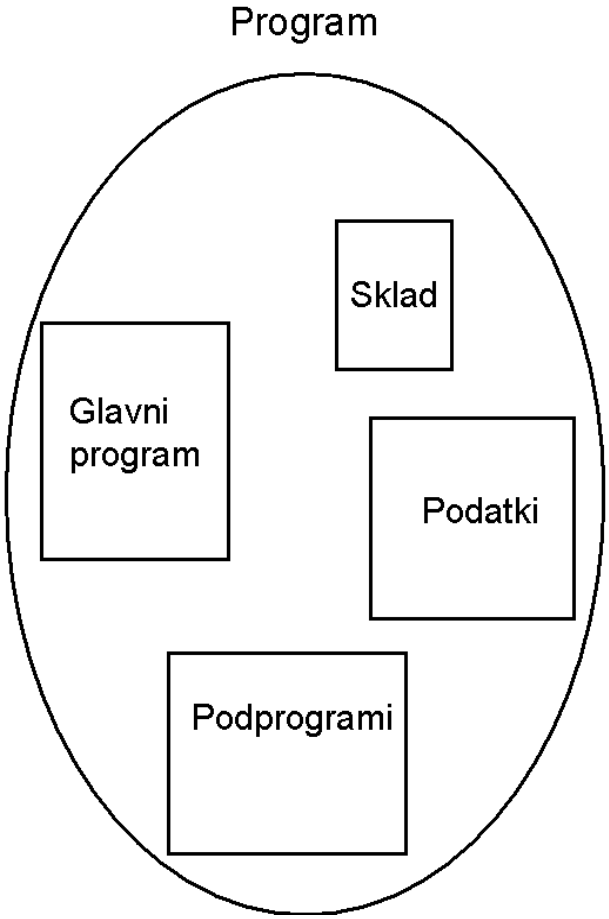
Segmentiran pomnilnik

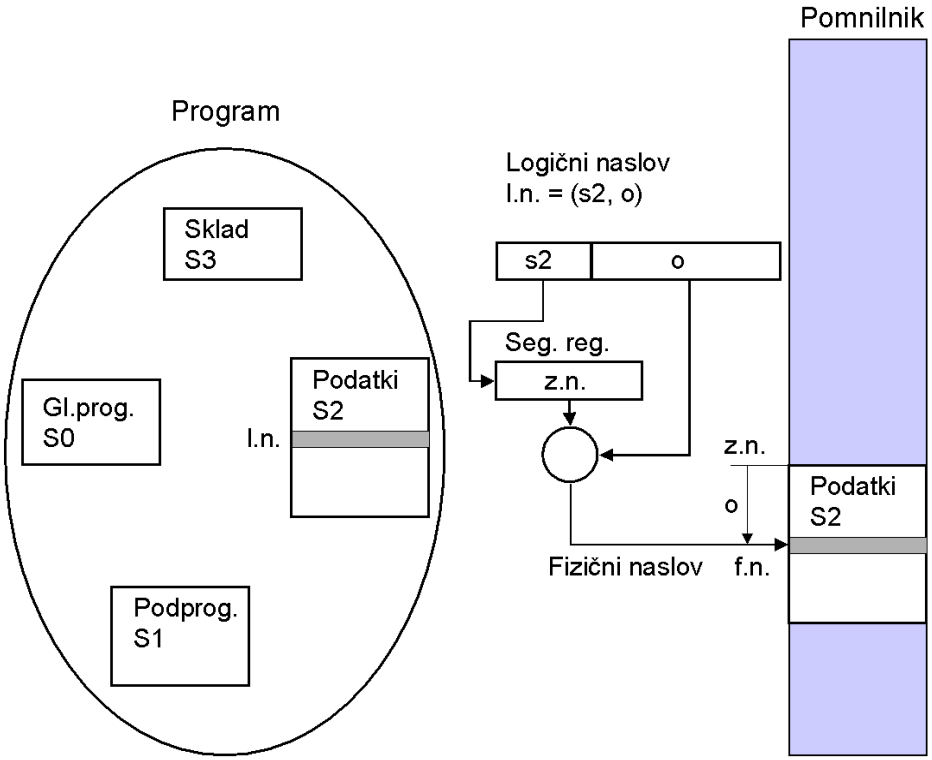
- Segmentiran pomnilnik temelji na izgledu naslovnega področja programa (pomnilnika), kot ga vidi uporabnik – programer.
- Programer ne vidi programa kot linerano zaporedje pomnilniških naslovov. Za programerja je program zbirka različno obsežnih kosov programa – segmentov. Logično naslovno področje sestavlja zbirka segmentov.



- Segment je kos programa, ki ima s stališča uporabnika (programerja) enoten pomen (npr. ukazi (glavni program, podprogrami) podatki, sklad, ...).
- Vsak segment ima za programerja svoje ime in velikost (dolžino).
- Logični naslov je torej dan z dvema količinama: imenom segmenta in odmikom znotraj segmenta.

Segmentiran pomnilnik – koncept

- Logično naslovno področje programa je deljeno na različno dolge dele, **segmente**.
- Segment se zvezno namesti v pomnilnik (če je v pomnilniku, potem je v pomnilniku v celoti). Segment je določen z začetnim naslovom v pomnilniku in dolžino segmenta.
- Logični naslov sestavlja ime (oznaka) segmenta in odmik znotraj segmenta.
- V času izvajanja programa se logični naslovi preslikajo v fizične naslove preko segmentnih registrov.
- Segmentni ragister vsebuje začetni fizični naslov segmenta v pomnilniku.
- Sočasno je lahko "aktivnih" toliko segmentov, kolikor je segmentnih registrov.



Intel 8086 in segmentiran pomnilnik

- Mikroprocesor 8086 omogoča segmentiran pomnilnik.
- Ima štiri segmentne registre (CS, DS, ES, SS) za ukazni, podatkovni, skladovni, ekstra segment. Velikost segmenta 64 Kb.
- Dopušča sočasno štiri aktivne segmente

$$(4 \times 64Kb)$$

  v naslovnem področju 1 Mb.
- Ne podpira izvedbe navideznega pomnilnika. Preslikava logičnega naslova v fizični naslov je "trdo" določena - fiksna.

$$(s, d) \rightarrow s \times 16 + d$$

- Preslikava logičnega v fizični naslov za kodni (ukazni) segment

3	4	6	5	CS (Kodni segmentni register)
2	4	1	B	IP (Programski števec)
3	6	A	6	B	Fizični naslov

- Podobno se preko segmentnih registrov DS, SS, ES, preračunajo drugi fizični naslovi.
- Razlog za segmentacijo pomnilnika:
  - razširitev naslovnega področja na 1 Mb.
  - kompatibilnost s predhodniki.

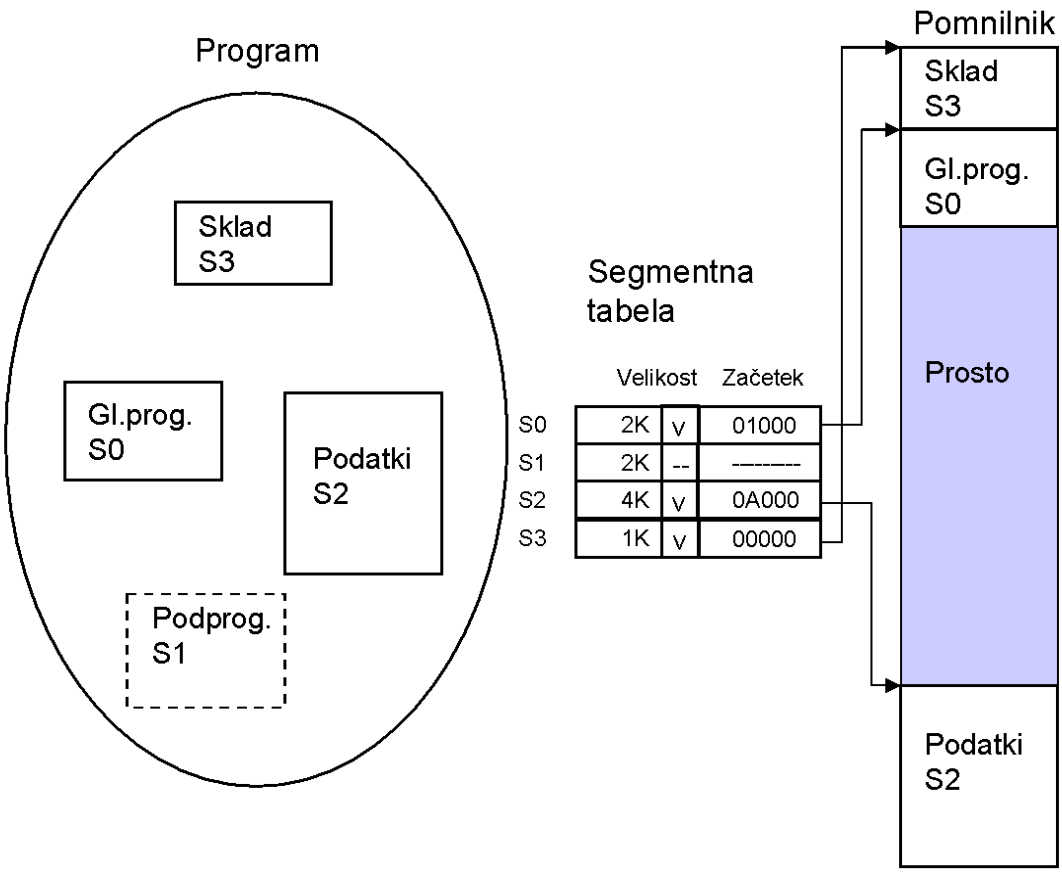
Segmentiran navidezni pomnilnik

- Segmentiranje (ponovitev)
  - Logični naslovni prostor (program) je deljen na segmente.
  - Logični naslov sestavljata dva dela: **ime** (oznaka) segmenta in **odmik** znotraj segmenta, (s, d). Logični naslovni prostor segmentiranega pomnilnika je "dvodimenzionalen".
  - Dvodimenzionalen logični naslov se v času izvajanja programa preslika v enodimenzionalen (linearen) fizični naslov.

$$l.n. = (s, d) \rightarrow f.n.$$

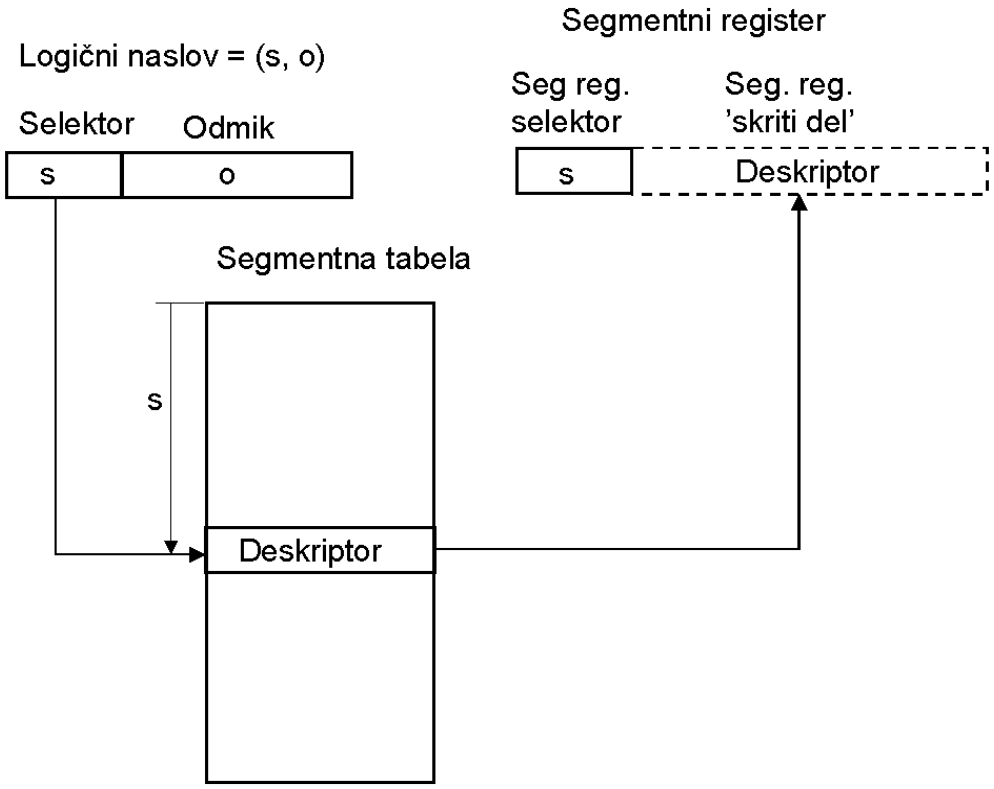
- Logični naslovi se preslikajo v fizične naslove preko segmentne tabele. Segmentna tabela se nahaja v glavnem pomnilniku.
- Segmentna tabela vsebuje "deskriptorje" segmentov. Deskriptor vsebuje preslikavo l.n. v f.n., velikost segmenta in dodatna določila (prisotnost v pomnilniku, spremenjenost, dovoljenja,...).
- Navidezni pomnilnik s segmentiranjem:
  - V pomnilnik se namestijo samo trenutno potrebni segmenti.
  - V primeru reference na segment, ki ni v pomnilniku, se izvajanje programa (začasno) prekine (nastopi izjema).
  - O.S. streže izjemi: v pomnilnik namesti zahtevani segment (na "zahtevo").
  - Obnovi izvrševanje (napredovanje) programa.

Segmentiran navidezni pomnilnik



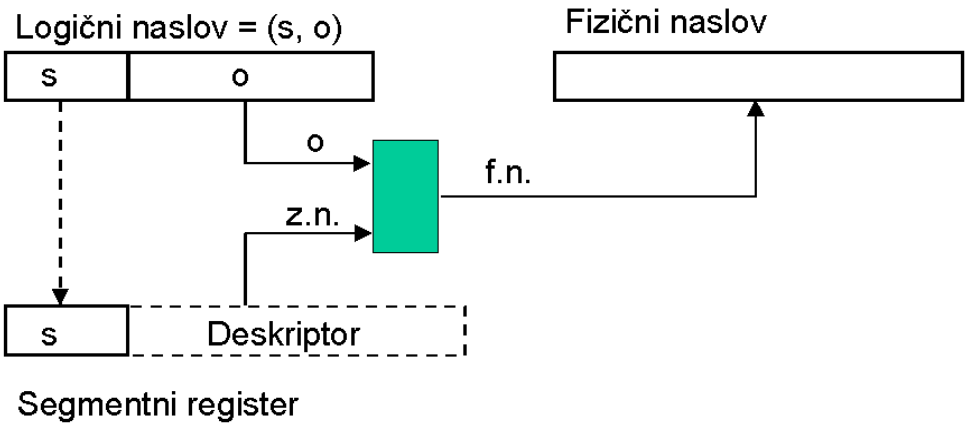
Segmentna tabela in asociativni segmentni registri

- Preslikava l.n. v f.n. preko segmentne tabele, ki se nahaja v pomnilniku, je počasna (vsaka referenca programa zahteva dva pomnilniška dostopa).
- V primeru ostranjenega pomnilnika se preslikava pohitri z naslovnim preslikovalnim predpomnilnikom (ATC, TLB). V primeru segmentiranega pomnilnika se preslikava pohitri z asociativnimi registri. Zadostuje že relativno malo segmentnih registrov.
- Aktivni "deskriptorji" segmentne tabele se nahajajo v hitrih (asociativnih) registrih (npr. Register podatkovnega segmenta, register ukaznega segmenta, ...).
- Segmentni registri se polnijo ob prvi referenci na ustrezen segment – referenca na deskriptor v segmenti tabeli.



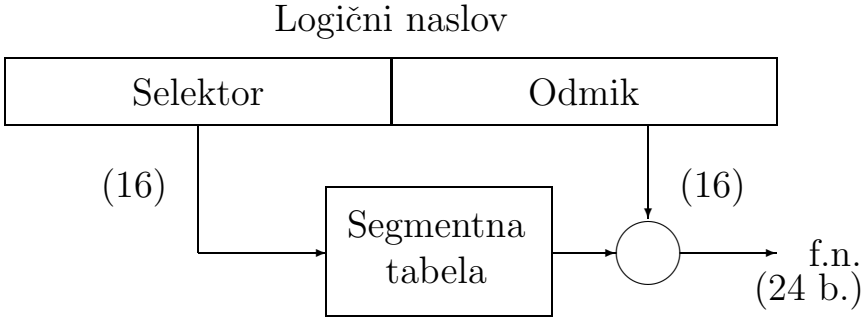
Asociativni segmentni registri

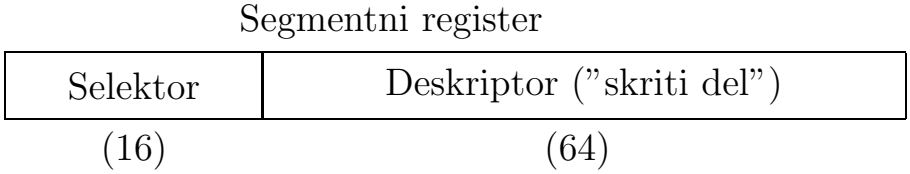
- 2-D logični naslov se preslika v 1-D (linearni) (fizični) naslov.
- Selektorski del logičnega naslova se primerja s selektorskim delom registra.
- V primeru ujemanja (zadetka) se na podlagi "skritega" dela segmentnega registra in odmika določi linearni, t.j. fizični naslov.



80286+ in navidezni pomnilnik

- Mikroprocesorji 80x86 podpirajo segementiran navidezni pomnilnik.
- Lahko delujejo v "realnem" pomnilniškem načinu, ki je v bistvu 8086 način, ali v "navideznem" (virtualnem oziroma zaščitenem) načinu.
- Mikroprocesor 80286 v zaščitenem načinu podpira segmentiran navidezen pomnilnik.
  - (Maksimalna) velikost segmenta: 64 Kb.
  - Maksimalno število segmentov: 16383 (2^{14}).
  - Logično naslovno področje: 1 Gb ($2^{14} \times 2^{16}$).
  - Fizično naslovno področje: 16 Mb (2^{24}).
- Segmentna tabela se nahaja v glvnem pomnilniku.
- **Načelna preslikava logičnega naslova v fizičen naslov**





- "Vidni" del segmentnega registra vsebuje selektor (ali začetni naslov segmenta v realnem načinu).
- "Skriti" del segmentnega registra vsebuje aktivni deskriptor iz segmentne tabele.
- Skriti del registra se polni avtomatično ob polnjenju (vpisu) vidnega dela.

Segmentiranje z ostranjenjem

- Logično naslovno področje programa se deli na različno obsežne dele – segmente ("najprej" segmentiranje).
- Vsak segment se deli na enako velike dele – strani (ostranjenje segmentov).
- Dvodimenzionalni logični naslovi se preslikajo v enodimenzionalne (linearne) naslove preko segmentne tabele. V pomnilniku se nahajajo samo nekateri segmenti.
- Segmentna tabela se nahaja v glavnem pomnilniku. Aktivni deskriptorji segmentne tabele se nahajajo v hitrih (asociativnih) segmentnih registrih.
- Ker so segmenti ostranjeni, linearni naslov še ni fizični naslov.
- Linearni naslovi se preko tabele strani preslikajo v fizične naslove. Tabela strani se nahaja v glavnem pomnilniku.
- V pomnilniku se nahajajo samo nekatere strani segmenta.
- Dodeljevanje na zahtevo (bistvo navideznega pomnilnika) se izvede na nivoju
  - segmentov (pravzaprav potem nimamo ostranjenja),
  - strani (pomnilnik je sicer segmentiran, vendar so v pomnilniku vsi segmenti),
  - kombinacije obeh (v pomnilniku so samo nekatere strani nekaterih segmentov).

80386 + in navidezni pomnilnik

- 80386 podpira segmentiran in ostranjen navidezni pomnilnik (ostranjeni segmenti). Ostranitev ni obvezna, segmentiranje je.
- Razlika segmentiranja med 80286 in 80386 + je samo v maksimalni velikosti segmenta (80386 - 4 Gb) in več (asociativnih) segmentnih registrih (CS, SS, DS, ES, FS, GS).
- Dve segmentni tabeli, globalna (GDT) in lokalna (LDT), vsaka v velikosti 8192, se nahajata v glavnem pomnilniku.
- GDT (Global Descriptor Table - hrani deskriptorje globalnih segmentov, skupne več programom).
- LDT - hrani deskriptorje segmentov posameznega programa.
- Asociativni segmentni registri vsebujejo deskriptorje aktivnih segmentov.
- Z različnim upravljanjem pomnilnika nastanejo različni "pomnilniški modeli":
  - Nesegmentiran, neostranjen: ("sploščen" model) Segmentni registri-deskriptorji inicializirani na isto vrednost (logično naslovno področje = fizično naslovno področje = 4 Gb).
  - Nesegmentiran, ostranjen: (logično naslovno področje: 4 Gb, fizično naslovno področje: 4 Gb).
  - Segmentiran, neostranjen: logično naslovno področje: do 64 Tb, (velikost segmenta x število segmentov = 4 Gb x 16383). Fizično naslovno področje: 4 Gb.
  - Segmentiran in ostranjen (olajša nameščanje v pomnilniku).

Deskriptorji segmentnih tabel GDT in LDT

(8 bajtov)

- Začetni naslov segmenta v glavnem pomnilniku (32 bitov),
- velikost segmenta (20 bitov) ("zrnatost" bajta/strani),
- DPL: prioritetni nivo deskriptorja (od višje proti nižji): 0, 1, 2, 3 (2 bita),
- dodatna določila:
  - P - bit prisotnosti: P = 1 - segment je v pomnilniku,
  - S - uporabniški/sistemski deskriptor: S = 0 - sistemski,
  - biti dostopnih dovoljenj (beri, piši, izvrši),
  - A - bit dostopa: A = 1 - segment je bil uporabljen,

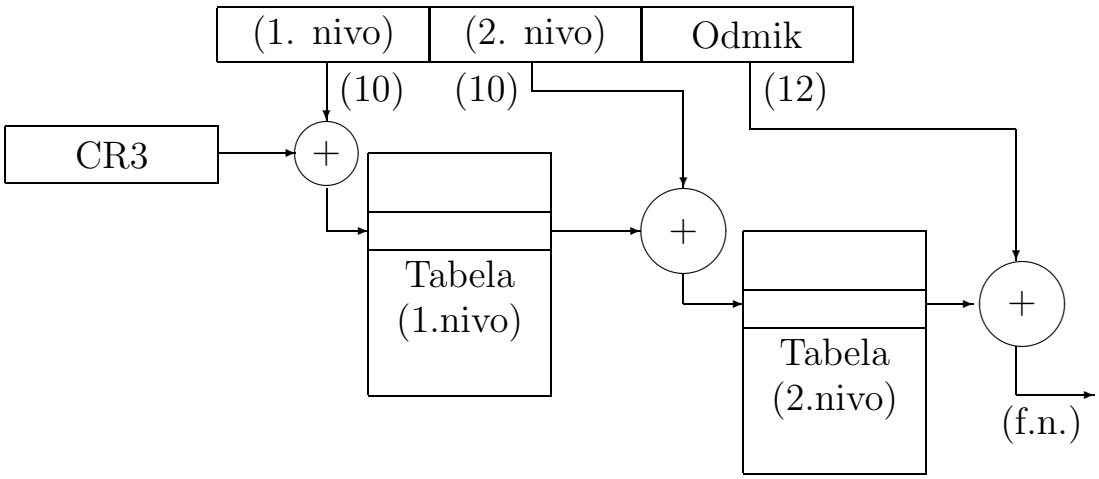
Zaščita dostopanja do segmenta

- CPL (Current Privilage Level),
  procesor deluje na enem izmed 4 prioritetnih nivojev (0,1,2: sistemski nivoji, 3: uporabniški nivo privilegijev). Trenutni nivo (običajno) določa prioritetni nivo aktivnega kodnega segmenta in je zabeležen v selektorju segmenta ("vidni" del segmentnega registra CS).
- RPL (Requsted Privilage Level),
  določa prioritetni nivo zahteve (reference) na (npr. podatkovni) segment, je sestavni del selektorja ("vidni" del segmentnega registra).
- Referenca na segment se dovoli, če sta CPL in RPL vsaj tako visoke prioritete kot DPL.
- V primeru, da je segment prisoten v pomnilniku (P = 1 - deskriptor v tabeli veljaven), sledi dostop do pomnilnika.
- V nasprotnem primeru sledi izjema.

80386 + in ostranjen pomnilnik

- Procesor daje na voljo (PG=1 v CR0) ostranitev segmentov. Segmenti so naprej deljeni na strani velikosti 4Kb.
- Linearen naslov se preko dvonivojskega sistema tabel strani, ki so v glavnem pomnilniku, preslika v (linearen) fizičen naslov.
- Preslikava l.n. v f.n. je pospešena s TLB (Translation Look-aside Buffer), ki zabeleži do 32 aktivnih preslikav (za do 32 aktivnih strani).

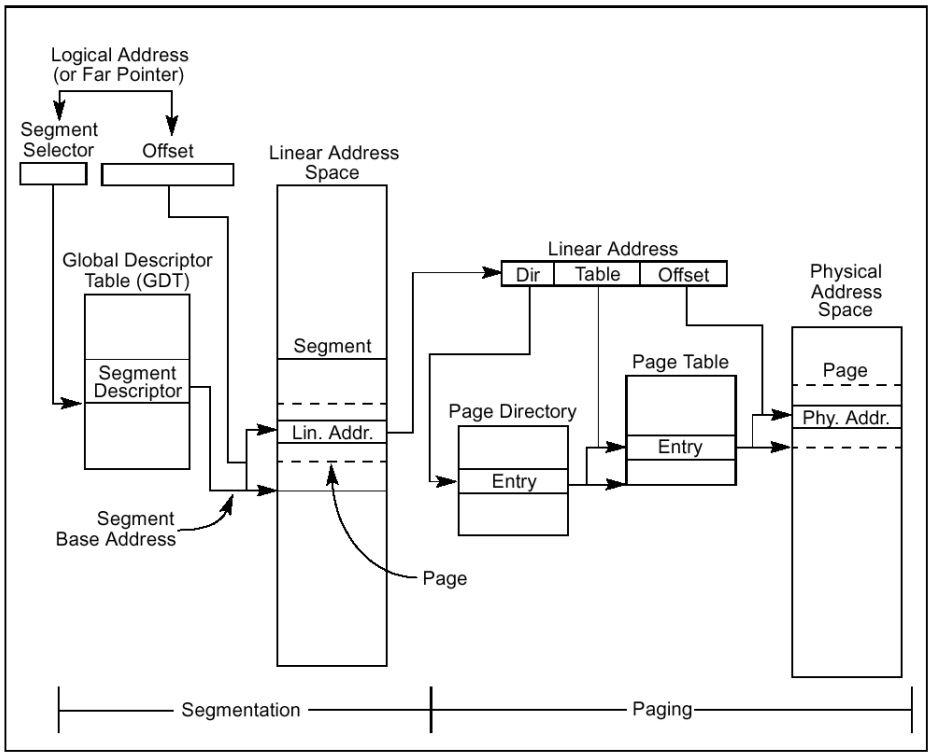
Linearni naslov po segmentiranju

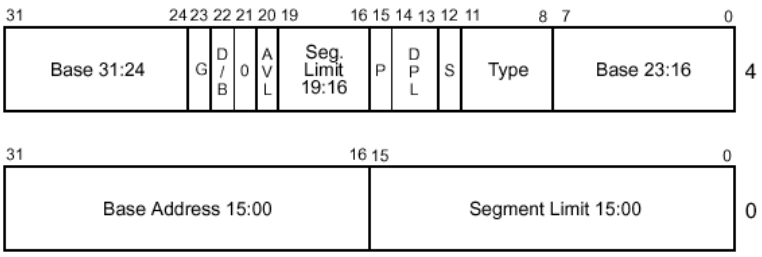


Deskriptor v tabeli strani

- Naslov tabele strani ali strani okvirja,
- P: (Present), stran prisotna,
- A: (Access), stran uporabljena,
- D (Dirty), stran spremenjena,
- U/S: (User/System), dovoljen uporabniški / sistemski dostop (prioritetni nivoji 0,1,2 so sistemski, 3 je uporabniški),
- R/W: (Read/Write), dovoljeno branje/pisanje.

80386 + in navidezni pomnilnik





AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type