

Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko



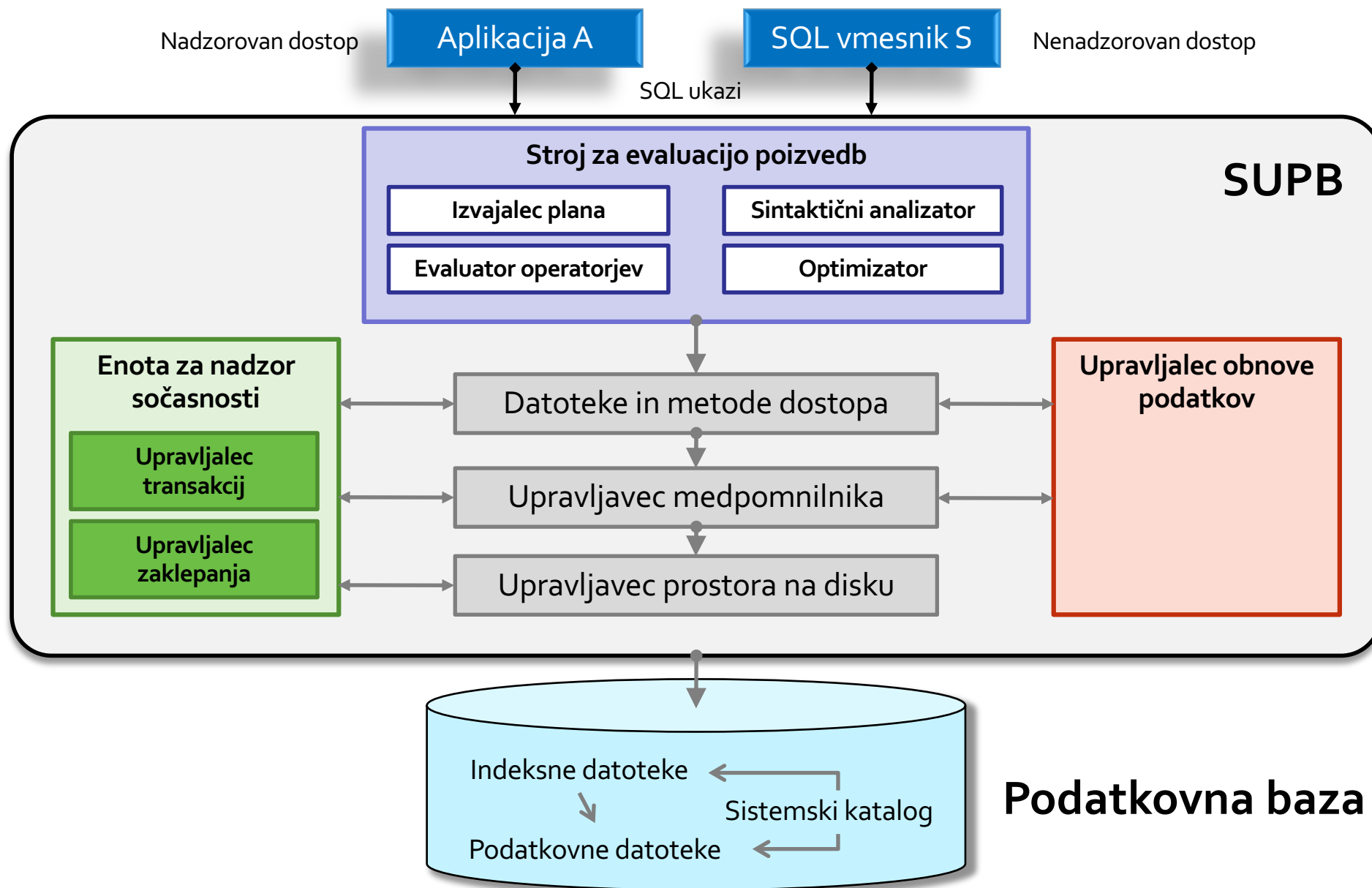
Predmet: Osnove podatkovnih baz

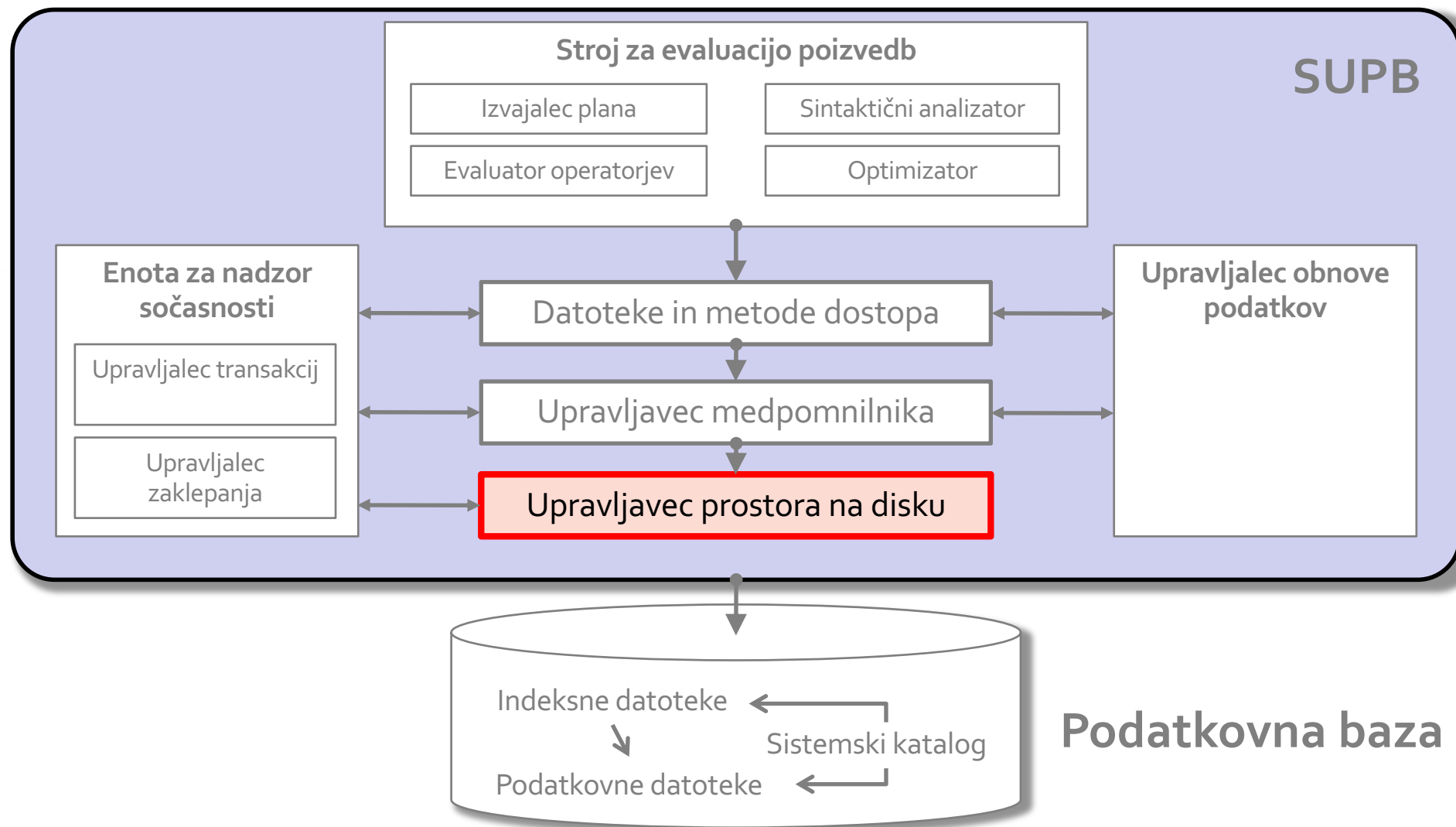
Modul:
Upravljanje z diskom in
pomnilnikom

Gradivo:
v.2015

Vsebina

- Zgradba SUPB
- Upravljanje s prostorom na disku
- Upravljanje z medpomnilnikom
- Strategije zamenjave strani v medpomnilniku





Upravljanje s prostorom na disku...

- Upravljanje z diskom - **upravljalac z diskom** (*Disk Space Manager*).
- Lastnosti:
 - Skrije podrobnosti strojne opreme in OS; omogoča, da ostale komponente SUPB vidijo podatke kot zbirko strani.
 - Izvaja ukaze za dodeljevanje in sproščanje prostora na disku ter branje in pisanje;
 - Vse operacije izvaja nad enotami - stran (*page*). Stran = blok na disku. Branje/pisanje strani → ena I/O operacija;

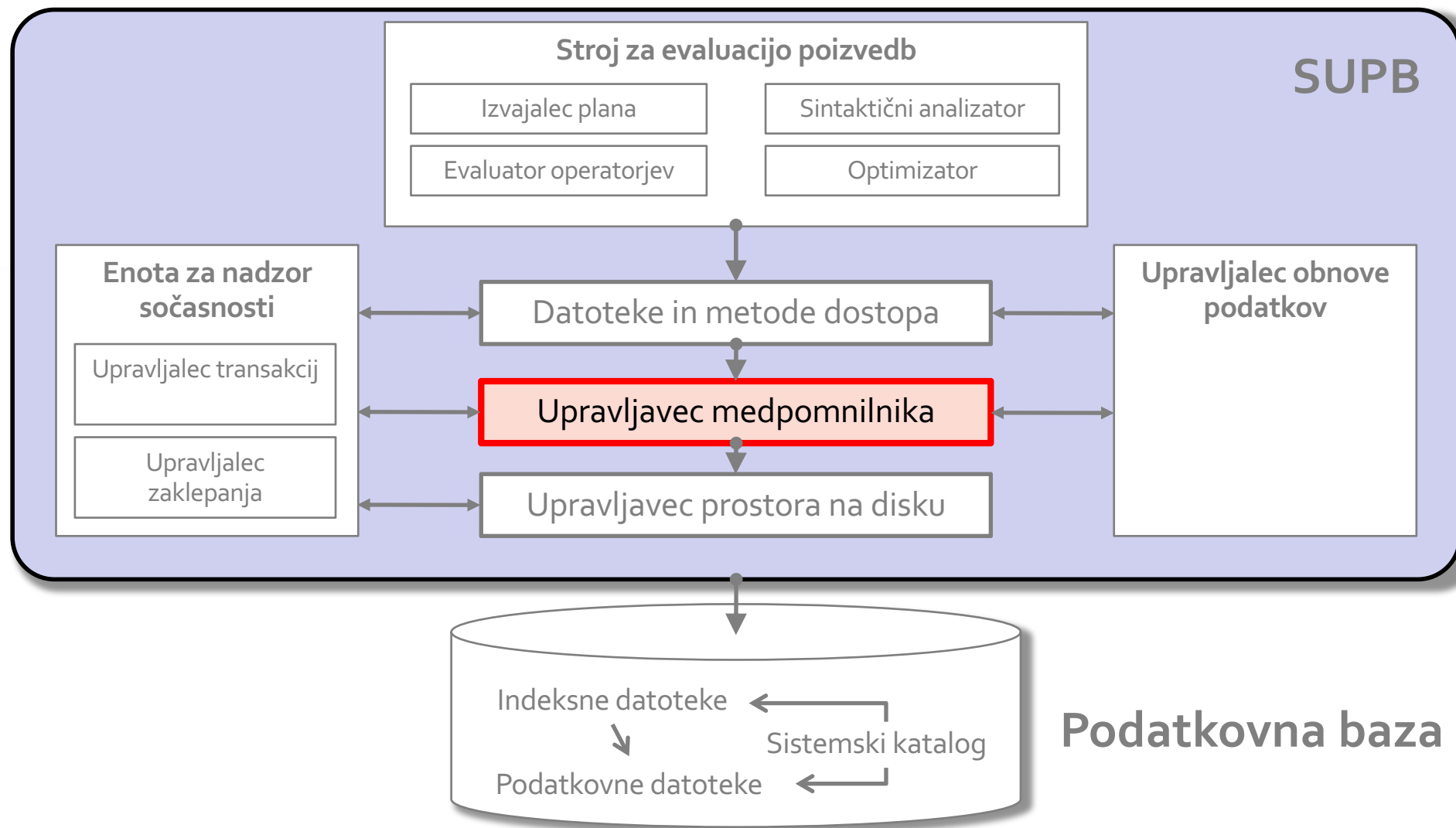


Upravljanje s prostorom na disku...

- Vzdržuje stanje zasedenih in prostih blokov na disku.
- Obstajata dva načina:
 - **Vzdrževanje seznama prostih blokov** (kazalec na prvi blok seznama se shrani na znano lokacijo na disku),
 - **Vzdrževanje bitne mape** (za vsak blok je v bitni mapi bit, ki označuje, ali je blok zaseden ali ne).

Upravljanje s prostorom na disku...

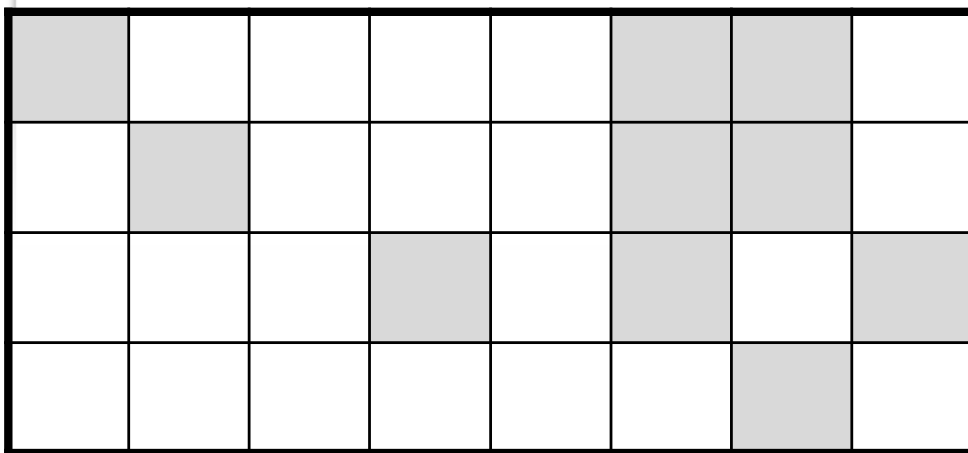
- Uporaba **datotečnega sistema** za upravljanje s prostorom:
 - Upravljalca z diskom lahko uporablja datoteke operacijskega sistema → celotna PB se nahaja v eni ali več datotekah.
 - V tem primeru je zadolžen za upravljanje prostora v teh datotekah.
 - Veliko PB ne uporablja datotečnega sistema, ampak svoj lastni sistem za upravljanje z diskom (popolnoma svoj ali pa razširja funkcionalnost datotečnega sistema OS). Razlogi:
 - PRAKTIČNI: bazo lahko uporabimo na več platformah,
 - TEHNIČNI: pri 32 bitnem naslavljanju se pojavi omejitev v velikosti datoteke.



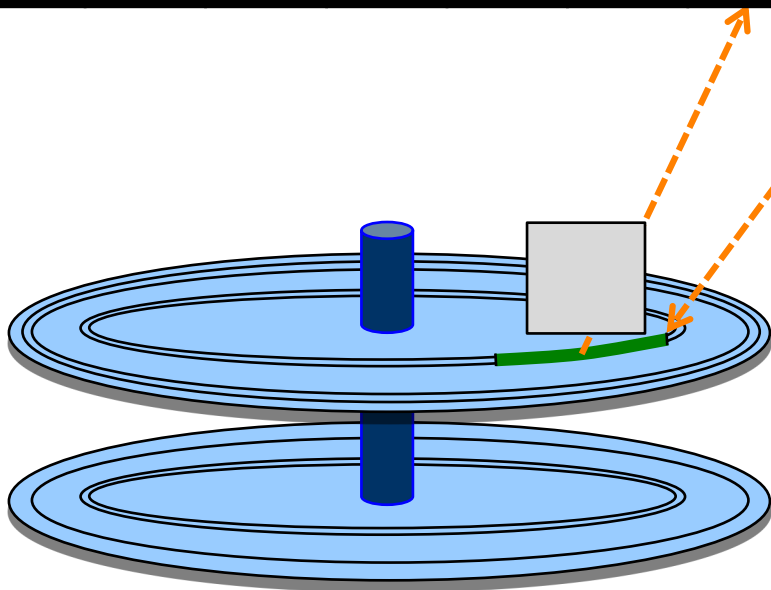
Upravljanje medpomnilnika...

- **Upravljaec medpomnilnika** programska plast, skrbi za prenašanje strani v pomnilnik.
 - upravlja z razpoložljivim pomnilnikom (*buffer pool*).
 - višjim plastem zagotavlja strani... prenese kar zahtevajo
 - višje plasti obveščajo o straneh, ki se sprostijo, spremenijo.

Upravljanje medpomnilnika...

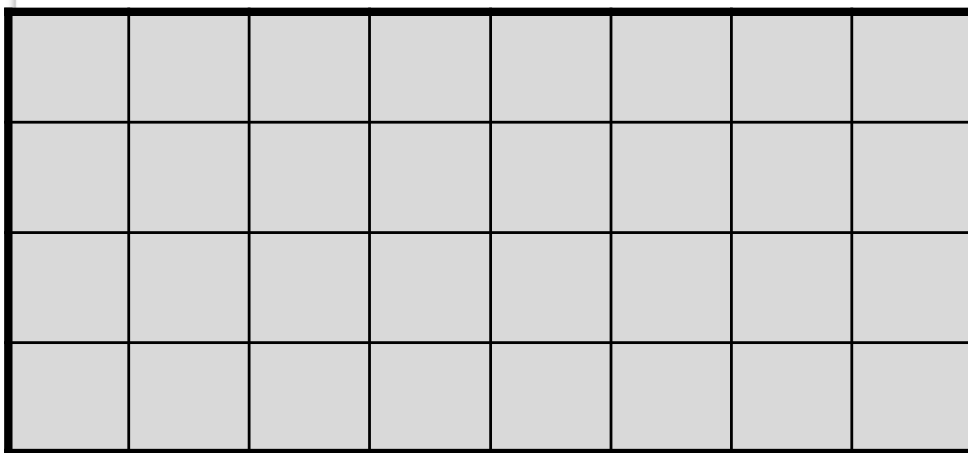


```
select * from employees
where emp_no = 10002;
```

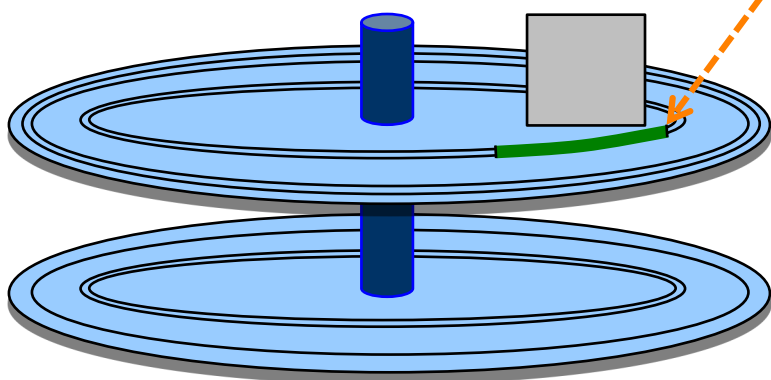


emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10011	1953-11-07	Mary	Sluis	F	1990-01-22
10012	1960-10-04	Patricio	Bridgland	M	1992-12-18
10013	1963-06-07	Eberhardt	Terkki	M	1985-10-20
10014	1956-02-12	Berni	Genin	M	1987-03-11
10015	1959-08-19	Guoxiang	Nooteboom	M	1987-07-02

Upravljanje medpomnilnika...



```
select * from employees  
where emp_no = 10002;
```



emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10011	1953-11-07	Mary	Sluis	F	1990-01-22
10012	1960-10-04	Patricio	Bridgland	M	1992-12-18
10013	1963-06-07	Eberhardt	Terkki	M	1985-10-20
10014	1956-02-12	Berni	Genin	M	1987-03-11
10015	1959-08-19	Guoxiang	Nooteboom	M	1987-07-02

Upravljanje medpomnilnika...

- Za vsak okvir v medpomnilniku dve spremenljivki:
 - `pin_count`: št. zahtev po neki strani brez sprostitev (št. trenutnih uporabnikov strani).
 - `dirty`: stran spremenjena: `true/false`.

- Začetno stanje okvirja:

- `pin_count = 0`
- `dirty = false`

P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F

Primer

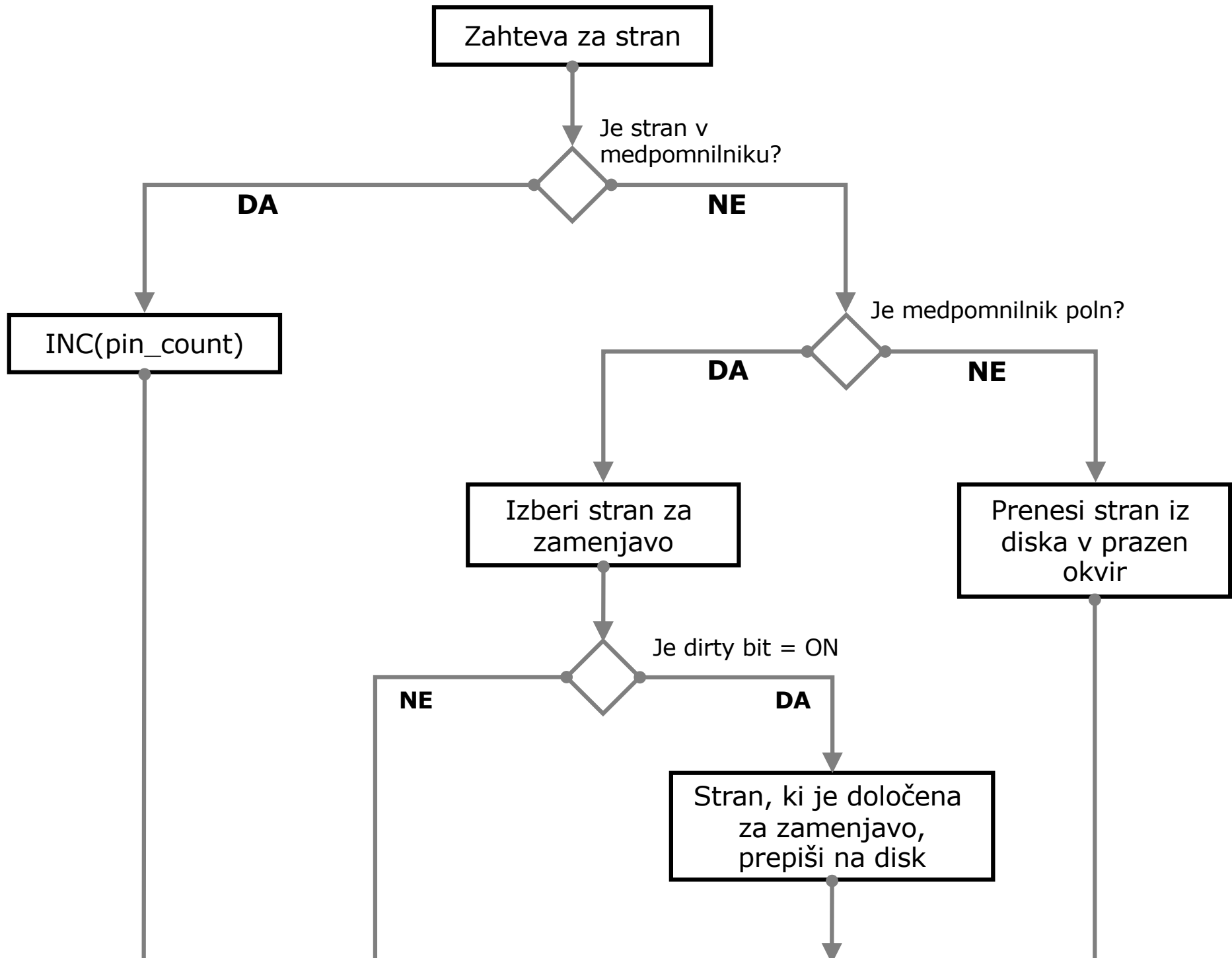
T1:
select * from employees
where emp_no = 10002;

T2:
update employees
set last_name = 'Julius'
where emp_no = 10008;

P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=1 D=F	P=0 D=F	P=0 D=F	P=1 D=T	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F

Postopek...

- Ko se pojavi zahteva po določeni strani, upravljalec z medpomnilnikom izvede naslednje:
 - če se stran nahaja v kakšnem od okvirjev, vrne pomnilniški naslov okvirja in poveča `pin_count` za 1,
 - sicer izvede naslednje:
 - izbere okvir za zamenjavo (z uporabo strategije za zamenjavo).
 - če je `dirty bit` okvirja, ki bo zamenjan, postavljen na “on”, se stran prepíše na disk.
 - stran se prenese iz diska v okvir, ki je določen za zamenjavo.
 - `pin_count` okvirja se postavi na 1
 - ko sistem sporoči, da se stran sprostila, se `pin_count` zmanjša za 1. Ko doseže vrednost 0, sistem prepíše stran nazaj na disk (če `dirty bit` ON)



Postopek

- Dodatna pravila:
 - Zahtevana stran ni v medpomnilniku, vsi okvirji zasedeni → izberi okvir s `pin_count=0`. Če več takih → uporabi **strategijo izbire!**
 - Ni iskane strani, ni strani s `pin_count=0` → čakaj na sprostitvev (**transakcija razveljavljena**).

Konfliktne spremembe...

- Nevarnost: če neko stran zahteva več neodvisnih transakcij, lahko pride do **konfliktnih sprememb**...

P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=2 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F
P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=0 D=F

Konfliktne spremembe

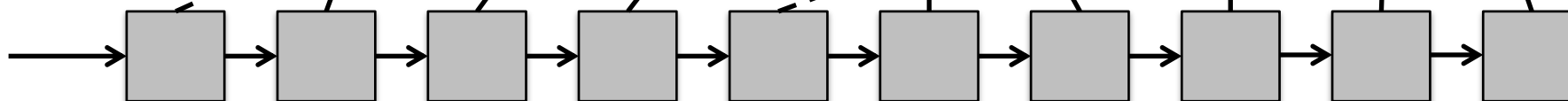
- Reševanje z zaklepanjem:
 - Obstaja **protokol zaklepanja**, za katerega skrbijo višje ravni SUPB (posebej **upravljalec transakcij**).
 - Vsaka transakcija lahko pridobi **deljeno** (*shared*) ali **ekskluzivno** (*exclusive*) zaklepanje preden lahko stran bere ali spreminja.
 - Ekskluzivno zaklepanje iste strani ne sme biti odobreno dvema transakcijama istočasno!

Zamenjava strani v medpomnilniku...

- **Strategija zamenjave strani** v medpomnilniku močno vpliva na učinkovitost SUPB.
- Različne strategije primerne za različne situacije.
- Strategija **LRU** – *Least Recently Used*
 - Vrsta kazalcev na okvirje s `pin_count = 0`
 - Ko stran postane kandidat za zamenjavo (`pin_count = 0`), okvir strani dodamo na konec vrste
 - Za zamenjavo izberemo stran iz okvirja, na katerega kaže prvi kazalec v vrsti
 - Problem: časovna kompleksnost

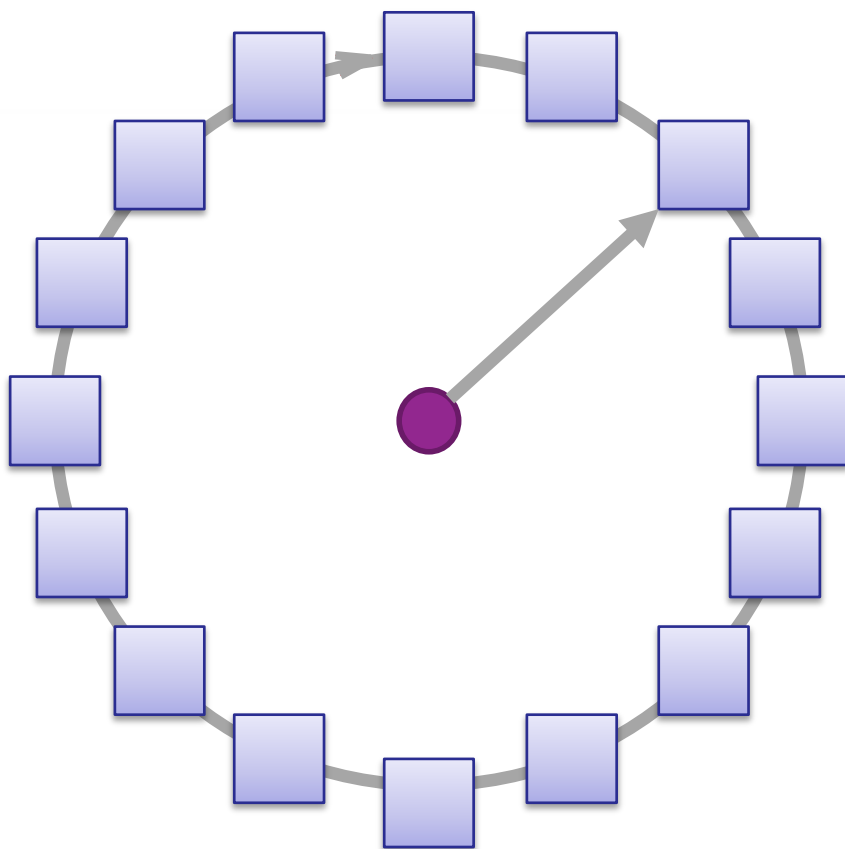
Zamenjava strani v medpomnilniku...

P=1 D=T	P=1 D=F	P=4 D=F	P=1 D=F	P=1 D=F	P=1 D=F	P=0 D=T	P=1 D=F
P=1 D=F	P=0 D=F	P=1 D=T	P=0 D=F	P=0 D=F	P=5 D=F	P=1 D=F	P=0 D=F
P=1 D=F	P=2 D=F	P=1 D=F	P=1 D=F	P=0 D=F	P=1 D=F	P=1 D=F	P=0 D=F
P=1 D=F	P=0 D=F	P=1 D=F	P=7 D=F	P=0 D=F	P=0 D=F	P=0 D=F	P=1 D=F



Zamenjava strani v medpomnilniku...

- Strategija **urne zamenjave** (*clock replacement*)



Vsaka stran ima:

- R - reference bit
- P - Pin count

Algoritem

Ko P postane 0 postavimo R=1;
Če potrebna zamenjava,
preverimo stran, kamor kaže kazalec;
Če velja $P > 0$ ali $R = 1$, pomaknemo kazalec naprej
Sicer stran uporabimo za zamenjavo.

Upravljanje navideznega pomnilnika z OS

- Upravljanje z navideznim pomnilnikom OS in medpomnilnikom pri SUPB podobno.
- Cilj pri SUPB: zagotoviti **učinkovit** dostop do podatkov, ko jih sistem rabi. Če je podatek v pomnilniku – dobro, sicer slabo.
- SUPB tipično realizira svojega upravljavca z medpomnilnikom:
 - lahko bolje predvidi zaporedje (vzorci dostopanja) kot tipičen OS. Strategija **vnaprejšnjega branja** (*prefetching*).
 - SUPB rabi več nadzora nad stranmi, ki se zapisujejo na disk, kot ga omogoča tipičen OS.