A HDD-k alternatívája
- Flash-memóriát alkalmaznak bennük, amik azután megtartják az adatot, miután az áramforrás megszűnik
- Készítenek hibrid meghajtókat is, amikor a HDD-be teszik az flash-memóriát (gyorsítótár, vagy külön használható)
- A szabad blokkok száma befolyásolja a működés sebességét: minél több a szabad blokk, annál gyorsabb a meghajtó
- TRIM/UNMAP parancs: az OS jelzi a meghajtó felé, hogy mely blokkok szabadíthatóak fel későbbi írás céljára
- Samsung PM1633a: 15.36 TB

HÁTTÉRMEMÓRIA: CSATOLÓFELÜLETEK

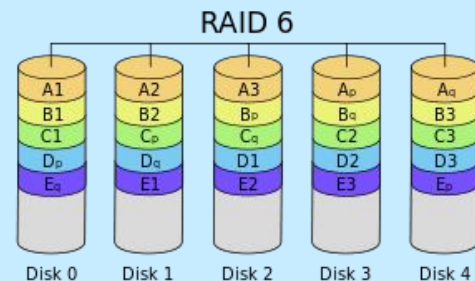
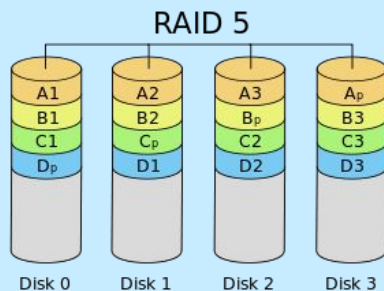
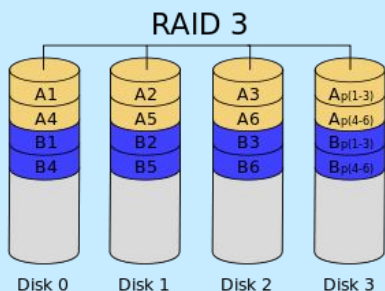
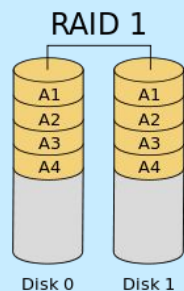
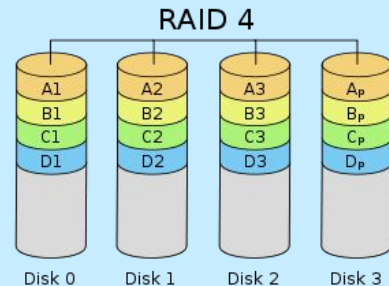
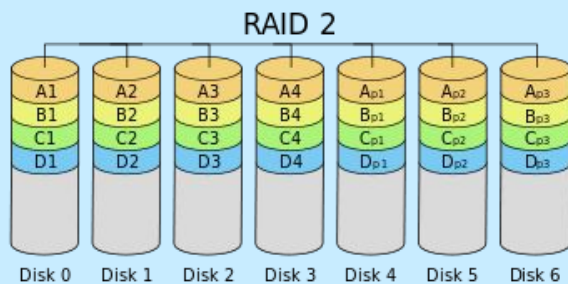
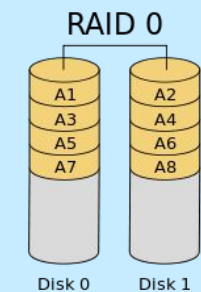
- IDE/PATA: Integrated Drive Electronics (1986)
- SCSI: Small Computer System Interface (1981)
- SATA: Serial AT Attachment (2003)
- SAS: Serial attached SCSI(2004)
- iSCSI: Internet Small Computer Systems Interface (2000)



HÁTTÉRMEMÓRIA: RAID

- Redundant Array of Independent Disks
- Levels
 - RAID 0: összefűzés vagy csíkozás
 - RAID 1: tükrözés
 - RAID 2: csíkozás + hibajavító kód tároló lemezek
 - RAID 3: 3-hoz hasonló, de csak paritásinfó van tárolva
 - RAID 4: 4-hez hasonló, csak nagy méretű csíkokkal
 - RAID 5: paritásinfó az összes meghajtón eloszlatva
 - RAID 6: 5 bővítés, paritás soronként és oszloponként
 - RAID 1+0: 4 lemez kell, először tükrözés, az után csíkozás
 - RAID 0+1: 4 lemez kell, először csíkozás, az után tükrözés

HÁTTÉRMEMÓRIA: RAID



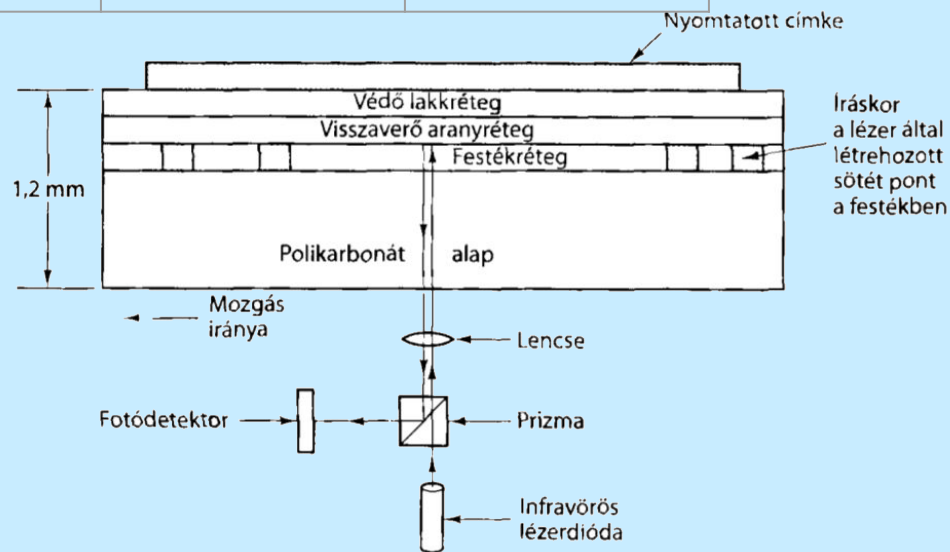
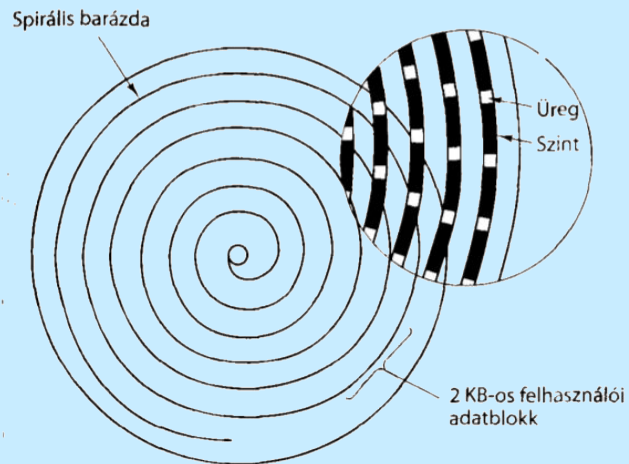
forrás: Wikipedia

HÁTTÉRMEMÓRIA: OPTIKAI MEGHAJTÓK

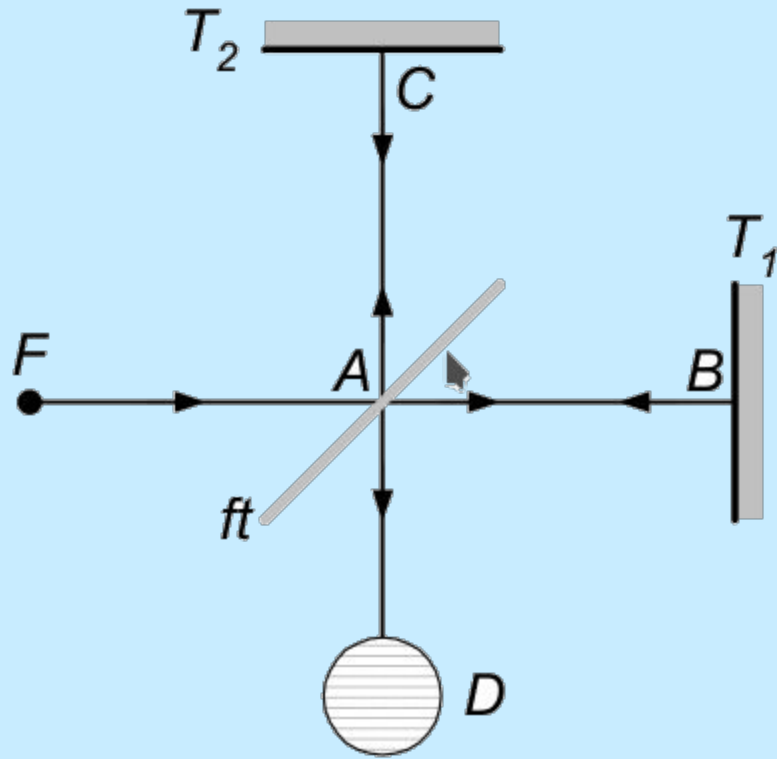
- A műanyag lemezen tárolt adattartalmat lézerfej segítségével írja / olvassa a meghajtó
- Egy gyári CD lemez készítése: nagy energiájú IR lézerrel 0,8 mikron átmérőjű lyukakat ésgetnek egy bevonattal ellátott üveg mesterlemezbe, amiről negatív öntőforma készül, végül az öntőformába olvadt polikarbonátot töltönek
- A CD visszaolvasakor egy lézerdióda 0.78 mikron hullámhosszú IR lézerrel megvilágítja a lyukakat a lemezen
- A lyukak mélysége a lézer hullámhossz negyede, ezért fáziseltolódás van a környezetről és lyukból visszavert

HÁTTÉRMEMÓRIA: OPTIKAI MEGHAJTÓK

CD	1982	700 MB	780 nm	1200 kbit/s
DVD	1995	17.08 GB	650 nm	10.5 Mbit/s
Blu-ray	2007	128 GB	405 nm	576 Mbit/s



MICHELSON-INTERFEROMÉTER



Az F fényforrásból kiinduló fény az A pontban eléri a fénysugár irányával 45° szöget bezáró ft félig áteresztő tükört.

A tükör a fényintenzitás egy részét átengedi, és ez a rész a T_1 tükörről visszaverődve visszaér az A pontba, majd egy része az ft tükörön visszaverődve a detektáló eszközbe (D) jut.

A fényintenzitás másik részét az ft tükör az eredeti fénysugárra merőleges irányban visszaveri, így az a T_2 tükörrre kerül.

Onnan visszaverődik, és egy része az ft tükörön áthaladva a detektáló eszközbe jut.

Hullámkád:

<http://www.falstad.com/ripple/>

HÁTTÉRMEMÓRIA: SZALAGOS MEGHAJTÓK

- Az adatok rögzítése szekvenciálisan mágnesszalagra történik
- 1951: Remington Rand – UNISERVO (224 kB)
- 2014: IBM – TS1150 (10 TB, 360 MB/s)
- “shoe-shining”: a mai gyors meghajtók puffer kifogyás esetén nem képesek azonnal megállni, vissza kell állniuk egy korábbi állapotba és újratekdeni az írást – ha ez gyakran megeshet az “fényesíti” a szalagot

HÁTTÉRMEMÓRIA: SZALAGOS MEGHAJTÓK



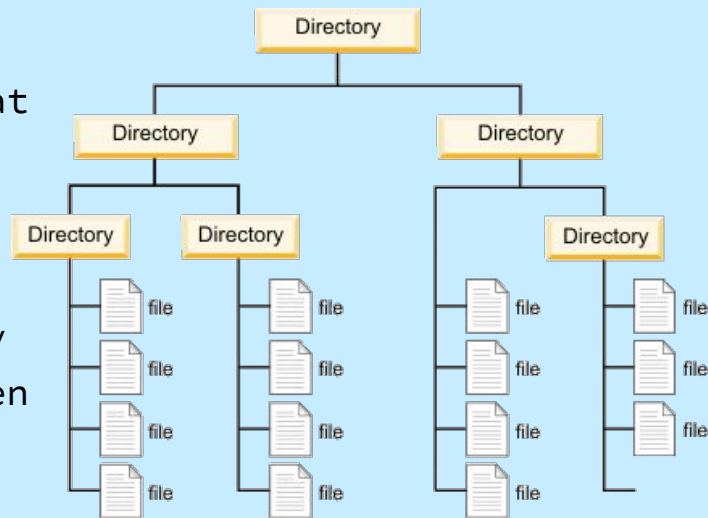
Remington Rand – UNISERVO



IBM – TS1150 & cardigate

FÁJLRENDSZEREK

- Problémák amik a fájlrendszerekhez vezettek:
 - A memória kicsi ahhoz, hogy minden adat elférjen benne
 - A memória illékony, a processzus végeztével nem érhető el az adat
 - Biztosítani kell, hogy egy adathoz egy időben több processzus is hozzáférhessen
- Az alapegység a fájl
- A fájlok a legtöbb esetben könyvtárakhoz vannak rendelve, melyek fa-struktúra szerint rendezettek



FÁJLRENDSZEREK: FÁJLOK

- Absztrakciós mechanizmus, lehetővé teszi az információ lemezen tárolását és visszaolvasását
- Fájlnev: karakterek sorozata, egyes fájlrendszerek az utolsó pontot követő részt kiterjesztésként értelmezik, mely utal a fájl típusára
- Fájl típusok:
 - Futtatható (bináris) fájlok
 - Adatfájlok (ASCII vagy bináris)
 - Speciális fájlok (operációsrendszer-specifikus)

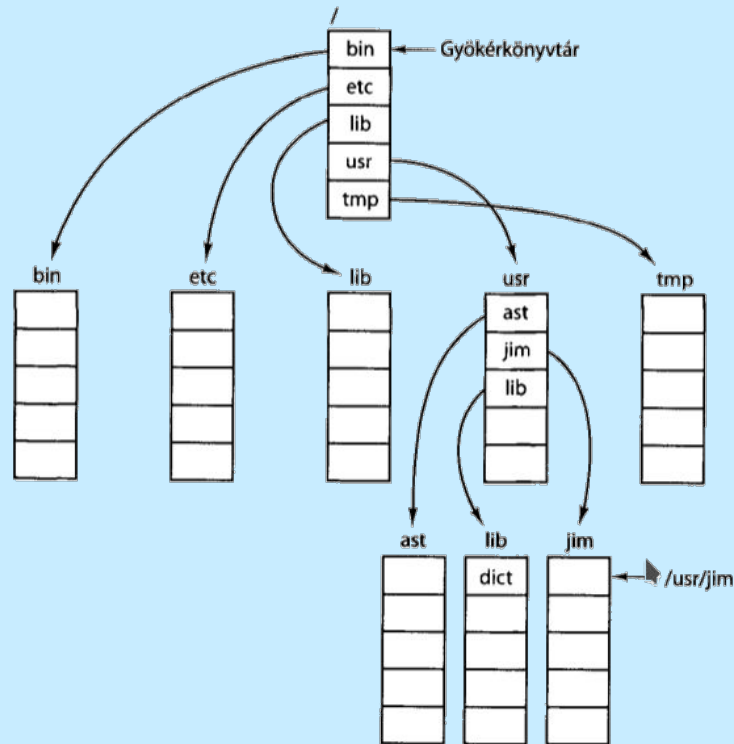
FÁJLRENDSZEREK: FÁJLOK

- Fájlattribútumok (táblázat)
- Fájlműveletek:
 - Létesítés
 - Törlés
 - Megnyitás (írásra, olvasásra)
 - Lezárás
 - Olvasás
 - Írás
 - Hozzáatldás (append)
 - Pozícionálás (seek)
 - Attribútum írás
 - Attribútum olvasás
 - Átnevezés
 - Zárolás (lock)

Mező	Értelmezés
Védelem	Ki érheti el a fájlt és milyen módon
Jelszó	Jelszó, amelyet az eléréshez meg kell adni
Létrehozó	A fájl létrehozójának azonosítója
Tulajdonos	Az aktuális tulajdonos azonosítója
Csak olvasható jelző	0, ha írás és olvasás megengedett, 1, ha csak olvasható
Rejtettségi jelző	0 a normál eset, 1, ha listázásban nem megjelenítendő
Rendszerjelző	0 normál fájl, 1 rendszerfájl esetén
Archív jelző	0, ha archiválva volt, 1, ha archiválásra kijelölt
ASCII/bináris jelző	0, ha ASCII, 1, ha bináris a fájl
Közvetlen elérés jelző	0, ha csak szekvenciális, 1, ha közvetlen elérésű a fájl
Ideiglenességjelző	0, ha normál fájl, 1, ha törölni kell a processzus befejeződésekor
Zároltságjelző	0, ha nem zárolt, 1, ha zárolt a fájl
Rekord hossza	A bájtok száma egy rekordban
Kulcs pozíciója	A kulcs pozíciója a rekordban
Kulcs hossza	A kulcsmező hossza bájtokban
Létesítési idő	A fájl létrehozásának dátuma és időpontja
Utolsó hozzáférés ideje	Az utolsó hozzáférés dátuma és időpontja
Utolsó módosítás ideje	Az utolsó módosítás dátuma és időpontja
Aktuális méret	A bájtok száma a fájlban
Maximális méret	A lehetséges maximális fájl méret bájtban

FÁJLRENDSZEREK: KÖNYVTÁRAK

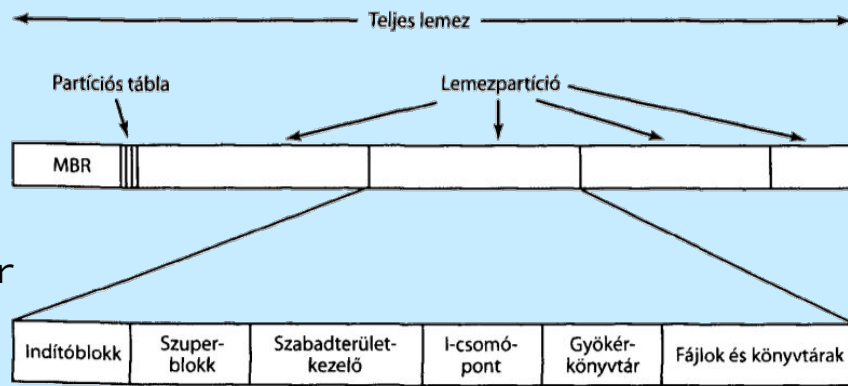
- Könyvtárszerkezet:
 - Egyszerű
 - Hierarchikus
- Útvonal megadása:
 - Abszolút
 - Relatív
- Könyvtári műveletek:
 - Létesít, töröl
 - Megnyit, lezár
 - Olvas
 - Átnevez
 - Kapcsol, lekapcsol (link, unlink)



FÁJLRENDSZEREK: SZERKEZET

- MBR partíciós tábla

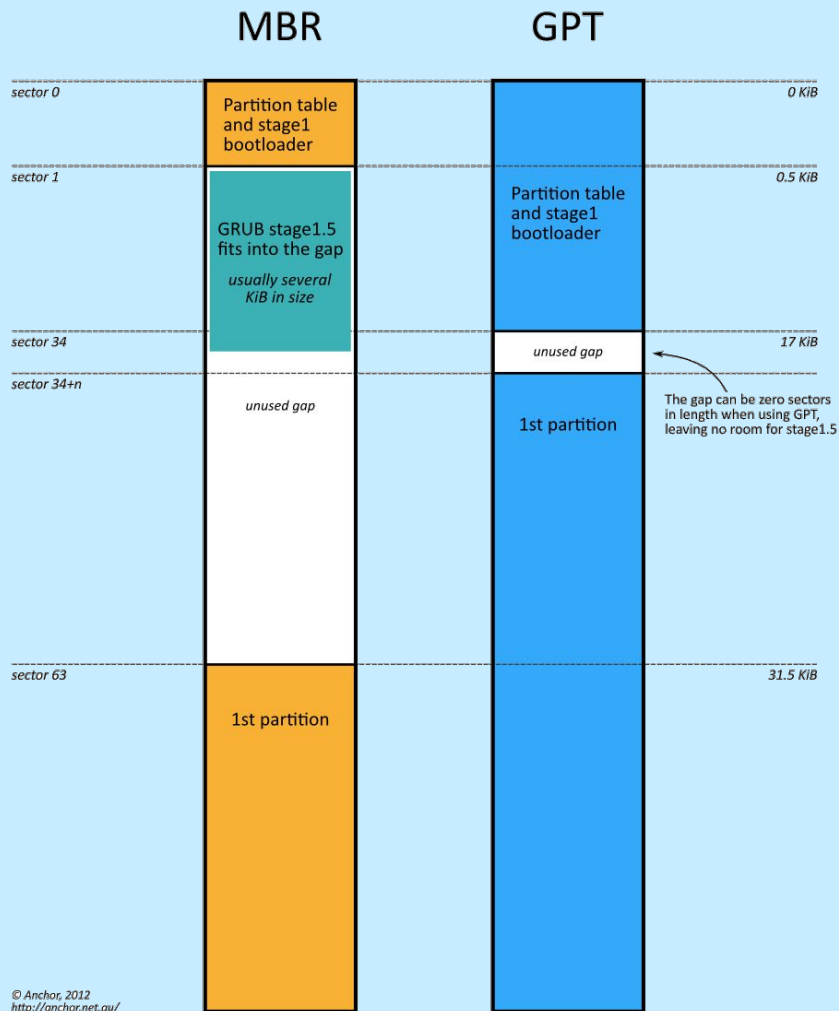
- A lemez partíciókra van osztva
- A lemez 0. szektorra az MBR (Master Boot Record)
- Az MBR-ben lévő kódot induláskor a BIOS tölti be
- Az MBR után következik a partíciós tábla
- A tábla tartalmazza, hogy a partíciók a lemezen hol helyezkednek el
- Minden partíció független fájlrendszert tartalmaz
- PC kompatibilis rendszerekben 4 elsődleges partíció lehet
- Egy elsődleges partíciót definiálhatunk kiterjesztett partícióként, ami logikai partíciók láncolt listáját tartalmazhatja



FÁJLRENDSZEREK: SZERKEZET

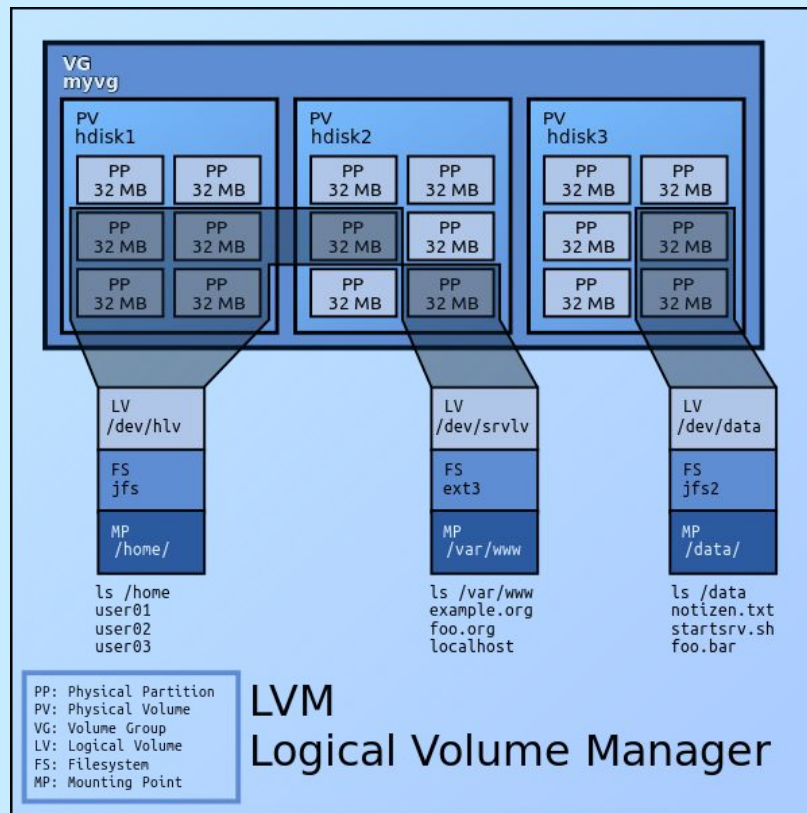
- GPT prtíciós tábla

- GUID-t (Globally Unique Identifier) használ a lemezek és a partíciók azonosítására
- Korlátlan számú partíció
- 64 bit LBA -> max. 2 ZiB (Zebibyte)
- kilo < mega < giga < tera < peta < exa < zetta < yotta
- Backup a lemez végén
- CRC32 ellenőrzőösszeg használata az adatsérülés detektálásához



FÁJLRENDSZEREK: LOGICAL VOLUME MANAGEMENT (LVM)

- Linux specifikus logikai kötetkezelés
- 1998-ban írta Heinz Maueishagen a HP-UX kötetkezelője alapján
- A fizikai partíciók fölött lévő újabb absztrakciós szint
- Szintjei:
 - Physical volumes (PV)
 - Volume groups (VG)
 - Logical volumes (LV)
- Leegyszerűsíti a partíciók kezelését



FÁJLRENDSZEREK: FONTOSABB PC FÁJLRENDSZEREK

- ext2: Natív Linux FS, felfelé kompatibilis
- ext3, ext4: az ext2 naplózó verziói
- reiserfs: robosztus FS, jól kezeli az adatkorrupciót
- jfs: naplózó FS, IBM fejlesztés
- xfs: magas teljesítmény, nagy fájlok esetében is
- zfs: FS és LVM egyben, a SUN fejlesztése
- nfs: hálózati fájlrendszer
- FAT, FAT32, exFAT: Microsoft MS-DOS FS és újabb verziói
- NTFS: Microsoft legfejlettebb FS-e