Radovan Kirchbaum
Fakultet organizacije i informatike
V a r a ž d i n

UDK 681.3.06
Stručni rad

I N T E G R I T E T P O D A T A K A U I N F O R M A C I J - S K I M S I S T E M I M A

Ovaj članak ukratko obradjuje teoretske i praktične probleme podrške i pomoći koju informacijski sistem osnovan na kompjuteru može pružiti u isključivanju namjernih ili nenamjernih pogrešaka, koje mogu nastupiti prilikom korištenja informacijskog sistema. Takva podrška, odnosno pomoć, moguća je samo u onim slučajevima kada se pogreške mogu sa sigurnošću raspoznati ili kada se mogu predvidjeti. Takvi su slučajevi (a) kada se skup dozvoljenih operacija obrade može formalno ograničiti, (b) pri korištenju sistema od strane više korisnika, kada postoji opasnost da medjusobno nezavisne operacije obrade različitih korisnika ne povoljno utječu jedne na druge,(c) kada nastupi disfunkcija sistema zbog nedostataka u programima ili zbog tehničkih defekata na strojevima i uredjajima, ili (d) kada neovlaštene osobe zahvaćaju podatke u bazi podataka informacijskog sistema. Prije obradjivanja pojedinog od navedenih područja problema daje se kratak pregled s razgraničenjem upotrijebljenih pojmova.

1. UVOD

Iako su problemi, koji se javljaju u vezi s integritetom podataka u informacijskim sistemima, prisutni i u sistemima s jednim korisnikom, oni se u sistemima s više istovremenih korisnika enormno povećavaju po broju, po vrsti i po složenosti. Zbog toga tradicionalna sredstva, kojima su raspolagali korisnici u svojoj nadležnosti, nisu više dovoljna za osiguranje integriteta podataka. U sistemima s više korisnika nužno mora informacijski sistem preuzeti odredjene kontrolne funkcije, a time i odgovornost za osiguranje integriteta podataka.

Podjemo li od gledišta korisnika, tada informacijski sistem osnovan na kompjuteru možemo definirati kao tehničku instancu koja korisniku pomaže u upravljanju i manipulaciji njegovim modelima. Ako tu pomoć interpretiramo kao bezuvjetno izvodjenje operacija obrade, u obliku koji je unaprijed zadan od strane korisnika,tada možemo reći da informacijski sistem korektno funkcionira samo kada ne mogu promaknuti namjerne ili nenamjerne pogreške korisnika,kada se različiti korisnici ne ometaju u radu i kada tehnička postrojenja rade pouzdano (3).

Nenamjerne pogreške u korištenju sistema dogadjaju se vrlo često. Uzroci mogu biti različiti, kao npr. trajniji gubitak pregleda ili nepotpuni pregled nad bazom podataka na strani korisnika, ili pak prosta nepažljivost korisnika u korištenju sistema (3). Ova vrsta pogrešaka može se u velikoj mjeri spriječiti ako se projektom informacijskog sistema korisniku omogući da unaprijed utvrdi operacije obrade koje se smiju izvoditi s odredjenim podacima ili skupovima podataka (dozvoljene operacije). U tom slučaju informacijski sistem može preuzeti kontrolnu funkciju kojom se preispituje svaki zahtjev za izvodjenje operacija obrade i donosi odluka da li će se zahtjevu udovoljiti, ili će se zahtjev otkloniti. Ako se na taj način uspiju otkloniti sve operacije obrade , koje bi s gledišta korisnika dovele do stvaranja neispravnog modela baze podataka, tada kažemo da informacijski sistem osigurava korisniku konzistentnost podataka. Izbor modela podataka i konkretizacija tog modela, koji se unaprijed obavljaju prilikom izgradnje i implementacije informacijskog sistema, već predstavljaju jedan korak u pravcu osiguranja konzistentnosti podataka jer nam ti postupci daju mogućnost odredjivanja dozvoljenih operacija obrade.

Služi li se sistemom u bilo kojem i svakom trenutku samo jedan korisnik i ako su u sistem unijeti ispravni podaci, tada ispravnost baze podataka i obradjenih informacija zavisi samo od korektne izvedbe medjugranice na kojoj se obavlja komunikacija izmedju korisnika i kompjutera. Medjugranicom (intermediate boundary) nazivamo granice izmedju podsistema (6). Nasuprot tome, koristi li sistem više korisnika istovremeno, tada usprkos korektnoj izvedbi medjugranice može doći do medjusobnog ometanja, a posljedica su neispravni rezultati obrade i neispravna stanja baze podataka (3). Uzrok ometanju je vremensko preklapanje više procesa obrade, do kojeg dolazi zbog toga što svaka operacija obrade, koju zahtijeva korisnik, traži odredjeno konačno vrijeme rada kompjutera(proces). Ako više korisnika radi istovremeno, tada očigledno može doći do vremenskog preklapanja različitih procesa. Koji medjusobni utjecaji mogu nastupiti i kojim se mjerama koordiniraju, to su pitanja obuhvaćena izrazom cjelovitost procesa.

Općenito uzevši, pouzdanost tehničkih uredjaja nije moguće kontrolirati. Tehnički defekti na strojevima i uredjajima, ili ne - predvidjena mehanička djelovanja, kao i neotkrivene koncepcijske i logičke pogreške u novo primijenjenim programima, mogu pro uzročiti pogrešne promjene u bazi podataka koje se uopće ne mogu ispraviti, ili je to vrlo teško moguće. Mjere za sprečavanje takvih pogrešaka obuhvaćene su pojmom osiguranje podataka.

Konzistentnost podataka, cjelovitost procesa i osiguranje podataka obuhvaćaju zaštitu protiv nenamjernih pogrešaka. Posebne mjere predvidjaju se za zaštitu od učinilaca namjernih pogrešaka, tj. protiv zahvata i promjena u bazi podataka s namjerom zloupotrebe. Za sprečavanje takvih zahvata uobičajen je naziv zaštita podataka.
Date (2) s pravom naziva blizancima probleme konzistentnosti podataka i cjelovitosti procesa s jedne strane, te probleme osiguranja i zaštite podataka s druge strane. Dok su prvi usmjereni na postizanje besprijekornosti stanja baze podataka, putem pravila konzistentnosti, drugi su usmjereni na rekonstrukciju baze podataka i sprečavanje neovlaštenog pristupa,sistim ciljem: besprijekornost stanja baze podataka.

2. KONZISTENTNOST PODATAKA

Pojam konzistentnosti podataka usko je povezan s pojmom sheme baze podataka. Shema baze podataka može se interpretirati i kao skup iskaza o sadržaju baze podataka. Ovi iskazi su nepromjenljivi za vrijeme obrade, a nazivaju se pravilima konzistentnosti. Pravilima konzistentnosti može se zahtijevati da informacije sadržane u bazi podataka budu kompletne u odredjenom smislu. Tako npr. kada je nešto poznato o studentu, tada je poznat njegov matični broj, ime i prezime, datum rodjenja i godina prvog upisa. Složenija pravila mogu zahtijevati npr. da suma troškova poslovanja neke organizacijske jedinice ne smije prekoračiti planirani iznos. Informacijski sistem osigurava integritet podataka u onoj mjeri u kojoj omogućava korisniku specificiranje pravila konzistentnosti, koja se tada automatski primjenjuju.

Specificiranje pravila konzistentnosti korisnik bi mogao izvršiti na osnovi raspoloživog jezika za definiranje podataka (Data Definition Language, DDL) na taj način da se za odredjivanje dozvoljenih operacija obrade poslužimo isključivo specifikacijama tipova i vrsti podataka (5,10). U tom slučaju dobit ćemo skup onih operacija obrade za koje je u konceptualnoj shemi tipa ili vrste podataka utvrdjen odgovarajući operator čiji je rezultat definiran u skladu s odgovarajućom shemom. Iz toga slijedi da je stanje baze podataka konzistentno ako je nastalo izvodjenjem niza dozvoljenih operacija obrade nad primitivnim elementima konzistentnog početnog stanja. Ukoliko se pomoću DDL-a ne mogu dovoljno precizno odrediti sve vrste dozvoljenih operacija obrade, te se kao dopuštene pojave operacije koje bi morale biti isključene, tada je potrebno izraditi dodatne iskaze koje nazivamo uvjetima konzistentnosti. U kojoj će mjeri u pojedinom slučaju biti potrebno zadati

takve uvjete, to zavisi od razvijenosti i moći raspoloživog DDL-a. Poštivanje dodatno zadanih uvjeta konzistentnosti moguće je nadzirati samo putem specijalnih programa, a često puta samo putem korisnikovih programa.
Korisnik će, prema tome, moći definirati pravila i uvjete konzistentnosti ako mu osiguramo odgovarajući općeniti jezik (4). Medjutim, može se dogoditi da korisnik specificira pravila konzistentnosti koja sama unutar sebe i medjusobno nisu konzistentna. Problem može biti riješen samo ako je jezik u kojem su izražena pravila konzistentnosti dovoljno jednostavan, tako da se može donijeti odluka o konzistentnosti samih pravila.

Iz primjene načela odgovornosti informacijskog sistema za osiguranje integriteta podataka proizlazi nužnost ispitivanja konzistentnosti stanja baze podataka uvijek nakon njezinog modificiranja (4). Trenutak kada će se ispitivati konzistentnost baze podataka mora biti pažljivo odredjen, jer ispitivanje skupa složenih pravila konzistentnosti može iziskivati pristup vrlo velikim dijelovima baze podataka, tako da samo ispitivanje može trajati u rasponu izraženom u satima kod malih baza podataka, pa sve do ne koliko tjedana kod vrlo velikih baza podataka. Stoga informacijski sistem mora biti sposoban u toku transformacije baza podataka odrediti, za zadano pravilo konzistentnosti, koliko će trajati ispitivanje, a time indirektno odrediti veličinu troškova prouzročenih ispitivanjem. Općenito možemo reći da ispitivanje konzistentnosti ima smisla ako je potrebno vrijeme limitirano linearnom funkcijom vremena potrebnog za izvodjenje obrade bez osiguranja integriteta.
Drugi razlog, koji iziskuje pažljivo odredjivanje trenutka za ispitivanje konzistentnosti baze podataka, nalazimo u činjenici da korisnik često mora izvesti niz modifikacija baze podataka prije nego što se ponovno postigne konzistentno stanje, kao što je to slučaj kod paralelnih i slijednih transakcija. Kad je ovaj problem uočen, došlo je do konstrukcije i uvodjenja pojma transakcije. Transakciju definiramo kao niz transformacija baze podataka na osnovi medjusobno zavisnih operacija obrade izvršenih od jednog korisnika, za koju se pretpostavlja da može konzistentno stanje transformirati u inkonzistentno (4). Održavanje pravila i uvjeta konzistentnosti još uvijek je u velikoj mjeri prepušteno korisniku, te mu je omogućeno da sve medjusobne zavisne operacije obrade, koje je potrebno izvesti, poveže stavljanjem u zagrade.Tako nastalu jedinicu tada nazivamo transakcijom. Početak i završetak transakcije nalaze se pod kontrolom korisnika. Kad god je neka transakcija završena, nastupa trenutak u kojem može otpočeti ispitivanje konzistentnosti baze podataka.

Pojam paralelne transakcije objasnit ćemo na primjeru baze podataka matičnog ureda. Promjena bračnog stanja neke osobe A u 'oženjen' smije se izvršiti samo ako su zadovoljeni uvjeti konzistentnosti: - da je polazno bračno stanje osobe A 'neoženjen',i - da je u bazi podataka evidentirana neka druga osoba B čije je polazno bračno stanje 'neudata'. Medjutim, navedeni uvjeti konzistentnosti još uvijek nisu potpuni jer se promjenom bračnog stanja osobe A mora istovremeno promijeniti i bračno stanje osobe B. Konzistentno stanje baze podataka bit će postignuto tek tada kada se provedu obje transakcije, tj. kada se promijeni bračno stanje od A (prva transakcija) i kada se promijeni bračno stanje od B (druga transakcija). Takve transakcije nazivamo paralelnim transakcijama.

Zbog nedostataka u konceptualnoj shemi nekog modela podataka ili zbog neprikladnosti DDL-a, može doći do situacije u kojoj se moraju provesti nedozvoljene transakcije kako bi se tek narednim transakcijama moglo ponovno doći do konzistentnog stanja koje se na drugi način ne može postići. Uzmimo jednostavan primjer i pretpostavimo da se iz skupa podataka o radniku ne mogu nezavisno jedan od drugoga zahvatiti podatak o školskoj spremi i podatak o dodatnim bodovima za školsku spremu. Pretpostavimo takodjer da se pri operacijama obrade ovih podataka mora voditi računa o relaciji opisanoj u slijedećoj tabeli:

Školska sprema	dodatni bodovi
- visoka	120
- viša	80
- srednja	40

Ako radnik naknadno stekne viši stupanj školske spreme od onog koji je prethodno imao, potrebno je izmijeniti podatak o školskoj spremi i podatak o dodatnim bodovima na koje je stekao pravo. No ako se podaci ne mogu zahvatiti nezavisno jedan od drugoga, to nije moguće obaviti jednom jedinom transakcijom. Stoga moramo najprije izvesti transakciju kojom mijenjamo podatak o školskoj spremi, a odmah nakon toga ćemo izvesti transakciju kojom ćemo izmijeniti i podatak o broju dodatnih bodova kako bismo na taj način ponovno postigli konzistentno stanje baze podataka. Drugu transakciju nazivamo nužnom slijednom transakcijom prve.

Za sada ne postoji jedinstvo shvaćanja o tome kako obuhvatiti i kontrolirati uvjete i pravila konzistentnosti, uključujući medjusobnu zavisnost transakcija. Uvjete i pravila konzistentnosti možemo informacijskom sistemu zadati u bilo kojem jeziku i zahtijevati da automatski poduzme potrebna ispitivanja i provede posljedične radnje.

3. CJELOVITOST PROCESA

3.1. Sinhronizacija skupova procesa

Kao što je u uvodu već napomenuto, problem integriteta izvanredno se povećava konkurentnim korištenjem informacijskog sistema od strane većeg broja korisnika. Budući da u takvom slučaju nije moguće izbjeći odredjeno vremensko preklapanje operacija, mora sistem osigurati uvjete u kojima korisnici neće interferirati jedan s drugim. U nastavku ćemo razmotriti situacije koje mogu nastati prilikom preklapanja operacija obrade. Prethodno je potrebno definirati pojam procesa pod kojim podrazumijevamo (a) izvodjenje slijeda pojedinačnih operacija obrade ako su prema pravilima i uvjetima konzistentnosti u odnosu medjusobne zavisnosti, i (b) izvodjenje jedne pojedinačne operacije obrade u svim drugim slučajevima. Poći ćemo od pretpostavke da svaka obrada za račun korisnika dovodi do konzistentnog stanja baze podataka, ako je započela od konzistentnog stanja. Takodjer ćemo pretpostaviti da se pojedinačne operacije obrade na medjugranici informacijskog sistema (točka na kojoj se obavlja komunikacija korisnika s kompjuterom) izvode bez ometanja. Iz toga slijedi da neki proces može biti izvrgnut utjecaju drugog procesa samo na početku pojedinačne operacije ili u intervalu izmedju dvije pojedinačne operacije koje slijede jedna iza druge.

Promotrimo slučaj kada izmedju dva podatka A i B postoji uvjet konzistentnosti takav da je A + B = konstanta. Uzmimo da postoji neki proces P koji se sastoji od dvije operacije:
A : = A + 100 i B : = B - 100 koje bi trebalo izvesti paralelno, ali se izvode vremenski jedna nakon druge u navedenom redoslijedu. Uzmimo da postoji i neki drugi proces R koji samo čita podatke A i B. Izvode li se procesi P i R paralelno tako da P najprije promijeni A, zatim R pročita A i B, a nakon toga P promijeni i B, imat ćemo situaciju u kojoj proces P nije izvrgnut utjecaju, ali je s gledišta procesa R stanje inkonzistentno, tj. R je na svom početku izvrgnut utjecaju procesa P.

Razmotrimo sada primjer gdje proces P rastućim redoslijedom mijenja startne osnovice radnika zaposlenih u jedinicama A i B neke organizacije udruženog rada, a neki drugi proces R mijenja podatak o rasporedu radnika a, b i c po jedinicama tako da ih premješta iz jedinice B u jedinicu A, s time da je uvjet konzistentnosti promjena startne osnovice svakog zaposlenog radnika. Izvode li se ovi procesi tako da najprije proces P promijeni startne osnovice svih radnika u jedinici A i startnu osnovicu radnika a u jedinici B bude prekinut procesom R koji promijeni razmještaj radnika

a, b i c iz B u A, nakon čega P nastavlja i završava svoju obradu, imamo situaciju u kojoj oba procesa ostavljaju po svom završetku inkonzistentno stanje baze podataka. Razlog je u tome što oba procesa izvode promjene u bazi podataka, pri čemu proces R polazi od inkonzistentnog stanja, dok proces P raspolaže u trenutku prekida i u trenutku svog nastavljanja s različitim stanjima baze podataka.
Uzmimo da su zadani podatak A i podaci B_1, B_2, ..., B_n s uvjetom konzistentnosti $A = (B_1 + B_2 + \ldots + B_n + B_{n+1} + \ldots + B_m)$. Neka postoji neki proces P koji najprije čita sve B_i, a zatim čita A, te proces R koji kreira novi element B_{n+1} i mijenja podatak A. Izvode li se ti procesi tako da P pročita sve B_i (i=1,...,n), bude prekinut procesom R koji kreira B_{n+1} i promijeni A, a nakon toga P pročita A, imamo situaciju u kojoj je stanje s gledišta procesa P inkonzistentno. Suprotno prvom primjeru, ovdje je prekinut proces koji čita, a proces koji mijenja podatke izvodi se odjednom u cjelini. Iako proces R ostavlja konzistentno stanje, dolazi do inkonzistentnosti s gledišta procesa P zbog toga što u momentu prekida raspolaže jednim, a u momentu nastavka drugim stanjem baze podataka. Mijenja li se u nekom procesu odredjeno područje baze podataka uvodjenjem novog elementa, kao što je ovdje slučaj s B_{n+1}, tada taj novi element nazivamo fantomom tog procesa zbog toga što se njegovo postojanje ne uočava neposredno, nego tek nastupanjem inkonzistentnosti u nekom drugom procesu.

Problemi ove prirode ne bi se nametali kad bi se pojedini procesi odvijali u redoslijedu strogo jedan iza drugoga. Medjutim, zbog potrebe da se poveća uspješnost informacijskih sistema postoji interes za paralelnim izvodjenjem različitih procesa u što većoj mjeri. Stoga sistem mora raspolagati mogućnostima koje udovoljavaju nužnim uvjetima za isključivanje utjecaja medju procesima: (U-1) ukupnost po nekom procesu pročitanih elemenata mora biti konzistentan isječak baze podataka, (U-2) ukupnost po nekom procesu pročitanih elemenata uvijek mora biti isječak onog stanja baze podataka kojim je proces započeo, (U-3) svaki proces mora u svakoj situaciji proizvesti konzistentno stanje baze podataka, i (U-4) ni jedan proces ne smije se prekidati dok ne završi stadij u kojem djeluje na bazu podataka, osim ako se namjerno prekida na unaprijed predvidjen korektan način. Ako su navedeni uvjeti ispunjeni za neki skup procesa koji se izvode paralelno, tada govorimo o skupu cjelovitih procesa.

Mjere za sprečavanje utjecaja medju procesima nazivamo mjerama za sinhronizaciju. Ako te mjere osiguravaju cjelovitost nekog skupa procesa, tada kažemo da je taj skup procesa ispravno sinhroniziran, što se teoretski može utvrditi pravilima sinhroniza

cije: neki skup procesa, koji se proizvoljno odvijaju, ispravno je sinhroniziran ako se pojedini procesi mogu dovesti i u neki drugi vremenski redoslijed, a ipak bude postignuto isto konačno stanje baze podataka i ako su ulazni podaci svakog procesa isti i u prvom i u drugom vremenskom redoslijedu (9).

Za svaki proces možemo reći da obradjuje neki odsječak baze podataka. Taj odsječak sastoji se od ulaznih podataka, koji predstavljaju operande, i od izlaznih podataka, tj. elemenata baze podataka koji se moraju odstraniti, promijeniti ili dodati, uključivši i one koji se na osnovi uvjeta konzistentnosti posljedično moraju promijeniti. Ulazni podaci mogu istovremeno biti izlazni podaci. Pri istovremenom odvijanju više procesa, uvjetima ispravne sinhronizacije bit će udovoljeno ako ulazni podaci jednog procesa ne pripadaju izlaznim podacima nekog drugog procesa i ako se skupovi izlaznih podataka različitih procesa ne presijecaju. Bilo koja dva procesa, za koje ovi uvjeti nisu ispunjeni, ne smiju se odvijati istovremeno. Zbog toga sistem mora raspolagati funkcijama koje procesima daju ekskluzivan pristup do odredjenog dijela baze podataka u ograničenom vremenu. Funkcije za davanje ekskluzivnog pristupa poznate su iz operacijskih sistema pod engleskim nazivima "locking" i "semaphores". U nastavku će se umjesto engleskih naziva upotrebljavati izraz pridržaj podataka ili izolacija podataka. Ekskluzivan pristup može se npr. postići na taj način da svaki proces sve svoje ulazne i izlazne podatke označi ekskluzivnom oznakom (engl: lock) koju ćemo nazivati izolatorom ili zaporkom, a nakon završene upotrebe ponovno ih stavi na raspolaganje drugim procesima skidanjem zaporke. U takvom slučaju jedan proces može se izvesti u cjelini odjednom jer pridržane ulazne i izlazne podatke ne može upotrijebiti ni jedan drugi proces.

U pogledu izbora momenta i vremena izoliranja podataka postoji više mogućnosti. Neki proces može izolirati skup potrebnih podataka odmah na svom početku, a staviti ih na raspolaganje drugim procesima tek nakon svog završetka.

Medjutim, on može svaki pojedini podatak osloboditi odmah nakon njegove upotrebe, a ako ne mora biti udovoljeno uvjetu U-2 ispravne sinhronizacije, tada može pojedini podatak izolirati tek u trenutku kada mu je stvarno potreban. Skup D (t,p) podataka izoliranih procesom p u momentu t naziva se skupom pridržanih podataka procesa p.
U pogledu načina izoliranja podataka razlikujemo izoliranje u jednoj razini od izoliranja u dvije razine. Kod izoliranja u jednoj razini stoje nam na raspolaganju samo dvije operacije za pridržaj

i oslobadjanje: lock (d) i unlock (d). Nedostatak ovog načina je u tome što se dva procesa, koji zahvaćaju iste ulazne podatke, ne mogu izvoditi istovremeno, čak ni tada kad ne mogu utjecati jedan na drugoga (npr. dva procesa čitanja podataka). Ovaj nedostatak otpada kod izoliranja u dvije razine gdje raspolažemo operacijama: lockr (d) - za izoliranje ulaznog podatka, i lockw (d) - za izoliranje izlaznog podatka. Podatak izoliran operacijom lockr ne može biti mijenjan, ali može biti upotrijebljen kao ulazni podatak od drugih procesa. Podaci izolirani operacijom lockw pridržani su isključivo za proces koji ih je izolirao, a drugi procesi ih ne mogu koristiti ni kao ulazne podatke. Medjusobno inkompatibilne su operacije lockr i lockw, a isto tako lockw i lockw, jer se neki podatak ne može označiti zaporkom lockr ako je već označen s lockw,niti se bilo koji podatak može označiti zaporkom lockw ako je već označen zaporkom lockr ili lockw. Naprotiv, dva i više procesa mogu isti podatak označiti zaporkom lockr.

Proces, koji pokuša izolirati potrebne mu podatke, a ne može to učiniti jer su podaci pridržani za druge procese, mora biti prekinut sve do momenta kad mu budu stavljeni na raspolaganje potrebni podaci skidanjem prethodno stavljenih zaporki. To se postiže sastavljanjem liste (indeksa) za svaki korišteni podatak, u koju se unose nazivi procesa kojima je taj podatak potreban s oznakom vrste zaporke koju treba staviti. Na osnovi takve liste možemo izgraditi različite strategije. Želimo li privilegirati procese koji čitaju u odnosu na procese koji mijenjaju podatke, tada ćemo svaki novi proces čitanja upisati na listu ispred prvog narednog procesa koji mijenja podatke. Pri tome postoji opasnost da procesi koji mijenjaju podatke uopće ne budu aktivirani. Ako želimo da svi procesi imaju isti prioritet, tada ćemo procese koji pridolaze upisati na kraj liste (FIFO).

Neki proces bit će brisan s liste tek kad je dosegao izvodjenje operacije unlock (d). U tom trenutku sistem provjerava da li neki od procesa s liste može biti aktiviran. Aktiviranje nekog procesa koji mijenja podatke nije moguće u trenutku brisanja jednog procesa čitanja ako se još izvode drugi procesi čitanja. Postupak prekidanja i aktiviranja procesa mora se odvijati prema posebnom protokolu, tj. mora udovoljavati odredjenim uvjetima koji osiguravaju cjelovitost (9): (a) bilo koji proces mora pridržati svaki podatak izoliranim za vrijeme njegove upotrebe, (b) ni jedan proces ne poduzima opetovano izoliranje podataka koje je već jednom stavio na raspolaganje drugim procesima, (c) svaki proces mora najkasnije pri svom završetku staviti na raspolaganje sve pridržane podatke i (d) ni jedan podatak ne smije istovremeno biti označen in-

kompatibilnim zaporkama. Schlageter (9) je dokazao da ovi uvjeti u specijalnim situacijama nisu sasvim dovoljni.

3.2. Postupci izoliranja podataka i rješavanja zastoja

Složenost kod izoliranja podataka proizlazi iz potrebe da se izoliraju elementi koji u toku postupka izoliranja još ne postoje i da se spriječi njihovo kreiranje. Postoji beskonačan broj elemenata baze podataka, koji potencijalno mogu biti kreirani, iako je skup kreiranih elemenata uvijek konačan. Problemi povezani s izoliranjem podataka analizirani su u izvještaju Eswarana i Chamberlina (4). Izoliranje podataka može biti opisano predikatima s beskonačnim ekstenzijama, ali zahtjevi uspješnosti nameću ograničenja formuliranju predikata kojima sistem mora manipulirati.

Označavanje ulaznih i izlaznih podataka, koje je potrebno izolirati, može se obaviti na različite načine. Jedna od metoda je izoliranje objekata kod koje se svaki potreban podatak pojedinačno izolira pod nadzorom protokola. Pri tome se podatak identificira nazivom, indeksom ili ključem. Prednost ove metode je što se jedan podatak izolira samo toliko vremena koliko je stvarno potreban. Nedostaci su joj što se ne može odjednom izolirati veća količina podataka, već samo podatak po podatak, pa postoji opasnost da neki drugi proces izvrši utjecaj u vremenu u kojem se provodi izoliranje; zatim što broj operacija izoliranja i broj lista može biti vrlo velik, te na koncu što ne pruža zaštitu protiv fantoma koji ne postoje u vrijeme izoliranja podataka.

Metoda izoliranja čvrstih skupova (tzv. fizičko predikatno izoliranje) omogućava da se odjednom izolira veća količina podataka jer se jednom operacijom izoliranja istovremeno pridržavaju, odnosno stavljaju na raspolaganje, svi elementi jednog skupa podataka. Skup podataka koji se izolira utvrdjuje se navodjenjem uvjeta, kao što je sadržaj odgovarajućih komponenti povezivanja na osnovi kojih se pronalaze elementi skupa i individualno izoliraju. Nedostatak te metode je što vrlo velik broj podataka može biti izoliran suviše dugo, pa dolazi do još većeg broja lista nego u prethodnoj metodi. Pored toga ova metoda ne pruža takodjer nikakvu zaštitu protiv fantoma.

Kod metode izoliranja promjenljivih skupova (tzv. logičko predikatno izoliranje) podaci koje je potrebno izolirati takodjer se opisuju navodjenjem uvjeta. Za razliku od prethodnog slučaja, umjesto eksplicitne izgradnje skupa izoliranih podataka, ovdje se prilikom svakog zahvata nekog procesa do bilo kojeg podatka ispituje nije li neki drugi proces zatražio izoliranje podataka s istim uvjeti-

ma, pa ako jeste, tada se novi podatak smatra izoliranim. Umjesto podataka ovdje se u stvari označavaju uvjeti izoliranja. Ova meto da zaštićuje od fantoma jer kreiranje novih podataka nije moguće tako dugo dok postoji odgovarajući uvjet izoliranja. Daljnja prednost je u manjem broju lista. Nedostaci su što nepotrebno izolira podatke, a naročito što ograničava mogućnost paralelnog izvo - djenja procesa. Naime, ako su u_1 i u_2 uvjeti izoliranja, nije uvijek moguće utvrditi da li su skupovi koji pripadaju nalozima lock (d_1) i lock (d_2) disjunktni u svim stanjima u kojima se nalazi baza podataka tokom obrade. Stoga se, sigurnosti radi, mora pretpostaviti da se radi o inkompatibilnim nalozima.

Izoliranje podataka ima za posljedicu opasnost od zastoja (deadlock). Svaki sistem s ograničenim brojem podataka može doći u stanje u kojem svaki aktivan proces čeka da mu se stavi na raspolaganje odredjeni podatak. Zastoji se stoga moraju spriječiti unaprijed, ili se nakon nastupanja moraju raspoznati i savladati. Stoga sistem mora raspolagati s unaprijed pripremljenim mjerama i strategijama. Navest ćemo one strategije koje su teoretski obra - djene, a nalazimo ih u primjeni (1).

Presequencing je postupak sprečavanja zastoja kod kojeg se unaprijed odredjeni procesi, za čije izvodjenje su potrebni isti podaci, dovode najprije u neki redoslijed i zatim izvode jedan za drugim. Budući da procesi moraju biti unaprijed poznati i odredjeni, dolazi u stvari do odlaganja njihovog izvodjenja. I svi potrebni podaci moraju biti unaprijed poznati i prilikom početka procesa izolirani, pa dolazi do izoliranja vrlo opsežnih skupova podataka. Postupak je prikladan samo za grupnu obradu (batch processing).

Preclaiming je takodjer postupak sprečavanja, kod kojeg se nastoji prekinute procese aktivirati odmah čim je moguće udovoljiti njihovim zahtjevima za podacima. Ovaj postupak prilagodjen je interaktivnom načinu rada. Sličan je prethodnom postupku po tome što se i ovdje mora za svaki proces, prije početka izvodjenja, izolirati skup svih potrebnih podataka. Stoga su mu i nedostaci slični, tj. izoliranje podataka vrši se na početku i u velikim skupovima. Pored toga potrebno je osigurati ograničenje vremena trajanja prekida procesa.

Preemption je postupak savladavanja zastoja kod kojeg se praktici ra izoliranje nekog podatka tek u trenutku neposredno prije njegove upotrebe. Pretpostavimo da je nastupio zastoj i da je zastoj otkriven pomoću bilo kojeg postupka. U tom slučaju će, prema ovom postupku, jedan od prekinutih procesa biti završen (termination) stavljanjem na raspolaganje drugim procesima podataka pridržanih za taj proces, ili će taj proces biti stavljen u neko prethodno

stanje (rollback). Zbog izbora postupka izoliranja ova strategija zahtijeva najveći radni prostor. Značajan joj je nedostatak što se poništavaju promjene izvedene procesima koji se vraćaju u neko prethodno stanje, pa se moraju opetovano izvršiti.To ima za posljedicu veliki utrošak vremena jer sve transakcije moraju biti protokolirane, ili se ugroženi dijelovi baze podataka moraju kopirati u kritičnim momentima.

Danas se najčešće upotrebljava mješavina preemption i preclaiming postupaka. U tu se svrhu svaki proces dijeli u dvije faze, jedno je faza izoliranja podataka (seize phase), a druga je faza izvodjenje procesa (execution phase). U fazi izoliranja podataka traže se ukupni podaci po redoslijedu. Ako dodje do zastoja, moguće je izvršiti vraćanje u prethodno stanje bez većih teškoća jer se još nisu dogodile promjene u bazi podataka.

Za raspoznavanje zastoja poznati su različiti postupci, a svi se osnivaju na principu konstrukcije grafa. Kod jednih se konstruira graf kojim se opisuje redoslijed zahvatanja podataka od strane različitih procesa. Kod drugih se konstruira graf procesa kojim se pokazuje koji procesi prekidaju druge i koje druge procese time što uzimaju u upotrebu neki podatak. Prilikom zastoja graf se u svakom slučaju ispituje na postojanje ciklusa, prema poznatim uspješnim algoritmima za tu svrhu.

4. OSIGURANJE PODATAKA

Izgradnja baze podataka povezana je s vrlo velikim investicijskim troškovima, velikim utroškom ljudskog rada i zahtijeva naročito mnogo vremena. Gubitak već i manjeg broja podataka može imati za posljedicu neupotrebljivost baze podataka. Stoga je nužno poduzeti mjere da se zaštitimo od gubitka baze podataka.

Mjere za osiguranje podataka moraju se poduzeti uvijek kad neka transakcija ne može normalno završiti. Uzroci i izvori smetnji su uglavnom zastoji, logičke pogreške u programu korisnika (na primjer, dijeljenje s nulom), ispadi sistema, mehaničke pogreške (oštećenja magnetskih vrpci, diskova ili bubnjeva), a nešto rjedje i viša sila (požar, poplava i sl.). Sprečavanje takvih smetnji ili njihovih posljedica nazivamo osiguranjem podataka. Osiguranje podataka ne smije dugoročno prouzrokovati veće troškove nego što bi ih prouzrokovao sam gubitak baze podataka, a mora se tako izvoditi da što manje ometa korisnike i da bude što manje uočljivo od strane korisnika. Prvi cilj osiguranja je izolacija pogreške kako bi se spriječila njezina propagacija u bazi podataka. Drugi cilj je ponovno izvodjenje prekinutih transakcija koje nisu prouzrokovale pogrešku, a bile su izvrgnute utjecaju pogreške. Danas se

uglavnom primjenjuju tri postupka koja će se opisati u nastavku (2; 7; 8; 10; 11).

Postupak protokoliranja sastoji se u prepisivanju cjelokupne baze podataka, najčešće na magnetske vrpce (tzv. sigurnosne vrpce). Prepisivanje se izvodi u odredjenim vremenskim intervalima koji zavise od učestalosti i broja transakcija. U vremenu izmedju dva kopiranja baze podataka sve transakcije i promjene baze podataka registriraju se na posebnim magnetskim vrpcama (journal, audit trail). Potpuno kopiranje baze podataka iziskuje vrlo mnogo vremena, za razliku od protokoliranja koje relativno malo opterećuje sistem. U slučaju potrebe rekonstrukcija baze podataka otpočinje kopiranjem sigurnosne vrpce na kojoj je definirana kontrolna točka (checkpoint) baze podataka, a nakon toga se u redoslijedu ponovno izvode transakcije zabilježene na žurnalu. Ovakav postupak zahtijeva vrlo mnogo vremena za rekonstrukciju baze podataka. Postupak protokoliranja interesantan je i za interaktivan način rada, zbog manjeg utroška vremena.

Postupak kopiranja sličan je prethodno opisanom postupku utoliko što se u odredjenim vremenskim intervalima kopira baza podataka, medjutim, promjene koje nastaju i transakcije koje se izvode izmedju dva postupka kopiranja, ne registriraju se paralelno. Ovaj postupak prikladniji je za grupnu obradu (batch processing), kao što su dnevne, tjedne ili mjesečne obrade transakcija, ali ga zbog manjih troškova koje prouzrokuje nalazimo i kod interaktivnog načina obrade transakcija. Protokoliranje je kod ovog postupka u potpunosti prepušteno odluci korisnika. Kod nekih područja primjene potrebno je da se posebno zabilježi stanje koje baza podataka ima u odredjenom trenutku. Tako na primjer, kod knjigovodstva, gdje je potrebno sačuvati stanja početkom i krajem godine, polugodišta, tromjesečja itd. Zbog toga se u unaprijed odredjenim trenucima izradjuje kopija baze podataka ili njezinih odgovarajućih dijelova. Najčešće se vodi po tri, a neki puta i više, prethodnih stanja, tzv. "generacija" baze podataka, koja možemo koristiti i za svrhe osiguranja podataka.

Dvostruko vodjenje je postupak kod kojeg se paralelno vode dva istovjetna primjerka baze podataka, tzv. Master i Backup. Svaka transakcija izvodi se paralelno u oba primjerka. Kod nastupanja smetnji izvedba obrade usmjerava se na neoštećeni primjerak, a oštećeni primjerak se nakon toga sanira pomoću neoštećenog. Kod ovog postupka uštedjujemo na protokoliranju transakcija i na ponovnom uspostavljanju ispravne baze podataka, medjutim te se prednosti naplaćuju zaposjedanjem dvostrukih prostora eksternih memorija. Ovaj postupak nas ne zaštićuje od pogrešnih transakcija do kojih je došlo iz zablude.

Potrebno je spomenuti da pohranjivanje velikih količina sigurnosnih vrpci i vrpci protokola zahtijeva posebnu pažnju i mjere (npr. klimatizacija) kako bi se spriječilo prljanje ili promjena sastava. Iziskuje takodjer i posebne troškove za osvježavanje vrpci ko je se mora poduzeti u razmaku od nekoliko mjeseci.

5. ZAŠTITA PODATAKA

Zaštita podataka obuhvaća sve mjere koje služe sprečavanju neovlaštenog korištenja podacima i neovlaštenog djelovanja na podatke. Pored korisnika ovlaštenih za korištenje informacijskog sistema postoje i neovlašteni korisnici koji iz različitih pobuda pokušavaju ostvariti vezu s informacijskim sistemom i koristiti pohranjene podatke. Moguće je u kompjuteru, u cilju zaštite od neovlaštenih korisnika, izgraditi zapravo dva informacijska sistema: jedan, pravi, za ovlaštene korisnike i drugi, lažan, koji prema neovlaštenim korisnicima ostavlja utisak da se radi o pravom informacijskom sistemu. S obzirom na iznijeto mjere zaštite podataka svrstane su u različite kategorije, prema načinu sprečavanja veze s pravim informacijskim sistemom (2; 7; 8;10; 11): (a) fizička zaštita, sprečavanjem uspostavljanja fizičke veze izmedju sistema i nelegitimiranog korisnika, (b) kontrola identiteta (authentication), sprečavanjem komuniciranja izmedju pravog sistema i nelegitimiranog korisnika, ili izmedju lažnog sistema i legitimiranog korisnika, pomoću odredjene fizičke veze (linije), (c) operativna zaštita podataka (protection), sprečavanjem zahvaćanja onih dijelova baze podataka i pomoću onih operatora za čiju primjenu korisnik nije ovlašten, iako se radi o korisniku legitimiranom za upotrebu sistema (Wedekind /10/ govori o njuškanju, "browsing").

Prije realizacije odredjene mjere zaštite potrebno je utvrditi da li troškovi potrebni za realizaciju te mjere stoje u nekom razumnom odnosu prema ukupnoj šteti koju je moguće nanijeti bez te mjere. S druge strane, kod odredjivanja mjera zaštite potrebno je voditi računa o obavezama i ograničenjima koje nameću zakonski i drugi propisi.

5.1. Fizička zaštita

Neovlašteno uspostavljanje fizičke veze moguće je prodiranjem u komunikacijske sisteme drugih. Najveće opasnosti prijete od:(a) prisluškivanja uključivanjem u postojeće komunikacijske puteve, (b) hvatanja elektromagnetskih valova proizvedenih od kompjutera, i (c) neovlaštenog pristupa prostorijama računskog centra.

Za zaštitu od prisluškivanja u najnovije vrijeme proizvode se odašiljači i prijemnici za šifrirani prijenos podataka. Drugoj opasnosti se suprotstavljamo gradjevinskim mjerama, kao što je elektromagnetska izolacija računskog centra. Neovlašteni ulazak u prostorije računskog centra mora se sprečavati organizacijskim mjerama, kao što je zaključavanje računskog centra, kontrola osoblja, nadzor nad perifernim uredjajima, memoriranje podataka u šifriranom obliku itd.

5.2. Kontrola identiteta

Informacijski sistem mora se prema korisniku, kao i prema komunikacijskom sistemu, čiji je krajnji uredjaj spojen s kompjuterom, odnositi s razumnom sumnjom. Korisnik sa svoje strane takodjer mora biti u mogućnosti utvrditi da li radi s pravim, očekivanim, informacijskim sistemom. To se ostvaruje dokazivanjem identiteta, najčešće putem lozinki koje su isključivo poznate ovlaštenom korisniku i pravom informacijskom sistemu. Sigurnost utvrdjivanja identiteta , a time i stupanj zaštite podataka, zavisi od stupnja tajnosti upotrijebljenih lozinki, ili od težine njihovog provaljivanja.
Mjere za očuvanje tajnosti lozinki usmjerene su na postizanje njihove što manje i što kraće vidljivosti, odnosno izbjegavanje vidljivosti lozinka. Kod grupne obrade (batch processing) korištenje lozinki je neprikladno jer je nemoguće sačuvati njihovu tajnost. Medjutim, u interaktivnom načinu rada, kod korištenja terminalskih jedinica sa zaslonima ili pisaćim mašinama, mogu se lozinke učiniti nevidljivima. Ove mjere ne pružaju dostatnu zaštitu protiv pažljivog promatrača, a nikakvu protiv prisluškivanja i hvatanja signala na komunikacijskim putevima, pa moraju biti dopunjene postupcima koji otežavaju krivotvorenja. Budući da se ovi postupci razlikuju u djelovanju protiv različitih oblika zloupotrebe, često se kombinirano primjenjuju.

Kod korištenja liste lozinki svakom korisniku stavlja se na raspolaganje jedna unaprijed izradjena lista s lozinkama. Prilikom svakog uspostavljanja veze i korištenja informacijskog sistema korisnik izabire bilo koju lozinku s liste, koja se nakon upotrebe briše s liste i time prestaje biti upotrebljivom. Lista lozinki pohranjena u kompjuteru takodjer se smanjuje brisanjem upotrijebljenih lozinki. Kod korištenja lozinki s ograničenim vremenom valjanosti nova lozinka dodjeljuje se legitimiranom korisniku u odredjenim vremenskim razmacima. U trećoj metodi koriste se lozinke generirane od strane sistema jer se utvrdilo da je izbor lozinki, ako ga obavlja čovjek , često pod utjecajem individual-

nih navika i okolnosti života, što omogućava relativno lako otkrivanje upotrijebljene lozinke. Potrebno je paziti da lozinke budu lako pamtive. U postupku preslikavanja lozinki sistem emitira korisniku, prilikom uspostavljanja veze, neki niz brojeva iz kojeg korisnik nekim jednostavnim postupkom proizvodi novi niz znakova i unosi u sistem. Postupak koji izvodi korisnik, iako jednostavan, mora biti takav da se teško pamti na osnovi prostog promatranja. Sistem izvodi isti algoritam i usporedjuje rezultate i tek u slučaju pozitivnog rezultata korisnik može pristupiti izboru neke od programiranih lozinki. Ova metoda je povoljna i stoga što korisniku oduzima izravni izbor lozinke.

Za kontrolu identiteta informacijskog sistema za sada ne raspolažemo dovoljno sigurnim postupcima. Ovi postupci svakako su potrebni jer se lažni sistem može ponašati kao pravi sve do trenutka dok ne sazna lozinku korisnika (masquerading). Postoji postupak koji je predložen za korištenje uz šifrirani prijenos podataka: korisnik unese u terminal lozinku i saopći je sistemu, a nakon toga unosi i svoj šifarski kod u terminal. Sistem na osnovi lozinke pronalazi odgovarajući šifarski kod korisnika pomoću kojeg će dešifrirati poruke koje prima od tog korisnika. Mogućnost sporazumijevanja izmedju korisnika i sistema postoji samo i jedino u slučaju primjene istog koda. Lažni sistem ne raspolaže mogućnostima dešifriranja, pa ne može razumno ili ne može uopće reagirati, što će pobuditi sumnju na strani korisnika. Na isti način pravi sistem može otkriti nelegitimirane korisnike.

5.3. Operativna zaštita podataka

Najveći problem u zaštiti podataka čini zaštita od ilegalnih zahvata osoba koje su na bilo koji način legitimirane za pristup sistemu, kao što je slučaj kod Time-Sharing sistema ili kod sistema s više različitih korisnika. Taj problem do sada nije u potpunosti riješen i još uvijek je predmetom intenzivnih istraživanja. Sada poznate mjere zaštite spadaju u tri kategorije.

Izolacija je mjera kod koje se polazi od pretpostavke da legitimni korisnici nekog informacijskog sistema mogu ostvariti pristup i do onih dijelova baze podataka koje nisu ovlašteni upotrebljavati. Zbog toga se za svakog korisnika izradjuje njegova vlastita baza podataka i jedna vlastita kopija sistema programa, tako da korisnik raspolaže vlastitom "virtuelnom mašinom" izoliranom od svih drugih. Pored toga može u informacijskom sistemu postojati i dio baze podataka koji je zajednički za sve korisnike i do kojeg svi imaju jednako pravo pristupa.

Kontrolirano korištenje je mjera kod koje se za svaku informaciju, memoriranu u sistemu, utvrdjuje tko je može koristiti i na koji način. Prava pojedinih korisnika na zahvaćanje odredjenih informacija možemo zamisliti u obliku matrice čiji redovi označavaju korisnike, a stupci elemente baze podataka i operatore koji se mogu primijeniti na odgovarajući element baze podataka. Ovakva matrica je rijetko posjednuta. Realizacija ovakvih metoda često se pojednostavnjuje stavljanjem sigurnosnih zaporki podacima koje treba zaštititi, a s druge strane davanjem korisnicima sigurnosnih ključeva kojima mogu skinuti istoimene zaporke. Troškovi zaštite i upravljanja podacima mogu se značajno sniziti ako se zaporke dodjeljuju većim skupovima podataka.

Pored metoda kontroliranog korištenja mogu se dodatne mjere zaštite, koje nisu kao standardne predvidjene u sistemu, realizirati putem programa korisnika, pa u tom slučaju govorimo o zaštitnim mjerama progranim od korisnika. Kod toga je moguće mjere zaštite izgradjivati u skladu sa svojstvima memoriranih informacija, na primjer, sprečavanje pristupa pojedinim elementarnim podacima o cijenama proizvoda i dr. poslovnim tajnama, a dozvoljavanje pristupa statističkim iskazima o informacijama sadržanim u bazi podataka. Vrlo je malo informacijskih sistema koji ovakve mogućnosti standardno stavljaju na raspolaganje korisnicima.

Za sada ne postoji neka potpuna metodologija rješavanja problema zaštite podataka. Poznato je da gradja cjelokupnog sistema mora biti što je moguće jednostavnija, tako da ostane preglednim broj mogućnosti prodiranja u sistem. Time se olakšava i manuelna kontrola programa u pogledu otklanjanja pogrešaka koje ne nastupaju u toku normalnog korištenja i koje je zbog toga vrlo teško otkriti. Odluke o zahvaćanju podataka moraju se donosi na temelju odobrenja, a ne isključivanjem, zbog toga što će pogreške u uskraćivanju odobrenja biti mnogo lakše uočene nego u obratnom slučaju gdje pogreška daje mogućnost prodiranja od strane neovlaštenih korisnika.

Mjere zaštite, koje se konkretno provode u nekom informacijskom sistemu, moraju se objaviti i biti opće poznate. Time se njihova valjanost provjerava od strane većeg broja korisnika, zainteresiranih za zaštitu podataka. Na taj način svaki korisnik može i sam prosuditi da li mu primijenjene mjere zaštite pružaju dovoljnu sigurnost.

6. ZAKLJUČAK

Problemi integriteta još uvijek su tek na putu da budu shvaćeni i poznati. Svakako postoji velik prostor za poboljšanje postojećih rješenja, naročito u pogledu onih funkcija informacijskih sistema koje će im povećati djelotvornost. Uspješnost u smislu propusne

moći (throughput) i količine obradjenih transakcija (transaction rate) problem je od najvećeg interesa. Općenito se smatra da postojeći sistemi ne ostvaruju uspješnost na onoj razini koja bi se mogla ostvariti mada se takva tvrdnja može dokazati jedino realizacijom neke alternative koja daje bolje rezultate.

Značajni napori ulažu se u razvijanje više mogućnosti za specificiranje pravila konzistentnosti, kao i za izoliranje podataka, na način koji će sistem moći uspješno primjenjivati. Pri tome je naglasak i na više funkcija i na većoj uspješnosti, dakle kumulativno, jer bi bilo trivijalno osigurati funkcije ignorirajući uspješnost.

Problemi istovremenog izvodjenja procesa obrade povezani sa sprečavanjem zastoja (deadlock prevention) nisu riješeni na zadovoljavajući način, a značajno se povećavaju u tzv. multiprocessing sistemima, s bazama podataka distribuiranim u mreži kompjutera. Ni postupci koji omogućavaju rekonstrukciju i brzo popravljanje baze podataka (recovery) za sada ne zadovoljavaju.

Kao što vidimo, u ovoj oblasti postoje ozbiljni problemi koji se sve više povećavaju kako se grade veće i obuhvatnije baze podataka i kako se sve više povećava broj korisnika informacijskih sistema. Stoga možemo očekivati intenzifikaciju istraživalačkih napora, kako na programskoj, tako i na tehničkoj razini pojedinih uredjaja. Danas nam stoje na raspolaganju nedovoljne mogućnosti, naročito u centralnim jedinicama (granični registri, segmentiranje, zaštita čitanja/pisanja memorijskih jedinica itd.) za koje se sve do nedavno smatralo da ne mogu biti usko grlo informacijskog sistema.

L I T E R A T U R A :

1. *Bayer,R. On the Integrity of Data Bases and Resource Locking. U. Lecture Notes in Computer Science, 39. Springer Verlag, 1976, str. 339-359.*

2. *Date, C.J. An Introduction to Data Base Systems. 2.izd. Addison-Wesley, 1977, str.283-315.*

3. *Demo, B. On the Data Base Integrity Concept. Informatica Bled, 1977, 4 103.*

4. *Eswaran,K.P.; Chamberlin, D. D. Functional Specifications of a Subsystem for Database Integrity. U: D.S.Kerr (red.) International Conference on Very Large Data Bases. ACM New York, 1975, str.48-68.*

5. Juščenko,E.L.(red.). Informacionnie sistemi obščego naznačenija. Naslov originala: Codasyl Systems Committee, Feature Analysis of Generalized Data Base Management Systems, Technical Report. Statistika Moskva, 1975.

6. Langefors,B. Theoretical Analysis of Information Systems. Auerbach Philadelphia, 1973.

7. Martin J. Principles of Data-Base Management. Prentice-Hall, 1976, str.271-291.

8. Martin, J. Security, Accuracy and Privacy in Computer Systems. Prentice-Hall, 1973.

9. Schlageter,G. Access Synchronization and Deadlock-Analysis in Data Base Systems: An Implementation Oriented Approach. Information Systems 1, 1975, str.97-102.

10. Wedekind, H. Datensicherheit in Datenbanksystemen. U. Lecture Notes in Computer Science, 39. Springer Verlag, 1976, str. 315-336.

11. Wiederhold, G. Database Design. McGraw-Hill,1977,str.483-563.

Primljeno: 1979-10-15

Kirchbaum R. Datenintegrität in den Informations-systemen.

ZUSAMMENFASSUNG

Dieser Artikel befasst sich mit den theoretischen und praktischen Problemen der Hilfestellungen, welche ein rechnergestütztes Informationssystem bieten kann, um Fehler auszuschliessen, die bei seiner Benutzung auftreten können. Solche Unterstützung ist nur in den Fällen möglich, in denen Fehler auch als solche erkennbar oder vorhersagbar sind. Solche Fälle liegen vor(a) wenn sich die Menge der erlaubten Operationen formal begrenzen lässt, (b) wenn sich sonst unabhängige Operationen bei Mehrbenutzerbetrieb gegenseitig zu beeinflussen drohen,(c)wenn Fehlerverhalten durch Programm-oder technische Defekte auftritt, oder (d) wenn Unbefugte in die Datenbasis eingreifen.Vor einer eingehenden Behandlung der einzelnen Problembereiche ist ein kurzer Ueberblick,mit einer Abrenzug der verwendeten Begriffe, gegeben.