

# Architektúra, címezési módok

Koschek Vilmos

vkoschek@vonalkod.hu

## Miről lesz szó?

- Címzés fogalma, címezési módok
- Virtuális tárkezelés
- Példa: Intel

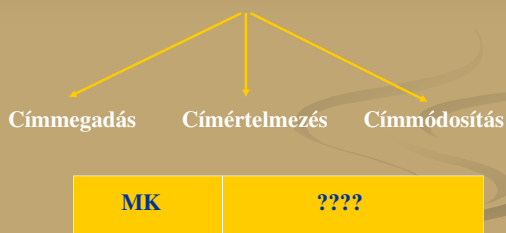
Címzés

Koschek Vilmos

2

## Fogalom

A címezési mód az az út (algorithmus), ahogyan az operandus címét meghatározzuk.



Címzés

Koschek Vilmos

3

## Címértelmezés

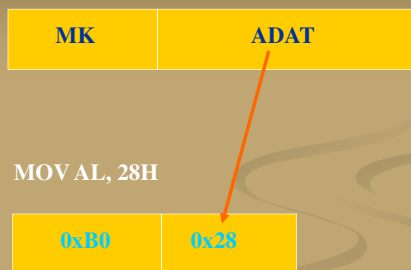
- Adat
  - Közvetlen adatmegadás
  - Rejtett címzés
- Cím
  - Közvetlen címzés
  - Közvetett címzés
- Valós/ virtuális (később !)

Címzés

Koschek Vilmos

4

## Közvetlen (immediate) adatmegadás

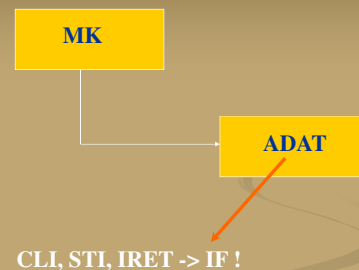


Címzés

Koschek Vilmos

5

## Rejtett (inherent), vagy implicit címzés



Címzés

Koschek Vilmos

6

## Közvetlen (direkt) címzés

MK	CÍM
	113
	112
	111
	110
	109
	108
	107
	106
	105
	104
	103
	102
	101
	100

MOV AL, [106]

AL = 0x33

Címzés Kosztek Vilmos 7

## Közvetett (indirekt) címzés

MK	CÍM
	113
	112
	111
	110
	109
	108
	107
	106
	105
	104
	103
	102
	101
	100

MOV AL, [SI]

AL=0x33

Indirekció

Többszörös is lehet!

Címzés Kosztek Vilmos 8

## Cím megadás

- Abszolút
- Rövidített (relatív)
  - Lapcímes
  - Bázisregiszteres

Címzés Kosztek Vilmos 9

## Abszolút

MK	CÍM
	113
	112
	111
	110
	109
	108
	107
	106
	105
	104
	103
	102
	101
	100

- Címezhető tár
- Méret
- Sebesség

Címzés Kosztek Vilmos 10

## Rövidített

MK	Cím
	113
	112
	111
	110
	109
	108
	107
	106
	105
	104
	103
	102
	101
	100

Bázis

Cím

13

- Címezhető tár
- Méret
- Sebesség
- Lapcímes
- Relatív

Címzés Kosztek Vilmos 11

## Lapcímes

MK	Lapon belüli cím	254
	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
	11	11
	12	12
	13	13
	14	14
	15	15
	16	16
	17	17
	18	18
	19	19
	20	20
	21	21
	22	22
	23	23
	24	24
	25	25
	26	26
	27	27
	28	28
	29	29
	30	30
	31	31
	32	32
	33	33
	34	34
	35	35
	36	36
	37	37
	38	38
	39	39
	40	40
	41	41
	42	42
	43	43
	44	44
	45	45
	46	46
	47	47
	48	48
	49	49
	50	50
	51	51
	52	52
	53	53
	54	54
	55	55
	56	56
	57	57
	58	58
	59	59
	60	60
	61	61
	62	62
	63	63
	64	64
	65	65
	66	66
	67	67
	68	68
	69	69
	70	70
	71	71
	72	72
	73	73
	74	74
	75	75
	76	76
	77	77
	78	78
	79	79
	80	80
	81	81
	82	82
	83	83
	84	84
	85	85
	86	86
	87	87
	88	88
	89	89
	90	90
	91	91
	92	92
	93	93
	94	94
	95	95
	96	96
	97	97
	98	98
	99	99
	100	100
	101	101
	102	102
	103	103
	104	104
	105	105
	106	106
	107	107
	108	108
	109	109
	110	110
	111	111
	112	112
	113	113

Lapcímes

Lap címes:

- Lap
- Lapcím
- Eltolás, előjel nélküli
- PC, 0,

Lapon belüli cím

2

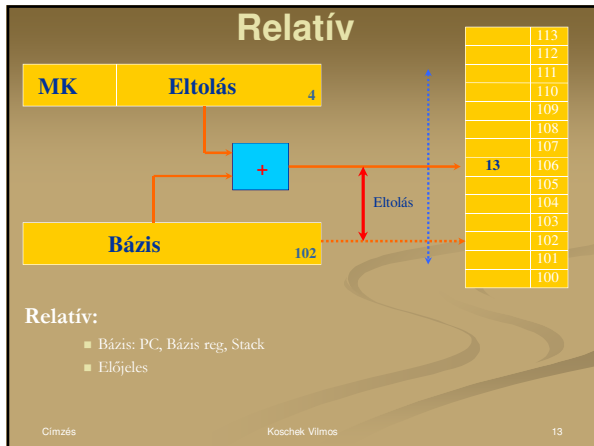
0.lap

1.lap

2.lap

3.lap

Címzés Kosztek Vilmos 12



## Címmódosítás

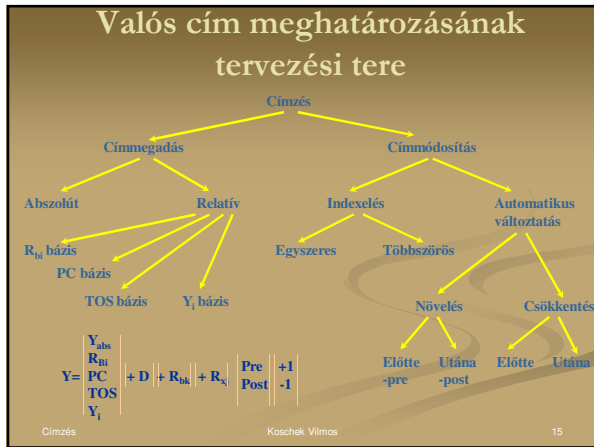
**Indexelés:** egymás utáni, azonos típusú adatok (tömb) címzésére szolgál.

**MOV AL, CIM [SI]**

**Automatikus módosítása:**

- Autoinkrementálás / autodekrementálása
- Címzés előtt (pre) / után (post)

Címzés      Koschek Vilmos      14



## Intel (16/32) processzorok

- Programozható regiszterek
- Utasításforma
- Címzési lehetőségek

Címzés      Koschek Vilmos      16

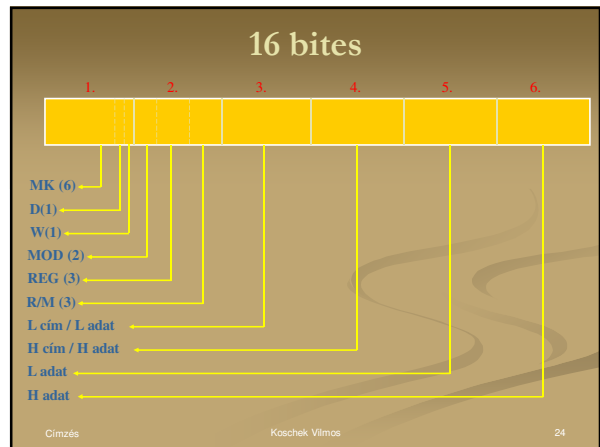
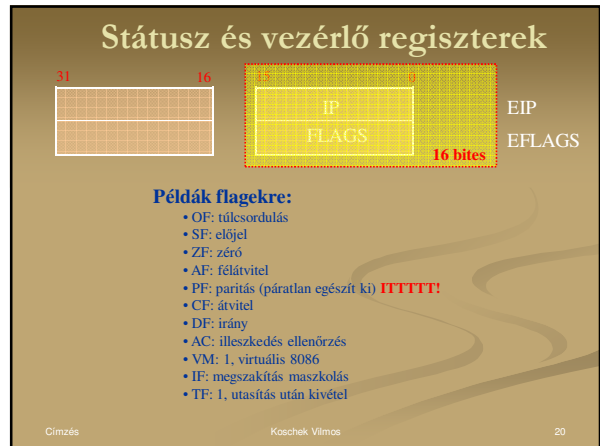
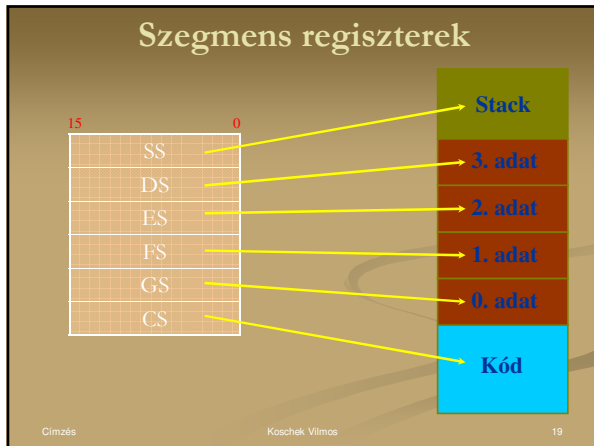
## Programozható regiszterek

Programozó által használható regiszterek.

- Általános regiszterek
- Szegmens regiszterek
- Státusz és vezérlő regiszterek

Címzés      Koschek Vilmos      17





## Címzési mód bájta

Lehetséges átvitel

- R -> M
- R <- M
- R -> R

D=0 : regiszterből  
D=1 : regiszterbe

OP1: REG : regiszter  
OP2: MOD-R/M: regiszter/memória

D	REG	MOD R/M	Irány
0	R	M	R->M
1	R	M	R<-M
0	R	R	R->R
1	R	R	R->R

**MOV AX, BX ?**

Címzés Koschek Vilmos 25

## Címzési mód bájta értékei

REG	W=1	W=0	MOD R/M	00	01	10	11 (W!)
AX	AX	AL	000	bx+si	bx+si+D	bx+si+D	AL/AX
CX	CX	CL	001	bx+di	bx+di+D	bx+di+D	CL/CX
DX	DX	DL	010	bp+si	bp+si+D	bp+si+D	DL/DX
BX	BX	BL	011	bp+di	bp+di+D	bp+di+D	BL/BX
SP	SP	AH	100	si	si+D	si+D	AH/SP
BP	BP	CH	101	di	di+D	di+D	CH/BP
SI	SI	DI	110	d.cím	bp+D	bp+D	DI/SI
DI	DI	BI	111	bx	bx+D	bx+D	BI/DI

**MOV AX, BX ?**

Címzés Koschek Vilmos 26

## 32 bites

- MK: 1-2 bájta
- Adat: 4 bájta
- Cím: 4 bájta
- SIB bájta: Scale, Index, Base

1	2	4	8	*	+	EAX EBX ECX EDX EBP ESI EDI ESP
---	---	---	---	---	---	--

Címzés Koschek Vilmos 27

## Címzési lehetőségek

**Valós mód**

11	22	33	44
44	33	22	11
44	44	44	44

**Védett mód**

Később...

Eltolás (16)

Szegmens (16)

Fizikai cím (20)

Címzés Koschek Vilmos 28

## Logikai cím számításának forrásai

Memória hivatkozás típusok	Szegmens bázis	Lehetséges másik szegmens	Eltolás
Utasítás leltvá	CS	nincs	EIP
Stack művelet	SS	nincs	ESP
Változók (kivéve a következőket)	DS	CS,ES,SS,FS,GS	Effektív cím
String forrás	DS	CS,ES,SS,FS,GS	ESI
String cél	ES	nincs	EDI
BP, mint bázisregiszter	SS	CS,ES,SS,FS,GS	Effektív cím

Címzés Koschek Vilmos 29

## Még néhány példa...

- Regiszteres címzés: INC ECX
- Bennfoglalt (közvetlen adat): MOV EAX, 112233H
- Direkt memória: ADD CÍM, AL
- Regiszter indirekt: MOV [SI], 7
- Bázis címzés: SUB [BX]CÍM, AH
- Indexelt címzés: MOV CÍM[EDI], 25
- Bázis indexelt címzés: ADD [EBX]CÍM[ESI], AL
- String címzés:
  - MOV SI, CÍM1
  - MOV DI, CÍM2
  - MOVSB
 --> [DS:SI] --> [ES:DI]
- I/O port címzése:
  - Direkt: OUT 30, AL
  - Indirekt: IN AX, DX

Címzés Koschek Vilmos 30

## Virtuális tárkezelés

- Fogalom
- Lapcímzés
- Szegmentálás
- Példa: Intel

Címzés      Koschek Vilmos      31

## Fogalom

Címzés      Koschek Vilmos      32

## Lapcímzés, működés

Virtuális címér      Lap – azonos méret !      Valós címér, OPT

Címzés      Koschek Vilmos      33

## Win7....

1. Kattintson ide a Rendszer párbeszédpanel megnyitásához.
2. A bal oldali ablakablán kattintson a Speciális rendszerbeállítások elemre. Amennyiben a rendszer rendszergazdai jelszót vagy megerősítést kér, írja be a jelszót vagy adjon megerősítést
3. A Speciális lap Teljesítmény területén kattintson a Beállítások gombra.
4. Kattintson a Haladó fülre, majd a Virtuális memória területén kattintson a Módosítás gombra.
5. Törölje A lapozófájl méretének automatikus kezelése az összes meghajtón jelölőnégyzetet.
6. A Meghajtó [Kötetcímke] kattintson arra a meghajtóra, amely a módosítani kívánt lapozófájl tartalmazza.
7. Kattintson az Egyéni méret gombra, írja be az új méret értékét megabájrt mértékegységben a Kezdeti méret (MB) vagy a Maximális méret (MB) mezőbe, kattintson a Beállítás gombra, majd kattintson az OK gombra.

Címzés      Koschek Vilmos      34

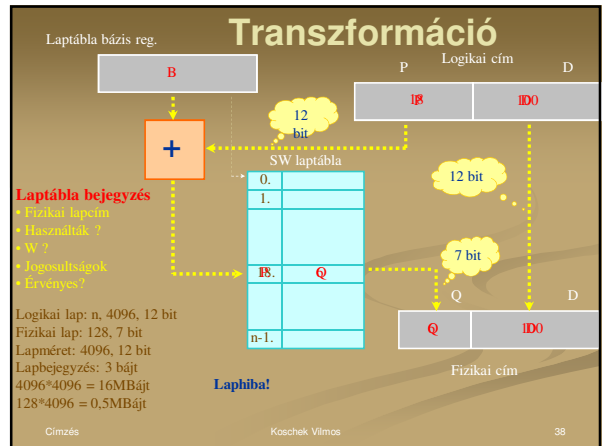
## Win7...

Címzés      Koschek Vilmos      35

## Kérdések

- Lap mérete ?
- Transzformáció ?
- Hozzáférési idő ?
- Melyik lapot dobjam el ?

Címzés      Koschek Vilmos      36



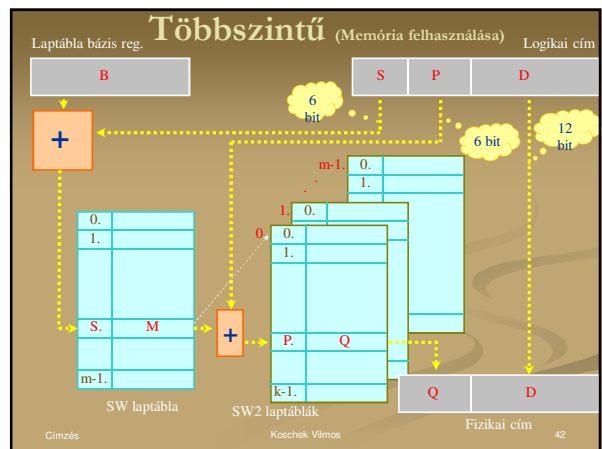
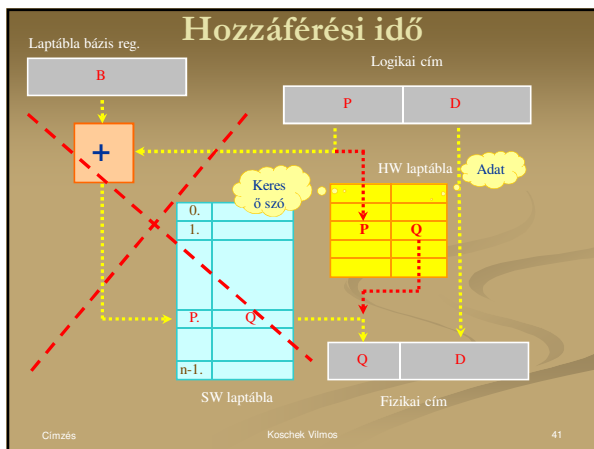
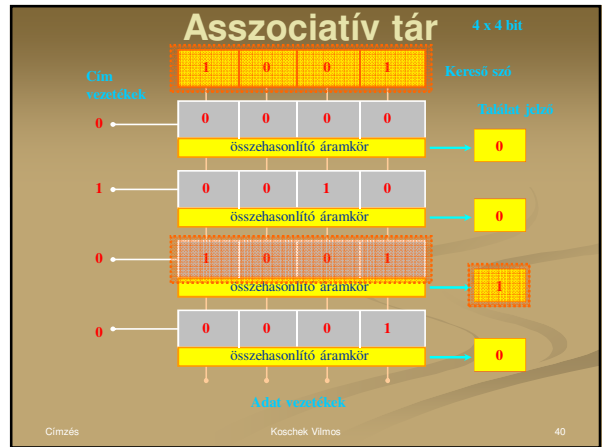
### Két „apró” probléma...

- Memóriaolvasás? +1? -> hozzáférési idő
- Memória felhasználásának hatékonysága?
  - 10 alkalmazás:  $10 * 4096 * 3 = 120\text{KBájtt}$
  - $128 * 3 = 384\text{Bájtt}$

Címzés

Koschek Vilmos

39



## Kétszintű lapcímzés

- $m=6$  bit (S) -> SW laptábla, 64 bejegyzés  
 $k=6$  bit (P) -> m darab SW2, 64 bejegyzés  
 10 alkalmazás:  $10 \times 100\text{KBajt}$   
 1 db SW, 10 db SW2  
 $64 \times 3 + 10 \times 64 \times 3 = 2112$  Bajt
- Memóriaolvasás? +2? -> hozzáférési idő  
 Asszociatív tár:  
 P,Q -> S,P,Q!

S	P	Találat
0	0	*
0	1	*
1	0	1/2 ✓
1	1	✓

Címzés Koschek Vilmos 43

## Melyik lapot dobjam el ?

Lapigény

```

    graph TD
      A[Lapigény] --> B[Van szabad fizikai lap]
      A --> C[Nincs szabad fizikai lap]
      B --> D[OK]
      C --> E[El kell dobni egyet, na de melyiket?]
      E --> F[A legalkalmasabbat ...]
      F --> G[Későbbi lapváltások száma]
      F --> H[Kevés munkával jár]
      G --> I[Lapváltási algoritmusok]
      H --> J[„Tiszta” lap]
    
```

Címzés Koschek Vilmos 44

## Lapváltási algoritmusok

- FIFO
- LRU
- FINUFO

Címzés Koschek Vilmos 45

## FIFO

A tárban levő lapok közül a legrégebbit! -> First in first out

Eltárolt lapok elrendezése

Címzés Koschek Vilmos 46

## LRU

A tárban levő lapok közül a legrégebben használt! -> Least recently used

Eltárolt lapok elrendezése

$T_w$ : munka  
 HB, KSZ  
 HW

**HB==0**  
 • Nem történt hozzáférést  
 • KSZ++

**HB==1**  
 • Történt hozzáférést  
 • KSZ=0  
 • HB=0

**Kiválasztás: KSZ ?**

Címzés Koschek Vilmos 47

## FIFO, LRU összehasonlítása

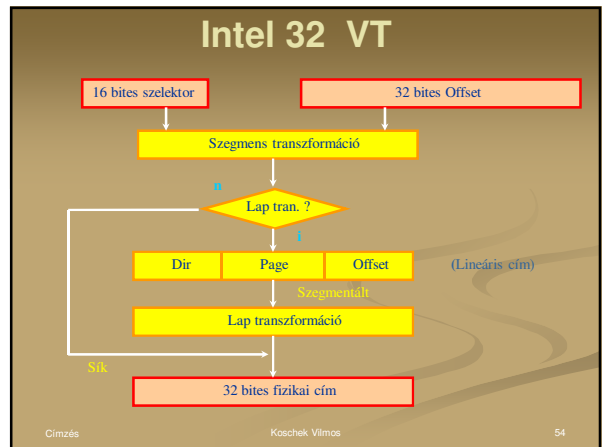
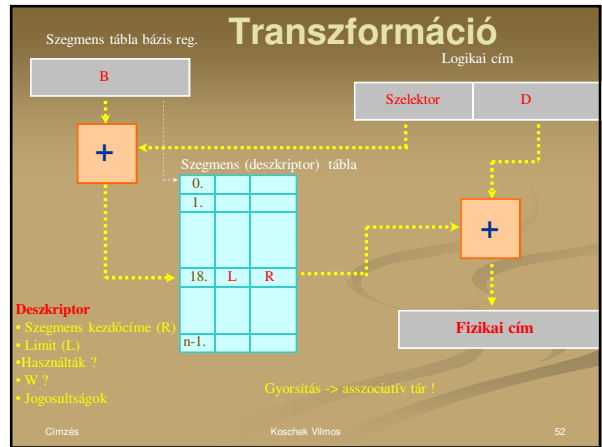
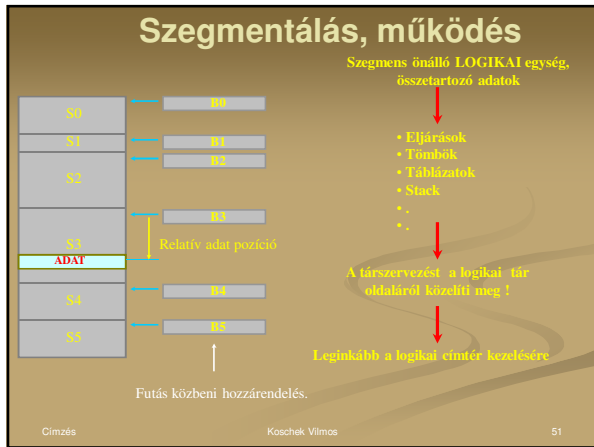
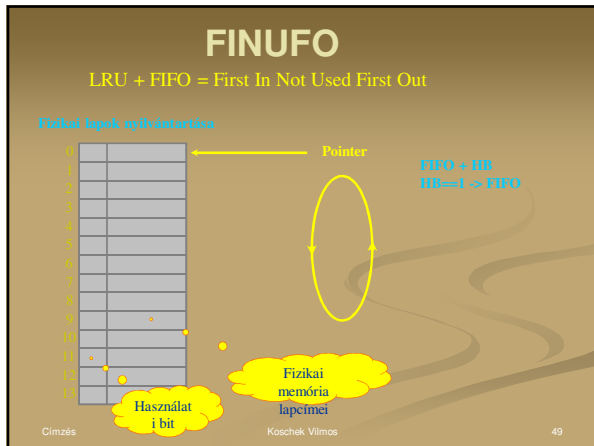
4 db logikai, 3 db fizikai lap

CIMLÁNC	1	2	3	1	2	4	1	2	3	1	2	4	1
FIFO					1	2	3	4					1
LRU				3			4						3

**Akkor melyik a jobb? És miért?**  
**És mikor egyforma?**

Címzés Koschek Vilmos 48





## Transzformáció

### Sík (flat) címtér

- A memória egyetlen 4Gbájtos tömb
- Cím: 32 bites
- Szegmens reg.-t csak egyszer kell beállítani

### Szegmentált címtér

- 64 tbájt logikai címtér
- Cím: 16 bites szelektor+ 32 bites offset
- 16383 db 4Gbájtos szegmens

Címzés      Koschek Vilmos      55

## Szegmens transzformáció

The diagram illustrates the transformation of a logical address into a linear address. It shows a 32-bit **Szelektor** (selector) and a 32-bit **Offset** being combined with a **Deszkriptor** from a **Deszkriptor tábla** (descriptor table). The descriptor table contains entries 0, 1, ..., n-1. The **LPTB: GDTB** (Linear Pointer Table / Global Descriptor Table Base) is used to find the descriptor. The final result is the **Lineáris cím** (linear address).

Logikai cím -> Lineáris cím

Címzés      Koschek Vilmos      56

## Szelektor

- Látható rész:** 16 bites szegmens regiszter
  - 2 bit, RPL: szelektort létrehozó eljárás védelmi szintje
  - 1 bit, 0-GDT, 1-LDT
  - 13 bit, 8192 bejegyzés
- Nem látható rész:** deszkriptor, proc kezeli!

$$2^{13} * 2^1 * 2^{32} = 2^{46}$$

RPL - Responder's Privilege Level

Címzés      Koschek Vilmos      57

## Deszkriptor

- Bázis, 32 bit
- Limit, 20 bit
- Granulaty bit,
  - G=0-Limit\*1 bájt -> 1Mbájt
  - G=1-Limit\*4096 bájt -> 4Gbájt
- Type 3 bit, leíró típusa: read, write, code, data, ....
- DPL 2 bit, deszkriptor privilegium szint
- Present 1 bit
- Accessed 1 bit

Címzés      Koschek Vilmos      58

## Lap transzformáció

Lineáris cím -> Fizikai cím

The diagram shows the transformation of a linear address into a physical address. It starts with a **Lineáris cím** (linear address) divided into **DIR**, **PAGE**, and **OFFSET** fields. The **DIR** field is used to access the **Lapkönyvtár** (page directory) table, which provides the **LKB** (Linear Kernel Base). The **PAGE** field is used to access the **Laptábla** (page table) table, which provides the **LTB** (Linear Table Base). The **OFFSET** field is used to access the specific byte within the page. The final result is the **Fizikai cím** (physical address).

Címzés      Koschek Vilmos      59

## Bejegyzések LKB, LTB

- Present 1 bit
- U/S, R/W 2 bit, lapszintű tárvédelem
- Accessed 1 bit
- Dirty 1 bit
- 20 bit, laptábla/lap fizikai címének felső része

LKB, LTB = 4 bájt  
DIR/10 bit + PAGE/10 bit + OFFSET/12 bit = 32 bit

- DIR/10 + CR3 => LKB (DIR\*4)
- LKB/20 + PAGE/10 => LTB (30)
- LTB/20 + OFFSET/12 => FIZIKAI CÍM

Címzés      Koschek Vilmos      60

## Asszociatív tár

Lineáris cím => fizikai cím kétszintes => lassú



32 elemű asszociatív tár, TLB:  
Translation Lookaside Buffer



LKB – LTB párokat tárolja



**98%**

<http://nik.bmf.hu/broczko/tantargy.htm>