

Bevezetés a számítástechnikába nagy ZH összefoglaló

Ekart Csaba - 2016



PÁZMÁNY PÉTER
KATOLIKUS EGYETEM

Tartalom

I. Számrendszerek, számábrázolás, karakterkódolás	4
A számrendszer	4
Alaki és helyiértékek, egész és tört számok.....	4
Átváltás számrendszerek között.....	4
Az adatmennyiség mértékegységei és gépi számábrázolás.....	4
MSB/LSB	5
Túlcsordulás, lebegőpontos ábrázolás.....	5
Karakterek és karakterkészletek.....	5
II. Operációs rendszerek.....	7
Az operációs rendszer fogalma	7
Az operációs rendszer, mint erőforrás-kezelő.....	7
Az operációs rendszerek történelme.....	7
Processzus	8
Fájlok	9
Parancsértelmező.....	9
Rendszerhívások.....	9
Virtuális memória	10
Struktúrák, kernel típusok	10
III. Architektúra	12
Számítógép architektúrák.....	12
Többszintű számítógépek.....	12
Számítógépek termékiskálája	12
Processzor felépítése és működése.....	13
Utasítás végrehajtás	14
A CISC és a RISC	14
Moore törvénye (<i>Moore's Law</i>).....	15
Memória: Bitek és címzés.....	15
Bájtrend.....	16
Gyorsítótár	16
Basic Input Output System (BIOS)	16
Fizikai jelátviteli módszerek.....	17

IV. Adattárolás	18
Háttérmemória, hierarchia.....	18
A merevlemez.....	18
Solid State Drive (SSD).....	18
Csatolófelületek.....	19
Háttérmemória: Raid.....	19
Optikai meghajtók	20
Michelson-interferométer.....	20
Szalagos meghajtók	21
Fájlrendszerek és fájlok	21
Könyvtárak.....	22
Szerkezet	22
Logical Volume management (LVM)	23
Fontosabb PC fájlrendszerek	24

I. 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, ezen kívül még megkülönböztetjük a sorrendiségen alapuló számrendszereket (pl.: római számok).
- A számrendszerek **alapja** [base, radix] meghatározza az egyes pozíciókra írható számok maximumát. A **számjegyek** [digit] a számok írására használt karakterek.
- Az informatikában leginkább a 2-es a 8-as és a 16-os számrendszert használjuk, a hétköznapiakból fakadóan természetesen a hagyományos 10-es számrendszert is.

Alaki és helyiértékek, egész és tört számok

- Egy számjegy értékét megadhatjuk alaki értékéből és helyiértékéből az alábbi módon: **számjegy értéke = alaki érték · helyiérték**. Az **alaki érték** a számjegyhez tartozó értéket jelenti, míg a **helyi érték** a számrendszer alapjának pozíció szerinti hatványa.
- Egész és tört számok felírása:**
 - 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 számrendszerek között

- A kettes számrendszerbe történő átváltás a képen lévőhöz hasonlóan, történik, más számrendszerekbe való átváltás is hasonlóan végezhető el. (ábrán: $41_{(10)} = 101001_2$)
- Nem egész számok nem mindig írhatók fel véges számjeggyel (pl.: az $1/3_{10} = 0.3333\dots_{10} = 0.1_3$)

41	: 2	=	20	marad 1
20	: 2	=	10	marad 0
10	: 2	=	5	marad 0
5	: 2	=	2	marad 1
2	: 2	=	1	marad 0
1	: 2	=	0	marad 1

Az adatmennyiség mértékegységei és gépi számábrázolás

- Alapegysége a **bit**, mely 0-t vagy 1-est, igaz vagy hamis értéket tárolhat.
- 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

- Nem negatív egész számok:** megegyezik a bináris számok leírásával, tehát kettes számrendszerben tároljuk, összeadás műveletét értelmezzük a nemnegatív egész bináris számokon, az összeadás módszere triviális.

- **Negatív számok előjelbites ábrázolása:**
 - A legnagyobb helyiértéken a szám előjelét tároljuk.
 - A maradék helyiértéken a számot
 - Pl. 8 bites változó méret esetén $-32_{(10)} = 1010\ 0000_{(2)}$
- **2-es komplement ábrázolás**
 - A számnak veszem az egyes komplementét és hozzáadok még egyet.
 - A 2-es komplement nagy előnye, hogy nincs szükség általa kivonás műveletre. Elég, ha a kivonandónak hozzáadjuk a kettes komplementét az eredeti számhoz.

	-2^5	2^4	2^3	2^2	2^1	2^0	
A	0	1	0	0	1	0	(+18)
1-es kompl.	1	0	1	1	0	1	
+1	0	0	0	0	0	1	
2-es kompl.	1	0	1	1	1	0	(-18)

MSB/LSB

- **MSB:** Most Significant Bit
- **LSB:** Least Significant Bit
- Architektúránként különböző
- 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.**
- **Endianness¹ (bájtrend):** Az endianness jelzi azt a tulajdonságot, ami bizonyos adatok, többnyire kisebb egységek egymást követő sorozata – tárolási és/vagy hálózaton való továbbítási sorrendjét jellemzi. **big-endian / little-endian** (-> Swift: Guliver utazásai)

Túlcsordulás, lebegőpontos ábrázolás

- Egész számok ábrázolása esetén meghatározott számú biten korlátozott az ábrázolható számok nagysága, ebből következően lehetséges **alul- vagy túlcsordulás.**
- A megvalósítástól függően eredményezhet:
 - **levágást**, azaz a nem ábrázolható részt levágjuk és nem használjuk fel, illetve
 - **szaturációt**, azaz a legnagyobb vagy legkisebb ábrázolható értéket tároljuk.
- **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. Pl. : $380_{(10)} = 3.8 \cdot 10^2$

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

¹ <https://hu.wikipedia.org/wiki/B%C3%A1jtrend>

- UTF-8
 - Változó 1-6 Byte hosszú tárolás
 - ASCII kompatibilis és önszinkronizáló (nem kell a string elejéről kezdeni az olvasást, hogy elkülönüljenek a karakterek)

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0 0	000		NULL	32 20	040		 	Space	64 40	100		@	@	96 60	140		`	`
1 1	001		SoH	33 21	041		!	!	65 41	101		A	A	97 61	141		a	a
2 2	002		SoTxt	34 22	042		"	"	66 42	102		B	B	98 62	142		b	b
3 3	003		EoTxt	35 23	043		#	#	67 43	103		C	C	99 63	143		c	c
4 4	004		EoT	36 24	044		$	\$	68 44	104		D	D	100 64	144		d	d
5 5	005		Enq	37 25	045		%	%	69 45	105		E	E	101 65	145		e	e
6 6	006		Ack	38 26	046		&	&	70 46	106		F	F	102 66	146		f	f
7 7	007		Bell	39 27	047		'	'	71 47	107		G	G	103 67	147		g	g
8 8	010		Bsp	40 28	050		((72 48	110		H	H	104 68	150		h	h
9 9	011		HTab	41 29	051))	73 49	111		I	I	105 69	151		i	i
10 A	012		LFeed	42 2A	052		*	*	74 4A	112		J	J	106 6A	152		j	j
11 B	013		VTab	43 2B	053		+	+	75 4B	113		K	K	107 6B	153		k	k
12 C	014		FFeed	44 2C	054		,	,	76 4C	114		L	L	108 6C	154		l	l
13 D	015		CR	45 2D	055		-	-	77 4D	115		M	M	109 6D	155		m	m
14 E	016		SOOut	46 2E	056		.	.	78 4E	116		N	N	110 6E	156		n	n
15 F	017		SIn	47 2F	057		/	/	79 4F	117		O	O	111 6F	157		o	o
16 10	020		DLE	48 30	060		0	0	80 50	120		P	P	112 70	160		p	p
17 11	021		DC1	49 31	061		1	1	81 51	121		Q	Q	113 71	161		q	q
18 12	022		DC2	50 32	062		2	2	82 52	122		R	R	114 72	162		r	r
19 13	023		DC3	51 33	063		3	3	83 53	123		S	S	115 73	163		s	s
20 14	024		DC4	52 34	064		4	4	84 54	124		T	T	116 74	164		t	t
21 15	025		NAck	53 35	065		5	5	85 55	125		U	U	117 75	165		u	u
22 16	026		Syn	54 36	066		6	6	86 56	126		V	V	118 76	166		v	v
23 17	027		EoTB	55 37	067		7	7	87 57	127		W	W	119 77	167		w	w
24 18	030		Can	56 38	070		8	8	88 58	130		X	X	120 78	170		x	x
25 19	031		EoM	57 39	071		9	9	89 59	131		Y	Y	121 79	171		y	y
26 1A	032		Sub	58 3A	072		:	:	90 5A	132		Z	Z	122 7A	172		z	z
27 1B	033		Esc	59 3B	073		;	;	91 5B	133		[[123 7B	173		{	{
28 1C	034		FSep	60 3C	074		<	<	92 5C	134		\	\	124 7C	174		|	
29 1D	035		GSep	61 3D	075		=	=	93 5D	135]]	125 7D	175		}	}
30 1E	036		RSep	62 3E	076		>	>	94 5E	136		^	^	126 7E	176		~	~
31 1F	037		USep	63 3F	077		?	?	95 5F	137		_	_	127 7F	177			Del

charstable.com

Az ASCII kódtábla

II. Operációs rendszerek

Az operációs rendszer fogalma

- A számítógépet közvetlenül **gépi nyelv** szintjén programozhatjuk, ez azonban kényelmetlen lenne.
- Minden **alrendszer** (utasításkészlet, memóriaszervezés, I/O rendszer, sínstruktúra) a **programozónak** kellene lekezelnie a programjában, de ez nagyon nagy **energiapazarlással** járna az ember részéről, nem akarja.
- Az **operációs rendszer** elrejtí elölünk a hardvert: absztrakciós rétegeket hoz létre, hogy a hardver elérése a felhasználói programokból egyszerű legyen.
- Kiterjesztett vagy virtuális gépet biztosít a felhasználónak.

Az 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**.

Az operációs rendszerek történelme

- Számítógép generációkon keresztül tekintjük át az operációs rendszerek fejlődését.
- **I. generáció (1945-1955):** Vákuum csövek és kapcsolótáblák.
 - *Howard Aiken, Neumann János, J. Presper Eckert, John William Mauchley és Konrad Suse* számítógépek építésében értek el sikereket, először relék, majd **vákuum csövek alkották a gépeket**.
 - Programozás **gépi nyelven** történt, nem voltak programozási nyelvek sem, eleinte **kapcsolótáblákat** cserélgettek, majd az 50-es években megjelentek a **lyukkártyák**.
- **II. generáció (1955-1965): tranzistorok és kötegelt rendszerek**
 - Ma ezeket a gépeket nevezzük mainframe-eknek.
 - **Assembly, FORTRAN nyelvek** használata, lyukkártyán, később szalagon vitték be a programokat, melyek az eredményt a nyomtatóra küldték.
 - **Tudományos és mérnöki számításokra** használták, pl. parciális differenciálegyenletek numerikus megoldására, lényegében egy „**turbó számológép**”
 - Operációsrendszerek: **Fortran Monitor System** és **IBSYS** (IBM rendszere a 7094-re).
- **III. generáció (1965-1980): integrált áramkörök**
 - A korábbi szó-orientált gépek mellett megjelentek a **karakter-orientált gépek** (pl. 1401), melyeket **bankok 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 várta, azalatt a memória egy másik részében számítást végzett.

- **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.
- 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 rendszerbe** torkollott. A Unix **forrása nyílt** volt, mindenki saját, inkompatibilis változatot kezdett fejleszteni. Két fő változata: **System V és a BSD**.
- Az **IEEE POSIX** nevű **szabványát** a legtöbb mai Unix betartja
- **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.
 - 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 processzaira (*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.
 - Amikor a **Unix forrását lezárta az AT&T** a hallgatók előtt, Prof. Tanenbaum hozzáfogott a **Minix** fejlesztéséhez, melyet kisméretűnek tartott 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 terminál meghajtót is készített, később egy lemezmeghajtót és fájlrendszert, majd az eredményeket a USENET-en comp.os.minix csoportban közzétette, és segítőkre lelt így **1994. március 13-án megszületett a Linux 1.0**.

Processzus

- A **processzus** végrehajtás alatt lévő program, melyhez tartozik egy címtartomány (a memória egy szelete), amin belül a processzus írhat/olvashat.
- A processzus rendelkezik egy **regiszterkészlettel** is (utasításszámláló, veremmutató, stb.).

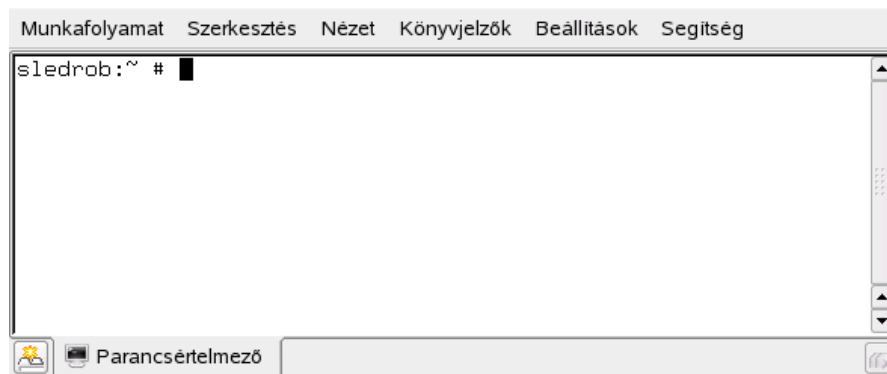
- Egy adott processzushoz tartozó információkat az operációs rendszer a **processzustáblázatban** tárol, és ezeket az adatokat használja fel, amikor az időbeosztás során az adott processzusnak újból ad egy CPU időszeletet.
- A processzusok **fa-struktúrát** alkotnak

Fájlok

- Az operációs rendszer 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 útvonala 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.

Parancsértelmező

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



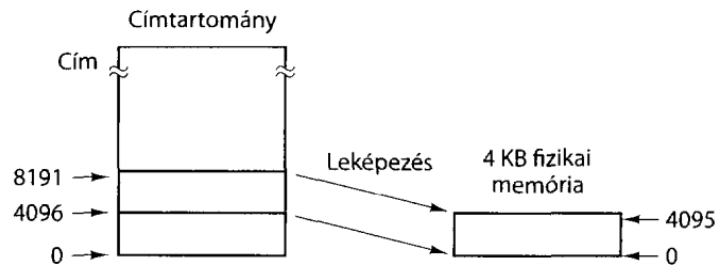
Parancsértelmező: bash

Rendszerhívások

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

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*).

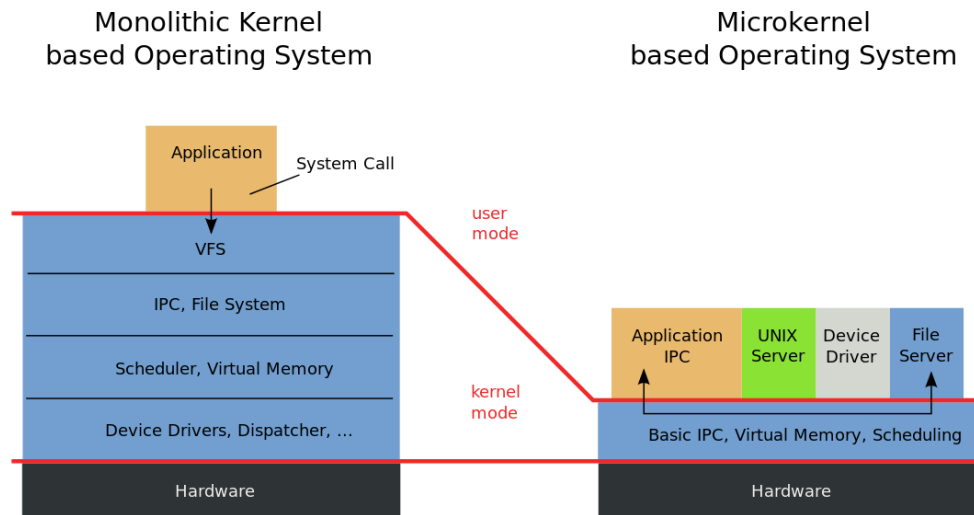


Virtuális memória működése, grafikus ábrázolása

- Ezáltal megtörtént a címtartomány és a memóriarekeszek fogalmának különválasztása.
- **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, kernel típusok

- **Monolitikus kernel:**
 - „Struktúrája a struktúrátlanság.”
 - Az operációs rendszer eljárások gyűjteménye, **bármelyik hívhatja a másikat korlátozás nélkül, nagy szabadságot ad.**
 - 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.**
 - *Ilyen a UNIX, és az ezen a rendszeren alapuló Linux kernel is, de az MS-DOS is még ezek közé tartozott.*
- **Microkernel:**
 - Szemben a monolitikus rendszerekkel a **mag méretét minimalizálják**, és külső forrásból éri el a kernel azokat a komponenseket, amiket nem tartalmaz.
 - *Pl.: Ilyen volt a Minix, vagy akár a Nokiás telefonokon futó Symbian OS is.*
- *A monolitikus és micro kernelen kívül megkülönböztetjük még a nanokernel, az exokernel, illetve a „hibrid kernel” is, melyek nevei a kernel méretére utalnak.*



A kernelek méretének grafikus összehasonlítása

- **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

- **Virtuális gépek**

- Egy adott hardver/szoftver architektúra **emulálás**át jelenti. (pl.: VMWare vagy Virtualbox)
- Első: VM/370 (1979): különválasztották a multiprogramozást és a hardver eléréséhez használt kiterjesztett gépet.

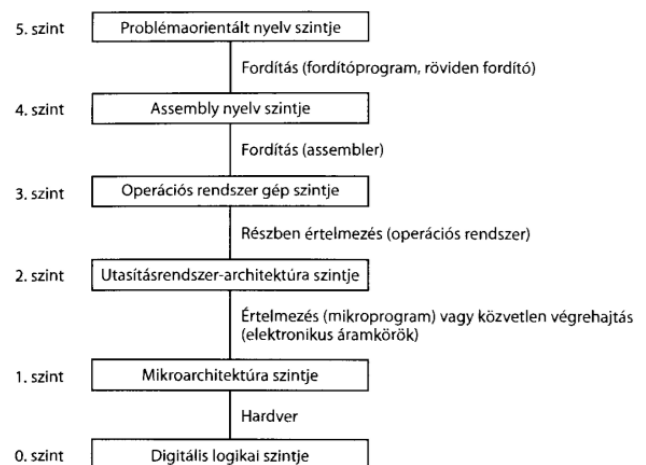
III. Architektúra

Számítógép architektúrák

- **Digitális számítógép:** problémák megoldása utasítások révén.
- **Program:** utasítások sorozata.
- Az elektronikus áramkörök az 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 a másikra!
- A **gépi nyelvek** kényelmetlenek az ember számára, ezért strukturálták a működést absztrakciók sorozatára, innen ered a **strukturált számítógép-felépítés**.

Többszintű számítógépek

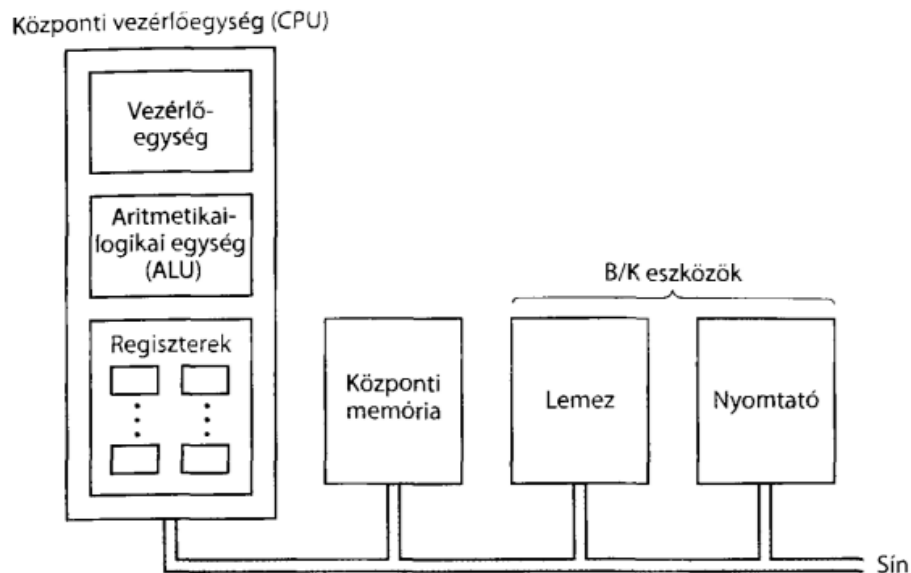
- 5. szint: Magas szintű nyelvek
- 4. szint: Szimbolikus nyelv
- 3. szint: Bővített utasítások
- 2. szint: ISA
- 1. szint: ALU
- 0. szint: kapuk
- -1. szint: elektronika



Számítógépek termékskálája

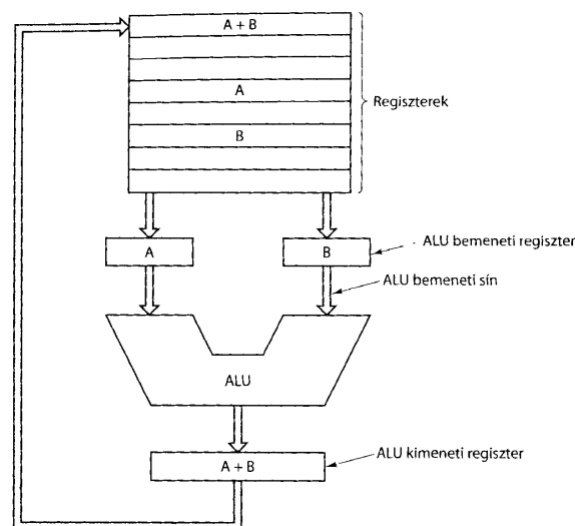
- **Eldobható számítógépeknek** nevezzük az olyan relatív primitív számítógépeket, melyek mindössze egy, de mindenesetre kevés használatra vannak fejlesztve (pl. zenélő képeslap).
- A mikrokontroller vagy **mikrovezérlő** egyetlen lapkára integrált, általában vezérlési feladatokra optimalizált cél-számítógép.
- A **személyi számítógép** (angolul: *personal computer, PC*) olyan számítógép, amely nem egy központi számítógép terminálja (munkaállomása), hanem önálló, egyetlen *személy* (az ún. *végfelhasználó*) által kezelt, hordozható méretű gép, saját billentyűzettel, processzorral, operatív memóriával és monitorral.
- A **szerver** (az angol *server* szóból) vagy kiszolgáló az informatikában olyan (általában nagy teljesítményű) számítógépet vagy szoftvert jelent, ami más számítógépek számára a rajta tárolt vagy előállított adatok felhasználását, a szerver hardver erőforrásainak (például nyomtató, háttértárolók, processzor) kihasználását, illetve más **szolgáltatások elérését teszi lehetővé**.
- **Elosztott rendszerek** (klaszterek) a **több számítógépből** álló rendszerek. Napjaink nagy számítógépes, vállalati rendszerei erre épülnek.
- A **nagyszámítógép**, (angol kifejezéssel: *mainframe*, szakmai körökben elterjedt kifejezéssel *nagygép* vagy *nagy vas*) azokat a nagy és „drága”, **nagy teljesítményű** számítógépeket jelenti, amelyeket főleg kormányzati intézmények, nagyvállalatok, és bankok használnak az üzletileg kritikus alkalmazásaik futtatására. Általában **nagy mennyiségű adat** kezelésére, tárolására képesek, amelyeket a statisztika, a nyilvántartások, ERP rendszerek, és a pénzügyi tranzakció feldolgozási folyamatok igényelnek.

Processzor felépítése és működése



A processzor részei és azok kapcsolata

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



A CPU felépítése

- **Adatút (Data path):**
 - Regiszterek
 - ALU (Arithmetical Logical Unit), a műveletek elvégző digitális áramkör.
 - Sínek vagy buszok
- Az **ALU** a bemeneti regiszterekből olvas be és az eredményt a kimeneti regiszterekbe írja. A fontos **ALU műveletek** az összeadás, kivonás, illetve az összehasonlítás nullával.

Utasítás végrehajtás

- **A végrehajtás lépései (betöltő-dekódoló-végrehajtó ciklus):**
 - A soron következő utasítás **beolvasása** a memóriából az utasításregiszterbe.
 - Az utasításszámláló 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.

A CISC és a RISC

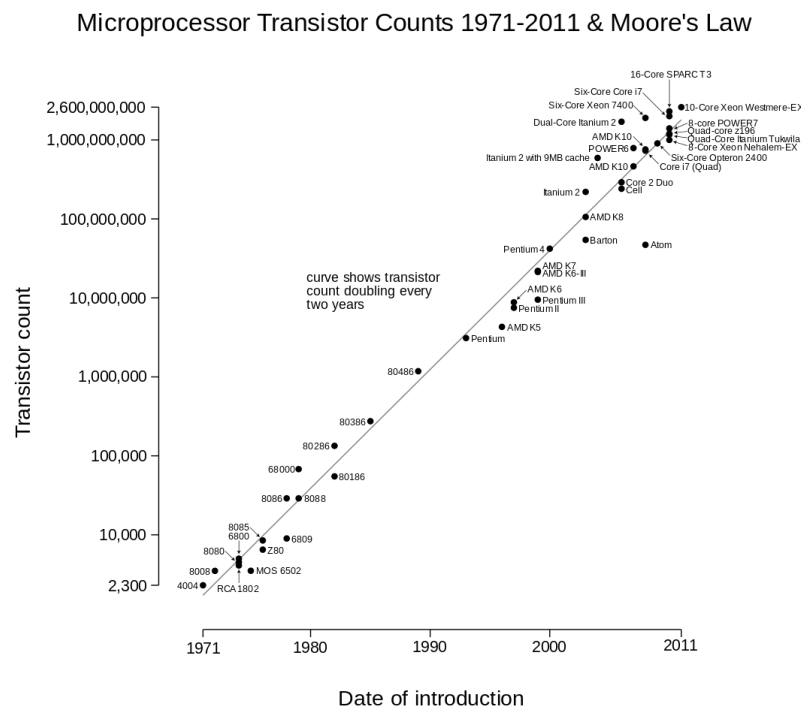
- **CISC:** Complex Instruction Set Computer, vagyis "összetett utasításkészlettel rendelkező számítógép" olyan **processzorokat** jelent, melyek **utasításkészlete** jóval **több, bonyolultabb utasítást** tartalmaz, mint a **RISC processzorok** utasításkészlete. A CISC processzorok utasításai általában **több elemi műveletet** végeznek **egyszerre**, így a gépi kódú programjaik **rövidebbek**, jobban átláthatók egy ember számára.
- **RISC:** A RISC angol rövidítés (*reduced instruction set computing, redukált utasításkészletű számítástechnika, vagy konkrét gépre alkalmazva reduced instruction set computer, „csökkentett utasításkészletű számítógép”*) egy, a számítógépek processzorának tervezésénél alkalmazott tervezési stratégia.
- 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)
- A CISC processzorok hátránya a RISC processzorokkal szemben az, hogy **a bonyolultabb utasítások sokszor jelentősen lassabban hajthatók végre**, és így a rövidebb programok ellenére is a végeredmény a **lassabb programfutás** lesz. Másik hátránya az, hogy a komplex utasítások jóval **bonyolultabb felépítésű processzorokat igényelnek**, melyek fejlesztése és tesztelése **költségesebb**.

CISC processzorok	RISC processzorok
Az utasítások összetettek, több gépi ciklust igényelnek.	Egy gépi ciklus alatt végrehajtható egyszerű utasítások.
Több utasítás is igénybe veheti a tárolót.	Csak a LOAD/STORE utasítások használhatják a tárat.
A pipelining feldolgozás kismértékű	Jelentős pipelining feldolgozás
Mikroprogram által vezérelt utasítás-végrehajtás	Huzalozott utasítás-végrehajtás
Változó hosszúságú utasítások	Rögzített hosszúságú utasítások
Sokféle utasítás és címezési mód	Kevés utasítás és címezési mód
Bonyolult mikroprogram	Bonyolult fordítóprogram
Kevés regiszter	Nagyméretű regisztertár

- 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.

Moore törvénye (*Moore's Law*)

- Gordon E. Moore (*Intel*): „Az integrált áramkörök összetettsége 18 hónaponként megduplázódik.” (1965)



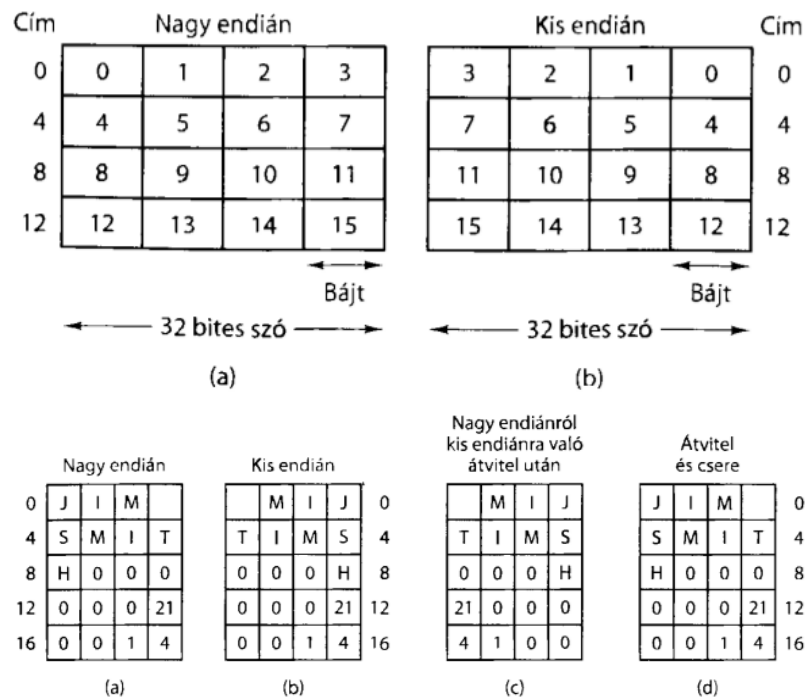
Moore törvénye grafikonon ábrázolva

Memória: Bitek és címzés

- A memória alapegysége a **bit** (0 vagy 1).
- Minél több feszültségszintet kell megkülönböztetnünk, annál bonyolultabb feladat, ezért "hatékony" 2-es számrendszert használni.
- **BCD** (*Binary coded decimal*): 4 biten tárol egy 10-es számrendszerbeli számot.
- 1944
 - **decimális**: 0001 1001 0100 0100
 - **bináris**: 0000 0111 1001 1000
- A memória egyforma, k méretű cellákba van rendezve, melyek 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: $2^{64} \cdot 1 \text{ Byte}$*)

Bájtsorrend

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



Gyorsítótár

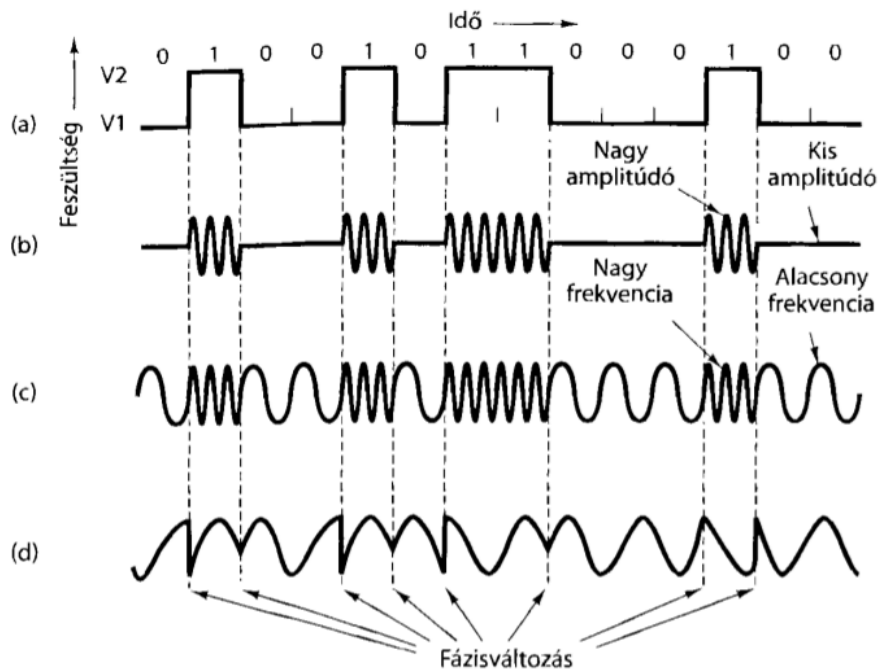
- CPU gyártók célja **sebesség növelése**, míg a memória gyártók a kapacitást növelik.
- Gyors memóriát a CPU lapkán kell elhelyezni, ez azonban drága lenne, a **sínen kapcsolt memória jóval olcsóbb**.
- **Hibrid megoldás**: kevés, gyors memória (*cache*) a CPU lapkán és sok, de lassú memória a sínen keresztül elérve.
- **Lokálási 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 bővítkártyák is saját **BIOS**-al rendelkezhetnek.
- Korábban hasonló funkciót töltöttek be a **firmware**-ek, 1975-ben vezették be a BIOS-t, mint (könnyen) módosítható változatot.
- A hardver és a szoftver közötti kapcsolat szerepét tölti be.
- **Feladatai**:
 - Hardver ellenőrzése
 - Hardvervezérlők betöltése
 - Operációs rendszer betöltése
 - Interfész biztosítása az operációs rendszer számára a hardware elemek eléréséhez.

Fizikai jelátviteli módszerek

- a) Kétszintű jel
- b) **Amplitúdómoduláció:** Az amplitúdómoduláció (*rövidítve: AM*) a jelátvitelben az amplitúdó változtatása, mely ezáltal az átvendő információt hordozza.
- c) **Frekvenciamoduláció:** Frekvenciamoduláció (*FM*). Frekvencia moduláció esetén a vivő hullám pillanatnyi frekvenciája a moduláló jel pillanatnyi amplitúdójával arányos.
- d) **Fázismoduláció:** nagyfrekvenciás elektromágneses hullámok fázisszögének változtatása a továbbítandó információt tartalmazó elektromos jel(ek) függvényében.



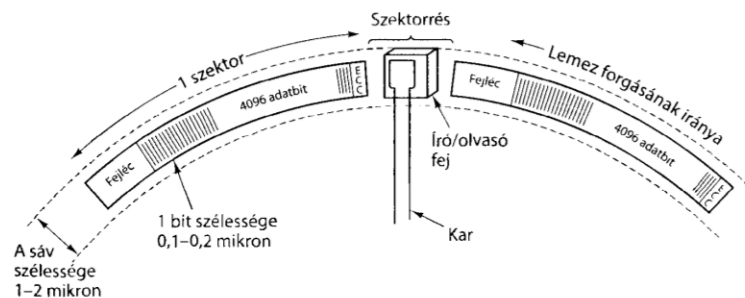
IV. Adattárolás

Háttérmemória, hierarchia

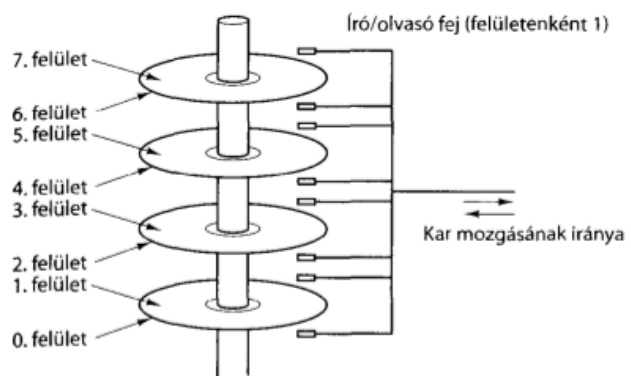
- Egy bájt tárolásának költsége **fentről lefelé csökken**, ahogy a sebesség is.
- **Jellemző elérési idők:**
 - **Regiszterek:** 1-5 ns
 - **Memória:** 10-50 ns
 - **SSD:** 0.1-0.3 ms
 - **Mágneselem:** 3-12 ms



A merevlemez



- A **merevlemez** (angolul **hard disk drive**, rövidítése **HDD**) egy számítástechnikai adattároló berendezés. Az adatokat kettes számrendszerben, mágnesezhető réteggel bevont, forgó lemezeken tárolja.
- **Alumínium lemez, mágnesezhető** bevonattal ellátva.
- Az indukciós tekercset tartalmazó fej a lemez felszíne felett légpárnán lebeg.
- Egy teljes körül fordulás alatt felírt **bitsorozat** a sáv, melyek **szektorokra** vannak bontva.



Solid State Drive (SSD)

- Az **SSD** (magyarul: *félvezető alapú meghajtó, tartós állapotú meghajtó vagy szilárdtest-meghajtó*) félvezető memóriát használó adattároló eszköz. **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 szabadíthatók fel későbbi írás céljára.

Csatolófelületek

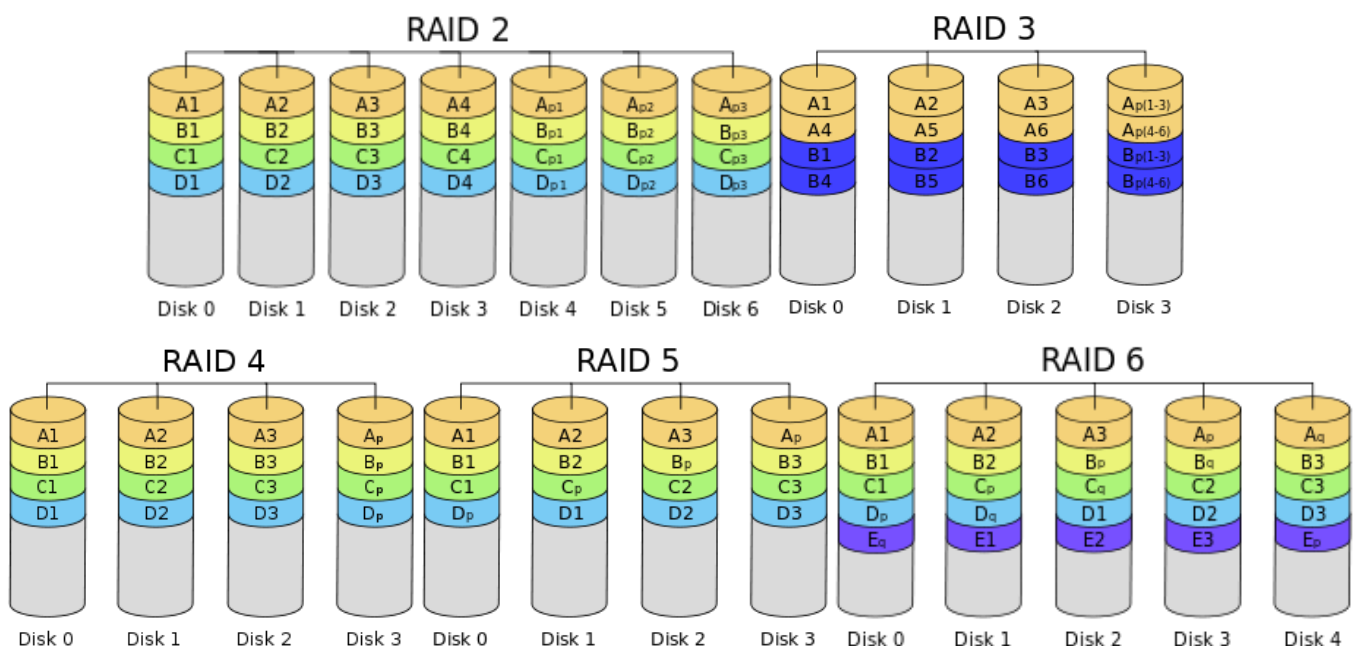
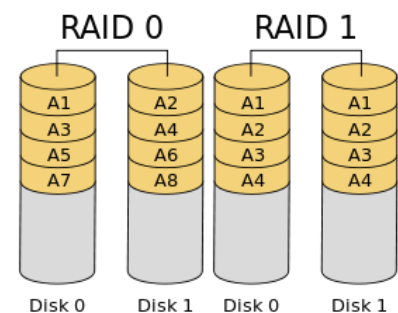
- **IDE/PATA:** Integrated Drive Electronics (1986)
- **SCSI:** Small Computer System Interface (1981)
- **SATA:** Serial AT Attachment (2003)
- **SAS:** Serial attached SCSI(2004)
- **iSCSI:** Internet Small Computer Systems Interface (2000)

Háttérmemória: Raid

- A **RAID** (angolul *Redundant Array of Inexpensive Disks* vagy *Redundant Array of Independent Disks*) tárolási technológia, mely segítségével az adatok elosztása vagy replikálása **több fizikailag független merevlemezen**, egy **logikai lemez** létrehozásával lehetséges. Minden RAID szint alapján véve vagy az **adatbiztonság növelését** vagy az **adatátviteli sebesség növelését** szolgálja.

• Raid levels²:

- **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 nagyméretű csíkokkal
- **RAID 5:** paritásinfó az összes meghajtón elosztatva
- **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



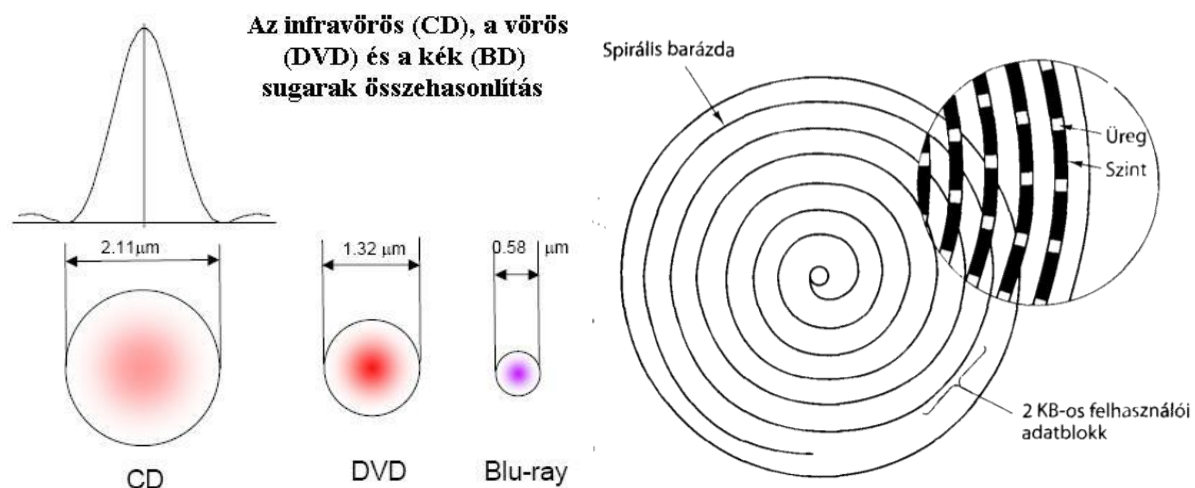
Raid szintek grafikusán ábrázolva

² <https://hu.wikipedia.org/wiki/RAID>

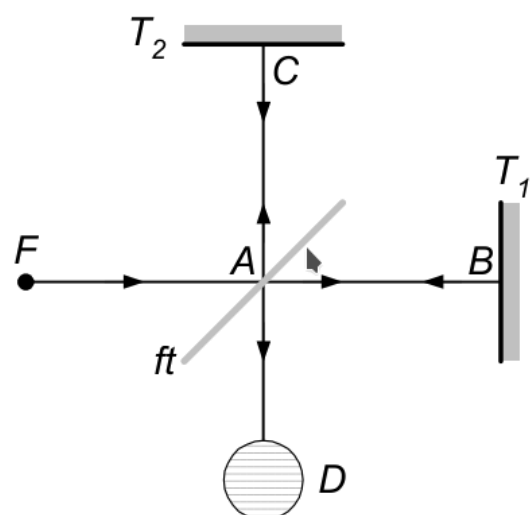
Optikai meghajtók

- 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 égetnek egy bevonattal ellátott üveg mesterlemezbe, amiről negatív öntőforma készül, végül az öntőformába olvadt polikarbonátot töltenek.
- A CD olvasásakor 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 fény között, amik az interferencia miatt gyengítik egymást.

Típus	Megjelenés éve	Kapacitás	Méret	Sebesség
CD	1982	700 MB	780 nm	1200 Kbit/s
DVD	1995	17.08 GB	650 nm	10.5 Mbit/s
Blue-ray	2007	128 GB	405 nm	576 it/s

Michelson-interferométer³

- 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 tükörről 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 tükörről áthaladva a detektáló eszközbe jut.



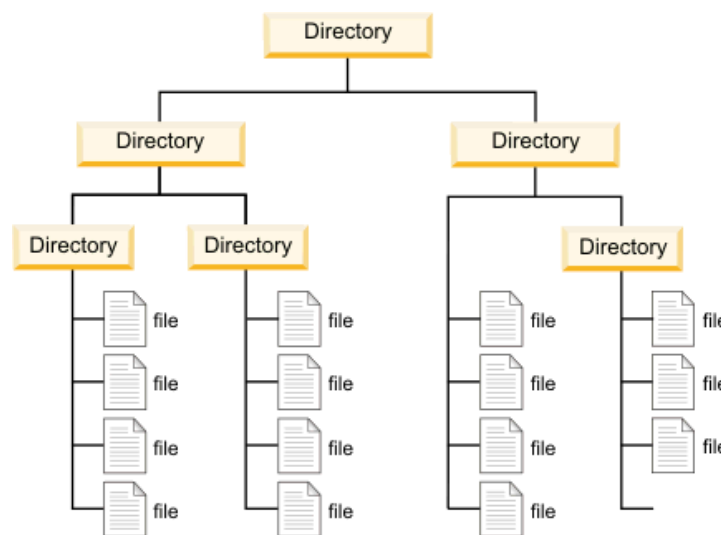
³ <http://www.falstad.com/ripple/>

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 újakezdeni az írást - ha ez gyakran megesik az **„fényesíti”** a szalagot.

Fájlrendszerek és fájlok

- Problémák amik a **fájlrendszerekhez** vezettek:
 - **A memória kicsi** ahhoz, hogy minden adat elérjen 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.



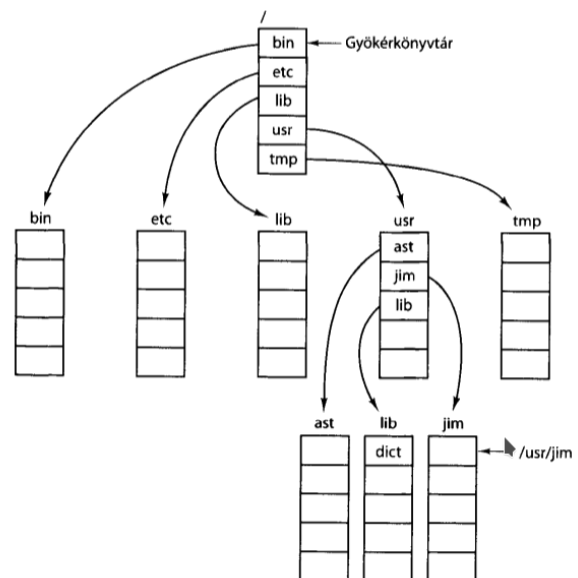
- **Absztrakciós mechanizmus**, lehetővé teszi az információ lemezen tárolását és visszaolvasását.
- **Fájlnev**: karakterek sorozata, egyes fájlrendszerek az utolsó pontot követő részt kiterjesztésként értelmezik, mely utal a fájl típusára.
- **Fájltípusok**:
 - Futtatható (bináris) fájlok
 - Adatfájlok (ASCII vagy bináris)
 - Speciális fájlok (operációsrendszer-specifikus)
- **Fájlműveletek**:
 - Létesítés / Törlés
 - Megnyitás (írásra, olvasásra) / Lezárás
 - Olvasás ésírás
 - Hozzáadás (append)
 - Pozícionálás (seek)
 - Attribútum írás és 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
Zárótsági jelző	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

Fájlattribútumok

Könyvtárak

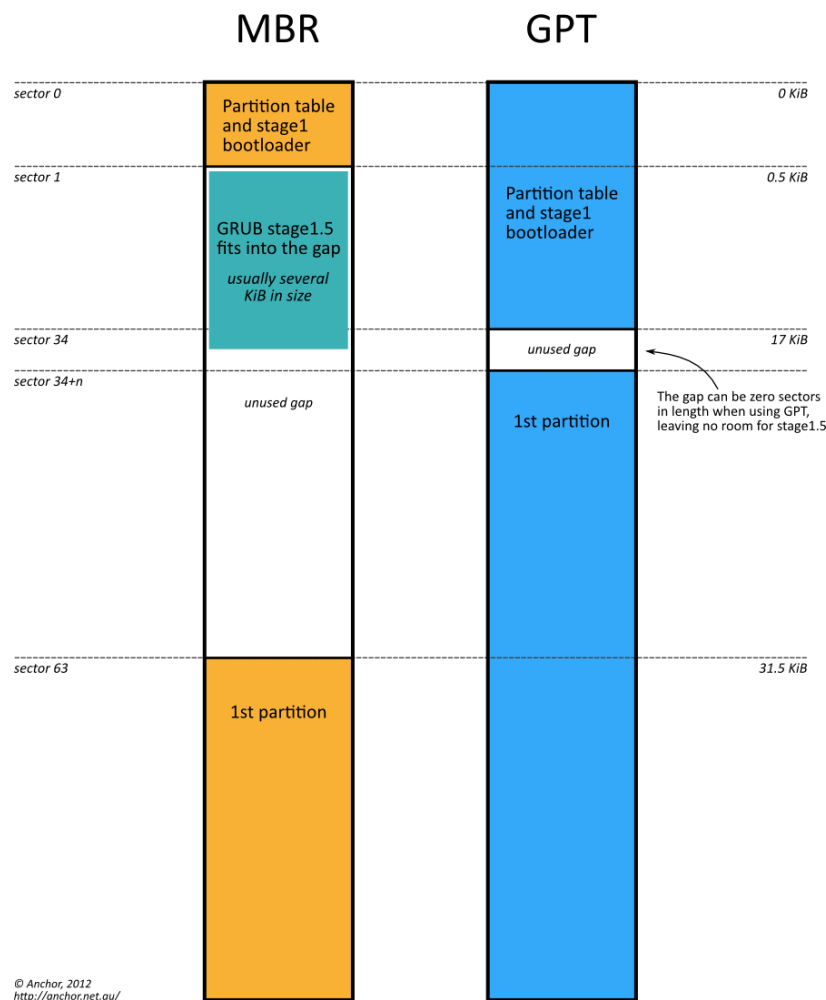
- **Könyvtárszerkezet:**
 - Egyszerű
 - Hierarchikus
- **Útvonal megadása:**
 - Abszolút
 - Relatív
- **Könyvtári műveletek:**
 - Létesít, töröl
 - Megnyit, lezár
 - Olvas
 - Átnevez
 - Kapcsol, lekapcsol (*link, unlink*)



Szerkezet

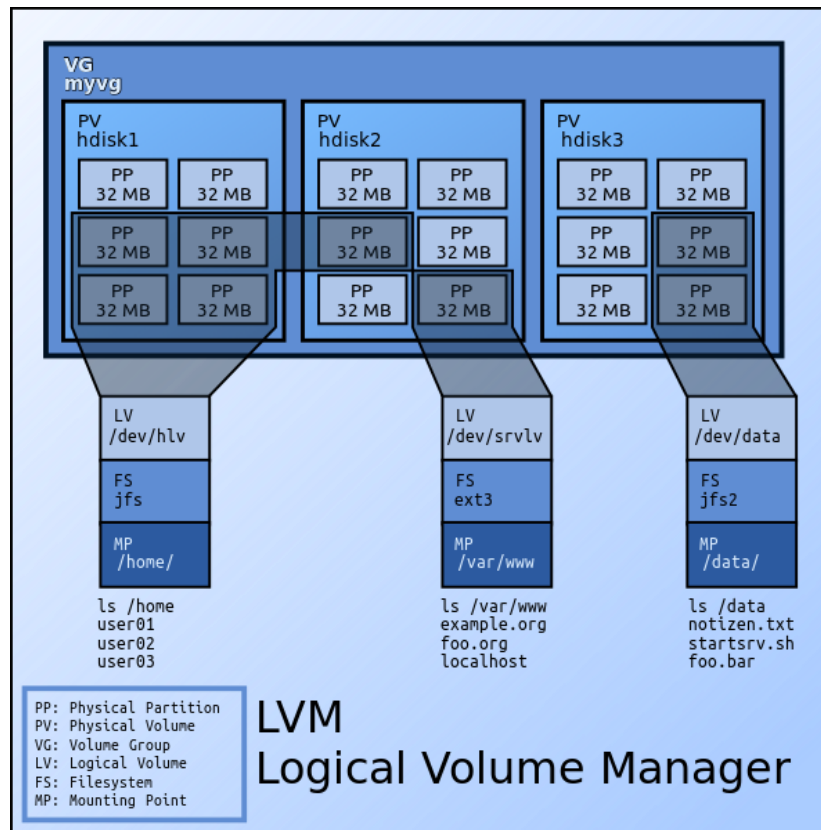
- A **Master Boot Record (MBR)** vagy más néven a partíciós szektor:
 - A lemez partíciókra van osztva
 - A lemez 0. szektor a MBR
 - (Master Boot Record)
 - Az MBR-ben lévő kódot induláskor
 - a BIOS tölti be
 - Az MBR után következik a
 - partíciós tábla
 - A tábla tartalmazza, hogy a partíciók a lemezen 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
- **A GUID Partition Table (GPT)** vagy másnéven GUID partíciós 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 < peta < exa < zetta < zetta
 - Backup a lemez végén
 - CRC32 ellenőrzőösszeg használata az adatsérülés detektálásához



Logical Volume management (LVM)

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



Fontosabb PC fájlrendszerek

- **ext2**: Natív Linux FS, felfelé kompatibilis
- **ext3, ext4**: az ext2 naplózó verziói
- **reiserfs**: robosztus FS, jól kezeli az adatkorruptiót
- **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