DEO VIII: UPRAVLJANJE MEMORIJOM

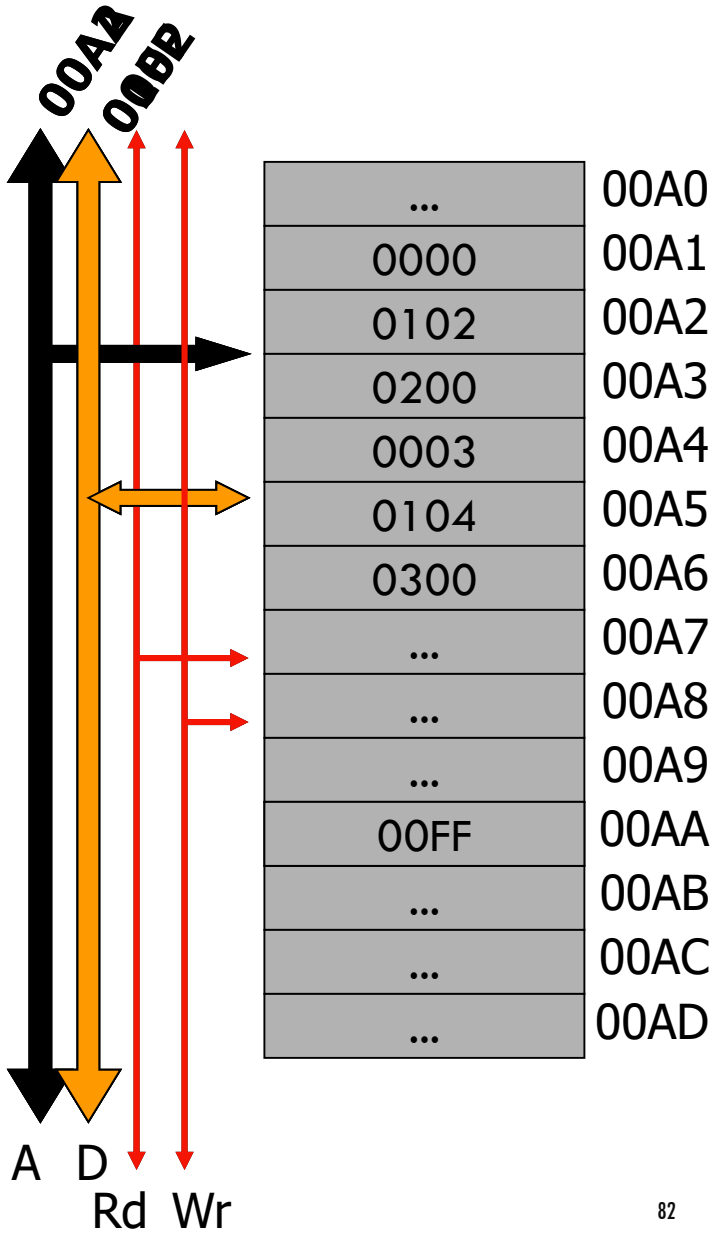
DEO VIII: UPRAVLJANJE MEMORIJOM

Nakon odslušanog ovog dela, trebalo bi da možete da:

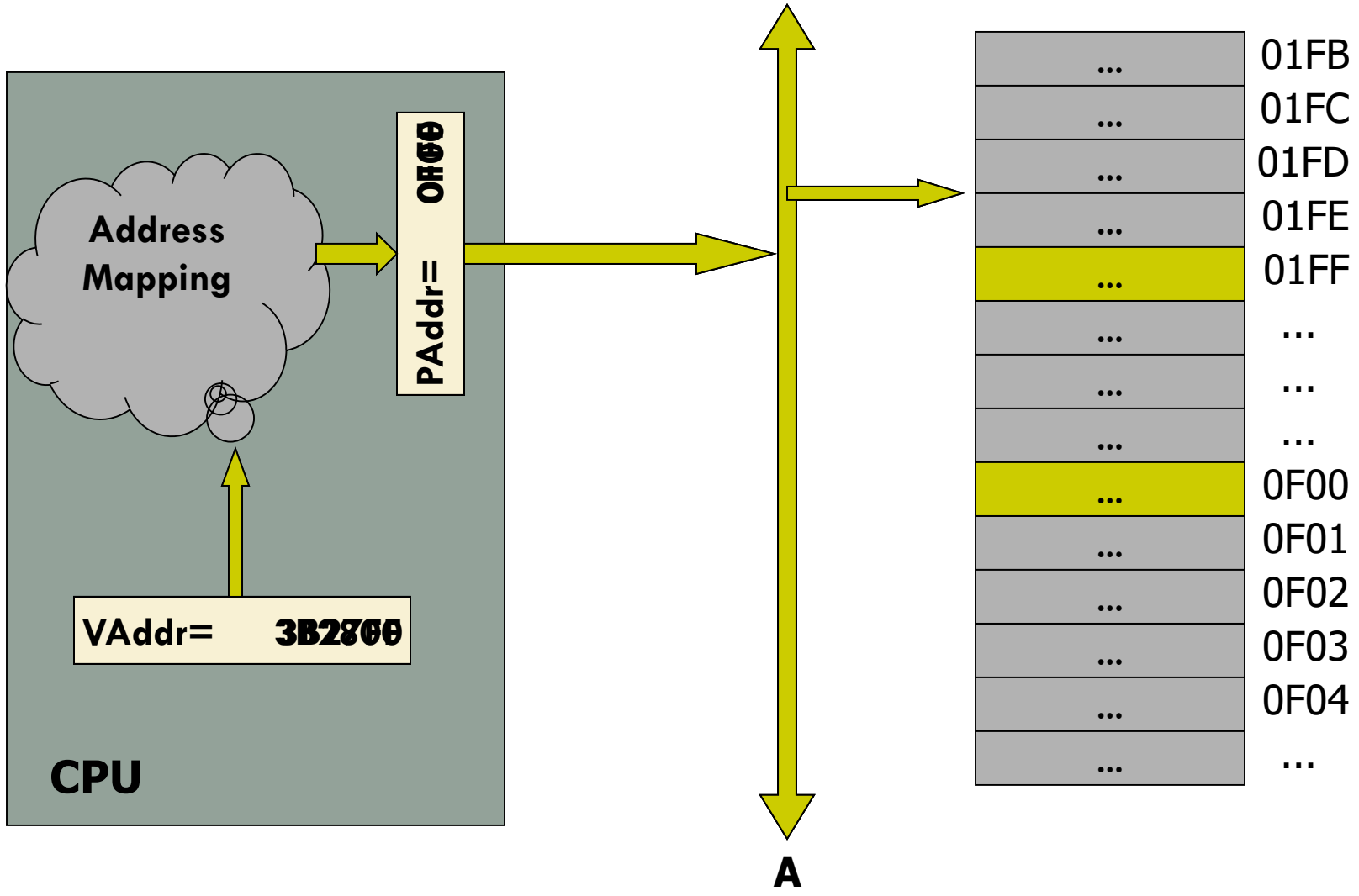
- objasnite kada se mogu dodeliti adrese instrukcijama i podacima;
- objasnite logički i fizički adresni prostor;
- objasnite dinamičko učitavanje;
- objasnite preklapanje i dinamičko povezivanje;
- objasnite šta je prebacivanje;
- navedete i objasnite prednosti i nedostatke kontinualne alokacije memorije;
- objasnite straničnu organizaciju memorije;
- objasnite segmentnu organizaciju memorije;
- objasnite zašto se koristi virtuelna memorija;
- objasnite šta predstavlja tabela preslikavanja.

POJAM VIRTUELNE MEMORIJE

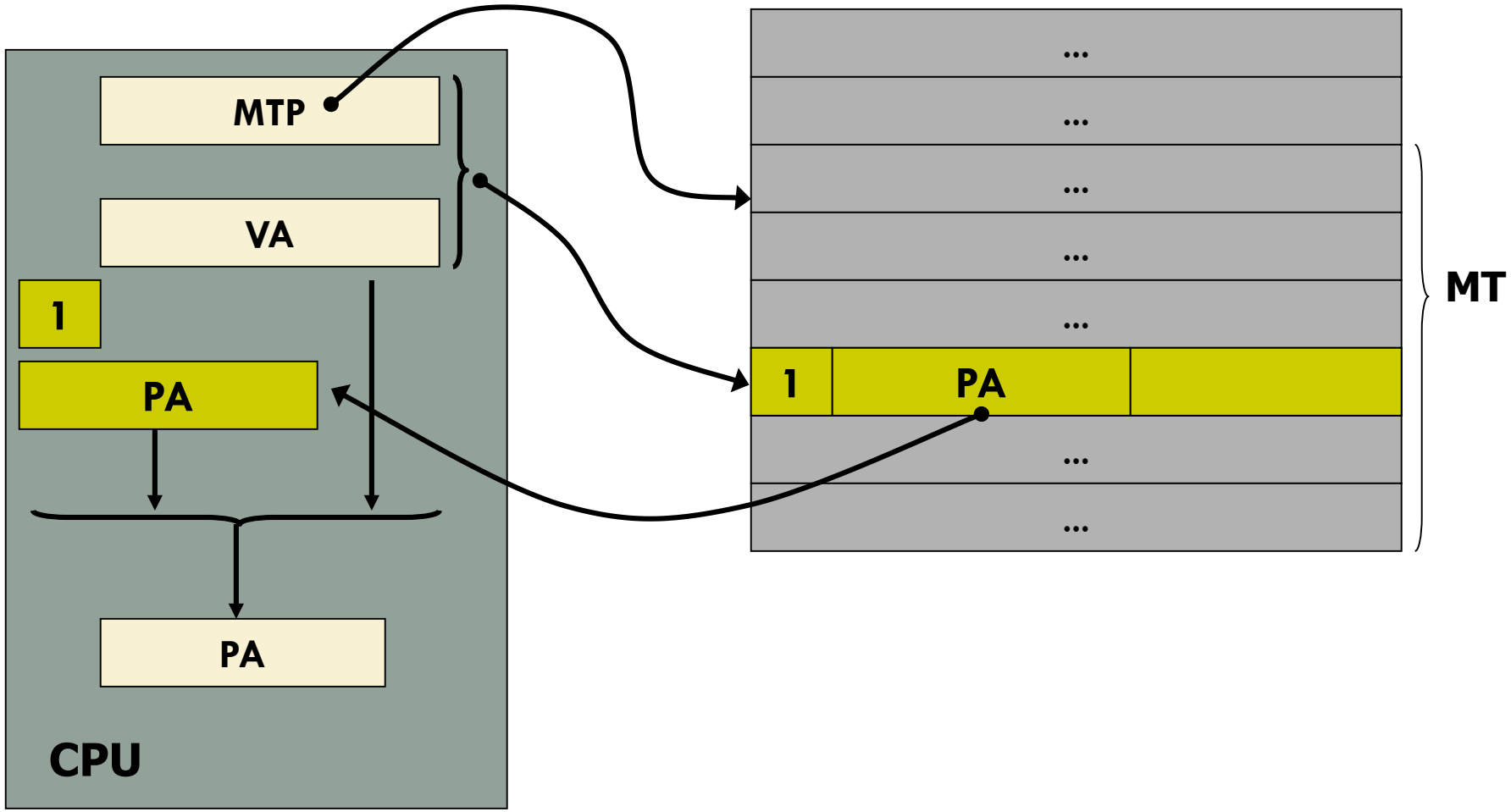
- Fizička memorija računara: linearno uređen niz ćelija sa mogućnošću *direktnog pristupa (direct access)*
- Direktan pristup: svaka ćelija ima svoju *adresu (address)* preko koje joj se može pristupiti
- RAM (*Random Access Memory*): memorija sa mogućnošću čitanja i upisa; gubi sadržaj gubitkom napajanja
- ROM (*Read Only Memory*): memorija samo sa mogućnošću čitanja; obično čuva sadržaj po gubitku napajanja; služi za smeštanje osnovnog sistemskog programa za "podizanje" sistema – učitavanje OS *(bootstrap program)*



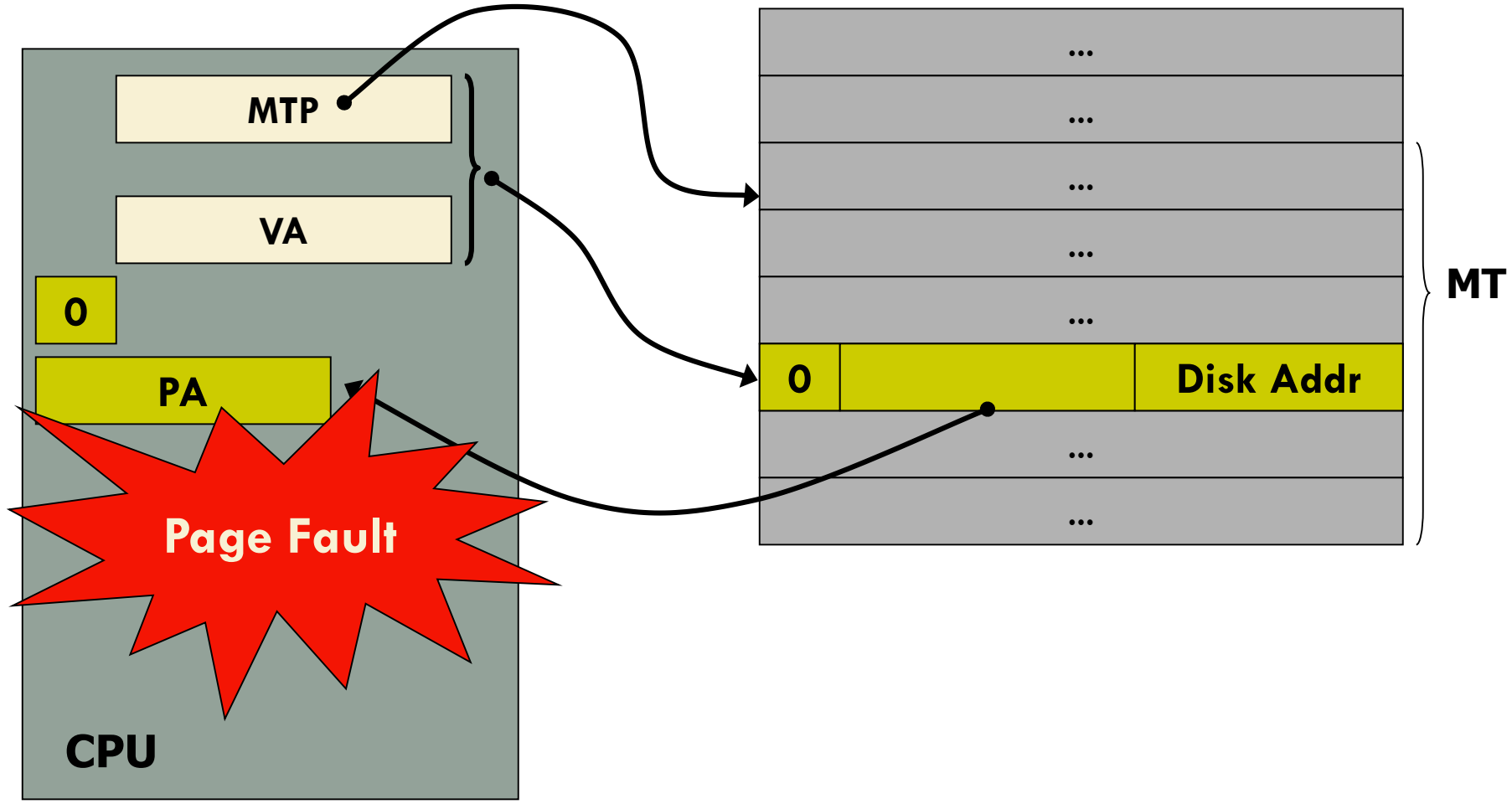
PRESLIKAVANJE ADRESA



PRESLIKAVANJE ADRESA



PRESLIKAVANJE ADRESA

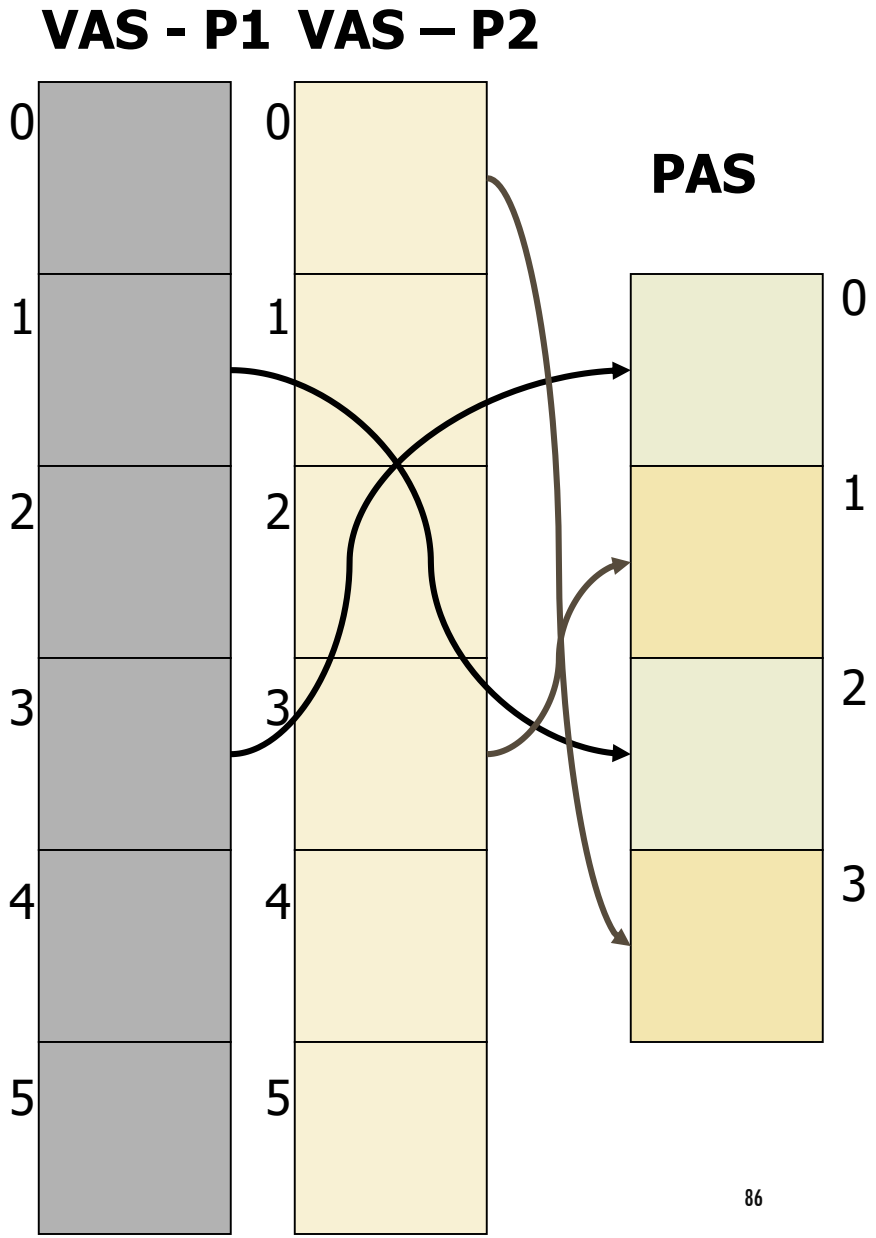


STRANIČNO PRESLIKAVANJE

- Fizička i virtuelna memorija podeljene su na blokove iste veličine – *stranice (page)*
- Jedna stranica virtuelnog adresnog prostora preslikava se u jednu stranicu fizičkog adresnog prostora (blok)

PMT - P1

0	...	...
1	2	...
0	...	...
1	0	...
0	...	...
0	...	...

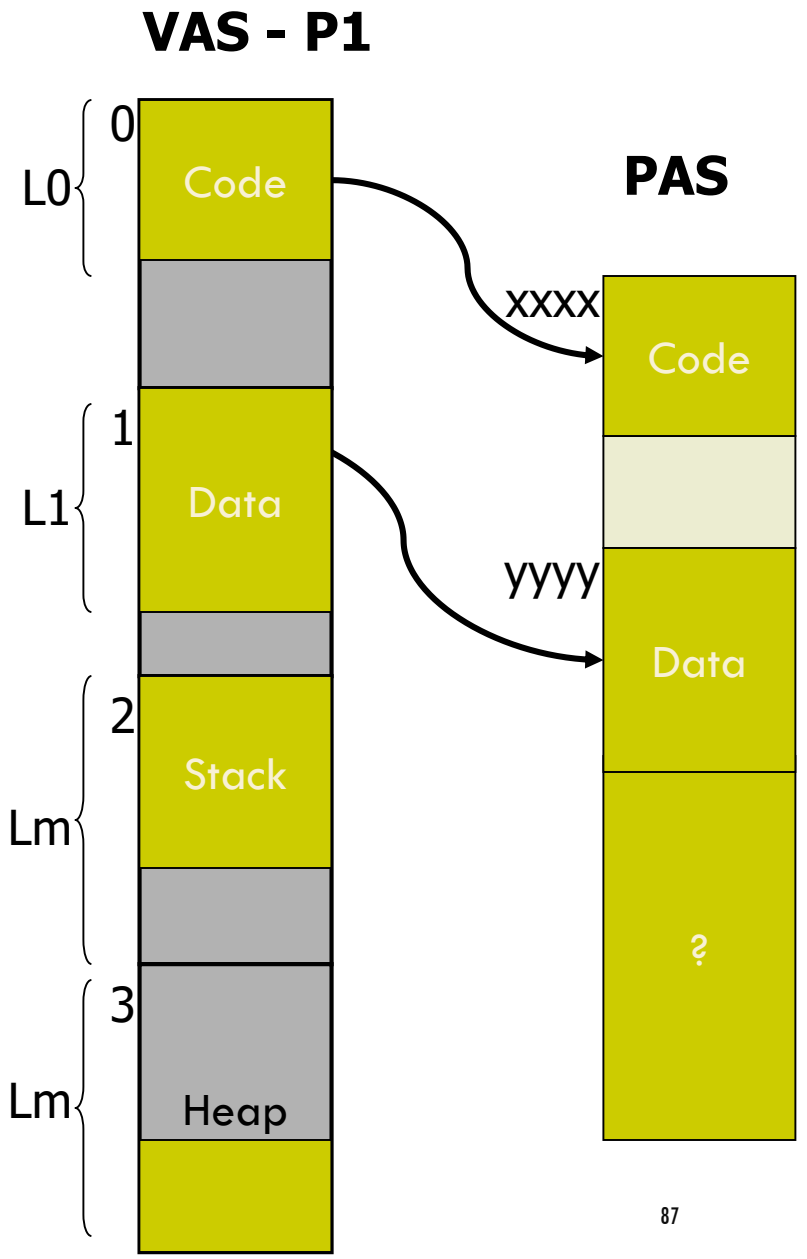


SEGMENTNO PRESLIKAVANJE

- Virtuelna memorija je podeljena na logičke celine različite stvarne veličine - *segmente (segment)*, ali iste *maksimalne veličine*, npr. kod, statički podaci, stek, dinamička memorija itd.
- OS pronalazi prostor potrebne veličine za smeštanje segmenta

SMT - P1

		Size	
1	**xxxx**	**L0**	**...**
1	**yyyy**	**L1**	**...**
0	**...**	**Lm**	**...**
0	**...**	**Lm**	**...**



SEGMENTNO-STRANIČNO PRESLIKAVANJE

- Zadatak: prikazati šematski strukturu VA i PA i postupak preslikavanja VA u PA.
- Prednosti:
  - stranice su logičke celine, pa je prirodno uvoditi zaštitu pristupa segmentu i prekoračenja veličine; npr. Code segment je *read-only*
  - nema eksterne fragmentacije
  - jednostavan zadatak za OS u pogledu alokacije prostora – nema potrebe za "upasivanjem" bloka memorije
- Nedostaci:
  - najsloženiji hardver, najmanje efikasno preslikavanje (u odnosu na ostala preslikavanja)
  - interna fragmentacija

TLB

- Problem: bez obzira na preslikavanje, procesor mora da dovlači deskriptor iz memorije za svako preslikavanje VA u PA (nekoliko puta tokom instrukcije) – neefikasno!
- Ideja: obezbediti malu, ali efikasnu memoriju koja čuva samo deo podataka potrebnih za preslikavanje, kao što to radi *keš (cache)* memorija – *Translation Lookaside Buffer* (TLB)
- TLB treba da obezbedi brzo preslikavanje određenog dela VA u podatke iz deskriptora potrebne procesoru:
  - ako je podatak u TLB-u, procesor ga dobija brzo
  - ako nije, TLB dovlači podatak iz deksriptora čitanjem iz tabela u memoriji