

# A permanens tár kezelése

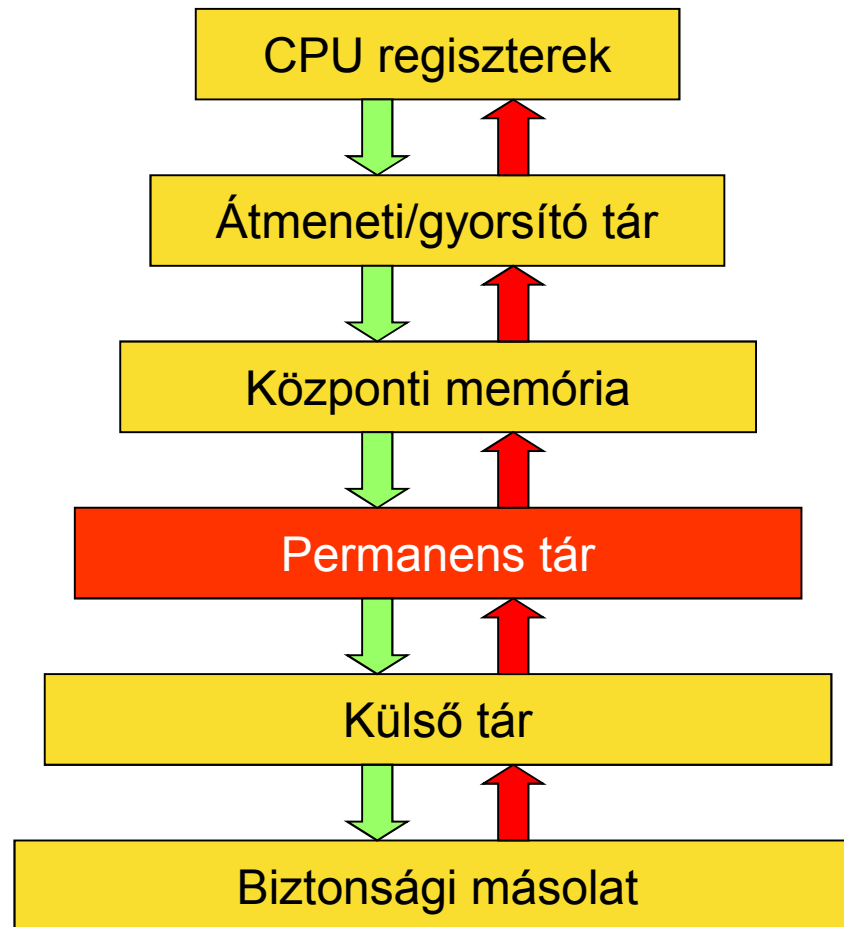
dr. Kovácsházy Tamás

9. anyagrész, A permanens tár kezelése



Méréstechnika és  
Információs Rendszerek  
Tanszék

# Permanens tár



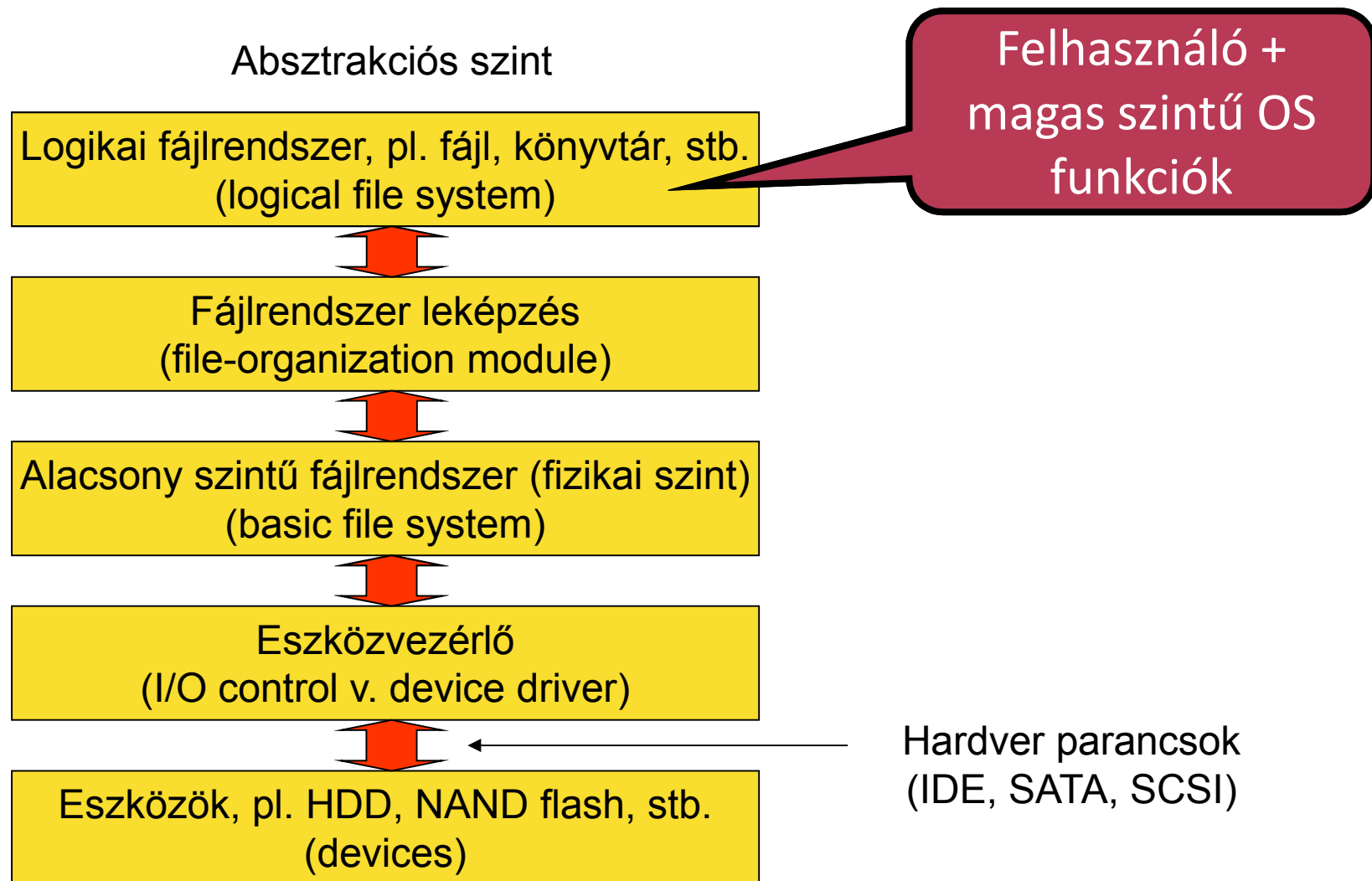
# Permanens tár kezelése

- Permanens tár vagy háttértár:
  - A központi memóriával összehasonlítva
  - Nagyságrendekkel nagyobb tároló terület.
  - Nagyságrendekkel lassabb.
    - Adatátviteli sebesség.
    - Késleltetés.
  - Nem felejtő tárolás.
  - Blokk alapú szervezés.
    - Az OS ebben kezeli, ennél kisebb egységekben nem gondolkozik.
    - Blokként olvasható, írható, törölhető.
    - Kivéve egyes beágyazott rendszereket.
      - NOR flash memória szervezésű.
      - NOR flash-ből direkt módon futtatható az OS és a programok.

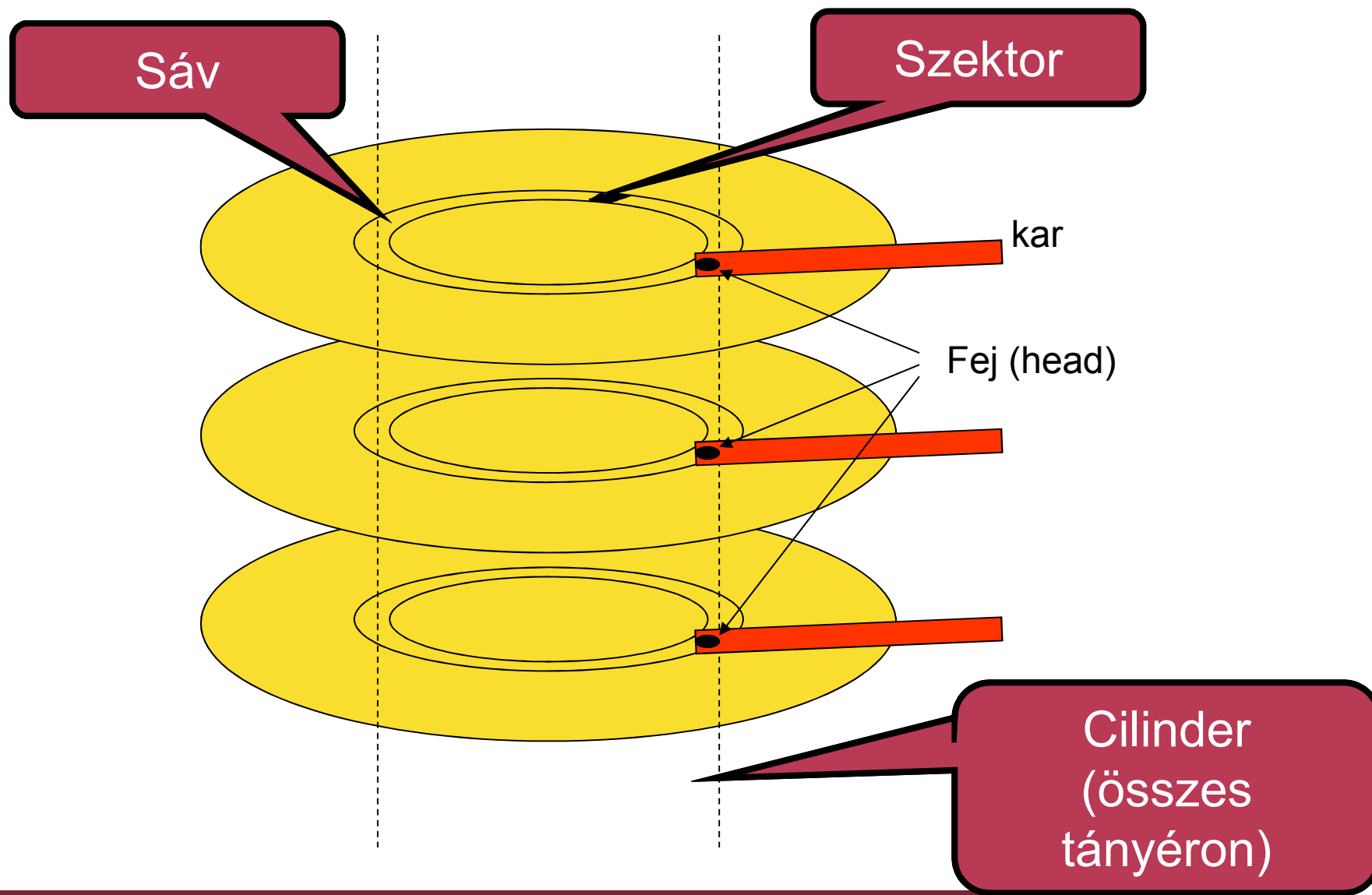
# Fájl leképzése a HW tároló elemre

- A fájl (file) a permanens táron az adattárolás logikai egysége.
  - Névvvel rendelkezik (named collection).
  - A nevével lehet hivatkozni rá.
  - Tetszőleges méretű (fájlrendszer és az eszköz kapacitása a korlát)
- Az operációs rendszer feladata a logikai egységek (fájlok) leképzése valódi fizikai egységekre.
- Ezt az operációs rendszerben egy többszintű hierarchikus/réteges rendszer, különböző absztrakciós szintekkel oldja meg.
- Legalacsonyabb szinten többnyire valamilyen speciális HW van (HDD, Flash drive, stb.).
  - Kivétel az u.n. RAM drive...

# Absztrakciós szintek (egyszerűsített)



# HDD szerkezete



# Eszközök: HDD

- Merevlemez (HDD):
  - Forgó mágnesezhető tányérok (platter).
  - Egy mozgatható karra (arm) vannak szerelve az író/olvasó fejek (head).
  - A tányérokot cilinderekre (cylinder), és azokat sávokra (track) azokat meg szektorokra (sector) osztjuk.
  - A cylinder, sáv és szektor együtt azonosítja az írható/olvasható adatblokkot.
  - A gyakorlatban már a fizikai eszközök egy logikai leképzést alkalmaznak (Logical Block Addressing).
  - LBA: 48 vagy 64 bit napjainkban
  - 512 byte / 4 Kbyte szektor méret áttérés
    - Jelenleg folyik
    - Teljesítmény ingadozik apró implementációs részletek függvényében

# A HDD tényleges sebessége

- Erősen függ attól hogy éppen hol helyezkedik el a fej, és ahhoz képest az elérendő adatblokk (szektor), és milyen sebességgel forognak a tányérok.
- Több szintű optimalizáció:
  - Diszk ütemezés (Disk scheduling).
    - Milyen sorrendben optimális a diszk-hez érkező I/O kérések kiszolgálása?
  - HDD szintjén (SATA NCQ, SCSI).
  - Operációs rendszer szintjén a párhuzamos írások/olvasások ütemezése során.
  - Prefetch...
- Több szintű cache:
  - HDD szintjén (16-64 Mbyte jelenleg).
  - HDD controller cache (drága RAID kontrollereken, tipikusan  $n \cdot 100$  Mbyte)
  - Operációs rendszer szintjén:
    - Disk cache, dinamikusan változó méret.



# Eszközök: NAND Flash tároló

- Az alacsony szintű interface azonos a merevlemezéével.
  - Solid State Disk (SSD) SATA v. IDE interfésszel.
  - PEN drive USB interfésszel.
    - A kártya olvasók ugyan így működnek, csak a tároló cserélhető.
- Olvasás gyors, független az adatot tároló blokk elhelyezkedésétől:
  - Nem kell fejeket mozgatni és szektort pozícióba forgatni.
  - RAM jelleggel érhető el.
- Az írás (valójában a törlés) problémás:
  - Véges számú alkalommal törölhető egy blokk.
  - Az írás/törlés lényegesen lassabb is.
    - Párhuzamosítható több blokk írása gyorsíthat, ha az eszköz támogatja.
  - Wear leveling (eszköz és/vagy OS szintű)
    - Pl. OS szinten speciális fájlrendszerek: JFFS2, YAFFS, UDF (optikai tárolókon), ZFS
  - TRIM (SSD + OS támogatás): Írások csökkentése, és a „write amplification” jelenség elkerülése a cél

# Eszköz csatlakozás

- A hoszt számítógéphez csatlakozó megoldások (Host-Attached Storage):
  - Direkt csatlakozás: SATA/eSATA, IDE, SCSI, SAS, stb.
  - Indirekt csatlakozás:
    - USB, Firewire alapú alagút (tunnel).
    - RAID (Redundant Array of Inexpensive Disks).
- Hálózati tárolók eszközök (Storage-Area Networks, SAN):
  - Hálózati alagút (tunnel) a hoszt és a tároló eszköz között.
  - Speciális protokollok: Fibre channel
  - Ethernet és/vagy TCP/IP alapú: iSCSI, AoE
- A NAS (Network-Attached Storage, File megosztás, stb.) nem ezen a szinten valósul meg.

# USB

- USB mass storage device class.
- SCSI parancs készlet kerül átküldésre az USB buszon transzparens módon.
  - Az OS számára egy SCSI buszon keresztül csatlakozó eszköznek tűnik.
  - Az USB csupán egy alagutat képez az eszköz és az OS között.

# RAID

- **Tények:**
  - A merevlemezek olcsók.
  - Nem megbízhatóak (mozgó alkatrészek, érzékenyek).
  - Lassúak.
- **Ötlet:**
  - Használjunk belőle többet egyszerre.
  - Több redundáns alkalmazása növeli a megbízhatóságot.
  - Több párhuzamos használata növeli a sebességet.
  - Hozzuk létre egy virtuális diszket a fizikai diszkekből.
    - Az OS majd azt (a virtuálisat) kezeli...
    - Ezt a virtuális diszket hívják RAID tömbnek (RAID array)
- **Megvalósítás:**
  - HW RAID vezérlők.
  - SW RAID megoldások.
    - Az alaplapi megoldások is ilyenek szinte kivétel nélkül.
    - Szerver alaplapokban esetleg van HW RAID.

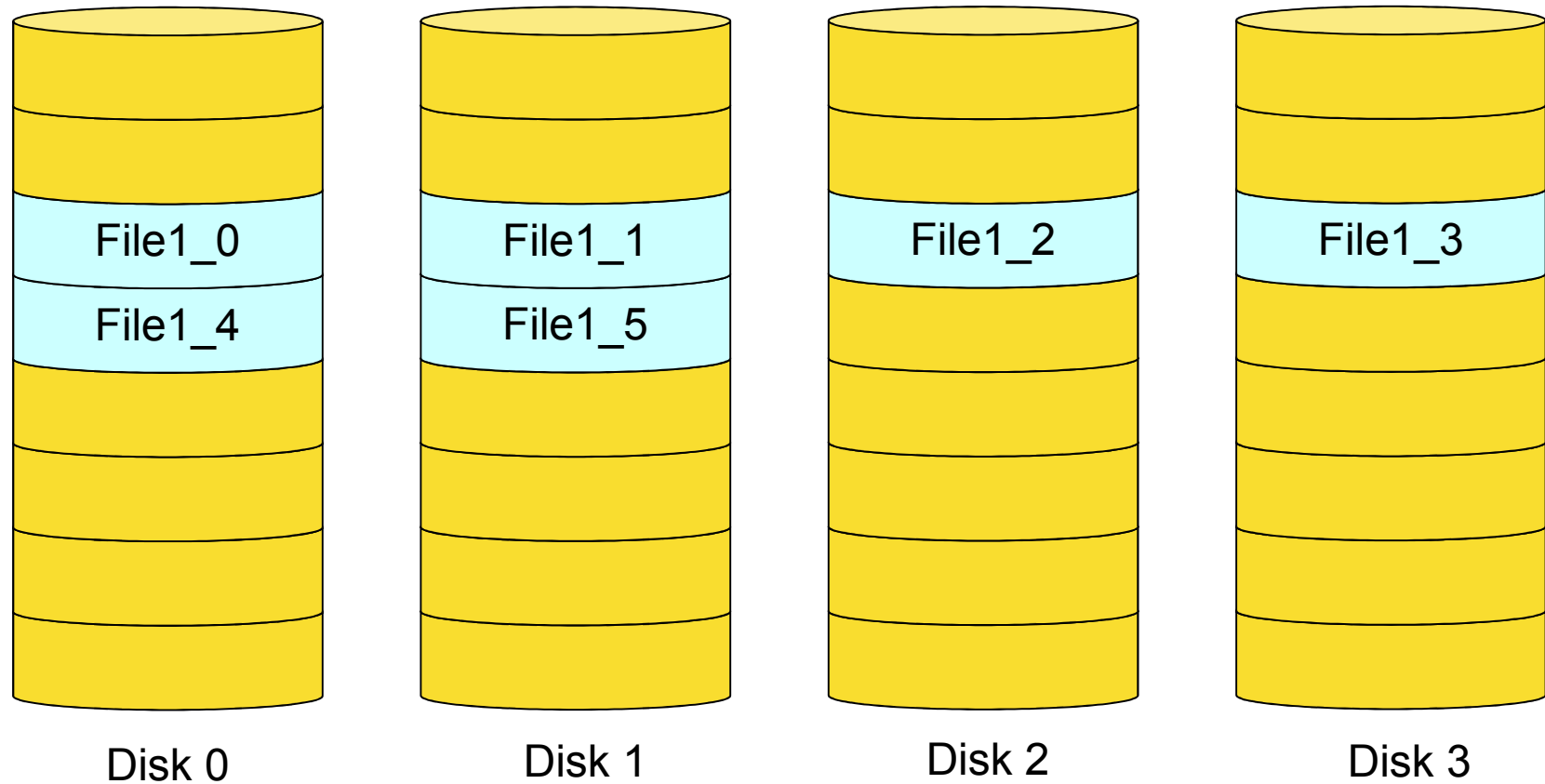
# RAID szintek

- RAID 0-6 és egymásba ágyazott (nested) szintek.
- RAID 0-1 szabványok általában SW implementációval és kevés (2db) diszkkal.
- RAID 2-4 szabványokat ritkán használják.
- RAID 5 és 6 alkalmazása tipikus nagyobb számú diszk esetén.
  - Sok diszk (4 vagy annál több).
- Egymásba ágyazott szintek:
  - RAID 1+0 és RAID 0+1.
- Vannak gyártó specifikus nem szabványos megoldások...

# RAID level 0

- RAID 0 (striped disks).
- Több diszk párhuzamos használata.
- A file részei N diszkre kerülnek:
  - Az egyes részek egymástól függetlenül elérhetők.
    - A diszkek tároló kapacitása összeadódik.
    - N azonos diszk esetén a RAID 0 virtuális diszk olvasásai és írási adatátviteli sebessége maximum N szerez közelébe nő.
    - A hozzáférési idő közel eléri egy diszk hozzáférési idejét.
    - Bármelyik diszk meghibásodása esetén az adat elveszik.

# RAID level 0

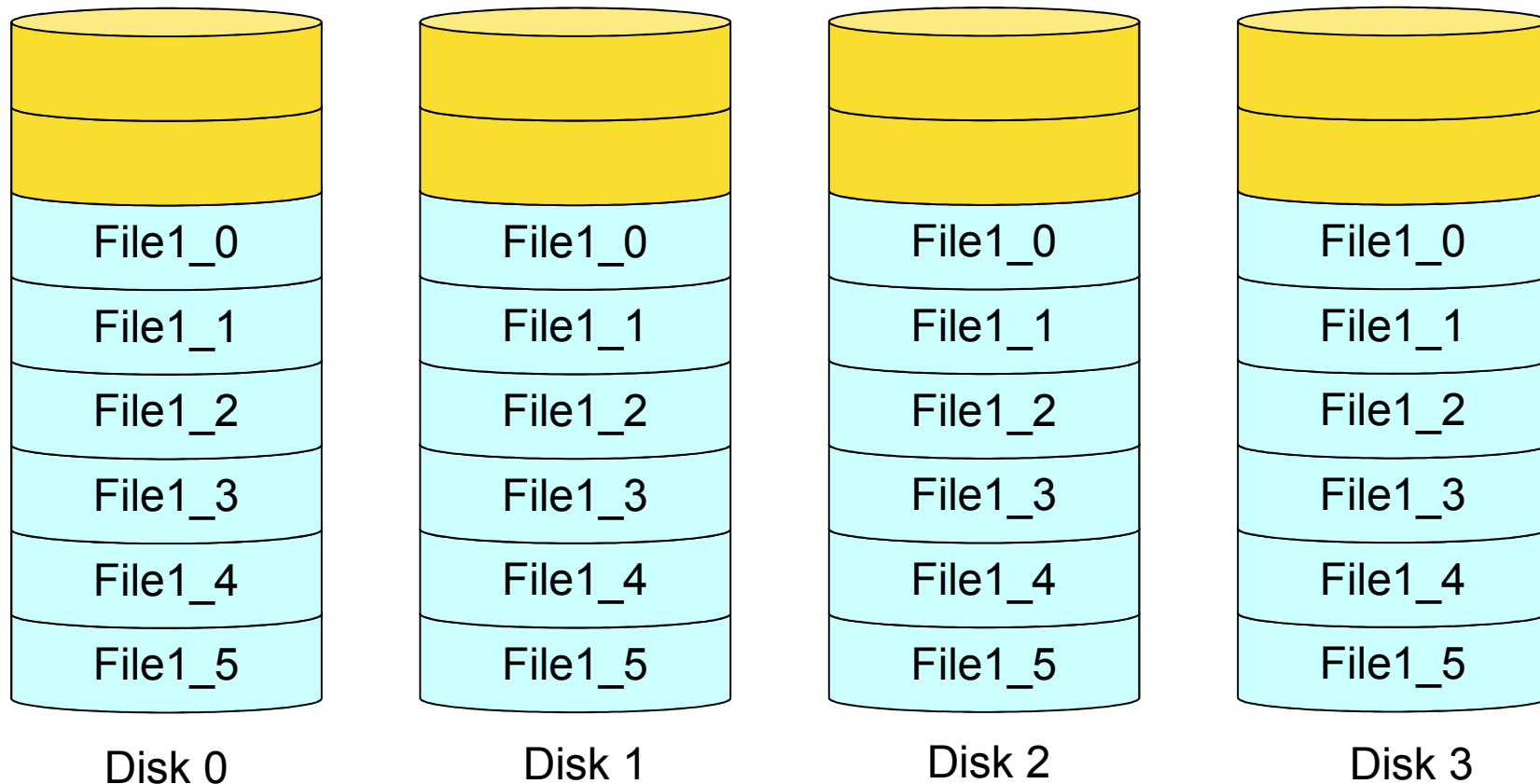


# RAID level 1

- RAID 1 (mirroring).
- Több diszk redundáns használata.
- A file minden része minden (N) diszkre kikerül.
  - Azonos diszkeket feltételezve a tároló terület egy diszk tároló területével azonos.
  - Az adatátviteli sebesség lassabb mint egy diszk sebessége.
  - A hozzáférési idő nő.
  - Speciális esetben az olvasási sebesség N szerezre nőhet.
    - Feltételezve a diszk meghibásodásának más módon történő észlelését (nem kell az azonosságot ellenőrizni többségi szavazással).
  - Egy működőképes diszk esetén az adat elérhető.



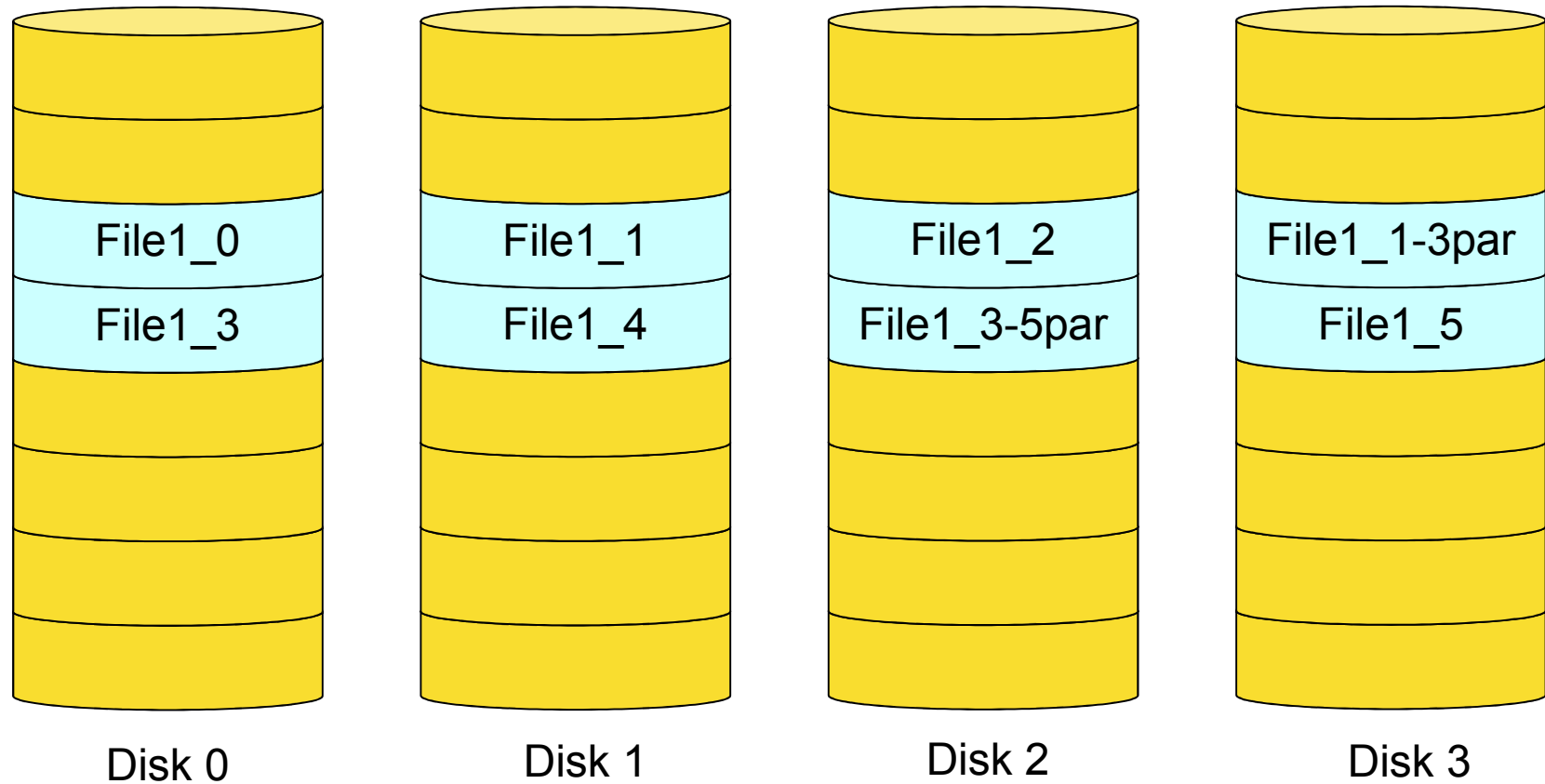
# RAID level 1



# RAID level 5

- RAID 5 (block interleaved distributed parity).
- Több diszk redundáns és párhuzamos használata.
- Adat és paritás elosztása  $N+1$  diszkre.
  - A sebesség tekintetében közel áll az  $N$  diszket használó RAID 0-hoz (HW támogatás esetén).
  - A kapacitása  $N$  diszk tároló kapacitásával egyenlő
  - 1 diszk meghibásodása esetén az adat elérhető.
  - 2 vagy több diszk meghibásodása esetén az adat elveszik.
  - Az adat nem feltétlenül állítható helyre.
    - Csendes/néma hibák (silent error).
    - A 2. meghibásodás észlelése a tömb újraépítése során.

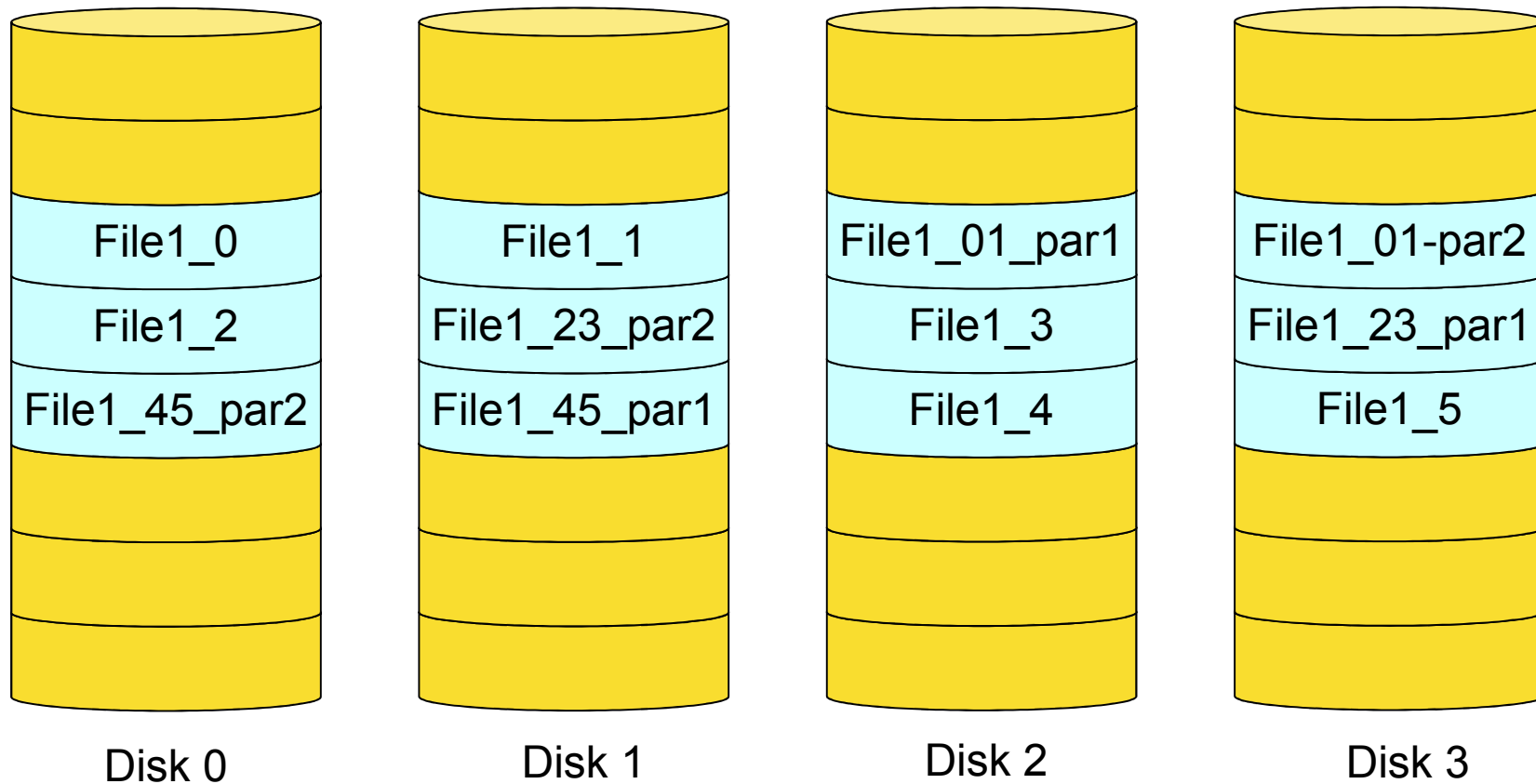
# RAID level 5



# RAID level 6

- RAID 6 (block interleaved dual distributed parity).
- Több diszk redundáns és párhuzamos használata.
- Adat és paritás elosztása  $N+2$  diszkre.
  - A sebesség tekintetében közel áll az  $N$  diszket használó RAID 0-hoz (HW támogatás esetén).
  - A kapacitása  $N$  diszk tároló kapacitásával egyenlő
  - 2 diszk meghibásodása esetén az adat elérhető.
  - 3 vagy több diszk meghibásodása esetén az adat elveszik.
  - Az adat nagyobb valószínűséggel állítható helyre a RAID 5-höz képest.
    - Ha 1 diszk meghibásodása esetén azt azonnal pótoljuk, a tömb helyreállítása során egy csendes/rejtett hiba is javítható
    - Azonnal javítani kell egy hiba esetén!

# RAID level 6



# RAID kritikája

- Hamis biztonságérzet:
  - Csak a merevlemez egyedi, véletlen meghibásodása ellen véd.
    - Pl. A tápegység meghibásodik, és tönkreteszi a HDD-éket is.
  - Nem véd az SW hibáktól, illetéktelen hozzáféréstől, stb.
  - Nem pótolja a biztonsági másolatokat, csak a rendelkezésre állási időt és a sebességet növelheti.
- A HW RAID vezérlők drágák:
  - 8 portos SATA RAID RAID 5 és RAID 6 támogatással kb. 200 eFt.
  - Drágább, mint a hozzá csatlakozó diszkek.
  - Komplet gépet lehet venni ekkora összegből...
- Az SW RAID megoldások elsősorban RAID 0 és RAID 1 esetén alkalmazhatóak.
  - Lassú a RAID 5 és 6 bonyolult kódolásának SW megvalósítása.

# RAID előnyei

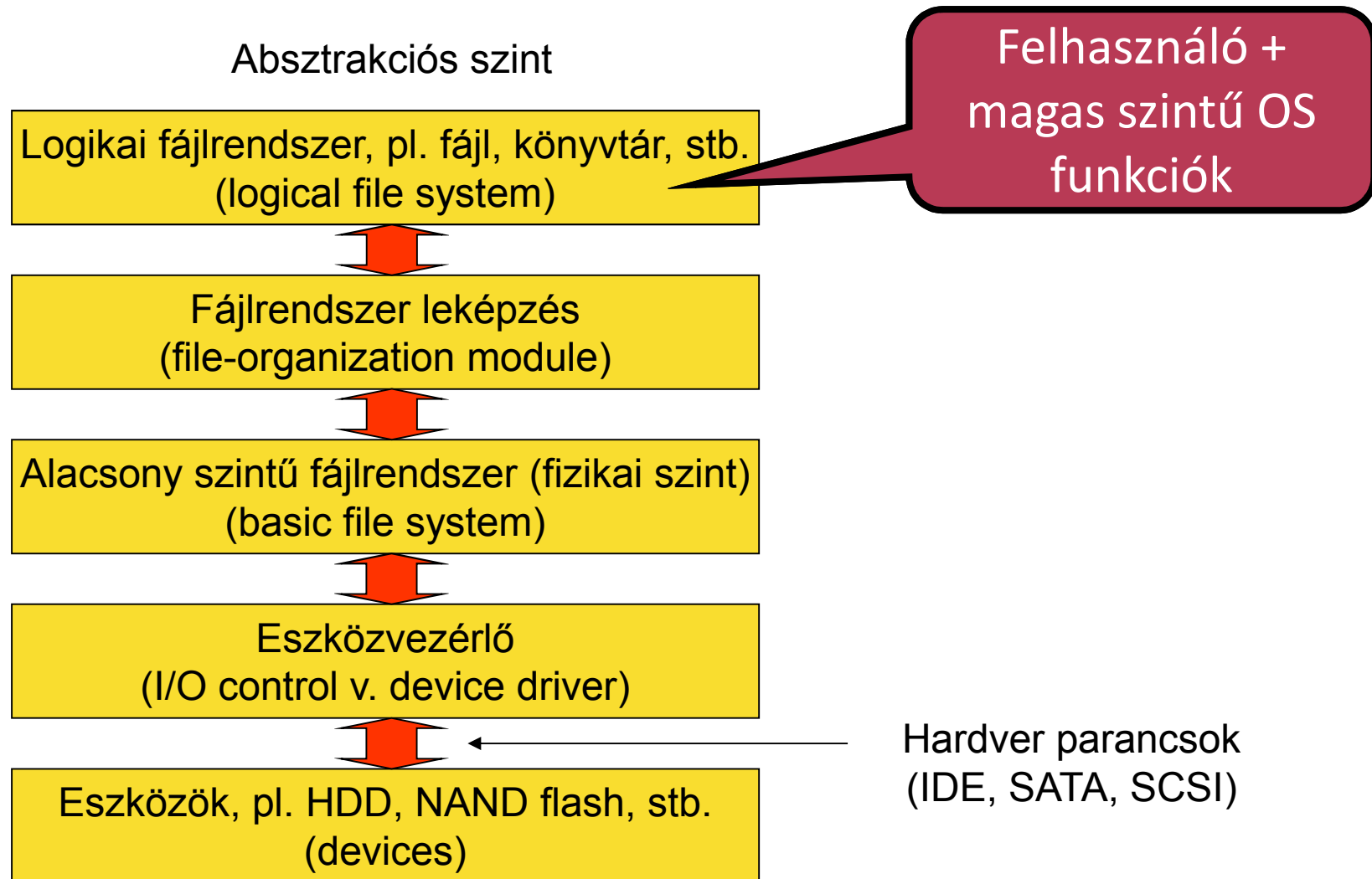
- A RAID 1 és RAID 5, 6 megvéd a tipikus véletlen HDD hibák által okozott azonnali rendszer leállástól.
  - A HDD a leggyengébb láncszem:
    - Pl. Google HDD statisztikák.
    - [http://labs.google.com/papers/disk\\_failures.pdf](http://labs.google.com/papers/disk_failures.pdf)
    - Hőmérséklet nem befolyásolja a működést olyan mértékben, mint korábban gondolták.
    - SMART (Self-Monitoring, Analysis, and Reporting Technology) hatásos a hibák előrejelzésére, de nem jelzi biztosan előre a hibákat.
    - Ki használja a SMART-ot? (Mindenki próbálja ki.)
      - Smartmontools + GSmartControl vagy HDD Guardian
  - CD-ROM-os történet (nem csak a HDD lehet hibás)...
- A RAID 0 és RAID 5, 6 gyorsítja a diszk hozzáférést.
  - A HDD a leggyengébb láncszem ebből a szempontból is.

# Hálózati tároló eszközök (SAN)

- Hálózati „tunnel” a hoszt és a tároló eszköz között.
  - Alacsony, blokk szintű megoldás.
  - Többnyire SCSI parancsokat küldenek át.
  - A tároló eszköz virtualizációja:
    - Teljes értékű, tetszőlegesen skálázható, partícionálható, bootolható, stb.
    - Mintha lokálisan lenne a merevlemez csatlakoztatva.
    - Általában egy géphez csatlakoztatható csak (kivéve fürtök/clusters).
- Megoldások:
  - Speciális protokollok: Fibre channel (drága, dedikált HW).
  - Ethernet és/vagy TCP/IP alapú: iSCSI, AoE.
    - Olcsó/ingyenes, részben SW, de legalább firmware támogatás is szükséges (pl. bootolás iSCSI-ről BIOS-ból, lásd Intel hálózati kártyák).
- Konvenciók:
  - Target: Hálózati tároló eszköz, amihez a fizikai tároló eszközök direkt módon vagy további SAN szinteken keresztül csatlakoztatva vannak.
  - Initiator: A kliens, ami használja a tároló eszközöket.
  - Elnevezési konvenciók a használandó tároló eszköz azonosítására.
  - Initiator szintű hozzáférés ellenőrzés.



# Absztrakciós szintek (egyszerűsített)



# Alacsony szintű fájlrendszer

- Fizikai diszk blokkok írását és olvasását végzi.
- Egyben feladata a működés során használt információk cache-elése is.
  - Puffer cache (buffer cache):
    - A lapozás során külön cache szolgál a pagefile cache-elésére, és külön a blokkok cache-elésére (beleértve a pagefile cache tartalmát is).
    - Kétszeres cache-elés.
  - Egységes puffer cache (unified buffer cache):
    - A cache a blokk szinten működik, nincs külön pagefile cache.
  - Egységes virtuális memória (unified virtual memory):
    - A lapozás és a fájlrendszer szinten megvalósított, operációs rendszer egészére vonatkozó disk cache összevonása.
    - A file lényegében virtuális memóriára van leképezve
    - Pl. Linux, Windows, stb. ezt használja.
    - Pl. Linux: Amíg van szabad fizikai memória, abba a file-okat cache-eli.

# Fájrendszer leképzés

- Logikai blokkok leképzése fizikai blokkokra (allocation).
  - Folytonos allokáció (contiguous allocation).
  - Láncolt listás allokáció (linked allocation).
  - Indexelt tárolás (indexed allocation).
- Az üres helyek menedzselése (free-space management):
  - Bit vektor (bit vector).
  - Láncolt lista (linked list).
  - Szabad helyek csoportjaink listája (Grouping).
  - Számlálás (Counting).
  - Egybefüggő szabad területek nyilvántartása (Space maps).
  - Az üres helyekkel nem fogunk foglalkozni a tárgyban...

# Folytonos allokáció (contiguous allocation) 1.

- A fájl egy folytonos fizikai blokk sorozatot foglal el.
  - Hozzáférés egyszerű és gyors HDD esetén (a sáv folytonosan olvasható a forgó tányérról).
  - Növekvő méretű fájloknak a helyfoglalás problémás:
    - Milyen méretű szabad helyet allokáljunk?
  - Új fájlok számára megfelelő szabad hely megtalálása nehéz, külső tördelődés lép fel.
    - Fájl törlése után a méretének megfelelő számú blokk felszabadul.
    - Erre a helyre kisebb vagy egyenlő méretű fájl írható.
      - Ugyan azokat az algoritmusokat használhatjuk, mint a memória foglalás során (first fit, next fit, best fit, worst fit).

## Folytonos allokáció (contiguous allocation) 2.

- **Növekvő fájlok:**
  - A best fit allokációs stratégia különösen veszélyes.
  - A fájl nagyobb szabad helyre másolása (erőforrás igényes).
- **Külső tördelődés csökkentése:**
  - Teljes másolás egy üres diszkre majd vissza (off-line).
    - Rendszerleállással jár.
    - Hosszú ideig tart és erőforrás igényes.
  - Futási idejű (on-line) töredezettség csökkentés (defragmentation).

# Láncolt listás allokáció (linked allocation) 1.

- A könyvtárakat leíró adatstruktúrák tartalmazzák az első és az utolsó blokk azonosítóját.
- Minden blokk tartalmazza a következő blokk azonosítóját
- A fájlhoz tartozó blokkok tetszőleges helyen lehetnek a diszken.
- Nincs külső töredezettség.
- Problémák:
  - Szekvenciális fájl elérésre alkalmas, a fájlba indexelni viszont nehéz (n. blokk direkt elérése).
  - A blokkokban lévő azonosítók helyet foglalnak.
  - Sérülékeny (azonosítók fűzik a blokkokat össze).
  - Sok fejmozgást okoz (seek), ha a blokkok el vannak szórva a diszken.

# Láncolt listás allokáció (linked allocation) 2.

- Pl. a FAT fájlrendszer ezt használja.
- Töredezettség mentesség ebben az esetben mást jelent:
  - Cél a fejmozgás minimalizálása egy file olvasása során.
    - SSD esetén értelmetlen (nincs fej, sok írás)
  - A fájlok egymás utáni blokkokon történő tárolását tűzi ki célul.
  - Ezt is defragmentation-nak hívják...
  - Ilyen célból ajánlott időnként pl. a Windows operációs rendszerek alatt a töredezettség mentesítő programok futtatása.
    - Jelentősen nőhet a fájlkezelés sebessége.
    - Az írást is gyorsítja, nem csak az olvasást!
      - Összefüggő szabad helyek...

# Indexelt tárolás (indexed allocation).

- Index blokkok használata:
  - Egyes blokkokat fájlhoz tartozó indexek tárolására allokálunk.
- Szekvenciális és indexelt elérésre is alkalmas.
- Sérülékeny (az index blokkok sérülése a fájlt elérhetetlenné teszi).
  - Az index blokkokat viszont könnyű többszörözni (replikálni).
- Sok fejmozgást okoz (seek), ha a blokkok el vannak szórva a diszken.
  - Itt is lehet a láncolt listás töredezettség mentesítéshez hasonló algoritmusokat használni a fejmozgás minimalizálására.



# Logikai fájlrendszer

- Operációs rendszer specifikus.
  - Operációs rendszer specifikus API a tetején.
  - Tipikus API függvények:
    - Create, Delete, Read, Write, Set/Get attributes, stb.
- Metaadat tárolása (minden, kivéve a tényleges adatot).
- Fájlok:
  - Absztrakt adattípus (objektum, fájl mutató).
  - Adat, név (name), típus (type) tulajdonságok (attributes).
  - Kölcsönös kizárás (file locking).
- Könyvtárak (Directory/Folder).
- Kötetek (Volume/Drive).

# Fájl

- A fájl (file) a permanens táron az adattárolás logikai egysége.
- Tulajdonságok:
  - Név:
    - Elnevezési konvenciók (Windows/UNIX).
    - Egyedi azonosító egy könyvtáron belül?
      - Többnyire igen, de bizonyos OS-ekben nem...
  - Típus (megadja a kezelés módját):
    - Pl. Windows kiterjesztés, vagy egy speciális tulajdonság.
  - Tulajdonosok, jogosultságok.
  - Hozzáférési időpontok.
    - Pl. Létrehozás, módosítás (írás), hozzáférés (írás/olvasás).
  - stb.

# Könyvtárak

- Az információ hierarchikus tárolása...
- Kialakítások:
  - Egyszintű, korai rendszerekben használták.
  - Kétszintű, még az 1990-es évek közepén is használták (pl. IBM OS/400).
  - Fa struktúrájú fájlrendszer.
  - Aciklikus irányított gráf struktúrájú fájlrendszer.
  - Általános gráf alapú fájlrendszer.

# Aciklikus irányított gráf struktúra

- Egy fájl vagy alkönyvtár több könyvtárban található meg.
- Csak egy példányban létezik!!!
- Pl. UNIX/Linux hard és symbolic links
  - Egy fájlhoz több úton el lehet jutni.
  - A „hard vagy symbolic link” jellegű kötés felismerhető fájl attribútumok alapján.
    - A fájlrendszeren végig lehet iterálni (felismerhetők a linkek)...
  - Mi történjen, ha törölünk egy fájlt, amelyhez több úton el lehet jutni?
    - Csak a hivatkozás törlődik, ha azt töröljük.
    - Nem törlődik, amíg az összes hivatkozást nem törlik.
    - Törlődik minden hivatkozással együtt.

# Általános gráf struktúra

- Operációs rendszerekben ritkán használják.
  - Hogyan lehet benne pl. keresni?
  - Mi legyen a kereső algoritmus leállási feltétele?
  - A WEB-es dokumentumok pl. egy ilyen struktúrát valósítanak meg, amikor egymásra hivatkoznak.
    - Bár az nem egy tényleges fájlrendszer...

# Kötetek

- A logikai fájlrendszer szintjén a legmagasabb egység.
- Megfelel egy fizikai vagy logikai partíciónak a fizikai tároló eszközön.
- Hogyan jelennek meg az operációs rendszerben?
  - Nevükön kell rá hivatkozni (pl. Windows C:)
  - A fájlrendszerben a könyvtár hierarchiába tetszőleges helyre leképezhető (UNIX/Linux mount, újabb Windows-ok)

# Adatszerkezetek az eszközön

- Az alacsony szintű adatstruktúrák.
  - Boot szektor (boot control block).
    - Ezt tölti be a BIOS/EFI/firmware, és utána ez alapján történik az OS betöltése.
  - Partíciós tábla (volume control block).
    - A partíció specifikus adatokat itt tárolják.
    - Partíció méretei, használt és szabad hely, azokra mutató referenciák, stb.
  - Fájrendszer specifikus információ.
    - Könyvtárstruktúra leírói.
    - Fájl leírók (File Control Block).

# Adatvesztés...

- A fájlok egy időben a memóriában és a permanens tárolóeszközön is jelen vannak.
  - Eltérő állapotban lehetnek:
    - Pl. a memóriában már új adatelemekkel...
  - A metadatokat, az allokációs adatstruktúrákat is módosítjuk.
  - Meghibásodás vagy a tápfeszültség elvesztése inkonzisztenciát okozhat.
- Konzisztencia ellenőrzése.
- A konzisztencia visszaállítása.
  - Tranzakció orientált fájlrendszerek.
    - Log-structured, log-based transaction oriented, journaling fájlrendszerek.
    - Pl. NTFS, EXT3 és EXT4.
    - NEM a biztonság a cél itt (adatvesztés lehet), a cél a konzisztencia!
- Biztonságos rendszerleállítás, szünetmentes táplálás (UPS).
- Adatmentés és visszaállítás.
  - A mentésből a helyreállítást tesztelni kell.
  - Amíg nincs sikeres helyreállítási teszt, nem beszélhetünk adatbiztonságról!



# Széles körben elterjedt fájlrendszerek 1.

- FAT (File Allocation Table)
  - 8+3 karakteres fájlnev, a hosszú fájlnev külön fájlba tárolva...
  - FAT16 (előtte volt FAT12 is).
    - Max. 2GB partíció méret.
    - 32767 könyvtár bejegyzés.
    - Ma is használják (Pendrive v. kisebb memória kártya).
  - FAT32
    - 2 TByte (TiB SI rendszerben) partíció méret.
      - Más okokból van egy 64 vagy 128 GByte-os limit egyes korai MS operációs rendszerekben.
    - File méret: 4 GByte-1 byte
      - Sokan nem tudják, és nem értik, miért nem tudnak nagy file-okat hordozható diszkekre másolni...
- NTFS (New Technology File System)
  - $2^{64}$  Byte (16 EB) - 1 KByte max. fájl méret,  $2^{32}-1$  fájl, stb.
  - $2^{64}$  darab szektor egy partíción belül.
  - 256 karakter hosszú fájlnev.
  - Tranzakció alapú.
  - Töredezés mentesítés ennél is szükséges.

# Széles körben elterjedt fájlrendszerek 1.

## ■ FAT (File Allocation Table)

- 8+3 karakteres fájlnev, a hosszú fájlnevek miatt
- FAT16 (előtte volt FAT12 is).
  - Max. 2GB partíció méret.
  - 32767 könyvtár bejegyzés.
  - Ma is használják (Pendrive v. kisebb lemezek).
- FAT32
  - 2 TByte (TiB SI rendszerben) partíció méret.
    - Más okokból van egy 64 vagy 128 KiB-os fájl méretkorlát a FAT32 rendszerekben.
  - File méret: 4 GByte-1 byte
    - Sokan nem tudják, és nem értik, miért nem tudnak nagy file-okat hordozható diszkekre másolni...

A FAT és NTFS fájlrendszerk nem különböztetik meg a kis- és nagybetűket.  
(Erre az átlag felhasználók sem képesek...)

## ■ NTFS (New Technology File System)

- $2^{64}$  Byte (16 EB) - 1 KByte max. fájl méret  $2^{32}-1$  fájl, stb.
- $2^{64}$ . darab szektor egy partíción belül.
- 256 karakter hosszú fájlnev.
- Tranzakció alapú.
- Töredezés mentesítés ennél is szükséges.

# Széles körben elterjedt fájlrendszerek 2.

- EXT2
  - Alapértelmezett Linux fájlrendszer korábban.
  - Van Windows driver is, lehet Windows alatt is használni.
    - Rendszerpartíción nem.
  - Maximális fájlméret: 16 GByte - 2 TByte (blokk mérettől függ).
  - Fájlok maximális száma:  $10^{18}$ .
  - Maximális fájlnev hossza: 255 byte (kis- és nagybetű érzékeny).
  - Maximális partíció méret: 2-32 TB (Linux kerneltől függő).
  - A töredezés lassan történik, ritkán van szükség töredezés mentesítésre (off-line lehetséges).
- EXT3
  - Az EXT2 javított, tranzakció kezeléssel kiegészített verziói.
  - Htree alapú indexelés: több könyvtárat tesz lehetővé
  - Ez a javasolt Linus fájlrendszer, kivéve a flash eszközöket.
  - EXT2 és EXT3 között egyszerű a konverzió (visszafelé is).
- EXT4: További bővítések (nagyobb táruk kezelése, extents, stb.).
  - Egyre gyakrabban már ezt telepítik alapértelmezett módon a disztribúciók.
- CD-ROM/DVD fájlrendszerek (ISO 9660, Rock Ridge, Joliet, El Torito kiterjesztések)
  - Máig is 2/4 Gbyte a fájlméret korlát (DVD ezért van szétszedve darabokra).

# NAS (Network-Attached Storage)

- Fájrendszer szintű hálózati fájlmegosztás.
  - Többnyire nyomtató is...
- Például:
  - Network File System (NFS).
    - Elsősorban UNIX, de van Windows megvalósítás is.
  - Server Message Block / Common Internet File System (SMB/CIFS)
    - Elsősorban Windows, de Linux/UNIX is (SAMBA).
- Fájrendszer szintű megosztás.
  - A hálózaton a könyvtárakra és fájlokra vonatkozó utasításokat küldünk át.
  - Jellegzetesen párhuzamosan több felhasználó érheti el.
  - Felhasználó szintű jogosultságok.
  - A kliens és a szerver fájlkezelési konvenciónak eltéréséből problémák lehetnek.
    - Pl. UNIX és Windows fájlnev, fájl tulajdonságok, stb.
- A HTTP nem ilyen, komplett fájl osztható meg, nem lehet indexelni a fájlba.
  - Fájl elérésnek is hívják

# Permanens tár megfigyelése

- Alacsony (blokk) szinten
  - Sysinternals: Disk Monitor (diskmon.exe)
- Magas (fájl) szinten
  - Sysinternals: Process Monitor (procmon.exe)