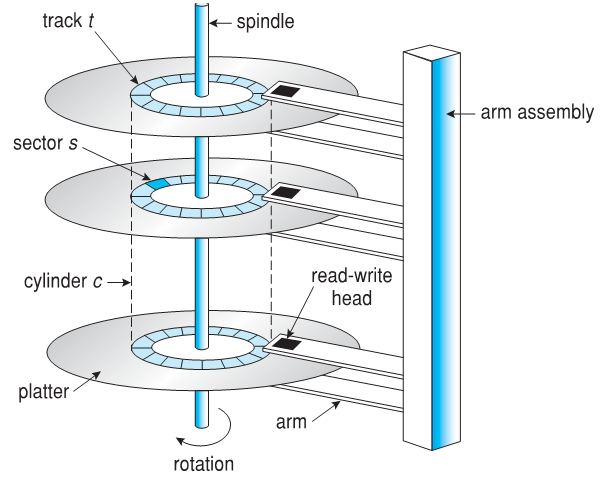
**Chapter 9**

Mass Storage System/Tárolórendszer – 11. előadás

Alapvetések

* Neumann elv 🡪 Memóriába kell rakni valamit, ha KI AKARJUK VÉGEZNI, de ez adat nem jó ha elveszik ha nincs áram 🡪 Háttértárak megjelenése

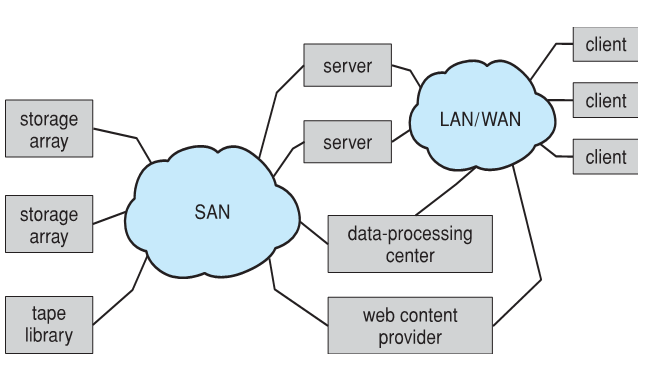
Háttértártípusok

* Mágnesszalag
  + Lassú, de ár/kapacitás és megbízhatóság szempontjából nagyon jó
  + Szekvenciális olvasás
* Merevlemez / Hard Disk Drives
  + Író – Olvasó fej 🡪 Nem ér hozzá a tányérhoz (jobb esetben)
    - Mozgó alkatrész 🡪 Not good, nem jó rugdosni
    - Fejmozgatási stratégia fontos lehet
  + Tányérok 🡪 Sávok 🡪 Szektorok (min 512 byte)
  + RPM 4200-15000
  + Kapacitás 500Gb – 10+TB
  + Örökéletre megőrzi, ha nem zavarja be valami mágneses tér
  + Könnyen eltávolíthatóak
  + Az alaplap kettéválasztja a bővítő eszközt az I/O-tól
    - PCI bus-t ezért nem használják
  + Seek time 🡪 3 ms – 12 ms
  + SATA-t használt
* EIDE/ATA, SATA, USB
  + - ATA – Advanced Technology Attachment
    - EIDE – Electronic integrated device
    - SATA – Sequential ATA Elméleti max sebesség 🡪 6 Gbit🡪 Gyakorlatban 1 Gbit
    - USB – Universal Serial Bus
    - Ezek mind azt írják le, hogy kerül át az adott adat a háttértárról a memóriába
    - Lehetséges probléma 🡪 Késlekedési idő 🡪 Lemez ütemezési stratégia
* Solid State Drive /SSD
  + Non-Volatile memory used like a hard-drive
  + Megbízhatóbb, mint a HDD
  + Sokkal drágább
  + Kisebb élettartam 🡪 A sok írást nem szereti
  + Kisebb kapacitás
  + Sokkal gyorsabb, mint egy HHD
  + PCI bus-hoz csatlakozik, hogy a sebességet ki tudja használni
  + Nincs mozgó alkatrész, nincs seek time

Háttértár struktúra

* Szektor alapú 🡪 Egymás után sok szektor
  + 1 szektor 512 Byte
  + Szektorok sorszáma 0.-tól kezdődik
* Amikor formázunk egy tárolót, szektorokból logikai blokkot készítünk
  + NFTS 🡪 Egy blokk lehet 4096 byte 🡪 8 szektor
  + Tárterületvesztés lehet 🡪 1 byte-os file elfoglal 4096 byte-ot

Tároló alhálózat / Storage Area Network SAN

* Elmaszkolja, hogy mi és miért látszódik
* Hálózaton kersztüli elérés 🡪 Gyorsabb mint az I/O
* Nagyon gyors, nincs adatvesztés
* Nagyon drága , I mean NAGYON

Lemezütemezési Stratégiák

* Lokálisra törekszünk, de lehet bárhol 🡪 Mi most lokálissal foglalkozunk
  + A kérdésre, hogy muszáj-e, hogy ez lokális legyen, mindig nem a válasz
* Író-Olvasó fej mozgatása, csesztetése
* HDD-n ütemezési stratégiák
  + FCFS First Come First Served
    - Lassú
  + SSTF Shortest Seek Time First
    - Sokkal jobb, gyorsabb
  + SCAN – Természetes legyező mozgás
    - Csökkenti a szükséges fejmozgásokat
    - Teljesen elmegy mindkét oldalig 🡪 Felesleges
  + C-SCAN
    - Mint a Round Robin 🡪 Folyamatos legyező mozgás körbe-körbe
  + C-LOOK
    - C-SCAN, de kiveszi a felesleges mozdulatokat, nem megy ki a legszélére, ha nem kell

|  |  |
| --- | --- |
| FCFS | SSTF |
| SCAN | C-SCAN |
| C-LOOK |

Melyik a legjobb lemezütemezési stratégia?

* Nincs legjobb, szituációfüggő
* Leggyakrabban az SSTF az optimális
* SCAN / C-SCAN 🡪 Heves I/O műveleteknél jó
* Fájlrendszerhez hasonlóan, minden használathoz megvan mi az ideális

Swap – Space Management

* Linuxon 🡪 Magától csinál egy Swap partíciót ahhoz megfelelő fájlrendszerrel
  + Lehet állítani
  + Really good
* Windows
  + Gyökérbe berakunk egy Swap fájlt
  + Mi a fasz
  + Lol

Raid Struktúra

* Miért van rá szükség?
  + 1 db disk a hardver bus maximum 50%-át használja ki
    - „50% ott van parlagon”
  + Erre a megoldás a Raid Tömbök
* Mit jelent a Raid struktúra?
  + Több disk összekötve, hogy maximálisan kihasználjuk a sávszélességet
  + 2 disk = 2\* 50% =100% 🡪 We likey
* RAID típusok
  + RAID 0 – Nem redundáns csíkozás
    - Nagyon gyors, a megbízhatóság oltárán
    - Felcsíkozza az adatokat 🡪 Egyik csík egyik diskre, másik másikra
      * Ha elszáll az egyik elszáll az egész
    - Nem redundáns, minden adat csak egyszer van meg
      * Adatvesztés veszélye fennáll
  + RAID 1 – Mirrored Disks
    - Nagyon megbízható, a sebesség oltárán
      * Minden másolást effektíve kétszer végez el mindkét diskre
    - Minden adat kétszer van meg
      * Ha elszáll az egyik, ott a másik, no problem
  + RAID 2,3,4,5
    - Hibrid megoldások
    - Csíkozott, és redundáns, stb