

Fábián Zoltán – Hálózatok elmélet

Operációs rendszerek

A számítógépek története

- ! Neumann elv
- ! Első generációs számítógépek
 - § Elektroncsövek
- ! Második generációs számítógépek
 - § Tranzisztorok, félvezetők
- ! Harmadik generációs számítógépek
 - § Integrált áramkörök
- ! Negyedik generációs számítógépek
 - § VLSI áramkörök, processzor

A személyi számítógépek története

- ! A játékszámítógépek megjelenése
- ! IBM – a PC megjelenése
 - § Intel – processzor
 - § Microsoft – operációs rendszer
 - § A többi IBM
- ! Apple
 - § Macintosh – ablakozós, egerészős

Az operációs rendszerek korszakai

- ! Open Shop számítógéphasználat
 - § Betöltő program
 - § Monitor program (betöltés és a futás állapotát figyeli)
- ! Closed Shop számítógép használat
 - § Operátor végez minden munkát
 - § (60-as évektől)
 - § Köteget (batch) programvégrehajtás
- ! Terminál alapú számítógépes használat
 - Multitaskos gépek
 - Multiuser
- ! Interaktív rendszerek
 - § A felhasználó a géppel párbeszéd formájában dolgozik – az összes mai Pc kategóriájú rendszer
- ! Valós idejű rendszerek
 - § Folyamatirányítás, vezérlés (repülőgép, Autó fedélzeti computere)

Processzor belső felépítése

- ! ALU – Arithmetical, logical unit
 - § logikai kifejezések kiértékelése,
 - § egész számokkal végzett műveletek
- ! Utasításvégrehajtó
- ! Perifériakezelő
- ! Cache – gyors memória
- ! Regiszter – olyan memória, ami magában a processzorban található
 - § 8, 16, 32, v 64 bit információ helyezkedhet el benne
- ! Lebegőpontos számolási egység

Processzorok gyorsítása

- ! Órajel növelése – korlátok valahol 3-4 GHZ körül (hőtermelődés miatt)
- ! Az adatbusz növelése (8 bit, 16 bit, 32 bit, 64 bit)
- ! Pipeline – Az utasítások több lépcsőben hajtódnak végre
- ! Párhuzamos végrehajtás => Utasítás előfeldolgozó => több pipeline hajtja végre az utasításokat
- ! Hyperthreading – Egy processzor látszólag több magnak látszik
- ! Több magos processzorok – A magok egymástól függetlenül hajtják végre egy-egy program utasításait

Processzor

- ! CISC – komplex utasításkészlet
 - § Sokféle gépi kódú utasítás
 - § Az utasítások változó processzor ciklus alatt futnak le
 - § Kevés regiszter
 - § A fordítóprogram egyszerűbb lehet
- ! RISC – Egyszerűsített utasításkészlet
 - § Kevés gépi kódú utasítás
 - § Minden utasítás ugyanannyi processzorciklus alatt fut le
 - § Sok processzor regiszter használata
 - § A fordítóprogram bonyolultabb

Perifériakezelés

- ! Spooling – ciklikusan lekérdezi a processzor a perifériákat. Pl. Win esetén a nyomtató alrendszer
- ! IRQ – Megszakítások használata - a periféria utasítja a processzort, hogy adatokat cseréljen vele. Egér, billentyűzet
- ! DMA – Direct Memory Access – VGA, HDD

Multiprogramozás

- ! Multitask – Egy időben több folyamat fut
- ! Multiuser – Egy időben több felhasználó használhatja a rendszert
- ! Folyamat – processz – Olyan program, ami megkapta a memóriában a vezérlést
- ! Processzek jellemzői
 - § PID – Processz azonosító
 - § Memória – Az a memóriarész, amit menet közben foglal

Erőforrások

- ! Megosztható erőforrás – több folyamat használja egy időben
- ! Nem megosztható erőforrás – Kizárólagosan használja egy időben az erőforrást egy folyamat – pl-. Nyomtató
- ! Erőforrás használata
 - § Le kell foglalni az erőforrást.
 - Statikus – Egyszer foglalja le az erőforrást
 - Dinamikus – Menet közben szükség szerint foglalja le az erőforrást
 - § Mikor már nincsen szükség az erőforrásra => felszabadítja az erőforrást
- ! Kiemelt erőforrások
 - § Minden processz használja – memória, processzoridő, háttértár
- ! Preemptív erőforrás – amikor egyik folyamat elveheti a másiktól az erőforrást
- ! Non-preemptív – amikor nem elvehető az erőforrás a másiktól
- ! Az erőforrásokat az erőforrás kezelő alkalmazások kezelik (resource manager)

Holtpont

- ! Olyan helyzet, amikor a folyamatok más folyamatok által lefoglalt erőforrásokra kölcsönösen várnak. Csak akkor futnak tovább a folyamatok, ha a szükséges erőforrások felszabadulnak.
- ! Kialakulásának feltételei
 - § Kizárólagosan lefoglalható erőforrások
 - § Nincsen automatikus felszabadítás
 - § Kölcsönös kizárás

Holtponi helyzet megoldásai

- ! Ha már kialakult a holtponi
 - § Kilőjük az egyik alkalmazást – Melyiket? Több stratégia létezik (legkevesebb fájl, legrégebben fut, legkevesebbet futott, sok erőforrást foglal, stb...)
- ! Holtponi megelőzése
 - § One Shot allocation – a futás kezdetekor minden szükséges erőforrást lefoglalunk
 - § Precedencia a nem elvehető erőforrások esetén – Kategóriákba osztjuk őket => Egy kategóriából csak egyszer lehet foglalni, utána már csak magasabb szintű erőforrást lehet foglalni
 - § Bankár algoritmus – amikor az alkalmazás erőforrást kér – az erőforrás-kezelő lejátsza, hogy mi történik akkor, ha megadja az erőforrást. Ha nem lesz holtponi, akkor odaadja.

Folyamatok - Processzek

- ! Program – A háttértáron tárolt változat
- ! Processz (folyamat) – a memóriában futó változat
- ! Folyamatleíró blokk – ez írja le a folyamat állapotát. Tartalma
 - § PID – Processz ID
 - § Programszámláló
 - § Regiszterek tartalma
 - § Memóriaterület adatai
 - § Használt perifériák és fájlok jellemzői

Szálak - Thread

- ! A folyamatok által elindított gyerek folyamat
 - § Közös memóriaterületet használ a szülővel
 - § Függetlenül fut a szülőtől
- ! Multithread alkalmazás
 - § Elindul egy szál
 - § Elindít más szálakat
 - § Pl. Microsoft Word – a háttérben megy a helyesírás ellenőrző

Mi az operációs rendszer?

- ! Szoftver
- ! Felépítése
 - § Kernel
 - § Shell
- ! Mi a feladata
 - § Hardver kezelése (kernel + driverrek)
 - § Szoftverek futtatása (kernel része a loader)
 - § Felhasználói kommunikáció (shell)
 - Parancssoros
 - Grafikus
 - § Felhasználó kezelés
 - § Jogosultságrendszer
 - § Memóriakezelés
 - § Erőforrás kezelés
 - § Folyamat kezelés
 - § ...

Operációs rendszerek működési módjai

i Kernel mód

- § Olyan üzemmód, amelyben futó folyamatok megszakíthatják a felhasználói módban futó alkalmazásokat
- § Az operációs rendszer kernel, és device driverek működnek ebben az üzemmódban

i Felhasználói mód

- § Kisebbs prioritású folyamatok. A kernel mód folyamatait semmilyen formában nem szakíthatják meg
- § A felhasználói programok, a shell dolgozik ebben az üzemmódban

Megszakítási folyamat

- ! Megszakítások fajtái
 - § Szoftveres – int utasítás
 - § Hardveres – külső hardvereszköz indítja
 - § Kivételkezelés (0-val való osztás, memóriavédelem)
 - § Trap – programozók hibakezelése
- ! A megszakítás folyamata
 - § Kérés (Int utasítás vagy hardver megszakítás)
 - § A regiszterek és a PC mentése a verembe
 - § Vezérlés a megadott kódra
 - § Paraméterek átvétele veremből
 - § A feldolgozó kód lefut
 - § Return utasítás hatására – veremből adatok vissza a regiszterekbe
 - § Vissza a vezérlés a megszakított folyamathoz
- ! A megszakítás letiltható (szoftver, hardver)
- ! Kivételkezelés – A kivételeket a programozó is kezelheti. Ha nem teszi. A program leáll

A kernel részei és feladatai

- ! Eszközkezelők
- ! Megszakításkezelők
- ! Rendszerhívások
- ! Erőforráskezelők
- ! Processzor ütemezés
- ! Memóriakezelés
- ! Állomány és fájlkezelés

A programok és folyamatok kezelésének folyamatai

- ! A shell segítségével tölti be a rendszer a programokat, de a programbetöltést a kernel folyamatai irányítják
- ! Programok futtatásának folyamata
 - § Program betöltése háttértárról
 - § Memória allokáció - relokáció
 - § Processzorütemező beállítása
 - § Futtatás
 - § Programok kiléptetése a memóriából
 - § Alkalmazás leállítása
 - § Erőforrások (fájlok, nyomtató, grafikus felület, Memóriát felszabadítja, stb...)
 - § A processzorütemező kiveszi a sorból

A kernel fizikai felépítése

- ! Monolitikus kernel – Olyan kernel, amikor minden modul egyszerre betöltődik a memóriába
 - § Pl. Régebbi Linux, Netware
- ! Réteges rendszer
 - § A eszközközkezelők, a shell külön rétegben helyezkedik el
 - § Moduláris:
 - § Többször felhasználható modulok (dll-ek)
- ! Virtuális gép alapú – Amikor egy szoftver emulál egy hardvereszközt. Ebben a szoftverben futnak más szoftverek
 - § Pl: Hyper-V, Virtual PC, Vmware, XEN
- ! Kliens-szerver modell alapú
 - § Az alkalmazások egy része fizikailag más helyen működhetnek, mint a szerver más része.
- ! Vegyes szerkezetű

Memóriakezelés

i Fizikai memória

§ Félvezetőkől előállított memóriamodulok

§ RAM - (Random Access Memory) - R/W – írható, olvasható, pl DRAM, SDRAM,

- A dinamikusan frissítendőek : Nagyon rövid időnként a memóriakezelő kiolvassa a memória tartalmát, majd visszaírja a korábbi állapotot, ha nincs frissítés, akkor elveszik az információ.
- Olcsó áramkörök

§ ROM – Read Only Memory – Csak olvasható. Lassabb, mint a RAM. Egyes fajtái UV fénnel vagy Elektromos árammal törölhetők és újraprogramozhatók (EPROM, EEPROM)

§ Flash memória – Külön áramforrás nélkül is megtartja állapotát.

- Nem lehet akárhányszor újraírni!
- Lassabb, mint a DRAM.
- Drágább, mint a DRAM

i Virtuális memória

§ Olyan memóriakezelés, amikor a fizikai memória méreténél nagyobb memóriáról tudnak a rendszerben futó alkalmazások. Általában a plusz helyet a háttértáron kialakított terület biztosítja

§ Pl. Windows: pagefile, Linux:page partíció

i Címzési módok: szegmens:offset

§ 32 bit szegmens és 32 bites offset esetén 2^{64} a megcímezhető memória mérete ~ 4 millió * 4 Millió byte

§ Átlapolts címzés – ugyanazon a címen több elektronika is elérhető felváltva

§ Lehet a szegmenshatár kisebb, mint a szegmens mérete => ugyanannak a fizikai címnek több címe is lehet.

§ Lehetnek változó szegmensméretek is

i Asszociatív memória – Olyan címző áramkör van a processzor és a fizikai memória között, hogy az adatok mozgathatnak nem az adat mozog, hanem a címe változik. A processzor a kívánt sorrendben látja az adatokat. Sokkal gyorsabb, a módosítás, drágábbak az ilyen memóriák

Memóriakezelés

i Memóriakezelő dolga

- § Virtuális vagy fizikai memória foglalás, felszabadítás
- § Osztott (shared) memória létrehozása
- § Fájlok osztott memóriához hasonló elérése (pipe)
- § Memóriavédelem
- § Kernel támogatás

i A memóriefoglalás folyamata

- § Egy lépéses foglalás – A címtartomány és a virtuális memóriát egy lépésben foglalja – pazarló
- § Két lépés (reserve – Címtartomány foglalása, commit – Virtuális memória foglalása) – gazdaságosabb. A commit csak akkor fut le, ha ténylegesen szükség van a memóriára
- § Laponként foglal területet a memóriakezelő. A lapok mérete 2-4 kb.

Memóriagazdálkodás – Fizikai tárkezelés

- ! Rögzített címzés – a szegmens és offszet is rögzített
- ! Áthelyezhető címzés – a cím szegmens része a bázisregiszterben van
- ! Fizikai tárkezelés
 - § Overlay – Állandó rész a memóriában, cserélődő modulok a háttértáron
 - § Tárcsere
 - Swapping – teljes memóriát cseréljük
 - Állandó partíciók – a partíciókat cseréljük (belső elaprózódás) – egy partíció – egy alkalmazás. A partícióban van szabad terület
 - Rugalmas partíciók – külső elaprózódás (sok apró üres memóriaszelet van a partíciók között) => garbage collection
 - Paging – A memóriát kis méretű lapokban kezeljük. A memórialapok és a folyamatok kapcsolatát laptáblák határozzák meg.

Memóriagazdálkodás - Virtuális tárkezelés

- ! A folyamatok kódjának éppen csak a futó része és környezete legyen a fizikai memóriában
- ! Kevés háttértár művelet kell => lokalitási elv => az alkalmazások 95%-ban sorfolytonosan helyezkednek el a háttértáron és a memóriában is
- ! A fizikai és a virtuális tár lapokra van osztva.
 - § Laptábla tárolja az egyes fizikai lapok helyét és tulajdonságait
 - § A laptáblában a helyen kívül az is benne van, hogy bent van-e a fizikai memóriában a lap.
 - Laphiba - ha a kívánt lap nincs a fizikai memóriában => azt be kell tölteni (esetleg betöltjük a környezetét is!)
 - Előtte ki kell menteni valamelyik korábban használt lap tartalmát . Ezt végzik a lapcsere algoritmusok.
 - Piszkos lapot nem kell menteni (nem írtunk bele)
 - Cél: a lehető legkevesebb laphiba legyen működés közben

Memóriagazdálkodás 3

! Lapkiosztási elvek

- § Egyenletes, arányos, prioritásos
- § Globális és lokális lapkiosztási elv

! Vergődés

- § Ha túl sok a laphiba, akkor a rendszer a legtöbb időt a memóriakezeléssel tölti
- § Pl. Kevés fizikai memória egy szerverben – a válaszidők drámaian megnőnek

! Lapcsere stratégiák

- § OPTimális (legkésőbb lesz szükség)
 - § FIFO – First In first Out
 - § LRU - Last Recently Used
 - § Second Chance – Második esély (Mielőtt kivennénk egy lapot a fizikai memóriából előtte még félretesszük egy kicsit, hátha szükség lesz rá)
- ! Programozó is tud javítani a laphibák számán megfelelő programozási technikákkal

Az operációs rendszer programozása

- i Közvetlen hardver programozás (játékgép, DOS)
 - § A BIOS funkciókat, portokat közvetlenül kellett programozni, hogy a megfelelő teljesítményt ki lehessen hozni.
 - § Assembly nyelvű programozás gyakori
 - § Device drivereket kellett írni (Video, hang)
- i Multitasking rendszer
 - § Nem szabad közvetlenül a hardvert programozni. (pl. DirectX használata)
 - § Figyelni kell az operációs rendszer feltételeire, jogosultságkezelésre, stb...
 - § A rendszer többi moduljának is kell kapnia processzoridőt
- i A programozás lehetőségei
 - § Rendszerhívások használatas
 - BIOS hívások
 - Operációs rendszer hívások
 - § API-k használata (=Application Programmink Interface)
 - A programok ezen keresztül szabványosan férnek hozzá az operációs rendszer szolgáltatásaihoz. Pl. File megnyitás, ablakkezelés, stb...

Shell fajtái

i Parancsfelület

- § Parancsnyelv (batch, power shell, sh, jscript, vbscript, php, stb...) – ha van hozzá értelmező
- § CMD.EXE, Command.com,
- § BASH,
- § Environment variables – a rendszer minden részében használható általános paraméterek. (SET PATH, TEMP, SET, FOR, stb...)

i Grafikus felület

§ Grafikus elemek

- Minden grafikus elemet az alábbiak jellemeznek:
 - Típusa (ablak, gomb, stb...)
 - Mérete (szélesség, magasság)
 - Hely (bal felső sarok koordinátái)
 - Melyik rétegben fekszik

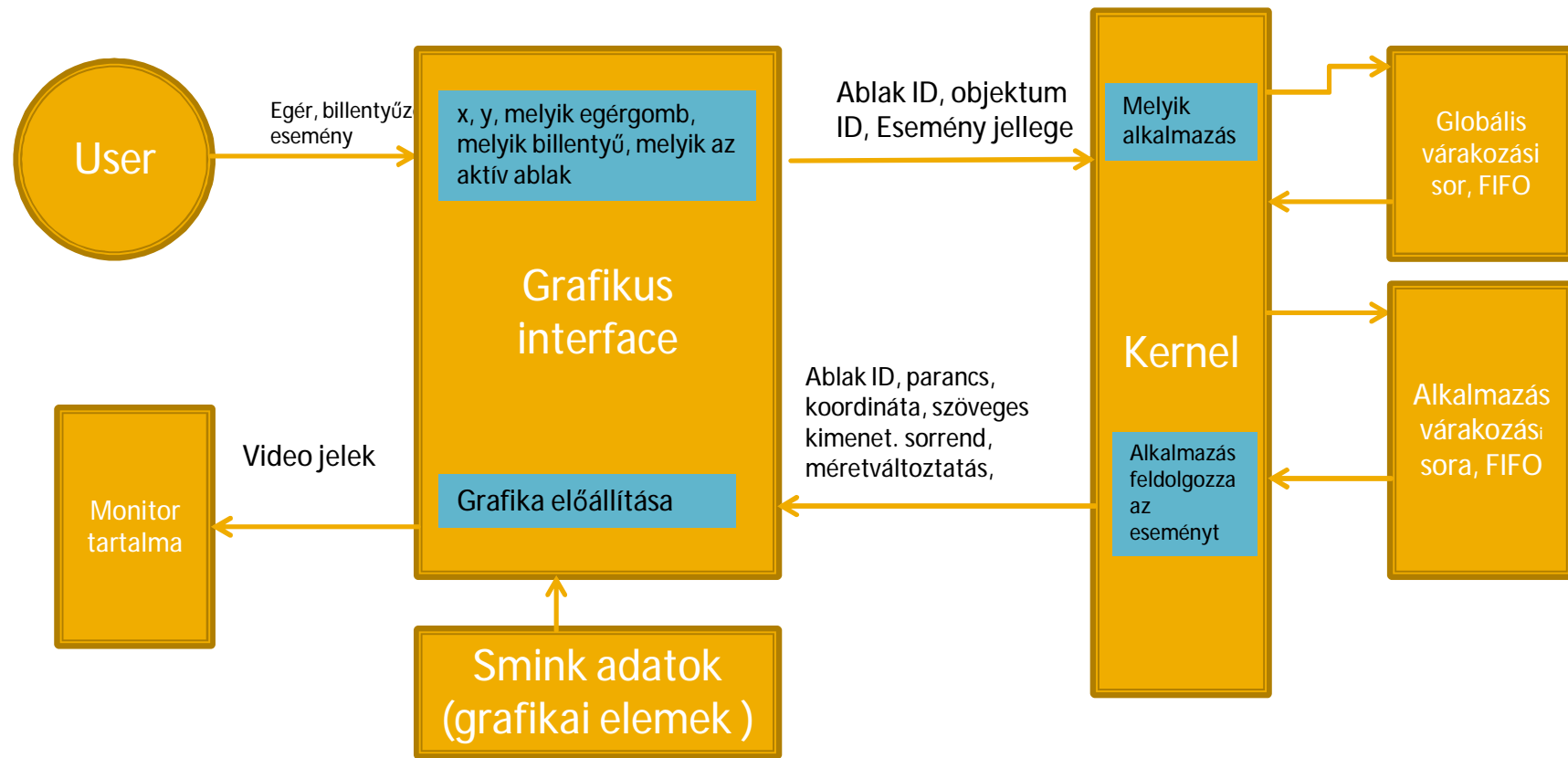
§ Ablakok jellegzetes részei (menük, keretek, stb...)

§ Események- a grafikus felületen végzett egérművelet vagy billentyűművelet

§ Eseménykezelő: Az események kezelését végző program

§ Az ablakok takarásának problémája

A grafikus interface működése



Háttértárak

! Miért szükséges a háttértár

§ Az alkalmazások és adatok tárolása

! Háttértárak típusai

§ Szekvenciális elérésű

- ! Mágnesszalag
- ! Lyukszalag
- ! Lyukkártya

! Véletlen elérésű

§ Csak olvasható

- Optikai lemez – véletlen elérésű
- Holografikus – (kísérleti, csak olvasható)

§ Írható-olvasható

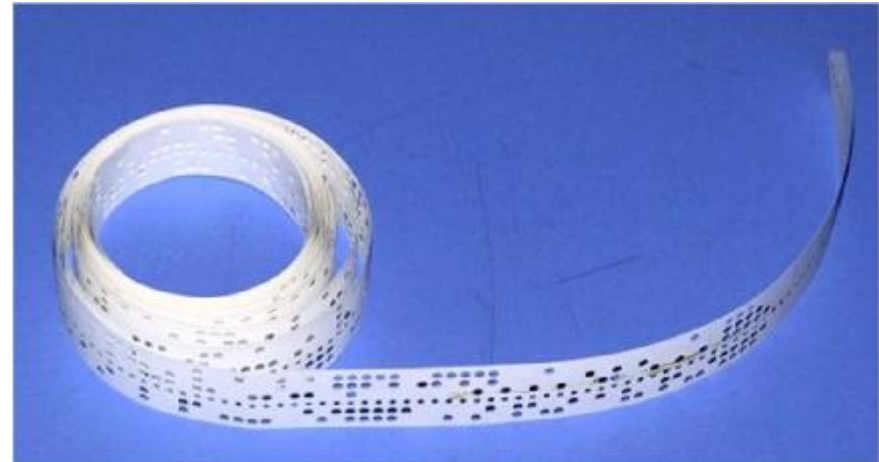
- Mágneslemez (floppy, merevlemez) – leggyakoribb
- Flash háttértár
- Már nem használt

Már nem használt háttértárak

! Lyukkártya



! Lyukszalag
6 lyuk egy sorban



Mágnesszalag

i Felépítése

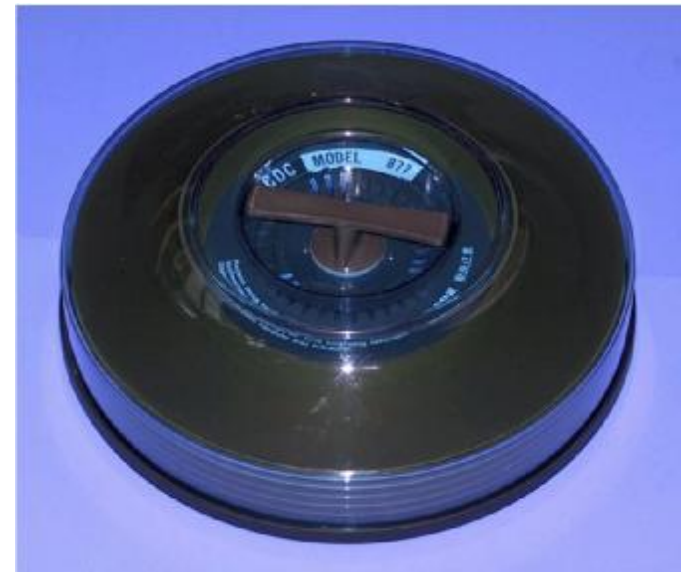
- § Vékony műanyag hordozón mágnesezhető réteg
- § Az író-olvasó fej relatív elmozdulási sebessége és a szalag hossza adja meg tárolható adat mennyiségét
- § Videomagnók – forgó fej, ferde sávok
- § Számítógépes rendszerek- álló fej, merőleges sávok
- § Kb: 2GB-200GB kapacitás

i Tulajdonságai

- § Lassú adatelérés
- § Szekvenciális adatelérés
- § Lassú keresés
- § Mentési, archiválási célokra
- § Nem sérülékeny
- § Gazdaságos
- § Időnként át kell másolni az adatokat, mert a menetek átmágnesezik egymást

i Eszköz

- § Streamer, DAT magnó, Videomagnó



Mágneseleme - Floppy

! Felépítése

- § Hajlékony műanyag lemezen vékony mágnesezhető réteg
- § Egy oldalas, két oldalas
- § Az író olvasó fej hozzáér a lemez anyagához
- § Szabványos méret: 5,25" , 3,5"

! Tulajdonságai

- § Mechanikailag sérülékeny (koszolódik, megsérül, megnyomódik)
- § Kis adatsűrűség
- § Kevés adat tárolására elegendő (360 Kb, 720 Kb, 1,2 Mb, 1,44 Mb)
- § Lassú az adatelérés (kb. 150 Kb/s)
- § Lassan elveszti a mágnesességét



Merevlemez fizikai felépítése

i Fizikai felépítés

- § A lemezek állandó forgássebességgel forognak, 5400-7200-10000-15000 rpm fordulatszámmal (Rotation Per Minute)
- § Manapság általában egy fém alapú lemezen, vékony mágnesezhető réteg
- § Író-olvasó fej módosítja a mágneses állapotot.
- § Az író-olvasó fej a lemez 1 nanométer magas légpárnán repül a lemez felett
- § Több lemez is lehet, lemezenként két fejjel
- § Nagy adatsűrűség,
- § Hagyományosan változó adatsűrűség
- § Gyors adatelérés (kb. 5-10 msec)
- § Nagy átviteli sebesség (max.40-90 MByte/s)
- § Méret:
 - 5,25" – régebben,
 - 3,5" – asztali gépek,
 - 2,5" – notebookokba
- § Szabványos interface: IDE, SATA, SCSI
- § Cache (8-32MB)



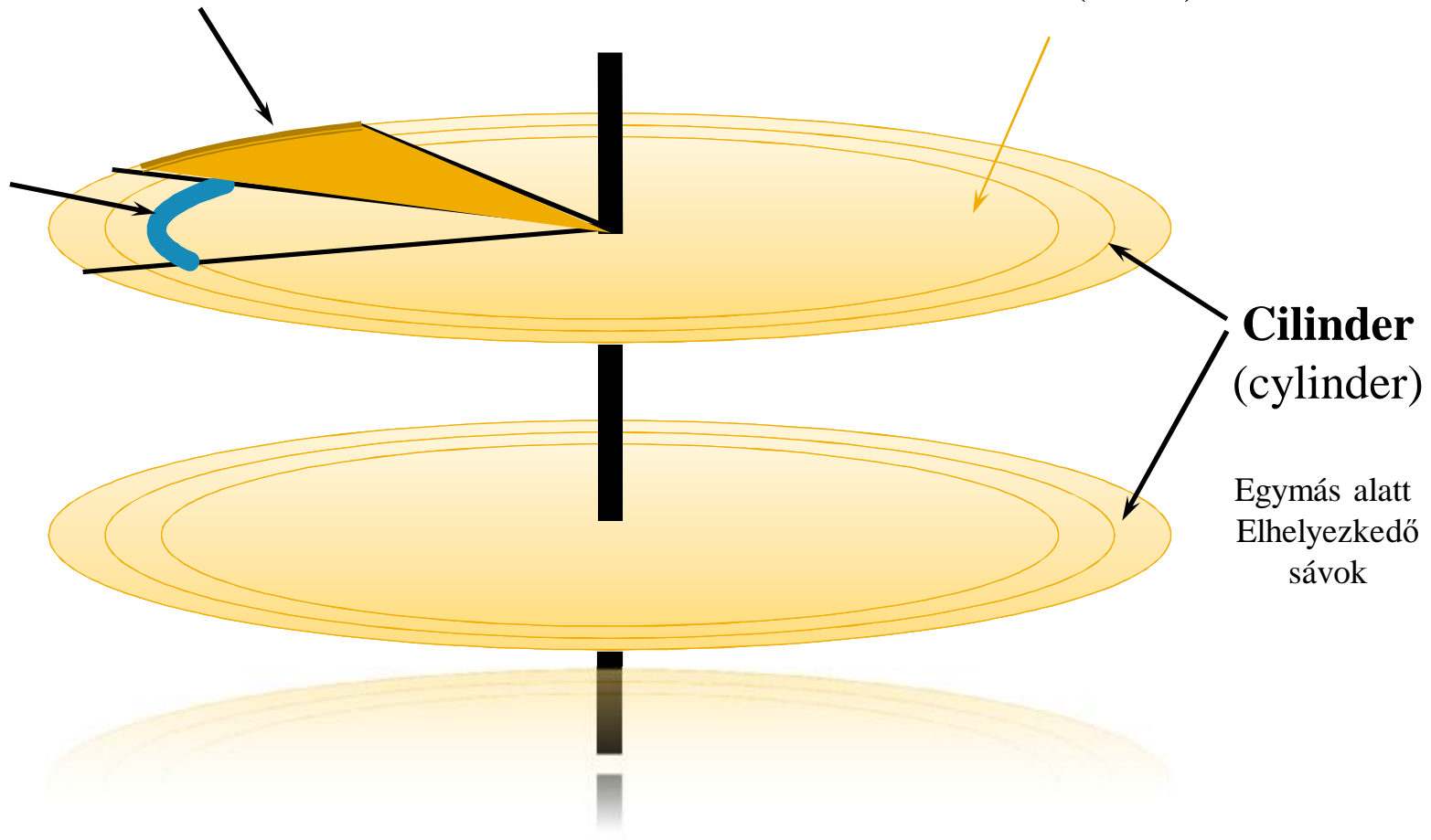
Merevlemezekkel kapcsolatos fogalmak

! Fogalmak

- § Seek time – fejmozgási idő
- § Latency time – elfordulási idő
- § Transfer time – Az adatátvitel ideje
- § Sáv (track) – egy lemez egy sávja
- § Szektor – A lemez legkisebb olvasható egysége
- § Cylinder – egymás feletti lemezek azonos sávjai
- § Cluster – Több egymás utáni szektor

szektor (sector)

sáv (track) - 2



Cylinder
(cylinder)

Egymás alatt
Elhelyezkedő
sávok

A merevlemezek logikai felépítése

i Logikai felépítés – a lemezen lévő adatok címzése

§ Az azonos cylinderben lévő adatokat a fejek elmozdulás nélkül olvassák

§ A szektor a legkisebb fizikai olvasási egység

§ A fejek egyszerre mozdulnak

i Az adatok elérése

§ A felhasználói alkalmazás írja vagy olvassa az adatokat (melyik partíció, melyik könyvtár)

§ Operációs rendszer

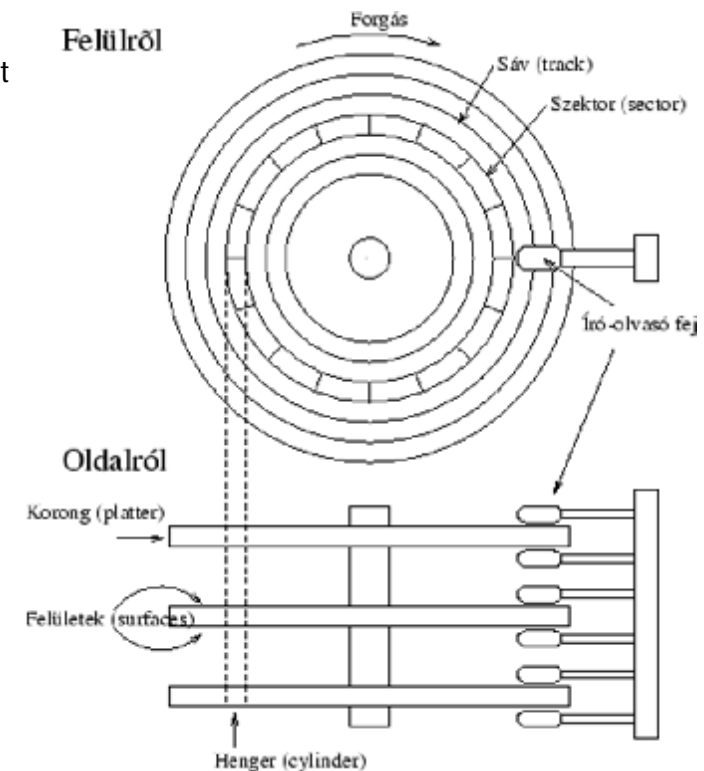
- => A kernel a fájlrendszerből kideríti, hogy a partíció melyik részén van az adat (Partíció, cluster – logikai írási egység)
- => A lemezkezelő driver megadja a lemez melyik CHS helyén található a kérdéses hely
- => A BIOS vezérli a fejmozgást és utasítja a HDD-t az adott szektor elérésére

§ A merevlemeznek BIOS-a

- => A számítógéptől érkező parancsokat a HDD BIOS-a értelmezi és lefordítja a valódi fizikai geometria szerinti mozgásokra

§ Az operációs rendszer lineárisan címzi az adatokat, a driver C,H,S-re fordítja le őket:

- $b = s * (i * t + j) + k$
- s – egy sávban a szektorok száma
- t – A cylinderben lévő sávok száma
- i – a megadott cylinder
- j – a fej sorszáma
- k – a sávon belüli szektor sorszáma
- b – Az operációs rendszer által kért sorszám



Merevlemez főbb tulajdonságai

! Tárolókapacitás

§ ez jellemzi a winchestert abból a szempontból, hogy mennyi adat fér rá: kezdetekben csak pár megabájt volt, manapság már 40 GB – 2 TB között mozog.

! Írási és olvasási sebessége

§ ezt nagyban befolyásolja a lemez forgási sebessége, amely jellemzően 5400, 7200, 10 000 vagy 15 000 fordulat/perc (rpm).

§ A merevlemez átviteli sebességének növelésének érdekében beépítenek egy gyorsítótárat (cache-t).

Optikai lemezek - Meghajtók

i Lemez felépítése

- § Felső burkoló lakkréteg
- § Alumínium – tükröző réteg
- § Adathordozó

i Működése

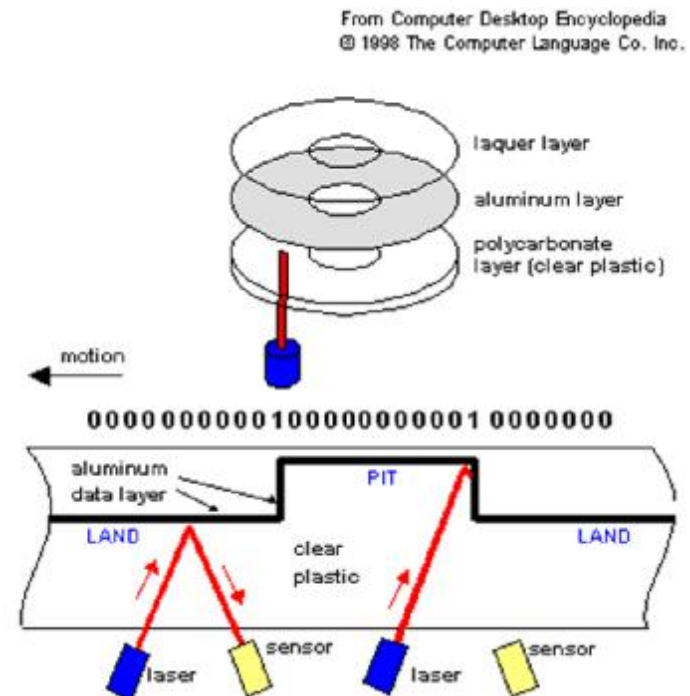
§ Alulról világító lézer

- Alacsony energia – olvasás
- Magas energia – Írás, újraírása

§ Visszaverődő fény érzékelése

- PIT – optikai lyuk – nincs visszavert fény
- Az információt az optikai lyuk hossza szolgáltatja!

§ A lakkréteg sérülékeny



Merevlemez csatolófelülete

i Csatolófelület:

§ ezen keresztül történik az adatátvitel

§ több fajta létezik:

- ATA (PATA),
- SATA (SATA I, SATA II, SATA III),
- SCSI, SAS (Serial Attached SCSI),
- FC (Fiber Channel) – 4GB/s
 - Csavart érpár – 100 MB/s
 - Optikai kábel – 200-400 MB/S



Optikai adathordozó adattárolása

- ! Az adatokat belülről induló spirális pályán rögzítik – szektoronként
- ! A pálya két oldalán plusz információ az író olvasó fej pályán tartása céljából
- ! Állandó adatsűrűség
- ! Az egymás után következő bit információk szétszórva helyezkednek el a lemezen
- ! Plusz hibajavító bitek találhatóak a lemezen (kb. 50MB / CD ROM)

Optikai meghajtók tulajdonságai

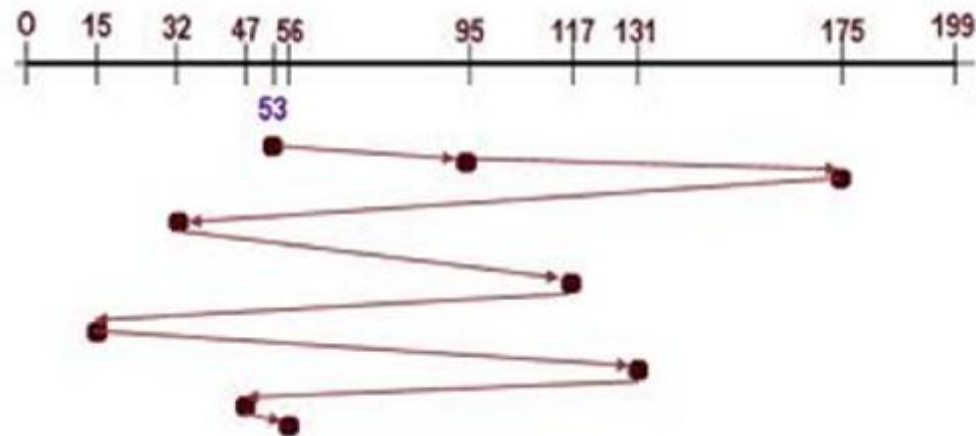
- ! Lassú fejmozgás – egyidejű több folyamat olvasása lelassítja – 100 msec elérés => Célszerűen lineáris olvasás a jó.
- ! CD olvasó – kis energiájú lézer
- ! CD író – nagyobb energiájú lézer
- ! Lemezek fajtái
 - § CD-ROM – csak olvasható (nyomtatott- írott)
 - § CD-RW – Írható, olvasható, újraírható
 - Lemez írása: a hordozó műanyagban szerkezeti változások jönnek létre a hő hatására
 - Lemez törlése: a lézerrel a pitek szerkezetét homogenizálni lehet
 - Korlátozott számban írható újra, mert a lemez anyaga „elfárad”
- ! DVD
 - § Felépítése ugyanaz, mint a CD
 - § 4,5 GB anyag fér rá
 - § Kétoldalas – akkor a duplája
 - § Két rétegű – akkor a duplája
 - § Más színű fény – „Blue Ray” – a nagyobb frekvenciájú fényt jobban lehet fókuszálni

Lemezműveletek ütemezése

- ! Több folyamat verseng a perifériaért – cél az átlagos hozzáférési idő csökkentése
- ! A fejmozgás optimalizálásának céljai
 - § Állandó válaszidő
 - § Átbocsátási képesség növelése
 - § Válaszidő szórásának minimalizálása
- ! Feltétel: egyenletes adateloszlás a lemezen

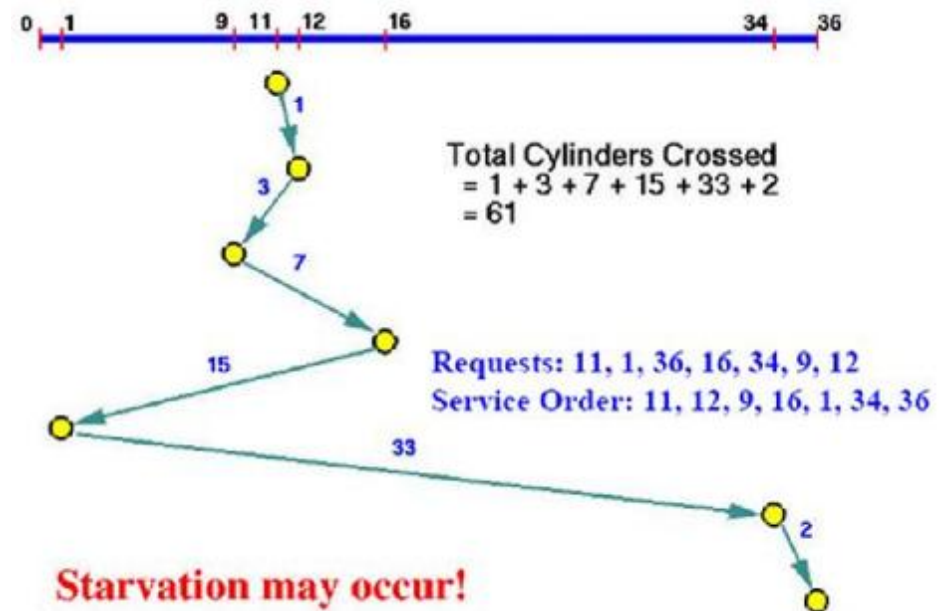
FCFS – First Come First Served

- ! A kérések beérkezésének sorrendjében történik a kiszolgálás
 - § Kicsi átviteli sebesség
 - § Nagy átlagos válaszidő
 - § Szórás kicsi



SSTF – Shortest Seek First Time – Legrövidebb fejmozgási idő

- ! Az aktuálshoz legközelebbi cylinderhez tartozó kérést szolgálja ki
 - § Nagy szórás
 - § Kiéheztetés léphet fel
 - § Közepes átbocsátási sebesség
 - § Kis átlagos válaszidő



SCAN - Pásztázó

- i Az aktuális fejmozgási irányhoz tartozó kéréseket szolgálja ki
 - § Közepes válaszdő
 - § Nagy átbocsátás
 - § Kis szórás
 - § A középső cylindereket többször olvassa

NSCAN – N lépéses pásztázó

- ! Egy irányba mozogva N kérést elégít ki
- ! A következő kéréseket irányváltás után elégíti ki
 - § Nagy átbocsátás
 - § Kis válaszidő
 - § Kis szórás

C-SCAN – Körbeforgó pásztázó

- ! Csak egy irányú fejmozgás van
- ! N lépéses is lehet
 - § Nagy átbocsátás
 - § Kis válaszidő
 - § Kis szórás

Kombinált módszerek

- ! Alacsony terhelésnél – SCAN
- ! Közepes terhelésnél – C-SCAN
- ! Nagy terhelésnél – N-SCAN

További teljesítménynövelés

- ! Az adatok nem sorfolytonosan helyezik el egy cylinderen belül. A lemezműveletekhez szükséges idő alatt elfordul a lemez. Akkor kell a következő szektornak odaérni a fejhez
- ! Lemez tömörítése –
 - § Gyors processzor: a tömörítéshez szükséges idő kevesebb, mint az több adat olvasásához szükséges idő (50%-os tömörítési arány mellett)
- ! Gyakran használt adatok a középső cylinderen
- ! Az egymás utáni adatok több cylinderen
- ! Cache használata
- ! Több adatblokk átvitele egy írással

Adatok biztonságos tárolása

- ! Backup – volt szó róla
- ! RAID 0-6 – Redundáns adattárolás
- ! RAID 0 – Nem redundáns tárolás. A több háttértáron az egymás utáni adatok külön háttértáron vannak
- ! RAID 1 - Lemez tükrözés
- ! RAID 2 – Minden bitet más tároló tárol + paritás biteket használunk
- ! RAID 3 – Elég egy paritás bit
- ! RAID 4 – egy HDD paritás blokkokat tartalmaz, de ugyanaz, mint a RAID 3
- ! RAID 5 – Ugyanaz, mint a RAID 4, de a paritás blokkok elosztva vannak a HDD-ken
- ! RAID6 – Ugyanaz, mint a RAID 5, de több paritás blokk van egy-egy lemezen

SSD háttértárak (Solid State Drive)

- i Qvantummechanikai elven működik

- i DRAM SSD

 - § Saját akkumulátorral működik

 - § Nagy energiafelvétel

 - § Melegedés

 - § 80-800 \$/GB

- i Flash alapú SSD

 - Lassabb, mint a DRAM

 - Csak korlátozott számú írás lehetséges – hibajavító és észlelő algoritmusok

 - SLC – Single Level Cell – 1 bit/memóriacella – gyorsabb, drágább

 - MLC – Multi Level Cell – 2 vagy 4 bit/memóriacella

 - 3-10\$/GB

SSD tulajdonságok

i Csatolófelület - SATA

i Előnyök

- § Gyorsabb: olvasás: 250MByte/s, írás: 170 MB/s
- § Hozzáférési idő 85 -115 µsec (HDD: 5-10 milisec)
- § Alacsony áramfelvétel
- § Csendes
- § Ütésálló
- § Állandó teljesítmény
- § Széles hőmérsékleti tartományokban működik (~70 °C-ig, HDD 5-55°C-on)

i Hátrány

- § Drágább
- § Kis kapacitás (max. 256 GB)
- § Hirtelen áramkimaradás, mágneses mező változására érzékeny
- § Újraírások száma korlátozott (100 000 – 5 millió)
- § Lassabb írás a Flash memóriákál

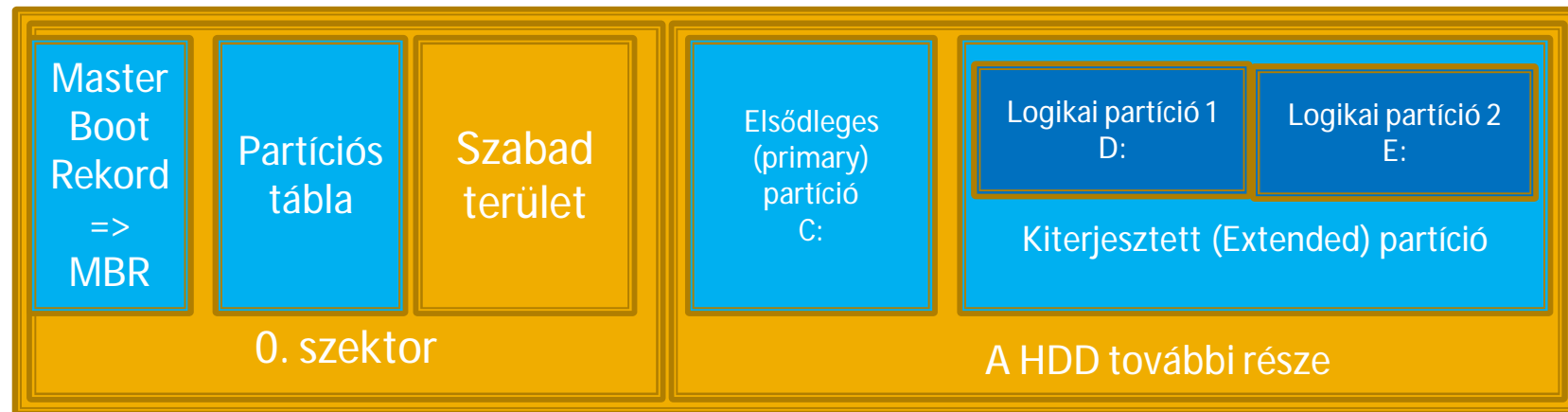
- i Új tendencia: RAID-be kötött memóriák, amelyek a kieső hibás memóriablokkokat képesek helyettesíteni – Intel

http://prohardver.hu/teszt/ssd-teszt_mitoszok_es_tenyek/az_ssd.html

Partíciók

- ! 1 Merevlemezen max 4 elsődleges (Primary partíció) vagy
- ! 3 elsődleges és 1 kiterjesztett (Extended) partíció lehet
- ! A kiterjesztett partícióban tetszőleges számú logikai partíció lehet
- ! Egy partíció lehet aktív is. Bootolás csak aktív partícióról történhet.

HDD logikai felépítése – a bootolás folyamata



- A bootolás folyamata
 - A BIOS a POST teszt után átadja a vezérlést a MBR-ra
 - Az MBR megállapítja, hogy melyik partíció aktív
 - Átadja a vezérlést az aktív partíción lévő boot rekordnak (0. szektor) (általában Windows rendszereken ez az 1. elsődleges partíció)

A partíciók létrehozásával meg is lehet határozni a partíció fájlrendszerét

Multi boot rendszer

- ! 1 HDD több bootolható operációs rendszer
=> Boot Manager programok
- ! A partíciós tábla utáni részre töltik fel a kódjukat vagy
- ! Egy rejtett aktív partícióról indul a program, majd átállítja azt, hogy melyik partíció legyen rejtett vagy aktív
- ! Módosítja az MBR-t.

Fájrendszer

i Microsoft

§ FAT (File Allocation Table) fájlrendszer

- Használhatja: MS-DOS, Windows 9x, Win NT, OS/2, Linux, Unix
- File Allocation tábla szerepe: Itt tárolják a partíció foglaltsági térképét, a fájlok darabjainak egymás utáni helyeit.
- 1 bejegyzés a FAT-ben = 1 cluster a partíción (Cluster = legkisebb írható-olvasható egység)
- 2 FAT példány egymás után, ha az egyik megsérülne, akkor a másik még tárolja az adatokat

§ NTFS – New Technologie File System

- Windows NT idején jelent meg
- Sok új tulajdonság

i Linux

§ extFS1, 2, 3, 4

FAT

- § A hosszú fájlok végének lassú elérése
- § Hibák lehetségesek – láncolások megszakadása esetén a fájl vége elérhetetlenné válik
- § FAT12 – csak floppy-kon
- § FAT16 – kisebb HDD-ken (Max. 2GB-ig)
 - Hosszú fájlnev (256 karakter) kezelése és 8+3 fájlnev kezelése
 - 4 attribútum lehet: Read Only, Hidden, Achív, System, (+ Volume, Dir)
 - A gyökérkönyvtárban lévő könyvtárak max. száma 4096
- § Fat32- Win98-től használják
 - Max: 2 TB-os partíció

NTFS

i A Windows NT-vel jelenik meg

- § 2 TB-os maximális méret
- § Fájlnev: max. 256 karakter
- § Hibatűrő – Tranzakció kezelő – rendszer – minden írás előtt készít egy mentést az írást érintő clusterek tartalmáról.
- § A fájlok nyilvántartása \$MFT (Master File Table). Két példányban helyezkedik el.
- § Futás közben képes a hibás szektorokat felismerni és áthelyezni a tartalmat másik helyre és a hibás szektort kizárja.
- § Hozzáférési lista alapú ellenőrzés => jogosultságkezelés
- § Többszörös fájl hozzáférés => megnyitott fájlok megnyitása írásra
- § Hard link
- § Tömöríthető könyvtár
- § Kapcsolódási pontok (más partíciók bekapcsolása a fájlrendszerbe)
- § Kvótázás
- § Indexelés
- § 6 különböző attribútum
- § Titkosítás – Csak az adott user képes hozzáférni az állományokhoz.

- § FAT konverziója NTFS-re pl. convert D: /fs:ntfs

Lemezkezelési műveletek

- ! Particionálás – A partíciók rendszerét alakítjuk ki
- ! Formázás – Egy patíció fájlrendszerét alakítjuk ki
- ! Lemez ellenőrzése – A lemezen lévő logikai hibák keresése és ha lehetséges kijavítása:
chkdsk C:
chkdsk C: /f
chkdsk c: /f /r
- ! Töredezettségmentesítés

Linuxos fájlok elhelyezkedése

i Inode

§ Minden fájl bejegyzés egy inode-dal kezdődik

§ Inode-nak van

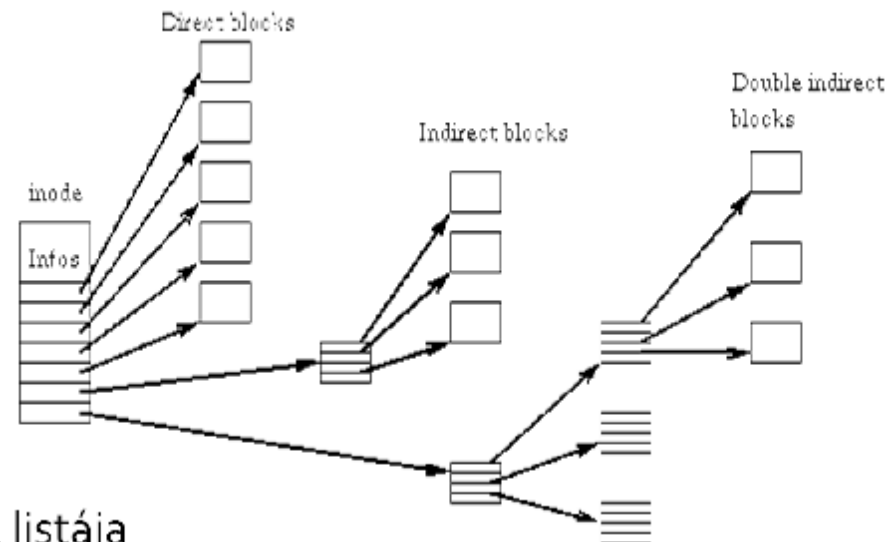
- Fejléc
- Mutatók a fájl blokkjaira
- Utolsó 2 blokk mutat újabb inode-okra

§ A directory-k is inode szerkezetűek

§ Minden directory könyvtárbejegyzések listája

§ A 2. inode a root könyvtár

§ Hard linkek vannak – inode-ra mutatnak.



i Tulajdonságok

§ Gyors fájllelérés – Gyorsabb, mint a FAT és az NTFS

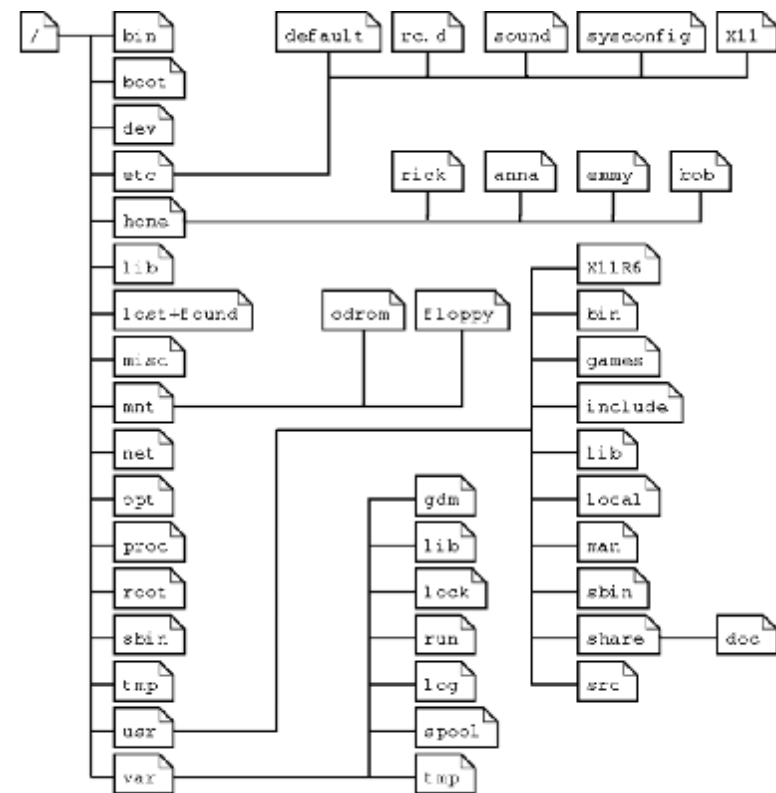
§ Ha megszakad a láncolás sok fájl tartalom veszik el

Linux fájlrendszerek – Ext2 ...Ext4

- ! Ext2 – Az első fontos Linuxos fájlrendszer
 - § Nem tranzakciókezelő
 - § Max fájlrendszer méret 16TB
 - § Max fájl méret 2 GB
- ! Ext3 – Ext2 módosítása
 - § Tranzakciókezelő, tehát hibatűrő. Szabályozható, hogy mit naplózzon
 - § 16TB-ig nőhet a fájlrendszer mérete
 - § 2TB lehet a legnagyobb fájl
 - § Könyvtárak mélysége 32000
- ! Ext4 – Az Ext3 továbbfejlesztése
 - § 1 EB-ig nőhet a fájlrendszer mérete (1 EB=1024PB = 1024*1024 TB)
 - § 16 TB lehet a legnagyobb fájl
 - § Könyvtárak mélysége korlátlan

Linux fájlrendszer tipikus példái

- Bin – futtatható állományok
- Boot – bootoláshoz szükséges cuccok
- Dev – device driverek
- Home – Userek könyvtárai
- Etc – rendszer konfiguráció
- Lib – közös könyvtárak
- ...
- Mnt – floppy, cd meghajtó
- Tmp - temporary
- Usr – userokhoz tartozó cuccok
- Var – változó cuccok



Fájlkezelés, könyvtárkezelés

i Fájlnevek

- § A fájlnev egyedi
- § Problematisa a fájlnev kódolása
- § FAT – kis betű nagy betű nem számít, Linux különböznek
- § 256 karakteres hosszú fájlnev
- § Felhasználói nevek
- § Rendszer szintű nevek: CON, LPT1, LPT2, COM1, COM2, PRN

i Tipikus fájl műveletek

- § Létrehozása – Editorral vag más felhasználói programmal
- § Megnyitása írásra, olvasásra, írásra és olvasásra – Általában egyszeres hozzáférés a fájlhoz. NTFS esetén a „stream” fájlokat egy időben többen is használhatják.
- § Fájl másolása – Lassú, mert két példány jön létre
- § Fájl mozgatása – Partíción belül gyors, mert csak a könyvtárbejegyzést kell átírni
- § Fájl törlése – Nem történik a fájl tartalmának törlése, hanem csak a fájlbejegyzést törlik

Speciális fájlrendszerek

- ! Windows 2003-tól
 - § EFS fájlrendszer: Virtuális fájlrendszer, fizikai gépek különböző könyvtáraiból lehet fájlrendszert létrehozni.
- ! Sharepoint szerver alapú fájlrendszer:
 - § SQL adatbázisban tárolják a fájlokat
 - § WebDAV klienssel hozzáférhető

Jogosultságkezelés

- ! Minden objektumra van jogosultság
 - § Könyvtárak
 - § Fájlok
 - § Device (Printer, scanner, stb...)
 - § Alkalmazások
 - § Szolgáltatások
- ! Különböző rendszerek különböző koncepciók
 - § Linux
 - § Netware
 - § Windows NT család

Jogosultságkezelés – közös elvek

- i A userok csoportokba szervezhetők
 - § Ha egy csoportnak a tagja a user, akkor minden jog, amit a csoport kap, rá is érvényes
 - § A csoportok egyes esetekben hierarchiát is alkothatnak
 - § Hogyan adjunk jogot?

Hogyan osszuk a jogosultságokat?

1. Feladatok, szervezeti egységek alapján csoportokat alkotunk
2. Megadjuk a csoportoknak a jellemző jogait
3. Felvesszük a felhasználókat
4. Besoroljuk a megfelelő csoportba
5. Megadjuk nekik a saját jogaikat
6. Megadjuk a kivételes tiltásokat

Linux jogosultságkezelése 1

- ! Istencsászár: root – mindenhez van joga, ezért veszélyes használni
 - § Root-tal nem lépünk be távolról
 - § Másik user-t használunk
- ! Minden objektum
 - § Olvasható (r) - 4
 - § Írható (w) - 2
 - § Futtatható (x) - 1
- ! Minden objektumnak van
 - § Tulajdonosa
 - § Csoportja
 - § Mindenki más

Linux jogosultságkezelése 2

! Példa

- § Tulajdonos (rw-) => 6
- § Csoportja (rw-) => 6
- § Mindenki más (r--) => 4
- § => rw-rw-r-- => 664
- § Mit jelent a 775? => rwxrwxrw-

! Chmod – paraméterek

§ Művelet

- + jog hozzáadás
- - jog elvétel
- = engedély beállítása
- r,w,x jogok fajtái

§ Kinek adjuk a jogot

- u tulajdonosnak adunk jogot
- g Csoportnak adunk jogot
- o mindneki másnak adunk jogot
- A mindenkinek jogokat adunk

§ -R rekurzív jogok adása

Linux jogosultságkezelése 3 chmod példák

- ❗ `chmod u+x gyakorlas.txt` # Futtatási jogosultságot ad a fájl tulajdonosának.
- ❗ `chmod go-rx gyakorlas.txt` # Visszavonja az olvasási és futtatási jogosultságot a csoport tagjaitól és mindenki mástól.
- ❗ `chmod a=r gyakorlas.txt` # A fájl jogosultságait csak olvashatóra állítja minden felhasználó számára.
- ❗ `chmod 444 gyakorlas.txt` # A fájl jogosultságait csak olvashatóra állítja minden felhasználó számára.

Fájl tulajdonosának módosítása

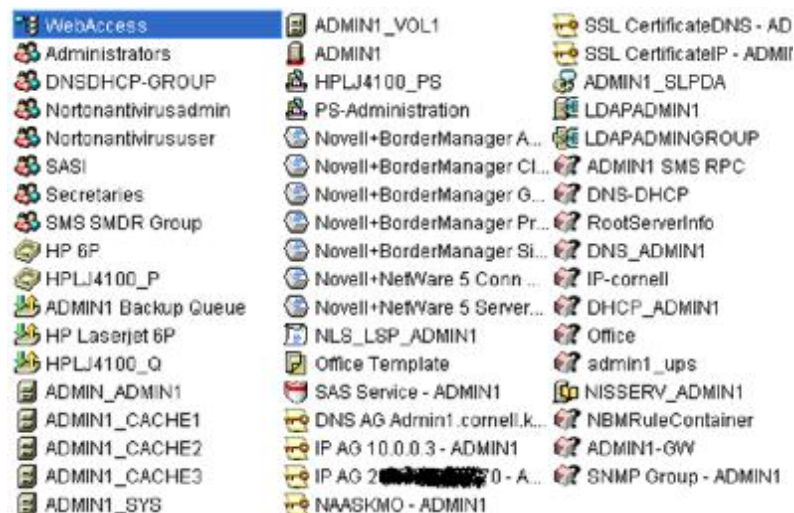
- ! A fájl tulajdonosának és csoportjának megváltoztatására alkalmas parancsok:
- ! `chown #` A fájl tulajdonosát változtatja meg.
- ! `-c #` Azon állományok nevét jeleníti meg, melyeknek a tulajdonosa megváltozott.
- ! `-f #` Tiltja a hibaüzenetek megjelenítését.
- ! `-R #` A fájlok tulajdonosát az alkönyvtárakban is módosítja.
- ! `-v #` A módosításokról részletes listát készít.
- ! Példák:
 - § `chown zsozso gyakorlas.txt #` A fájl "zsozso" tulajdonába kerül.
 - § `chgrp felhasznalo gyakorlas.txt #` A fájl a "felhasznalo" csoportba kerül.
 - § `chown zsozso:felhasznalo gyakorlas.txt #` A fájl "zsozso" tulajdonába és a "felhasznalo" csoportba kerül.

Webszerverek jogosultságai

- ! Root user – nem használjuk
- ! Apache – user: Futtatja az Apache-ot => php-t
- ! Xy-user – csak a saját könyvtárstruktúráját láthatja

Netware jogosultságszisztere

- ! Novell Directory Services – Címtár
 - § Jogosultságok kezelése
 - § Userok adatai
 - § Beléptető, authetnikáló rendszer
 - § Elosztott
 - § Szinkronizálható több Netware szerver között
 - § AD-vel korlátozottan kompatibilis
- ! Kezelése
 - § Régen Windows kliens program
 - § Console One kliens



NetWare-ek jogosultságkezelése

i A szerver nem hozzáférhető a userok számára, konzolnak nem kell jogosultságkezelés

i Speciális user: Supervisor = Istencsászár

i Auditáló user = Nem tud semmit csinálni, csak ellenőrzi a többi usert.

i Csak könyvtárakra és fájlokra vonatkozó jogok

§ Read – Olvashat egy fájlt

§ File Scan – Listázhat egy könyvtárat

§ Write – Írhat egy fájlba (könyvtárba)

§ Execute – Futtathat egy fájlt

§ Create – Létrehozhat egy fájlt, könyvtárat

§ Modify – Módosítani lehet egy fájl tartalmát

§ Delete – Fáj(könyvtár) törlési jog

§ Supervisor – Mindenre van joga

§ Access – Másnak átadhatja a felhasználó a jogait

i File-ok saját attribútumai

§ Read only

§ Hidden

§ System

§ Archív

i Userok jogosultságrendszere

§ Userok tagjai lehetnek csoportoknak

§ Csoportok tagjai lehetnek más csoportoknak

§ Userok beállíthatják, hogy más userekkel egyenrangúak legyenek

i Jogosultságok lehetnek az NDS-re is

§ Olvasási

§ Módosítási

§ Létrehozási

§ Auditálása

Windows NT család jogosultságkezelése

- ! Beépített felhasználók – lokális userek
 - § Alapértelmezett felhasználók és csoportok => Demo
 - § Administrator = Rendszergazda,
 - § Vendég
 - § Jogosultsági szintek: Rendszergazdák, Kiemelt felhasználók, Felhasználók, Vendégek
- ! Userek - Profilok
 - § Lokális profil
 - § Roaming profil - később

Active Directory

i Active Directory = Címtár

§ Tartalma

- Felhasználók nevei és egyéb adatai,
- Jelszó
- Viszonya a rendszerben
- UID – User ID – ez azonosítja a felhasználót

§ DNS-re épülő szolgáltatás

§ Feladatai

- Bejelentkeztetés – jogosultságkezelés
- Címtár adatok kiszolgálása különböző alkalmazások részére (pl. Exchange)

§ Gépeket is domainba kell léptetni -> Utána a gépbe belépést az AD engedélyezi

i Active Directory felhasználók

§ Alapértelmezett felhasználók és csoportok => Demo

§ Konténerek – A felhasználók csoportjainak egyedi tárolóhelye

- Nem lehet két egyforma felhasználói név az AD-ben.

i Gépek beléptetése - hogyan kell?

i Group Policy-k

§ Az AD-ben lévő csoportokra alkalmazható szabályok gyűjteménye

- Userekre vonatkozhat
- Munkaállomásokra vonatkozhat
- Szerverekre vonatkozhat
- Alkalmazásokra vonatkozhat
- Eseményekre (pl. belépés, kilépés)

Jogosultságok osztása

- ! Lokális jogok – engedély, tiltás
 - § Fájlokra és könyvtárakra adható jogosultságok (olvasás, írás, módosítás, futtatás, tulajdonos) – Lokális jogok
 - § A jogok öröklődnek alapesetben a könyvtárstruktúrában
 - § El lehet venni az öröklődést
- ! Megosztási jogok – engedély, tiltás
 - § A jogok öröklődnek
- ! A kétféle jog eredője lesz az effektív jog

Célszerű kialakítás

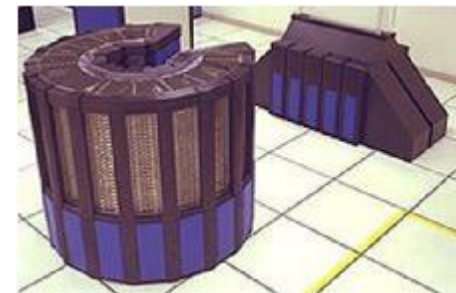
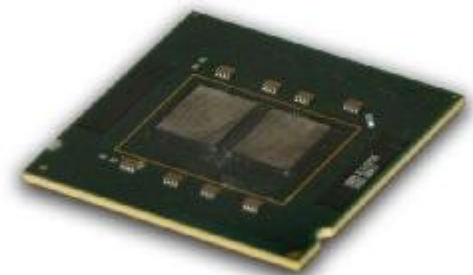
- ! Konténerek tükrözzék a személyek és csoportok kialakítását
- ! Usernév tükrözze a valódi felhasználót
- ! A csoportok legyenek a felhasználók tényleges csoportjainak leképezései
- ! Csak annyi jogot, amennyi kell a munkához, de ahhoz legyen elég
- ! A csoportok tagjai kapják a megfelelő jogot és csak a specialitások legyenek egyénileg beállítva
- ! Megosztásokban lévő könyvtárak elérésének finom szabályozása
 - § A Megosztási jog: mindent engedélyezzünk
 - § Lokális jog: csak azok az engedélyek, amelyek szükségesek
 - § Az öröklést a korlátozott könyvtárra nem szabad engedélyezni, de a korlátozott könyvtár jogai tovább öröklődhetnek lefelé

Célszerű tevékenységek, tipikus esetek

- ! Számítógép beléptetése AD-be
- ! Meglévő felhasználók gépeinek beléptetése AD-be, profilok átmásolása
- ! Nagy számú felhasználó létrehozása AD-ben
 - § Scriptekkel – Power Shell, VBScript, Jscript

Szorosan csatolt rendszerek

- ! Céljuk a számítási teljesítmény növelése, működési biztonság növelése
- ! Megoldási módok
 - § Több processzor vagy több mag
 - Hyperthreading is egyfajta virtuális több magot jelent
 - § Közös memória
 - § Közös háttértár
 - § Buszrendszeren kommunikálnak az processzorok
 - § Például egy alaplapon 2 vagy 4 db processzor
 - § Egy tokban több processzormag
- ! Példák



Lazán csatolt rendszerek (cluster)

! Céljai

- § nagyobb teljesítmény elérése
- § Szolgáltatások futtatásának biztosítása
- § Felhő (Cloud) szolgáltatások

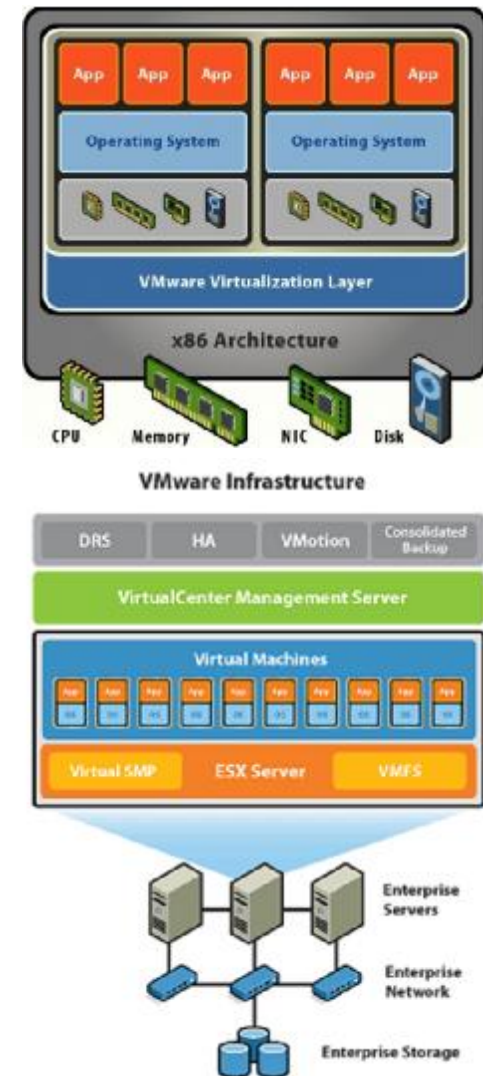
! Megvalósítás

- § Több egyforma teljes értékű számítógép nagy sebességű gerinchálózaton (backbone) összekötve.
- § Egy vagy több közös háttértár
- § Minden gépen futnak a megfelelő alkalmazások, hogy egymástól átvehessék a feladatokat
- § Ha az egyik gép kiesik, akkor automatikusan átköltöznek a szolgáltatások a másikra



Virtualizálás – virtuális gépek

- i Amikor egy szoftver emulál egy hardver vagy egy szoftveres környezetet
- i Céljai
 - § A hardver optimálisabb kihasználása
 - § Biztonságosabb szolgáltatások (több gépen virtualizálunk)
 - § Más hardver típus virtualizálása
 - § Szoftver platform virtualizálása
 - § Tesztelési cél – Vírusteszt labor
- i Operációs rendszer virtualizálási megoldások
 - § A hardveren közvetlenül fut egy virtualizációs szoftver
 - A virtualizációs felületen több virtuális gép fut és azokon vannak telepítve az operációs rendszerek
 - A virtuális gépek egymással virtuális hálózatot alkothatnak
 - Pl. Hypervisor VMWare ESX szerver
 - § Egy operációs rendszer futtat ez virtualizációs szoftvert
 - A virtuális gépek egymással és a host számítógéppel alkothatnak virtuális hálózatot
 - Lassabb megoldás, mert a host operációs rendszert is ki kell szolgálni
 - Pl. Virtual PC, VMWare Workstation
 - § Terminál emuláció
 - Távoli gépen igazi szerver fut. A felhasználó csak azt látja, mintha egyedül használná az operációs rendszert
- i Problémák
 - § A processzor jogosultsági szintjeinek megoldása
 - § Device driverek
 - § Egyes esetekben sebességi problémák
 - § Megfelelő hardver szükséges hozzá



Más hardver virtualizálása, szoftveres környezet emulálása

- ! Más hardver virtualizálása a fejlesztőeszköz biztosítása miatt
 - § Pl. C64 emulátor, Android emulátor, Mac emulátor PC-n vagy Mac-en PC emulátor
- ! Szoftveres környezet biztosítása
 - § JAVA Virtual Machine – A JAVA programok a JVM-ben futnak és elvileg nem léphetnek ki belőle