

Operációs rendszer szintje

Operating System Machine (OSM)

Ezen a szinten programozóknak rendelkezésre állnak a felhasználói módban használható ISA szintű utasítások és az operációs rendszer által hozzáadott utasítások: **rendszerhívások (system calls)**. Ezeket az operációs rendszer eljárásai valósítják meg (értelmezés).

Máté: Architektúrák

12. előadás

1

Virtuális memória

Régen nagyon kicsi volt a memória. Sokszor nem fért el az egész program a memóriában.

Overlay (átfedés): A program több része fut ugyanazon a memória területen, mindig az aktuálisan futó rész van a memóriában, a többi rész mágneslemezen van.

A programozó dolga a feladat átfedő részekre bontása, és a részek mozgatása a memória és a háttér tároló között.

Ma már sokkal nagyobb ugyan a memória, de még sokkal nagyobb lehet a **címtartomány (address space)**.

Máté: Architektúrák

12. előadás

2

Virtuális címtartomány: azok a címek, amelyekre a program hivatkozni tud.

Fizikai címtartomány: azok a címek, amelyek tényleges memória cellát címeznek.

A virtuális és fizikai címtartomány ugyanolyan méretű lapokra van osztva (**6.3. ábra**). A fizikai „lapokat” **lapkeretnek (page frame)** nevezzük.

Lap méret: 512 B – 64 KB (– 4 MB), **mindig 2 hatványa.**

Máté: Architektúrák

12. előadás

3

Lap	Virtuális címek	Lapkeret	Fizikai címek
N	-	n	-
...	... - - ...
4	16384 - 20479	4	16384 - 20479
3	12288 - 16383	3	12288 - 16383
2	8192 - 12287	2	8192 - 12287
1	4096 - 8191	1	4096 - 8191
0	0 - 4095	0	0 - 4095

A virtuális címtartomány sokkal nagyobb, mint a fizikai!

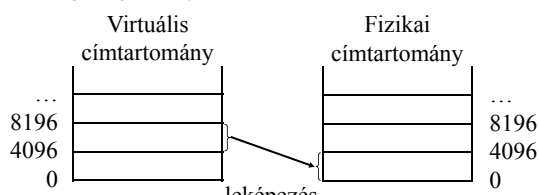
Mit kell tenni, ha olyan címre történik hivatkozás, amely nincs a memóriában?

Máté: Architektúrák

12. előadás

4

1. Egy lapkeret (pl. a 0-4095) tartalmának lemezre mentése.
2. A kérdéses lap megkeresése a lemezen.
3. A kérdéses lap betöltése a lapkeretbe.
4. A memória térkép megváltoztatása: pl. a 4096 és 8191 közötti címek leképezése a betöltött lapkeret címtartományába.
5. A végrehajtás folytatása.



6.2. ábra

Máté: Architektúrák

12. előadás

5

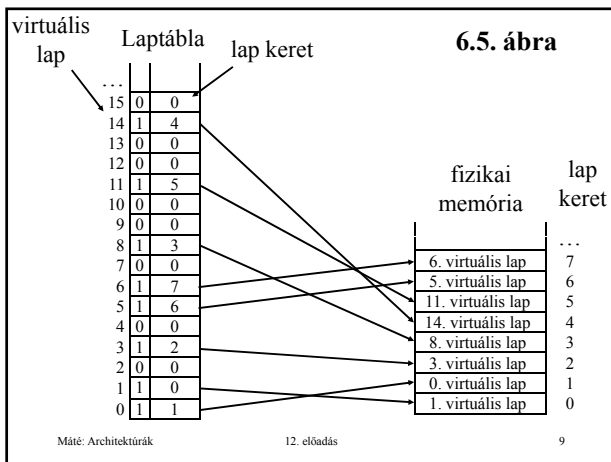
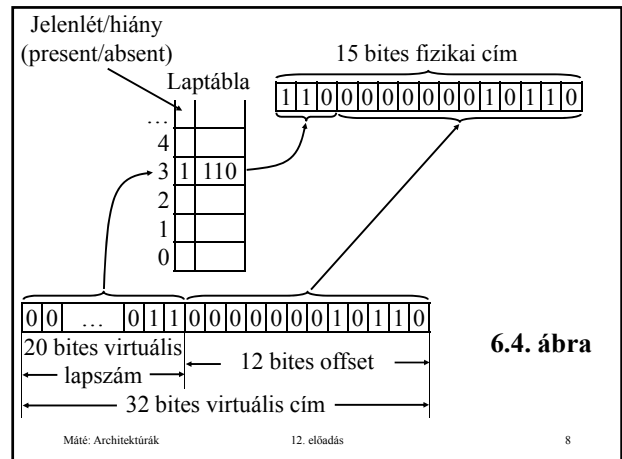
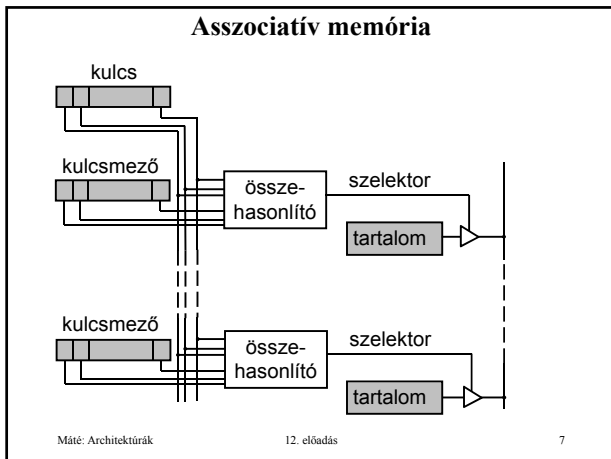
A virtuális címek fizikai címekre történő leképezését az **MMU (Memory Management Unit – memória kezelő egység)** végzi.

Memória térkép (memory map) vagy laptábla (page map) kapcsolja össze a virtuális címeket a fizikai címekkel. Pl. 4 KB-os lapméret és 32 bites virtuális cím esetén 1 millió virtuális lap van, ezért 1 millió bejegyzésű laptáblára van szükség. 32 KB fizikai memória esetén csak 8 lapkeret van, ezért a leképezés megoldható 8 cellás asszociatív memóriával is (a gyakorlatban több ezer lapkeret van, és az asszociatív memória igen drága).

Máté: Architektúrák

12. előadás

6



Laphiány (page fault): a lap nincs a memóriában.

Kérésre lapozás (demand paging): lapozás csak laphiány esetén. A program egyetlen bájta sem kell bent legyen a memóriában, csak a másodlagos tárolón.

Máté: Architektúrák 12. előadás 10

Munka halmaz (working set): az operációs rendszer feladata megállapítani

Tanenbaum: a legutóbbi k memória hivatkozásban szereplő lapok halmaza.

Wikipedia: a legutóbbi τ időben hivatkozott memória lapok halmaza.

Ha a munkahalmaz nagyobb, mint a lapkeretek száma, akkor gyakori lesz a laphiány. A nagyon gyakori laphiányt **vergődésnek (thrashing)** nevezzük.

Időosztásos rendszerekben a kérésre lapozás nem kielégítő, a munkahalmazban lévő a lapok előre visszatölthetők.

Máté: Architektúrák 12. előadás 11

Lapkezelési eljárások: melyik lap helyett töltsük be a kért lapot?

LRU (Least Recently Used, legrégebben használt): általában jó, de nem jó pl. 9 lapon átnyúló ciklus esetén, ha csak 8 memória lap van (**6.6. ábra**).

7. Virtuális lap	7. Virtuális lap	7. Virtuális lap
6. Virtuális lap	6. Virtuális lap	6. Virtuális lap
5. Virtuális lap	5. Virtuális lap	5. Virtuális lap
4. Virtuális lap	4. Virtuális lap	4. Virtuális lap
3. Virtuális lap	3. Virtuális lap	3. Virtuális lap
2. Virtuális lap	2. Virtuális lap	2. Virtuális lap
1. Virtuális lap	1. Virtuális lap	0. Virtuális lap
0. Virtuális lap	8. Virtuális lap	8. Virtuális lap

Máté: Architektúrák 12. előadás 12

FIFO (First-in First-Out, először be, először ki): egyszerűbb (de most ez se jobb, mint LRU).

Csak a módosult (**dirty**, szennyezett) lapokat kell visszaírni, a tisztát (**clean**) nem (**szennyezés bit**). Most is előnyös, ha az utasítások és az adatok elkülönülten helyezkednek el a memóriában: az utasításokat nem kell visszaírni.

Máté: Architektúrák 12. előadás 13

Lapméret és elaprózódás

Ha egy program k lapon fér el, akkor általában a k -dik lap nincs tele.

Ha a lap mérete n , akkor programonként átlagosan $n/2$ bájt kihasználatlan:

belső elaprózódás (internal fragmentation).

A belső elaprózódás ellen a lap méretének csökkentésével lehet védekezni, de ez a laptábla méretének növekedéséhez vezet.

A kis lap előnytelen a lemez sávszélességének kihasználása szempontjából is, viszont kisebb a vergődés kialakulásának valószínűsége.

Máté: Architektúrák 12. előadás 14

Szegmentálás

Egy fordítóprogramnak a következő célokra kellhet memória (6.7. ábra):

- szimbólum tábla,
- forrás kód,
- konstansok,
- elemzési fa,
- verem.

Rögzített memória felosztás esetén ezek egyike kicsinek bizonyulhat, miközben a többi nem használja ki a rendelkezésére álló tartományt.

Virtuális címtartomány

Máté: Architektúrák 12. előadás 15

Szegmentálás (6.8. ábra)

Szegmentált memóriában minden tábla a többbitől függetlenül nőhet vagy zsugorodhat.

Máté: Architektúrák 12. előadás 16

Szegmens (6.9. ábra)

A programozó számára látható logikai egység. Minden szegmens címtartománya 0-tól valamilyen maximumig terjed. A szegmens tényleges mérete ennél kisebb lehet. A program számára a címtartomány két dimenziós: (szegmens, offset).

Általában egy szegmensben csak egyféle dolgok vannak: vagy kód vagy konstans vagy ...

Különböző tárvédelmi lehetőségek:

- kód: csak végrehajtható, nem írható, nem olvasható,
- konstans: csak olvasható
- ...

Máté: Architektúrák 12. előadás 17

A szegmentálás és a virtuális memória összehasonlítása (6.9. ábra)

Szemponok	Lapozás	Szegmentálás
Tudnia kell róla a programozónak?	Nem	Igen
Hány lineáris címtartomány létezik?	1	Több
Meghaladhatja-e a virtuális címtartomány nagysága a fizikai memória méretét?	Igen	Igen
Könnyen kezelhetők a változó méretű táblák?	Nem	Igen
Mi ennek a technikának a lényege?	Nagy memória szimulálása	Több címtartomány biztosítása

Máté: Architektúrák 12. előadás 18

A szegmentálás megvalósítása

Lapozással: Minden szegmensnek saját laptáblája van. A szegmens néhány lapja a memóriában van.

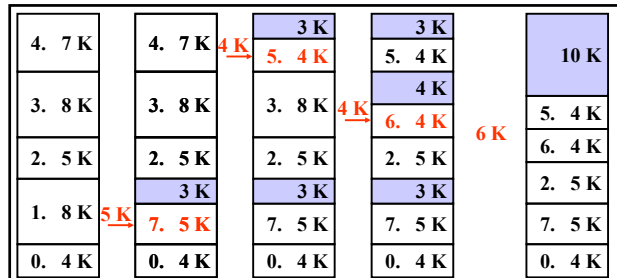
Cseréléssel: Teljes szegmensek mozognak a memória és a lemez között. Ha olyan szegmensre hivatkozunk, amely nincs a memóriában, akkor betöltődik. **Külső elaprózódáshoz (external fragmentation)** vezethet (6.10. ábra).

Lyukacsosodásnak (checkerboarding) is nevezik.

Máté: Architektúrák

12. előadás

19



Összepréselés: idő igényes, de időnként kell.

Legjobb illesztés (best fit) és **első illesztés (first fit)** algoritmus. Az utóbbi gyorsabb és jobb is az általános hatékonyság szempontjából.

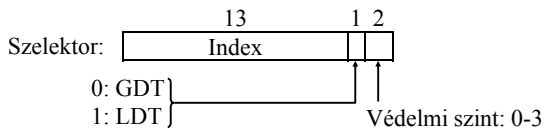
Máté: Architektúrák

12. előadás

20

Pentium 4 (6.12-14. ábra)

A szegmens regiszter tartalmazza a szelektort.



A szelektor (6.12. ábra) indexe választja ki a leíró (descriptor) a lokális (LDT, Local Descriptor Table) vagy globális leíró táblából (GDT, Global Descriptor Table). (6.13. ábra).

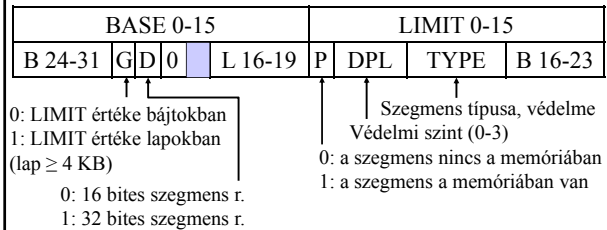
A 0. leíró használata csapdát eredményez (hiba).

Máté: Architektúrák

12. előadás

21

Pentium 4 kódszegmensének leírója (6.13. ábra)

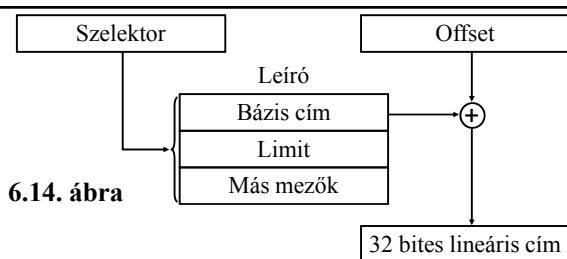


Ha **P=0**, csapda: nem létező szegmens, vagy be kell tölteni a szegmenst.

Máté: Architektúrák

12. előadás

22



Ha offset (a szegmens elejéhez viszonyított relatív cím) a szegmens határán túl van, csapda (hiba).

Lapozást tiltó flag (a globális vezérlőregiszter bitje):

Ha engedélyezett: lineáris cím = virtuális cím
Ha tiltott: lineáris cím = fizikai cím

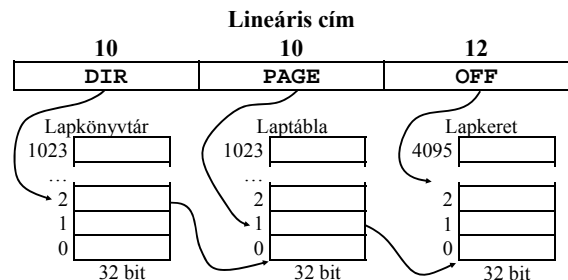
Máté: Architektúrák

12. előadás

23

Lapkönyvtár (page directory 6.15. ábra)

A 32 bites lineáris címek és a 4 KB-os lapok miatt egy szegmenshez egymillió lap is tartozhat. Túl sok!
Minden futó programhoz egy **lapkönyvtár** tartozik.
Minden bejegyzés egy **laptáblára** mutat, vagy sehova.



Máté: Architektúrák

12. előadás

24

A lapkönyvtárnak azokhoz a mutatóihoz, amelyek nem mutatnak sehova, nem kell helyet foglalni a laptábla számára (pl. csak két db. 1 K, és nem egy 1 M bejegyzésű tábla kell egy 4 MB-nál rövidebb szegmenshez).

A táblákban minden bejegyzéshez 32 bit áll rendelkezésre. A mutatókhoz nem használt biteket a hardver az operációs rendszer számára hasznos jelzésekkel tölti ki (védelem, szennyezettség, hozzáférés, ...).

Speciális hardver támogatja a legutóbb használt lapok gyorsabb elérését.

Máté: Architektúrák 12. előadás 25

A Pentium 4 védelmi rendszere (6.16. ábra)

A futó program pillanatnyi szintjét a **PSW** tartalmazza.

A program a saját szintjén lévő szegmenseket szabadon használhatja.

Magasabb szinten lévő adatokhoz hozzáfér, de az alacsonyabb szinten lévő kezelés csapdát okoz.

Más szinten lévő eljárás hívásánál **CALL** helyett szelektort kell alkalmazni, ez egy **hívás kaput (call gate)** jelöl ki (más védelmi szintre csak szabványos – tehát ellenőrzött – belépési ponton lehet áttérni).

A szintek egy lehetséges felhasználása:

Máté: Architektúrák 12. előadás 26

Az UltraSPARC III virtuális memóriája

Virtuális cím 64 bites, egyelőre 44 bitre korlátozva.

Virtuális címtartomány megengedett zónák

44 bitre korlátozva ez a címtartomány folytonos.

Fizikai címtartomány maximum 41 bites.

A kód és adat lapokat külön kezeli.

Máté: Architektúrák 12. előadás 27

Lapméret: 8, 64, 512 KB és 4 MB (6.17. ábra).

Lap mérete	Virtuális lap címe (bit)	OFFSET (bit)	Fizikai lap címe (bit)	OFFSET (bit)
8 KB	51 (31)	13	28	13
64 KB	48 (28)	16	25	16
512 KB	45 (25)	19	22	19
4 MB	42 (22)	22	19	22

↑ 44 bitre korlátozva ↑ maximum 41 bit

Máté: Architektúrák 12. előadás 28

A memória kezelő egység (**MMU**) három szinten dolgozik:

- A legutóbb használt lapokat gyorsan megtalálja (hardver). A kód és az adat lapokat teljesen külön kezeli.
- A nem nagyon régen használtakat már lassabban (hardver segítségével).
- A nagyon régen használtakat csak hosszas keresés után (szoftveres úton).

Máté: Architektúrák 12. előadás 29

TLB (Translation Lookaside Buffer) a legutóbb használt 64 lap bejegyzését tartalmazza (6.18. ábra).

Környezet (context): processzus szám.

Asszociatív memória: Kulcs a keresett virtuális lap és a környezet.

TLB hiány (TLB miss) esetén: csapda.

Máté: Architektúrák 12. előadás 30

TLB hiány esetén TSB folytatja a keresést (szoftver).
TSB (Translation Storage Buffer): olyan felépítésű, mint egy direkt leképezésű gyorsító tár (operációs rendszer építi fel, és kezeli a központi memóriában).

The diagram shows a table with columns for 'Virtuális lap címé' (Virtual page address) and 'Virtuális lap' (Virtual page). The 'Virtuális lap címé' column is further divided into 'tag' and 'line'. The 'Virtuális lap' column is divided into 'Érvényes' (Valid), 'tag', 'Környezet' (Environment), 'Fizikai lapkeret' (Physical page frame), and 'Flag-ek' (Flags). Below this, a grid represents the mapping of virtual pages to physical frames.

TSB találat esetén egy TLB sor helyébe beíródik a kért lapnak megfelelő bejegyzés.

Máté: Architektúrák 12. előadás 31

TSB hiány esetén a fordítótábla (**translation table**) alapján keres. Ennek a táblának a szerkezetét az operációs rendszer határozza meg.

Egy lehetséges megoldás a tördeléses eljárás. Ebben az esetben a memóriába töltött virtuális lapok és a nekik megfelelő fizikai lapkeretek sorszáma listákba van helyezve. Ha a virtuális lap sorszáma p -vel osztva q -t ad maradékul, akkor csak a q -adik listát kell végignézni.

Ha ez se találja a keresett lapot, akkor nincs a memóriában.

Máté: Architektúrák 12. előadás 32

Virtuális memória és gyorsító tár

Két szintű hierarchia:
 Virtuális memória használatakor az egész programot lemezen tartjuk, fix méretű lapokra osztjuk. Lap hiány esetén a lapot a központi memóriába töltjük (operációs rendszer).
 Gyorsító tár esetén a központi memóriát gyorsító sorokra osztjuk. Gyorsító sor hiány esetén a gyorsító sort a gyorsító tárba töltjük (hardver).

Máté: Architektúrák 12. előadás 33

Mágneslemez (2.19. ábra)

I/O fej: vékony légrés választja el a lemeztől.

The diagram shows a cross-section of a magnetic disk head. It labels the 'Szeletörés' (Sector cut), 'Lemez forgásának iránya' (Direction of disk rotation), 'Fejléc' (Sector head), 'Író/olvasó fej' (Write/read head), and 'Kar' (Arm). It also indicates '4096 adatbit' (4096 data bits) and '1 bit szélessége 0,1-0,2 mikron' (1 bit width 0.1-0.2 microns). A note states 'A sáv szélessége 1-2 mikron' (Track width 1-2 microns).

Sáv (track, 5000-10000 sáv/cm),
Szelet (tipikusan 512B, 50.000-100.000 bit/cm), pl.: fejléc + 4096 bit (= 512B) adat + hibajavító kód (Hamming vagy Reed-Solomon).

Máté: Architektúrák 12. előadás 34

Szeletorés: hogy az írás ne rontsa el a szomszédos szeletet.

Formázott és formázatlan kapacitás.
Winchester lemez (IBM), légmentesen lezárt.
 Kezdetben 30 MB fix + 30 MB cserélhető.
 Az átmérő régen 50 cm, mostanában 3 – 12 cm közötti, söt, kisebb is lehet.

Lemezegység (2.20. ábra): közös tengelyen több (6-12) lemez. **Cilinder.**

Máté: Architektúrák 12. előadás 35

Keresési idő: sáv/cilinder keresés (seek) 5-10 ms.
Forgási késleltetés: átlagosan egy fél fordulat ideje, 3-6 ms (60-180 fordulat/sec).
Átviteli sebesség: 20-40 MB/sec.
 Maximális ↔ átlagos

Írás sűrűség:
Régen: belül maximális, kifelé egyre kisebb (forgás szög alapján).
Jelenleg: 10-30 zóna, a külső zónákban több szeletorés van egy sávon (2.21. ábra).

Máté: Architektúrák 12. előadás 36

Lemezvezérlő: vezérli a hardvert, nyilvántartja és átcímzi a hibás sávokat.
Szoftver parancsokat hajt végre: kar mozgatás, **READ, WRITE, FORMAT**, ... utasítások.
További feladatai: hiba felismerés/javítás, soros – párhuzamos és párhuzamos – soros átalakítás.

Máté: Architektúrák

12. előadás

37

Hajlékony (floppy) lemez: szerviz célokra (karbantartási információk tárolására) találták ki. Az I/O fej hozzáér a lemezhez: gyorsan kopik, ezért leáll, ha éppen nincs feladata. Kb. 0.5 s, míg a lemez fölpörög.

Máté: Architektúrák

12. előadás

38

Lemez vezérlés

PC-ken kezdetben CPU regiszterekbe töltött fej, cylinder, szektor címek alapján a **BIOS** (Basic Input Output System) vezérelt.

Seagate lemezegység: 20 bites szektor cím.

4 fej (4 bit), 306 cylinder (10 bit)

és sávonként 17 db 512 bájtos szektor (6 bit).

Később kevés lett 10 bit a cylinder címzésére.

IDE (Integrated Drive Electronics, max. 504 MB): a meghajtóba integrált vezérlő. Seagate kompatibilis! „Hazudnak” a **BIOS**-nak.

A címet a vezérlő fej-cylinder-szektor címre fordítja.

Máté: Architektúrák

12. előadás

39

EIDE (Extended IDE): **LBA** (logikai blokk címzés - Logical Block Addressing). Cím: $0 - 2^{28} - 1$.
Maximum 128 GB

ATA-3 (AT Attachment, AT kiegészítő), majd

ATAPI-4 (ATA Packet Interface, ATA-csomaginterfész) 33 MB/s

ATAPI-5 66 MB/s

ATAPI-6 100 MB/s, 48 bites szektor cím

Máté: Architektúrák

12. előadás

40

ATAPI-7 A korábbi 80 vezetékes szalagkábel helyett 7 vezetékes kerek kábelt alkalmaz (PCI express): jobb a légáramlás.
Kezdetben 150 MB/s soros átvitel, ami várhatóan hamarosan 1,5 GB/s fölé emelkedik.
5 V helyett 0.5 V: kisebb energia fogyasztás.

Máté: Architektúrák

12. előadás

41

Feladatok

Milyen utasítások érhetők el operációs rendszer szinten?

Mi az overlay technika lényege?

Mi a virtuális címtartomány?

Mi a fizikai címtartomány?

Mi a lap és mi a lapkeret?

Mi a lapozás?

Mi a memória térkép (laptábla)?

Mi az MMU?

Hogy működik az asszociatív memória?

Mi a laphiány?

Mi a kérésre lapozás?

Mi a munka halmaz (working set)?

Máté: Architektúrák

12. előadás

42

Feladatok

Mikor alakul ki vergődés?
Milyen lapkezelési eljárásokat ismer?
Mi a belső elaprózódás?
Mi az előnye, és mi a hátránya a kis lapméretnek?
Mit nevezünk szegmentálásnak?
Hogy valósítható meg a szegmentálás?
Mik a szegmentálás előnyei?
Mi a külső elaprózódás?
Mi az összepréselés (compaction)?
Hogy valósul meg a szegmens címzés a Pentium 4-en?
Mi a szelektor?
Milyen információt tartalmaz a Pentium 4 szelektora?
Milyen mezőket tartalmaz a kódszegmensek leírója?

Máté: Architektúrák

12. előadás

43

Feladatok

Mire szolgál az LDT (Local Descriptor Table) és a GDT (Global Descriptor Table)?
Hogy képződik a lineáris cím?
Hogy valósul meg Pentium 4-en a virtuális címzés?
Milyen a Pentium 4 védelmi rendszere?
Hogy hívható más védelmi szintű eljárás?
Jellemezze az UltraSparc III virtuális memóriáját!
Mi a TLB (Translation Lookaside Buffer)?
Milyen memóriában van a TLB?
Mi történik TLB hiány esetén?
Hogy szervezett a TSB (Translation Storage Buffer)?
Mi történik TSB hiány esetén?
Hasonlítsa össze a virtuális memóriát a gyorsító tárral!

Máté: Architektúrák

12. előadás

44

Feladatok

Hogy vannak tárolva az adatok a mágneslemezen?
Mi a sáv?
Mi a szektor?
Milyen adatokat tartalmaz egy szektor?
Miért van szükség a szektor részre?
Mi a formázatlan és formázott kapacitás?
Mi a lemez egység?
Mi a cilinder?
Milyen késleltetések lépnek fel a mágneslemez használatánál?
Mi a keresési idő?
Mi a forgási késleltetés?
Mi az átviteli sebesség?

Máté: Architektúrák

12. előadás

45

Feladatok

Mi az írássűrűség?
Mi a zónákra osztás szerepe?
Milyen feladatai vannak a lemezvezérlőnek?
Jellemezze a hajlékony lemezt (floppy)!
Milyen lemez vezérlést ismer?
Mi az IDE vezérlő fő jellemzője?
Mi az EIDE vezérlő fő jellemzője?
Mi az LBA lényege?
Jellemezze az ATA-3, ATAPI-4, -5, -6 vezérlőket!
Jellemezze az ATAPI-7 vezérlőt!

Máté: Architektúrák

12. előadás

46

Az előadáshoz kapcsolódó

Fontosabb tételek

Operációs rendszer szintje. Virtuális memória.
Virtuális memória. Lapméret, elaprózódás
Szigmentálás. A Pentium 4 és az UltraSPARC III virtuális memóriája
Mágneslemez, lemezvezérlők

Máté: Architektúrák

12. előadás

47