Bevezetés a számítástechnikába II. nagy ZH összefoglaló

Ekart Csaba - 2016



Tartalom

[I. Számítógépes hálózatok 4](#_Toc469003421)

[Társadalmi vonatkozások 4](#_Toc469003422)

[Protokollhierarchiák 4](#_Toc469003423)

[A rétegek tervezési kérdései 4](#_Toc469003424)

[Az OSI hivatkozási modell 4](#_Toc469003425)

[Az OSI és a TCP/IP hivatkozási modell összehasonlítása 5](#_Toc469003426)

[ARPANet 5](#_Toc469003427)

[Fizikai réteg: Adatátvitel 5](#_Toc469003428)

[Fizikai réteg: Vezetékes átviteli közeg 5](#_Toc469003429)

[Fizikai réteg: Vezeték nélküli adatátviteli 6](#_Toc469003430)

[Fizikai réteg: Egyéb kommunikációs megoldások 6](#_Toc469003431)

[Adatkapcsolati réteg: Tervezés szempontjai 6](#_Toc469003432)

[Adatkapcsolati réteg: Keretezés 6](#_Toc469003433)

[Adatkapcsolati réteg: Hibakezelés 7](#_Toc469003434)

[Adatkapcsolati réteg: Forgalomszabályozás 7](#_Toc469003435)

[Hálózati réteg: Nyújtott szolgáltatások 7](#_Toc469003436)

[Hálózati réteg: Tervezési kérdések 7](#_Toc469003437)

[Hálózati réteg: Forgalomirányítás 7](#_Toc469003438)

[Hálózati réteg: Forgalomirányítás 8](#_Toc469003439)

[Hálózati réteg: Az IP protokoll és az IP-címek 8](#_Toc469003440)

[Szállítási réteg: nyújtott szolgáltatás 8](#_Toc469003441)

[Alkalmazási réteg 8](#_Toc469003442)

[II. Operációs rendszerek 9](#_Toc469003443)

[Az operációs rendszer fogalma 9](#_Toc469003444)

[Az operációs rendszer, mint erőforrás-kezelő 9](#_Toc469003445)

[Az operációs rendszerek történelme 9](#_Toc469003446)

[Processzus 10](#_Toc469003447)

[Fájlok 11](#_Toc469003448)

[Parancsértelmező 11](#_Toc469003449)

[Rendszerhívások 11](#_Toc469003450)

[Virtuális memória 12](#_Toc469003451)

[Struktúrák, kernel típusok 12](#_Toc469003452)

[III. Architektúra 14](#_Toc469003453)

[Számítógép architektúrák 14](#_Toc469003454)

[Többszintű számítógépek 14](#_Toc469003455)

[Számítógépek termékskálája 14](#_Toc469003456)

[Processzor felépítése és működése 15](#_Toc469003457)

[Utasítás végrehajtás 16](#_Toc469003458)

[A CISC és a RISC 16](#_Toc469003459)

[Moore törvénye (Moore’s Law) 17](#_Toc469003460)

[Memória: Bitek és címzés 17](#_Toc469003461)

[Bájtsorrend 18](#_Toc469003462)

[Gyorsítótár 18](#_Toc469003463)

[Basic Input Output System (BIOS) 18](#_Toc469003464)

[Fizikai jelátviteli módszerek 19](#_Toc469003465)

[IV. Adattárolás 20](#_Toc469003466)

[Háttérmemória, hierarchia 20](#_Toc469003467)

[A merevlemez 20](#_Toc469003468)

[Solid State Drive (SSD) 20](#_Toc469003469)

[Csatolófelületek 21](#_Toc469003470)

[Háttérmamória: Raid 21](#_Toc469003471)

[Optikai meghajtók 22](#_Toc469003472)

[Michelson-interferométer 22](#_Toc469003473)

[Szalagos meghajtók 23](#_Toc469003474)

[Fájlrendszerek és fájlok 23](#_Toc469003475)

[Könyvtárak 24](#_Toc469003476)

[Szerkezet 24](#_Toc469003477)

[Logical Volume management (LVM) 25](#_Toc469003478)

[Fontosabb PC fájlrendszerek 26](#_Toc469003479)

# https://lh4.googleusercontent.com/hc2oKKw8iYP_ioPbi8by8BQcdVf5e0DsgUniIF1892zB4SfFzCXzL578q978XoVugzM6sceeNk2g00x60k88MwsXDywuypnua_12ma04gIKAlN_LzqI6nvlB5tKfTynxj18xZGp9cpAI. Számítógépes hálózatok

## Társadalmi vonatkozások

* A hálózatok széleskörű megjelenése új társadalmi, etikai és politikai problémákat vet fel (Laudon, 1995)
* Mi a gond a hírcsoportokkal, közösség portálokkal?
  + Amíg hobbiról, műszaki dolgokról folyik a beszélgetés nincs gond
  + Érzékeny témáknál jönnek elő a bajok, mint pl. politika, vallás, szexualitás
  + Gyakran kontrolláltalan, a vélemények megsérthetnek másokat
  + Nem csak szöveg, de kép, hang és videó anyagokat is publikálhatunk.
* Kormány és állampolgár viszonya: eszközt ad a titkosszolgálatok kezébe, megfigyelhetővé válnak az emberek.
* Anonim módon űzhető tevékenységek: levelezés, vásárlás-fizetés, stb.

## Protokollhierarchiák

* A hálózati hardverek és szoftverek bonyolultsága miatt strukturálásra volt szükség, ez vezetett protokollhiearchiák kialakításához.
* A legtöbb hálózatot úgy alakítják ki, hogy egymásra épülő rétegekre (layer) tagolják a rendszert.
* Minden réteg célja, hogy szolgáltatásokat (service) nyújtson a felette és alatta lévő rétegek számára.
* Protokoll: az egymással kommunikáló felek (->rétegek) közötti párbeszéd szabályait tartalmazza.

## A rétegek tervezési kérdései

* Minden rétegben kell legyen egy címzési mechanizmus, ami az üzenet küldőjét és vevőjét azonosítja.
* Hibavédelem: a fizikai átvitelt megvalósító áramkörök nem tökéletesek, hibajelző és hibajavító kódolásokat alkalmaznak a hibák kiszűrésére.
* Forgalomszabályozás: meg kell akadályozni, hogy a gyorsabban adó gépek elárasszák a hálózatot és ezzel ellehetlenítsék a lassabb adók kommunikációját.
* Multiplexelés, demultiplexelés: egy összeköttetésen több párbeszédet folytató folyamatpár átvitele.
* Forgalomirányítás (routing): hálózatok összekapcsolása.

## Az OSI hivatkozási modell

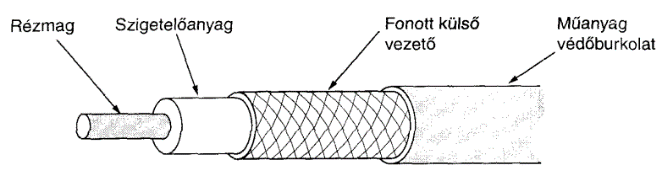
* ISO OSI: Open System Interconnection
* A különböző rétegekben használt protokolok első nemzetközi szabványosítása
* Rétegekre bontás szabályai:
  + A rétegek különböző absztrakciós szinteket képviseljenek.
  + Minden réteg jól definiált feladatot hajtson végre.
  + A rétegek szabványosításakor nemzetközi protokollokat kell figyelembe venni.
  + Rétegek határait úgy kell meghatározni, hogy a rétegek közötti információcsere minimális legyen.

## https://lh6.googleusercontent.com/eDBgmuYdnVJSbOyn383PSeypxAnFkiJymfUbS7_mkl5gKGmRl77WvnA6gOamRBXwCTa204GqX_RLq4ufF5b0aGym3vqkRYZNiIah6CANxgl4TY83RwKQcfn_5QqCHgv5z_GO2e5Yo1UAz OSI és a TCP/IP hivatkozási modell összehasonlítása

## ARPANet

* Az Internet őse, az Amerikai Védelmi Minisztérium kísérleti hálózata volt.
* Alapvető szempont volt, hogy lehetővé tegye tetszőlegesen sok hálózat zökkenésmentes összekapcsolását.
* Eleinte bérelt vonalakat használtak, később műholdakat és rádiós hálózatokat is hozzákapcsoltak.
* Másik fontos szempont volt, hogy amíg a forrás és célállomások jól működnek, a folyó beszélgetést ne zavarja alhálózatok kiesése (decentralizált és redundáns útvonalak).
* Két legismertebb protokollja nyomán vált híressé: TCP/IP.

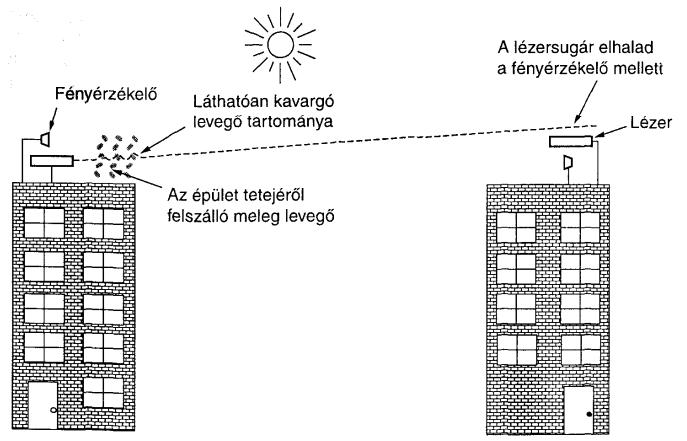
## Fizikai réteg: Adatátvitel

* A 19. század elején Jean-Baptiste Fourier francia matematikus bebizonyította, hogy bármely T periódusidővel rendelkező, periodikus g(t) függvény előállítható szinuszos és koszinuszos tagok (általában végtelen) összegeként.
* Az adatátvitel során energiaveszteség lép fel, ami az időtartományban változó lehet.
* Sávkorlátozott jelek: A 0 és fc vágási frekvencia között a jel kis mértékben torzul, felette arányosan nagyobb mértékben, ezért érdemes a jel frekvenciáját korlátozni.
* Sávszélesség: az a frekvenciatartomány, amin belül a csillapítás mértéke “nem túl nagy”
* Nyquist-Shannon tétel: a 0…f0 frekvenciatartományba eső időfüggvény véges számú minta segítségével, információvesztés nélkül átvihető, ha az fminta mintavételezési frekvencia f0 frekvenciának legalább kétszerese: fminta ≥ 2 f0

## Fizikai réteg: Vezetékes átviteli közeg

* Számos átviteli módszer létezik:
  + Mágneses adathordozó (nem szabad lebecsülni
  + annak a furgonnak az átviteli sebességét, ami
  + mágnesszalagokkal telepakolva száguld az autópályán)
  + Sodrott érpár (twisted pair)
  + Koaxális kábel
  + Fényvezető szálak

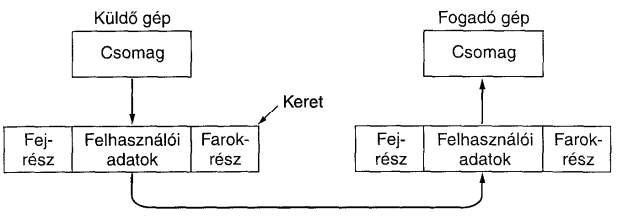
## Fizikai réteg: Vezeték nélküli adatátviteli

* Előnyös, ha zavaró a helyhez kötöttség, vagy, ha a vezetékes adatvitel kiépítése nehézkes lenne.
* Fajtái:
  + Rádiófrekvenciás átvitel
  + Mikrohullámú átvitel
  + Infravörös és mm-es hullámú átvitel
  + Látható fényhullámú átvitel
  + Ultrahang alapú átvitel
* Az ábrán egy, a lézeres adatátvitelt befolyásoló tényező látható.

## Fizikai réteg: Egyéb kommunikációs megoldások

* Kommunikációs műholdak
* Nyilvános kapcsolt telefonhálózat
* Mobiltelefon rendszer
* Kábeltelevízió (DOCSIS, …)

## Adatkapcsolati réteg: Tervezés szempontjai

* A probléma egyszerű lenne, azonban:
  + a kommunikációs áramkörök néha hibáznak
  + nem nulla késleltetéssel továbbítják az adatokat
  + véges az adatátviteli sebességük
* Megoldások:
  + Keretezés
  + Hibakezelés
  + Forgalomszabályozás

## Adatkapcsolati réteg: Keretezés

* Az érkező bitsorozat hibamentességét a fizikai réteg nem garantálja.
* A bitek száma lehet kevesebb vagy több, mint az elküldötteké, és az értékük is különbözhet az eredetitől.
* Az adatkapcsolati réteg feladata, hogy jelezze, illetve - ha szükséges - kijavítsa a hibákat.
* Keretezési módszerek:
  + Karakterszámlálás
  + Kezdő- és végkarakterek karakterbeszúrással
  + Kezdő- és végjelek bitbeszúrással
  + Fizikai rétegbeli kódolássértés

## Adatkapcsolati réteg: Hibakezelés

* Az adónak valamilyen visszacsatolást biztosítunk arról, hogy mi történik a vonal másik végén
* Hiba esetén az adó újra-adja a keretet
* Ha egy keret elveszik, a nyugta nem érkezne meg a küldőhöz a fogadótól és így csak várna az örökkévalóságig, ezért bevezették az időzítéseket
* Ha viszont a keret vagy a nyugta elveszik, az időzítő lejár, és jelzi az adónak, hogy valószínűleg hiba történt
* Megoldás: egyszerűen újra elküldeni a kere­tet, azonban sorszámozni kell, a duplikált küldés elkerülése miatt

## https://lh3.googleusercontent.com/N_to09DG72uUDdgQ009JQD8UB45i3OJsWZOyw4z5M0sz3DA5-d7Od7EawqMt4rt8BdHmFyy_gJS0Xb6V4snDVbVB5H_J3X0FAuxmMzuDnB9BVnKeGFisxG6Jv-UFfp4nffvf-u6YYQUAdatkapcsolati réteg: Forgalomszabályozás

* Feedback-based flow control (Visszacsatolás alapú forgalom- szabályozás): A vevő információt küld az adónak, amiben engedélyezi a további csomagok küldését
* Rate-based flow control (Sebesség alapú forgalom szabályozás): A protokollba be van épít­ve egy sebességkorlát, amelyet minden küldőnek minden adattovábbítás során tiszte­letben kell tartania
* *Támadási felület: DDoS (Distributed Denial of Service) attack, vagyis elosztott szolgáltatás- megtagadással járó támadás, pl. SYN flood.*

## Hálózati réteg: Nyújtott szolgáltatások

* A hálózati réteg feladata, hogy a csomagokat a forrástól egészen a célig eljuttassa (az adatkapcsolati rétegnél a keretet kellett továbbítani a vonal két végpontja között)
* A hálózati rétegnek ismernie kell a kommunikációs alhá­lózat (vagyis a routerek halmaza) topológiáját, és megfelelő útvonalakat kell találnia azon keresztül
* Úgy kell kiválassza a routereket, hogy elkerülje néhány kommunikációs vonal és router túlterhelését
* Ha forrás és a cél különböző hálózatokban vannak, ezt a problémát is a hálózati réteg oldja meg

## Hálózati réteg: Tervezési kérdések

* A szolgálatoknak függetleneknek kell lenniük az alhálózat kialakításától
* A szállítási réteg elől el kell takarni a jelenlevő alhálózatok számát, típusát és to­pológiáját
* A szállítási réteg rendelkezésére bocsátott hálózati címeknek egységes számozási rendszert kell alkotniuk, még LAN-ok és WAN-ok esetén is

## Hálózati réteg: Forgalomirányítás

* A forgalomirányító algoritmus (routing algorithm) a hálózati réteg szoftverének azon része, amely azért a döntésért felelős, hogy egy bejövő csomag melyik kimeneti vonalon kerüljön továbbításra
* Nem adaptív vagy statikus forgalomirányítás nem használ fel méréseket az útvonalak meghatározása során
* Az adaptív algoritmusok (adaptive algorithms) úgy változtat­ják forgalomirányítási döntéseiket, hogy tükrözzék a topológiában és rendszerint a forgalomban is történt változásokat

## Hálózati réteg: Forgalomirányítás

* A forgalomirányító algoritmus (routing algorithm) a hálózati réteg szoftverének azon része, amely azért a döntésért felelős, hogy egy bejövő csomag melyik kimeneti vonalon kerüljön továbbításra
* Nem adaptív vagy statikus forgalomirányítás nem használ fel méréseket az útvonalak meghatározása során
* Az adaptív algoritmusok (adaptive algorithms) úgy változtat­ják forgalomirányítási döntéseiket, hogy tükrözzék a topológiában és rendszerint a forgalomban is történt változásokat

## Hálózati réteg: Az IP protokoll és az IP-címek

* A ragasztó, amely az Internetet egyben tartja
* A kezde­tektől a hálózatok összekapcsolását szem előtt tartva tervezték
* Az Interneten a kommunikáció úgy történik, hogy a szállítási réteg az adatfolyamot datagramokra tördeli
* Egy IP-datagram egy fejrészből és egy szövegrészből áll, a fejrésznek van egy 20 bájtos rögzített része és egy változó hosszúságú opcioná­lis része
* A fejrészben található: verzió, szolgálat típusa, teljes hossz, azonosítás, élettartam, protokoll, CRC, ...
* Az Interneten minden hosztnak és routernek van egy IP-címe, amely a hálózat számát és a hoszt számát kódolja
* Jelenleg még az IPv4 az elterjedt, azonban ez kb. 2564 címet jelent, ami kevésnek bizonyult
* Az IPv6 32 helyett 128 bitesek, fontos jellemezői:
  + Megnövelt címtartomány
  + Közvetlen végponti címezhetőség
  + Automatikus konfiguráció
  + Hálózati mobilitás (roaminghoz hasonló)
  + Titkosítás, azonosítás (IPsec biztonsági protokoll)
  + Többszörös címezhetőség, szabványosított multicast

## Szállítási réteg: nyújtott szolgáltatás

* Feladata az, hogy megbízható, gazdaságos adatszállítást biztosítson a forráshoszttól a célhosztig, függetlenül magától a fizikai hálózattól vagy az aktuálisan használt kommunikációs alhálózatoktól
* TCP (Transmission Control Protocol)
  + megbízható, összeköttetés alapú protokoll, hibamentes bájtos átvitel
* UDP (User Datagram Protocol)
  + nem megbízható, összeköttetés nélküli protokoll - gyors, de nem biztosított a csomagok érkezési sorrendje

## Alkalmazási réteg

* Fontosabb alkalmazások:
  + Domain Name System (DNS) <- csak ezt részletezzük
  + E-mail
  + Multimédia (kép és hangátvitel)
  + Hipertext Transfer Protocol (HTTP) és World Wide Web (WWW)

# II. Operációs rendszerek

## Az operációs rendszer fogalma

* A számítógépet közvetlenül **gépi nyelv** szintjén programozhatjuk, ez azonban kényelmetlen lenne.
* Minden **alrendszert** (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** elrejti előlünk 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** cserélgették, majd az 50-es években megjelentek a **lyukkártyák**.
* **II. generáció** (1955-1965): **tranzisztorok** 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**: amig 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 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.
* Amikor **a Unix forrását lezárta az AT&T** a hallgatók elől, Prof. Tanenbaum hozzáfogott a **Minix** fejlesztéséhez, melyet kisméretűnek tartott meg, hogy a hallgatók is tudják futtatni gépjeiken, így tett az egyik hallgató **Linus Torvalds** is.
* Torvaldsnak **hiányzott pár funkció a Minixbő**l, 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.**

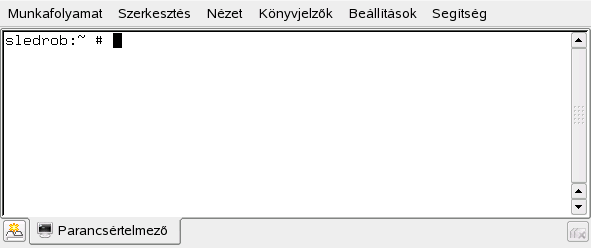
## Processzus

* 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.

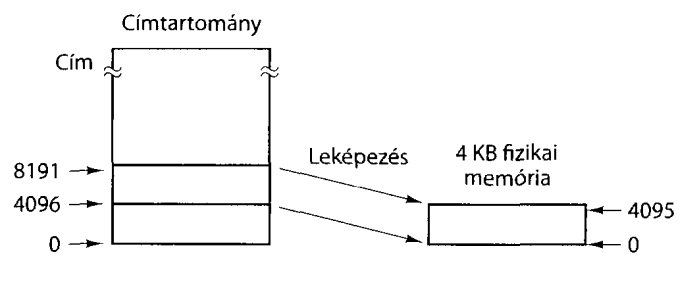
Parancsértelmező: bash

## Rendszerhívások

* A **rendszerhívásokkal** a felhasználói programok jelzik a rendszermag számára, hogy feladatot kell, végrehajtson.
* Gyakorlatilag olyan **eljáráshívások**, amik a magba más privilegizált operációs rendszer komponensbe tudnak belépni.
* **Csoportjai**:
  + Processzuskezelő rendszerhívások
  + Szignálkezelő rendszerhívások
  + Fájlkezelő rendszerhívások
  + Könyvtárkezelő rendszerhíváok
  + 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)*.

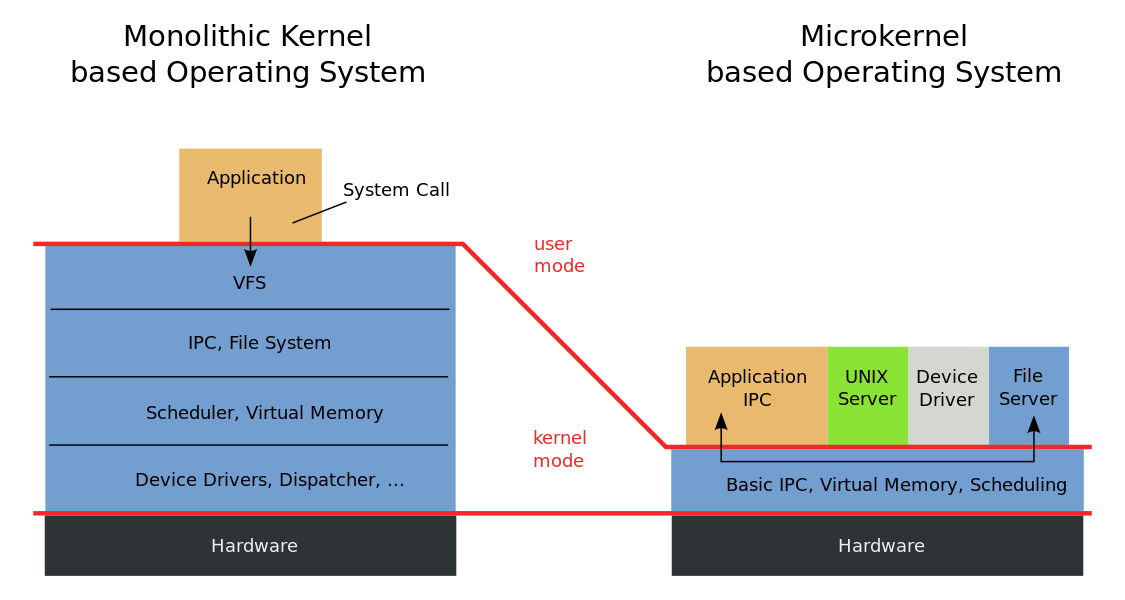


Virtuális memória működése, grafikus ábrázolása

* 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 memórirekeszeket a fizikai címtartomány címzi meg.
  + A memóriatérkép az egyes virtuális címeknek megfelelő fizikai címeket határozza meg.

## Struktúrák, kernel típusok

* **Monolitikus kernel**:
  + „Struktúrája a struktúrálatlanság.”
  + 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 nanokernelt, az exokernelt, illetve a „hibrid kernelt” 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 **emulálását** 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.**

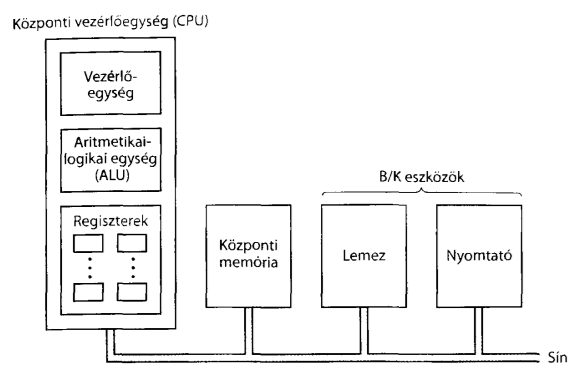
## Többszintű számítógépek

* 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

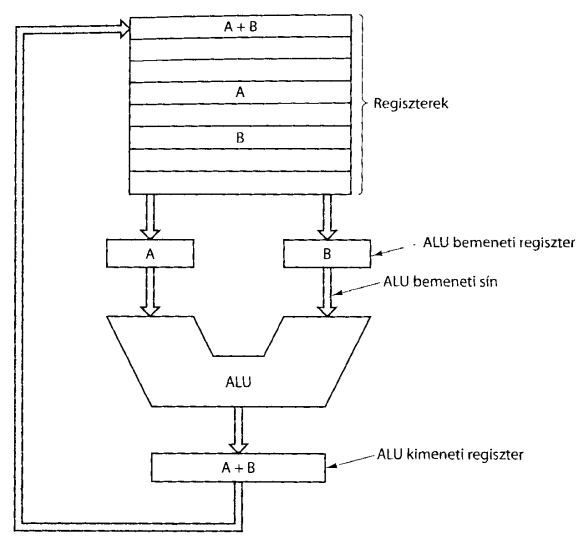
* **Eldobható számítógépeknek** nevezzük az olyan relatív primitív számítógépeket, melyek mindössze egy, de mindenesetre kevés használatra vannak fejlesztve (pl. zenélő képeslap).
* A mikrokontroller vagy **mikrovezérlő** egyetlen lapkára integrált, általában vezérlési feladatokra optimalizált cél-számítógép.
* 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 *(Artimetical 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

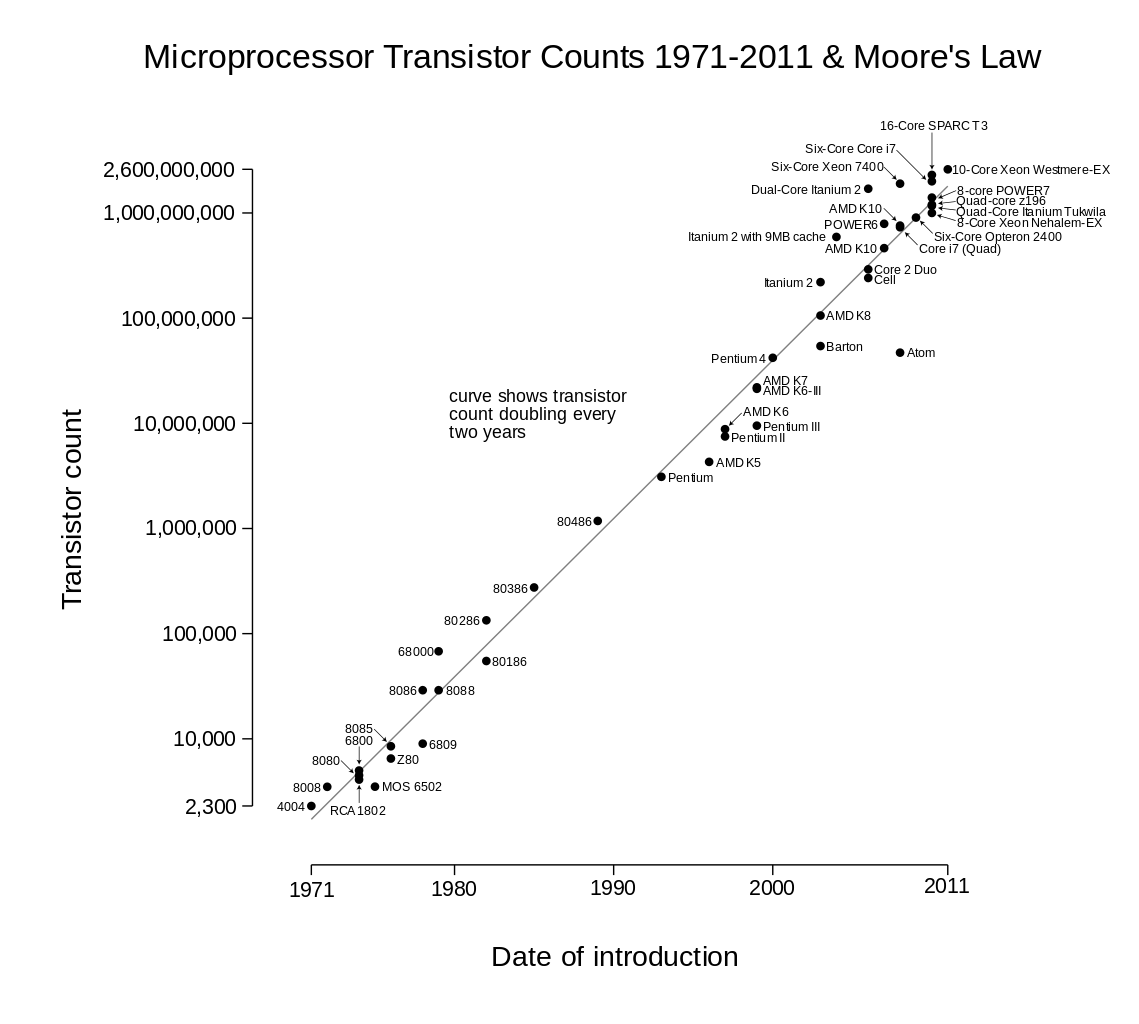
* **CISC**: Complex Instruction Set Computer, vagyis "összetett utasításkészlettel rendelkező számítógép" olyan **processzorokat** jelent, melyek **utasításkészlete** jóval **több, bonyolultabb utasítást** tartalmaz, mint a **RISC processzorok** utasításkészlete. A CISC processzorok utasításai általában **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 címzési mód | Kevés utasítás és címzési 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)*

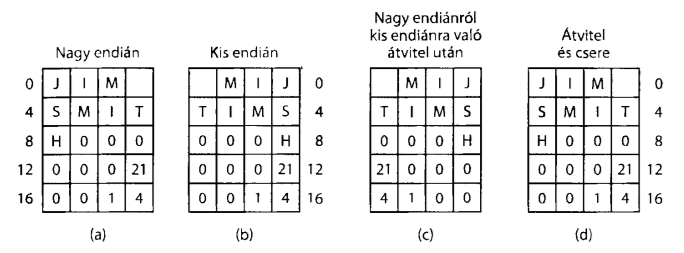
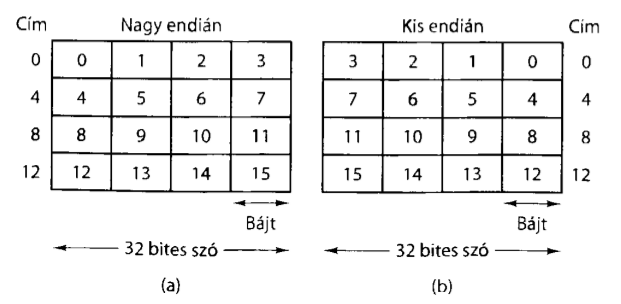


Moore törvénye grafikonon ábrázolva

## 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\*2k 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: 264 \* 1 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

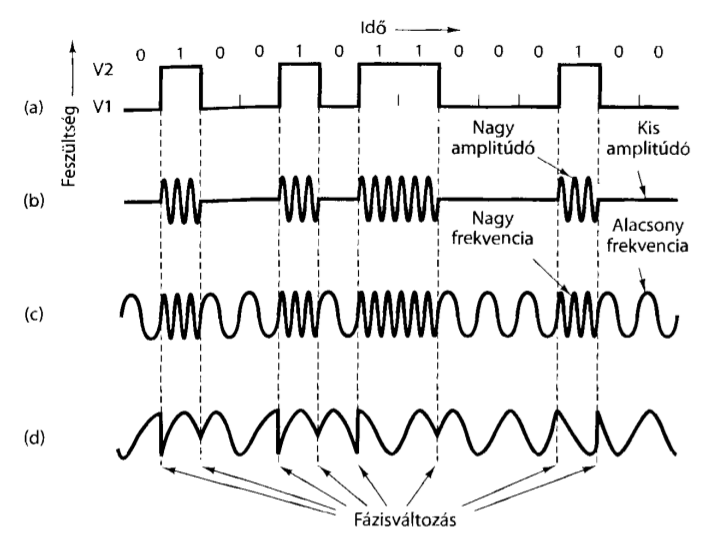
* CPU gyártók célja **sebesség növelése**, míg a memória gyártók a kapacitást növelik.
* Gyors memóriát a CPU lapkán kell elhelyezni, ez azonban drága lenne, a **sínen kapcsolt memória jóval olcsóbb.**
* **Hibrid megoldás**: kevés, gyors memória *(cache)* a CPU lapkán és sok, de lassú memória a sínen keresztül elérve.
* **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 alaplapin 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.
* 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

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

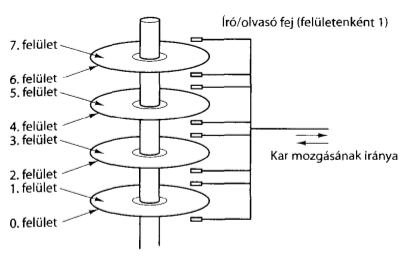


# https://lh3.googleusercontent.com/G01JMZ3kBvNKcERWUGUqXD2QsDXFEyLO-DHL1O4p5ibzyly5PrZEibJq0Ss0qQP1FrEdfJbti6KE-2lMBUL16bkPsLUbQdHuIiHcH3TNe5CJqqfLnNPel4EmB3vPg438rrc8jfBDcdoIV. Adattárolás

## 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
  + **Mágneslemez**: 3-12 ms

## https://lh6.googleusercontent.com/uvL0Ss8DD-lz9t74HishX0WQT80JplVvDIr--YhPJ7O7VbR8HEodCM7S4umVNFk1AMxR829eVL_9usQGQg5DSxR12xQVsuJMqnHbJ5KeCsFtoSR5FZK9rLfO2_vafhsdrbxVtPdN-t0A 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.

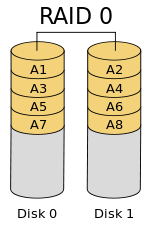
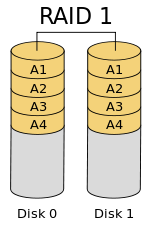
## Solid State Drive (SSD)

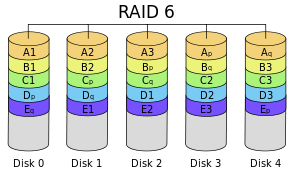
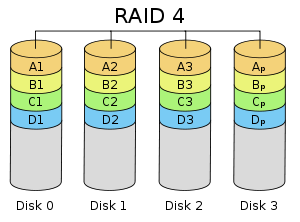
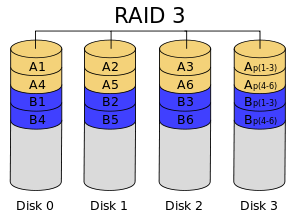
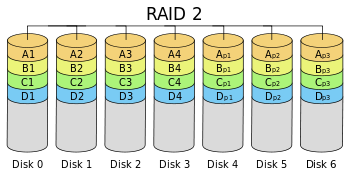
* Az **SSD** *(magyarul: félvezető alapú meghajtó, tartós állapotú meghajtó vagy szilárdtest-meghajtó)* félvezetős memóriát használó adattároló eszköz. **A HDD-k alternatívája.**
* **Flash-memóriát** alkalmaznak bennük, amik azután **megtartják az adatot**, miután az **áramforrás megszűnik.**
* Készítenek **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)

## Háttérmamória: 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 alapjában véve vagy az **adatbiztonság növelését vagy az adatátviteli sebesség növelését** szolgálja.
* **Raid levels**[[1]](#footnote-1):
  + **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 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

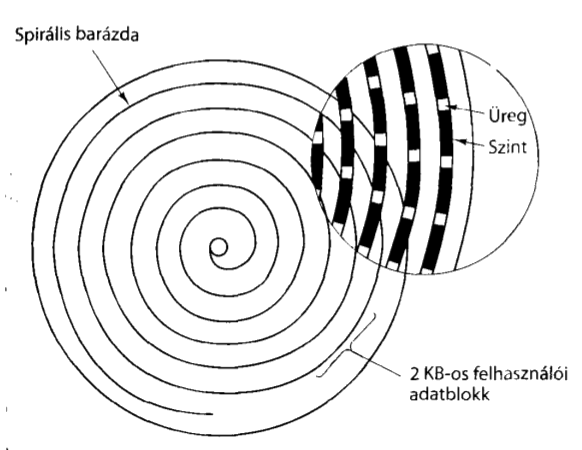
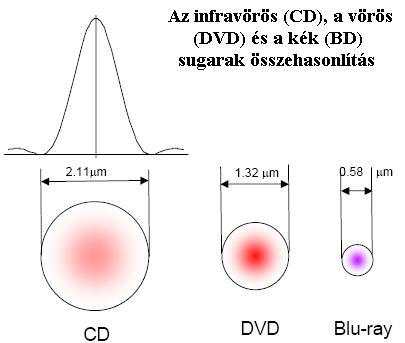


Raid szintek grafikusan ábrázolva

## 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 **lézerdiófa** 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 | Méretek | 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 | 1. it/s |



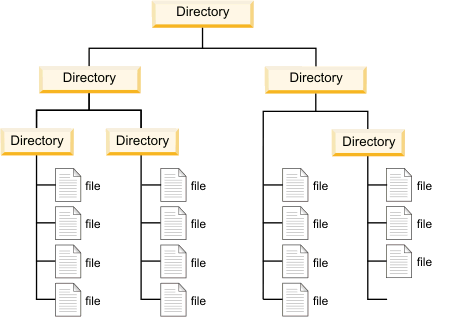
## https://lh5.googleusercontent.com/goP6S0dNDvXsDa_1gax4DpABAxQsX6kz7hmaXa6frBTAPLkfkmEUSbkitcyjlkKpJ0HCZmBoyMJJuMVgLnAbHQP_JC7FS4BveEgmWRMbFz4ZUwX8yIqTC33lVKJM2hEuVPplp7nCd10Michelson-interferométer[[2]](#footnote-2)

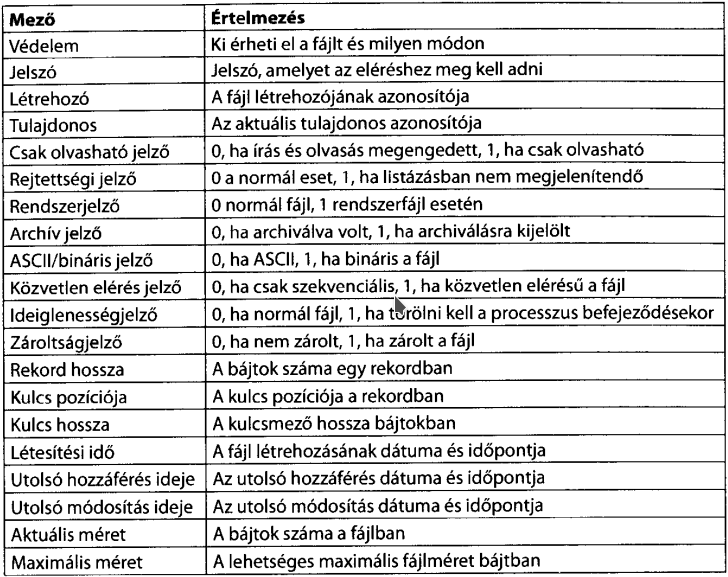
* Az F fényforrásból kiinduló fény az A pontban eléri a fénysugár irányával 45° szöget bezáró ft félig áteresztő tükröt.
* A tükör a fényintenzitás egy részét átengedi, és ez a rész a T1 tükörről visszaverődve visszaér az A pontba, majd egy része az ft tükrön visszaverődve a detektáló eszközbe (D) jut.
* A fényintenzitás másik részét az ft tükör az eredeti fénysugárra merőleges irányban visszaveri, így az a T2 tükörre kerül.
* Onnan visszaverődik, és egy része az ft tükrön áthaladva a detektáló eszközbe jut.

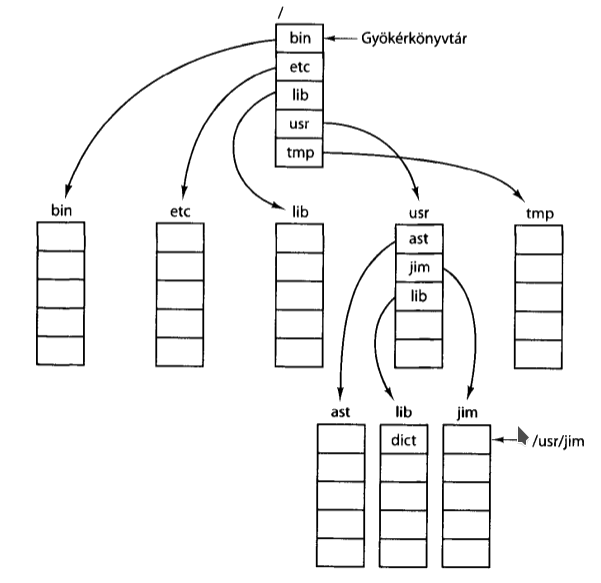
## Szalagos meghajtók

* 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 újrakezdeni 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 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.
* **Absztrakciós mechanizmus**, lehetővé teszi az információ lemezen tárolását és visszaolvasását.
* **Fájlnév**: 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
  + Hozzátoldás (append)
  + Pozícionálás (seek)
  + Attribútum írás és attribútum olvasás
  + Átnevezés
  + Zárolás (lock)



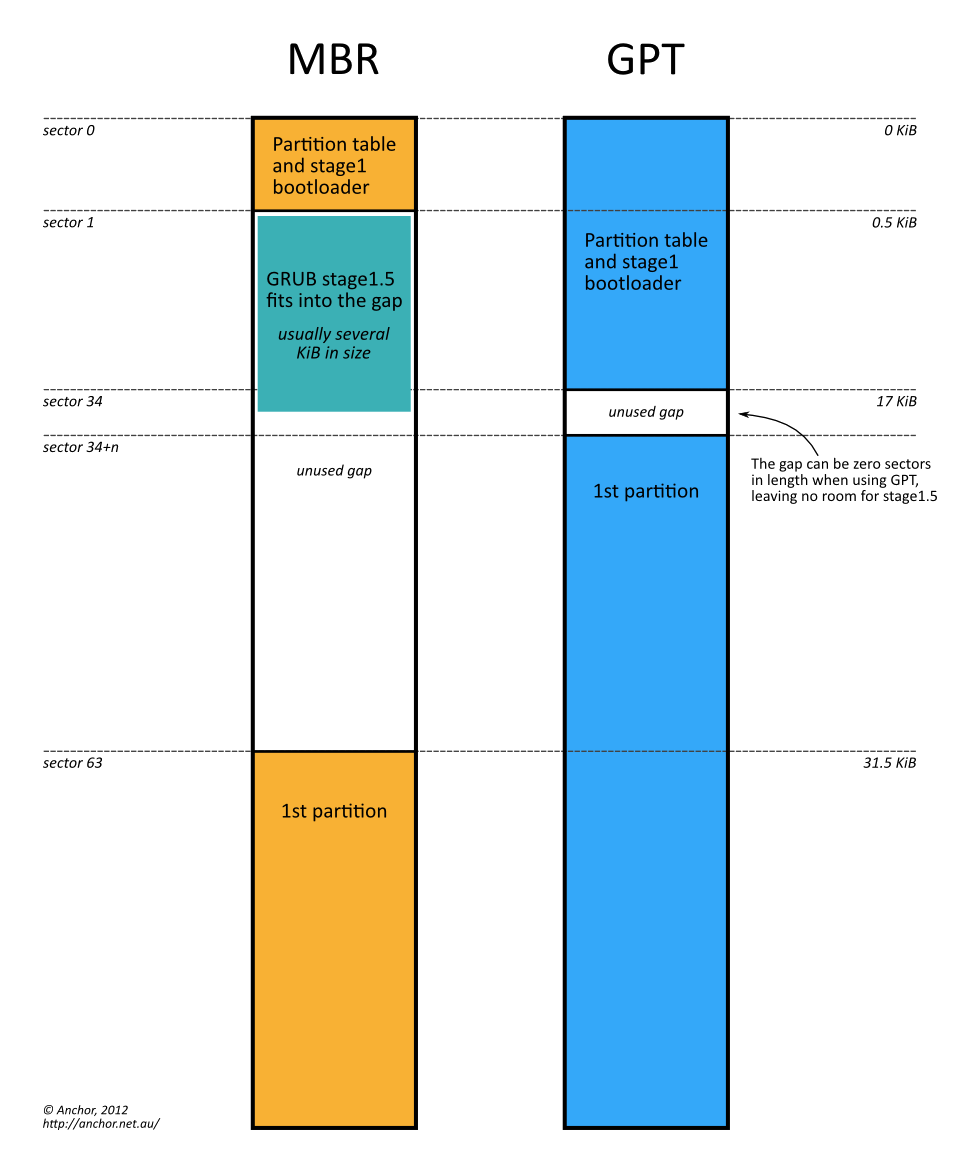
Fájlattribútumok

## Könyvtárak

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

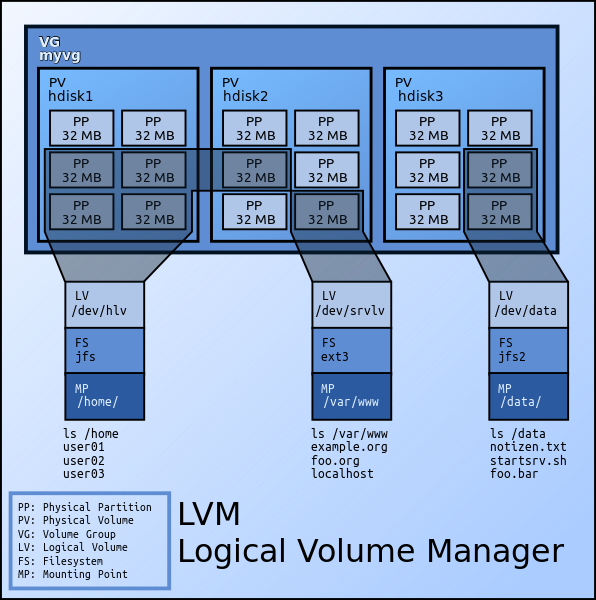
## Szerkezet

* A **Master Boot Record** (MBR) vagy más néven a partíciós szektor:
  + A lemez partíciókra van osztva
  + A lemez 0. szketora 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
* **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 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

1. *https://hu.wikipedia.org/wiki/RAID* [↑](#footnote-ref-1)
2. *http://www.falstad.com/ripple/* [↑](#footnote-ref-2)