

Operētajsistēmu inženierija

Adresu translācija un koplietošana lietojot segmentāciju

Kursu vada Leo Seļāvo

Adrešu apgabali

- Kodola programmas un lietojumprogrammas
 - Katra darbojas sava apgabalā (problēmu izolācija)
 - Kodola apgabalā atļauts izpildīt privelegētās instrukcijas
- Kodolam jā rūpējas par adrešu apgabaliem
 - Text, Data, Stack
 - Kodols darbojas savā (monolītiskā) apgabalā un nodrošina piekļuvi aparatūrai
 - Lietotāja programmas adresu apgabals – var palielināties un samazināties.

Galvenās operācijas

- Radīt
- Palielināt
- Samazināt
- Izņemt
- Pārslēgt (procesoru no vienas pie otras)
- Dalīties (sharing)

Divas galvenās pieejas

- Ar segmentāciju
- Ar adrešu tabulām
- Ieviest tikai programmatūrā pārāk dārgi
 - Programmatūra - konfigurē
 - Aparatūra - izpilda

Bez virtuālās atmiņas

- Fiziskā adrese = CPU adrešu maģistrāles “vadi”
- X86 ielāde
 - Sākas reālajā režīmā (real mode)
 - $\text{segment} * 16 + \text{offset} \Rightarrow \text{physical address}$
 - Nav aizsardzības
 - Tad pārslēdzas uz aizsargāto režīmu (protected mode), kas ļauj adrešu apgabalu veidošanu

Protected mode

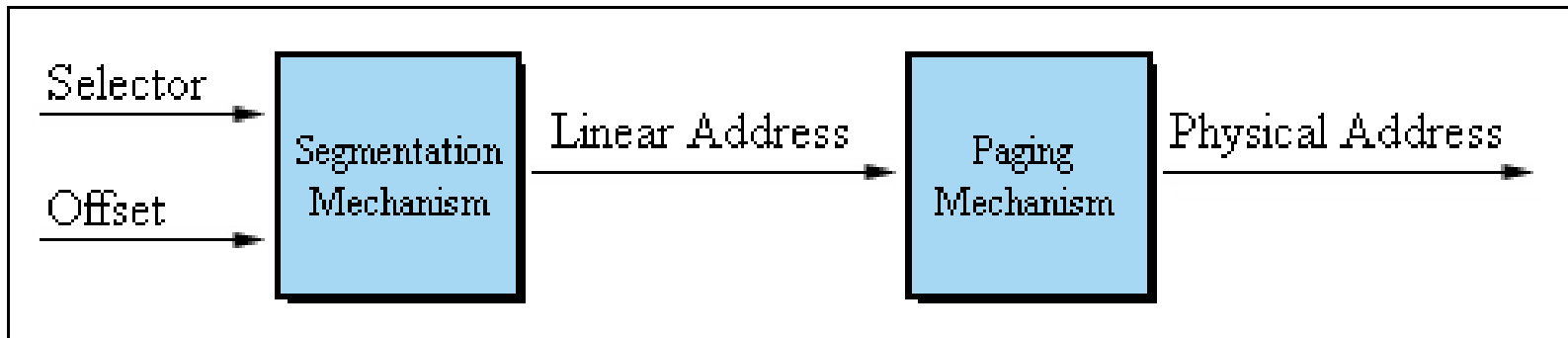
selector:offset (lógiskā adrese)

== SEGMENTATION ==>

Lineārā adrese

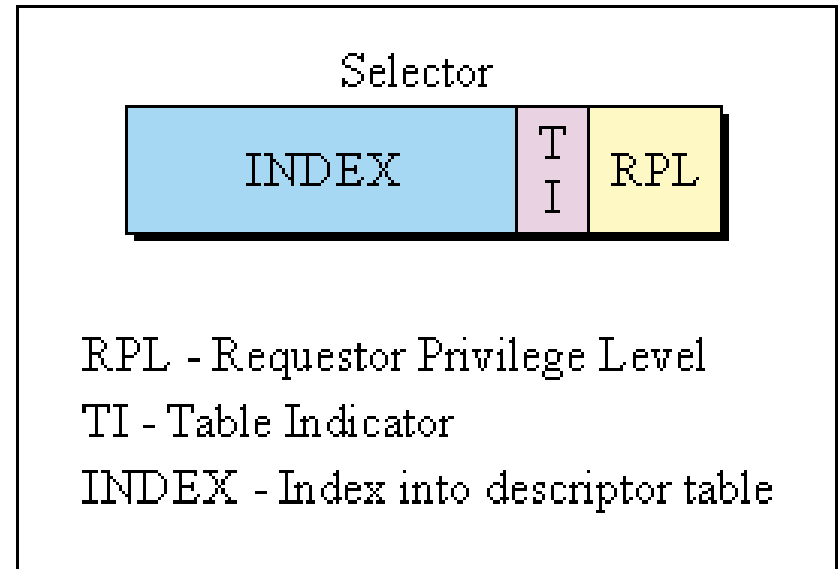
== PAGING ==>

Fiziskā (reālā) adrese



Protected mode - segmentācija

- Segmenta reģistrs satur segmenta “selektoru”
- Selektors satur indeksu uz GDT ierakstu
 - (Global Descriptor Table)
 - LGDT instrukcija ielādē CPU GDT
- Pāreja uz protected mode notiek ierakstot PE bitā CR0 reģistrā



Deskriptoru tabula

Descriptor Entry

Base 24-31	G	D / B	0	A V L	Seg Limit 16-19	P	D P L	S	Type	Base 16-23
Base Address 0-15					Segment Limit 0-15					

AVL - Available for use by the operating system

BASE - Segment Base Address

D / B - Default Segment Size (16 / 32 bits)

DPL - Descriptor Privilege Level

G - Granularity

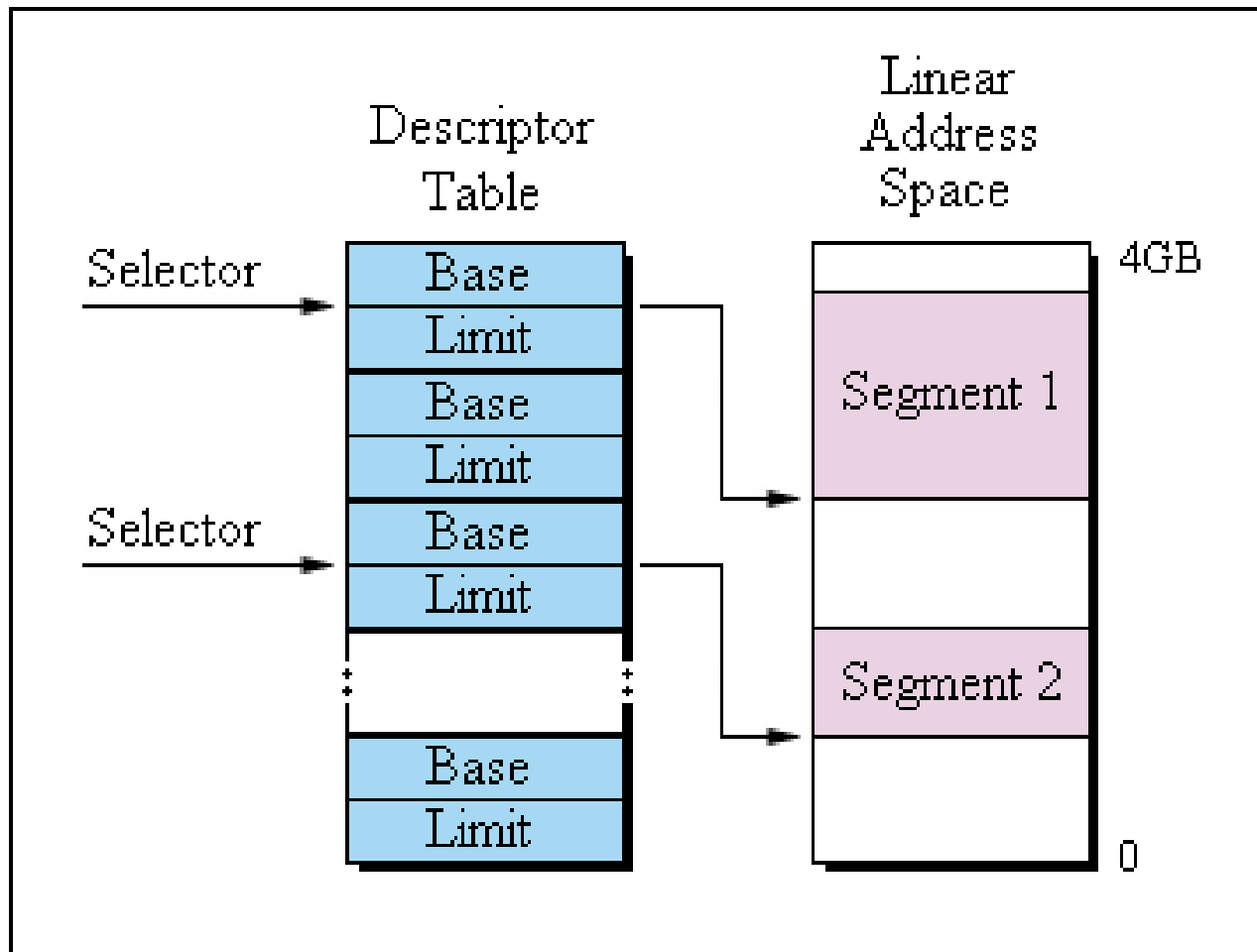
LIMIT - Segment Limit

P - Present Bit

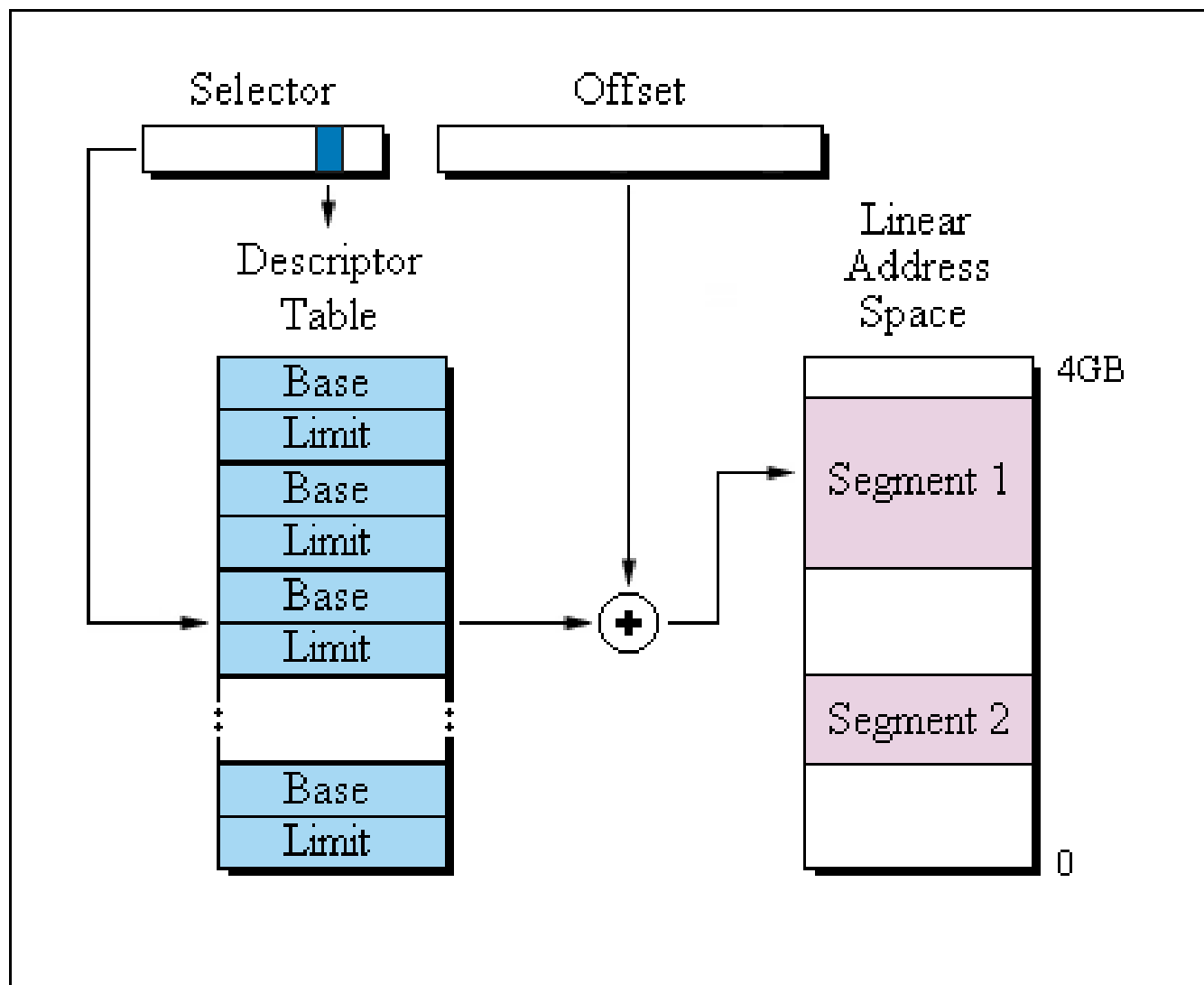
S - Descriptor Type (System / Application)

TYPE - Segment Type

Selektori un segmentācija



Lineārās adreses aprēķins



Drošība?

- Aizsardzība?
- Vai OS var ierobežot kuru atmiņu lai aplikācija lasa/raksta?
- App. Var ielādēt jebkuru selektoru segmenta atmiņā.
- Bet app var lietot tikai indeksus uz GDT
- App nevar mainīt GDT reģistru (privilģēts)
- Kā pēc app. Nevar rakstīt deskriptorus uz GDT?
- Kā sistēmas izsakukumi pāriet uz kodola apgabalu?
- App. nevar vienkārši samazināt CPL ...

Privilēģijas – kā pārlekt uz citu apgabalu?

- CPL – tekošais privilēģiju līmenis
 - CPL=0 – OS
 - CPL=3 – lietotājs
- Tai pašā CPL līmenī – **ljmp**
- Citā – jāsauc vārtejas (call gates)
 - Maina privilēģijas un CS un SS
 - Realizētas ar īpaša tipa deskriptoru iekš GDT
 - Konceptuāli līdzīgi pārtraukumiem, bet pārtraukumu deskriptori stāv IDT

Segmentācija

- Kā palielināt atmiņas apgabalu?
 - Paņem jaunu (lielāku) segmentu
 - Pārkopē no vecā
 - Nonuļļo atlikumu
 - Atbrīvo veco segmentu

- Vai var labāk?
 - Iapošana

Paging (lapošana)

