



Poglavje 8

Obnavljanje podatkovne baze

Povzeto po [4]

Podatkovne nesreče – Uvod..

- SUPB mora med drugim uporabnikom zagotavljati tudi nemoten dostop do celovitih – neokrnjenih podatkov
- Dostop do podatkov in njihovo celovitost lahko ogrozijo dogodki, ki se imenujejo podatkovne nesreče
- Podatkovne nesreče lahko povzročijo nekonsistentnost podatkov med seboj ali s stvarnostjo, ali pa onemogočijo nadaljnji dostop do PB

Podatkovne nesreče – Uvod..

- Postopke, s katerimi se ohranja konsistentnost PB po nesrečah, imenujemo obnavljanje (recovery). Te postopke izvaja SUPB pod nadzorom skrbnika PB
- Obnavljanje obsega:
 - Pripravo podatkov za obnovitev
 - Detekcijo podatkovne nesreče, ki je povzročila nekonsistentno stanje ali nedostopnost podatkov
 - Obnovitev PB v stanje pred nesrečo

Podatkovne nesreče – Uvod..

- Na PB lahko gledamo kot na sistem, ki se sme nahajati le v veljavnih stanjih, v katerih so podatki v PB med seboj konsistentni
- Prehod iz veljavnega stanja v veljavno stanje se izvede z ažuriranjem podatkov: skupino operacij, ki poizvedujejo in spreminjajo podatke

Podatkovne nesreče – Uvod..

- Zaporedje ažuriranj, ki povzročijo prehod PB iz enega v drugo veljavno stanje, se imenuje transakcija
- Med izvajanjem transakcije se PB lahko nahaja tudi v neveljavnih stanjih
- V neveljavnem stanju lahko PB tudi obstane, če se med izvajanjem zgodi podatkovna nesreča, ker se v tem primeru transakcija ne konča

Podatkovne nesreče – Uvod..

- Obnavljanje PB mora zagotoviti, da se PB po nesreči obnovi v eno izmed veljavnih stanj:
 - to je lahko zadnje veljavno stanje ali
 - eno izmed prej veljavnih stanj
- V vseh veljavnih stanjih so podatki med seboj sicer konsistentni, vprašanje je, ali so podatki konsistentni s stvarnostjo
- S stvarnostjo so najbolj konsistentni podatki zadnjega veljavnega stanja

Podatkovne nesreče – Uvod..

- Obnavljanje ne obsega samo povrnitev PB v neko veljavno stanje, ampak tudi zagotavljanje podatkov, ki so potrebni za obnavljanje
- Postopki za zagotavljanje obnovitvenih podatkov se morajo izvajati z ažuriranjem PB, v okviru transakcij

Podatkovne nesreče – Uvod

- Posledično je to dodatna obremenitev sistema, kar pomeni slabše performanse
- Čim hitrejšo in čim kakovostnejšo obnovitev PB po nesreči želimo, tem več dodatnih postopkov je potrebno izvajati med transakcijami

Transakcije..

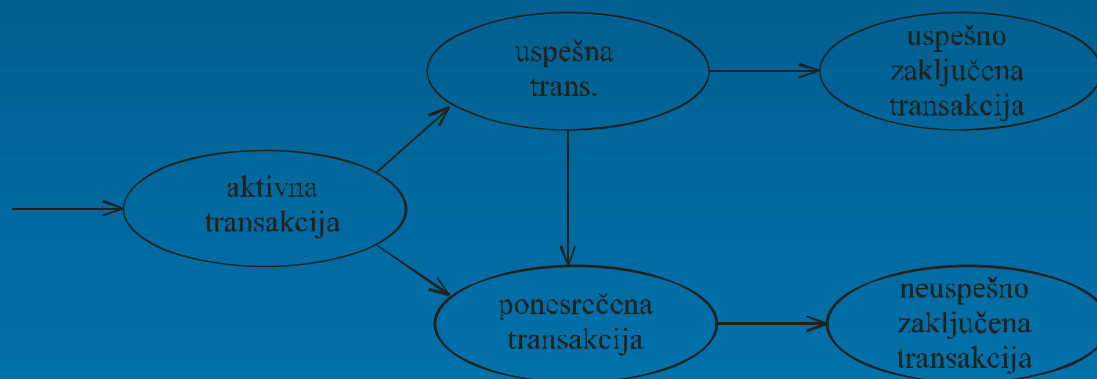
- Transakcija je logični vrstni red ažuriranj in povpraševanj, ki prevede PB iz enega v novo veljavno stanje. Transakcija ohranja konsistentnost podatkov v PB
- Če hočemo ohraniti konsistentno PB, se mora transakcija izvesti v celoti, ali pa sploh ne. Oziroma: izvesti se morajo vsa ažuriranja transakcije ali nobeno

Transakcije..

- Transakcija se prične z operacijo “*Začetek transakcije*”, konča pa ali z ukazom “*Pomni*” (COMMIT) ali “*Pozabi*” (CANCEL, ROLLBACK)
- Ukaz COMMIT: sporočilo SUPB, da so se vsa ažuriranja v okviru transakcije uspešno izvedla in da naj se vse spremembe v PB ohranijo
- Ukaz ROLLBACK: navodilo SUPB, naj se vsa v okviru transakcije izvedena ažuriranja razveljavijo

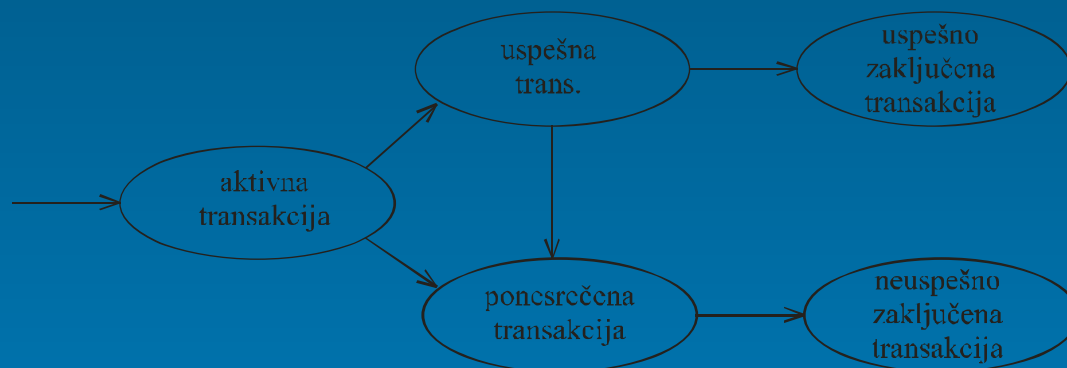
Transakcije..

- Življenski cikel transakcije lahko predstavimo z diagramom prehajanja stanj – transakcija se nahaja v več stanjih
- Transakcija je aktivna od ukaza “Začetek transakcije” do ukaza “Commit” (preide v stanje “uspešna transakcija”)



Transakcije..

- Pri nekaterih vrstah obnavljanja mora SUPB vsa ažuriranja uspešne transakcije šele uveljaviti v PB, ker se dejanska ažuriranja v PB še niso izvedla
- Ko je uveljavljanje ažuriranj izvedeno, postane transakcija uspešno zaključena (stanje “uspešno zaključena transakcija”)



Transakcije

- Če se med uveljavljanjem ažuriranj zgodi podatkovna nesreča, postane transakcija ponesrečena
- Tudi aktivna transakcija se lahko ponesreči in tako postane ponesrečena transakcija
- V takih primerih mora SUPB poskrbeti za razveljavitev vseh morebitnih ažuriranj v PB, nato pa postane transakcija “neuspešno zaključena”
- Če se transakcija ni ponesrečila po lastni krivdi (na primer zaradi neustreznih vhodnih podatkov), jo SUPB vrne v ponovno izvajanje

Vrste podatkovnih nesreč..

- Podatkovne nesreče se med seboj razlikujejo:
 - po vzrokih
 - po tem, kdo odkrije nesrečo
 - po posledicah, ki jih povzročé
- Od vrste podatkovnih nesreč so odvisni nadaljnji postopki obnavljanja
- Vrste podatkovnih nesreč:
 - transakcijske nesreče, ki jih odkrijejo uporabniški programi
 - transakcijske nesreče, ki jih odkrije SUPB
 - sistemske nesreče
 - diskovne nesreče

Vrste podatkovnih nesreč

- 1. in 2. skupina predstavljata nesreče lokalnega značaja – samo posamezna transakcija se ne zaključi uspešno
- 3. in 4. skupina predstavlja nesreče s širšimi posledicami

Transakcijske nesreče..

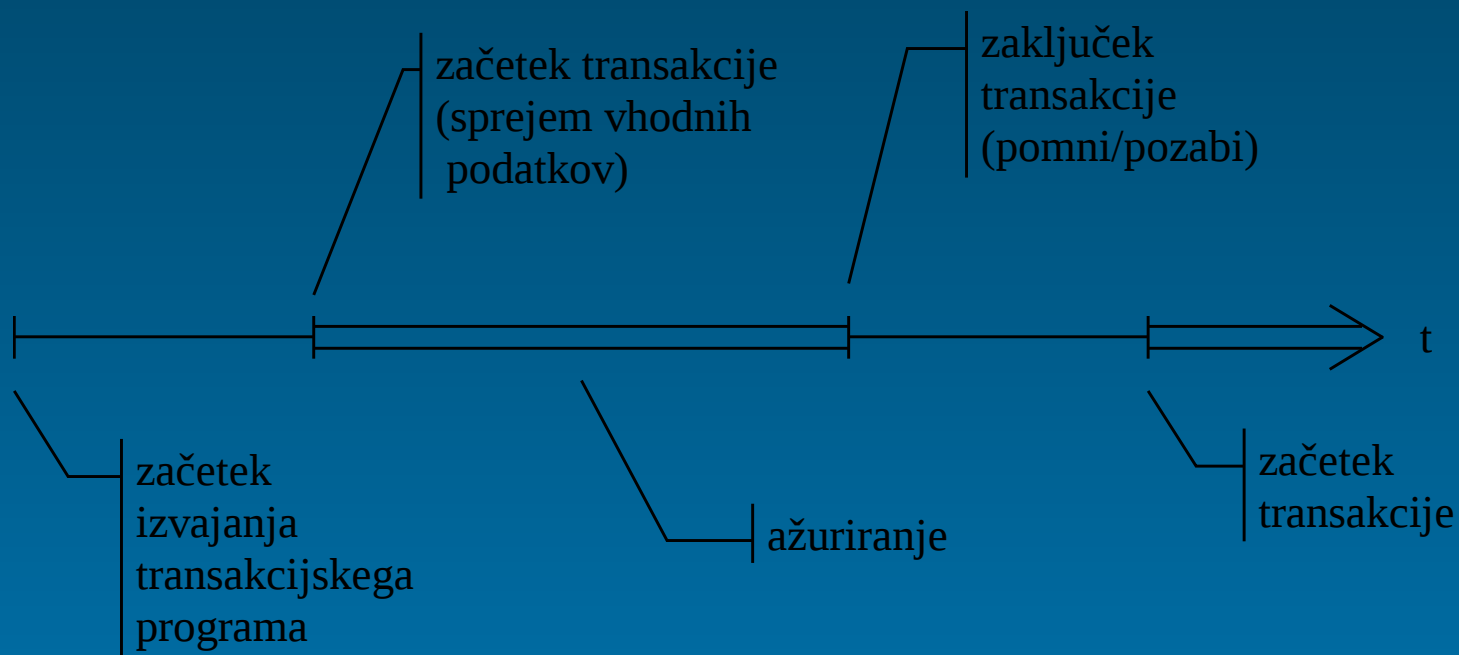
- Transakcijske nesreče, ki jih odkrijejo uporabniški programi:
 - Npr.: Med izvajanjem transakcije se lahko izkaže, da so vhodni podatki v protislovju s podatki v podatkovni bazi, in da transakcije ni mogoče uspešno izvesti. V takem primeru izda transakcijski program ukaz **Rollback** SUPB, katerega naloga je nato, da vsa že izvedena ažuriranja podatkovne baze v okviru transakcije razveljavi
- Pri izvajanju transakcije lahko pride do prekinitve izvajanja transakcijskega programa:
 - Razlog je lahko slabo napisan ali slabo preizkušen transakcijski program

Transakcijske nesreče..

- Transakcijo lahko prekine tudi proces za nadzor sočasnega izvajanja transakcij, če ugotovi, da se neka transakcija ne more sočasno izvajati z drugim, ker bi to ogrozilo konsistentnost PB. V tem primeru gre za transakcijsko nesrečo, ko jo odkrije SUPB.

Transakcijske nesreče

- Izvajanje transakcije:



Sistemske nesreče..

- Za sistemsko nesrečo štejemo izgubo podatkov v notranjem pomnilniku, zaradi prekinitve delovanja le-tega
- Napaka je lahko povzročena s:
 - prekinitvijo napajanja
 - napako pri branju ukaza ali podatka iz notranjega pomnilnika

Sistemske nesreče

- Sistemska nesreča povzroči prekinitev izvajanja trenutno aktivnih transakcij
- Pred nadaljnjo uporabo PB po sistemski nesreči, je potrebno obnoviti PB v zadnje veljavno stanje pred nesrečo in ponoviti izvajanje prekinjenih transakcij

Diskovne nesreče..

- Tudi podatki, ki se nahajajo na sekundarnem pomnilniku (trdi disk), se lahko uničijo ali izgubijo
- Razlogi za okvaro:
 - okvara diskovne površine
 - okvara bralno pisalnih glav
 - okvara krmilnika diska itd.
- Tudi pri teh nesrečah so lahko nekatere transakcije med svojim izvajanjem prekinjene

Diskovne nesreče

- Ko PB spet začne delovati, je potrebno obnoviti PB v zadnje veljavno stanje pred nesrečo in ponoviti izvajanje prekinjenih transakcij

Vrste obnavljanja PB – Uvod..

- Obnavljanje PB temelji na redundantnih podatkih
- Redundantni podatki se generirajo občasno ali sočasno z izvajanjem operacij v okviru transakcij

Vrste obnavljanja PB – Uvod..

- Podatki za obnavljanje:
 - kopija PB
 - vhodni transakcijski podatki, na osnovi katerih se izvajajo transakcije in oznaka transakcijskega programa
 - vrednosti zapisov v PB pred njihovim ažuriranjem (stare vrednosti zapisov) – omogoča razveljavitev ažuriranj neuspešno zaključenih transakcij
 - vrednosti zapisov v PB po njihovem ažuriranju (nove vrednosti zapisov) – omogoča ponovitev ažuriranj uspešno izvedenih transakcij, ki so se izgubila v podatkovni nesreči
- Za obnavljanje lahko uporabimo vse našete podatke ali le nekatere izmed njih

Vrste obnavljanja PB – Uvod..

- Obnavljanja se razlikujejo glede na:
 - hitrost obnovitve PB
 - pogostost in količino beleženja redundantnih podatkov
 - poslabšanja odzivnih časov, zaradi beleženja redundantnih podatkov
- Čim pogostejše in v čim popolnejšem obsegu se beležijo redundantni podatki, tem **hitrejša** je obnovitev PB

Vrste obnavljanja PB – Uvod

- S tem pa so povezani slabši odzivni časi SUPB in večji stroški zaradi dodatnega pomnilnega prostora
- Odločiti se je potrebno za nek pameten kompromis med odzivnostjo in stroški!!!

Dvojna podatkovna baza..

- V primeru vzdrževanja dvojne podatkovne baze obstaja več možnosti, kaj vse podvojiti:
 - najenostavnejši primer: diskovni krmilnik zapisuje sočasno podatke na 2 fizično ločena diska. Če se ena PB pokvari, se uporablja njena dvojnica
 - za večjo zanesljivost: podvoji se tudi diskovni krmilnik ali kar cel računalniški sistem
- Omenjena zaščita je relativno draga. Omogoča pa hitro obnavljanje PB predvsem po diskovnih in delno sistemskih podatkovnih nesrečah

Dvojna podatkovna baza..

- Ob tem je koristno uporabljati še kakšen drug način beleženja redundantnih podatkov, zaradi primera, ko postaneta nedostopni obe PB. Na primer: izpad elektrike

Dvojna podatkovna baza..

- Preprostejša varianta dvojne PB: kopija, kjer se vsebina PB periodično kopira na magnetni trak. Ali pa se PB delno ali v celoti “exportira” v datoteko na disk, ki je potem vir za obnovitev PB
- Kopija PB omogoča popolno obnovitev PB v primeru diskovnih nesreč, s čimer preide v veljavno stanje, v katerem se je nahajala v trenutku izdelave kopije

Dvojna podatkovna baza

- Med izdelavo kopije naj se transakcije ne bi izvajale, zato da se bo kopija PB zanesljivo nahajala v veljavnem stanju
- PB je možno tudi delno ali inkrementalno kopirati. To kopiranje se lahko izvaja tudi v času, ko je PB v uporabi

Obnavljanje s senčnimi stranmi..

- Obnavljanje s senčnimi stranmi je učinkovito predvsem pri transakcijskih podatkovnih nesrečah, ko je potrebno razveljaviti že izvedena ažuriranja PB
- Princip delovanja:
 - Strani notranjega pomnilnika, ki so bile ažurirane, se ne zapisujejo neposredno v PB, ampak na nezasedene bloke na disku
 - Če se transakcija zaključi uspešno, se omenjeni novi bloki na disku vključijo v PB, namesto starih neažuriranih
- Naslov vsake strani na disku je zapisan v indeksu strani

Obnavljanje s senčnimi stranmi..

- Obstajata dva indeksa strani:
 - tekoči indeks strani
 - senčni indeks strani



Obnavljanje s senčnimi stranmi..

- Na začetku transakcije sta oba indeksa enaka. Tekoči indeks se nahaja v notranjem pomnilniku, senčni pa na disku. Naslov senčnega indeksa na disku je zapisan v posebni startni datoteki
- Operaciji PreberiBlok(X) in IzpišiBlok(X) uporabljata za dostop do blokov na disku (povpraševanje, ažuriranje) tekoči indeks

Obnavljanje s senčnimi stranmi..



Obnavljanje s senčnimi stranmi..

- Ažurirni ukaz $Spremeni(X,x)$ se izvede v naslednjih korakih:
 - če se stran X s podatkom x še ne nahaja v notranjem pomnilniku, se izvede operacija $PreberiBlok(X)$
 - zapisu x , ki se nahaja na strani X notranjega pomnilnika, se vrednost ustrezno spremeni
 - če stran X z zapisom x še ni bila ažurirana (na disku), se na disku poišče še nerabljena stran in z njenim naslovom se v tekočem indeksu zamenja dosedanji naslov strani. Sicer ta nerabljena stran že obstaja za stran X

Obnavljanje s senčnimi stranmi..



Obnavljanje s senčnimi stranmi..

- V primeru transakcijske nesreče, ko se izda ukaz Pozabi, se tekoči indeks strani preprosto prepíše s senčnim indeksom in v PB ni sledu o ažuriranjih, ki so se v okviru transakcij izvedla
- Aktivirajo se senčne strani in transakcijski program lahko nadaljuje z izvajanjem naslednje transakcije

Obnavljanje s senčnimi stranmi..

- Ob uspešnem zaključku transakcije (ukaz Pomni), se izvede naslednje:
 1. vse ažurirane strani, ki se še nahajajo v pomnilniku, se izpišejo na disk,
 2. tekoči indeks strani se izpiše na prosto mesto na disk in to postane senčni indeks,
 3. naslov novega senčnega indeksa na disku se vpiše v startno datoteko.

Obnavljanje s senčnimi stranmi

- Če bi se med izvajanjem transakcije ali korakov 1-3 izvedla sistemska nesreča, bi SUPB ob ponovnem zagonu iz startne datoteke prebral naslov senčnega indeksa in na ta način prekinjena transakcija ne bi v PB pustila nobenih sledi
- Obnavljanje s senčnimi stranmi ščiti PB pred transakcijskimi in sistemskimi nesrečami
- Za zaščito pred diskovnimi nesrečami, je potrebno poskrbeti dodatno

Obnavljanje z dnevnikom in kopijo..

- To obnavljanje temelji na izdelavi kopije PB in izdelavi dnevnika:
 - s kopijo lahko PB obnovimo v veljavno stanje, ko se je izdelala kopija
 - v dnevnik se zapisujejo podatki, s katerimi je možno s kopijo obnovljeno PB obnoviti v zadnje veljavno stanje tik pred nesrečo
- Dnevnik vsebuje tudi podatke, s katerimi je možno ponovno izvesti tudi transakcije, ki so bile zaradi nesreče prekinjene

Obnavljanje z dnevnikom in kopijo..

- 2 vrsti obnavljanja PB z dnevnikom glede na čas izvedbe ažuriranja PB:
 - odloženo ažuriranje
 - sprotno ažuriranje
- **Odloženo ažuriranje:** vsa ažuriranja v okviru transakcije se najprej shranijo v dnevnik. Pri uspešnem zaključku transakcije se izvede dejansko ažuriranje PB
- **Sprotno ažuriranje:** Vsa ažuriranja se izvajajo v PB. V dnevnik se vpisujejo le podatki, ki so potrebni za morebitno obnavljanje PB

Obnavljanje z dnevnikom in kopijo

- Performanse:
 - Odloženo ažuriranje je učinkovitejše, če se v povprečju izvede več neuspešnih transakcij, ker v tem primeru ni treba spreminjati PB
 - Sprotno ažuriranje je učinkovitejše, če se v povprečju izvede več uspešnih transakcij

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- V dnevnik se beležijo:
 - zapisi, ki vsebujejo podatke, potrebne za izvedbo odloženega ažuriranja PB in
 - podatki za njeno morebitno obnavljanje
- Dnevnik je zaporedna datoteka, v katerem so zapisi urejeni po času njihovega nastanka
- Zapisi se v dnevnik vedno dodajajo na konec datoteke

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- Vsak zapis v dnevniku je opremljen:
 - z enolično oznako transakcije - Ti
 - s časom generiranja zapisa - t
- Zapis ob začetku izvajanja transakcije vsebuje:
 - oznako “Začetek”
 - oznako transakcijskega programa Pj
 - za vsak vhodni zapis pa par: (ImeZapisa, VrednostZapisa)

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- Zapis ob ukazu za ažuriranje vsebuje:
 - oznako NovaVrednost
 - vrsto operacije: Dodaj, Izbriši, Spremeni
 - par (NaslovZapisa, NovaVrednostZapisa)
- Zapis ob prehodu transakcije v stanje Uspešna transakcija:
 - oznako Pomni
- Zapis ob prehodu transakcije v stanje Neuspešna transakcija:
 - oznako Pozabi

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- Ker se sočasno lahko izvaja več transakcij so dnevniški zapisi teh transakcij lahko med seboj pomešani
- Ko se transakcija prične izvajati, SUPB v dnevnik zapiše zapis:
Dodaj(D, <Ti, t, Začetek, Pj, (V1, v1), ... (Vn, vn)>)

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- Ob vsakem ažuriranju, ki se pojavi v okviru transakcije, se v dnevnik doda naslednji zapis:

Dodaj(D, <Ti, t, NovaVrednost, vrsta-operacije, (X, x)>)

- Ob neuspešnem zaključku transakcije (ukaz Pozabi) se v dnevnik doda:

Dodaj(D, <Ti, t, Pozabi>)

in izbriše transakcijo iz liste aktivnih transakcij.

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje..

- Ob uspešnem zaključku trans. se v dnevnik doda zapis:

`Dodaj(D, <Ti, t, Pomni>)`

poleg tega pa se izvede tudi uveljavitev sprememb v PB

- Uveljavitev sprememb v PB se izvede tako, da se bere dnevniške zapise z oznako NovaVrednost za transakcijo Ti (po času naprej), ter se izvedejo v njih zapisani ukazi operacij nad PB:

`PoiščiPreberi(D, <Ti, t, NovaVrednost, vrsta-operacije, (X, x)>)`

Obnavljanje z dnevnikom in kopijo - Odloženo ažuriranje

- Ko so izvedene vse operacije za T_i , ki so zapisane v dnevniku, SUPB zbršiše T_i iz liste aktivnih transakcij

Obnavljanje z dnevnikom in kopijo - Sprotno ažuriranje..

- V dnevnik se beležijo enake vrednosti, kot pri odloženem ažuriranju, dodatno pa še zapisi z vrednostmi pred ažuriranjem (stare vrednosti)
- Zapisovanje ob začetku izvajanja transakcije, njenem uspešnem ali neuspešnem zaključku, je enako kot pri odloženem ažuriranju

Obnavljanje z dnevnikom in kopijo - Sprotno ažuriranje..

- Spremenjen je postopek pri ukazih za ažuriranje. Ob vsakem ukazu za ažuriranje se v dnevnik najprej doda zapis s staro vrednostjo:

Dodaj(D, <Ti, t, StaraVrednost, vrsta-operacije, (X, x)>)

- Nato se izvede ažuriranje podatkovne baze (spreminjanje, brisanje, dodajanje)

- In nato še vpis v dnevnik:

Dodaj(D, <Ti, t NovaVrednost, vrsta-operacije, (X, x)>)

Obnavljanje z dnevnikom in kopijo - Sprotno ažuriranje

- Ob uspešnem zaključku transakcije se v dnevnik doda zapis Pomni, transakcija pa se izbriše iz liste aktivnih transakcij
- Pri neuspešnem zaključku se v dnevnik najprej doda zapis Pozabi, nato pa je ažuriranja v bazi potrebno razveljaviti
- Po razveljavitvi se transakcija izbriše iz liste aktivnih transakcij

Obnavljanje z dnevnikom in kopijo - Razveljavitev transakcije..

- Pri sprotnem ažuriranju se vse spremembe zapisov sproti vpisujejo v podatkovno bazo
- V primeru transakcijske ali systemske nesreče, je za nadaljevanje dela najprej potrebno odstraniti iz PB vsa ažuriranja oz. nastale spremembe, ki so nastale s strani prekinjenih transakcij. Pravimo, da SUPB v izvede razveljavitev

Obnavljanje z dnevnikom in kopijo - Razveljavitev transakcije..

- Pri sprotnem ažuriranju se razveljavitev izvede tako, da se iz dnevnika bere stare zapise (StaraVrednost) od zadnjega proti prvemu po padajočem času t:

PoiščiPreberi(D, <Ti, t, StaraVrednost, vrsta-operacije,
(X, x)>)

Obnavljanje z dnevnikom in kopijo - Razveljavitev transakcije..

- Za vsak tak prebrani dnevniški zapis se izvede ažuriranje PB:
 - Če je bila operacija vrste Spremeni, se zapisu X priredi njegova stara vrednost: Spremeni (X,x=StaraVrednost)
 - Če je bila operacije vrste Dodaj, se dodani zapis izbriše: Izbriši (X)
 - Če je bila operacija vrste Izbriši, se izbrisani zapis doda nazaj v bazo: Dodaj (X,x=StaraVrednost)

Obnavljanje z dnevnikom in kopijo - Razveljavitev transakcije

- V primeru sistemske nesreče je potrebno razveljaviti ažuriranja večih transakcij
- V dnevniku so njihovi zapisi med seboj pomešani, vendar pa so urejeni po času, kar omogoča izvesti razveljavitev

Obnavljanje z dnevnikom in kopijo - Ponovitev ažuriranj transakcije

- Ažuriranja, ki jih izvede uspešna transakcija, je v primeru sistemskih in diskovnih podatkovnih nesreč treba ponoviti
- Vzroki:
 - ker so se ažuriranja izgubila ali pa
 - ni zanesljivo, da so se vsi datotečni vmesniki z ažuriranimi zapisi tudi izpisali na disk (sistemske nesreče)
- Po izvedeni ponovitvi se v PB zanesljivo nahajajo vsa ažuriranja, ki so jih izvedle uspešne transakcije

Obnavljanje z dnevnikom in kopijo - Obnavljanje po sistemskih in diskovnih nesrečah..

- Pri sistemskih podatkovnih nesrečah se prekine izvajanje trenutno aktivnih transakcij
- Ohrani se PB na disku, izgubijo pa se podatki v notranjem pomnilniku
- Po ponovnem zagonu je PB treba obnoviti, tako da najprej razdelimo transakcije na:
 - **prekinjene transakcije:** v dnevniku obstajajo njihovi zapisi
 - **uspešne transakcije:** tiste, za katere obstaja v dnevniku zapis z oznako “Pomni” in
 - **neuspešne transakcije:** tiste, za katere obstaja v dnevniku zapis z oznako “Pozabi”

Obnavljanje z dnevnikom in kopijo - Obnavljanje po sistemskih in diskovnih nesrečah

- Obnavljanje uspešnih transakcij:
 - Pri njih nismo prepričani, ali so se strani, ki so jih transakcije ažurirale, zanesljivo zapisale na disk
 - Zaradi tega je potrebno njihova ažuriranja ponoviti, tako pri uporabi sprotnega, kot odloženega ažuriranja
- Obnavljanje neuspešnih transakcij:
 - Pri teh transakcijah pri sprotnem ažuriranju nismo prepričani, ali so se njihova ažuriranja tudi že zares razveljavila v PB
 - Take transakcije je zato potrebno ponovno razveljaviti
- Obnavljanje prekinjenih transakcij:
 - Najprej izvedemo njihovo razveljavitev, nato jih vrnemo transakcijskim programom v ponovno izvajanje

Obnavljanje z dnevnikom in kopijo - Kontrolna točka..

- Pri sistemskih nesrečah se pojavi vprašanje: Za koliko časa nazaj je potrebno uspešne transakcije ponoviti in neuspešne razveljaviti?
- Odgovor: Ponoviti oz. razveljaviti je potrebno vse transakcije, katerih zapise najdemo v dnevniku

Obnavljanje z dnevnikom in kopijo - Kontrolna točka..

- **PROBLEM:** Dnevnik se začne pisati od trenutka, ko je bila izdelana kopija PB. To pomeni, da je v določenih primerih potrebno obnoviti in razveljaviti veliko število transakcij, kar pomeni dolgotrajno obnavljanje!!!
- **REŠITEV:** Čas obnavljanja skrajšamo z zahtevo po izpisu vseh datotečnih vmesnikov na disk. Tako smo prepričani, da so bile transakcije, ki so bile zaključene pred izpisom vmesnikov, zanesljivo uveljavljene ali razveljavljene v PB na disku
- Temu postopku pravimo Kontrolna točka

Obnavljanje z dnevnikom in kopijo - Kontrolna točka..

- Kontrolna točka se izvede v odvisnosti od števila transakcij na časovno enoto
- Ob sistemski nesreči je potrebno razveljaviti oz. ponoviti le transakcije, ki so bile aktivne v času izdelave kontrolne točke, ali so se pričele izvajati kasneje

Obnavljanje z dnevnikom in kopijo - Kontrolna točka

- Pri kontrolni točki se izvedejo naslednje operacije:
 1. prekine se izvajanje novih ukazov transakcijskih programov, dokončajo se vse razveljavitve in uveljavitve transakcij,
 2. izvede se izsiljeni izpis vseh datotečnih vmesnikov dnevnika na disk,
 3. izvede se izsiljeni izpis vseh datotečnih vmesnikov PB na disk,
 4. v dnevnik se doda zapis “Kontrolna točka” in izvede izsiljen izpis datotečnega vmesnika dnevnika na disk,
 5. v startno datoteko PB se zapiše naslov zapisa “Kontrolna točka” v dnevniku,
 6. nadaljuje se izvajanje ukazov transakcijskih programov.

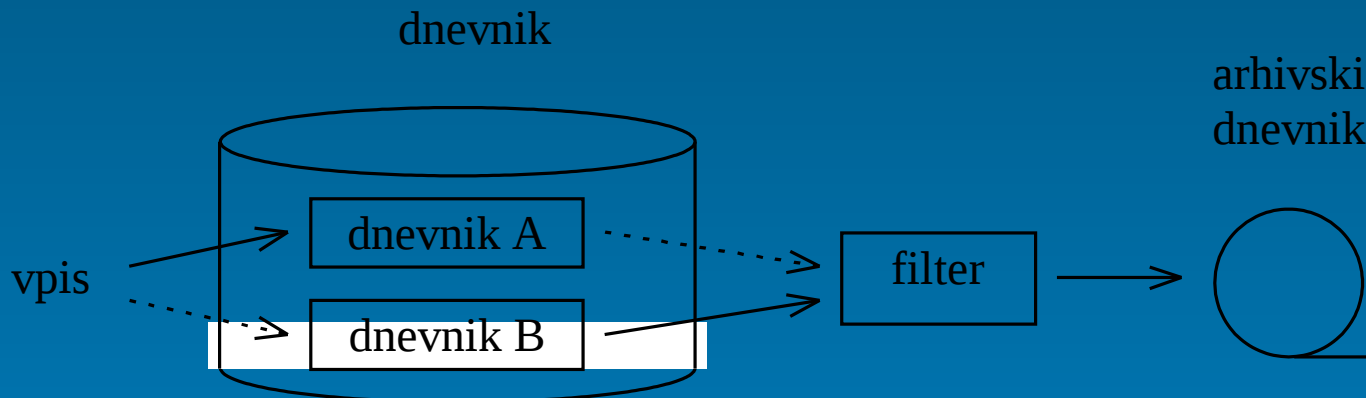
Obnavljanje z dnevnikom in kopijo – Dnevnik..

- Dnevnik in kopija PB predstavljata redundantne podatke, ki omogočajo obnovitev po sistemskih in diskovnih nesrečah
- Kopija in dnevnik se ne shranjujeta v fizično istem zunanjem pomnilniku, kot PB. S tem se zmanjša možnost uničenja vseh podatkov
- Dnevnik se uporablja tudi za tekoče razveljavljanje (sprotno ažuriranje) oz. uveljavljanje (odloženo ažuriranje). To mora biti izvedeno v čimkrajšem času

Obnavljanje z dnevnikom in kopijo – Dnevnik..

- Najpogostejša varianta realizacije dnevnika:

- dve dnevniški datoteki na disku in ena na magnetnem traku – arhivska
- Ob izvajanju transakcij se dnevniški zapisi vpisujejo v eno izmed datotek na disku (npr.: v dnevnik A)
- Ko se dnevnik A napolni približno 90%, se izvrši preklon vpisovanja
- Vsi dnevniški vpisi, ki se prično po preklopu, se vpisujejo v dnevnik B, zapisi transakcij, ki so se začele pred preklopom, pa se še naprej zapisujejo v dnevnik A.



Obnavljanje z dnevnikom in kopijo – Dnevnik

- Ko se zaključijo vse transakcije, ki se vpisujejo v dnevnik A, se izvede kontrolna točka. S tem so se vse transakcije v A zaključile pred kontrolno točko
- Edina nesreča, pri kateri bi še potrebovali dnevniške zapise iz dnevnika A, je diskovna nesreča
- Ker se pri ponovitvi transakcijskih ažuriranj rabijo le zapisi z oznako “NovaVrednost”, se le ti pri arhiviranju dnevnika A prepisejo v arhivski dnevnik. Temu pravimo filtriranje
- Arhivski dnevnik vsebuje le še tiste zapise, ki bi jih utegnili rabiti za obnavljanje, kar dosti zmanjša majhna velikost arhivskega dnevnika
- Ko se napolni dnevnik B, se postopek preklopa in arhiviranja ponovi