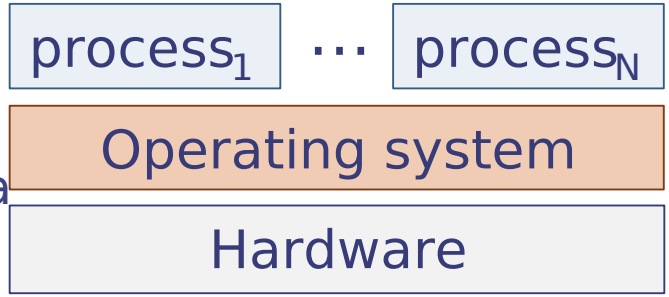
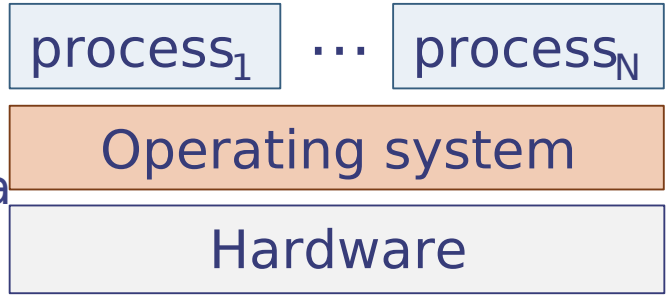
Virtualna Memorija

Podsetnik: Operativni Sistem

- Cilj OS:
  - Zastita I privatnost: Procesi ne mogu da pristupaju podacima ostalih procesa
  - Abstakcija: Sakriti detalje hardvera koji se nalazi ispod
    - npr, procesi otvaraju i pristupaju datotekama umesto pozivanja komandi direktno na periferijama
  - Menadzment resursa : Kontrolise kako procesi dele hardverske resurse (CPU, memoriju, disk, itd)

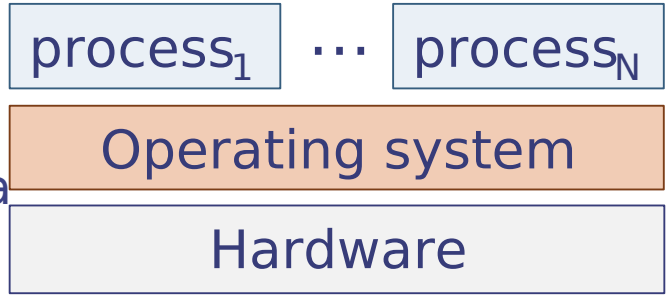


Podsetnik: Operativni Sistem



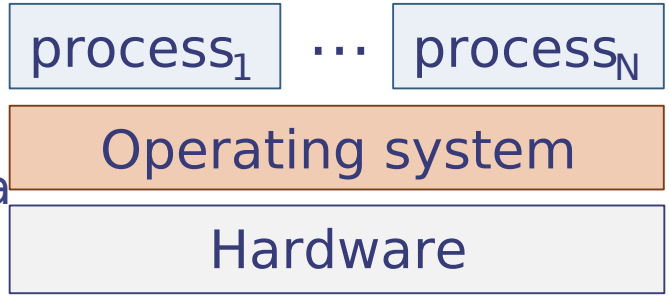
- Cilj OS:
  - Zastita I privatnost: Procesi ne mogu da pristupaju podacima ostalih procesa
  - Abstakcija: Sakriti detalje hardvera koji se nalazi ispod
    - npr, procesi otvaraju i pristupaju datotekama umesto pozivanja komandi direktno na periferijama
  - Menadzment resursa : Kontrolise kako procesi dele hardverske resurse (CPU, memoriju, disk, itd)
- Ovo je omoguceno zahvaljujuci:

Podsetnik: Operativni Sistem



- Cilj OS:
  - Zastita I privatnost: Procesi ne mogu da pristupaju podacima ostalih procesa
  - Abstakcija: Sakriti detalje hardvera koji se nalazi ispod
    - npr, procesi otvaraju i pristupaju datotekama umesto pozivanja komandi direktno na periferijama
  - Menadzment resursa : Kontrolise kako procesi dele hardverske resurse (CPU, memoriju, disk, itd)
- Ovo je omoguceno zahvaljujuci:
  - Korisnicki mod + nadzirani mod
  - Izuzeci i prekidi u cilju bezbedne tranzicije u nadzirani mod

Podsetnik: Operativni Sistem

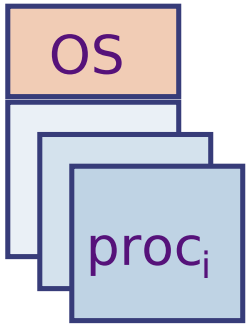


- Cilj OS:
  - Zastita I privatnost: Procesi ne mogu da pristupaju podacima ostalih procesa
  - Abstakcija: Sakriti detalje hardvera koji se nalazi ispod
    - npr, procesi otvaraju i pristupaju datotekama umesto pozivanja komandi direktno na periferijama
  - Menadzment resursa : Kontrolise kako procesi dele hardverske resurse (CPU, memoriju, disk, itd)
- Ovo je omoguceno zahvaljujuci:
  - Korisnicki mod + nadzirani mod
  - Izuzeci i prekidi u cilju bezbedne tranzicije u nadzirani mod
  - Virtualna memorija koja apstrakuje smestajne resurse na nekoj masini

Virtualna Memorija (VM)

Iluzija velikog, privatnog, uniformnog adresnog prostora

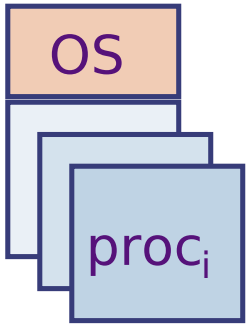
- Zastita i Privatnost
  - Svaki proces ima privatni adresni prostor

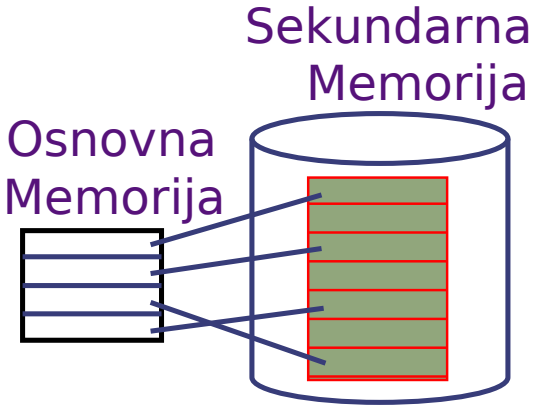


Virtualna Memorija (VM)

Iluzija velikog, privatnog, uniformnog adresnog prostora

- Zastita i Privatnost
  - Svaki proces ima privatni adresni prostor
- Stanice na zahtev (Deman Paging)
  - Omogucava pokretanje Programa vecih od velicine Osnovne memorije
  - Skriva razlicitosti Konfiguracija razlicitih masina

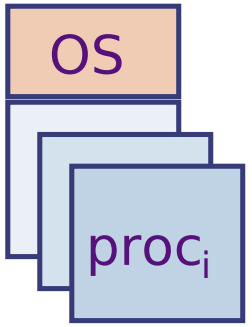


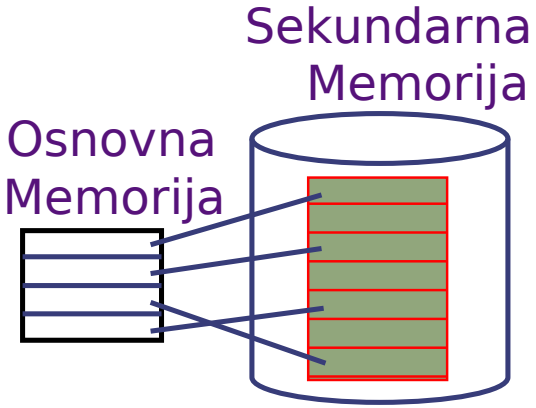


Virtualna Memorija (VM)

Iluzija velikog, privatnog, uniformnog adresnog prostora

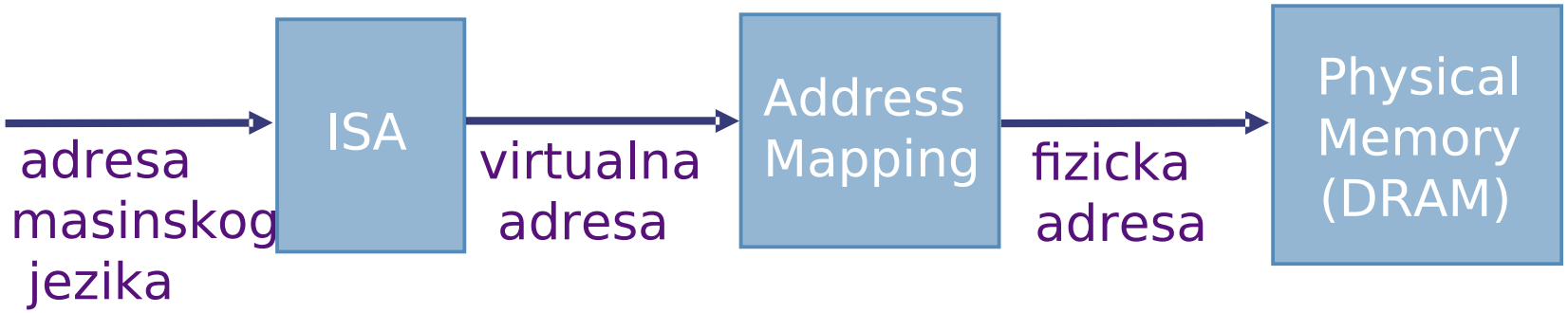
- Zastita i Privatnost
  - Svaki proces ima privatni adresni prostor
- Stanice na zahtev (Demand Paging)
  - Omogucava pokretanje Programa vecih od velicine Osnovne memorije
  - Skriva razlicitosti Konfiguracija razlicitih masina





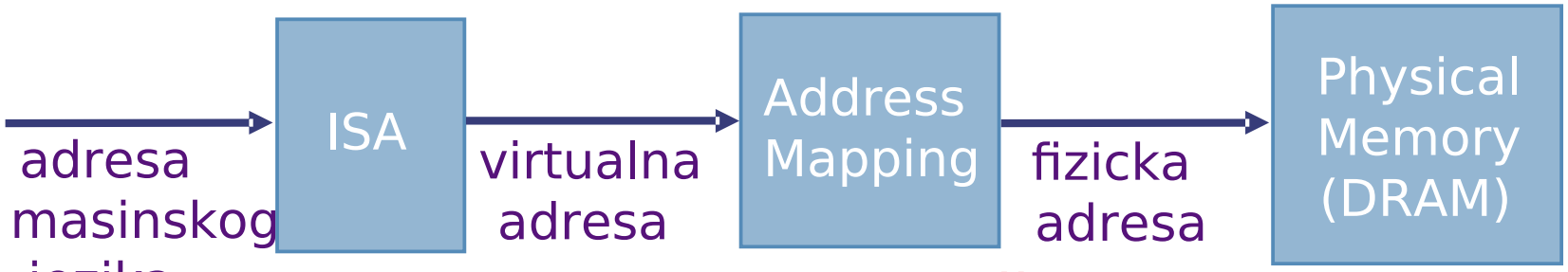
Cena VM je translacija adresa prilikom Svakog pristupa memoriji!

Nazivi za memorijske lokacije



- Adresa masinskog jezika
  - Specificirana je masinskim kodom
- Virtualna adresa
  - ISA specificira translaciju adrese masinskog koda u virtualnu adresu programske varijable (naziva se i *efektivna* adresa)
- Fizicka adresa
  - Operativni sistem specificira mapiranje virtualne adrese u odgovarajucu fizicku memorijsku lokaciju

Nazivi za memorijske lokacije

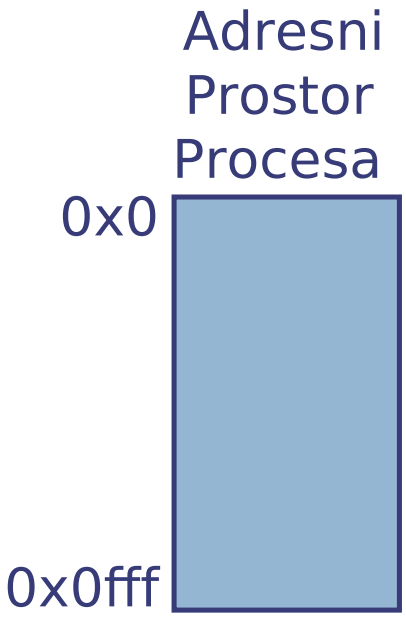


Segmentacija
ili Koriscenje Stranica

- Adresa masinskog jezika
  - Specificirana je masinskim kodom
- Virtualna adresa
  - ISA specificira translaciju adrese masinskog koda u virtualnu adresu programske varijable (naziva se i *efektivna* adresa)
- Fizicka adresa
  - Operativni sistem specificira mapiranje virtualne adrese u odgovarajucu fizicku memorijsku lokaciju

Segmentacija (Base-and-Bound)
Adresna translacija

Segmentacija (Base-and-Bound)
Adresna translacija

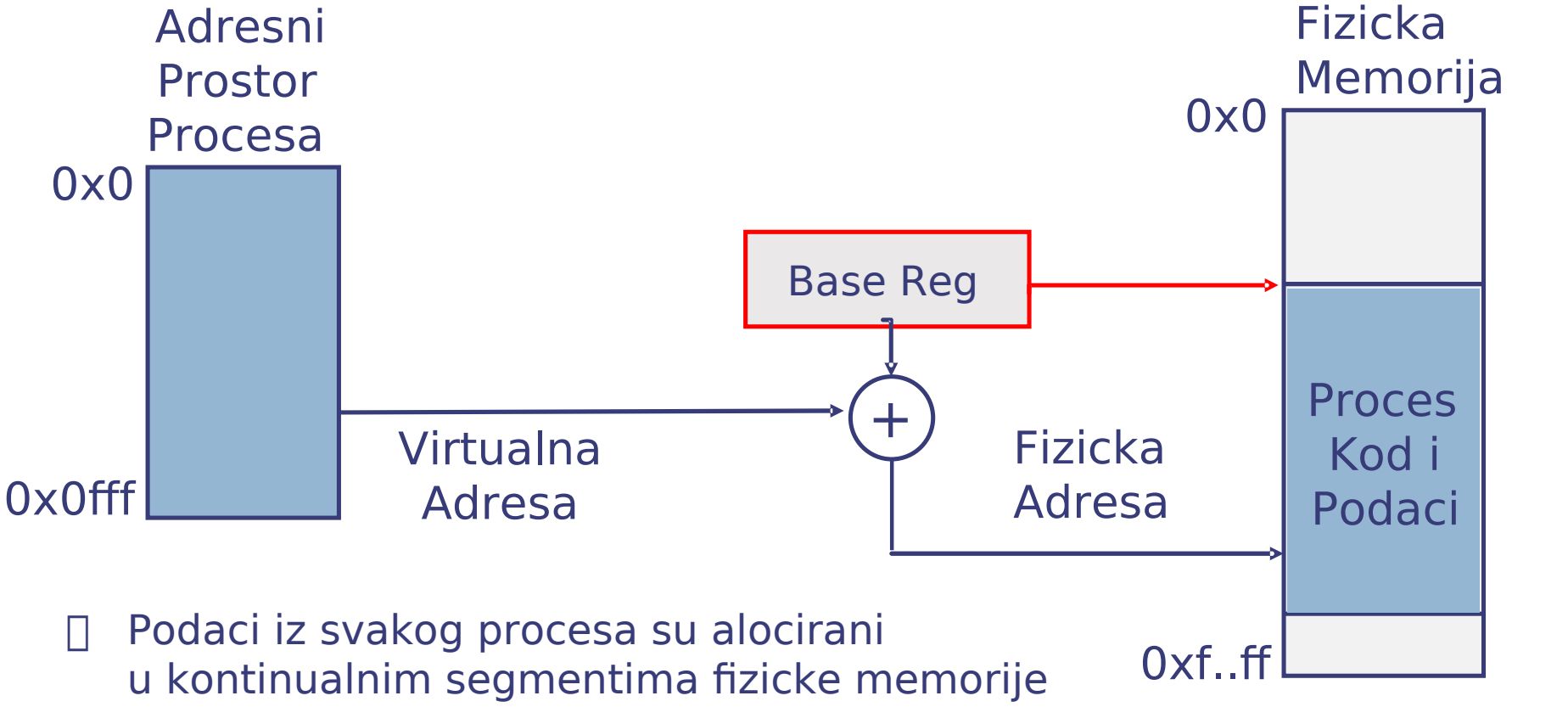


- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije

Segmentacija (Base-and-Bound) Adresna translacija

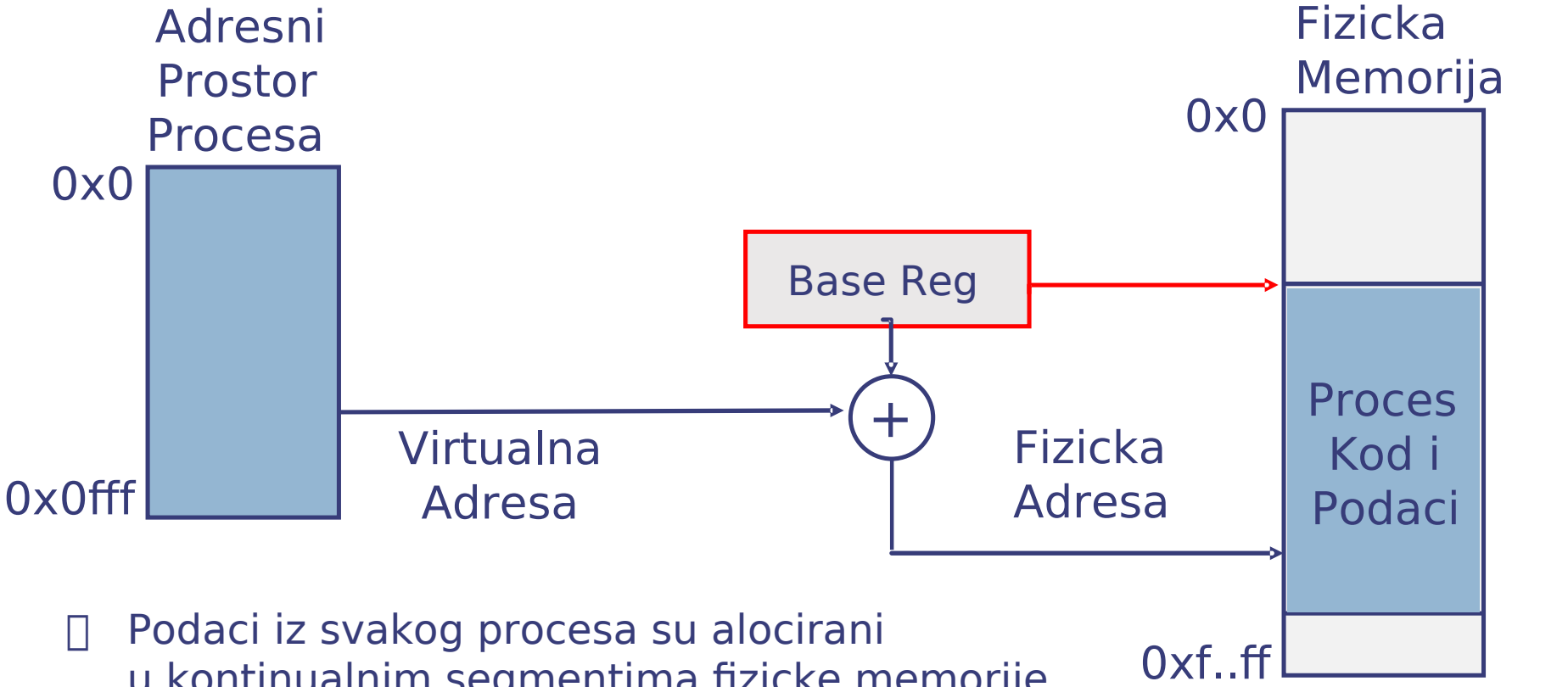
- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije

Segmentacija (Base-and-Bound) Adresna translacija



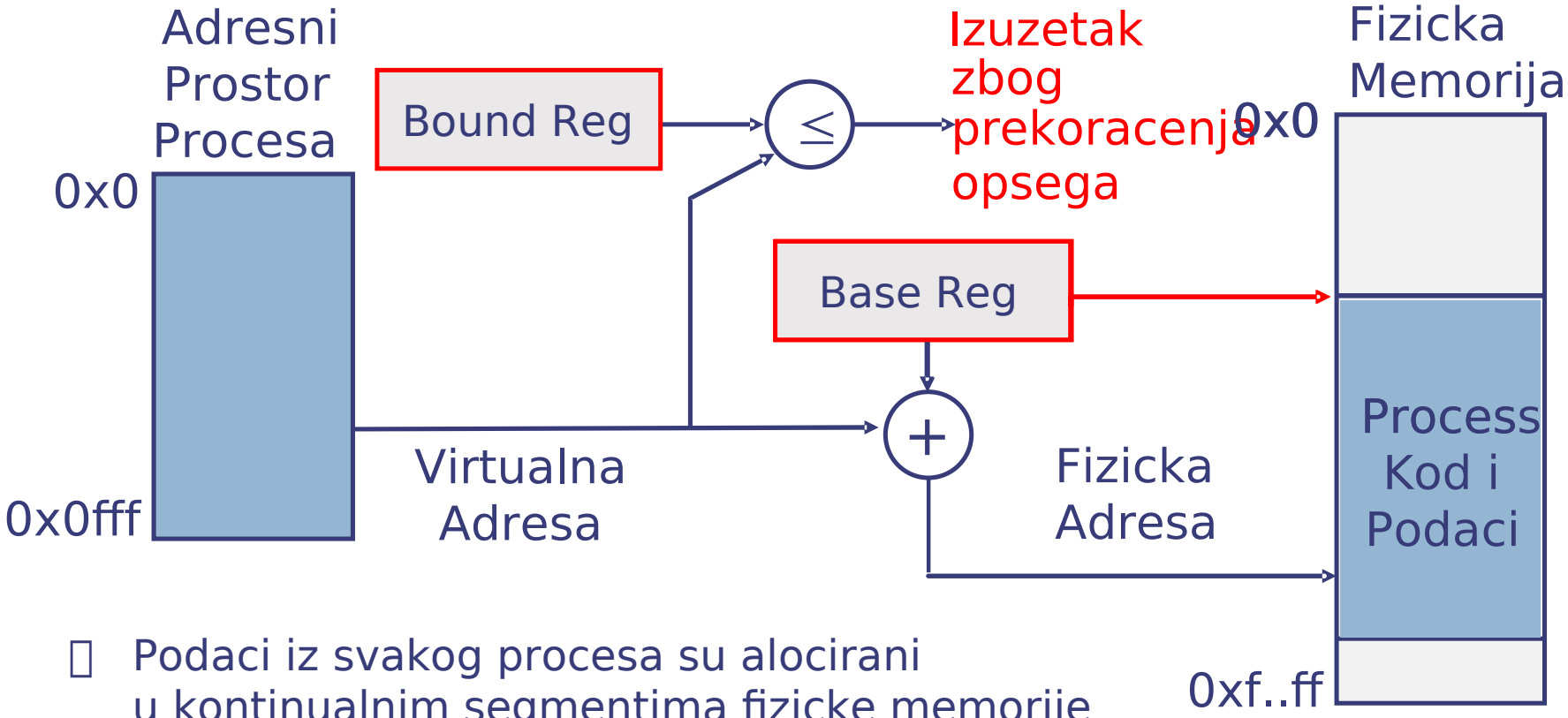
- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije

Segmentacija (Base-and-Bound) Adresna translacija



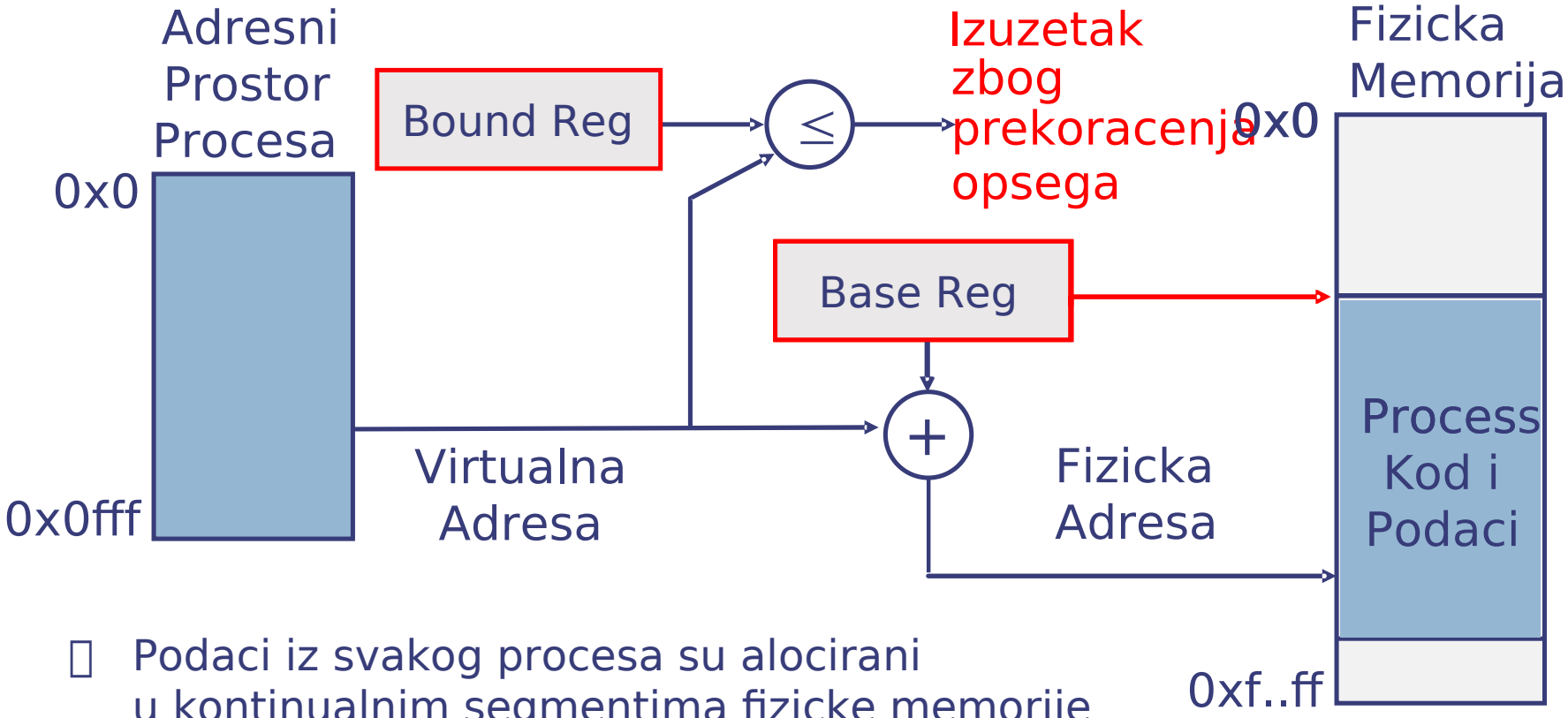
- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije
- Fizicka adresa = Virtualna adresa + bazna adresa segmenta

Segmentacija (Base-and-Bound) Adresna translacija



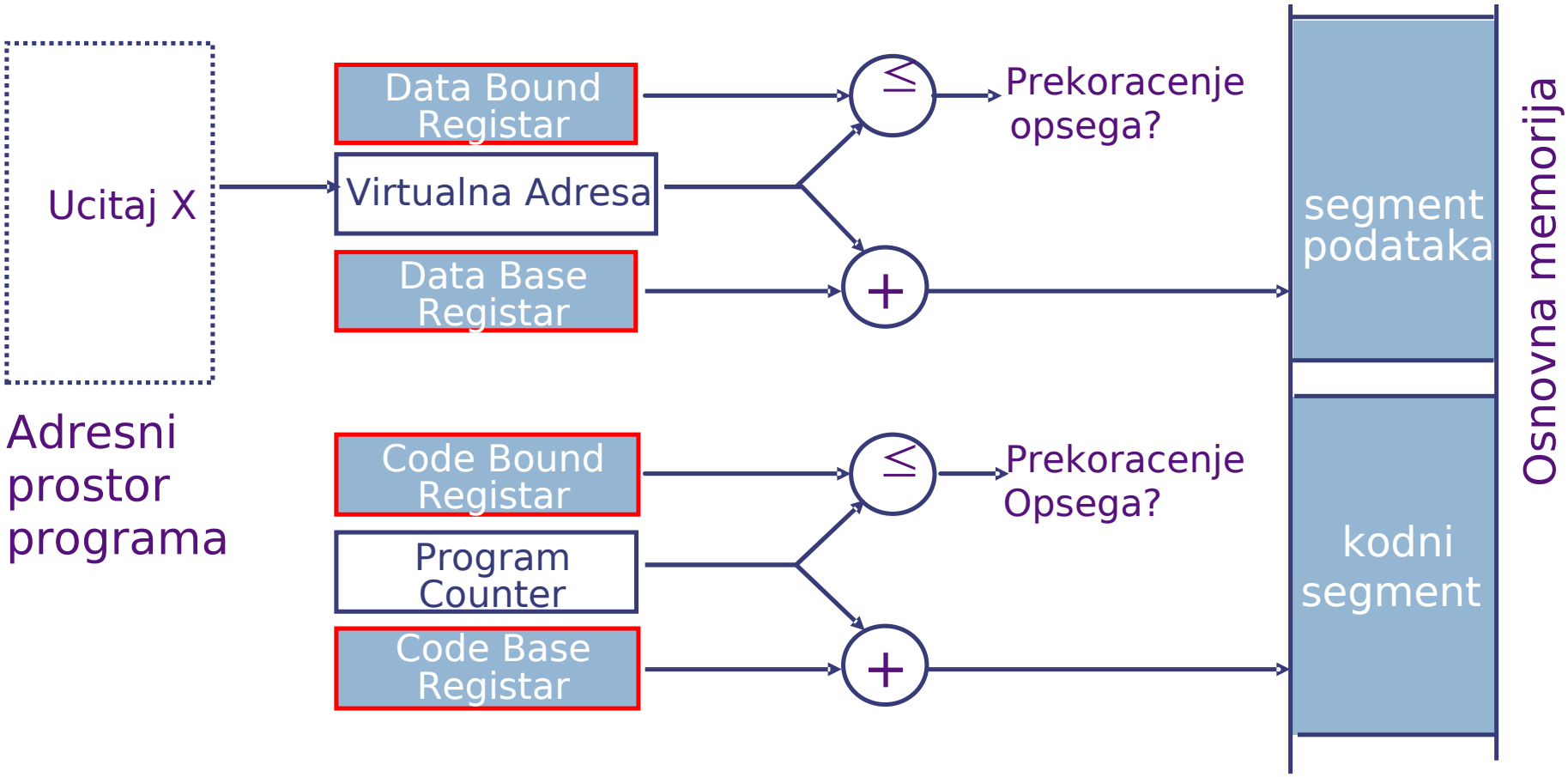
- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije
- Fizicka adresa = Virtualna adresa + bazna adresa segmenta
- Bound registar obezbedjuje bezbednost i izolaciju

Segmentacija (Base-and-Bound) Adresna translacija

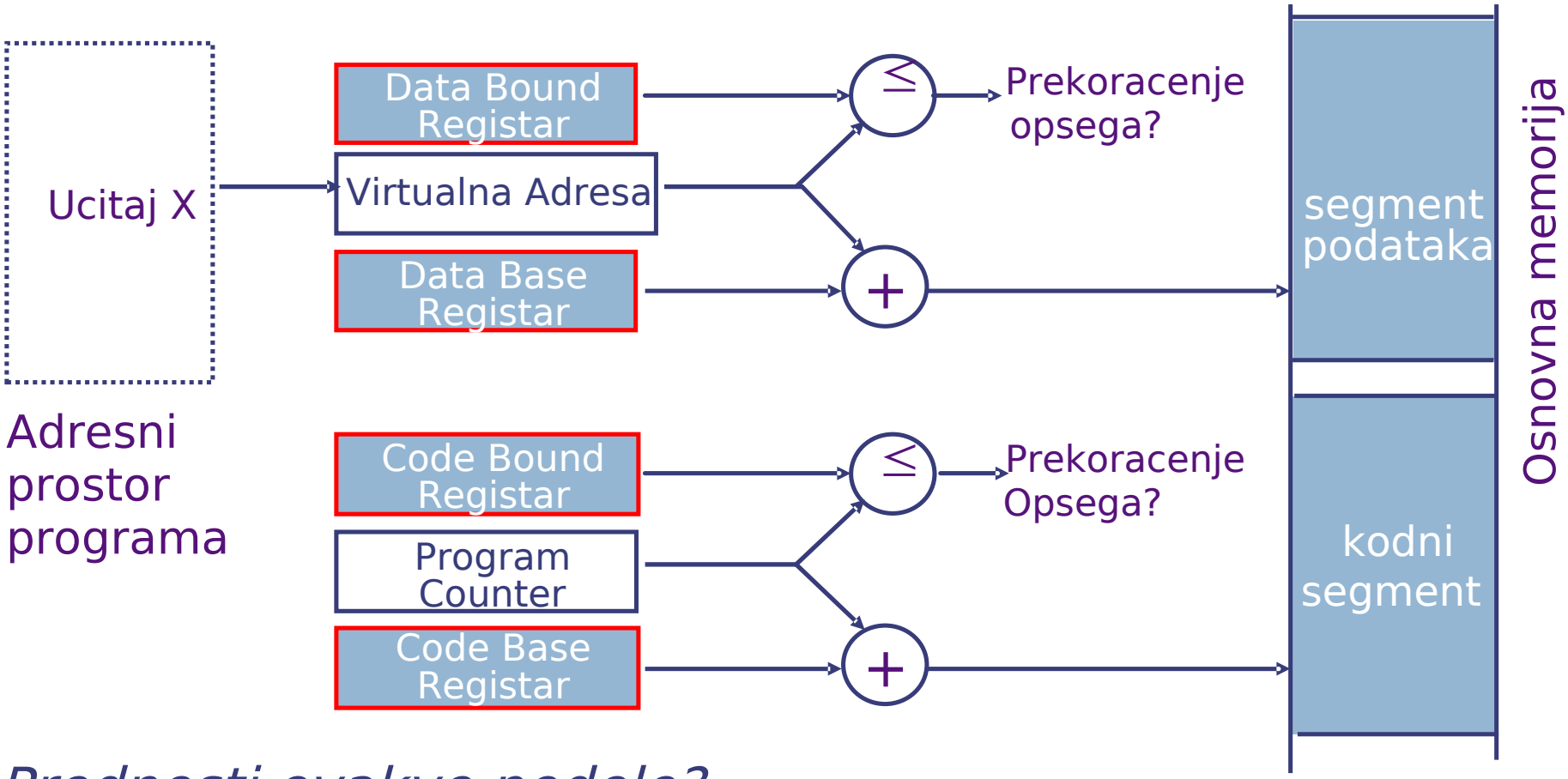


- Podaci iz svakog procesa su alocirani u kontinualnim segmentima fizicke memorije
- Fizicka adresa = Virtualna adresa + bazna adresa segmenta
- Bound registar obezbedjuje bezbednost i izolaciju
- Base i Bound registrima ne treba pristupati od strane koristnickih progr. (jedino su dostupni u nadziranom modu)

Odvojeni segmenti za kod i podatke

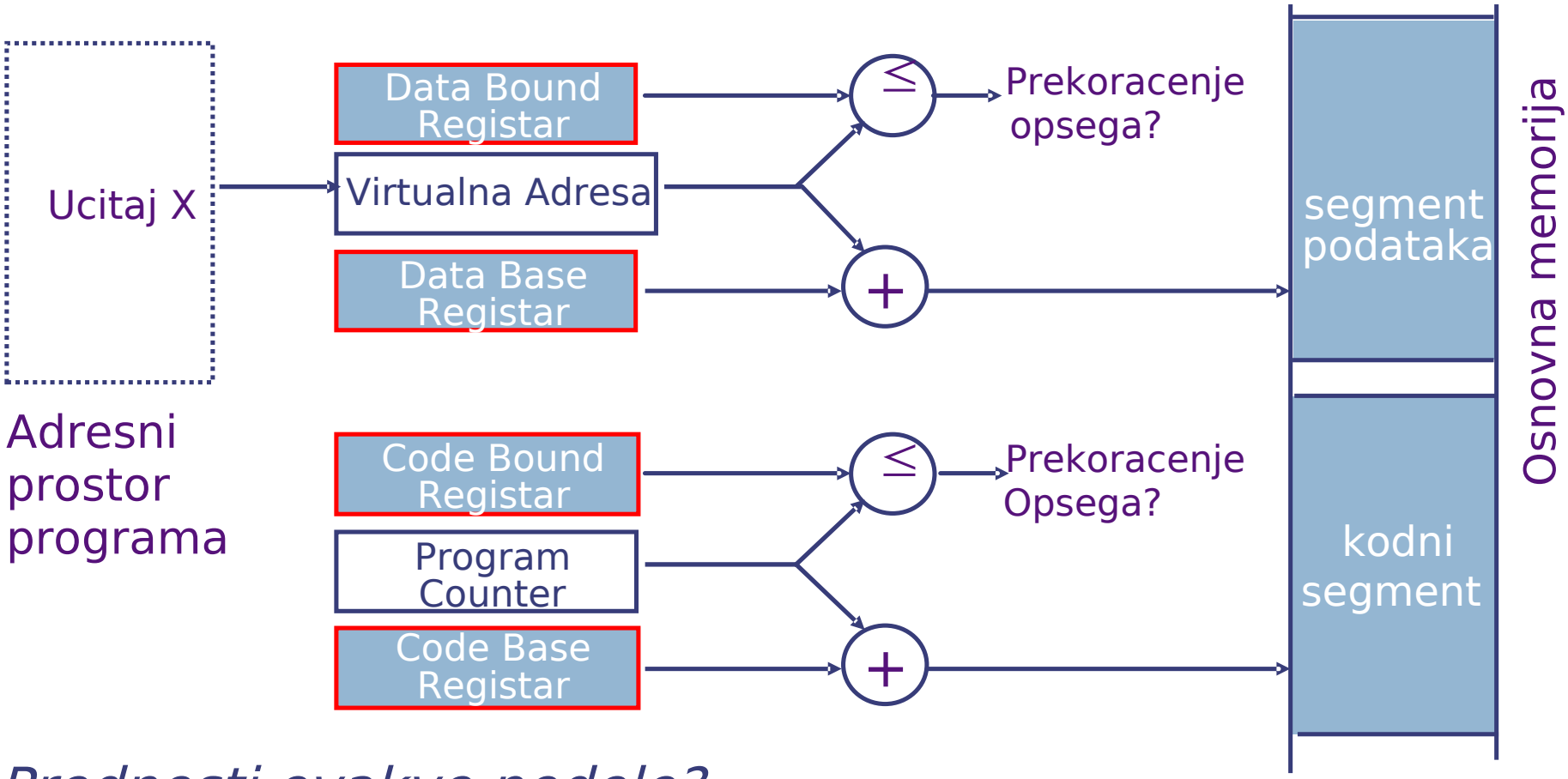


Odvojeni segmenti za kod i podatke



Prednosti ovakve podele?

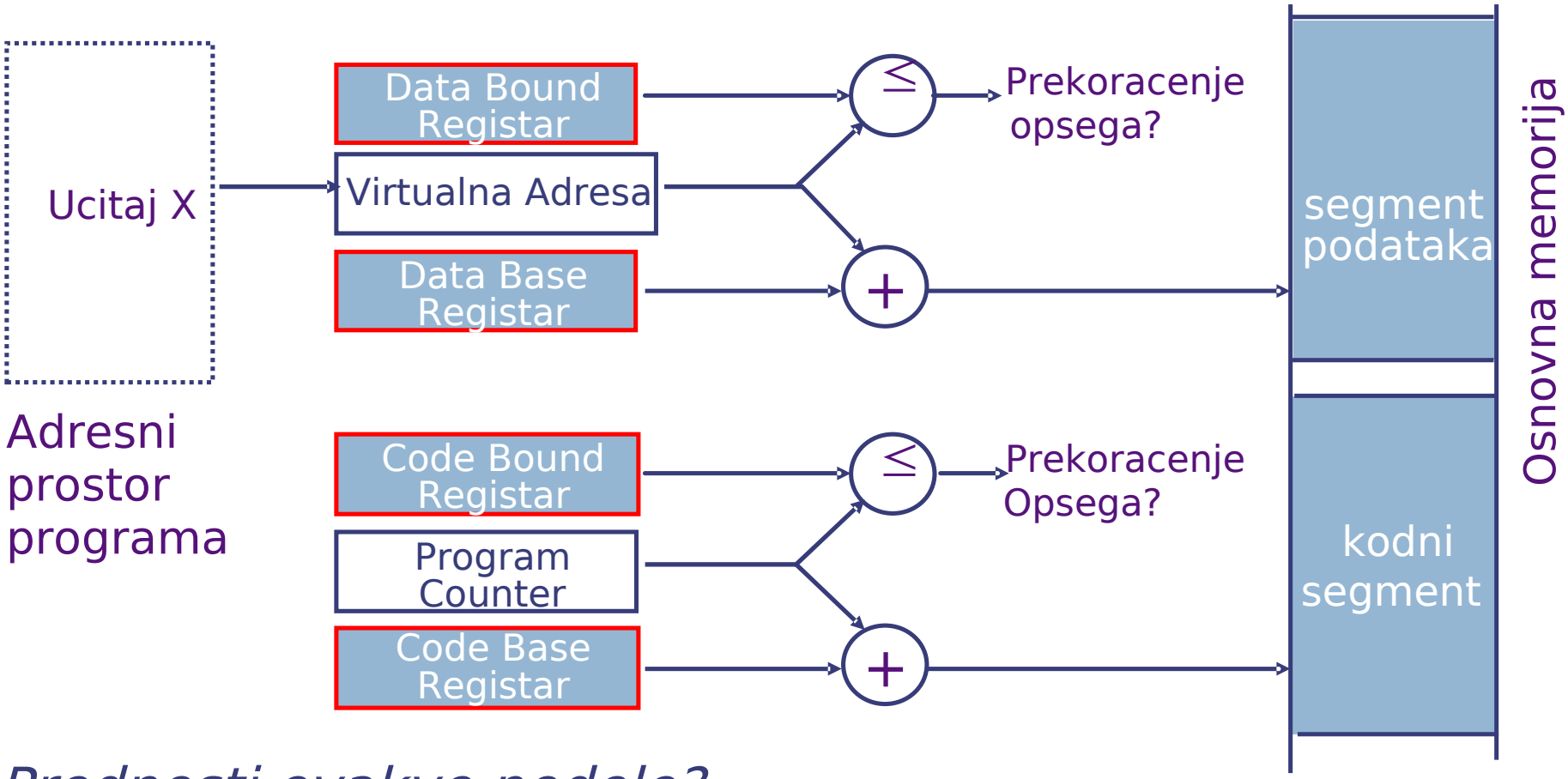
Odvojeni segmenti za kod i podatke



Prednosti ovakve podele?

Sprecava bagovite programe da greskom prepisu kod

Odvojeni segmenti za kod i podatke



Prednosti ovakve podele?

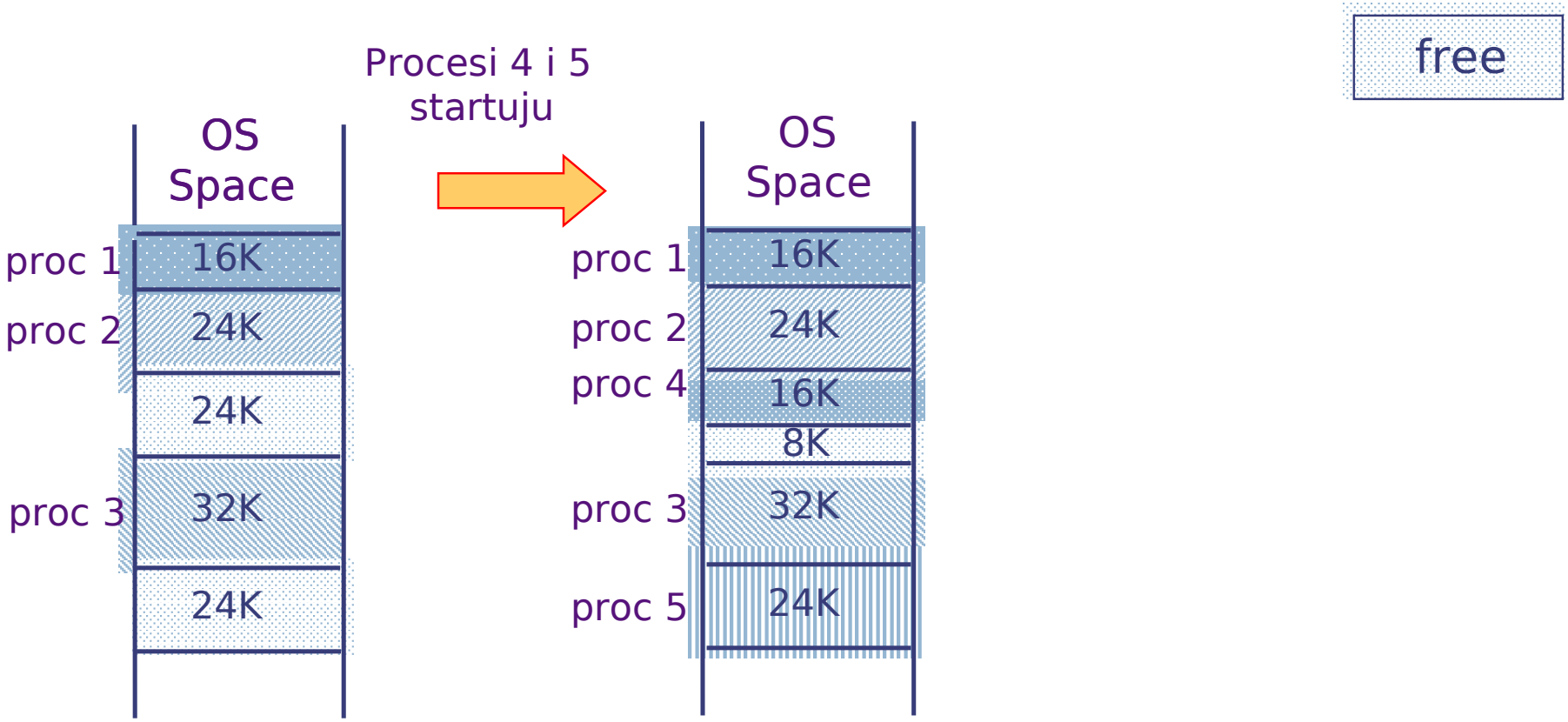
Sprecava bagovite programe da greskom prepisu kod

Vise procesa mogu da dele isti kodni segment

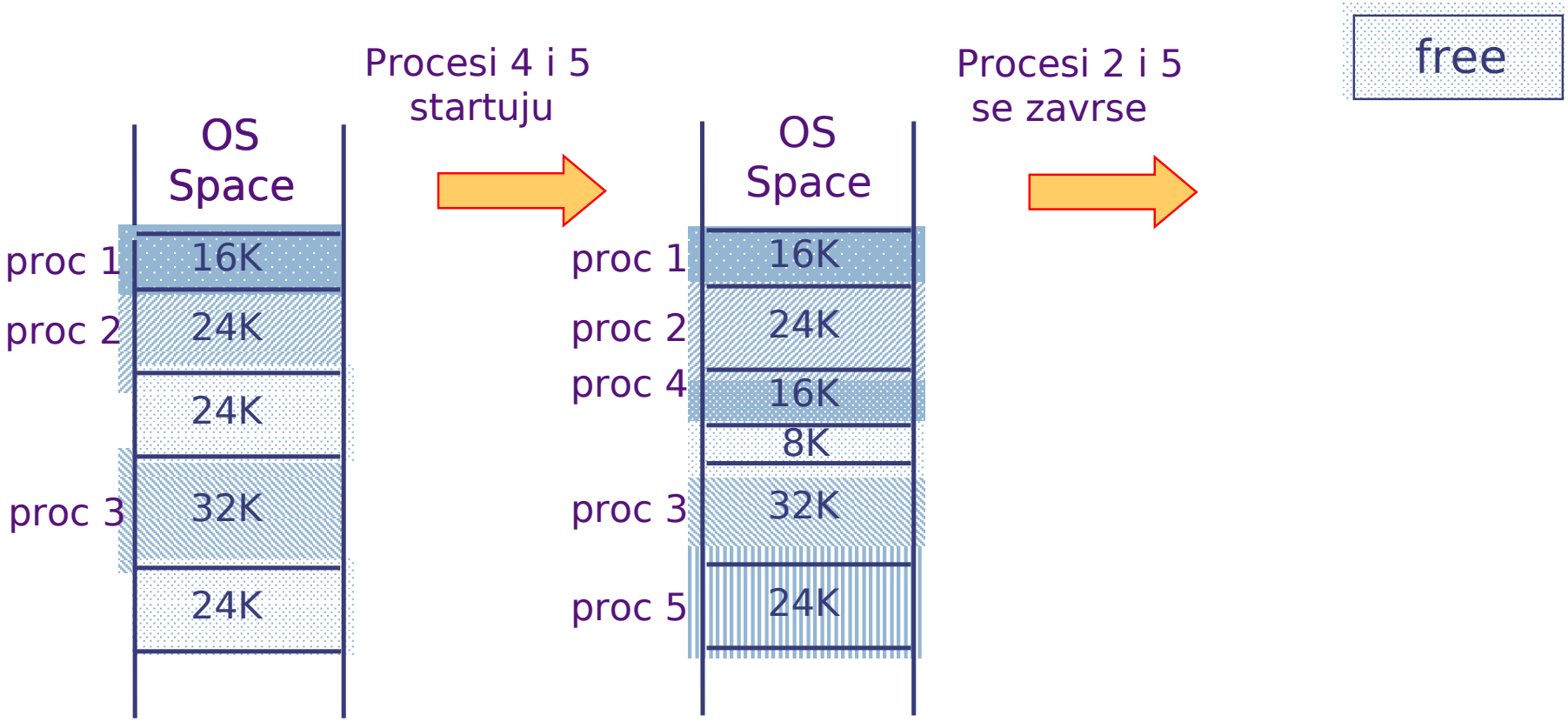
Fragmentacija memorije

Fragmentacija memorije

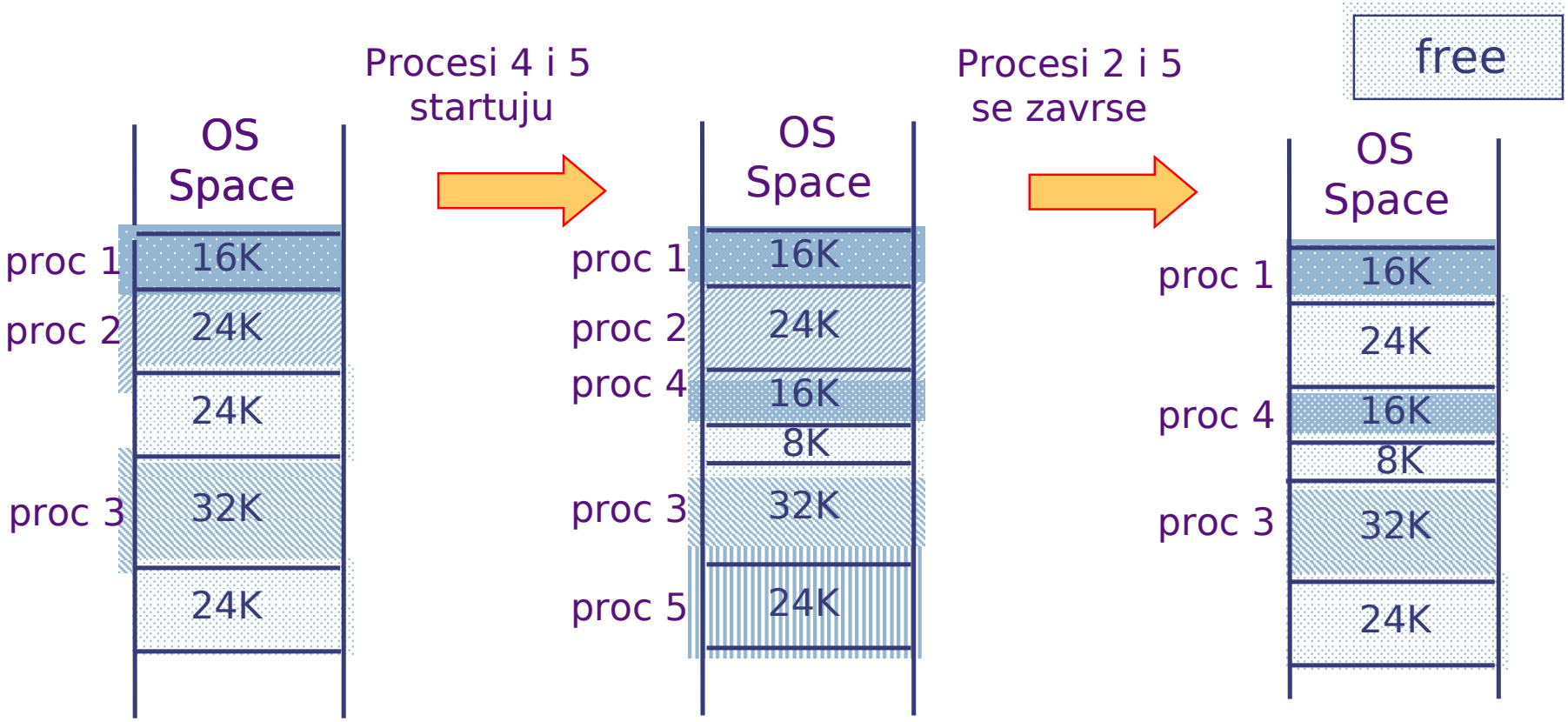
Fragmentacija memorije



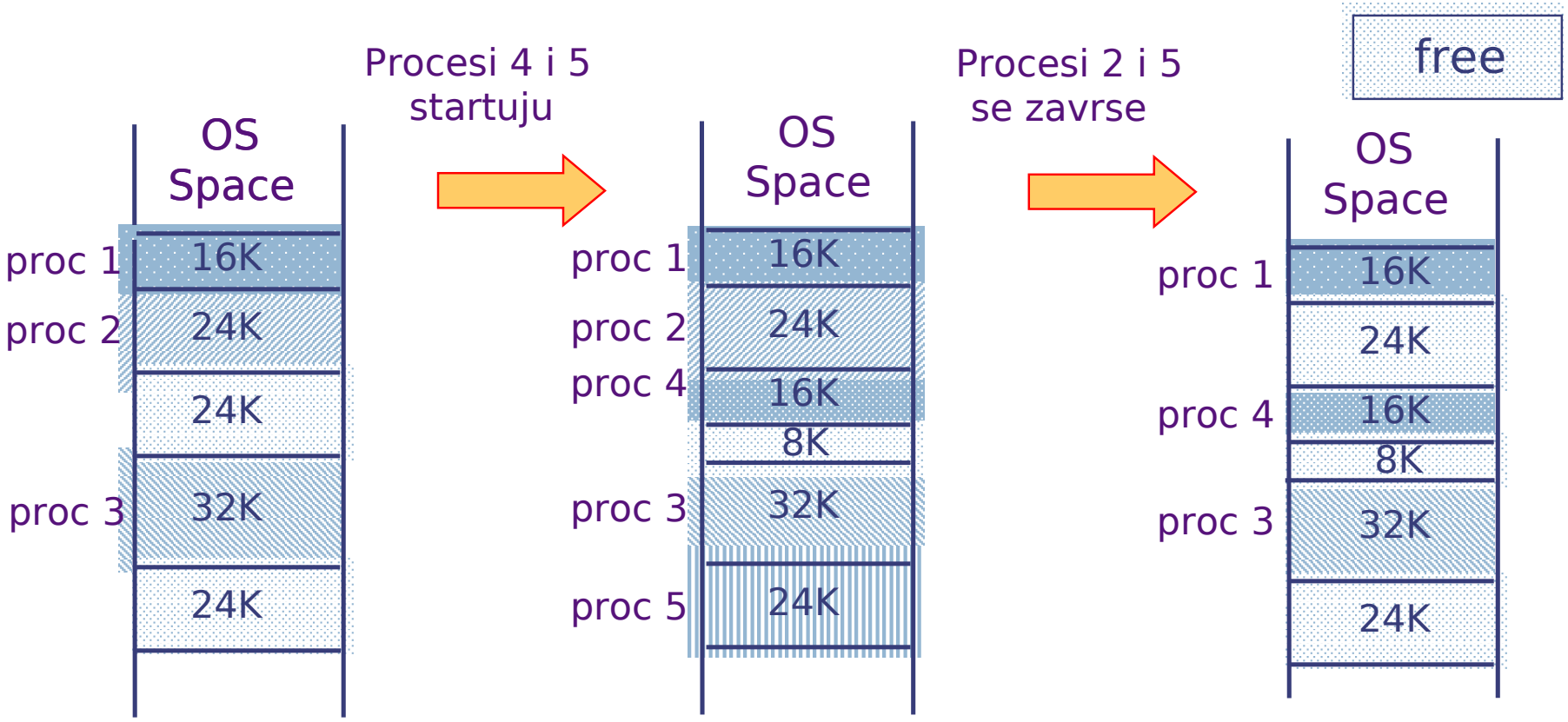
Fragmentacija memorije



Fragmentacija memorije



Fragmentacija memorije



Kako procesi pocinju i zavrsavaju se, memorija je “”Fragmentirana”. U nekom trenutku segmenti moraju Biti premestani kako bi se dobio kompaktan slobodni Prostor.

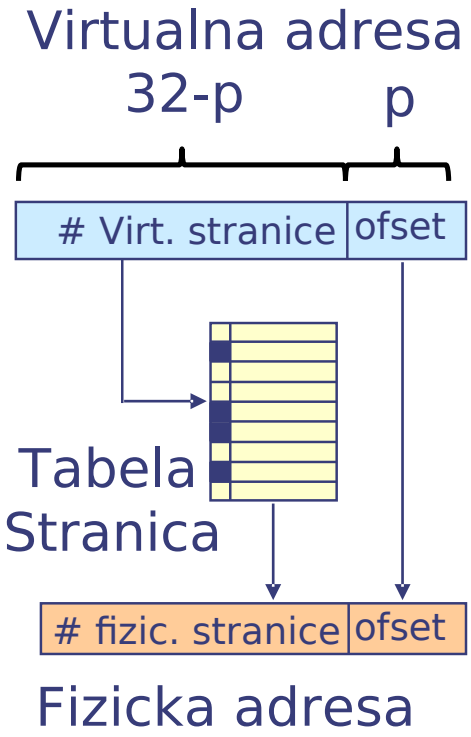
Koriscenje stranica (Pages)

- Izdeli fizicku memoriju u blokove fiksne velicine: stranice
  - Tipicna velicina stranice: 4KB

Koriscenje stranica (Pages)

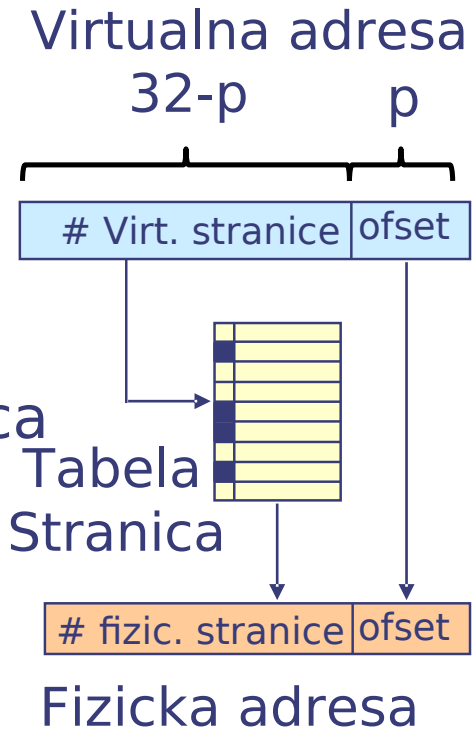
- Izdeli fizicku memoriju u blokove fiksne velicine: stranice
  - Tipicna velicina stranice: 4KB

- Interpretiraj virtualnu adresu kao par <broj virtualne stranice, ofset>

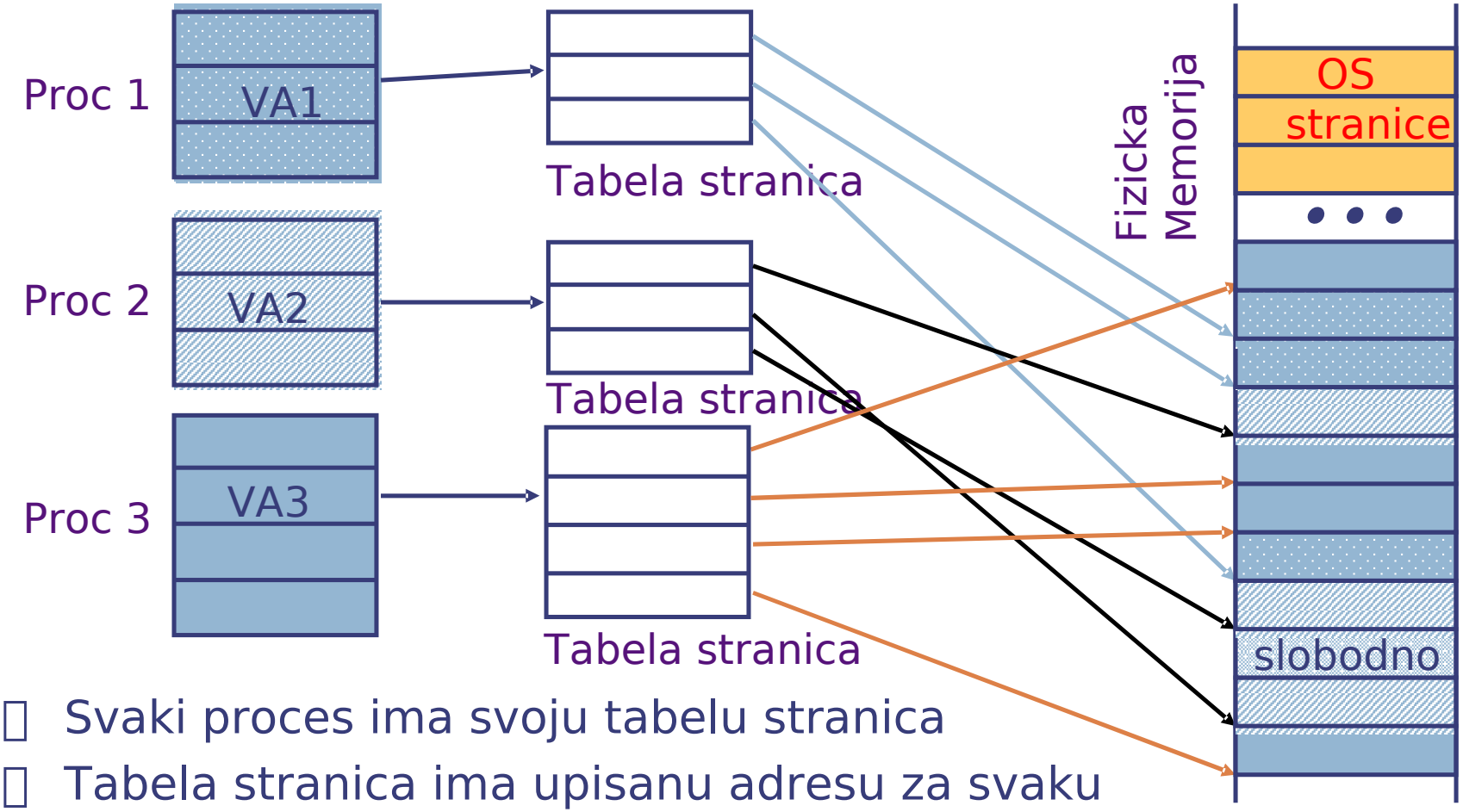


Koriscenje stranica (Pages)

- Izdeli fizicku memoriju u blokove fiksne velicine: stranice
  - Tipicna velicina stranice: 4KB
- Interpretiraj virtualnu adresu kao par <broj virtualne stranice, ofset>
- Koristi tabelu stranica za translaciju iz brojeva virtualnih u brojeve fizickih stranica
  - Tabela stranica sadrzi broj fizicke stranice (tj. pocetnu fizicku adresu) za svaki broj virtualne stranice

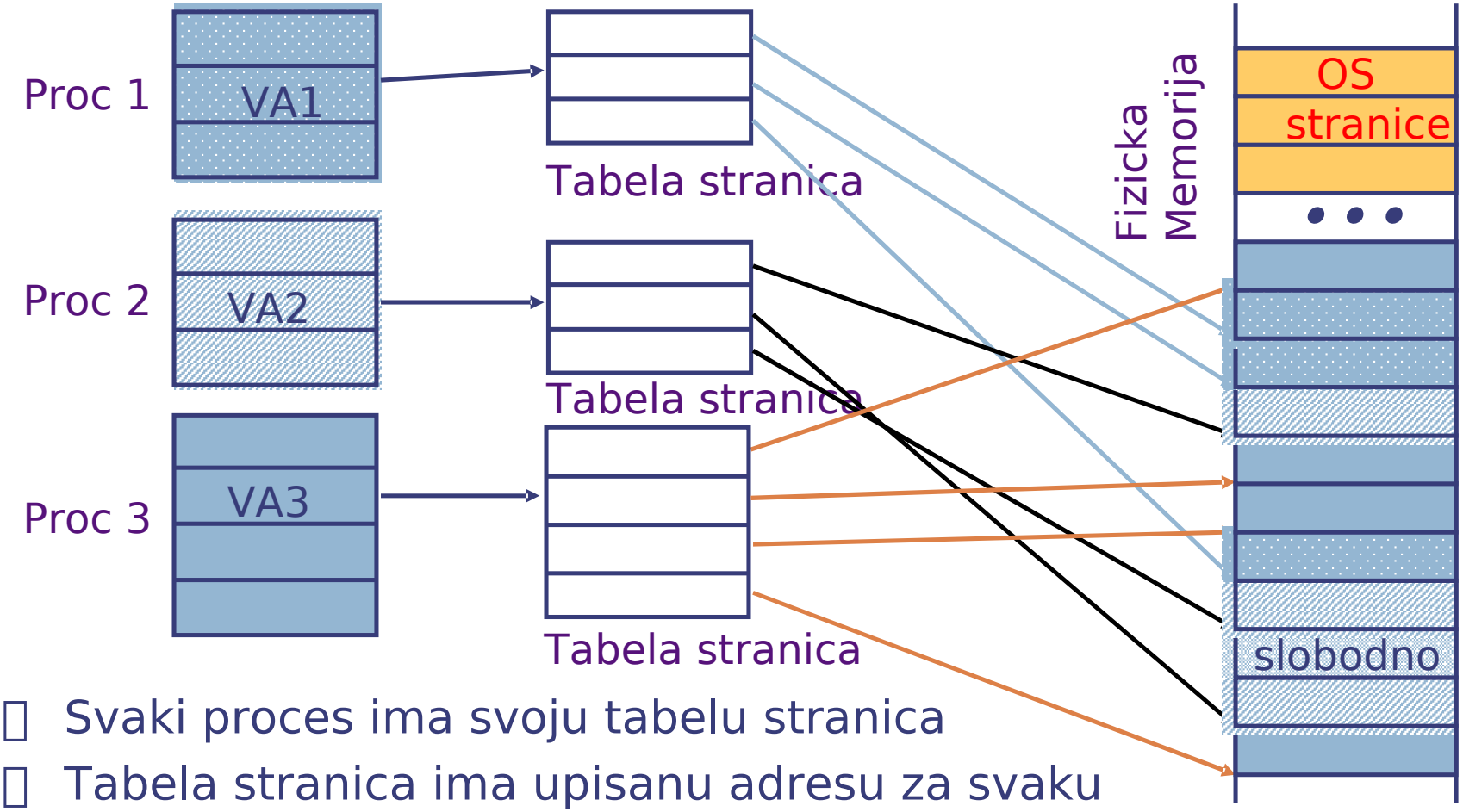


Privatni adresni prostor svakom proc.



- Svaki proces ima svoju tabelu stranica
- Tabela stranica ima upisanu adresu za svaku stranicu dodeljenu procesu

Privatni adresni prostor svakom proc.



- Svaki proces ima svoju tabelu stranica
- Tabela stranica ima upisanu adresu za svaku stranicu dodeljenu procesu

Tabele stranica omogucavaju smestanje stranica programa na ne-kontinualne lokacije

Stranice ili Segmentacija

Prednosti?

Stranice ili Segmentacija

Prednosti?

Koriscenje stranica sprecava fragmentaciju

Stranice ili Segmentacija

Prednosti?

Koriscenje stranica sprecava fragmentaciju

Stranice omogucavaju programu
vise virtualne memorije od postojece fizicke memorije (stranice na zahtev)

Stranice ili Segmentacija

Prednosti?

Koriscenje stranica sprecava fragmentaciju

Stranice omogucavaju programu
vise virtualne memorije od postojece fizicke memorije (stranice na zahtev)

Mane?

Stranice ili Segmentacija

Prednosti?

Koriscenje stranica sprecava fragmentaciju

Stranice omogucavaju programu
vise virtualne memorije od postojece fizicke memorije (stranice na zahtev)

Mane?

Tabele stranica su MNOGO vece od base/bound registara!

Stranice ili Segmentacija

Prednosti?

Koriscenje stranica sprecava fragmentaciju

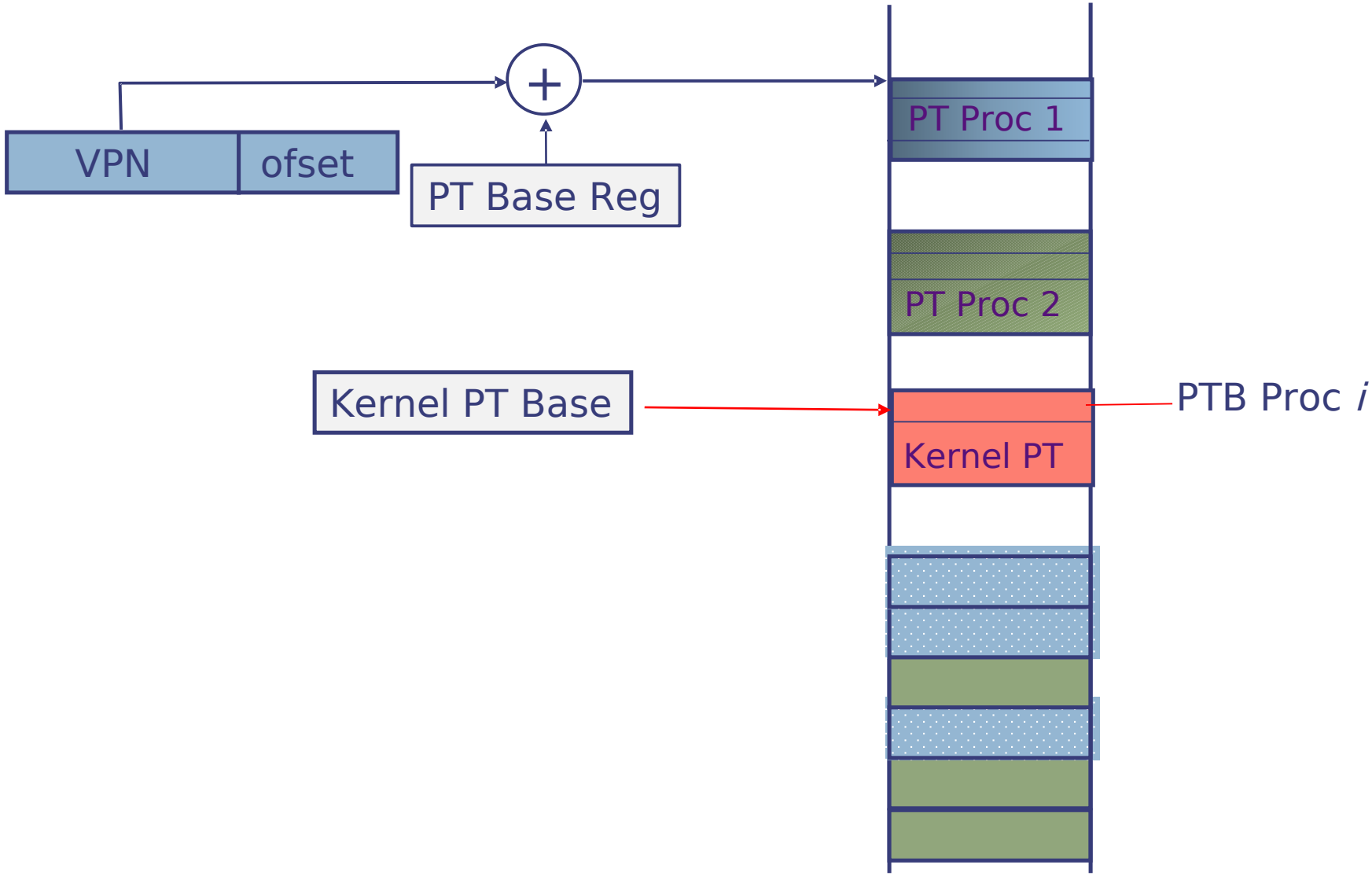
Stranice omogucavaju programu
vise virtualne memorije od postojece fizicke memorije (stranice na zahtev)

Mane?

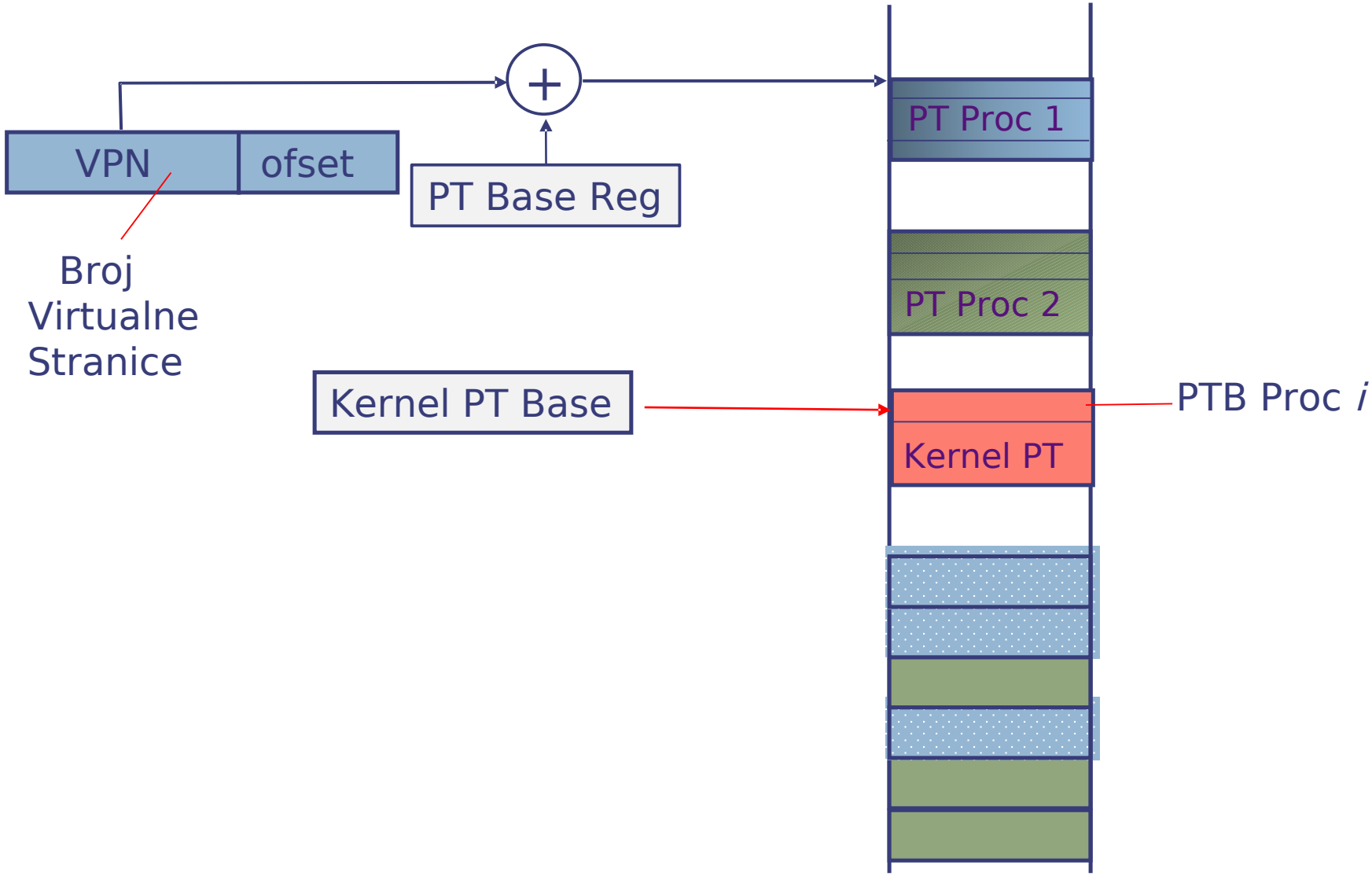
Tabele stranica su MNOGO vece od base/bound registara!

... gde skladistiti same tabele stranica?

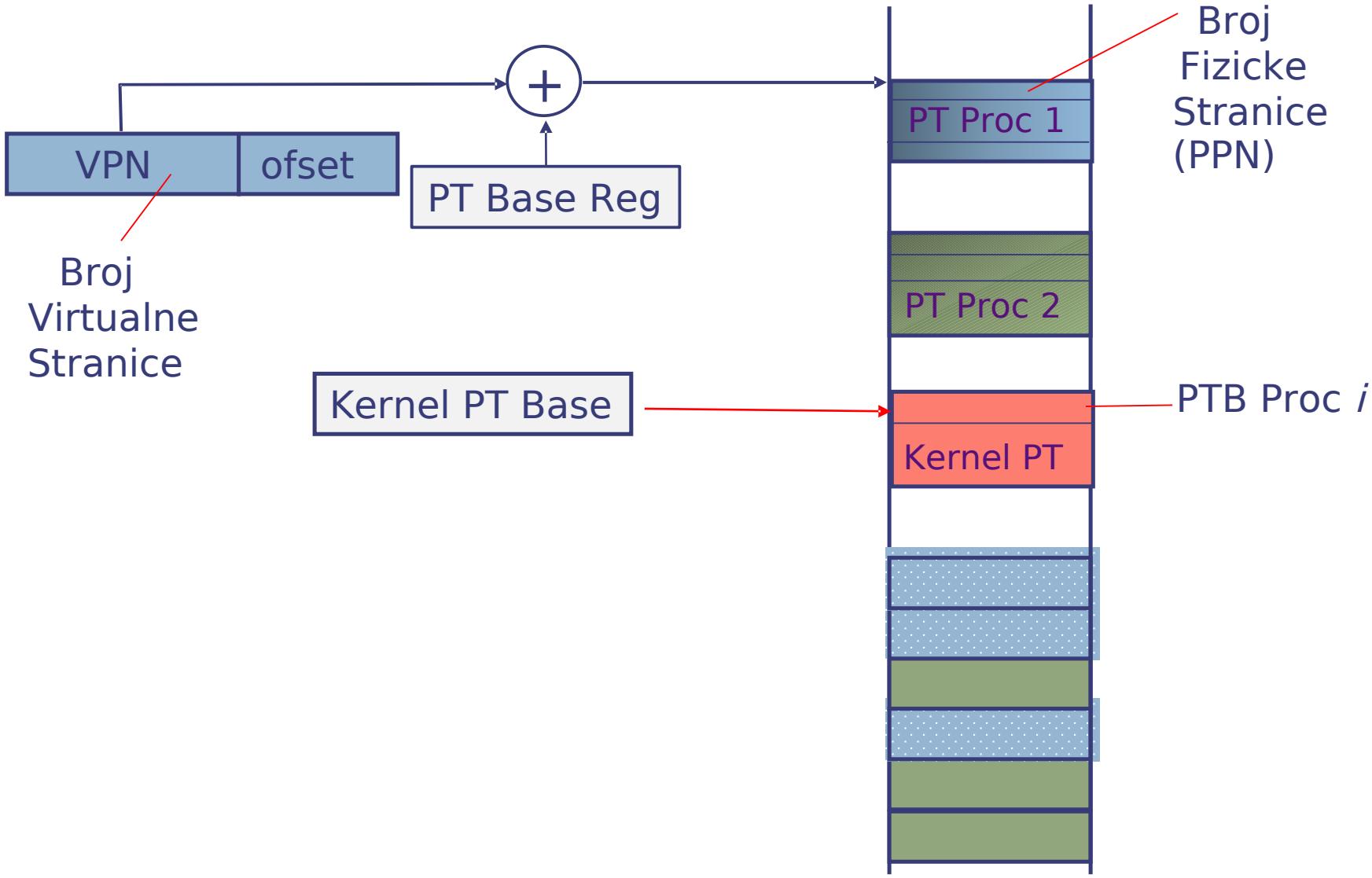
Tabele stranica smestene u memoriji?



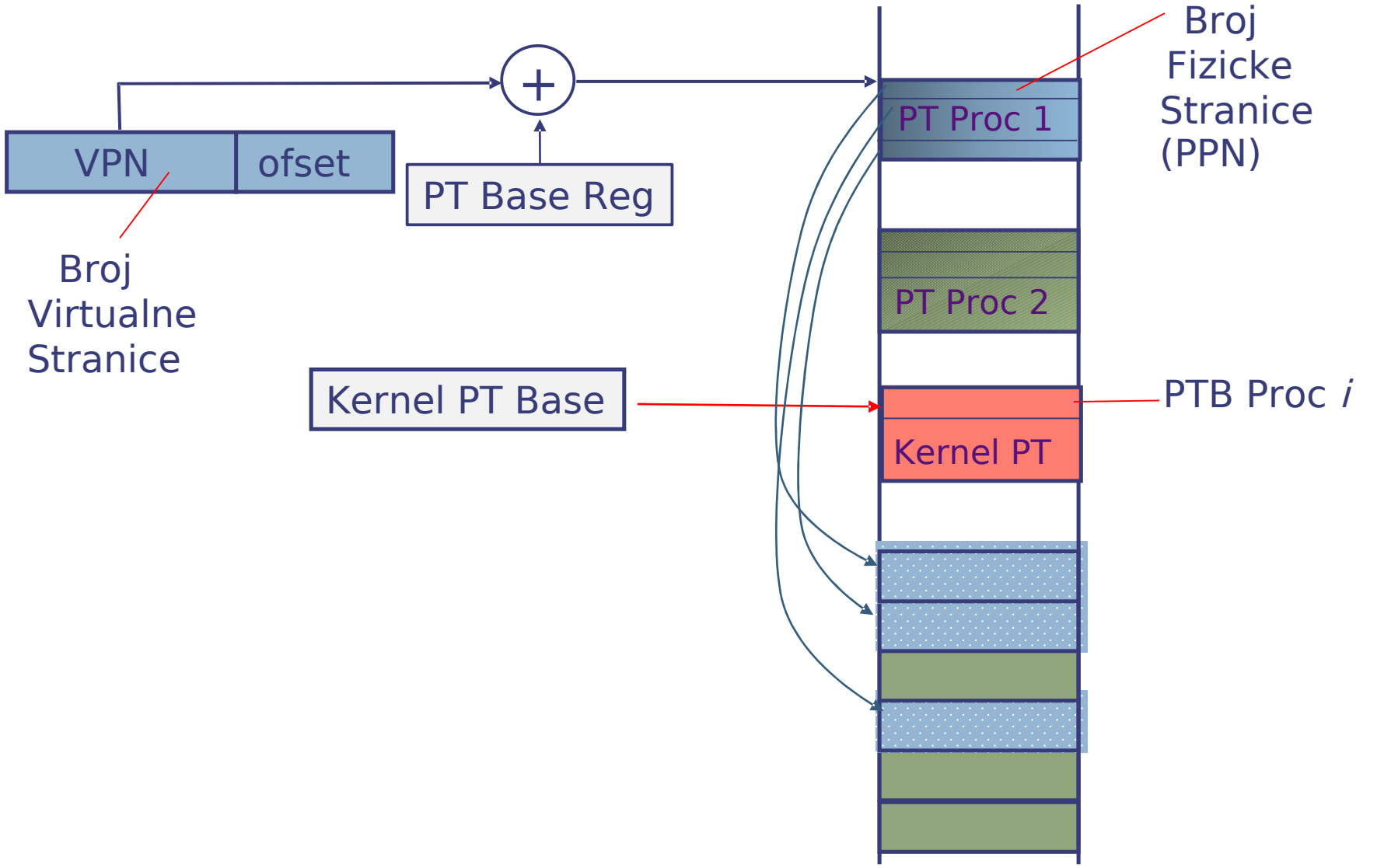
Tabele stranica smestene u memoriji?



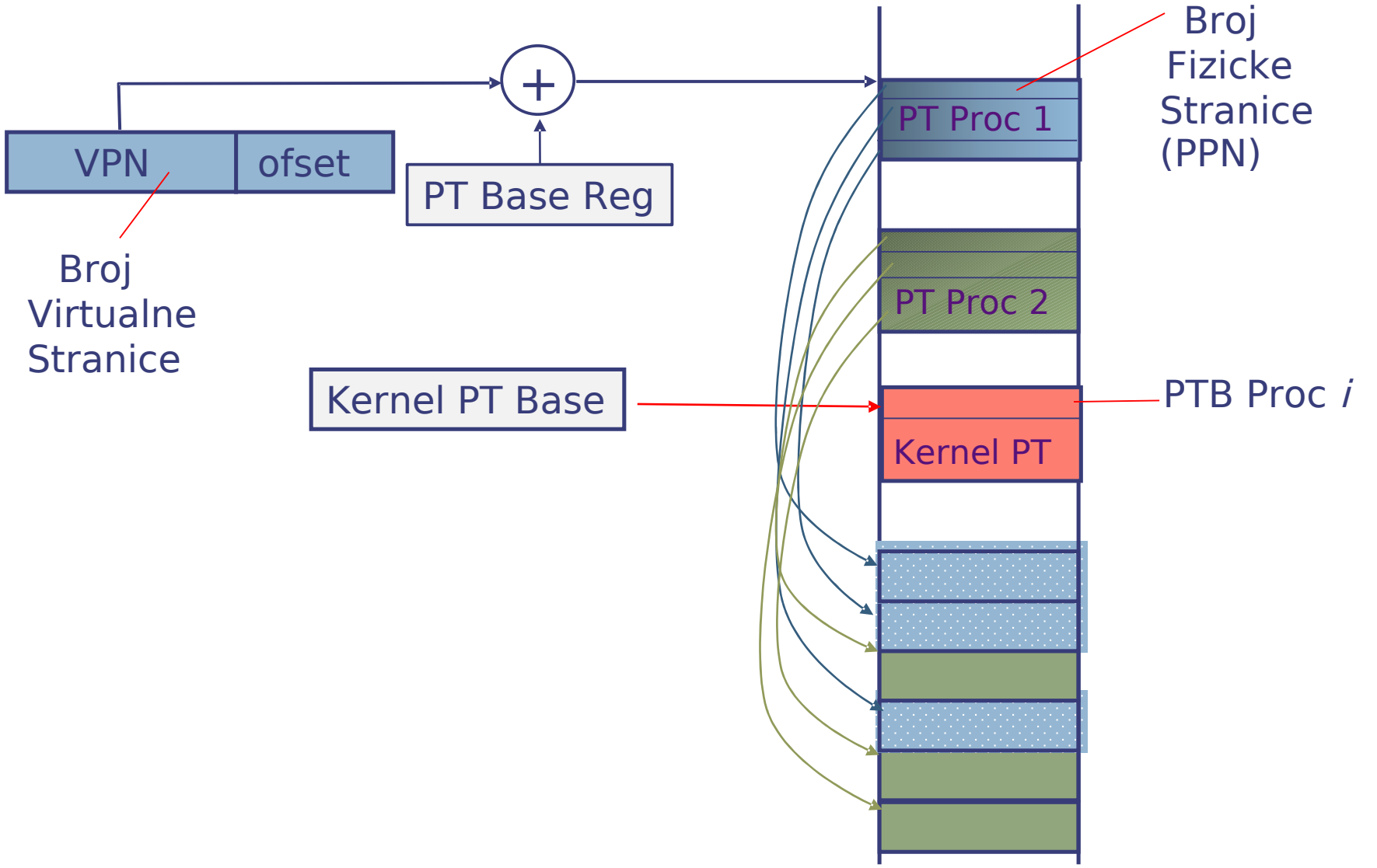
Tabele stranica smestene u memoriji?



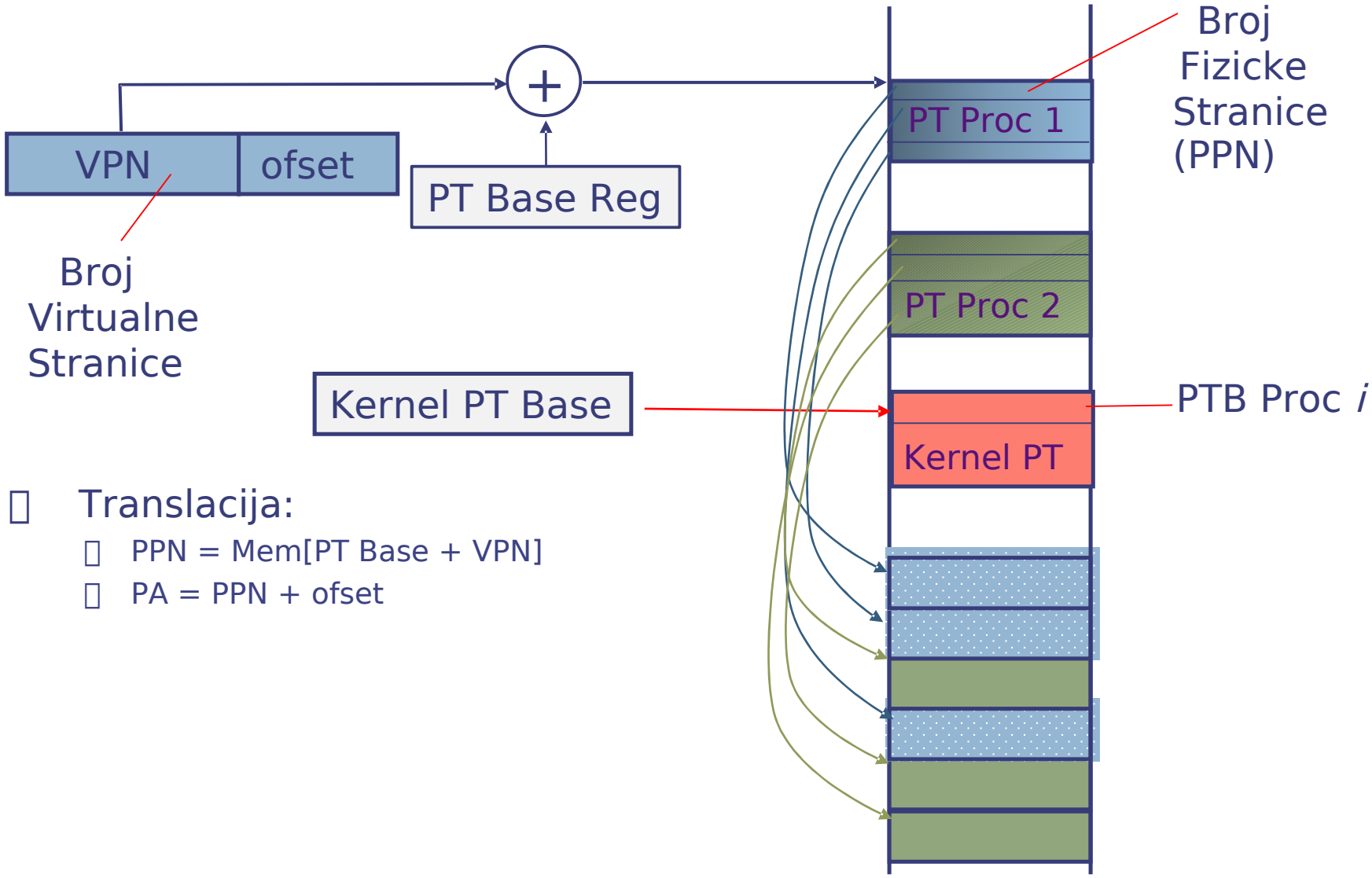
Tabele stranica smestene u memoriji?



Tabele stranica smestene u memoriji?

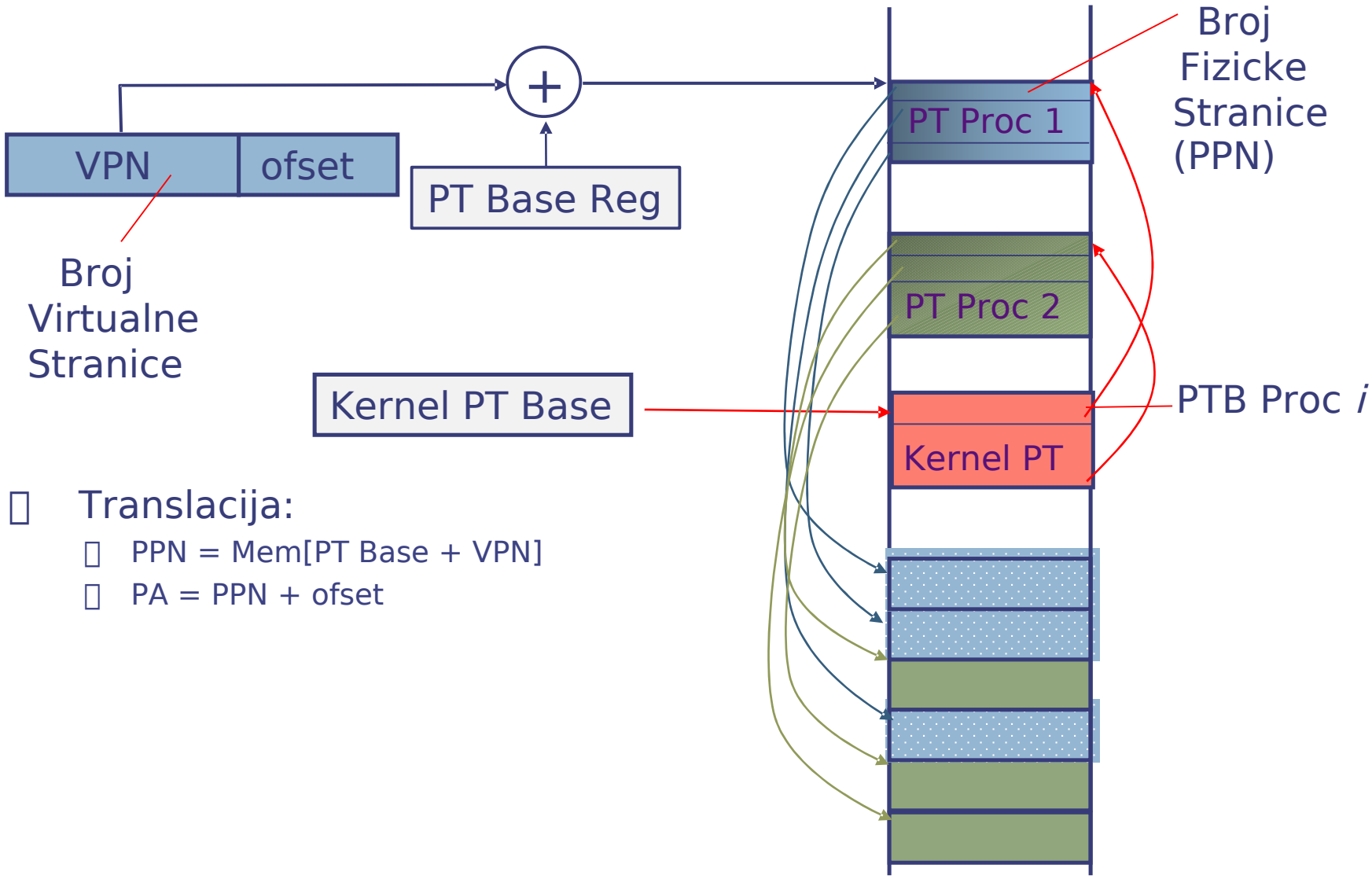


Tabele stranica smestene u memoriji?



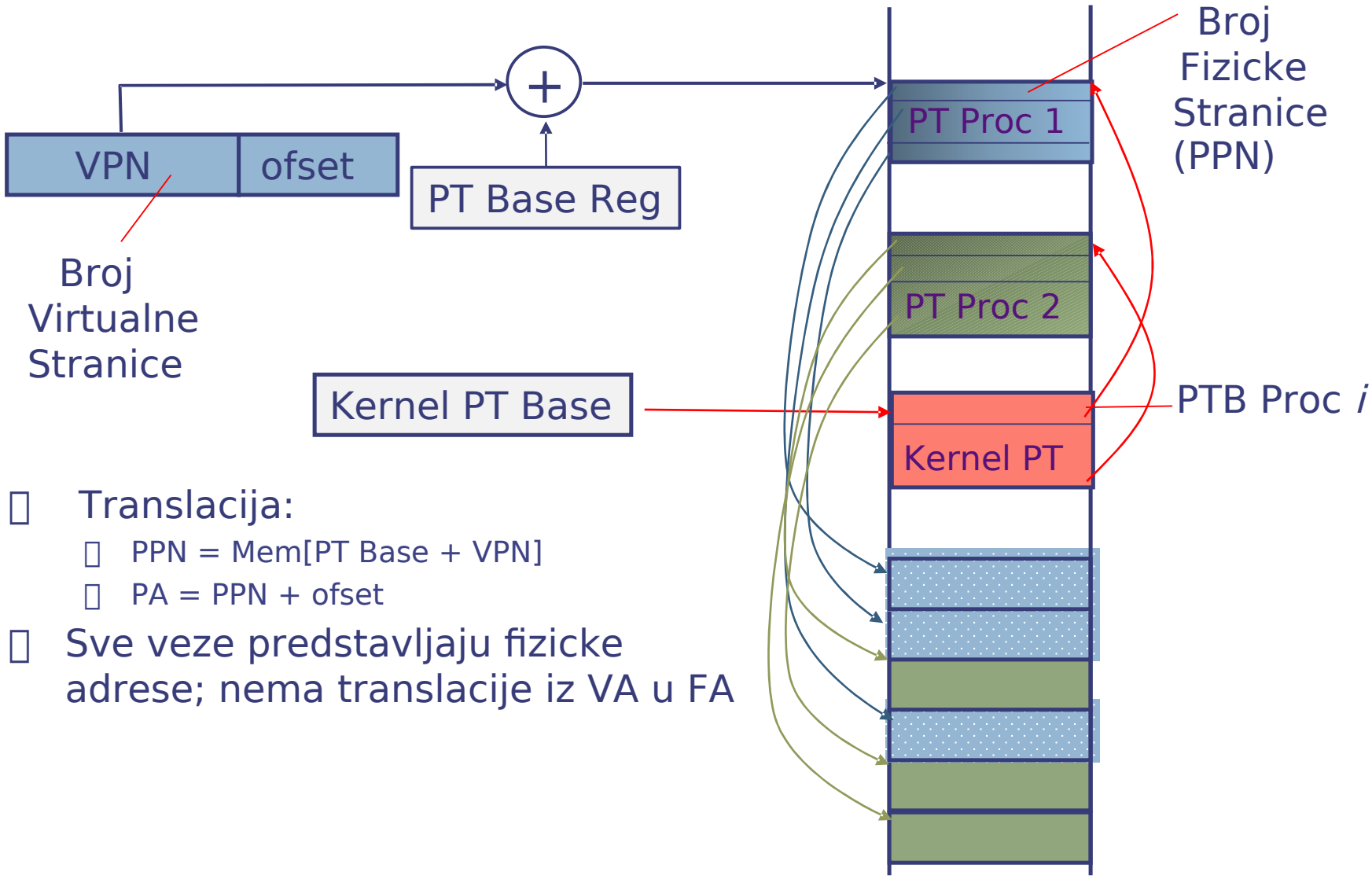
- Translacija:
  - PPN = Mem[PT Base + VPN]
  - PA = PPN + ofset

Tabele stranica smestene u memoriji?



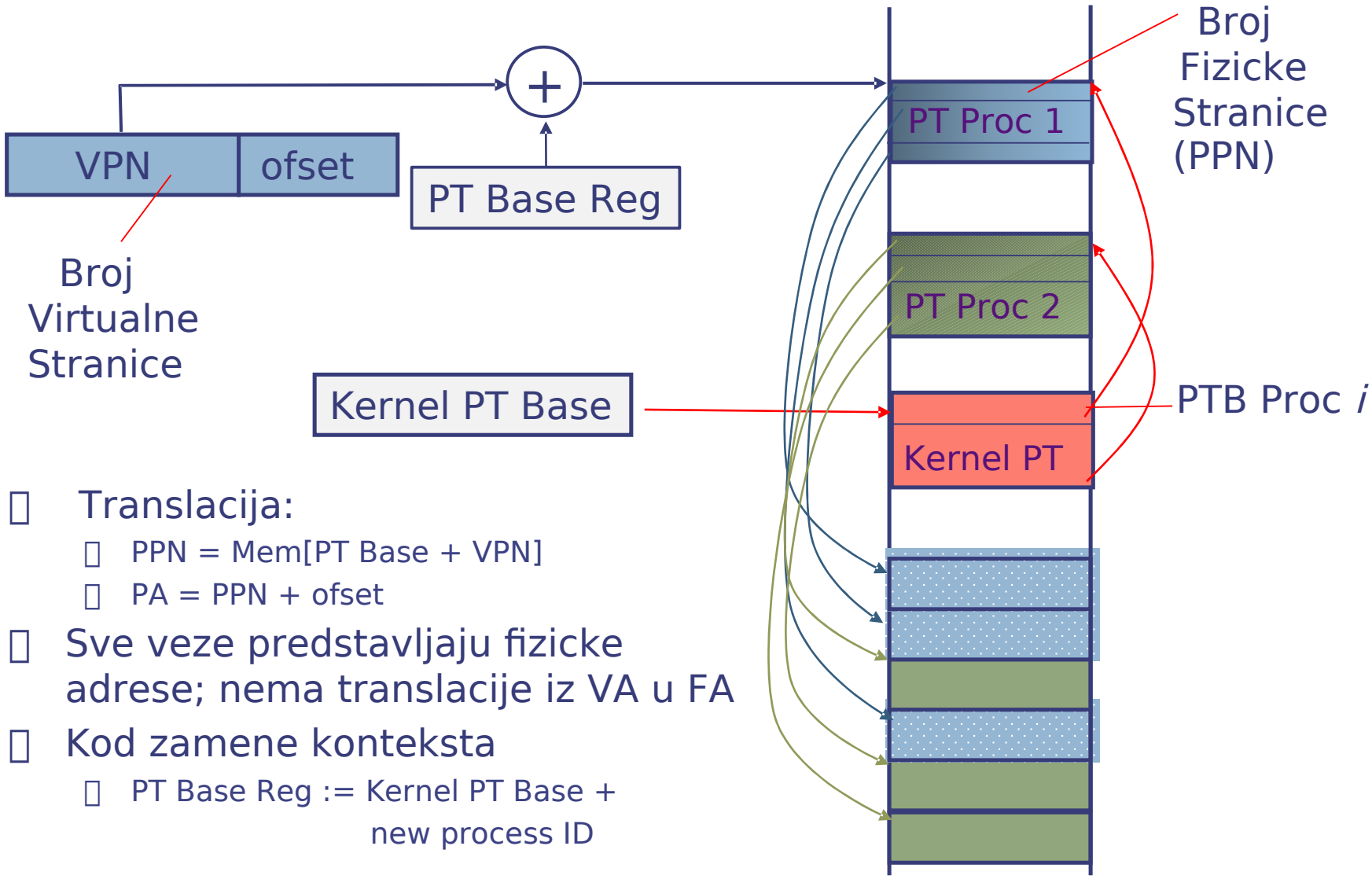
- Translacija:
  - PPN = Mem[PT Base + VPN]
  - PA = PPN + ofset

Tabele stranica smestene u memoriji?



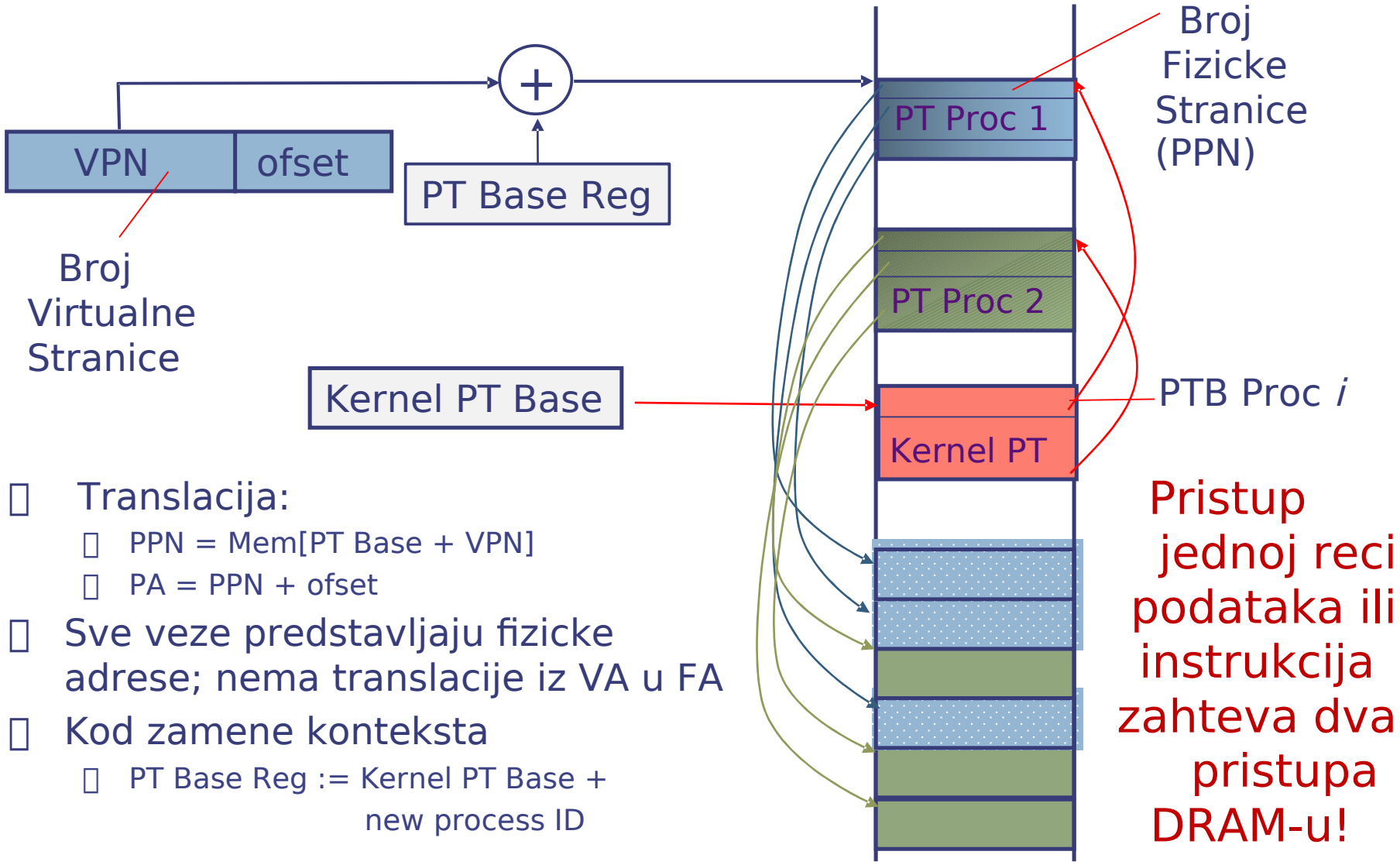
- Translacija:
  - PPN = Mem[PT Base + VPN]
  - PA = PPN + ofset
- Sve veze predstavljaju fizicke adrese; nema translacije iz VA u FA

Tabele stranica smestene u memoriji?



- Translacija:
  - PPN = Mem[PT Base + VPN]
  - PA = PPN + ofset
- Sve veze predstavljaju fizicke adrese; nema translacije iz VA u FA
- Kod zamene konteksta
  - PT Base Reg := Kernel PT Base + new process ID

Tabele stranica smestene u memoriji?



- Translacija:
  - PPN = Mem[PT Base + VPN]
  - PA = PPN + ofset
- Sve veze predstavljaju fizicke adrese; nema translacije iz VA u FA
- Kod zamene konteksta
  - PT Base Reg := Kernel PT Base + new process ID

Pristup jednoj reci podataka ili instrukcija zahteva dva pristupa DRAM-u!

Implementacija stranica

- Kako smanjiti suvisan pristup memoriji?
  - Dobar dizajn VM mora biti brz i efikasan sa stanovista memorije
- Sta ako stranice ne mogu da stanu u DRAM?

Implementacija stranica

- Kako smanjiti suvisan pristup memoriji?
  - Dobar dizajn VM mora biti brz i efikasan sa stanovista memorije
- Sta ako stranice ne mogu da stanu u DRAM?
  - Ako tabele stranica procesa ne mogu da stanu u DRAM?

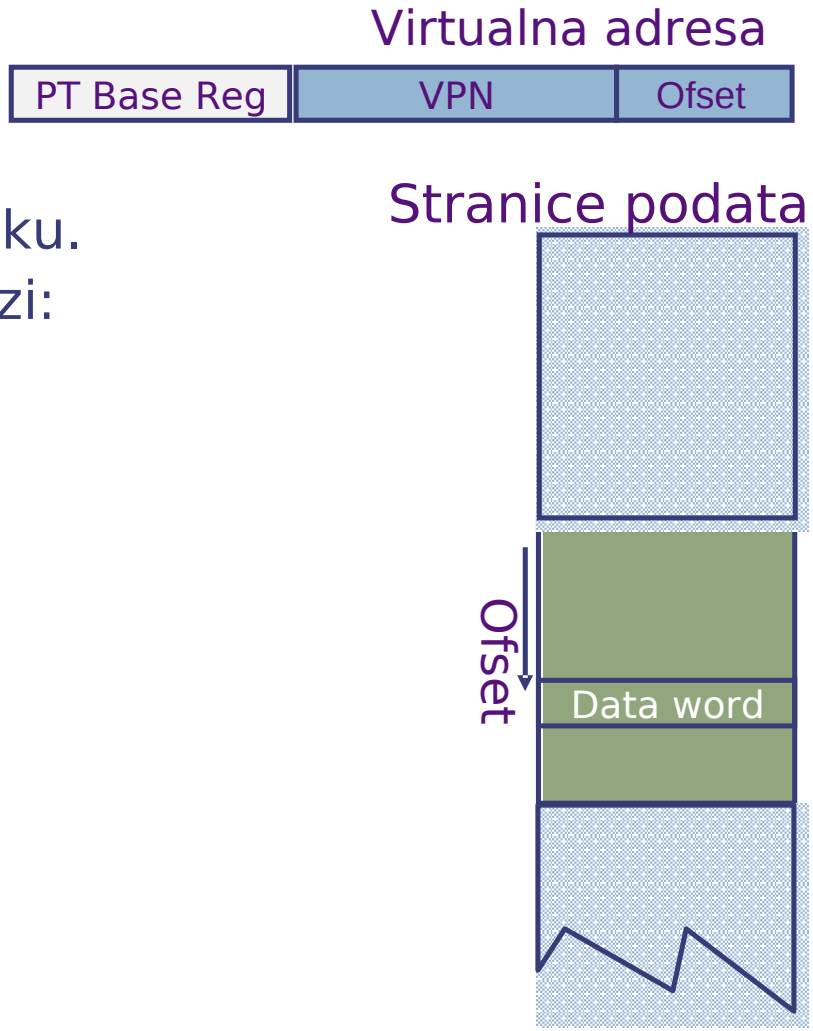
Implementacija stranica

- Kako smanjiti suvisan pristup memoriji?
  - Dobar dizajn VM mora biti brz i efikasan sa stanovista memorije
- Sta ako stranice ne mogu da stanu u DRAM?
  - Ako tabele stranica procesa ne mogu da stanu u DRAM?
  - Ako tabela stranica kernela ne moze da stane u DRAM?

Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

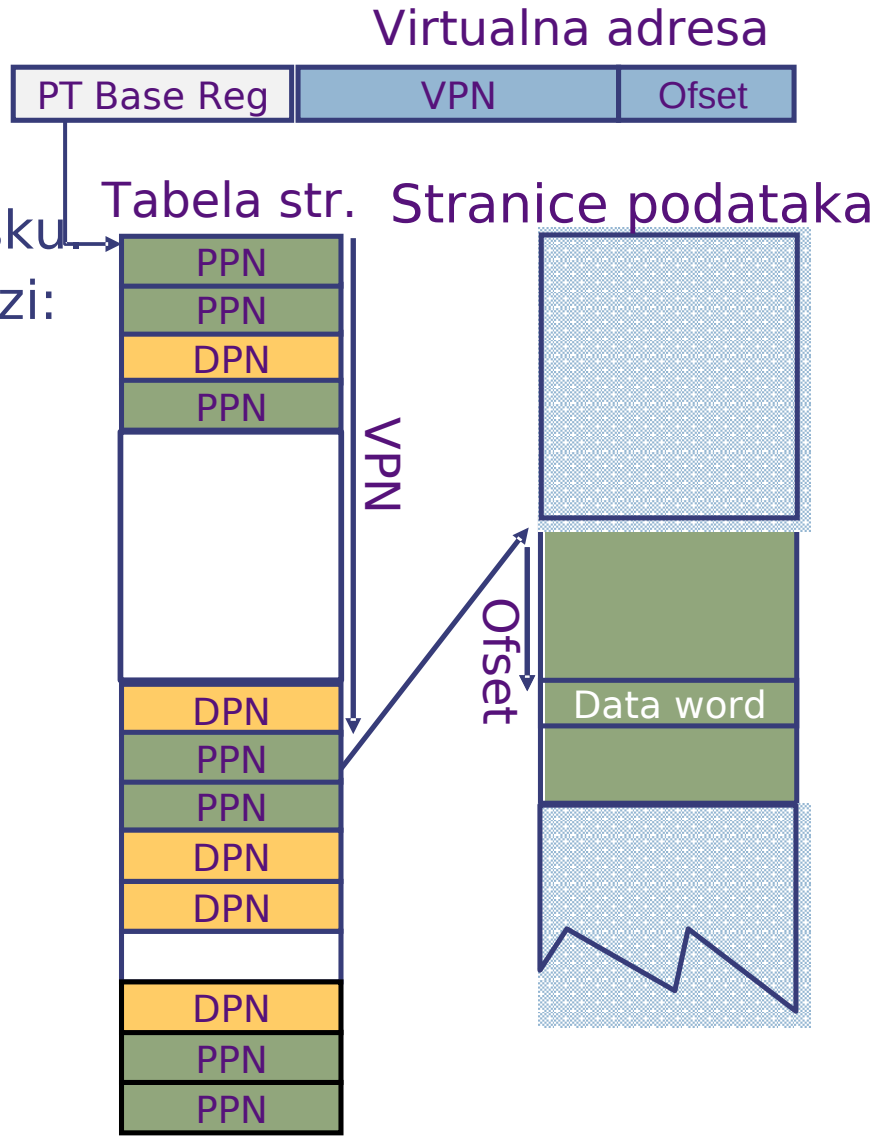
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

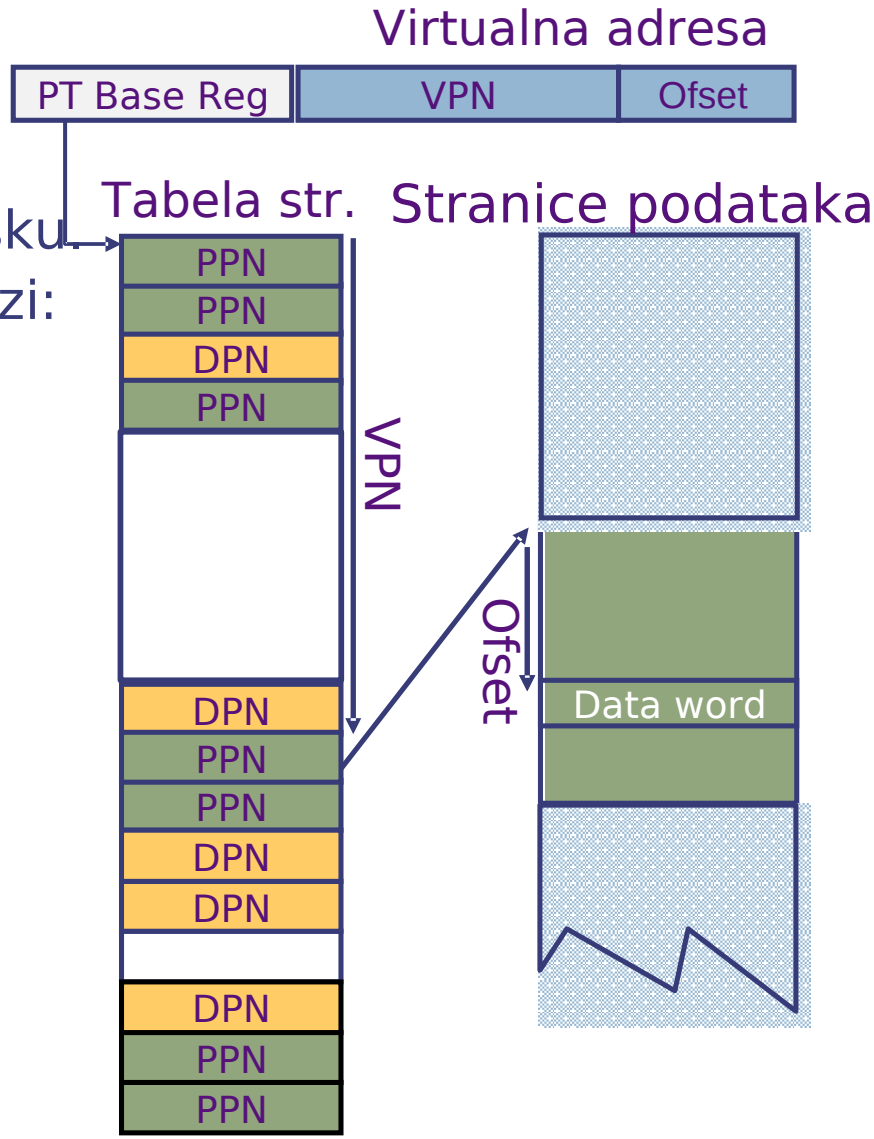
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

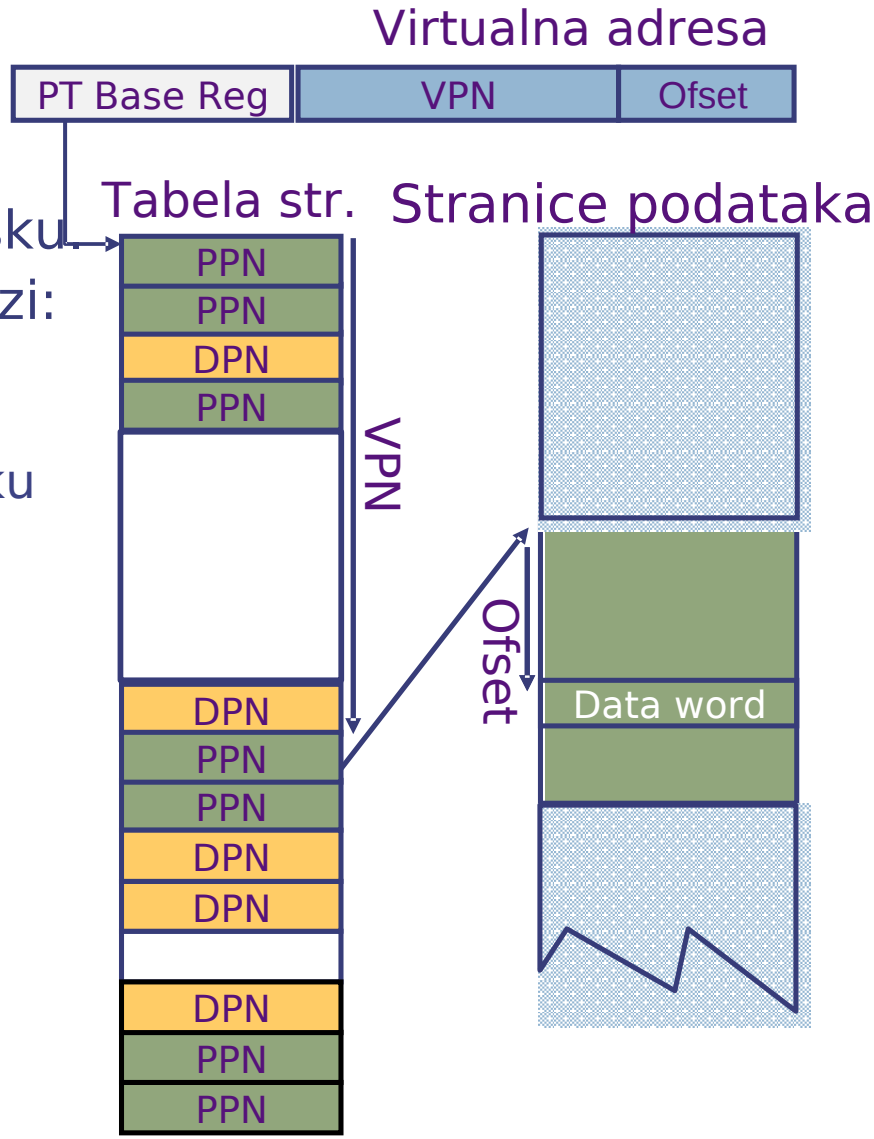
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

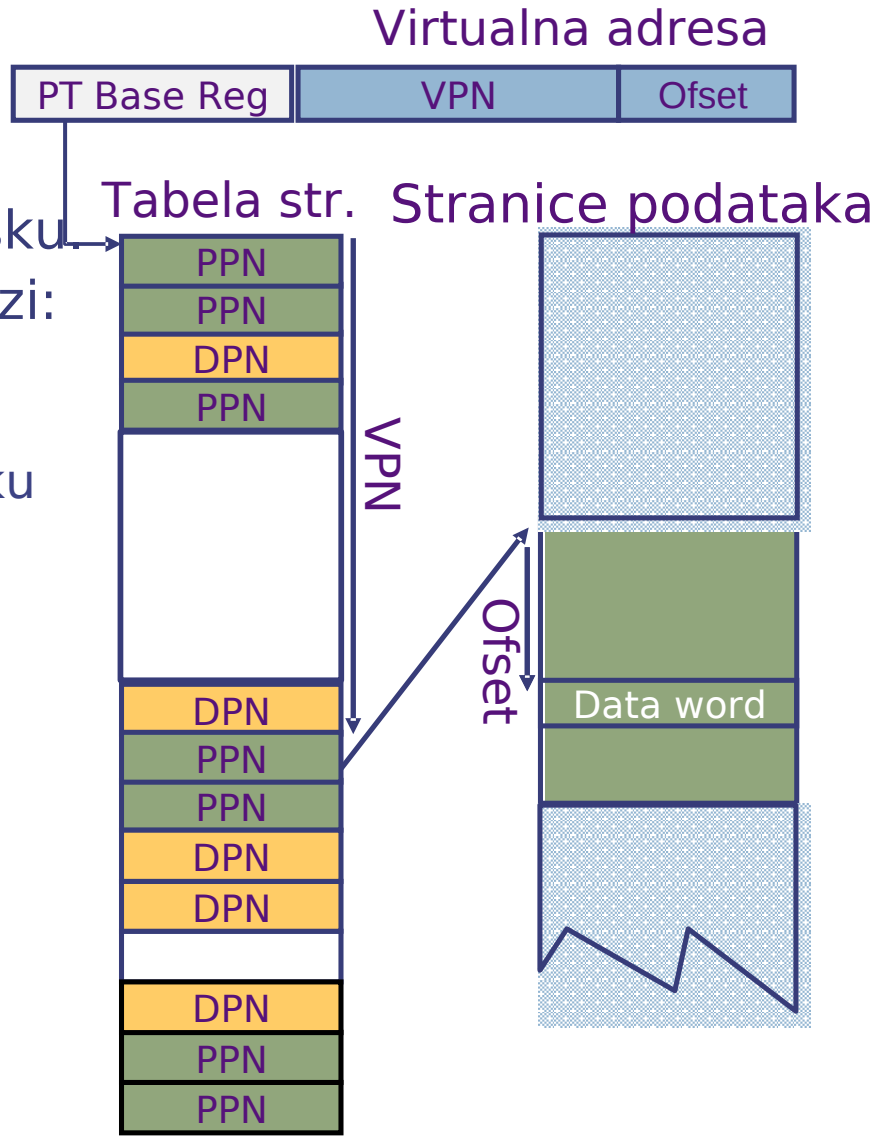
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.
  - PPN (broj fizicke stranice) za svaku stranicu prisutnu u memoriji



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

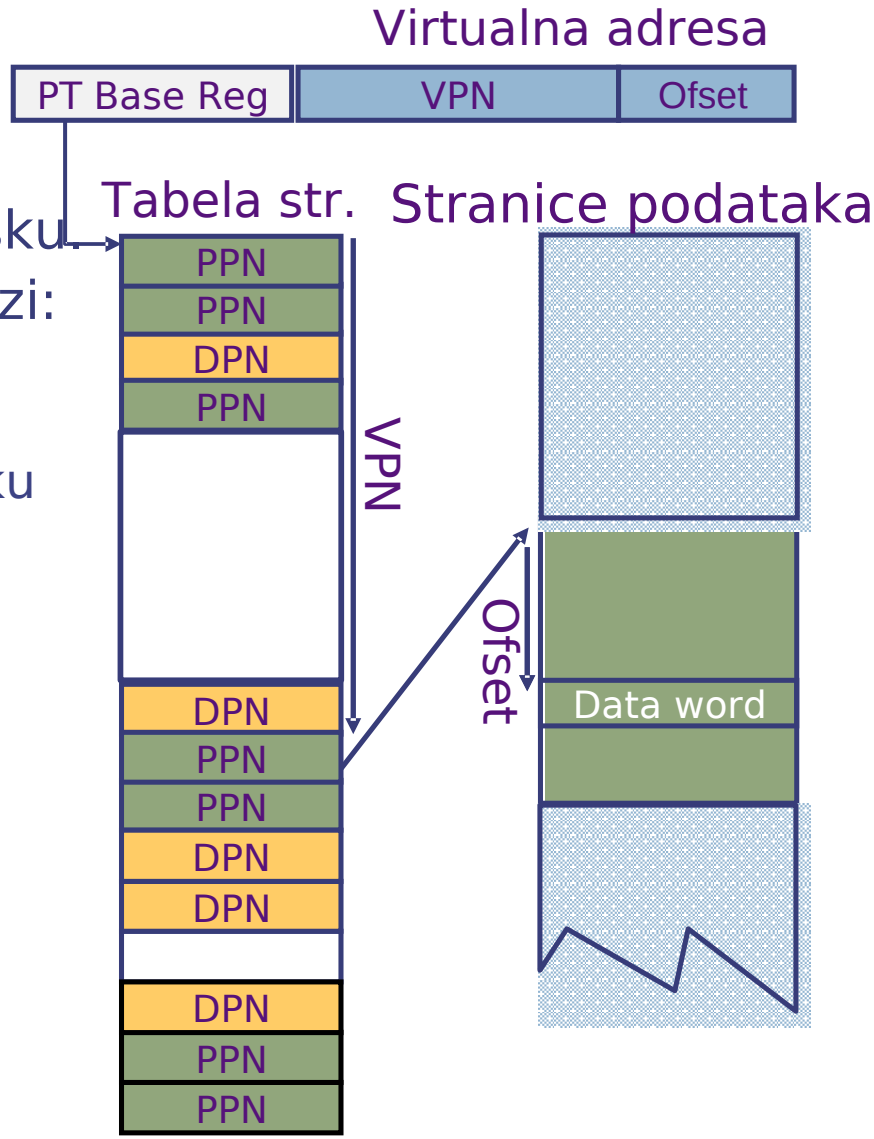
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.
  - PPN (broj fizicke stranice) za svaku stranicu prisutnu u memoriji
  - DPN (broj stranice na disku) za stranice koje se nalaze na disku



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

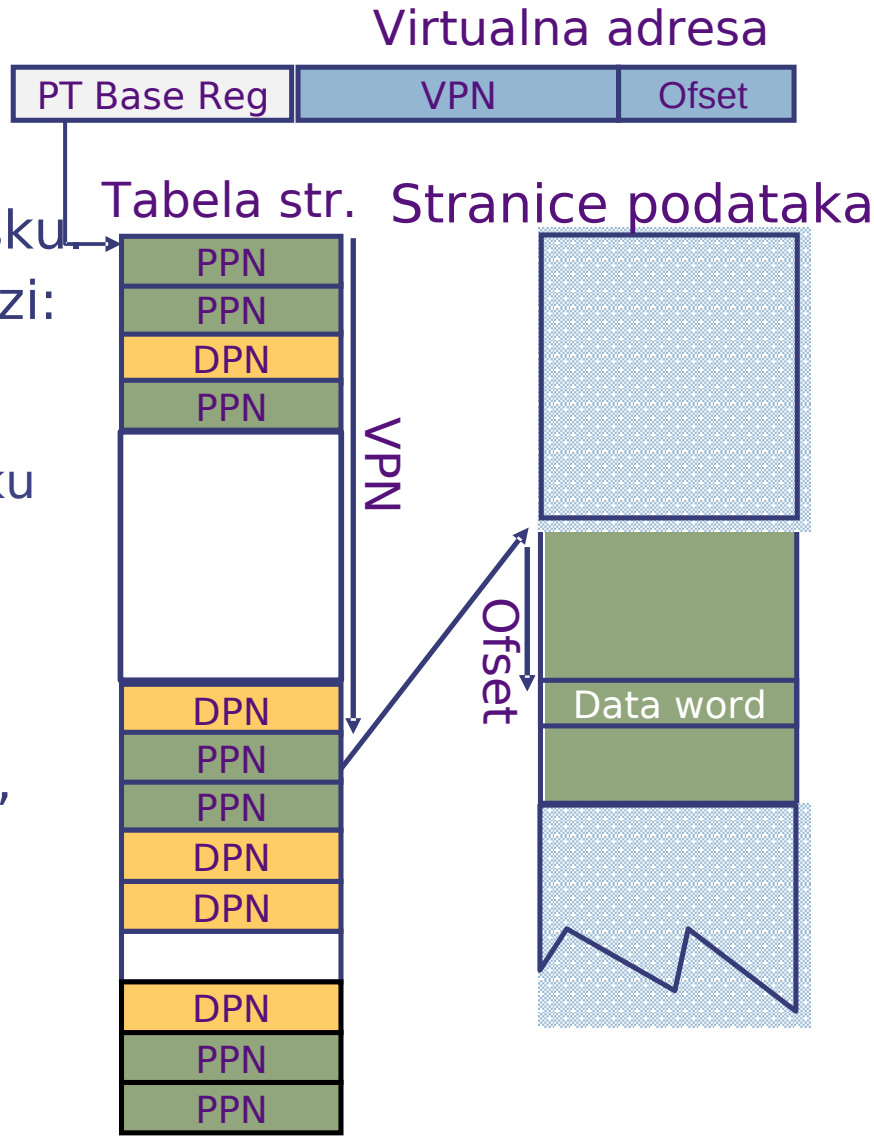
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.
  - PPN (broj fizicke stranice) za svaku stranicu prisutnu u memoriji
  - DPN (broj stranice na disku) za stranice koje se nalaze na disku
  - Bite za zastitu i koriscenje



Stranice na zahtev (Demand Paging)

Koristi osnovnu memoriju kao kes za disk

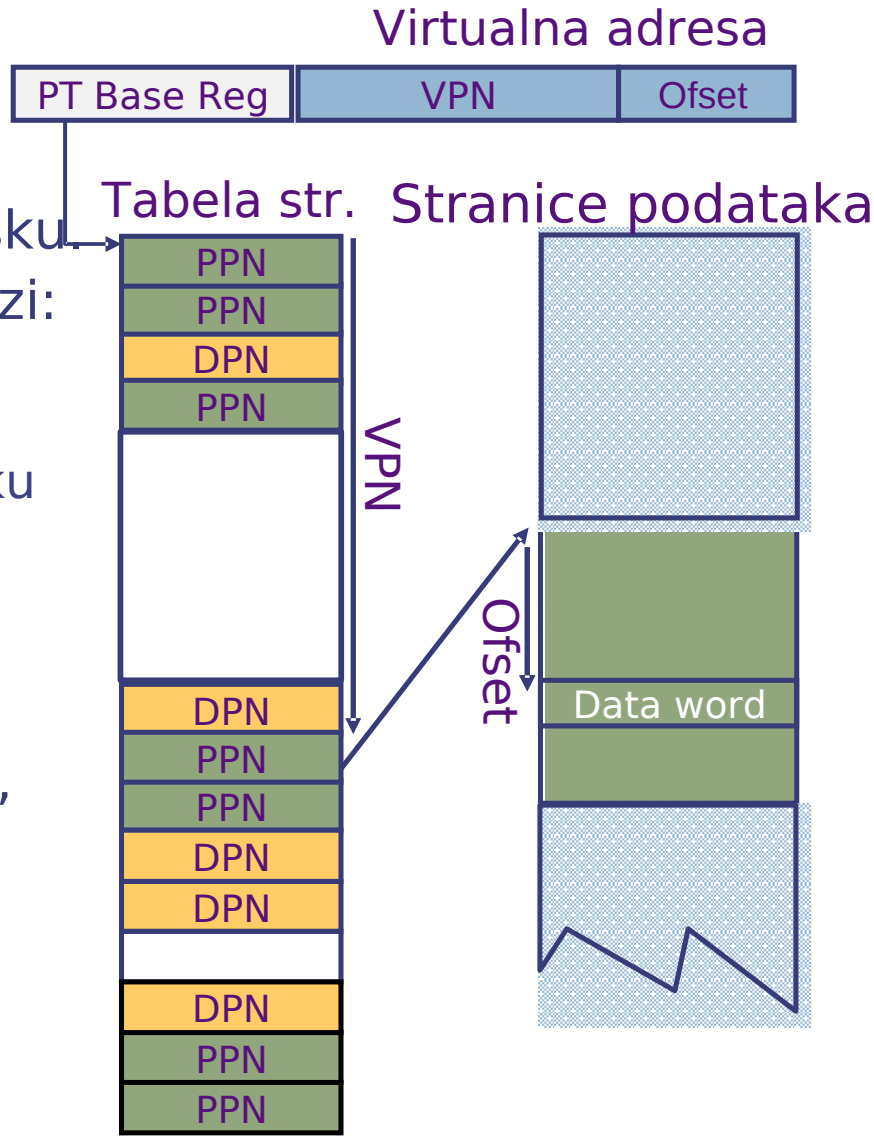
- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.
  - PPN (broj fizicke stranice) za svaku stranicu prisutnu u memoriji
  - DPN (broj stranice na disku) za stranice koje se nalaze na disku
  - Bite za zastitu i koriscenje
- Cak i ako sve stranice mogu stati, stranice na zahtev omogucavaju prihvat samo onoga sto je nuzno



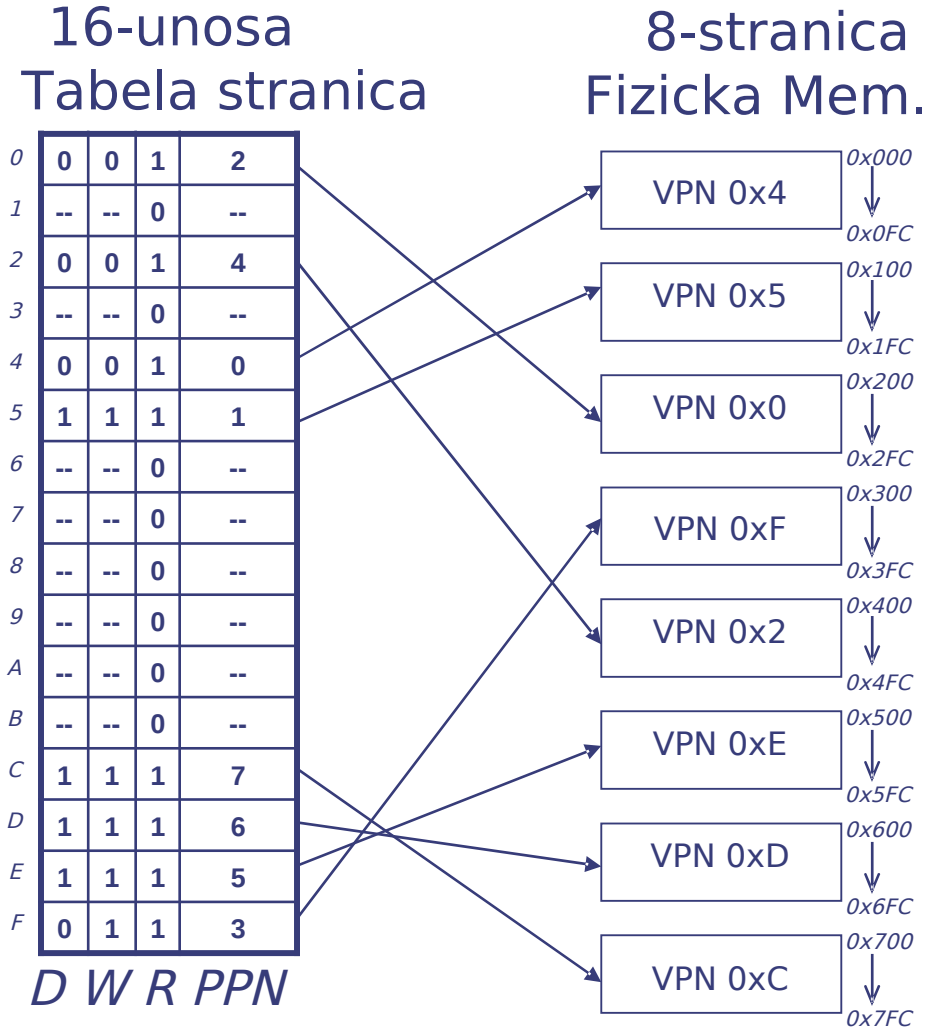
Stranice na zahtev (Demand Paging)

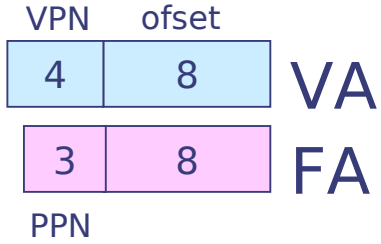
Koristi osnovnu memoriju kao kes za disk

- Sve stranice procesa ne mogu stati u osnovnu memoriju. Usled toga, DRAM je podrzan koriscenjem *swap prostora* na disku.
- Unos u tabelu stranica (PTE) sadrzi:
  - Resident bit kao indikacija da stranica postoji u osnovnoj mem.
  - PPN (broj fizicke stranice) za svaku stranicu prisutnu u memoriji
  - DPN (broj stranice na disku) za stranice koje se nalaze na disku
  - Bite za zastitu i koriscenje
- Cak i ako sve stranice mogu stati, stranice na zahtev omogucavaju prihvat samo onoga sto je nuzno
  - Kada proces startuje, sav kod i podaci su na disku; prihvataj stranice kako im se pristupa



Primer: VA <=> FA Translacija





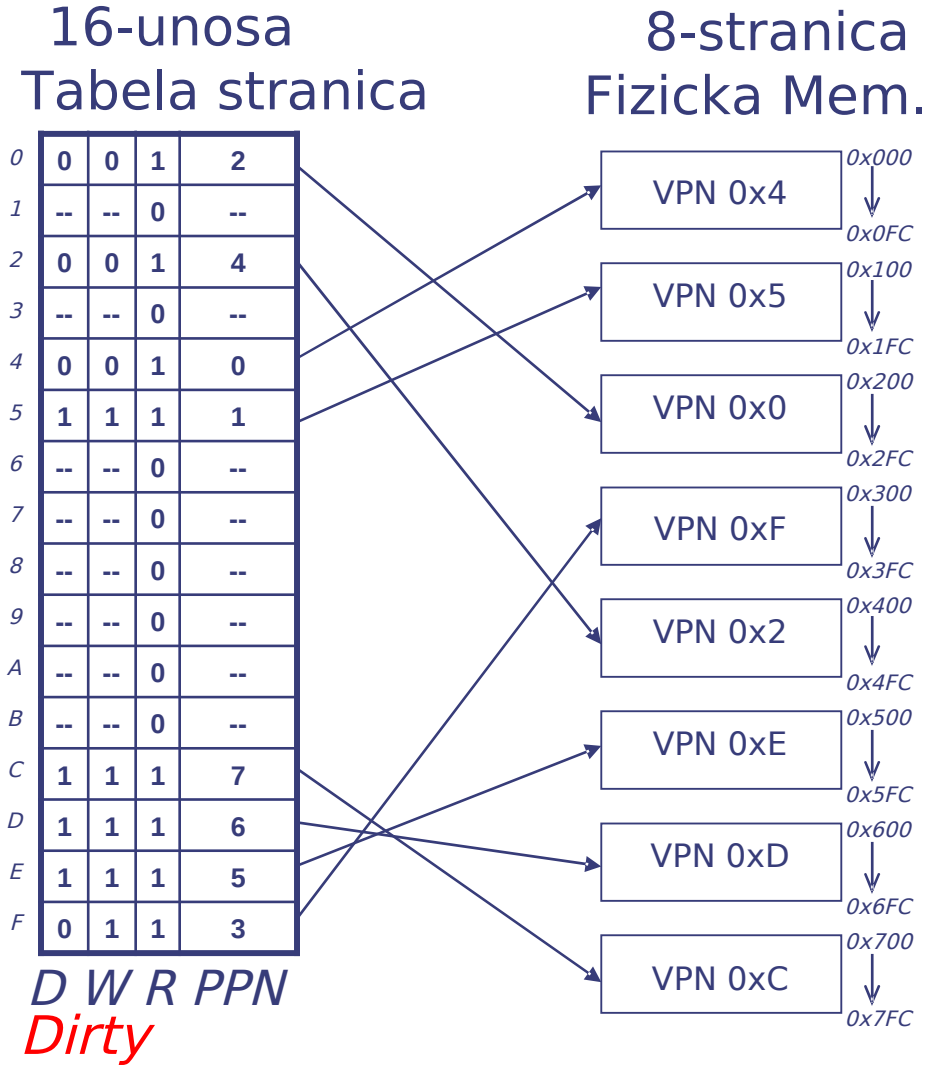
Setup:

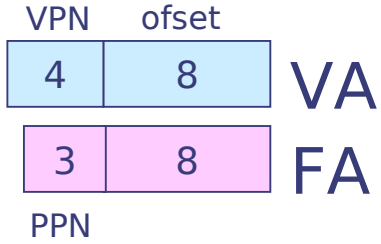
- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

Primer: VA <=> FA Translacija





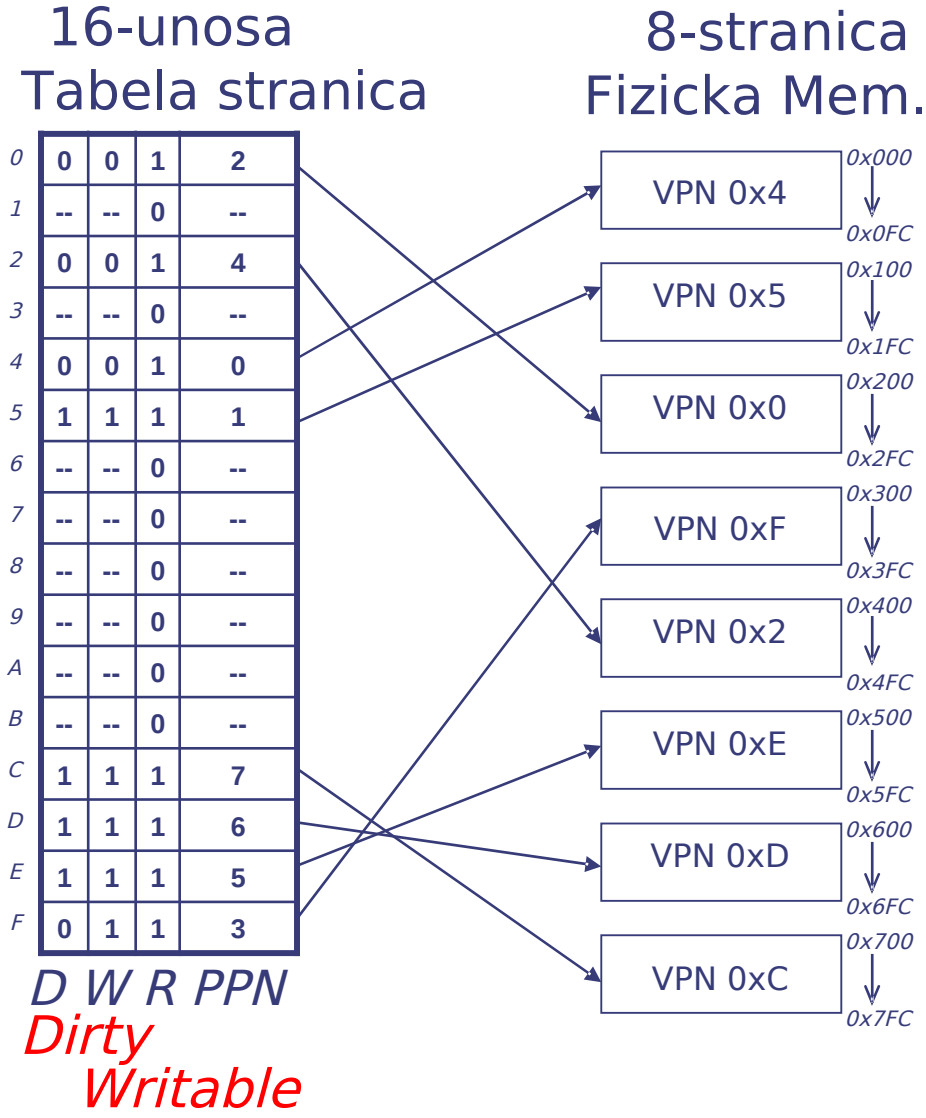
Setup:

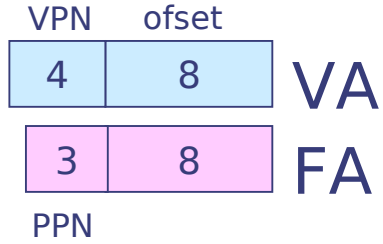
- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

Primer: VA <=> FA Translacija





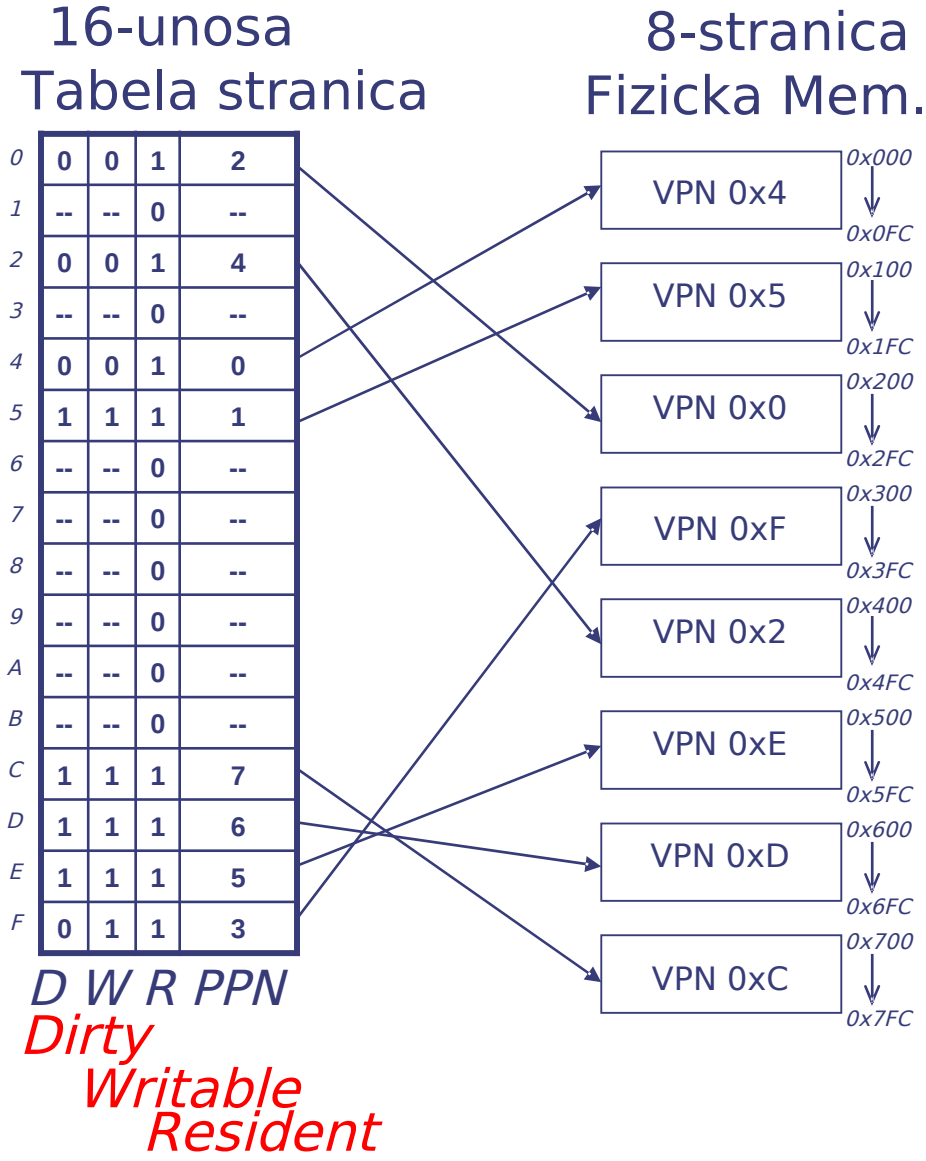
Setup:

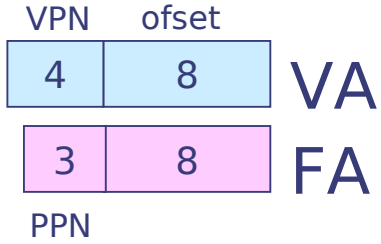
- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

Primer: VA <=> FA Translacija





Setup:

- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

Primer: VA <=> FA Translacija

16-unosa
Tabela stranica

	D	W	R	PPN
0	0	0	1	2
1	--	--	0	--
2	0	0	1	4
3	--	--	0	--
4	0	0	1	0
5	1	1	1	1
6	--	--	0	--
7	--	--	0	--
8	--	--	0	--
9	--	--	0	--
A	--	--	0	--
B	--	--	0	--
C	1	1	1	7
D	1	1	1	6
E	1	1	1	5
F	0	1	1	3

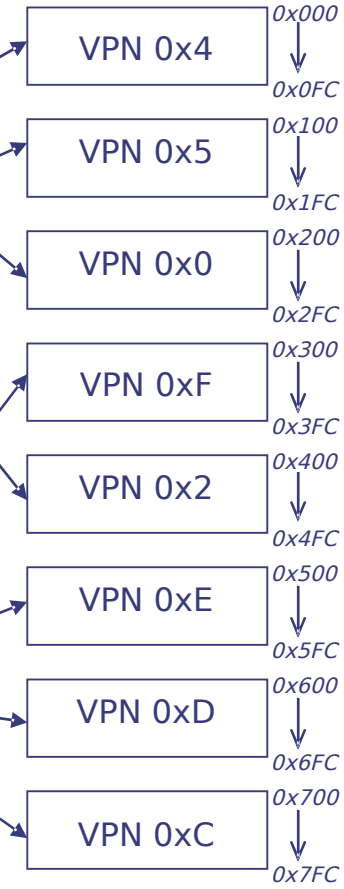
D W R PPN

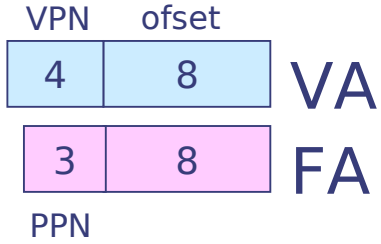
Dirty

Writable

Resident

8-stranica
Fizicka Mem.





Setup:

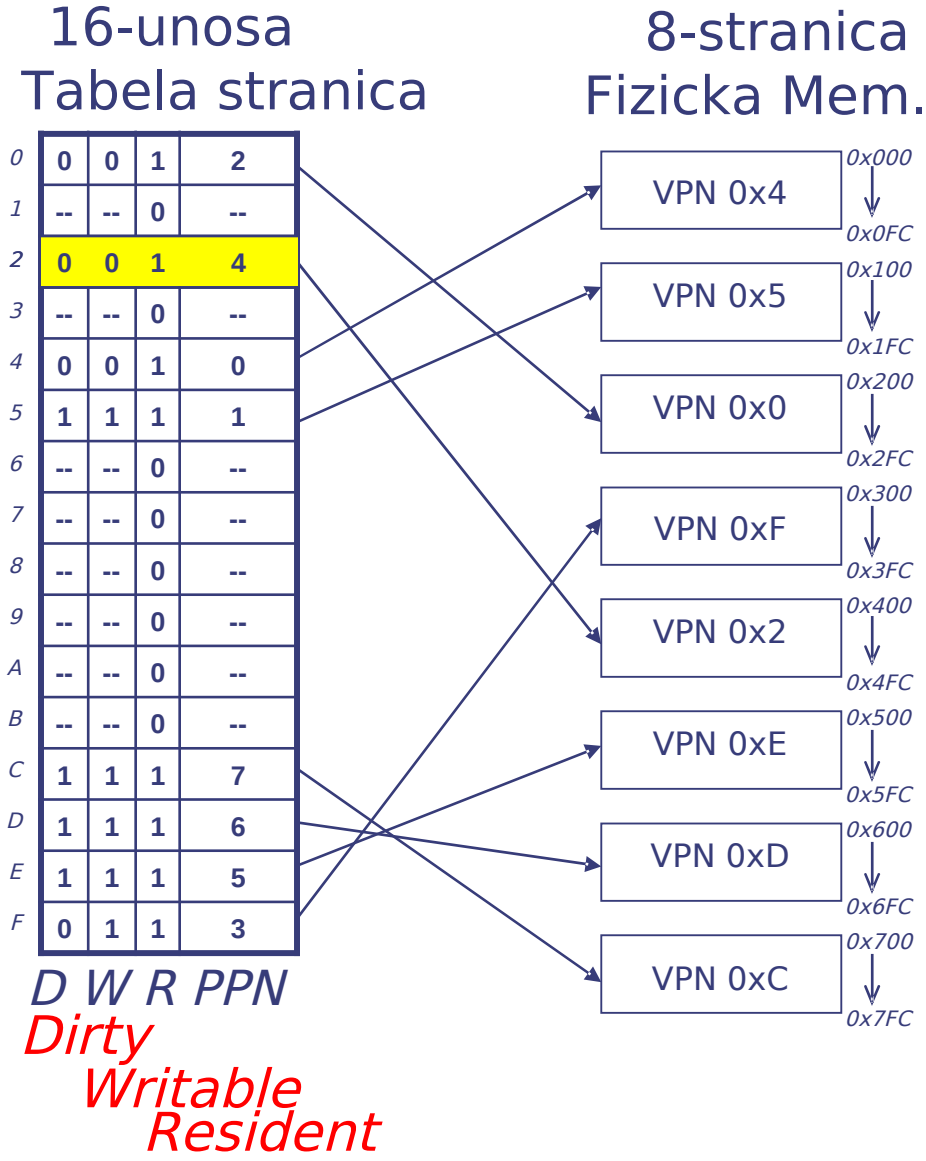
- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

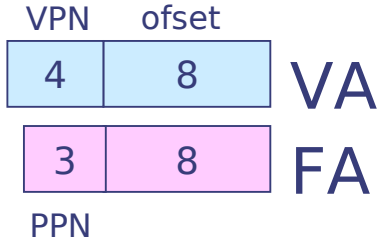
```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

VPN = 0x2

Primer: VA <=> FA Translacija





Setup:

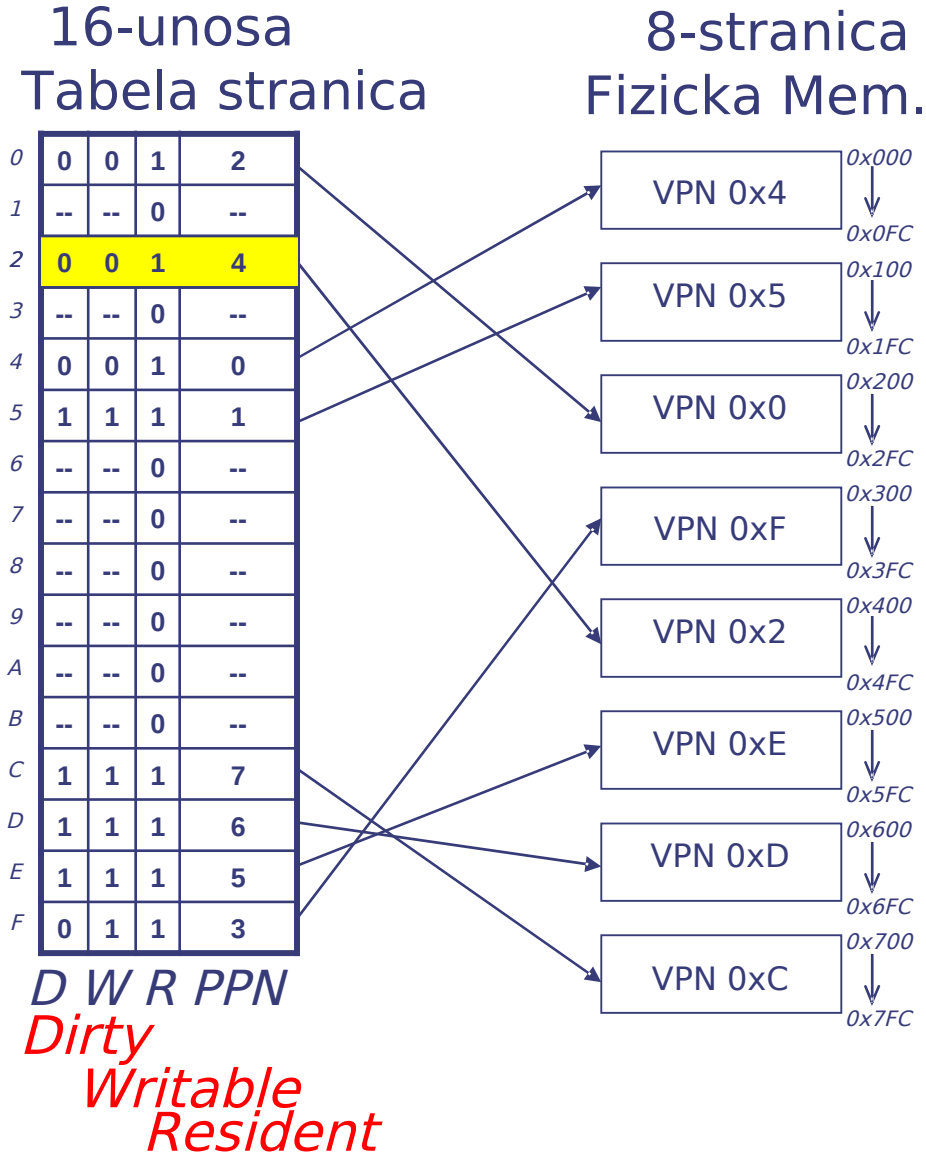
- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

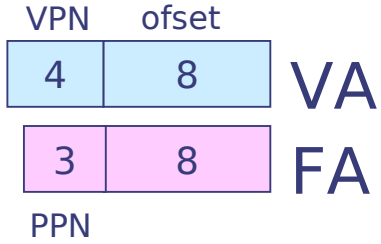
```
lw 0x2C8(x0)
```

VA = 0x2C8, PA = _____

VPN = 0x2

Primer: VA <=> FA Translacija





Setup:

- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

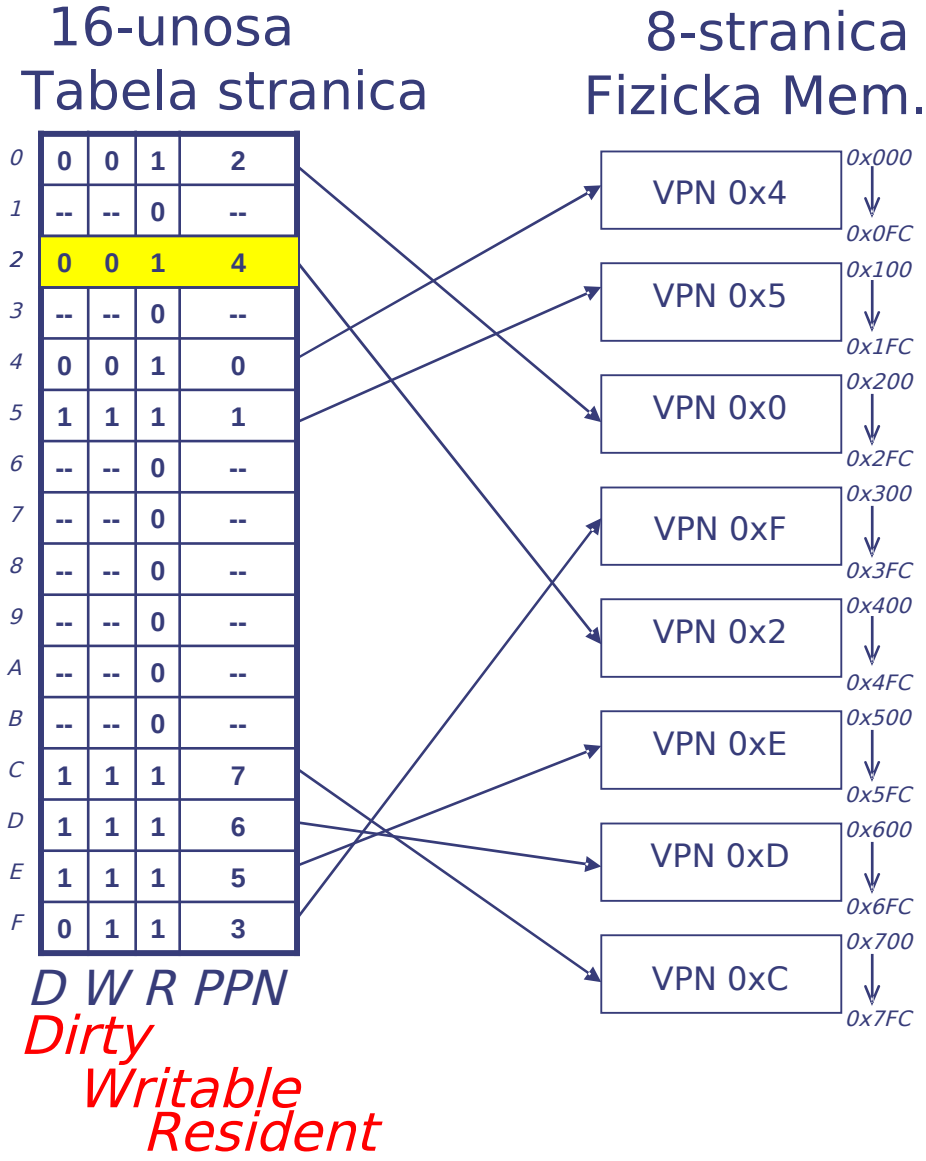
`lw 0x2C8(x0)`

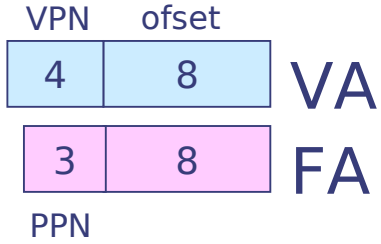
VA = 0x2C8, PA = _____

VPN = 0x2

→ PPN = 0x4

Primer: VA <=> FA Translacija





Setup:

- 256 bytes/page (2^8)
- 16 virtu. stranica(2^4)
- 8 fizickih stranica (2^3)
- 12-bit VA (4 vpn, 8 ofset)
- 11-bit FA (3 ppn, 8 ofset)

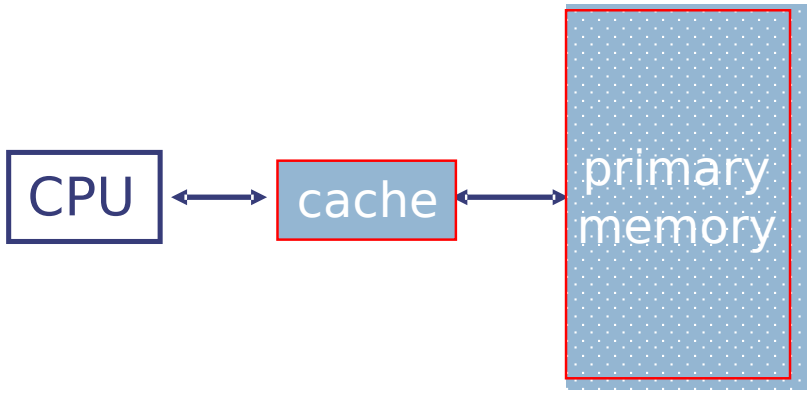
```
lw 0x2C8(x0)
```

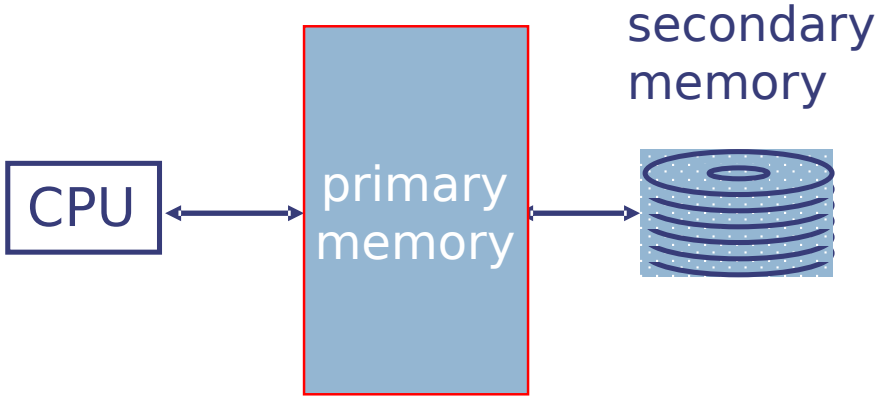
VA = 0x2C8, PA = 0x4C8

VPN = 0x2

→ PPN = 0x4

Kesiranje i Stranice na zahtev





Caching

- cache entry
- cache blok (~32 bytes)
- cache MR (1% to 20%)
- cache hit (~1 cycle)
- cache miss (~100 cycles)
- promasaj se obradjuje u *hardveru*

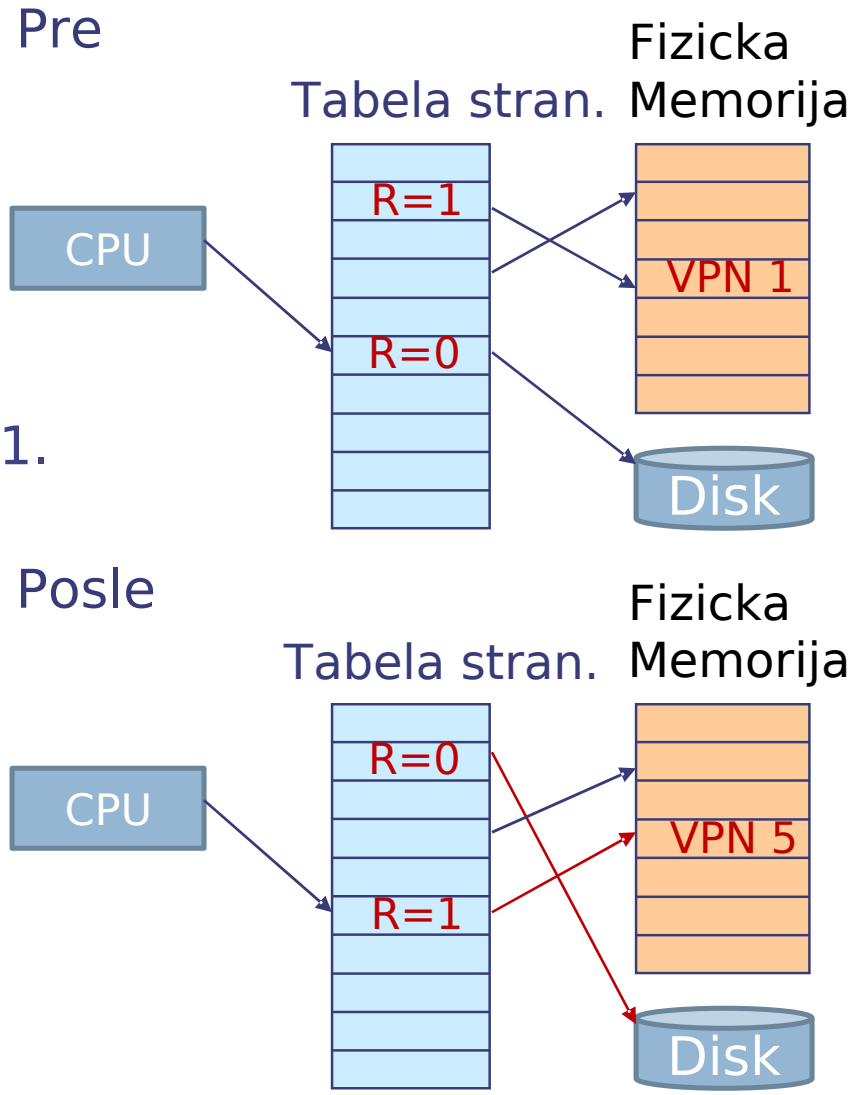
Demand paging

- page frame
- page (~4K bytes)
- page MR (<0.001%)
- page hit (~100 cycles)
- page miss (~5M cycles)
- promasaj se obradjuje uglavnom u softveru

Nedostupne stranice (Page Faults)

Pristup stranici koja nema validnu translacijui dovodi do izuzetka usled nedostupne stranice
OS rutina za obradu izuzetka je pozvana:

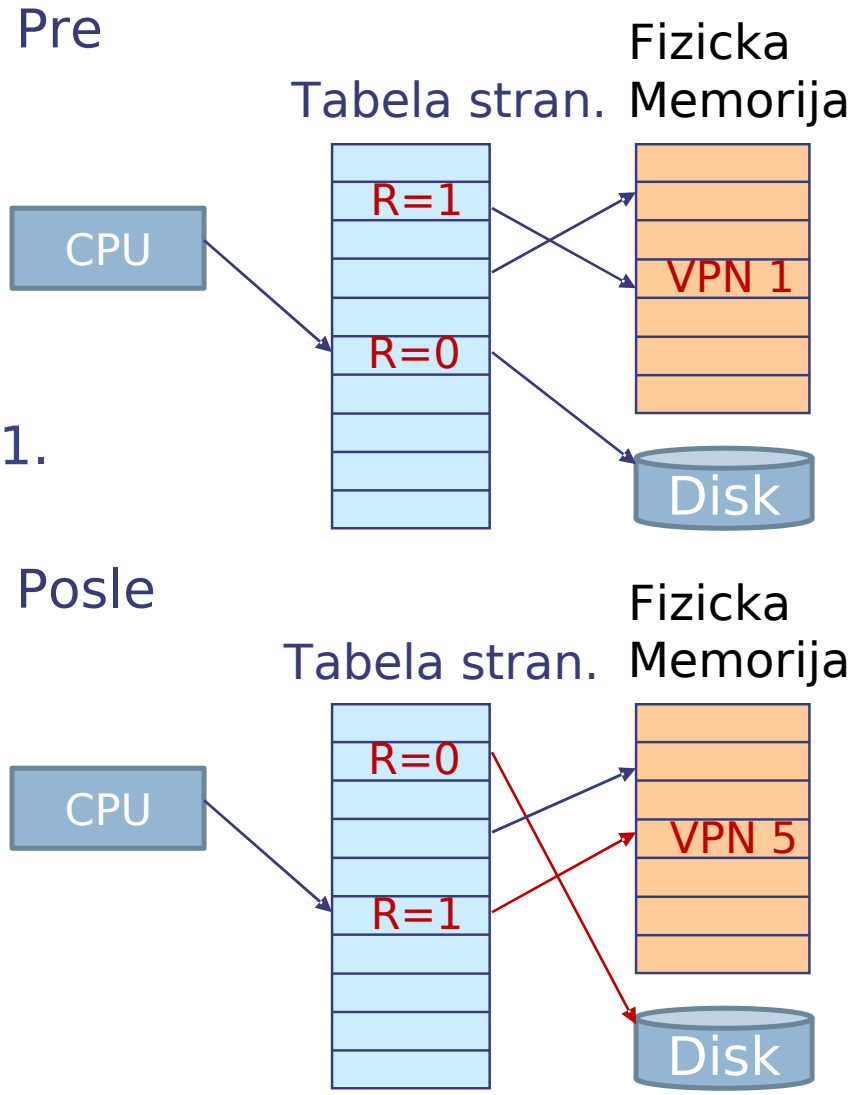
– Odaberi stranicu za zamenu, upisi je u memoriju ako je dirty=1. Oznaci je sa resident=0



Nedostupne stranice (Page Faults)

Pristup stranici koja nema validnu translacijui dovodi do izuzetka usled nedostupne stranice
OS rutina za obradu izuzetka je pozvana:

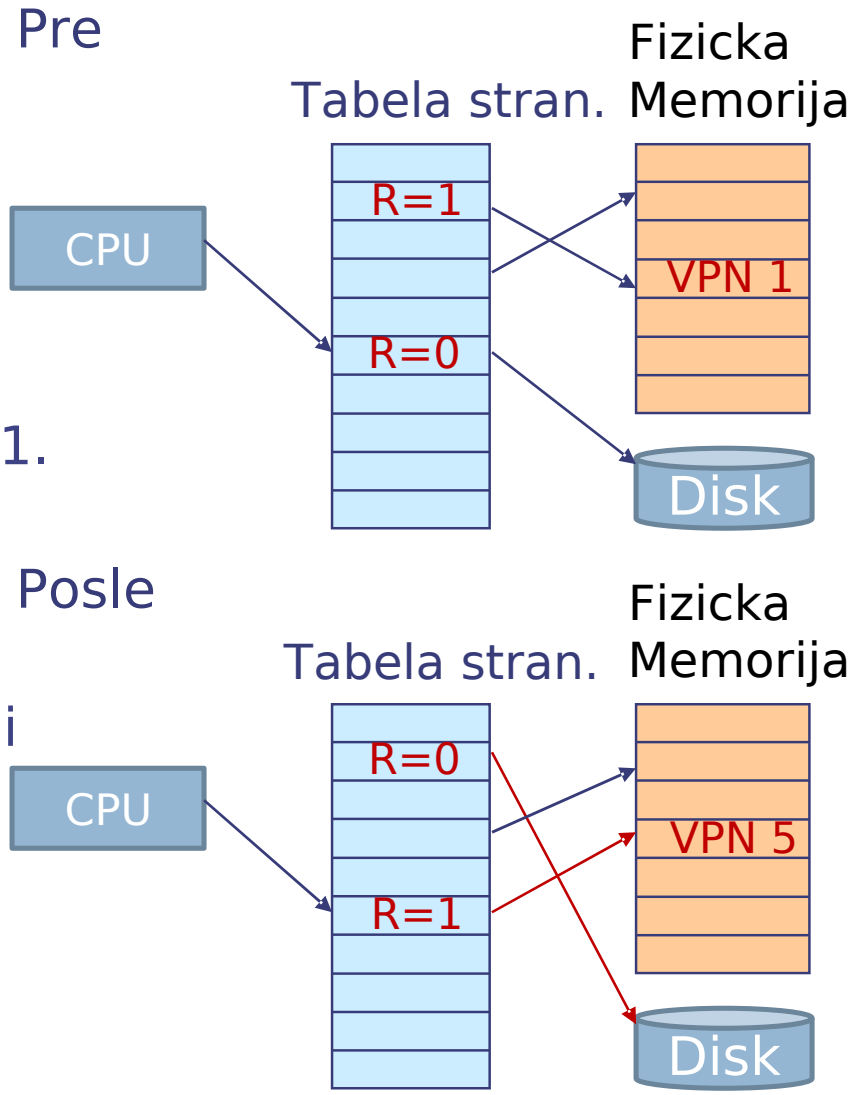
- Odaberi stranicu za zamenu, upisi je u memoriju ako je dirty=1. Oznaci je sa resident=0
- Ucitaj stranicu sa diska na mesto dostupne fizicke stranice



Nedostupne stranice (Page Faults)

Pristup stranici koja nema validnu translacijui dovodi do izuzetka usled nedostupne stranice
OS rutina za obradu izuzetka je pozvana:

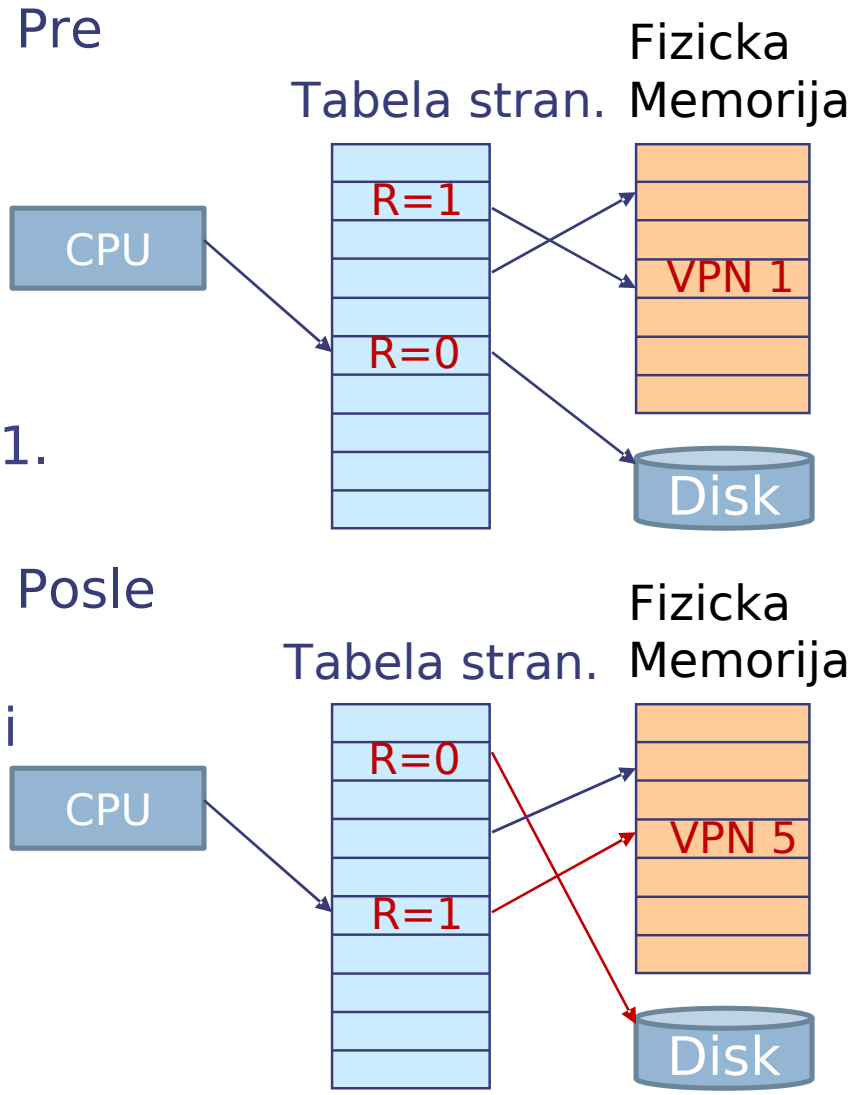
- Odaberi stranicu za zamenu, upisi je u memoriju ako je dirty=1. Oznaci je sa resident=0
- Ucitaj stranicu sa diska na mesto dostupne fizicke stranice
- Azuriraj tabelu stranica da se vidi nova prisutna stranica (resident)



Nedostupne stranice (Page Faults)

Pristup stranici koja nema validnu translacijui dovodi do izuzetka usled nedostupne stranice
OS rutina za obradu izuzetka je pozvana:

- Odaberi stranicu za zamenu, upisi je u memoriju ako je dirty=1. Oznaci je sa resident=0
- Ucitaj stranicu sa diska na mesto dostupne fizicke stranice
- Azuriraj tabelu stranica da se vidi nova prisutna stranica (resident)
- Vrati kontrolu programu koji ponovo pristupa memor. lokaciji

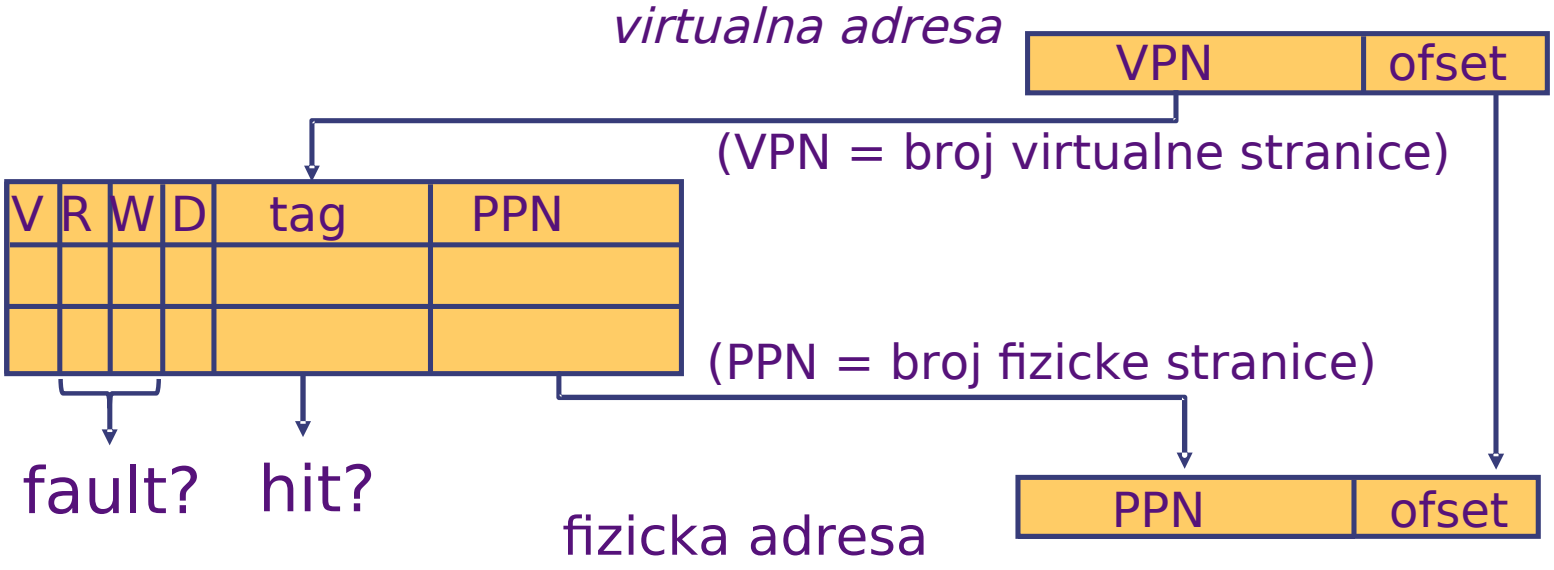


Bafer za kesiranje translacija
Translation Lookaside Buffer (TLB)

Problem: Translacija adresa je veoma skupa!
Svaki pristup memoriji zahteva pristup tabeli stranica

Resenje: *Kesirati translacije u TLB*

TLB hit ⇒ *Translacija u jednom taktu*
TLB miss ⇒ *Prolaz kroz tabelu stranica*



TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni

TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)

TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)
- Zamena procesa je skupa jer TLB moraju biti zamenjeni

TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)
- Zamena procesa je skupa jer TLB moraju biti zamenjeni
  - Alternativno, dodati ID procesa u TLB kako bi se izbeglo ovo

TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)
- Zamena procesa je skupa jer TLB moraju biti zamenjeni
  - Alternativno, dodati ID procesa u TLB kako bi se izbeglo ovo
- TLB promasaj: Prodji (“Walk”) kroz tabelu stranica. Ako je stranica u memoriji, ucitaj VPN<=>PPN translaciju u TLB. U suprotnom, oznaci nedostupnu stranicu (Page Fault)

TLB Dizajn

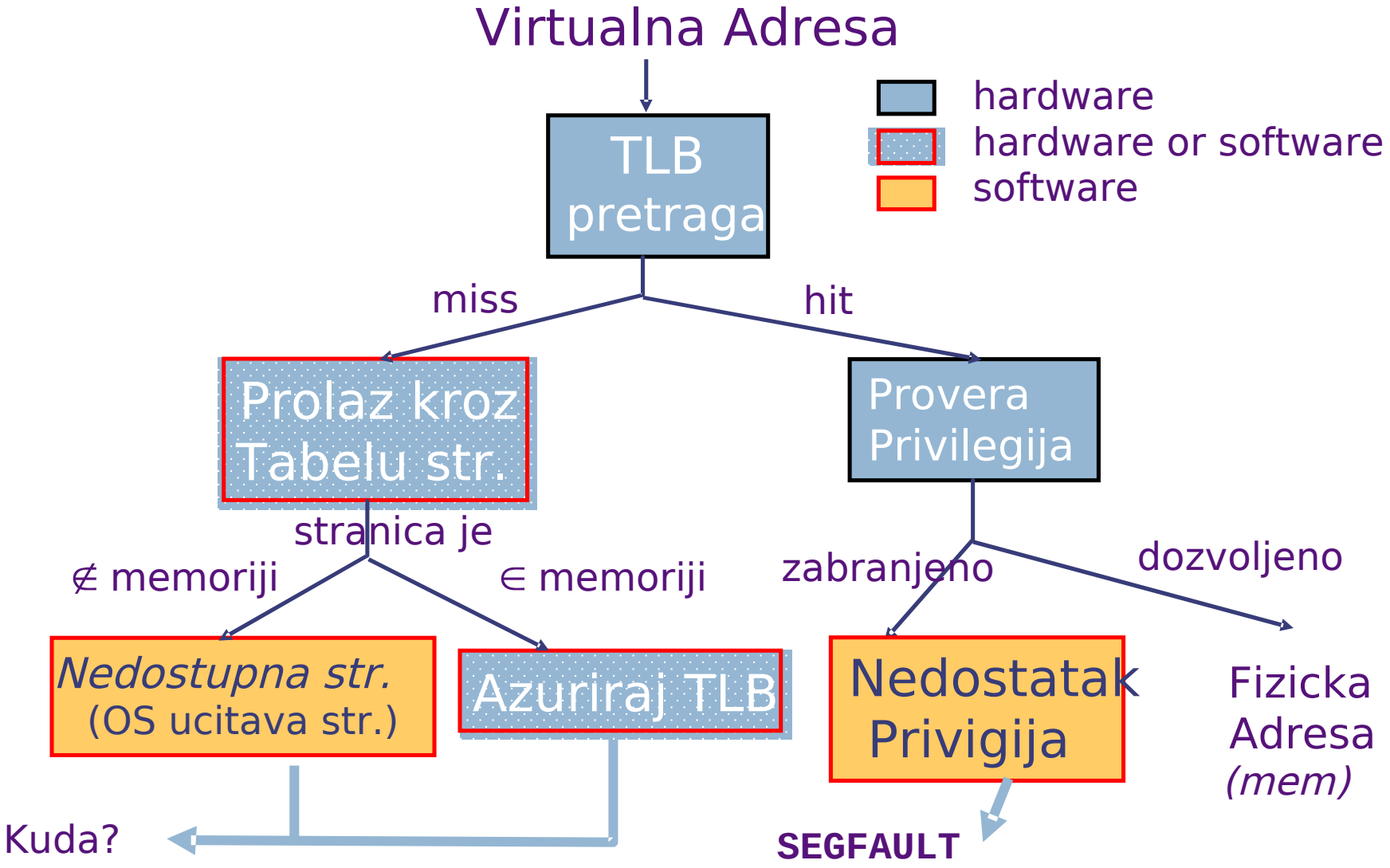
- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)
- Zamena procesa je skupa jer TLB moraju biti zamenjeni
  - Alternativno, dodati ID procesa u TLB kako bi se izbeglo ovo
- TLB promasaj: Prodji (“Walk”) kroz tabelu stranica. Ako je stranica u memoriji, ucitaj VPN<=>PPN translaciju u TLB. U suprotnom, oznaci nedostupnu stranicu (Page Fault)
  - Razresavanje nedostupne stranice se uvek vrsi u softveru

TLB Dizajn

- Tipicno 32-128 linija, 4 ili 8-grupa set-asocijativni
  - Moderni procesori koriste hijerarhiju TLB-ova (npr, 128-linija L1 TLB + 2K-linija L2 TLB)
- Zamena procesa je skupa jer TLB moraju biti zamenjeni
  - Alternativno, dodati ID procesa u TLB kako bi se izbeglo ovo
- TLB promasaj: Prodji (“Walk”) kroz tabelu stranica. Ako je stranica u memoriji, ucitaj VPN<=>PPN translaciju u TLB. U suprotnom, oznaci nedostupnu stranicu (Page Fault)
  - Razresavanje nedostupne stranice se uvek vrsi u softveru
  - Ali prolaz kroz tabelu stranica se gotovo uvek razresava u HW koriscenjem MMU *(memory management unit)*

Translacija adresa:

Sumiranje



Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag Data

VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom?
2. Koliko linija (ulaza) postoji u tabeli stranica?
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit)
4. Koliko stranica zahteva tabela stranica?
5. Koji deo virtualne memorije moze da bude “resident”?
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag Data

VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica?
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit)
4. Koliko stranica zahteva tabela stranica?
5. Koji deo virtualne memorije moze da bude “resident”?
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag Data

VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit)
4. Koliko stranica zahteva tabela stranica?
5. Koji deo virtualne memorije moze da bude “resident”?
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag Data

VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica?
5. Koji deo virtualne memorije moze da bude “resident”?
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag Data

VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica? 2^{23} bytes = 2^{13} stranica
5. Koji deo virtualne memorije moze da bude “resident”?
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag	Data		
VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica? 2^{23} bytes = 2^{13} stranica
5. Koji deo virtualne memorije moze da bude “resident”? $1/2^{8}$
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije?
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag	Data		
VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica? 2^{23} bytes = 2^{13} stranica
5. Koji deo virtualne memorije moze da bude “resident”? $1/2^{8}$
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije? [VPN=6] 0x804
7. Isto za 0x1080?
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag	Data		
VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica? 2^{23} bytes = 2^{13} stranica
5. Koji deo virtualne memorije moze da bude “resident”? $1/2^{8}$
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije? [VPN=6] 0x804
7. Isto za 0x1080? [VPN=4] 0x1480
8. Isto za 0x0FC?

Primer: TLB i Tabela stranica

Pretpostavka

- Virtualna memorija 2^{32}bytes
- Fizicka memorija 2^{24} bytes
- Velicina Str.2^{10} (1 K) bytes
- 4-linije FA TLB

TLB

Tag	Data		
VPN	R	D	PPN
0	0	0	3
6	1	1	2
1	1	1	9
3	0	0	5

Tabela Stranica

VPN	R	D	PPN
0	0	0	7
1	1	1	9
2	1	0	0
3	0	0	5
4	1	0	5
5	0	0	3
6	1	1	2
7	1	0	4
8	1	0	1
	…		

1. Koliko stranica moze biti smesteno u fizicku memoriju odjednom? 2^{14}
2. Koliko linija (ulaza) postoji u tabeli stranica? 2^{22}
3. Koliko bita po svakom ulazu tabele stranica? (Podrazumevano svaki ima PPN, resident bit, dirty bit) 16
4. Koliko stranica zahteva tabela stranica? 2^{23} bytes = 2^{13} stranica
5. Koji deo virtualne memorije moze da bude “resident”? $1/2^{8}$
6. Koja je fizicka adresa za virtualnu addresu 0x1804? Koje komponente su ukljucene u proces translacije? [VPN=6] 0x804
7. Isto za 0x1080? [VPN=4] 0x1480
8. Isto za 0x0FC? [VPN=0] nedostupna stranica

Problem: Velicina linearne tabele str.

Sa 32-bitnim adresama, 4 KB stranicama i 4-byte PTE:

- ⇒ 2^{20} PTE, tj. 4 MB tabela stranica *za svaki proces*
- ⇒ Cesto imamo stotine, cak i hiljade procesa aktivnih na masini. Koristiti GB-e memorije samo za tabele stranica?

Problem: Velicina linearne tabele str.

Sa 32-bitnim adresama, 4 KB stranicama i 4-byte PTE:

⇒ 2^{20} PTE, tj. 4 MB tabela stranica *za svaki proces*

⇒ Cesto imamo stotine, cak i hiljade procesa aktivnih na masini. Koristiti GB-e memorije samo za tabele stranica?

Koristiti vece stranice?

- Interna fragmentacija (nije sva memorija stranice koriscena)
- Veca kazna kod nedostupne stranice (vise vremena za ucitav.)

Problem: Velicina linearne tabele str.

Sa 32-bitnim adresama, 4 KB stranicama i 4-byte PTE:

⇒ 2^{20} PTE, tj. 4 MB tabela stranica *za svaki proces*

⇒ Cesto imamo stotine, cak i hiljade procesa aktivnih na masini. Koristiti GB-e memorije samo za tabele stranica?

Koristiti vece stranice?

- Interna fragmentacija (nije sva memorija stranice koriscena)
- Veca kazna kod nedostupne stranice (vise vremena za ucitav.)

Sta sa 64-bitnim virtualnim adresnim prostorom?

- Cak i stanice od 1MB zahtevaju 2^{44} 8-byte PTE-ova (35 TB!)

Problem: Velicina linearne tabele str.

Sa 32-bitnim adresama, 4 KB stranicama i 4-byte PTE:

⇒ 2^{20} PTE, tj. 4 MB tabela stranica *za svaki proces*

⇒ Cesto imamo stotine, cak i hiljade procesa aktivnih na masini. Koristiti GB-e memorije samo za tabele stranica?

Koristiti vece stranice?

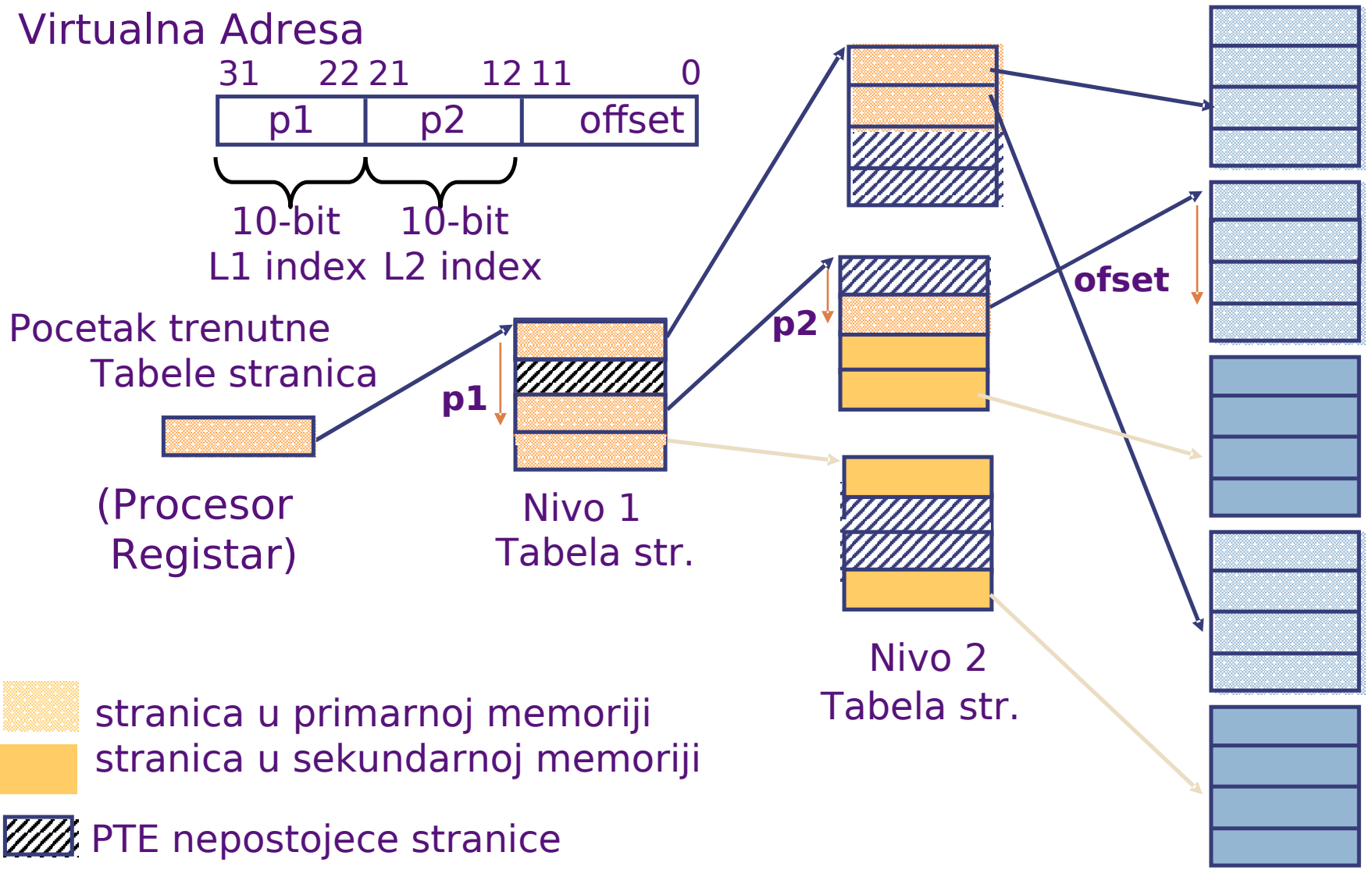
- Interna fragmentacija (nije sva memorija stranice koriscena)
- Veca kazna kod nedostupne stranice (vise vremena za ucitav.)

Sta sa 64-bitnim virtualnim adresnim prostorom?

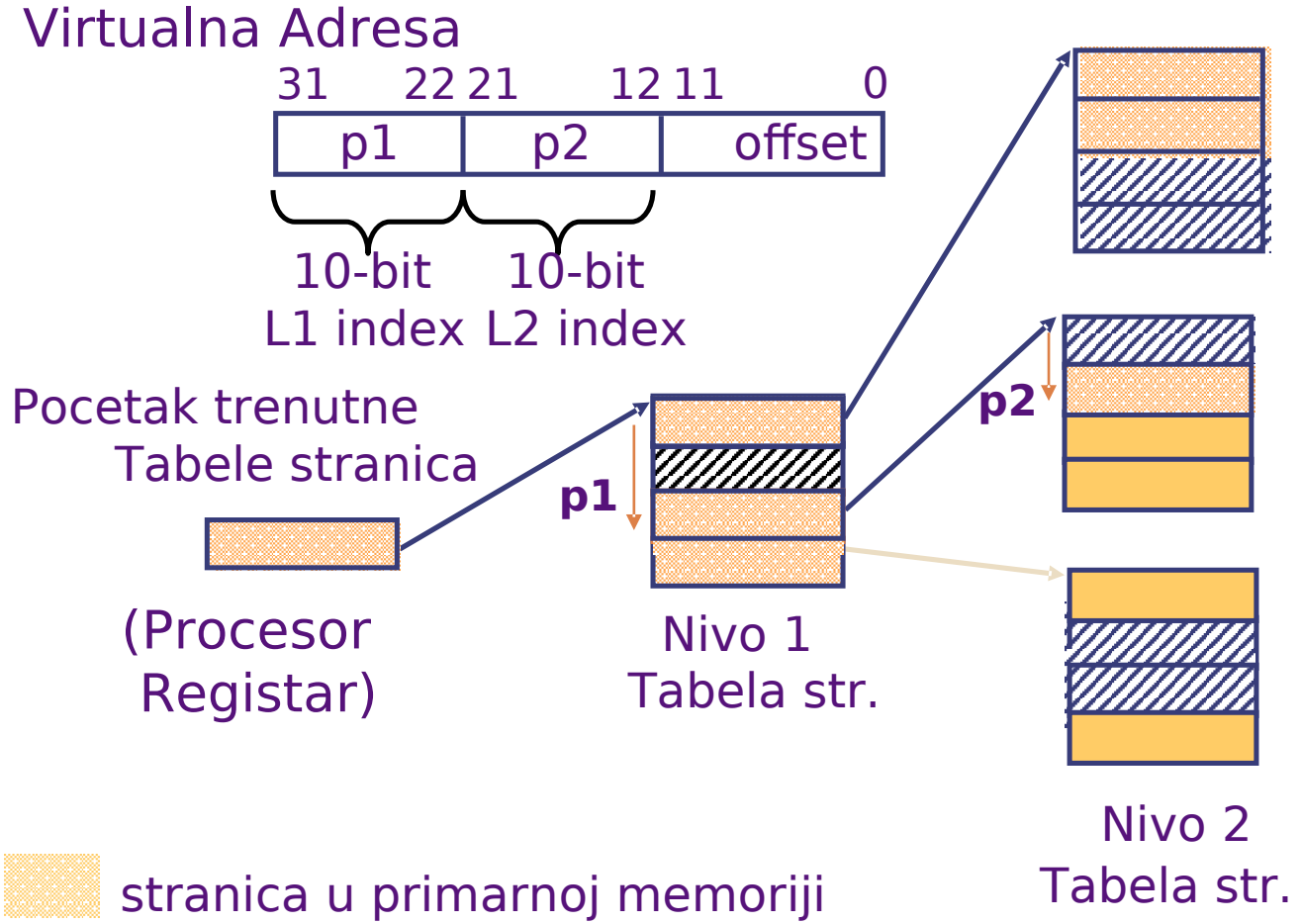
- Cak i stanice od 1MB zahtevaju 2^{44} 8-byte PTE-ova (35 TB!)

Resenje: Koriscenje format sa retkim tabelama stranica (velicina proporcionalna broju *mapiranih* stranica)

Hijerarhijske Tabele Stranica

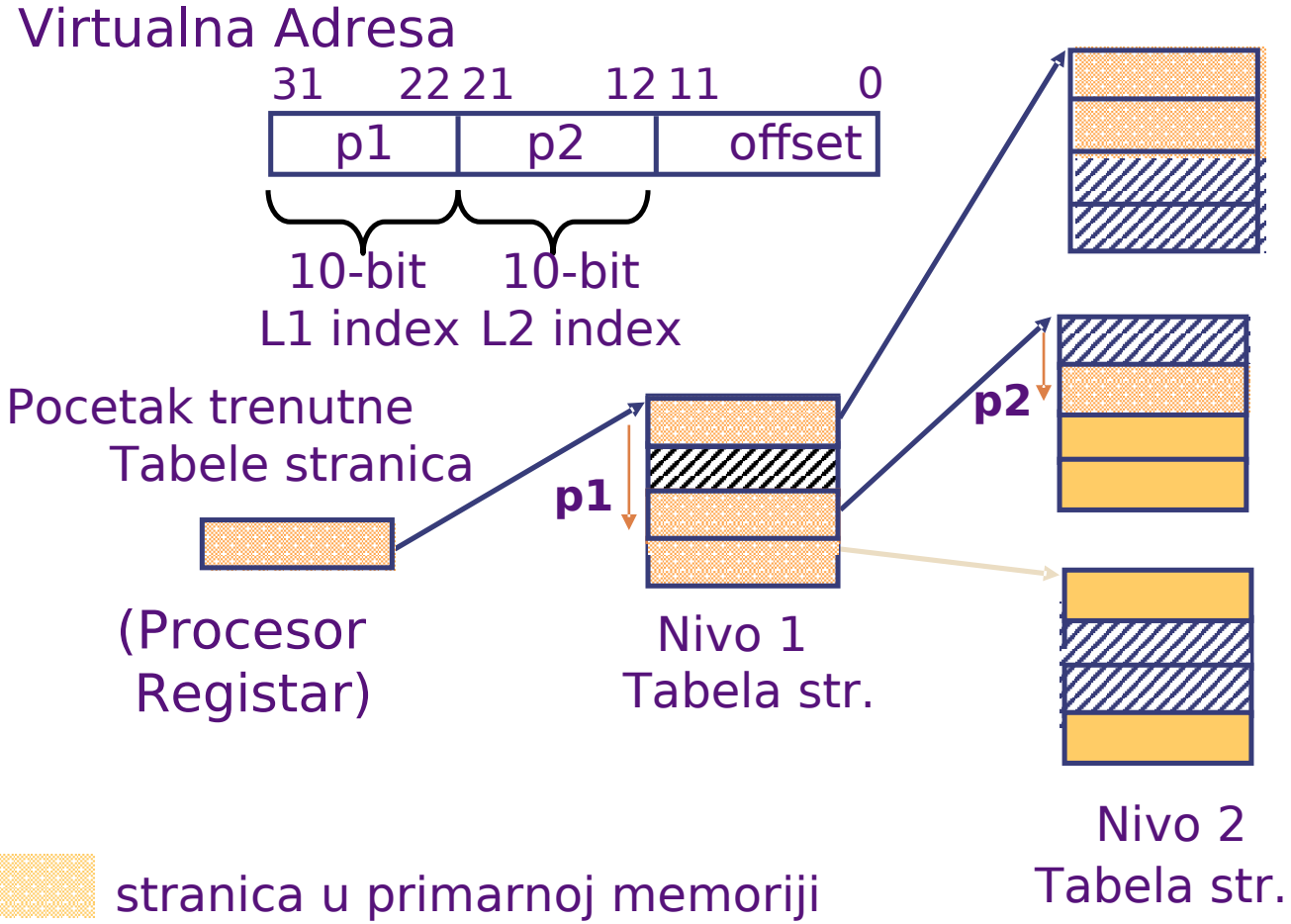


Hijerarhijske Tabele Stranica



Memorija za tebele stranica proporc. memoriji korisce. od stane procesa
-> manje stranica manje nivo 2 ts

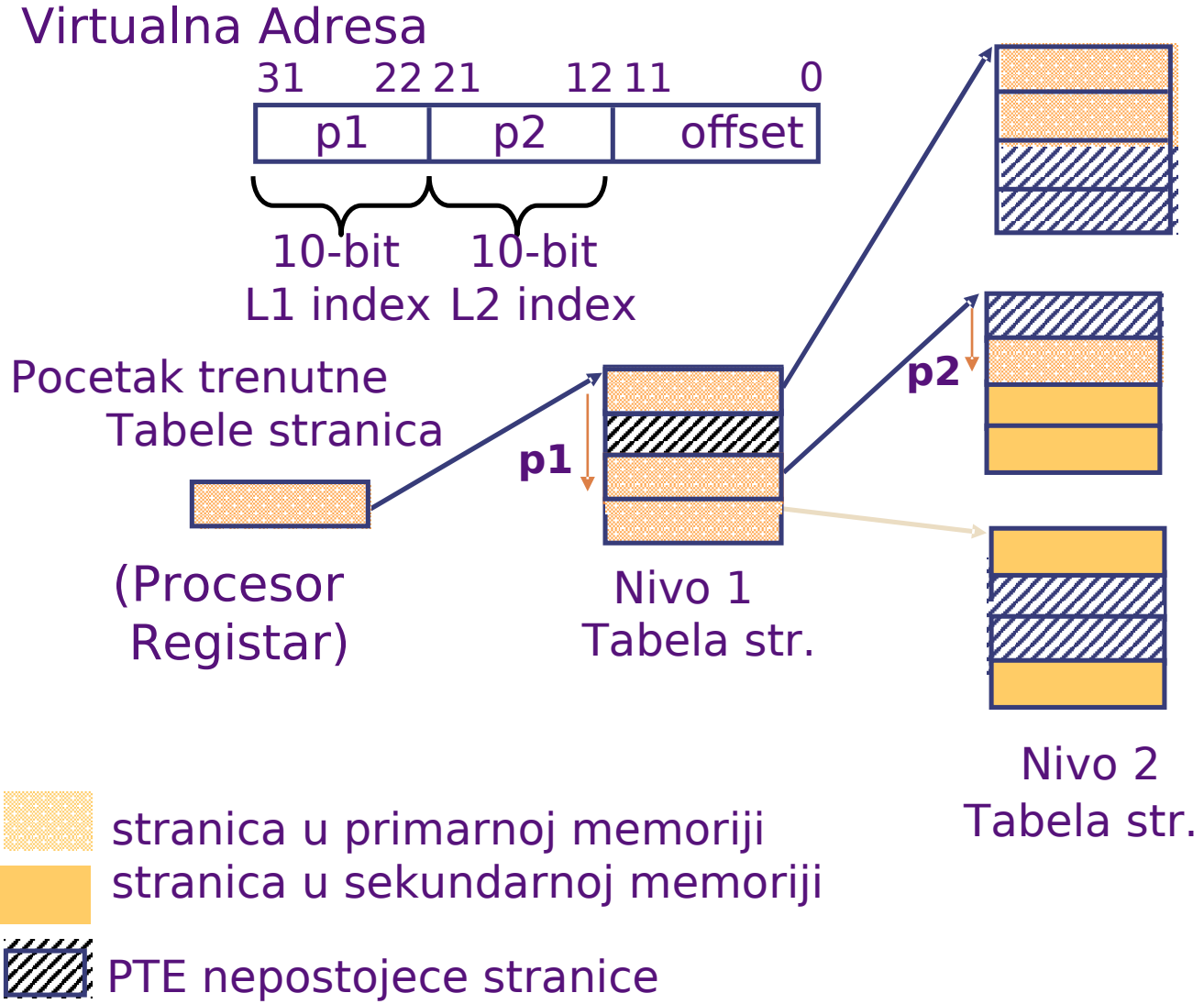
Hijerarhijske Tabele Stranica



Memorija za tebele stranica proporc. memoriji korisce. od stane procesa
-> manje stranica manje nivo 2 ts

Tabela stranica moze biti razbacana
Svaka ts nivoa 1 i nivoa 2 je mala, ugl staje u stranicu

Hijerarhijske Tabele Stranica



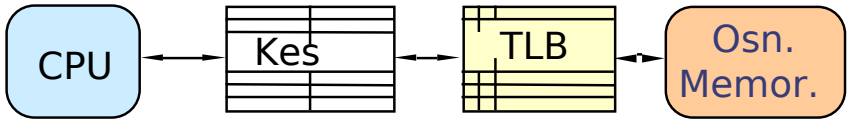
Memorija za tebele stranica proporc. memoriji korisce. od stane procesa
-> manje stranica manje nivo 2 ts

Tabela stranica moze biti razbacana
Svaka ts nivoa 1 i nivoa 2 je mala, ugl staje u stranicu

Prolaz kros ts podr. visestruke pristupe memoriji
TLB cini ovo retkom pojavom

Kesiranje sa Virtualnom Memorijom

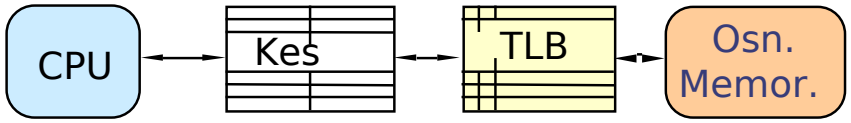
Virtualno-Adresirani
Kes



- Brzo: Bez VA<=>FA
  Translacije kod kes pogotka
- Problem: Mora se azurirati kes
  Nakon zamene konteksta

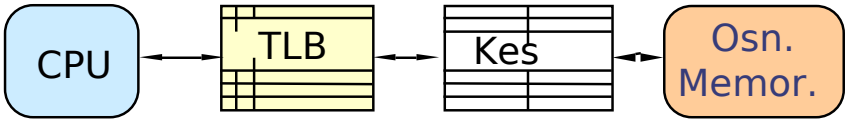
Kesiranje sa Virtualnom Memorijom

Virtualno-Adresirani Kes



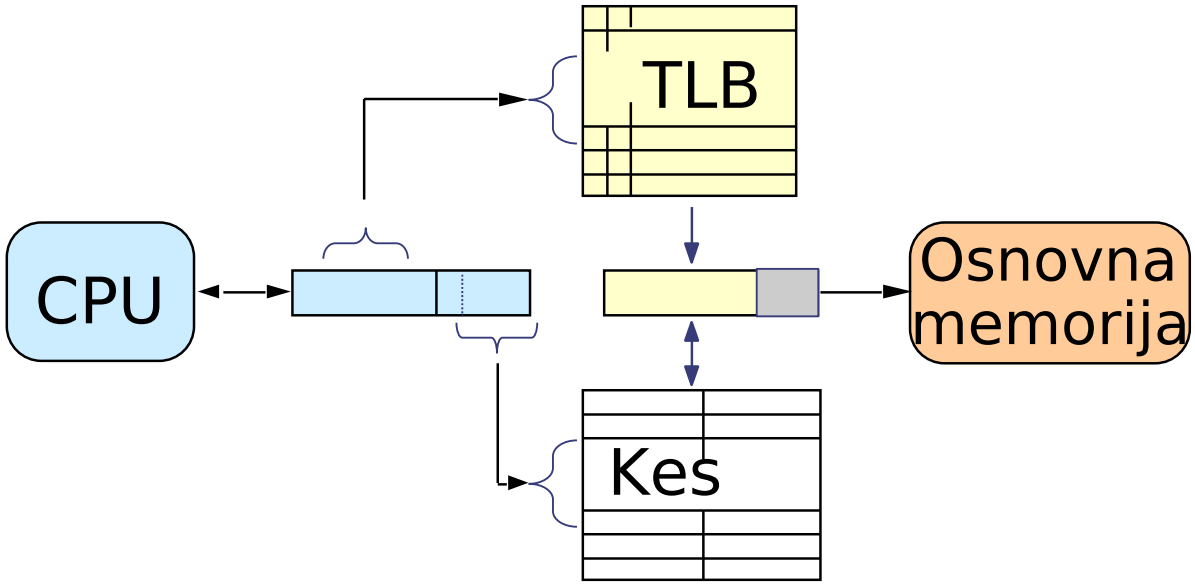
- Brzo: Bez VA<=>FA
  Translacije kod kes pogotka
- Problem: Mora se azurirati kes
  Nakon zamene konteksta

Fizicko-Adresirani Kes



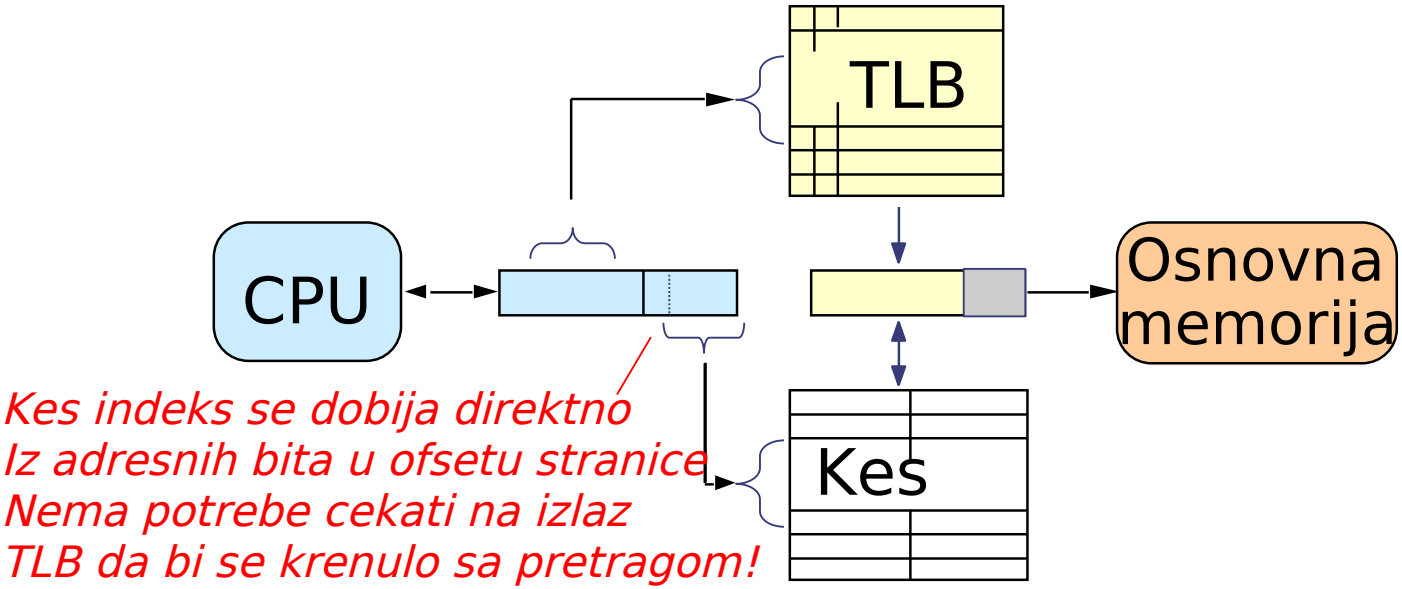
- Izbegava zastarele kes pod.
  Nakon zamene konteksta
- SPORO: VA<=>FA
  Translacija pre svakog
  Pristupa Kesu

Najbolje od oba: Virtualno Indeksirani, Fizicki-Tagovani Kes (VIPT)



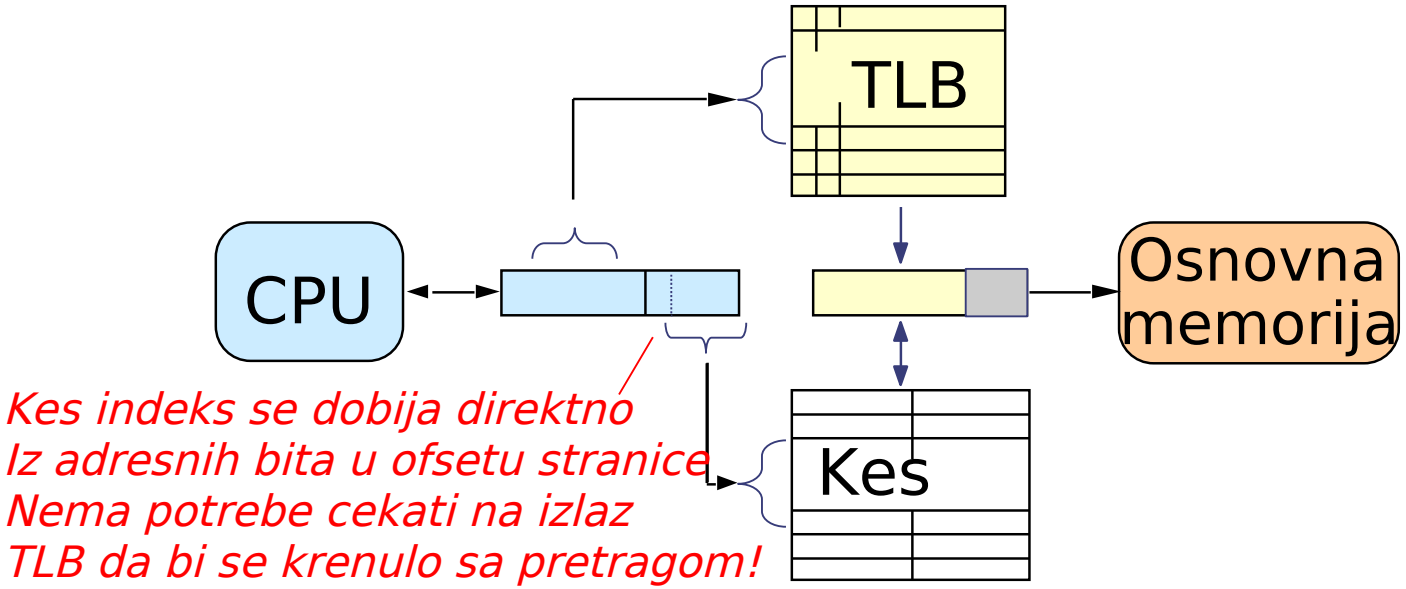
Ako su indeks biti podskup ofset bita u okviru stranice, pristup tag-ovima u fizickom kesu moze se obaviti *u paraleli* sa TLB pristupom. Tag iz kesa se poredi sa fizickom adresom stranice iz TLB-a da bi se odredio pogodak/promasaj (hit/miss).

Najbolje od oba: Virtualno Indeksirani, Fizicki-Tagovani Kes (VIPT)



Ako su indeks biti podskup ofset bita u okviru stranice, pristup tag-ovima u fizickom kesu moze se obaviti *u paraleli* sa TLB pristupom. Tag iz kesa se poredi sa fizickom adresom stranice iz TLB-a da bi se odredio pogodak/promasaj (hit/miss).

Najbolje od oba: Virtualno Indeksirani, Fizicki-Tagovani Kes (VIPT)



Ako su indeks biti podskup ofset bita u okviru stranice, pristup tag-ovima u fizickom kesu moze se obaviti *u paraleli* sa TLB pristupom. Tag iz kesa se poredi sa fizickom adresom stranice iz TLB-a da bi se odredio pogodak/promasaj (hit/miss).

Problem: Limitiran broj bita indeksiranja kesa→ Povecanje kapaciteta kesa jedino moguce povecanjem asocijativnosti!

Zakljucak

- Prednosti virtualne memorije
  - Zastita i privatnost: Privatni adresni prostori za svaki proces
  - Stranice na zahtev: Koriscenje osn. mem. za kesiranje diska

Zakljucak

- Prednosti virtualne memorije
  - Zastita i privatnost: Privatni adresni prostori za svaki proces
  - Stranice na zahtev: Koriscenje osn. mem. za kesiranje diska
- Segmentacija: adresni prostor svakog procesa kontinualan blok (segment) u fizickoj memoriji
  - Jednostavno: Base/bound registri
  - Problem sa fragmentacijom, nema zahteva za stranicama

Zakljucak

- Prednosti virtualne memorije
  - Zastita i privatnost: Privatni adresni prostori za svaki proces
  - Stranice na zahtev: Koriscenje osn. mem. za kesiranje diska
- Segmentacija: adresni prostor svakog procesa kontinualan blok (segment) u fizickoj memoriji
  - Jednostavno: Base/bound registri
  - Problem sa fragmentacijom, nema zahteva za stranicama
- Stranice: Adresni prostor svakog procesa je je smesten u vise stranice fiksne velicine. Tabela stranica mapira virtualne u fizicke
  - Smanjuje fragmentaciju
  - Omogucava stanice na zahtev: stranice mogu biti u memoriji ili disku
  - Zahteva pristup tabelama stranica za svako referenciranje memorije

Zakljucak

- Prednosti virtualne memorije
  - Zastita i privatnost: Privatni adresni prostori za svaki proces
  - Stranice na zahtev: Koriscenje osn. mem. za kesiranje diska
- Segmentacija: adresni prostor svakog procesa kontinualan blok (segment) u fizickoj memoriji
  - Jednostavno: Base/bound registri
  - Problem sa fragmentacijom, nema zahteva za stranicama
- Stranice: Adresni prostor svakog procesa je je smesten u vise stranice fiksne velicine. Tabela stranica mapira virtualne u fizicke
  - Smanjuje fragmentaciju
  - Omogucava stanice na zahtev: stranice mogu biti u memoriji ili disku
  - Zahteva pristup tabelama stranica za svako referenciranje memorije
- Efikasno koriscenje stranica
  - TLB: Kesiranje tabele stranica
  - Hijerarhijske tabele stranica: Obezbedjuju malu velicinu tabele str.