

STRUKTURE PODATAKA PREMA VRSTI I
NAČINU USPOSTAVLJANJA VEZE MEDJU
ELEMENTIMA STRUKTURE

UVOD

Prema S.K.Langeru (5) struktura nekog entiteta određena je načinom na koji su elementi, od kojih se entitet sastoji, sastavljeni u cjelinu. U pravilu svaki entitet možemo podijeliti na neke elemente. Kod muzičke ljestvice elementi mogu biti pojedini tonovi, kod ljestava pojedine prečke, kod flote pojedine plovne jedinice, kod pravca pojedine točke itd. Elementi, na koje smo podijelili entitet, posjeduju određena svojstva (oblik, volumen, težina, prostorni položaj, boja, naziv i sl.). S obzirom na ova svojstva elementi, unutar entiteta, stoje međusobno u određenim odnosima. Pod pojmom strukture podrazumijevat ćemo skup odnosa koji postoje među elementima strukture. U skladu s tim pod STRUKTUROM PODATAKA podrazumijevat ćemo skup odnosa koji postoje među elementima strukture podataka.

Elementi strukture podataka mogu biti pojedinačni podaci ili bilo kakve grupe podataka. Povezivanje pojedinačnih podataka u triplete, kojima su predstavljene pojedine informacije, potrebno je da bi se podaci uopće mogli koristiti (1,6). Povezivanje pojedinačnih tripleta, kojima su predstavljene pojedine informacije u veće semantički logičke cjeline kao što su logički slogovi ili invertne strukture, značajno ubrzava obradu raspoloživih podataka (7). Daljim povezivanjem ovakvih cjelina u odgovarajuće strukture moguće je postići dalje povećanje brzine obrade.

VRSTE VEZA

Dva elementa strukture, koji stoje u nekom odnosu, predstavljaju uređjeni par. U strukturi podataka za nas je od posebnog značenja veza između elemenata uređenog para. Ta veza treba da nam omogući da polazeći od prvog elementa pronadjemo drugi elemenat uređenog para na mediju za memoriranje.

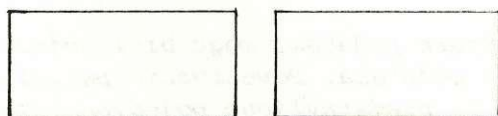
U strukturi podataka koristimo dvije vrste veza(2):

- fizičke veze i
- logičke veze.

Fizičku vezu između dva elementa uspostavljamo fizičkim redoslijedom, ili općenitije rasporedom upisa na medij za pohranjivanje podataka. Za dva elementa, koji su na mediju smješteni jedan pored drugog, kažemo da su fizički povezani. Ovakva veza je dvosmjerna. Fizičku vezu možemo rastaviti na dvije usmjerene veze od kojih jedna polazi iz prvog elementa uređenog para i ukazuje na drugi, a druga polazi iz drugog elementa i ukazuje na prvi. Uobičajeno je da se prvim elementom uređenog para smatra onaj element koji se u skladu s konvencionalnim načinom čitanja očitava prvi s medija.

Druga vrsta veze jest tzv. logička veza. Da bismo uspostavili logičku vezu, u prvi element uređenog para ugrađujemo adresu na kojoj je pohranjen drugi element para. Pod adresom podrazumijevamo takav opis određene mjesta na memoriji koji nam omogućava pronalaženje i pristup do tog mjesta. Svojstva logičke veze u velikoj mjeri ovise o obliku adrese. Međutim, logička veza je uvijek usmjerena veza. Ona nam pruža mogućnost da, polazeći od elementa u koji je adresa ugrađena, pronadjemo drugi element uređenog para. Kretanje u suprotnom smjeru logičke veze nije moguće. Na slici 1 šematski su prikazane vrste veza koje koristimo u strukturi podataka.

a) fizička veza



b) logička veza



Slika 1. Šematski prikaz vrsta veza koje se koriste u strukturi podataka.

VRSTE ADRESA U LOGIČKIM VEZAMA

Adresa u logičkoj vezi može biti dana na više različitih načina. Svojstva logičke veze u velikoj mjeri ovise o obliku adrese i zbog toga je potrebno da ukratko razmotrimo osnovne oblike koje adresa može imati. Postoje tri osnovna načina na koja može biti definirano mjesto na mediju:

- stvarna adresa,
- relativna adresa i
- simbolička adresa.

S t v a r n a a d r e s a jedne točke na mediju za memoriranje može biti dana u obliku udaljenosti te točke od početka memorije (broj adresibilnih jedinica u memoriji od početka memorije pa do te točke), ili u obliku neke vrste koordinata (cilindar, staza, slog) u koordinatnom sistemu kojim je obuhvaćen i početak memorije. Karakteristika svake stvarne adrese jest da je njome definirano određeno mjesto u memoriji s obzirom na početak memorije ili neku drugu čvrstu točku u memoriji.

R e l a t i v n a a d r e s a predstavlja u suštini udaljenost od neke referentne točke u memoriji s tim da položaj te točke u memoriji nije unaprijed određen. Tek kad znamo stvarnu adresu referentne točke, možemo iz relativne adrese izračunati stvarnu adresu. Relativna adresa nekog elementa u skupu podataka obično predstavlja udaljenost tog elementa od početka skupa. Prostor, koji na memoriji zauzimaju elementi tog skupa, redovito nije jednak prostoru koji nam u memoriji stoji na raspolaganju. Bez obzira na kojem dijelu memorije su elementi skupa pohranjeni, oni zapsjedaju uvijek isti prostor. Unutar skupa redoslijed podataka je neovisan o mjestu na kojem je skup pohranjen. Relativnom adresom, s obzirom na početak skupa, određen je položaj elementa unutar skupa. Kao relativna adresa često se koristi redni broj elementa u skupu.

S i m b o l i č k a a d r e s a bitno se razlikuje od stvarne i relativne adrese. Kao adresu elementa koristimo u ovom slučaju neki simbol kojim je element jednoznačno određen u skupu.

U tu svrhu možemo upotrijebiti svako semantičko svojstvo elementa čijim vrijednostima atributa je svaki pojedini element skupa jednoznačno određen. Često se kao simbolička adresa koristi identifikator logičkog sloga. Da bi simbol, kojim je element strukture jednoznačno određen, mogao biti upotrijebljen kao simbolička adresa, mora postojati način da se polazeći od simbola utvrdi stvarna adresa elementa. Takvu vezu između simbola i stvarne adrese možemo uspostaviti putem tabele koja sadrži popis simbola i odgovarajuće stvarne ili relativne adrese, odnosno putem algoritma pomoću kojeg iz simbola izračunavamo stvarnu adresu elementa na mediju za memoriranje.

S v o j s t v a p o j e d i n i h t i p o v a a d r e s a međusobno se dosta razlikuju. Svaki od triju mogućih oblika adrese ima određene prednosti i nedostatke. Jedino pomoću stvarne adrese moguće je pronaći element na mediju. Ukoliko imamo re-

lativnu ili simboličku adresu, morat ćemo uz pomoć određenog algoritma ili tabele odrediti stvarnu adresu prije nego što budemo u mogućnosti očitati elemenat s mjesta na koje je pohra - njen. Očito je da je pristup do određenog podatka i najjednostavniji i najbrži ukoliko je logička veza uspostavljena preko stvarnih adresa.

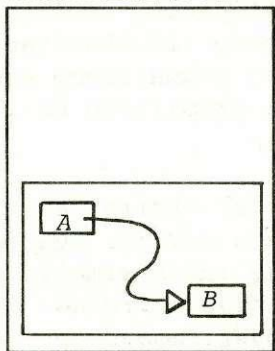
Problem održavanja logičkih veza baca, međutim, sasvim drugačije svjetlo na pojedine oblike adresa. Logička veza ukazuje na jedan određeni elemenat skupa i omogućava nam pristup do mjesta na kojem se elemenat nalazi. Ukoliko elemenat premjestimo na neko drugo mjesto u memoriji, postavlja se pitanje šta treba učiniti da bi logička veza odražavala novu situaciju. Operacije, koje treba izvršiti, u velikoj mjeri ovise o obliku adrese. Prije nego što pristupimo razmatranju samih operacija, bit će korisno da pogledamo koji razlozi mogu dovesti do promjene mjesta na koje je neki elemenat skupa odložen.

U principu postoje tri osnovna razloga koji do toga mogu dovesti:

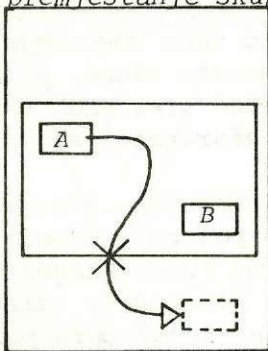
- čitav skup elemenata kao takav je promijenio mjesto na memoriji. Pri tome se redoslijed elemenata unutar skupa nije izmijenio,
- uredjenom skupu elemenata dodani su neki novi ili izbačeni neki od postojećih elemenata. Prilikom dodavanja novi element ćemo morati smjestiti na neko određeno mjesto koje mu s obzirom na uredjajnu relaciju pripada. Da bismo to mogli izvesti, morat ćemo premjestiti sve elemente, počev od tog mjesta za jedno mjesto prema kraju. Tek pošto smo na taj način oslobodili potreban prostor na memoriji, možemo upisati elemenat koji dodajemo. Prilikom izbacivanja nekog elementa iz skupa doći će do pomicanja svih elemenata koji slijede za jedno mjesto prema početku skupa,
- treći uzrok, koji može dovesti do premještanja elemenata, jest promjena uredjajne relacije kojom je utvrđen redoslijed elemenata u skupu. S obzirom da se u tom slučaju mijenja čitava struktura, ovaj slučaj ne spada u održavanje strukture, odnosno logičkih veza u strukturi. Zbog toga ovaj slučaj nećemo uzeti u obzir na ovom mjestu.

Ukoliko iz bilo kojeg razloga dodje do promjene mjesta na kojem se element nalazi, stvarna adresa elementa se promijenila i prema tome, ukoliko su logičke veze koje ukazuju na taj element dane u obliku stvarne adrese, one moraju biti ažurirane (slika 2).

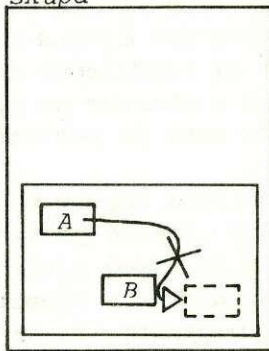
Stvarna adresa



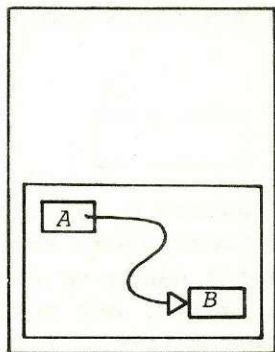
premještanje skupa



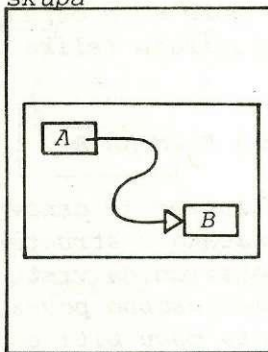
promjena unutar skupa



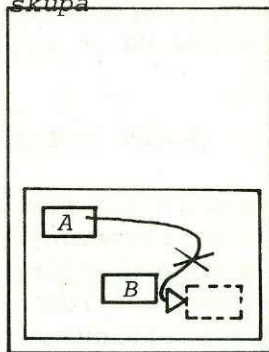
Relativna adresa



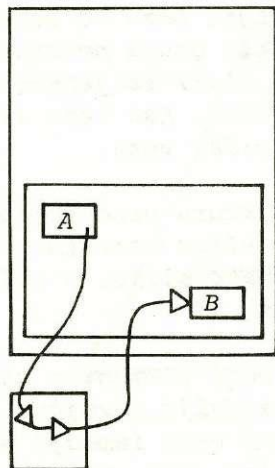
premještanje skupa



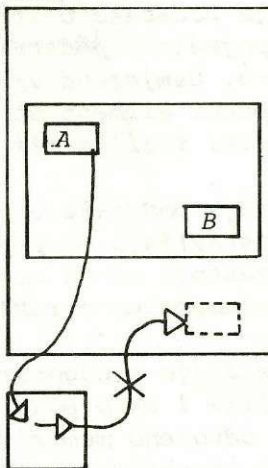
promjena unutar skupa



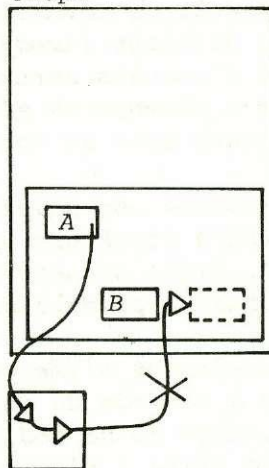
Simbolička adresa



premještanje skupa



promjena unutar skupa



Slika 2.

Ukoliko je logička veza uspostavljena pomoću relativne adrese, situacija se razlikuje utoliko što logičku vezu nije potrebno ažurirati, ako je do promjene mjesta na koje je upisan element došlo zato što je čitav skup premješten. Međutim, ukoliko je do promjene mjesta došlo zbog dodavanja novog ili izbacivanja jednog od postojećih elemenata skupa, položaj promatranog elementa s obzirom na početak uredjenog skupa promijenio se i logičku vezu je potrebno ažurirati (slika 2).

Simboličku adresu nije potrebno ažurirati bez obzira kakav je razlog izazvao promjenu položaja elementa na mediju. Ovo je, međutim, samo djelomično točno. Ažuriranje samih simbola, koji su ugrađeni u elemente strukture, zaista nije potrebno. Rekli smo, međutim, da uz sam simbol mora postojati mogućnost da se polazeći od simbola utvrdi stvarna adresa elementa na mediju. Ovisno o tome na koji način je ova veza između simbola i stvarne adrese uspostavljena može se u određenim slučajevima javiti potreba da se veze ažuriraju (slika 2).

NAČIN POVEZIVANJA ELEMENATA

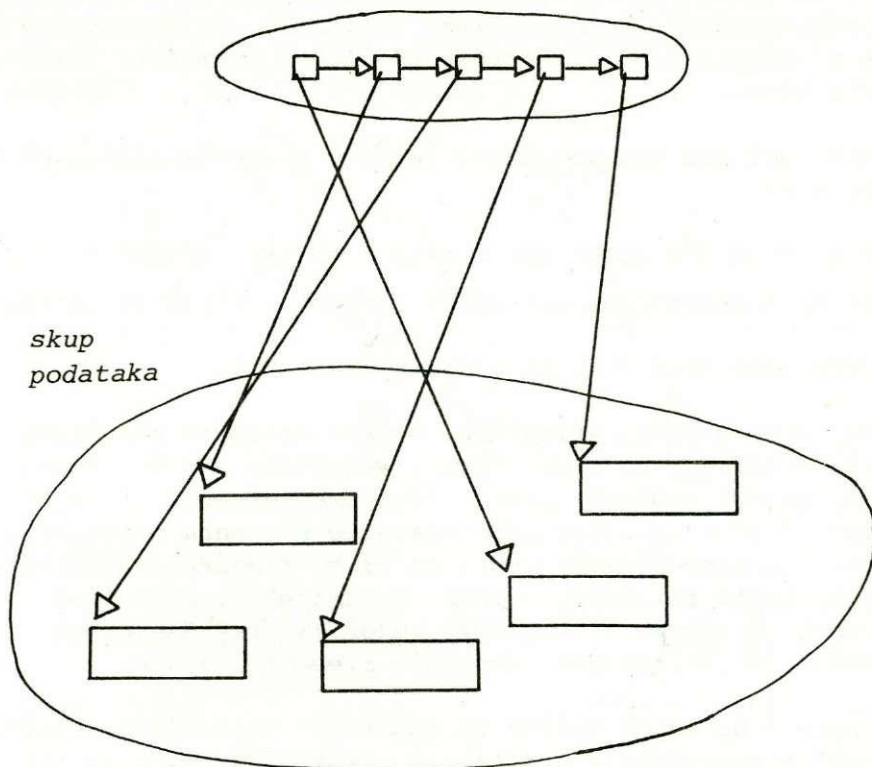
D. Severance (3) sistematizirao je osnovne faktore o kojima ovisi način povezivanja podataka u strukturu. Prema njemu strukture možemo podijeliti s obzirom na vrstu veze i način na koji su dva elementa strukture međusobno povezana. Naime, bez obzira na vrstu veze dva elementa mogu biti povezana direktno ili indirektno preko slike strukture.

Kod direktnog povezivanja podataka u strukturu veze se uspostavljaju direktno između pojedinih podataka ili grupa podataka, odnosno elemenata strukture. Usmjerena veza polazi iz jednog elementa i ukazuje na slijedeći element strukture. Kao veza između elemenata može se koristiti fizička ili logička veza.

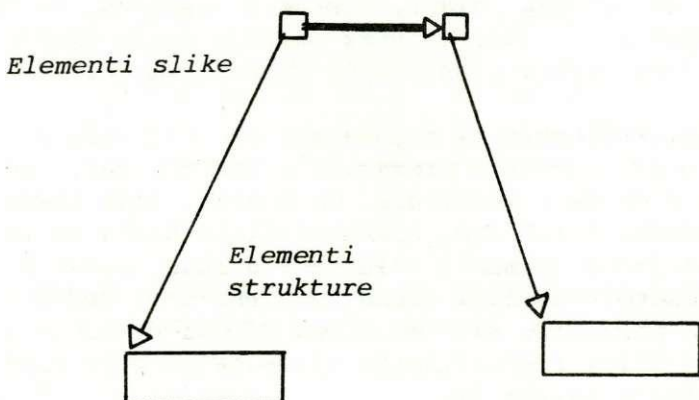
Kod indirektnog povezivanja podataka u strukturu veze između elemenata strukture uspostavljaju se preko slike strukture (slika 3). Slika strukture sastoji se od elemenata slike, a svi su jednoznačno pridruženi elementima strukture.

Svaki element slike povezan je s odgovarajućim elementom strukture. S obzirom da su slika i skup podataka dvije odvojene cjeline, koje su na mediju odvojeno memorirane, veze između elemenata slike i elemenata strukture su logičke veze. Logička veza je ugrađena u element slike i ukazuje na element strukture (slika 3).

Slika strukture



Slika 3. Šematski prikaz indirektnog povezivanja podataka.



Slika 4. Šematski prikaz indirektnog povezivanja dvaju elemenata strukture.

Veze između elemenata strukture uspostavljaju se tako da se povežu odgovarajući elementi slike (slika 4). Za međusobno povezivanje elemenata u slici strukture mogu se koristiti fizičke ili logičke veze.

Rezimirajući ono što smo dosad rekli o elementu slike, on mora sadržavati:

- adresu elementa strukture u skupu podataka (logička veza) i
- vezu koja ukazuje na slijedeći element u slici strukture.

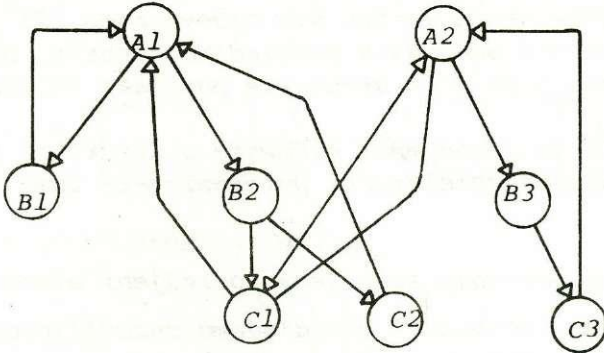
Ova veza može biti fizička ili logička.

Prateći veze u slici strukture, možemo elemente strukture obradivati u odredjenom redoslijedu. Međutim, ukoliko želimo pronaći odredjeni podatak preko strukture, element slike mora sadržavati i identifikator odgovarajućeg elementa strukture. Kao i u slučaju simboličke adrese, da bismo jednoznačno odredili sadržaj elementa strukture, možemo upotrijebiti bilo koje semantičko svojstvo elementa strukture, ukoliko vrijednosti atributa tog svojstva jedn. značno određuju element u skupu.

Osim upravo opisanog načina za uspostavljanje slike strukture indirektno povezivanje podataka u strukturu moguće je i preko tzv. bit slike. Bit slika je dvodimenzionalna matrica. Broj redaka i broj stupaca u matrici jednak je broju elemenata strukture. Po jedan redak i po jedan stupac matrice jednoznačno su pridruženi svakom pojedinom elementu strukture. Vezu između dvaju elemenata strukture prikazat ćemo tako da u presjecištu retka elementa, iz kojeg veza polazi, i stupca elementa, na koji veza ukazuje, upišemo 1. Ukoliko veza između dvaju elemenata ne postoji, u odgovarajuće presjecište upisat ćemo 0.

Redak i stupac matrice donekle bi odgovarali elementu slike strukture. Upisom u odgovarajuće presjecište registrirali smo vezu između dvaju elemenata strukture. Osim veze, koja ukazuje na slijedeći element strukture, element slike treba da sadrži adresu odgovarajućeg elementa strukture u skupu podataka i eventualno identifikator sadržaja elementa strukture. Redni broj retka mogao bi poslužiti kao relativna adresa elementa u skupu podataka. Međutim, identifikator elementa ne može biti sadržan u samoj matrici (slika 5).

Elementi povezani u strukturu



Redoslijed upisa elemenata na memoriju



Bit slika

	1	2	3	4	5	6	7	8
1	0	0	1	1	0	0	0	0
2	0	0	0	0	1	1	0	1
3	1	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	0
5	0	0	0	0	0	0	0	1
6	1	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0

prateća tabela

A1	1
A2	2
B1	3
B2	4
B3	5
C1	6
C2	7
C3	8

Slika 5. Šematski prikaz strukture, rasporeda elemenata na memoriji i odgovarajuća bit slika s pratećom tabelom.

PODJELA STRUKTURA PREMA VRSTI VEZA I
NAČINU POVEZIVANJA ELEMENATA

Vrsta veze kao i način povezivanja elemenata u strukturu značajno utječu na mogućnosti obrade podataka u strukturi kao i na mogućnosti održavanja strukture. Podjela strukture u četiri osnovne grupe, koju je izvršio D. Severance (3) na osnovu spomenutih faktora omogućava pregled mogućnosti, odnosno procjenu svojstava postojeće strukture podataka (slika 6).

Da bismo došli do određenog elementa u strukturi, moramo slijediti veze kojima su elementi povezani. Pri tome razlikujemo dva slučaja:

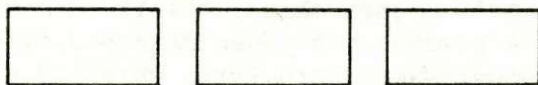
- za obradu je potreban samo jedan određen element i
- sve elemente u strukturi obrađujemo redoslijedom kojim su saržani u strukturi.

Kod direktnog povezivanja podataka u strukturu veze su uspostavljene direktno između elemenata strukture. Da bismo slijedili te veze, moramo se kretati od jednog elementa strukture do drugog. Zbog toga direktno povezivanje podataka je u principu pogodnije kad se obrađuju svi elementi strukture. Da bismo u takvoj strukturi došli do nekog određenog elementa, morat ćemo ispitati i prema tome na neki način obraditi sve elemente koji mu u strukturi prethode. Kod indirektnog povezivanja podataka u strukturu veze među elementima strukture nalaze se u slici strukture. Obradom slike strukture ustanovit ćemo gdje se nalazi jedan određen element strukture. S obzirom da je slika strukture relativno mala u usporedbi sa skupom podataka, njena obrada je u principu jednostavnija. Prema tome, pronalaženje određenog elementa u strukturi s indirektno povezanim podacima, nešto je brže nego kod odgovarajuće strukture s direktno povezanim podacima. S druge strane, obrada svih elemenata strukture svodi se na niz uzastopnih pronalaženja određenog elementa strukture (element koji u strukturi slijedi iza upravo obrađenog). Ovakva obrada može u najboljem slučaju biti jednako brza kao i kod strukture s direktno povezanim podacima, međutim u pravilu će biti nešto sporija.

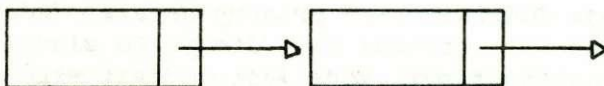
Veze među elementima strukture, kod direktnog povezivanja, s elementima slike strukture kod indirektnog mogu biti fizičke ili logičke. Jedini način da dodjemo do određenog elementa jest da pratimo veze među njima. Ovakva obrada svodi se na obradu svih elemenata prema određenom redoslijedu. Prema tome, kad govorimo o svojstvima veza i njihovom utjecaju na brzinu obrade, jedino što ima smisla to je procjena vremena potrebnog da bismo pronašli element na koji ukazuje usmjerena veza.

Direktno povezivanje podataka

a) fizičkim vezama

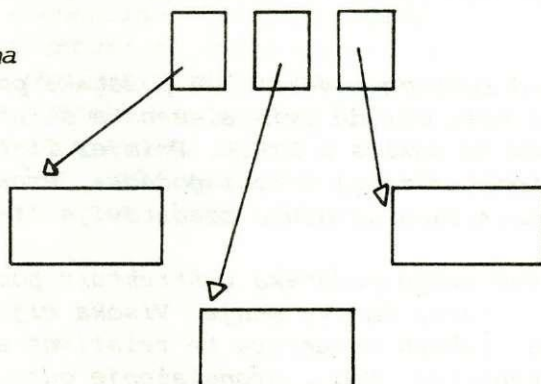


b) logičkim vezama

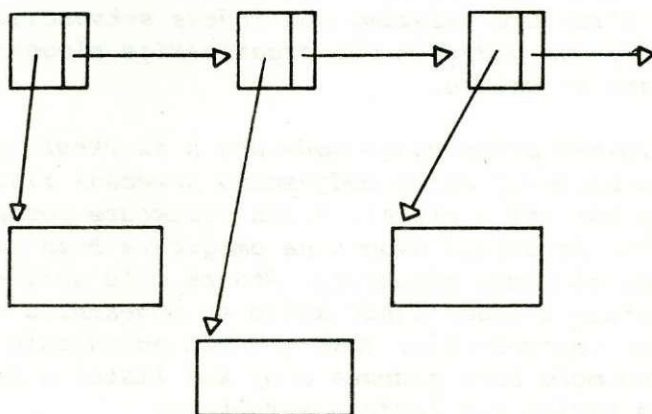


Indirektno povezivanje podataka

c) fizičkim vezama



d) logičkim vezama



Slika 6. Šematski prikaz četiriju teoretski mogućih načina povezivanja podataka u strukturu (prema Douque, 1975).

Karakteristika fizičke veze jest da je element koji slijedi ujedno i slijedeći element na mediju za memoriranje. Zahvaljujući tome obrada svih elemenata koji su povezani fizičkim vezama predstavlja najbržu moguću obradu. Veliki nedostatak fizičkih veza jest da je pomoću njih jedan skup podataka moguće povezati samo u jednu odredjenu strukturu, te da je neke strukture teško prikazati pomoću fizičkih veza (4).

U slučaju logičkih veza pristup do slijedećeg elementa zahtijeva znatno više vremena nego što je to slučaj kod fizičkih veza. Velika prednost ovih veza jest da isti skup podataka može biti povezan u niz različitih struktura, te da se pomoću logičkih veza može prikazati bilo koja struktura. Pri tome treba imati na umu da je uspostavljanje i održavanje logičkih veza relativno skupo.

Kombinacija direktnog povezivanja podataka pomoću fizičkih veza omogućava najbržu obradu svih elemenata strukture. Ovaj tip strukture često se naziva i LISTA. Primjer liste vrlo često možemo sresti u automatiziranoj obradi podatka. Praktično svaka sekvencijalno organizirana datoteka predstavlja listu.

Direktno povezivanje podataka u strukturu pomoću logičkih veza koristi se u praksi daleko manje. Visoka cijena uspostavljanja i održavanja ovakvih struktura te relativno spora obrada svih elemenata strukture, kao i pronalaženje odredjenog elementa u strukturi, razlog su da se ovakve strukture koriste samo u nekim posebnim slučajevima. Ovakva struktura inače se u literaturi često spominje pod nazivom LANČANA STRUKTURA. Primjer lančane strukture nalazimo kod indeks sekvencijalne organizacije. U rezervnom području ove organizacije slogovi su povezani u lančanu strukturu.

Indirektno povezivanje podataka u strukturu preko slike, pri čemu su elementi slike međusobno povezani fizičkim vezama, mnogo se koristi u praksi. Slika strukture poznata je pod nazivom INDEKS. Struktura ovog tipa omogućava brzo pronalaženje odredjenog elementa strukture. Što se tiče obrade svih elemenata strukture u redosljedu kojim su povezani u strukturu ona je nešto nepovoljnija. Iako u nekim odredjenim slučajevima ta brzina može biti jednaka onoj kod liste, u pravilu ona je ipak bliža brzini kod lančane strukture.

Kombinaciju indirektnog povezivanja podataka i logičkih veza medju elementima slike strukture u praksi možemo sresti jedino u strukturalnoj datoteci banke podataka BOMP i njoj vrlo srodnim DBOMP i TOTAL. Premda ovakvo povezivanje podataka pruža

moćnost prikazivanja nekih tipova struktura koje bi na drugi način bilo teško prikazati, mala brzina obrade svih elemenata strukture i vrlo sporo pronalaženje određenog elementa u strukturi razlog su da se ovakva struktura u praksi izbjegava. Naime, slika strukture u ovom slučaju po veličini može biti jednaka skupu podataka. Ovakvu sliku strukture nije moguće učitati i obradivati u glavnoj memoriji kompjutera, čime su u stvari izgubljene sve prednosti indirektnog povezivanja podataka.

Indirektno povezivanje podataka u strukturu preko bit slike pruža mogućnost prikazivanja i najkompliciranijih veza među elementima strukture. Ipak bit sliku relativno rijetko možemo susresti u praksi obrade podataka. Premda je cijena uspostavljanja i održavanja veza među elementima strukture relativno niska, promjena broja elemenata u strukturi je dosta komplicirana jer se pri tome mijenja veličina matrice koja predstavlja bit sliku. Ako se ovom nedostatku doda potreba za uspostavljanjem prateće tabele, čini mi se da se može objasniti relativno mala primjena bit slike. Ipak, mislim da bit slika zavredjuje da se ispita mogućnost njene primjene u određenim slučajevima.

L I T E R A T U R A

1. Bachman, C.W.: *Data Structuring Facilities in Commercially Available DBMS*, Computer Bulletin, september (1974) 20.
2. Domes, J. i Tkalac, S.: *Struktura podataka*, Informatologia Yugoslavica, 7 (1975) 49.
3. Douque, B.C.M.: *Pholas, a Modular Implementation of the DBTG Proposals*, u *Data Base Systems* (Editor Bunjan, C.J.), str.331, Infotech Information Ltd, Nicholson House, Maidenhead, Berkshire, England, 1975.
4. Engles, R.W.: *A Tutorial on Data-Base Organization*, IBM Systems Development Division, 1970, New York, N.Y.
5. Langer, S.K.: *The Study of Forms*, u *An Introduction to Symbolic Logic*, str.21, Dover Publications, Inc., New York, N.Y., 1967.
6. Tkalac, S.: *Entitet, informacija, podatak*, Informatologia Yugoslavica, u štampi.
7. Tkalac, S.: *Logički slog i invertna struktura*, Informatologia Yugoslavica, u štampi.