

Rekurzív eljárások megvalósításához veremre van szükség. Minden hívás esetén az eljárás paramétereit a verembe kell tenni, és ott kell elhelyezni a lokális változókat is!

Eljárás prológus: a régi verem keret mutató (**FP**) elmentése, új verem keret mutató megadása, és a verem mutató (**SP**) növelése, hogy legyen hely a veremben a lokális változók számára.

Eljárás epilógus: visszatéréskor a verem kitakarítása.

Máté: Architektúrák 12. előadás 1

A Hanoi tornyai probléma megoldása

5.47. ábra: Pentium 4 program.

5.48. ábra: UltraSPARC III program.
Eltolás rés!

Máté: Architektúrák 12. előadás 2

Vezérlési folyamat

- Szekvenciális vezérlés (5.39. ábra)
- Elágazás.
- Eljárás: 5.44. ábra.
- Megszakítások.
- Csapdák.
- Korutínok: 5.45. ábra. Párhuzamos feldolgozás szimulálására alkalmas egy CPU-s gépen.

g`oto` helyett jobb a ciklus vagy az eljárás alkalmazása!

Máté: Architektúrák 12. előadás 3

A Pentium 4 utasításai

- Egész utasítások legnagyobb része: 5.34. ábra.
- Egyéb utasítások (pl. lebegőpontosak).

Az UltraSPARC III utasításai

Összes egész utasítás: 5.35. ábra.
A utasításnévben **CC**: beállítja a feltételkódot.
ADD, ADDC, ADDCC, ADDCCC utasítások.
Szimulált utasítások (5.36. ábra), pl.:
MOV SRC, DST ≡ OR SRC, G0, DST

A 8051 utasításai (5.37. ábra)

Bit utasítások, pl. a 43. bit 1-re állítása:
SETB 43

Máté: Architektúrák 12. előadás 4

A feltételes ugró utasítások eldugaszolják a csővezeték

- Feltételes végrehajtás
- Predikáció

Máté: Architektúrák 12. előadás 5

Feltételes végrehajtás (5.51-52. ábra):

<code>if(R1 == 0)</code> <code>R2 = R3;</code>	<code>CMP R1, 0</code> <code>BNE L1</code> <code>MOV R2, R3</code> L1: ...	<code>CMOVZ R2, R3, R1</code>
<code>if(R1 == 0) {</code> <code>R2 = R3;</code> <code>R4 = R5;</code> <code>} else {</code> <code>R6 = R7;</code> <code>R8 = R9;</code> <code>}</code>	<code>CMP R1, 0</code> <code>BNE L1</code> <code>MOV R2, R3</code> <code>MOV R4, R5</code> <code>BR L2</code> L1: MOV R6, R7 MOV R8, R9 L2: ...	<code>CMOVZ R2, R3, R1</code> <code>CMOVZ R4, R5, R1</code> <code>CMOVN R6, R7, R1</code> <code>CMOVN R8, R9, R1</code>

CMOVZ R2, R3, R1 csak akkor hajtja végre `R2 = R3 - t`, ha `R1 = 0`.

Máté: Architektúrák 12. előadás 6

Predikáció, IA – 64 (5. 53. ábra)

64 predikátum regiszter: 1 bites regiszterek, többnyire párban. Az IA – 64 minden utasítása predikátumos. CMPEQ R1, R2, P4 beállítja P4-et és törli P5-öt, ha R1 = R2, különben P5-öt állítja be és P4-et törli.

if(R1 == R2) R3 = R4 + R5; else R6 = R4 – R5;	CMP R1, R2 BNE L1 MOV R3, R4 ADD R3, R5 BR L2 L1: MOV R6, R4 SUB R6, R5 L2: ...	CMPEQ R1, R2, P4 <P4>ADD R3, R4, R5 <P5>SUB R6, R4, R5
--	--	--

Máté: Architektúrák
12. előadás
7

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
8

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
9

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
10

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
11

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

Virtuális címtartomány

Fizikai címtartomány

leképezés

6.2. ábra

Máté: Architektúrák
12. előadás
12

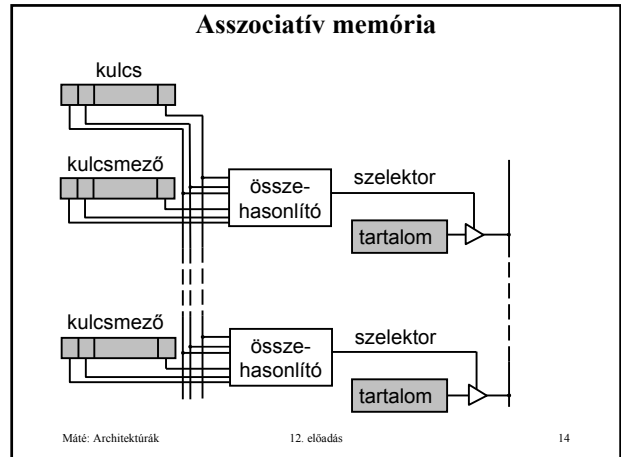
A virtuális címek fizikai címekre történő leképezését az MMU (Memory Management Unit – memória kezelő egység) végzi.

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

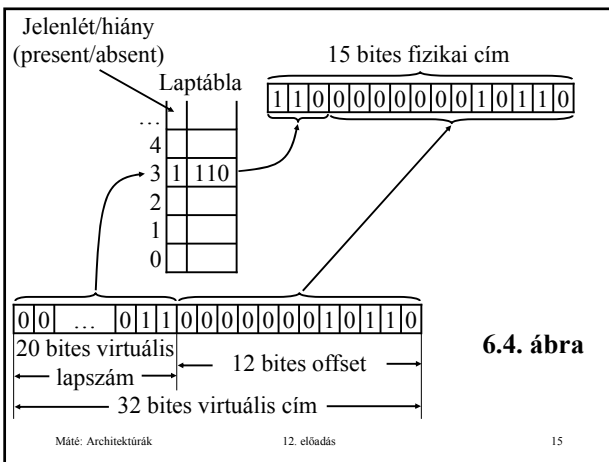
13



Máté: Architektúrák

12. előadás

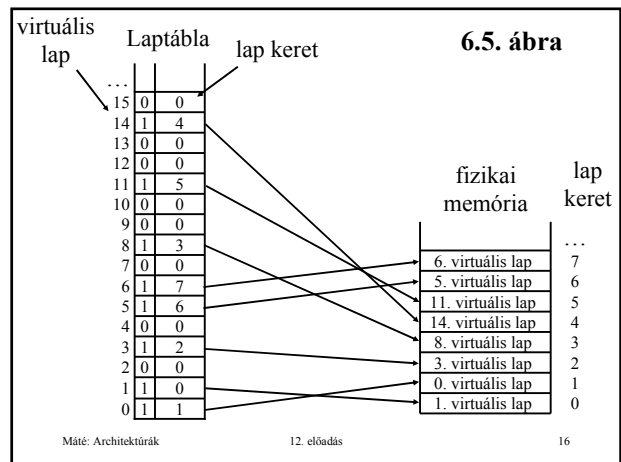
14



Máté: Architektúrák

12. előadás

15



Máté: Architektúrák

12. előadás

16

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

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

Időosztásos rendszereknél nem kielégítő!

Munka halmaz (working set): a legutóbbi k memória hivatkozásban szereplő lapok halmaza (az operációs rendszer feladata megállapítani). Időosztásos rendszerekben ezek a lapok előre visszatölthetők.

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

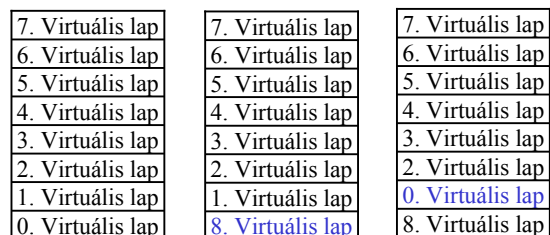
Máté: Architektúrák

12. előadás

17

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**).



Máté: Architektúrák

12. előadás

18

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 19

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 $n/2$ bájtt 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 20

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 21

Szegmentálás (6.8. ábra)

Szegmentált memóriában minden tábla a többitől függetlenül nőhet vagy zsugorodhat.

Máté: Architektúrák 12. előadás 22

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 23

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 24

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
25

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
26

Pentium 4 (6.12-14. ábra)

A szegmens regiszter tartalmazza a szelektort.

Szelektor:	13 Index	1	2
------------	-------------	---	---

0: GDT }
1: LDT } Védelmi szint: 0-3

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
27

Pentium 4 kódszegmensének leírója (6.13. ábra)

BASE 0-15				LIMIT 0-15			
B 24-31	G	D 0	L 16-19	P	DPL	TYPE	B 16-23

0: LIMIT értéke bájtokban
1: LIMIT értéke lapokban (lap ≥ 4 KB)
0: 16 bites szegmens r.
1: 32 bites szegmens r.

Szegmens típusa, védelme
Védelmi szint (0-3)
0: a szegmens nincs a memóriában
1: a szegmens a memóriában van

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

Máté: Architektúrák
12. előadás
28

Szelektor

Offset

Leíró	
Bázis cím	+
Limit	
Más mezők	

6.14. ábra

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
29

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.

Lineáris cím

10 DIR	10 PAGE	12 OFF
-----------	------------	-----------

Lapkönyvtár

1023	
...	
2	
1	
0	

32 bit

Laptábla

1023	
...	
2	
1	
0	

32 bit

Lapkeret

32 bit

Máté: Architektúrák
12. előadás
30

A lapkönyvtárnak azokhoz a mutatóihoz, amelyek nem mutatnak sehova, nem kell helyet foglalni a laptábla számára (pl. csak két db. ezer, és nem egy milliós 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 31

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 lévőek kezelése 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 32

Az UltraSPARC III virtuális memóriája

Virtuális cím 64 bites, egyelőre 44 bitre korlátozva.

44 bitre korlátozva ez a címtartomány folytonos.

Fizikai címtartomány maximum 41 bites.

A kód és adat lapokat külön kezeli.

Máté: Architektúrák 12. előadás 33

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 34

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 segítséggel).
- A nagyon régen használtakat csak hosszas keresés után (szoftveres úton).

Máté: Architektúrák 12. előadás 35

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 36

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).

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

Máté: Architektúrák 12. előadás 37

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 38

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 tár hiány esetén a gyorsító sort a gyorsító tárba töltjük (hardver).

Máté: Architektúrák 12. előadás 39

Mágneslemez (2.19. ábra)
I/O fej: vékony légrés választja el a lemeztől.

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

Máté: Architektúrák 12. előadás 40

Szektor rés: hogy az írás ne rontsa el a szomszédos szektort.
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, sőt, kisebb is lehet.
Lemezegység (2.20. ábra): közös tengelyen több (6-12) lemez. **Cilinder.**

Máté: Architektúrák 12. előadás 41

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 szektor van egy sávon (**2.21. ábra**).

Máté: Architektúrák 12. előadás 42

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

43

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

44

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 címezésére.

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

45

EIDE (Extended IDE): **LBA** (logikai blokk címezés - 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

46

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

47

SCSI (Small Computer System Interface) **lemezek:** sokkal gyorsabb átvitelt biztosít (**2.22. ábra**), drágábbak is.

SCSI: sín, vezérlő + maximum 7 (15) **SCSI** eszköz (lemez, nyomtató, CD, ...) csatlakozó.

A sín „átmegy” az eszközökön: az eszközöknek van egy bemenő és egy kimenő csatlakozója.

A visszaverődő jelek kiszűrése miatt az utolsó eszközön a sánt le kell zárni.

Minden eszköznek 0-7 (15) közötti azonosítója van.

Egyszerre több eszköz is aktív lehet (**EIDE:** csak egy).

Máté: Architektúrák

12. előadás

48

Feladatok

Mi az eljárás prológus?
Mi az eljárás epilógus?
Mi az eltolás rész?
Mit nevezünk korutinnak (társrutin, coroutine)?
Mit értünk feltételes végrehajtáson?
Mi a feltételes végrehajtás előnye?
Mit értünk predikáción?
Hogy küszöböli ki a feltételes végrehajtás és a predikáció a csővezeték elakadását? Jelent-e ez késleltetést a program futásában?

Máté: Architektúrák

12. előadás

49

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

50

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

51

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

52

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 cylinder?
Milyen késleltetések lépnek fel a mágneslemez használatánál?
Mi a keresési idő?
Mi a forgási késleltetés?
Mi az átviteli sebesség?

Máté: Architektúrák

12. előadás

53

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!
Mi a SCSI?

Máté: Architektúrák

12. előadás

54

Az előadáshoz kapcsolódó

Fontosabb tételek

Vezérlési folyamat. Korutinok,
Feltételes végrehajtás, predikáció
Operációs rendszer szintje. Virtuális memória.
Virtuális memória. Lapméret, elaprózódás
Szegmentálás. A Pentium 4 és az UltraSPARC III
virtuális memóriája
Mágneselemek, lemezvezérlők, SCSI