

IDA BAZA

KAJ JE IDA-BAZA?

BAZA je popoln sistem za upravljanje podatkovnih baz, usklajen z ostalimi izdelki IDA. BAZA združuje prednosti mnogih tovrstnih tujih proizvodov in domačih rešitev. Je vmesni člen med rizičnimi podatki na diskih in logičnimi slikami podatkov, kot jih vidijo uporabniki. BAZA je zasnovana tako, da poučarja centralno upravljanje z bazo podatkov, s tem, da so podatkovne strukture, dostop do podatkov in njihova zaščita definirani ločeno od programov. Uporabniki se lahko v celoti posvetijo problemskemu programiranju, medtem ko je upravljanje logične in fizične strukture ter zaščita podatkov prepuščena v fazi snovanja upravljalcu baze podatkov (data base administrator), v fazi izvajanja aplikacij pa BAZI. S tem se olajša delo programerjev in povečuje zanesljivost delovanja in varnost podatkov.

BAZO tvorijo *monitorji* in storitveni programi, ki omogočajo:

- definiranje struktur baze podatkov (DD),
- upravljanje in kreiranje podatkovnih struktur (DBF),
- realizacijo programskega dostopa do baze podatkov (DML),
- izvajanje V/I operacij, sprožanja in ustavljanja jedra sistema
- beleženje sprememb v fazi na nivoju logične transakcije in funkcionalno beleženje sprememb (logging), ki omogočajo restavriranje baze podatkov (recovery) v primeru nepredvidenih in nenormalnih prekinitev dela računalnika,
- beleženje uporabnikovih vhodnih transakcij (input logging), ki omogoča ponavljanje določene obdelave v paketnem režimu.

BAZA je kot ostali izdelki IDA prilagojena operacijskemu sistemu

DELTA/V, kot tudi DELTA/M. S stališča uporabnika je uporaba BAZE transparentna glede na operacijski sistem. Posebna skrb je posvečena izkoriščanju sistemskih resursov. Pomnilniški prostor se uporablja dinamično z možnostjo naslavljanja do 1 MB na operacijskem sistemu DELTA/M in 4 MB na DELTA/V. To zagotavlja bazi visoko učinkovitost in širok spekter uporabe tako za poslovne in tehnične, kot tudi za aplikacije v realnem času. BAZA je oblikovana tako, da omogoča sočasen dostop do baze podatkov 32 uporabnikom na DELTA/M in 100 uporabnikom na operacijskem sistemu DELTA/V. Možnost rezervacij na nivoju posameznih zapisov preprečuje motnje med uporabniki. Avtomatsko sproščanje zapisov po določenem času, če zapis potrebuje še kak drug program, preprečuje »dead lock« situacije.

ENOSTAVEN PROGRAMSKI DOSTOP DO PODATKOV

Jezik za manipuliranje z bazo podatkov (DML) je izveden s stavki za klic - CALL. To pomeni, da lahko kot gostiteljski jezik, v katerem pišemo uporabniške programe, uporabljamo praktično katerikoli višji programski jezik. BAZA definira štiri osnovne tipe programskih operacij podatkov v bazi:

- branje GET (GETD, GETP, GETR)
- dodajanje INSERT (INSA, INSB, INSG)
- spreminjanje REWRITE (RWRG)
- brisanje DELETE (DELG)

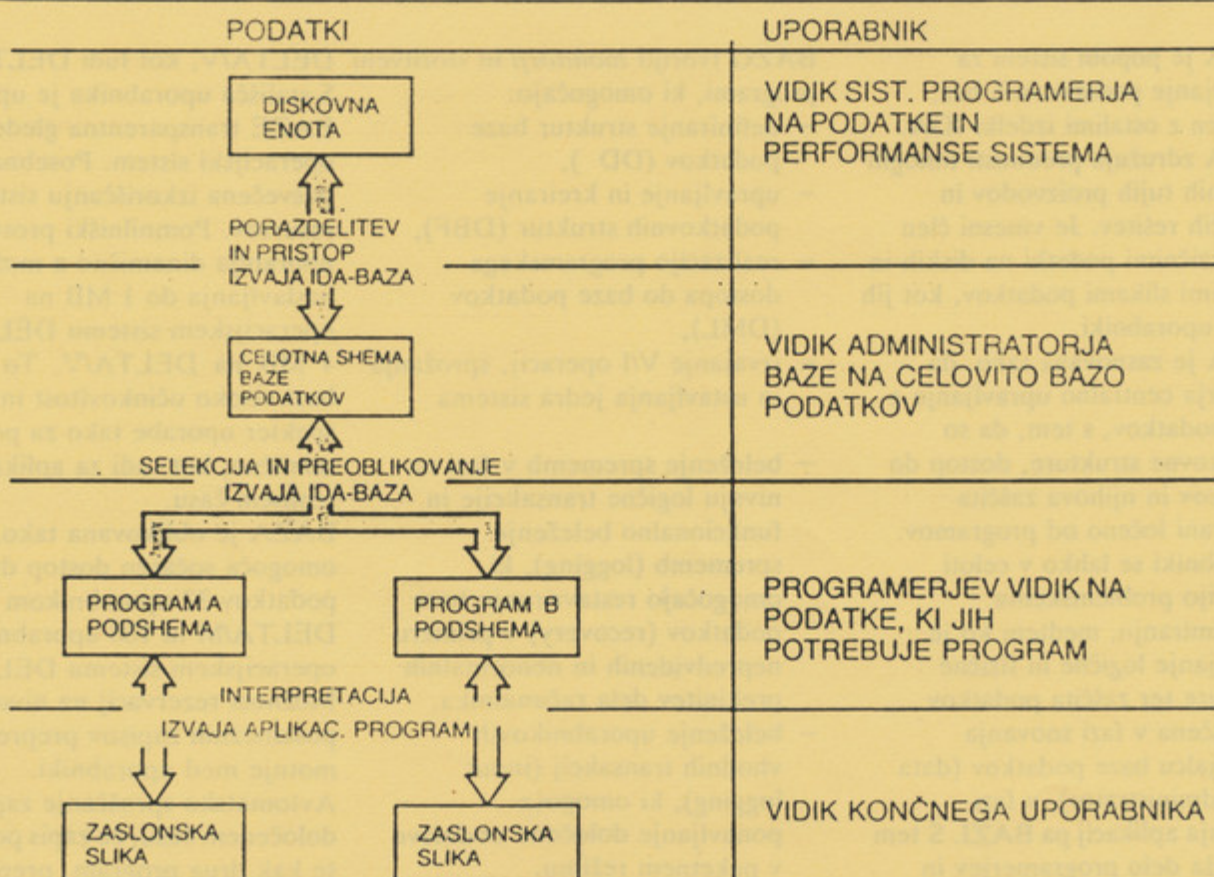
DML je pravzaprav kot podprogram z več vstopnimi

točkami, ki ob klicu izvede kontrolo parametrov in poskrbi za nadaljnjo izvedbo zahtevane akcije. Komunikacija med uporabniškim programom in BAZO poteka preko internih registrov in področij, ki so navedena v listi parametrov in katerih programski opis je eden od izhodov programa DDC. Ti opisi se lahko vključujejo v DATA DIVISION

uporabniškega programa v izvorni kobilski kodi. Ko se kontrola vrne uporabniškemu programu, moramo testirati DB STATUS register, da ugotovimo, ali je bila zahtevana akcija uspešna ali ne, in če ne, zakaj ne. Z ozirom na vzrok neuspešnosti akcije so lahko v uporabniškem programu

predvideni ustrezni ukrepi.

Kot prvi klic mora biti izveden klic z vstopno točko HELLO, v katerem sporočimo BAZI, katera polja bomo uporabljali kot interne registre, s katero podshemo bo program delal in seveda geslo. S tem se vzpostavi komunikacija med BAZO in programom. Izstopna točka BYE služi za zaključek dela s trenutno aktivno podshemo (zapiranje nizov). Vse spremembe na bazi se fizično ne realizirajo takoj, pač pa jih je treba potrditi z vstopno točko CONFIRM ali pa jih razveljaviti z vstopno točko CANCEL. Ta lastnost BAZE je tesno povezana z njeno sposobnostjo restavriranja na nivoju transakcije.



INTEGRITETA BAZE PODATKOV

Posebna skrb je posvečena ohranjanju integritete baze podatkov. V normalnem delovanju se skrb za integriteto odraža v možnosti določitve načina uporabe baze podatkov.

Tako se na nivoju podsheme definira način uporabe zapisov, ki je lahko SKUPEN (shared) ali PRIVILEGIRAN (privileged) in določa, ali lahko tudi duge podsheme istočasno uporabljajo iste zapise ali ne. Za vsak zapis podsheme določimo tip pristopa, ki je lahko:

- GETP – dovoljeno je le čitanje zapisov po fizičnem zaporedju,
- GET – čitanje zapisov po ključu,
- ADD – dodajanje zapisov po fizičnem zaporedju,
- DEL – brisanje zapisov,
- RWR – ažuriranje zapisov ali katerakoli kombinacija naštetih možnosti.

Za vsak tip zapisa SCHEME lahko definiramo do 16 zapisov

PODSHEME, to pomeni, da je praktično možno določiti tip pristopa za vsako poletje. Ker BAZA omogoča dostop do podatkov več uporabnikom, se ob možnosti ažuriranja potrebni zapisi zaklenejo za druge programe, ki bi želeli te zapise istočasno uporabiti. Vsak program lahko zaklene večje število zapisov (run-time parameter).

Če zaklenjen zapis želijo uporabljati tudi drugi programi, BAZA zapis po določenem času sprosti in s tem prepreči situacijo »dead lock«.

Za ohranjanje integritete baze v primeru napak, ki se lahko odražajo kot programske, sistemske, diskovne ali kot napake logične transakcije, poskrbi BAZA z beleženjem (logging) in restavriranjem (recovery).

Glede na vrsto napake in način restavriranja BAZA pozna:

- transakcijsko beleženje in restavrirno proceduro, ki je za

uporabnika transparentna,

- funkcijsko beleženje in restavrirni storitveni program, ki ga sproži upravljalca BAZE.

Ko se program izvaja in se podatki v bazi ažurirajo, BAZA vse spremembe beleži, vendar fizično spremembo ne zvede, vse dokler program ne zaključi logične transakcije in eksplicitno zahteva spremembe na bazi (DML operacija CONFIRM).

Ta lastnost omogoča, da se baza ažurira le, če je bila logična transakcija pravilno izvedena, poleg tega pa omogoča BAZI, da z avtomatsko restavrirno proceduro odpravi učinke vseh nepopolnih logičnih transakcij. Na ta način se ohranja integriteta baze pri vseh vrstah napak. Če je baza fizično uničena, se restavriranje izvede na osnovi predhodne kopije baze (BACK-UP) in vseh zapisov zabeleženih po spremembi (after image logging).

STRUKTURA BAZE PODATKOV

Struktura baze podatkov je opisana s SCHEMO, ki je logični opis celotne baze podatkov in zajema:

- definicije vseh TIPOV ZAPISOV
- definicije povezav med zapisi, ki so združene v LOGIČNI SCHEMI
- fizično strukturo baze
- parametre, ki določajo izvajanje operacij nad bazo in so zajeti v operativnem področju
- definicije PODSHEM.

Tipi zapisov v SCHEMI vsebujejo opise vseh relevantnih podatkov za področje aplikacij, ki jih pokriva podatkovna baza.

Take zapise v bazi imenujemo LOGIČNI ZAPISI. Vse enake logične zapise v bazi imenujemo zbirka zapisov.

PODSHEMA je programsko okno v bazo podatkov, opisuje logično

sliko baze podatkov s stališča aplikativnega programa.

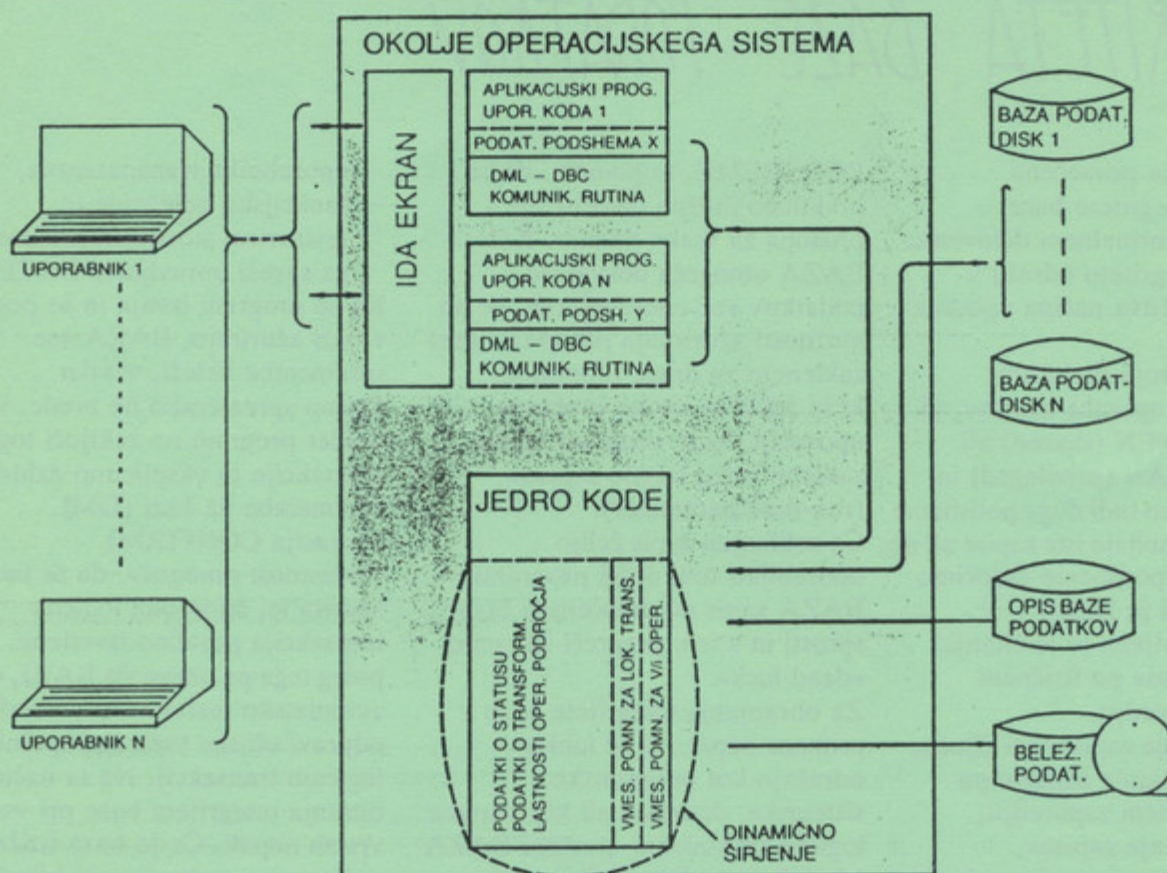
PODSHEMA definira zapise, kakršne uporablja aplikativni program. Zato imenujemo te zapise PROGRAMSKI ZAPISI. Programski zapis lahko vsebuje vse ali pa le nekatere podatke, ki jih vsebuje logični zapis. V isti PODSHEMI smemo iz enega logičnega zapisa definirati več programskih zapisov in za vsakega svoj način uporabe.

BAZA hrani podatke v za to predvidenih datotekah na diskih, ki se imenujejo podatkovni kontejnerji. Fizična struktura baze je popolnoma neodvisna od logične:

- vsi logični nizi zapisov so v enem kontejnerju,
- vsak logični niz je v svojem kontejnerju,

- en logični niz je lahko razdeljen v več kontejnerjih,
- vse kombinacije prejšnjih možnosti.

Podatkovni kontejnerji so lahko locirani na vseh diskovnih enotah, ki jih podpira operacijski sistem. Formalno so podatkovni kontejnerji standardne datoteke, za delo z njimi se uporabljajo standardni storitveni programi. Nad zapisi v SCHEMI definiramo realcije s pravilom SET, ki relacijo poimenuje. Vse relacije so združene v LOGIČNI SCHEMI, ki je priključena k shemi. V relaciji sta po dva logična zapisa, od katerih je eden nadrejen (owner) in drugi podrejen (member). Z ozirom na to, pozna BAZA 3 vrste zapisov:



SHEMATSKI PRIKAZ IDA-BAZE

- SAMOSTOJEN tip zapisa, ki je identificiran z direktnim ključem; v relaciji nastopa le kot nadrejen zapis, lahko pa se v podatkovni bazi pojavlja tudi samostojno.
 - KOMBINIRAN tip zapisa, ki je identificiran z direktnim ključem; v relaciji nastopa kot nadrejen, podrejen ali istočano kot nadrejen in podrejen v različnih relacijah. Zapisi tega tipa so lahko povezani v skupine, ne morejo pa se pojavljati samostojno.
 - ODVISNI tip zapisa, ki nima direktnega ključa; v relaciji nastopa le kot podrejeni zapis. Te vrste zapisi so vedno povezani in podrejeni enemu ali več drugim tipom zapisov.
- Vsaki relaciji pripada tudi ustrezen ključ povezave. Med dvema zapisoma je možno definirati tudi več različno poimenovanih relacij.

Recipročne relacije niso mogoče. BAZA torej omogoča zgraditi mrežno strukturo baze podatkov v skladu s priporočili CODASYL-a. BAZA upravlja tudi s sekvenčnimi podatkovnimi strukturami. Vključevanje sekvenčnih datotek v bazo podatkov pomeni:

- enoten koncept za vse vrste podatkov,
- prihranek v naslovnem prostoru programa,
- skupno definiranje in uporabljanje vseh podatkov.

Koncept SHEMA – PODSHEME omogoča neodvisnost podatkov od programov. Ne glede na to, kako se baza dopolnjuje in širi z neprestanimi zahtevami po spremembah in postopnem dograjevanju informacijskega sistema se te spremembe odražajo le v SHEMI, medtem pa lahko kljub temu ostanejo PODSHEME

vsebinsko nespremenjene; vključujejo iste zapise, elemente in povezave kot prej. OPIS PODSHEME ni del programa, pač pa je shranjen v posebni knjižnici na disku. Pri inicializaciji navede program ime PODSHEME in njeno geslo. BAZA potem napolni opis PODSHEME v za to predviden pomnilniški prostor v programu. Strukturo baze podatkov definira *prevajalnik DDC*, ki omogoča, da kreiramo zeleno podatkovno strukturo: shemo, podshemo, fizično strukturo, logično shemo, operativno področje ali tip zapisa.

IDA BAZA

STA JE IDA-BAZA

Baza je savršen sistem za upravljanje bazama podataka, usklađen sa ostalima proizvodima IDA. BAZA sjedinjuje prednosti mnogih inostranih proizvoda ove vrste i domaćih rešenja. Predstavlja međukariku između različitih podataka na diskovima i logičnih slika podataka, kako ih vide korisnici.

BAZA je zasnovana tako da naglašava centralno upravljanje bazom podataka, time što su strukture podataka, pristup podacima i njihova zaštita definisani odvojeno od programa. Korisnici mogu u celini da se posete problemskom programiranju, dok je upravljanje logičnom i fizičkom strukturom, kao i zaštita podataka prepustena u fazi zasnivanja upravljaču bazom podataka (data base administrator), a u fazi sprovođenja aplikacija BAZI. Time je rad programera olakšan i povećava se pouzdanost rada i bezbednost podataka.

BAZU čine monitori i servisni program koji omogućuju:

- definisanje struktura baze podataka (DD)
- upravljanje i kreiranje struktura podataka (DBF)
- realizacija programskog pristupa bazi podataka (DML)
- sprovođenje V/I operacija, startovanja i zaustavljanja jezgra sistema
- beleženje promena u fazi na nivou logične transakcije i funkcionalno beleženje promena (logging) i koje omogućuju restauriranje baze podataka (recovery) u slučaju nepredviđenih i nenormalnih prekida rada računara
- beleženje korisnikovih ulaznih transakcije (input logging) koje omogućuje ponavljanje određene obrade u paketnom režimu.

BAZA je, kao i ostali proizvodi IDA, prilagođena operacionom sistemu DELTA/V, kao i DELTA/M. Sa stanovišta korisnika, upotreba BAZE je transparentna obzirom na operacioni sistem. Posebna je pažnja posvećena iskoristavanju sistemskih resursa. Memorijski prostor se koristi dinamično sa mogućnošću adresovanja do 1 MB na DELTA/M i 4 MB na DELTA/V. Ovo obezbeđuje BAZI visoku efikasnost i širok spektar upotrebe, kako za poslovne i tehničke, tako i za aplikacije u realnom vremenu.

BAZA je tako oformljena da omogućuje istovremeni pristup bazi podataka 32 korisnika na DELTA/M i 100 korisnicima na DELTA/V operacionom sistemu. Mogućnost rezervacija na nivou pojedinih zapisa sprečava smetnje među korisnicima.

Automatsko oslobađanje zapisa posle određenog vremena, ako je zapisu potreban još neki drugi program, sprečava "dead lock" situacije.



JEDNOSTAVAN PROGRAMSKI

PRISTUP PODACIMA

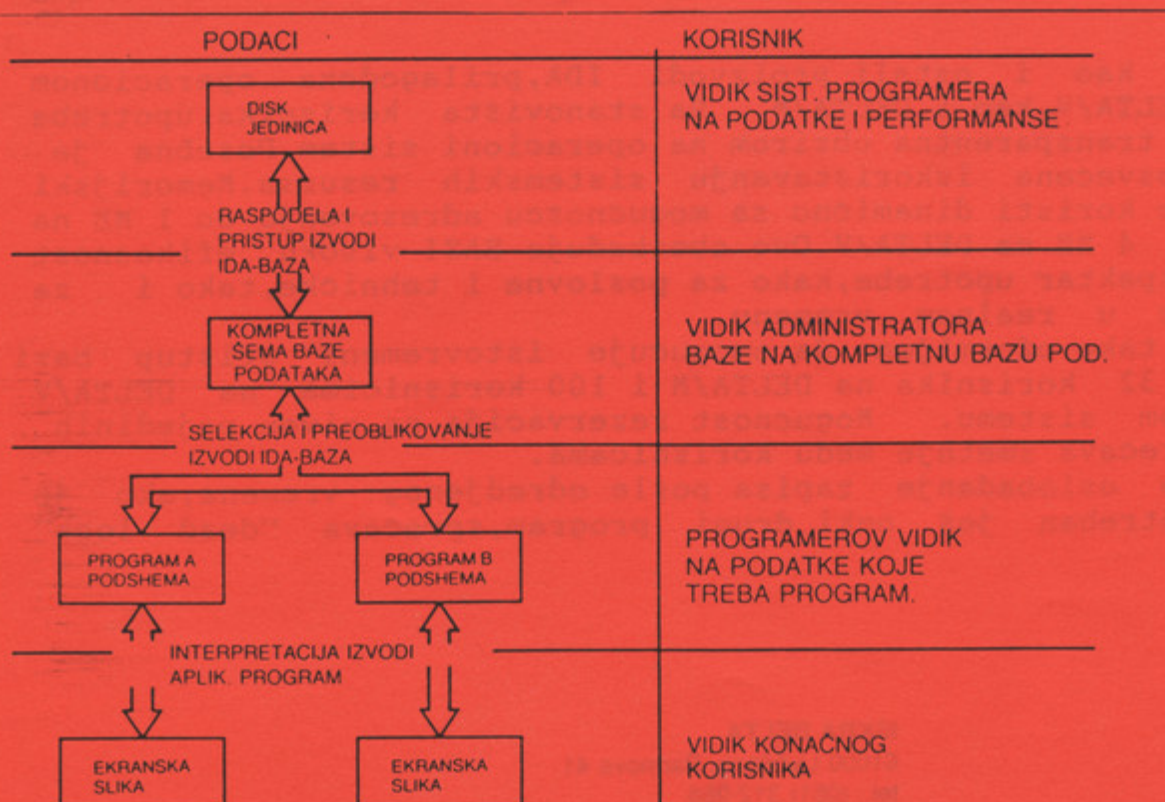
Jezik za manipulisanje sa bazom podataka (DML) izveden je pomoću naredbi za poziv - CALL. To znači da možemo kao ugostiteljski jezik, s kojim pišemo korisničke programe, da upotrebimo praktički bilo koji viši programski jezik. BAZA definiše četiri osnovne tipove programskih operacija podataka u bazi:

- čitanje GET (GETD, GETP, GETR)
- dodavanje INSERT (INSA, INSB, INSG)
- menjanje REWRITE (RWRG)
- brisanje DELETE (DELG)

DML je ustvari kao potprogram sa više ulaznih tačaka, koji prilikom poziva vrši kontrolu parametra i brine se o daljem sprovođenju tražene akcije.

Komunikacija između korisničkog programa i BAZE odvija se preko internih registara i područja koja su navedena u listi parametara i čiji je programski opis jedan od izlaza programa DDC. Ovi opisi mogu da se uključe u DATA DIVISION korisničkog programa u izvornom cobolskom kodu.

Kada se kontrola vraća korisničkom programu, treba da testiramo DB STATUS registar, kako bismo ustanovili, da li je tražena akcija bila uspešna ili nije, a ako nije, zbog čega. Obzirom na uzrok neuspešnosti akcije mogu u korisničkom programu da budu predviđene odgovarajuće mere. Kao prvi poziv treba da bude izvršen sa ulaznim tačkom HELLO u kojem poručujemo BAZI, koja ćemo polja koristiti kao interne registre, sa kojom će podšemom program raditi i , naravno, lozinku. Time se upostavlja komunikacija između BAZE i programa. Izlazna tačka BYE služi za završetak rada sa trenutno aktivnom podšemom (zatvaranje nizova). Sve promene na bazi fizički se ne realizuju odmah, nego ih treba potvrditi sa ulaznim tačkom CONFIRM ili ih poništiti sa ulaznom tačkom CANCEL. Ova osobina BAZE usko je povezana sa njenom sposobnošću restauriranja na nivou transakcije.



IDA MULTI BAZA

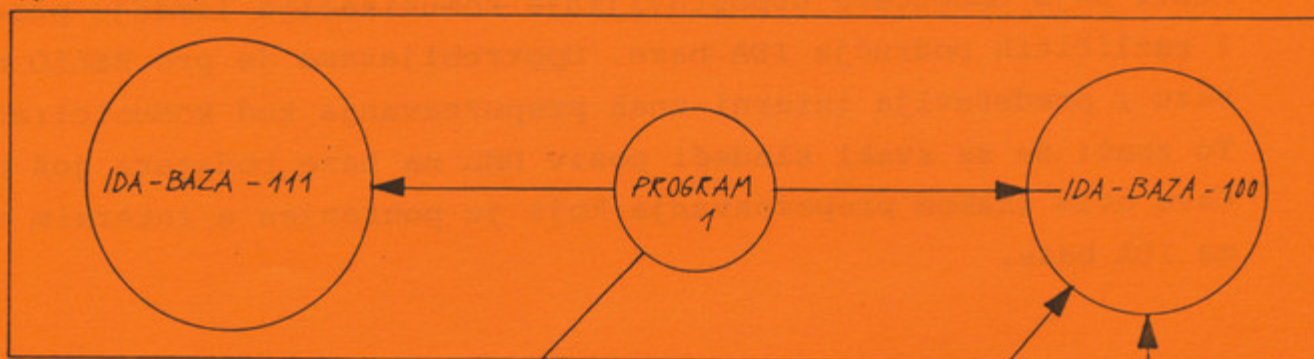
IDA multi baza omogoča dostop do več baz podatkov v istem programu. Taka oprema je nujna pri koordiniranju velikih informacijskih sistemov, saj je nekatere računalniške obdelave potrebno decentralizirati, druge pa koordinirati.

Proizvod IDA distribuirana baza uspešno rešuje problem dostopa do podatkov v globalni bazi podatkov na nivoju računalniške mreže. Ne omogoča pa dostopa do večjih baz podatkov v istem programu. Tak način dostopa je potreben takrat, kadar več parcialnih baz podatkov predstavlja globalno bazo podatkov. To problematiko rešujeta skupaj IDA distribuirana baza in IDA multi baza.

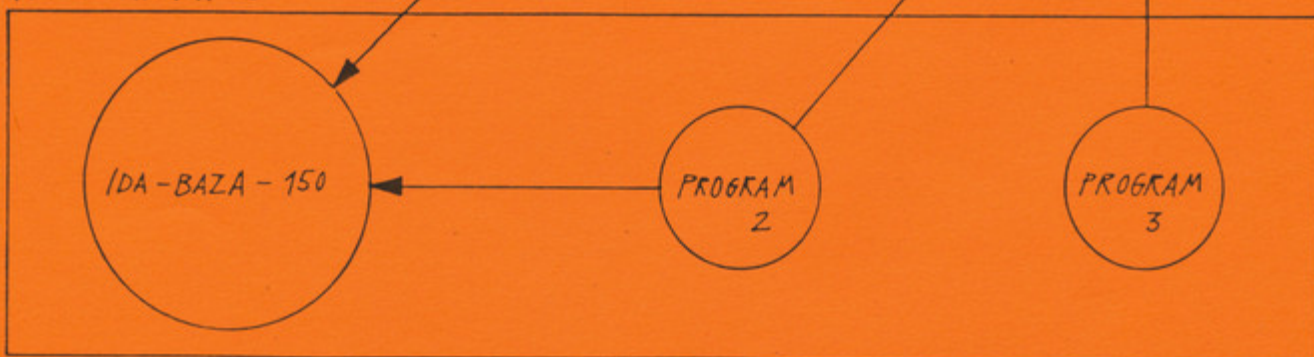
Prav tako dobro pa sama IDA multi baza združuje vire podatkov na nivoju ene centralne procesne enote.

Pri realizaciji IDA multi baze je bilo potrebno dodati nov DML ukaz, MULTI, ki omogoča vzpostavljanje komunikacij med programom in različnimi področji IDA baze. Uporabljamo ga pred HELLO na bazo in predstavlja interni razpoznavni znak pri komuniciranju. To pomeni, da se vsak nadaljni klic DML na bazo podatkov se označi s tem razpoznavnim znakom, ki je shranjen v internih registrih IDA baze.

RAČUNALNIK 1



RAČUNALNIK 2



IDA AVTOMATSKO LOGIRANJE TRANSAKCIJ

Pri razvoju aplikacij s starejšimi informacijskimi tehnologijami so bila poleg napornega dela porabljena ogromna sredstva. Zato je pri uvajanju novih tehnologij nujno potrebno narediti nekakšen pripomoček, ki bi omogočal njihovo uporabo pri že obstoječih aplikacijah.

IDA logiranje transakcij pri IDA bazi je tehnologija, pri kateri je potrebno spremeniti programe, ki so bili napisani še predno je bila tehnologija izdelana, ali pa so bili preneseni iz baze podatkov TOTAL, ki tega načina logiranja ni poznala.

Prednosti LOGIRANJA TRANSAKCIJ so sicer splošno znane, vendar jih na tem mestu vsaj naštejmo:

- zanesljiva obnovitev podatkov v slučaju napak aparaturne ali programske opreme,
- kratek čas obnovitve podatkov,
- enostavnejši način pisanja aplikacij, saj zagotavlja tudi logično skladnost podatkov (transakcija je v celoti izvedena ali zavrnjena).

AVTOMATSKO LOGIRANJE TRANSAKCIJ je nastalo ravno zato, da bi lahko uporabili tehnologijo LOGIRANJA TRANSAKCIJ v starih aplikacijah in s tem ohranili že opravljeno vloženo delo in zagotovili zanesljivost delovanja aplikacije v pogojih nezanesljivega napajanja, nezanesljive aparaturne in programske opreme, ter skrajšali čas, ki je v takem primeru potreben za obnovitev podatkov.

IDA AUTOMATSKO LOGIRANJE TRANSAKCIJA

Kod razvoja aplikacija starijim informacionim tehnologijama osim napornog rada utrošena su i ogromna sredstva. Zbog toga je kod uvođenja novih tehnologija neophodno potrebno izraditi nekakvo sredstvo koje bi omogućavalo njihovu upotrebu kod već postojećih aplikacija.

IDA logiranje transakcija kod IDA baze je tehnologija kod koje treba promeniti programe koji su bili napisani pre nego što je tehnologija izradjena ili su bili preneti sa baze podataka TOTAL koja taj način logiranja nije poznavala.

Prednosti LOGIRANJA TRANSAKCIJA su inače opšte poznate, a na ovom mestu ih nabrajamo:

- pouzdano obnavljanje podataka u slučaju greške aparaturene ili programske opreme,
- kratko vreme obnavljanja podataka,
- jedinstveniji način pisanja aplikacija jer obezbedjuje i logičnu skladnost podataka (transakcija je u celini izvršena ili odbijena).

AUTOMATSKO LOGIRANJE TRANSAKCIJA je nastalo upravo zbog toga da bi se mogla primeniti tehnologija LOGIRANJA TRANSAKCIJA u starim aplikacijama i da bi se sačuvao već obavljeni posao i trud koji je uložen i da bi se obezbedila pouzdanost delovanja aplikacije u uslovima nepouzdanosti napajanja, nepouzdanost aparaturene i programske opreme i da bi se skratilo vreme koje je u takvom slučaju potrebno za obnavljanje podataka.

IDA "GET FAST"

Pri večini aplikacij je potrebna hitra obdelava podatkov. Največkrat se za to uporablja funkcija "GETP" ali "GETG". Ker gre največkrat za izdelavo določenih pregledov ali za formiranje sekvenčnih datotek vseh ali večine podatkov, je hitrost izvajanja teh funkcij še kako pomembna. Če upoštevamo, da branje podatkov iz baze predstavlja večino vseh operacij, postane hitrost teh funkcij še pomembnejša.

Namen proizvoda IDA "GET FAST" je predvsem povečati hitrost izvajanja vseh funkcij in izboljšati prekrivanje programov na rovaš povečane porabe zmogljivosti pomnilnika.

Delovanje produkta IDA "GET FAST" je za uporabnike popolnoma nepomembno in je vezano samo na pristopne pravice posameznega programskega zapisa. Če uporabimo zapis, ki dovoljuje spremembo podatkov v bazi podatkov, potekajo vse funkcije branja v normalnem režimu - kot do sedaj. Če pa uporabimo programski zapis, ki ima samo pravico branja podatkov, se avtomatsko vse funkcije izvajajo v režimu FAST.

Branje v režimu FAST je približno dvakrat hitrejše od dosedanjega. Ker pa v tem režimu ni komunikacije in ni potrebna sinhronizacija na podatkih, je prekrivanje med procesi večje. Zato pri normalni uporabi IDA Baze pričakujemo se večje razlike med režimom FAST in dosedanjim režimom dela, saj prekrivanje lahko pride do izraza sele pri večuporabniškem načinu dela.

IDA, OCT FAST

... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..

... ..
... ..
... ..
... ..
... ..

... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..

... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..

IDA "GET FAST"

Kod većine aplikacija potrebna je brza obrada podataka. Najčešće se zato upotrebljava funkcija "GETP" ili "GETG". Pošto se najčešće radi o izradi određenih pregleda ili formiranju sekventnih datoteka svih ili većine podataka, brzina vršenja ovih funkcija je i te kako važna. Ako imamo u vidu da čitanje podataka iz baze predstavlja većinu svih operacija, brzina ovih funkcija biva još značajnija.

Namena proizvoda IDA "GET FAST" je pre svega u tome da poveća brzinu vršenja svih funkcija i da poboljša prekrivanje programa na račun povećanog korišćenja kapaciteta memorije.

Funkcionisanje proizvoda IDA "GET FAST" je za korisnike potpuno beznačajno (i vezano je samo na pristupna prava pojedinog programskog zapisa. Ako upotrebimo zapis koji dozvoljava promenu podataka u bazi podataka, sve funkcije čitanja se u normalnom režimu odvijaju kao do sada. Međutim, ako upotrebimo programski zapis koji ima samo pravo čitanja podataka, automatski se sve funkcije vrše u režimu FAST.

Čitanje u režimu FAST je približno dvaput brže od dosadašnjeg. Međutim, pošto u tom režimu nema komunikacije i nije potrebna sinhronizacija na podacima prekrivanje između procesa je veće. Zbog toga kod normalne upotrebe IDA baze očekujemo još veće razlike između režima FAST i dosadašnjeg režima rada, jer prekrivanje može doći do izražaja tek kod načina rada sa više korisnika.

IDA „GET FAST“

Kod vodnje aktivnosti, kot tudi pri drugih aktivnostih, je treba upoštevati, da se pri uporabi izdelka "GET FAST" ne sme uporabljati v kombinaciji s katerikoli drugimi zdravili, ki vsebujejo kofein ali druge stimulante. Če imate kakršnokoli zdravilno stanje, morate pred uporabo izdelka "GET FAST" posvetovati s svojim zdravnikom. Izdelek "GET FAST" ne sme uporabljati pri nosečnosti in dojenju. Izdelek "GET FAST" ne sme uporabljati pri otrocih, mlajših od 18 let.

Izdelek "GET FAST" je namenjen uporabi kot pomoč pri koncentraciji in pozornosti. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo občutljivost na kofein ali druge stimulante. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo zdravilno stanje, ki ga lahko poslabša uporaba izdelka "GET FAST".

Izdelek "GET FAST" je namenjen uporabi kot pomoč pri koncentraciji in pozornosti. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo občutljivost na kofein ali druge stimulante. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo zdravilno stanje, ki ga lahko poslabša uporaba izdelka "GET FAST".

Izdelek "GET FAST" je namenjen uporabi kot pomoč pri koncentraciji in pozornosti. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo občutljivost na kofein ali druge stimulante. Izdelek "GET FAST" ne sme uporabljati pri osebah, ki imajo zdravilno stanje, ki ga lahko poslabša uporaba izdelka "GET FAST".

IDA DISTRIBUIRANA BAZA

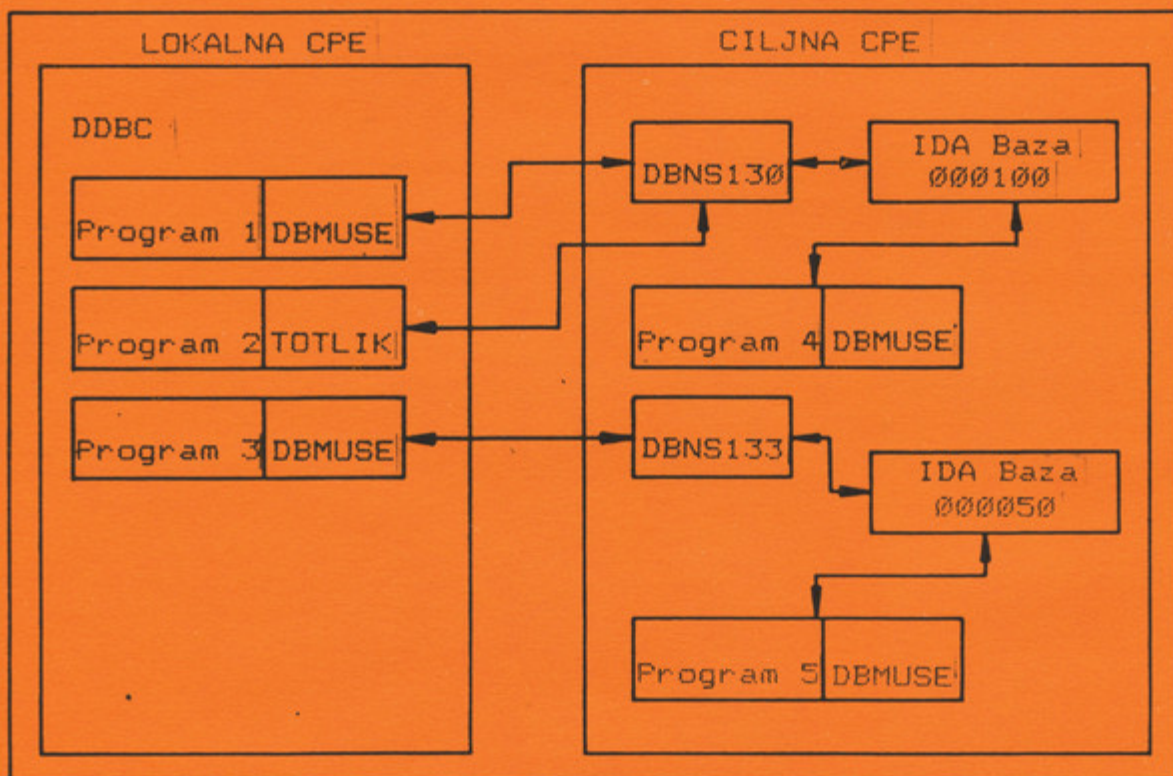
IDA distribuirana baza omogoča dostop do skupne baze podatkov iz vseh vozlišč računalniške mreže.

Omogoča sočasen dostop do skupne baze podatkov iz vseh CPE (centralnih procesorskih enot), ki so vključene v konfiguracijo DELTANET.

Vsaka baza podatkov, ki je fizično prisotna na neki CPE, je z vidika programov te CPE lokalna, kar izboljšuje storilnost.

Zagotovljena je večprogramska zaščita in integriteta baze podatkov.

Uporaba IDA distribuirane baze je nepomembna za programe in podatke - isti programi lahko uporabljajo lokalno (standardno) IDA Bazo ali IDA distribuirano bazo. Prav tako je dovoljena sočasna uporaba programov, ki uporabljajo konvencijo (vmesnik) IDA ali TOTLIK DML.



IDA distribuirana baza skupaj z IDA Multi Bazo povečuje funkcionalnost programske proizvodnje v večprocesorskem okolju, oziroma v mreži DELTANET.

IDA distribuirana baza je proizvod, ki je optimalno prilagojen mreži DELTANET in omogoča uporabniku, da na enostaven način uporablja prednosti te arhitekture.

IDA distribuirana baza se prav tako uspešno uporablja v mrežah CLUSTER.

DATA DISTRIBUTION

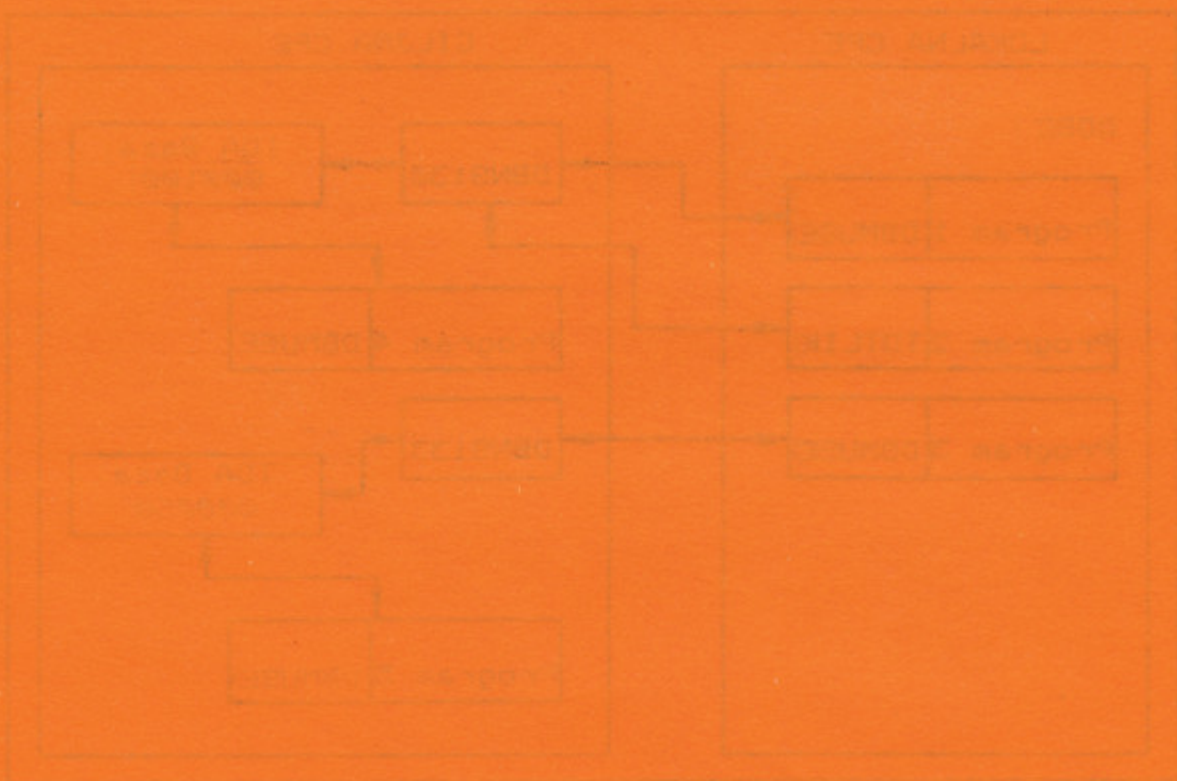
The distribution of data is a key factor in determining the efficiency of a system. It is essential to understand the characteristics of the data and how it is stored and accessed. This document discusses the various factors that influence data distribution and provides a framework for analyzing and optimizing data storage and retrieval.

The primary factors that affect data distribution are the volume of data, the frequency of access, and the complexity of the data. These factors determine the most appropriate storage and access methods. For example, a large volume of data that is accessed frequently may require a different storage strategy than a smaller volume of data that is accessed infrequently.

The distribution of data is also influenced by the physical characteristics of the storage medium. For example, the speed of access is affected by the physical distance between the data and the access point. This is why data that is accessed frequently should be stored closer to the access point.

The distribution of data is also influenced by the logical structure of the data. For example, data that is organized into a hierarchical structure may be accessed differently than data that is organized into a flat structure.

The distribution of data is a complex and multi-faceted problem. It requires a deep understanding of the data and the system in which it is used. This document provides a framework for analyzing and optimizing data distribution, but it is not a substitute for a detailed analysis of a specific system.



The distribution of data is a key factor in determining the efficiency of a system. It is essential to understand the characteristics of the data and how it is stored and accessed. This document discusses the various factors that influence data distribution and provides a framework for analyzing and optimizing data storage and retrieval.

The primary factors that affect data distribution are the volume of data, the frequency of access, and the complexity of the data. These factors determine the most appropriate storage and access methods. For example, a large volume of data that is accessed frequently may require a different storage strategy than a smaller volume of data that is accessed infrequently.

The distribution of data is also influenced by the physical characteristics of the storage medium. For example, the speed of access is affected by the physical distance between the data and the access point. This is why data that is accessed frequently should be stored closer to the access point.

The distribution of data is also influenced by the logical structure of the data. For example, data that is organized into a hierarchical structure may be accessed differently than data that is organized into a flat structure.

The distribution of data is a complex and multi-faceted problem. It requires a deep understanding of the data and the system in which it is used. This document provides a framework for analyzing and optimizing data distribution, but it is not a substitute for a detailed analysis of a specific system.

IDA DISTRIBUIRANA BAZA

IDA distribuirana baza omogućava pristup zajedničkoj bazi podataka sa svih čvorišta računarske mreže.

Omogućava istovremeni pristup zajedničkoj bazi podataka sa svih CPJ (centralnih procesorskih jedinica) koje su uključene u konfiguraciju DELTANET.

Svaka baza podataka koja je fizički prisutna na nekoj CPJ, je sa aspekta programa ove CPJ lokalna, što poboljšava efikasnost.

Obezbedjena je zaštita više programa i integritet baze podataka.

Upotreba IDA distribuirane baze je bez značaja za program i podatke - isti programi mogu upotrebljavati lokalnu (standardnu) IDA bazu ili IDA distribuiranu bazu. Takođe je dozvoljena istovremena upotreba programa koji koriste konvenciju IDA ili TOTLIK DML.

IDA distribuirana baza zajedno sa IDA Multi bazom povećava funkcionalnost programske proizvodnje u višeprocessorskoj okolini odnosno u mreži DELTANET.

IDA distribuirana baza je proizvod koji je optimalno prilagodjen mreži DELTANET i korisniku omogućava da na jednostavan način koristi prednosti ove arhitekture.

IDA distribuirana baza se takođe uspešno upotrebljava u mrežama CLUSTER.

11/11/11

THE DISTRIBUTION OF

The distribution of the population of the United States in 1950 is shown in the following table. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.

The population of the United States in 1950 was 150,697,000. The population of the United States in 1950 was 150,697,000.