Baze podataka

Katedra za računarstvo
Elektronski fakultet u Nišu

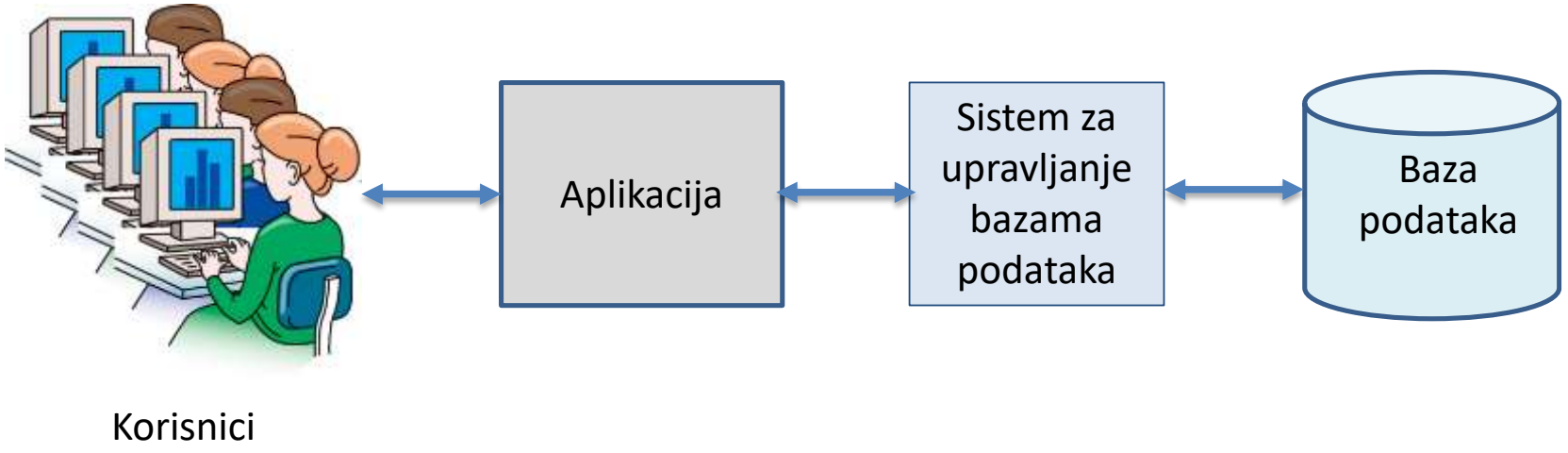
Transakciona obrada

Prof.dr Leonid Stoimenov

Pregled

- Višekorisnički DBMS
- Transakcije
- Upravljanje transakcijama
  - Kontrola konkurencije
  - Oporavak baze podataka

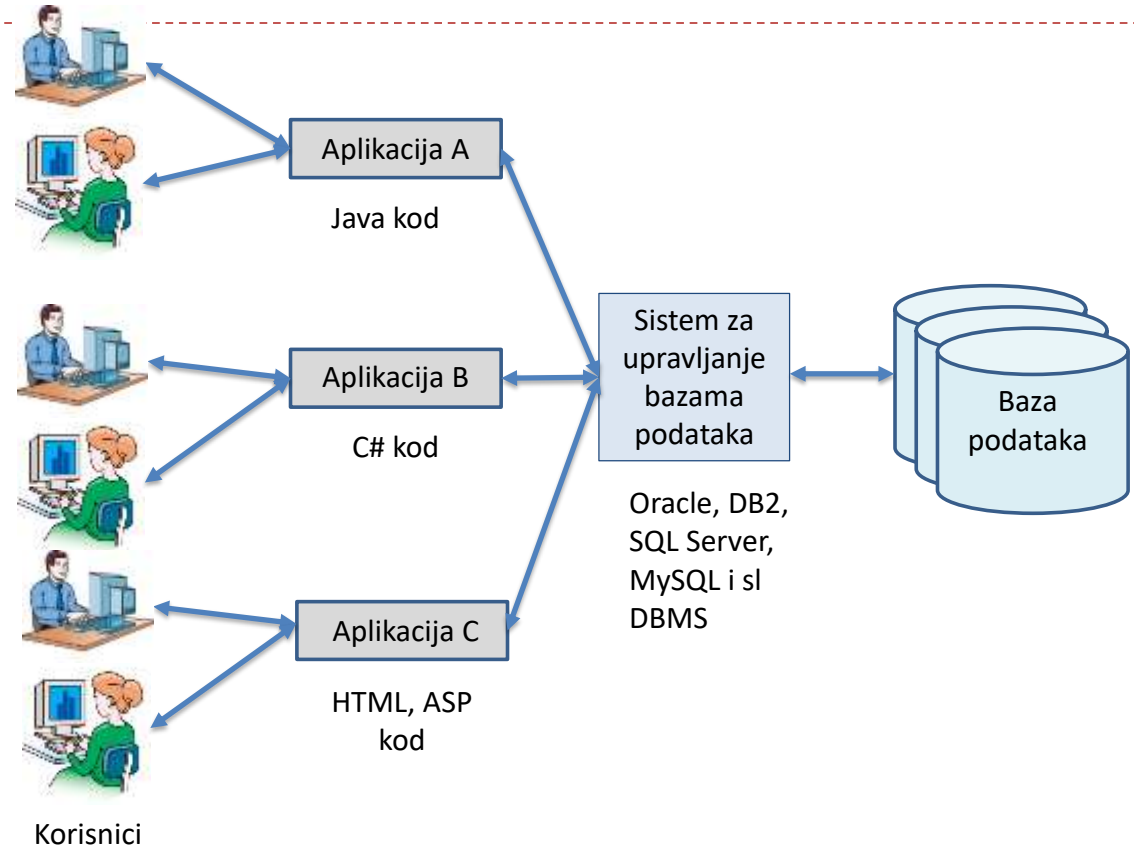
Višekorisnički DBMS



Sistem baza podataka

- Bazu podataka obično koristi više različitih aplikacija
  - Aplikativni programi pisani na nekom programskom *host* jeziku
  - Interaktivni upit ("ad hoc" upit) realizovan na nekom upitnom jeziku

Višekorisnički DBMS



- Višekorisnički DBMS omogućava većem broju korisnika da konkurentno koriste bazu podataka
  - višekorisnički DBMS zahteva i višekorisnički OS

Transakcije

- Različiti korisnici mogu pokrenuti više različitih izvršenja istog "programa" DBMSa koji pristupa i eventualno ažurira neka polja u bazi podataka

- **Transakcija** je program u izvršenju DBMSa koji predstavlja logičku jedinicu obrade baze podataka
  - sadrži jednu ili više operacija koje pristupaju bazi podataka (umetanje, brisanje, modifikacija, pretraživanje,…)

- Transakcija mora da vidi bazu podataka kao konzistentnu
- Konkurentno korišćenje baze podataka preko DBMSa krije u sebi određene opasnosti
  - **Primer:** Jedna transakcija učita u svoj radni prostor neki podatak iz baze podataka, a druga ga odmah nakon toga u bazi podataka izmeni. Prva transakcija će koristiti netačan podatak.

ACID svojstva transakcije

- ***Atomičnost (Atomicity).*** **Transakcija je atomska jedinica obrade što znači da se mora izvršiti u celini**
  - Za očuvanje atomičnosti transakcije zadužen je deo DBMSa **Menadžer oporavka**
    - U slučaju kraha sistema menadžer oporavka mora obezbediti **poništavanje** uticaja transakcije na bazu podataka
- ***Konzistentnost (Consistency).*** Korektno izvođenje svake transakcije treba da prevede bazu podataka iz jednog u drugo konzistentno stanje
  - Za očuvanje ovog svojstva odgovoran je **programer aplikacije** nad bazom podataka ili DBMS komponenta za **proveru integriteta**

ACID svojstva transakcije (nast.)

- ***Izolovanost (Isolation).*** **Sve promene koje nad bazom podataka čini jedna transakcija su nevidljive za ostale transakcije sve dok ih transakcija ne komituje (potvrdi).** Ako se ovo ograničenje strogo poštuje, rešava se problem privremenog ažuriranja, tj. nije neophodno kaskadno vraćanje transakcije unazad. Različite metode konkurencije i oporavka uvode različite varijante ovog ograničenja
  - Za očuvanje ovog svojstva odgovorna je DBMSova **metoda kontrole konkurencije**
- ***Postojanost (Durability).*** **Sve promene koje transakcija učini nad bazom podataka nakon komitovanja nesmeju biti izgubljene**
  - Za očuvanje ovog svojstva odgovoran je DBMSov **menadžer oporavka**

„Granice“ transakcije

- Korisnik specificira granica transakcije u aplikacionom programu – uz eksplicitno navođenje naredbi
  - **BEGIN TRANSACTION** i
  - **END TRANSACTION**
  - Sve operacije za pristup bazi podataka koje se nalaze između ove dve naredbe smatraju se jednom transakcijom
- Jedan aplikacioni program može sadržati više transakcija
- U DBMS-u se implementira protokol **kontrole konkurencije** i **zaključavanja podataka** koji upravlja konkurentnim pristupima bazi podataka

COMMIT i ROLLBACK

- **COMMIT** specificira uspešan kraj transakcije
  - Sve promene BP koje su posledica operacija transakcije su zapamćene
  - Te promene su nadalje vidljive drugim transakcijama
- **ROLLBACK** signalizira da je neuspešan kraj transakcije
  - Sve promene BP koje su posledica tranakcije se poništavaju
  - Stanje je kao da transakcija nikad nije ni izvršena

Primer transakcije u SQL-u

```
UPDATE authors
SET au_fname = 'John'
WHERE au_id = '172-32-1176‘
```

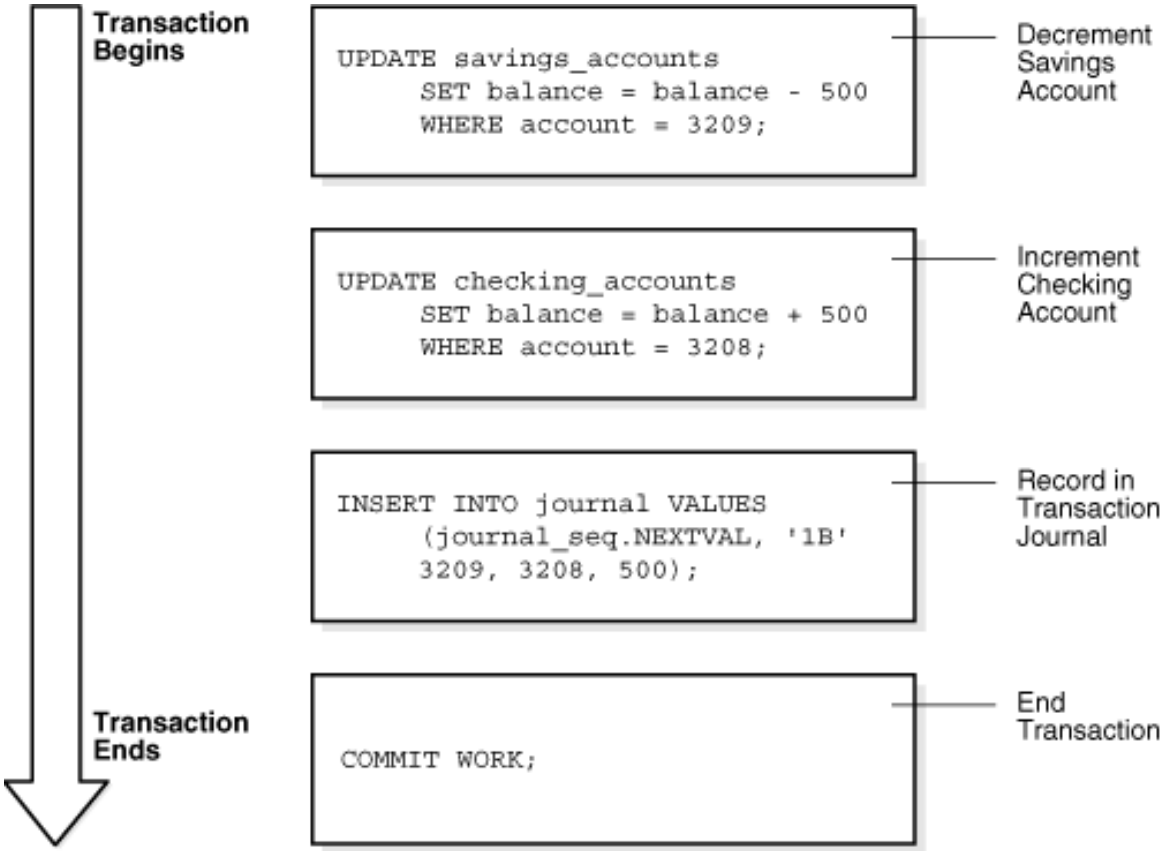
Autocommit transaction

```
BEGIN TRANSACTION
   UPDATE authors  SET au_fname = 'John'
    WHERE au_id = '172-32-1176'

   UPDATE authors SET au_fname = 'Marg'
    WHERE au_id = '213-46-8915'
COMMIT
END TRANSACTION
```

Primer transakcije u SQL-u

- BEGIN TRANSACTION



- END TRANSACTION

Transakcije na nivou DBMSa

- Transakcije se mogu posmatrati preko operacija za pristup i ažuriranje elemenata (objekata) baze podataka
  - **read_item** (kraće **read** ili **r**)
  - **write_item** (kraće **write** ili **w**)

Primer transakcije

- Transakcija T1

```
read_item(X);
X=X-N;
write_item(X);
read_item(Y);
Y=Y+N;
write_item(Y);
```

- Transakcija T2

```
read_item(X);
X=X+M;
write_item(X);
```

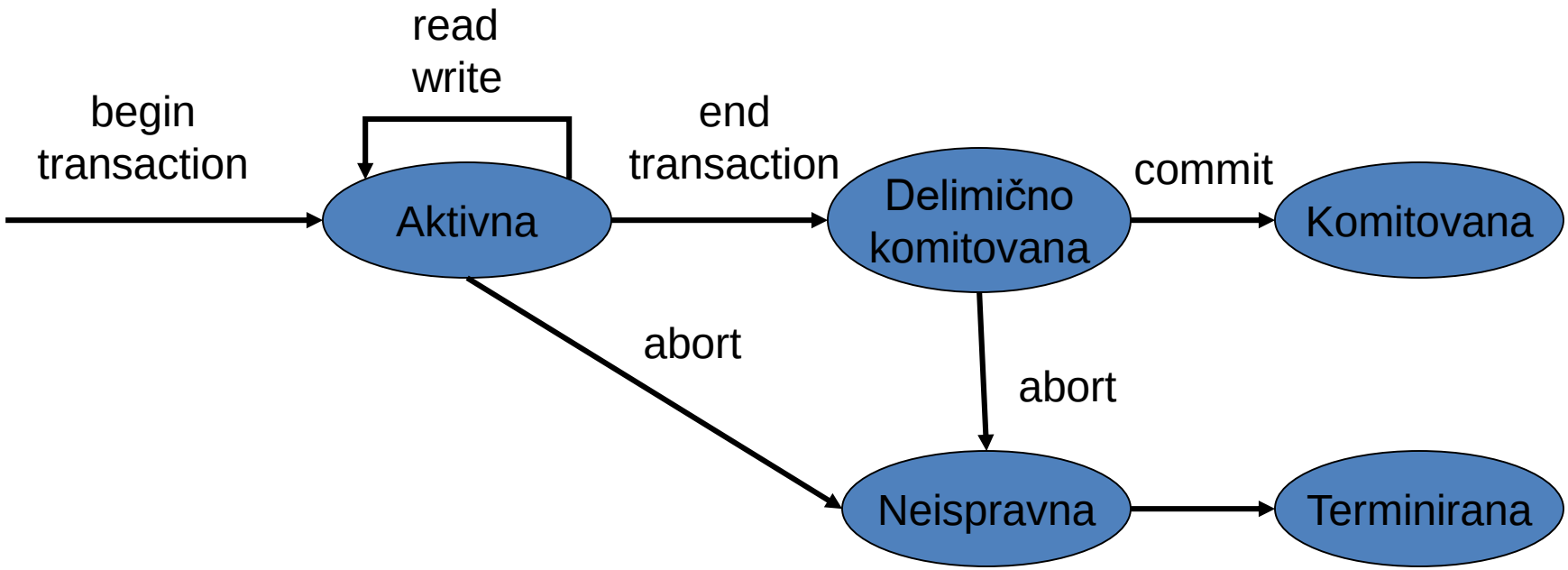
Operacije nad transakcijama

- **begin_transaction:** markira početak izvodjenja transakcije
- **read** ili **write:** specificira operacije čitanja i upisa elemenata baze podataka koje se izvršavaju kao deo transakcije
- **end_transaction:** označava da su završene *read* i *write* operacije iz transakcije i markira krajnju granicu izvršenja transakcije. U ovoj tački je neophodna provera da li su promene u bazi podataka trajno izvedene (tj. da li je transakcija **komitovana**) ili je izvedeno **abortovanje** transakcije iz nekog razloga

Operacije nad transakcijama

- **commit_transaction:** daje informaciju da je transakcija uspešno izvedena tako da su promene izvršene u okviru transakcije sigurno komitovane u bazu podataka i da neće biti poništene
- **rollback (abort):** signalizira da je transakcija završena neuspešno tako da treba ponoviti sve operacije ažuriranja koje su ovom transakcijom obuhvaćene
- **undo:** slična je operaciji rollback osim što se primenjuje na jednu operaciju, a ne na čitavu transakciju
- **redo:** ukazuje da određene operacije transakcije treba ponovo izvesti da bi osigurali uspešnost izvodjenja svih operacija komitovane transakcije nad bazom podataka

Dijagram stanja transakcije



Menadžer transakcija

- Menadžer transakcija forisira ACID svojstva
  - Određuje redosled izvršenja
  - Koristi COMMIT i ROLLBACK da obezbedi atomičnost
  - Katanci i timestamps se koriste da se obezbedi konzistentnost i izolacija za konkurentne transakcije
  - Održava Log za obezbeđivanje postojanosti kod pada sistema

Kontrola konkurencije

- **Kontrola konkurencije** je proces upravljanja simultanim (konkurentnim) operacijama nad bazom podataka da bi se onemogućila njihova međusobna interferencija
- DBMS prepliće akcije različitih transakcija da bi poboljšao performanse – povećao protočnost ili poboljšao vreme odgovora za kratke transakcije
  - Nisu sva preplitanja dozvoljena

Zašto je neophodna kontrola konkurencije?

- Mogu se javiti različiti problemi kada se transakcije izvršavaju konkurentno bez kontrole
- Karakteristični su sledeći problemi:
  - problem gubitka pri ažuriranju
  - problem privremenog ažuriranja
  - problem nekorektnog sabiranja
  - problem neponovljivog čitanja
- Primeri transakcija

T1	T2
read(X);	**read(X);**
X:=X-N;	**X:=X+M;**
write(X);	**write(X);**
read(Y);	
Y:=Y+N;	
write(Y);	

Problem gubitka pri ažuriranju

- Vrednost X je nekorektna jer T2 očitava vrednost X pre nego što T1 izmeni njenu vrednost u bazi podataka i stoga je izgubljena ažurirana vrednost X iz transakcije T1
- Primer:
  - Ulaz: X=5,N=2,M=1
  - Izlaz: X=6, tačno je X=4

T1	T2
read(X);	
X:=X-N;	
	read(X);
	X:=X+M;
write(X);	
read(Y);	
	write(X);
Y:=Y+N;	
write(Y);	

Problem privremenog ažuriranja ili prljavog čitanja

- Ako iz nekog razloga dođe do pada transakcije nakon ažuriranja nekog elementa (u našem slučaju elementa X), potrebno je taj element vratiti na staru vrednost
- Međutim, može se desiti da neka druga transakcija pristupi tom elementu pre nego što se element vrati na staru vrednost
- Ova druga transakcija čita privremenu vrednost tog elementa zbog čega se javlja neispravnost

T1	T2
read(X);	
X:=X-N;	
write(X);	
	read(X);
	X:=X+M;
	write(X);
read(Y);	
Pad transakcije	

Problem netačnog sumiranja

- T3 čita X posle oduzimanja N i čita Y pre dodavanja N, tako da je zbir neispravan
- U ovoj transakciji su nepravilno isprepletane operacije transakcija T1 koja ažurira elemente baze podataka X i Y i transakcije T3 koja čita te iste elemente

T1	T3
	suma:=0;
	read(A);
	suma:=suma+A;
read(X);	
X:=X-N;	
write(X);	
	read(X);
	suma:=suma+X;
	read(Y);
	suma:=suma+Y;
read(Y);	
Y:=Y+N;	
write(Y);	

Problem neponovljivog čitanja

- Javlja se kada transakcija T čita element podatka X dva puta
- Ako neka druga transakcija T' ažurira element podatka X između ova dva čitanja, T će dobiti dve različite vrednosti za isti element podatka X

Metode zaključavanja

- Za kontrolu konkurentnog pristupa podacima koristi se **zaključavanje**
- Kada transakcija pristupi bazi podataka koristi **katanac (lock)** da zabrani ostalim transakcijama pristup bazi podataka da bi se preventivno delovalo na pojavu nekorektnih rezultata
- Postoji nekoliko varijacija metode zaključavanja, ali osnove metode su zajedničke:
  - Transakcija mora postaviti katanac na element podatka pre operacije čitanja i upisa u bazu podataka
  - Katanac sprečava ostale transakcije da modifikuju ili čak i da čitaju taj element (ekskluzivni katanac)
  - Može se zaključati cela baza, jedan slog ili jedno polje u slogu
    - To je granularnost zaključavanja

Oporavak baze podataka

- **Oporavak baze podataka** je proces vraćanja (restauracije) baze podataka u korektno stanje nakon pojave neke neispravnosti

Zašto je neophodan oporavak baze podataka?

- Pri lansiranju neke transakcije DBMS mora obezbediti sledeće:
  - da se sve operacije transakcije uspešno izvedu i da se njihov efekat trajno zapiše u bazu podataka, ili
  - da transakcija ne utiče na bazu podataka, niti na bilo koju drugu transakciju
- DBMS mora osigurati da se ne desi da samo neke operacije transakcije imaju uticaj na bazu podataka
  - To se može desiti ako transakcija padne u toku njenog izvodjenja

Razlozi zbog kojih može doći do pada transakcije (1)

1. ***Pad računara:*** Za vreme izvođenja transakcije došlo je do greške u hardveru ili softveru. Pri tom može doši do gubitka sadržaja memorije
2. ***Greška u sistemu ili u transakciji.*** Neke operacije u transakciji mogu dovesti do pada sistema. Na primer, prekoračenje ili deljenje nulom, nedozvoljena vrednost nekog parametra, logička greška u programu ili operater mogu prekinuti izvršenje transakcije
3. ***Transakcija je detektovala lokalnu grešku ili uslov izuzeća***. U toku izvođenja transakcije se može desiti neki događaj koji zahteva otkazivanje transakcije. Na primer, nisu nađeni podaci neophodni za rad

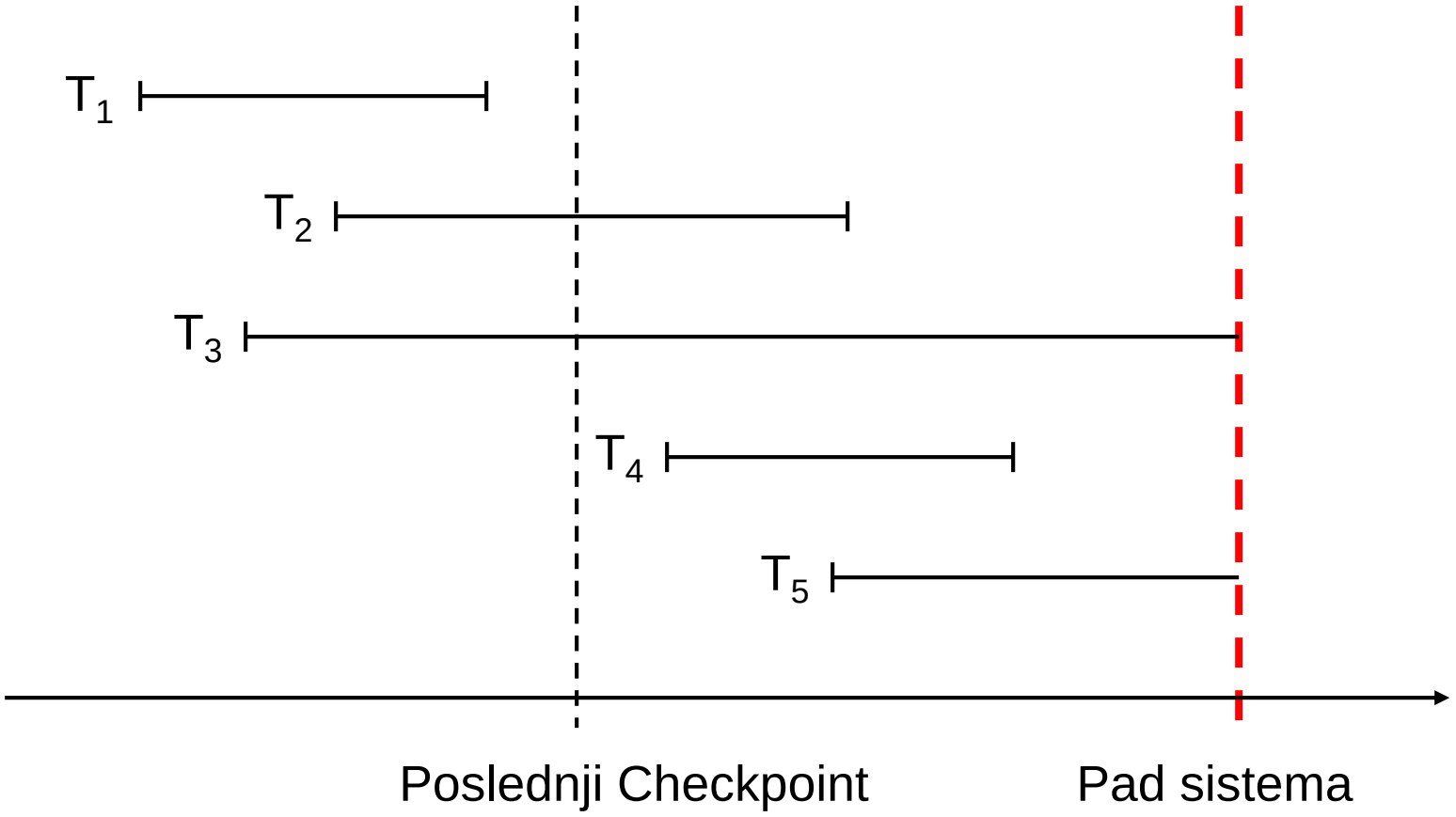
Razlozi zbog kojih može doći do pada transakcije (2)

4. ***Prisilna primena kontrole konkurencije.*** Metoda kontrole konkurencije može doneti odluku o abortiranju transakcije, naravno uz mogućnost njenog restartovanja.
5. ***Neispravnost diska***. Može doći do gubitaka podataka sa diska zbog njegove neispravnosti ili zbog problema u izvodjenju operacije čitanja sa diska ili zapisa na disk.
6. ***Fizički problemi i katastrofe.*** Može doći do fizičkog uništavanja medijuma ili do montiranja pogrešnog medijuma.

Transakcije i oporavak

- Transakcije su **osnovne jedinice oporavka** u sistemu baze podataka
- **Zadatak menadžera oporavka** je da garantuje 2 od 4 ACID svojstva transakcije – **atomičnost i postojanost**
- Menadžer oporavka treba da osigura, pri oporavku od nesipravnosti, da se **svi efekti** transakcije permanentno zabeleže u bazi podataka **ili nijedan** od njih
- Situacija je komplikovana jer upis u bazu podataka nije atomična akcija, pa je moguće da je transakcija izvela operaciju komitovanja, a da njen efekat još nije permanentno upisan u bazu podataka

Primer oporavka transakcija



Oporavak transakcije

UNDO i REDO: liste transakcija

UNDO = sve transakcije startovne na poslednjem checkpoint-u

REDO = prazna lista

Za svaku stavku u Log-u, počev od poslednjeg checkpoint-a

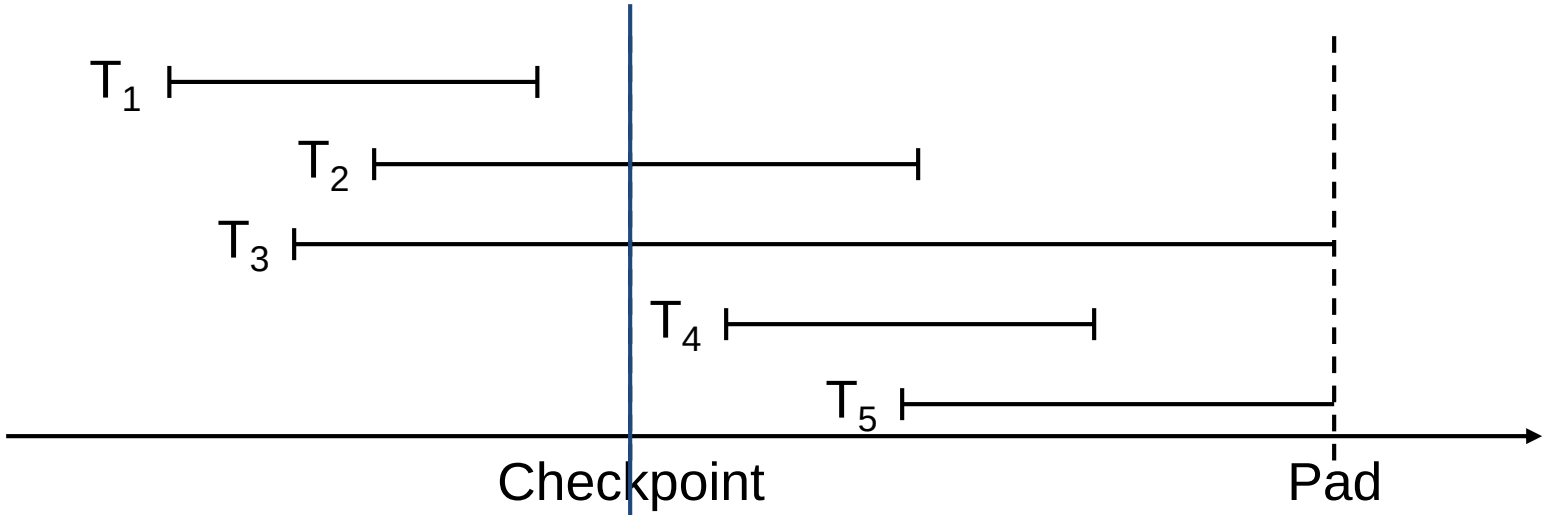
Ako nađen BEGIN TRANSACTION za T

Dodaj T u UNDO

Ako nađen COMMIT za T

Prebaci T iz UNDO u REDO

Oporavak transakcije



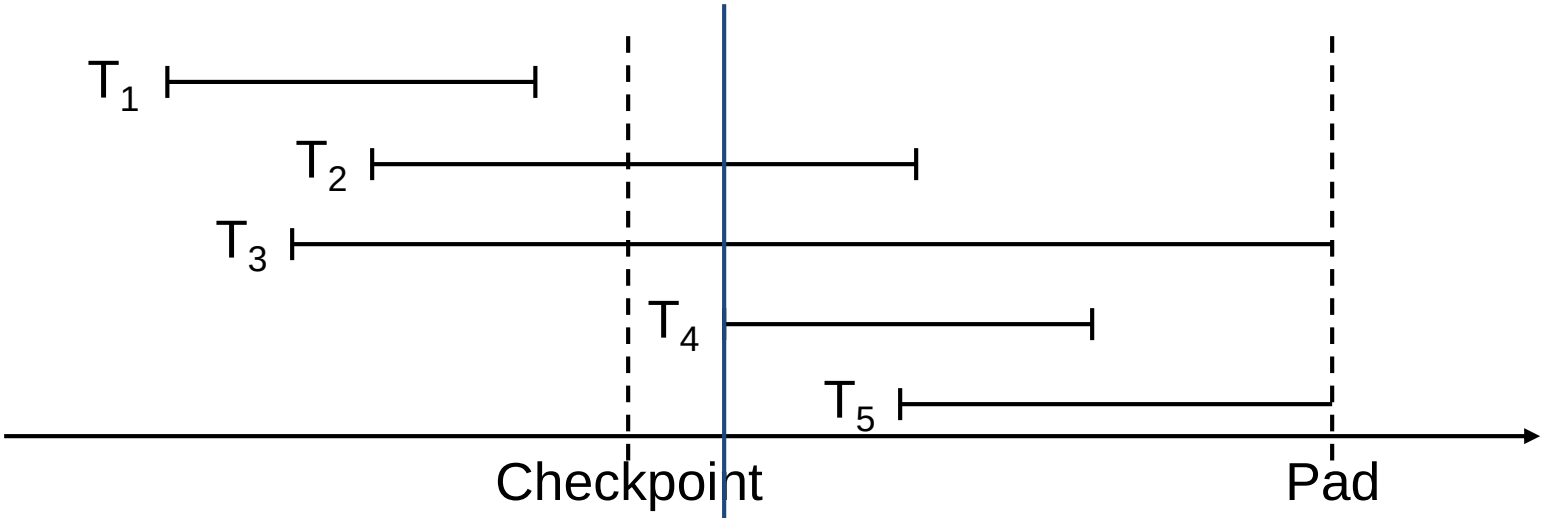
UNDO: T_2, T_3

REDO:

Poslednji Checkpoint

Aktivne transakcije: T_2, T_3

Oporavak transakcije



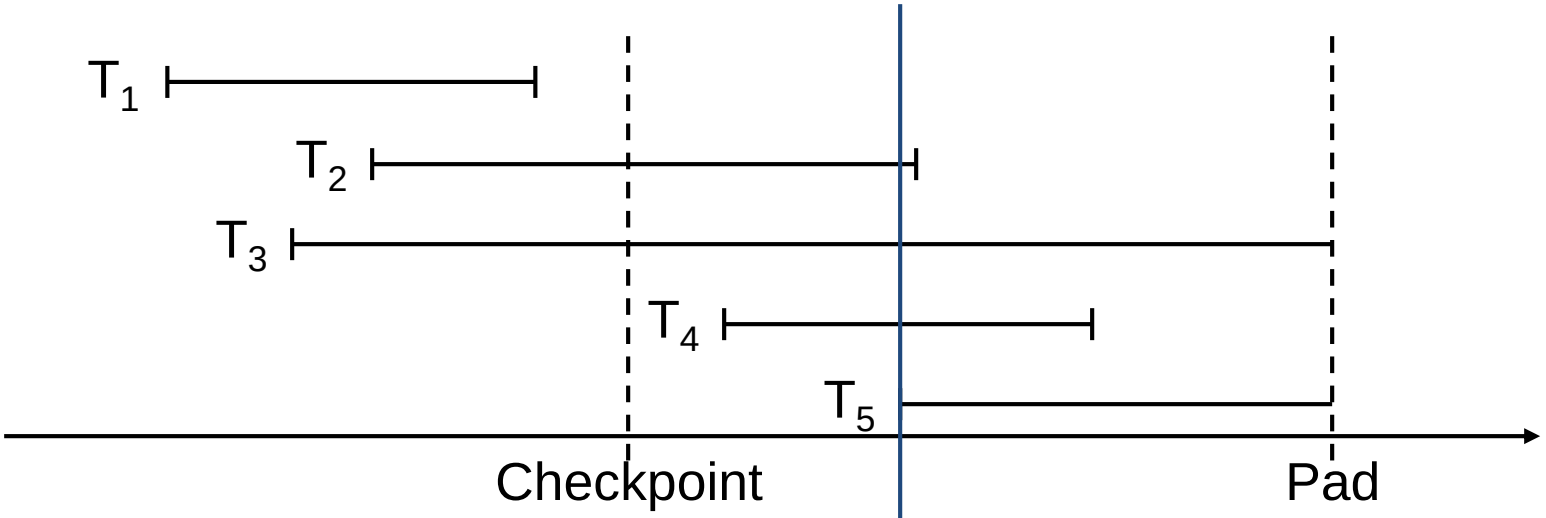
UNDO: T_2, T_3, T_4

REDO:

T4 počinje

Dodaj T_4 u UNDO

Oporavak transakcije



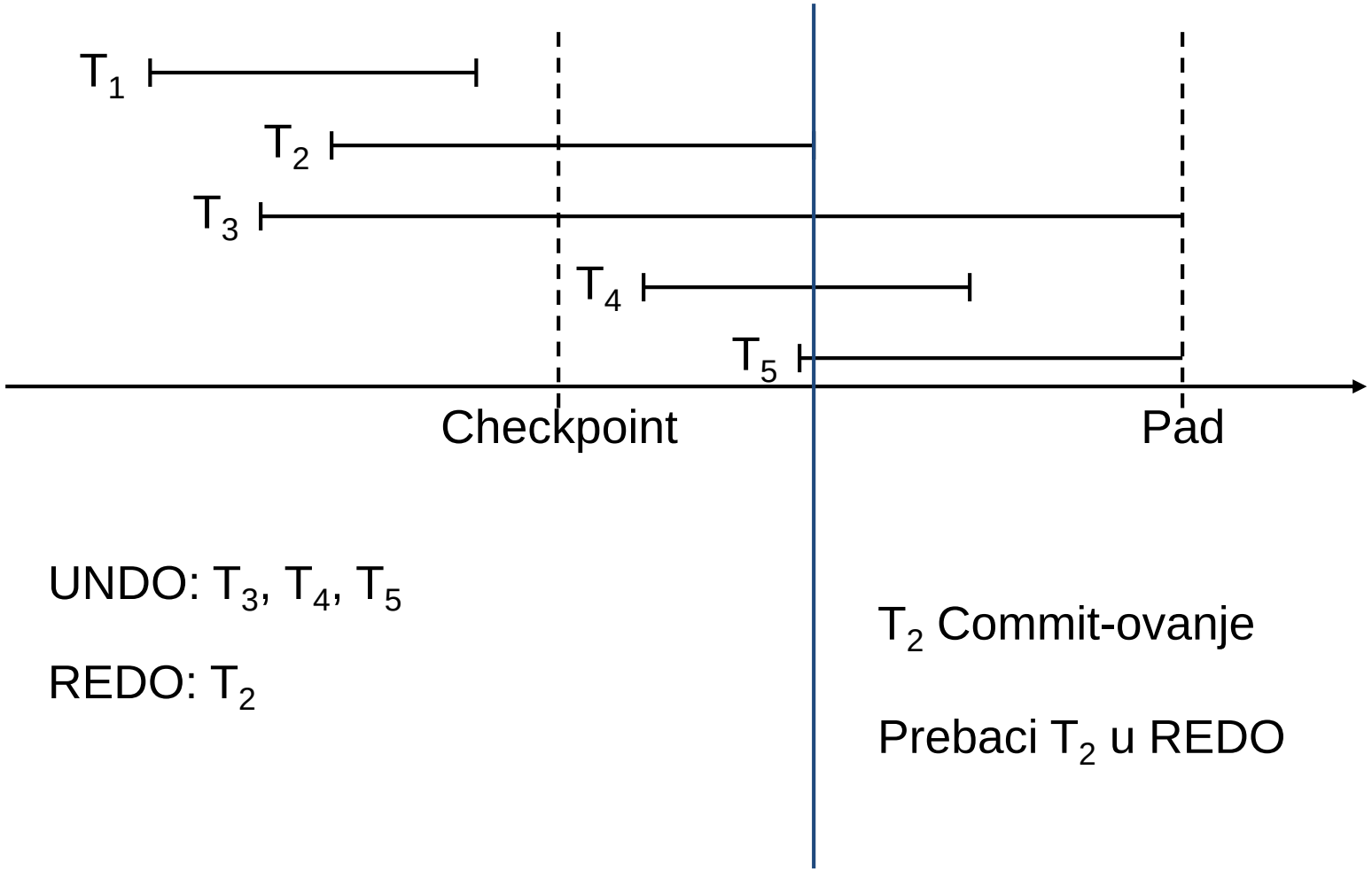
UNDO: T_2, T_3, T_4, T_5

REDO:

T_5 počinje

Dodaj T_5 u UNDO

Oporavak transakcije



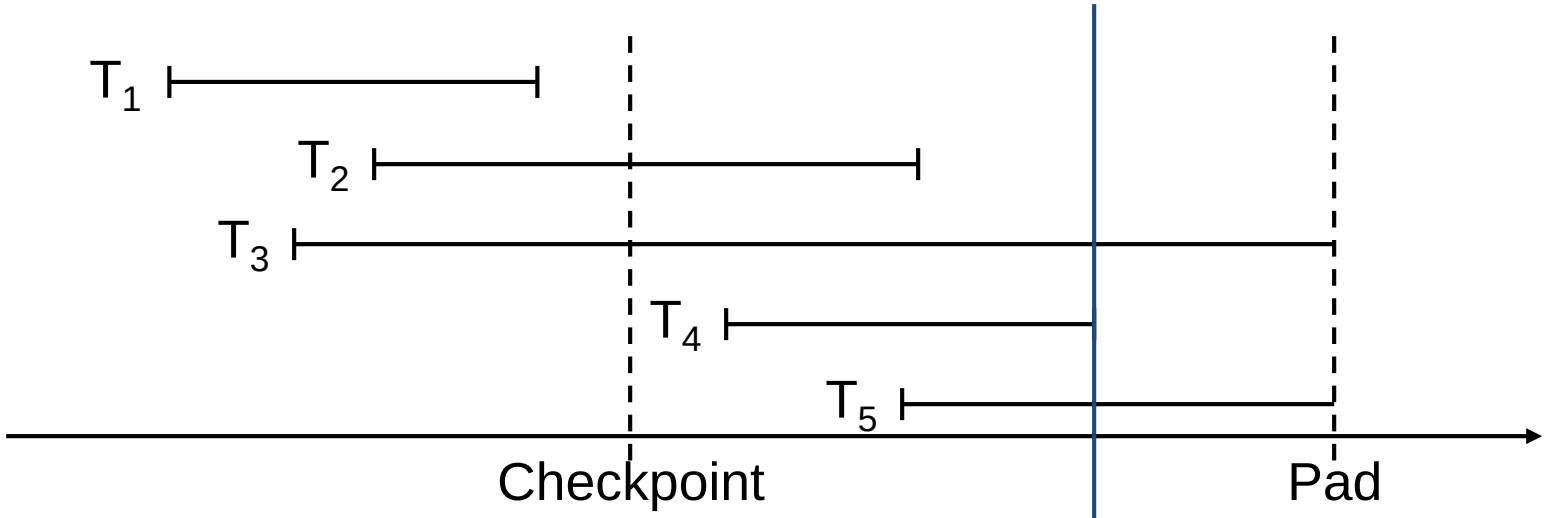
UNDO: T_3, T_4, T_5

REDO: T_2

T_2 Commit-ovanje

Prebaci T_2 u REDO

Oporavak transakcije



UNDO: T_3, T_5

REDO: T_2, T_4

T_4 Commit-ovanje

Prebaci T_4 u REDO

Transakciona obrada

Pitanja ???