

1. Skaiciavimo sistemos.

Skaiciavimo sistema yra visuma budu ir priemoniu, leidzianti uzrasyti ar kitaip pateikti skaicius.

Naudojamos yra kelios skaiciavimu sistemos: dvejetainė, astuntainė, desimtaine, sesioliktaine.

Dvejetainė - turi du simbolius, yra elektroniniu schem. pagrindas, nes tik tokia schem. leidzia skaitinemis reiksmemis nustatyti el. shem. buvi. Pvz.: kruvio buvimas gali reiksti 1, o nebuvimas 0 (ir atvirksciai).

Desimtaine - yra pagrindine, visuotinai priimta sk. sistema (turi 10 skirtingu simboliu).

Sesioliktaine - turi 16 simboliu, dazniausiai naudojama skaiciu pervedimui tarp dvejetaines ir desimtaines sk. sistemu. Sesiolika simboliu iki devinto yra parasomi skaiciumi, o toliau raide (A, B, C, D, E, F).

Astuntaine - yra analogiska sesioliktainei, gali buti naudojama pervedimui dvejetainio skaiciaus i desimtaine sistema, taciau yra 8 simboliai, o pervedinejant is BIN i DEC yra naudojamos triados, bet ne tetrados.

Bet kurioje skaiciavimo sist. gali buti atliekami aritmetiniai veiksmai (sudetis, atimtis, dalyba, daugyba ir t.t.), ir bet kurioje skaiciavimo sist. tas pats skaicius gali buti uzrasomas skirtingai, taciau skaiciaus dydis lieka tas pats.

2. Skaiciu pervedimas is vienos skaiciavimo sist. i kita.

Norint bet kurioje skaiciavimo sistemoje, bet koki skaiciu perversi i kita skaic. sist., naudinga pirmiausia ji pasiversti i dvejetaini, o tik paskui i norima skaic. sistemos skaiciu.

Verta zinoti, kaip is desimtaines skaiciavimu sistemos pasiversti dvejetainiu. Reikia atlikti paprasta dalybos veiksmas is dviejų. Pvz:

DEC '12' = BIN '1100' --> 12:2=6 ir liekana 0 arba 12 | 0 --> BIN '1100'

6:2=3 ir liekana 0 6 | 0

3:2=1 ir liekana 1 3 | 1

1 1

Dvejetainis skaicius uzrasomas nuo galo.

Skaiciai is astuntaines ir sesioliktaines skaic. sist. yra pervedami atitinkamai triadomis ir tetradomis. Pvz:

HEX '1F3' --> 1 F 3

BIN '000111110011' --> 0001 1111 0011

OCT '123' --> 1 2 3

BIN '001010011' --> 001 010 011

Pervedant is vienos sk. sistemos i kita skaicius, kuriuos sudaro sveikoji ir trupmenine dalys, ir sveikoji ir trupmenine dalys pervedamos atskirai. Visada galioja budas, jog is bet kokios sistemos reikia pasiversti i BIN, o paskui i norima skaic. sist.

Tegu duota trupmena DEC '0.257'

.0257 x 16 = 4.112 = 4 + .112,

.112 x 16 = 1.792 = 1 + .792,

.792 x 16 = 12.672 = C + .672,

.672 x 16 = 10.752 = A + .752,

.752 x 16 = 12.032 = C + .032

DEC '.257' = HEX '.41CAC...' = BIN '.100 0001 1100 1010 1100'

Dazniausiai tenka apvalinti trupmenas, nes pervedimas gali testis neribotai.

3. Dvejetainė aritmetika.

Dvejetaines aritmetikos pagrindas - sudetis, atimtis, daugyba.

Dvejetainiu sk. daugybos ir dalybos veiksmai yra analogiski desimtainiu sk. daugybos ir dalybos veiksmais.

Sudedant skaicius vieneto pernesimas vyksta i didesniaja (i kaire) tada, kai sudedama 1+1=10.

Skaicius atimant, prireikus vyresniosios skilties vieneta pernesti i desiniaja skilti (pasiskolinti), pastarojoje skiltyje atsiranda du vienetai (t.y. 10), o tarpinese nulinese skiltyse - po viena vieneta, nes kitas vienetas

keliauja iki maziausios skilties i kuria skolinamasi. Todel pastarojoje skiltyje pries atimant, atsiranda du dvejetainiai vienetai.

4. Logines operacijos.

Pagrindines logines operacijos yra tiesa, kuri zumima skaiciumi 1, ir netiesa - skaiciumi 0.

Logines operacijos vadinamos ARBA (logine sudetis, disjunkcija), IR (logine daugyba, konjunkcija), ISSKIRTINE ARBA (neekvivalentumas), NE (loginis neiginys, inversija).

Logines operacijos zymimos simboliais:

ARBA - v, IR - ^, ISSKIRTINE ARBA - (+), NE - \

Logines operacijos rezultatas gali buti tik tiesa 1, arba netiesa 0.

Logineje operacijoje ARBA, bitu palyginimo rezultatas visais atvejais yra 1, isskyrus ta atveji, kai yra palyginami nuliai, tada rezultatas bus 0. Logines operacijos IR, visais bitu palygino atvejais rezultatas yra 0, isskyrus ta atveji, kai palyginami yra vienetai, tada rezultatas bus 1.

Logineje operacijoje ISSKIRTINE ARBA lyginami skirtingi baitai, turi rezultata 1, o vienodi 0.

Loginio neiginio NE rezultato analogiskose vietose vienetai yra pakeiciami nuliais, o nuliai vienetais.

Kompiuterio atmintyje esanciu dvejetainiu strukturu tiesiogiai vizualiai matyti yra neimanoma, tai yra daroma operacijai pateikiant sablona, t.y. is anksto zinoma dvejetaine struktura. Paskui pagal logines operacijos rezultata, spresti, kas ten buvo.

5. Loginiu operaciju pavyzdziai:

v ARBA	^IR	(+)ISSKIRTINE ARBA	-\NE
00001111	10010010	01111101	01101011
10000000	00000000	11111111	-----
-----	-----	-----	10010100
10001111	11111001	10000010	

6. Dvejetaines operacijos diskreciuose kompiuteriu elementuose.

Kompiuteriu elementai nuolat kinta, tobuleja, o dvejetaine aritmetika lieka ta pati. Diskretusis elementas vaizduojamas schema, kuria sudaro elektros grandine, susidedanti is magnetiniu (a,b) ir elektromagnetiniu itaisu (A,B), kontaktu ir elektros sroves saltinio, bei isejimo, operacijos isejimo tasko C.

Dvejetaine informacija schemai pateikiama elektros sroves buvimu arba nebuvimu A ir B taskuose. Itamos buvimas taskuose A, B, C reiskia 1, o nebuvimas 0.

7. Operacijos ARBA, IR schema.

ARBA

IR

8. Informacija.

Informacija - tai ziniu apie pasauli visuma. Ta visuma sudaro zinios apie ivairiausius objektus, nuo elementariausiu iki labai sudetingu. Informacija gali buti skirstoma ir rusiuojama pagal ivairius pozymius.

Pagal apdoravimo periodus, ji gali buti pradine, tarpine, rezultatine, kuri skirta galutiniam jos naudojimui. Pagal sprendziamu uzdaviniu tipus, inf. gali buti operatyvi, uzdara, komercine, moksline, ekonomine, filosofine ir t.t. Pagal reiksme - teisinga, klaidinga, pilna, nepilna, priestaringa, nereikalinga, ir kt. Vieno ir to paties pavadinimo inf. gali atitikti ne vienam, okeliems rusiavimo kriterijams.

Galutiniu vartotoju laikomas asmuo arba organizacija, turintys savo inf., kuria reikia, butina arba tikslinga apdoroti kompiuteriu. Tokia inf. negali buti pilnai apdorujama esama programine sisteme, taikomaja, tipine ar kitokia iranga.

Tokios inf. apdorujimui reikalinga sukurti specialia progr. iranga, kuri savarankiskai issprestu minetos inf. apdorujimo uzdavinius.

Informacijos pateikimoforma yra lenteles. Jomis pateikiama informacija nesunku transformuoti i matem. konstrukcija - reliacines aibes. reliacines aibes israiska turi didele dalis technines, technologines, ekonomines, mokslines ir kt. informacijos.

9. Informacijos apdorujimo lygiu samprata.

Informacijos apdorujimo lygiu laikytini tokie atvejai:

1) kompiuterinis informacijos apdorujimas, kuris inf. nekeicia, naujos informacijos negaunama, komp. naudojamas, kaip fizinis irankis zm. darbui tobulinti.

2) Vartotojui pateikiama kito vartotojo informacija, apdorujant ja i kitam vartotojui reikiama forma, eile, apdorujimo trukme ir pan.

3) Inf. apdorujimo rezultatas yra nauja inf., kuri isreiskiami skaiciais, zodziais, simboliais, kodais, kurie yra zinomu ir nauju atributu reiksmes.

Informacijos apdorujimo lygiai yra salyginiai. Kompiuteris yra tik irankis, kuris ivairiais aspektais palengvina zm. darba. Kompiuterine sistema jokios naujos inf., siuo atveju, sukurti negali.

10-11. Duomenys, pareikstys.

Duomeniu laikoma pareikstis, kuri pateikiama logiskai savarankisku, daugiau nedalomu sakiniu. Toks sakiny turi logini veiksmi ir pagrinda, bei paziminius ir papildinius.

Pavyzdys: "INSTITUCIJOJE b,d METU PRADZIOJE DIRBO c DARBUOTOJU, KURIE SIOJE INSTITUCIJOJE TURI e METU DARBO STAZA".

Cia loginis veiksmys yra zodis DARBUOTOJU, o pagrindas - c. Pateiktoji pareikstis yra logiskai savarankiska ir negali prarasti ar igyti nauju pozymiu ar papildiniu.

12-13. Duomenu identifikavimas, duomenu lenteles.

Jeigu turimas ne vienas domuo, o keletas ar keliolika duomenu apie institucijas bei ju darbuotoju darbo staza, tai galima sudaryti tokiu duomenu hipotetine lentele. Sudarant tokia lentele, pirmiausia yra isskiriami bendri visu duomenu pozymiai. Tokiu pozymiu galetu buti laiko momentas (metu pradzia). Sis pozymis yra uzrasomas lenteles priekyje ir tarnauja

lenteles identifikatoriaus sudetine dalimi.

Kita lenteles identifikaoriaus sudetine dalimi galetu buti subjektas, kurio duomenims skir- ta lentele. Tai instituciju visumos tam tikra dalis: kurio tai verslo visos institucijos.

Po to nustatomi ir koduojami tam tikra eile surasyti atributai.

Taigi identifiukuota lenteles schema sudarys identifikatoriaus atributu schema ir pacios len- teles schema.

Identifikatoriaus schema:

a - pacios lenteles, kaip savarankisko duomenu darinio, kodas;

B - subjekto atributas - nuoroda, kad butu pateikta, kokia veiklos srity apima institucijos, kuriu duomenys pateikiami lenteleje.

D - laiko faktoriaus kodas, siuo atveju reiskiantis metu pradzia, o pildant lentele nurodoma koku metu pradzios duomenys yra lenteleje.

Duomenu lenteleje eilutes gali buti isdestytos ivairia tvarka: pagal kurio nors atributo skaitiniu reiksmiu didejima arba mazejima, pagal tekstiniu atributu reiksmiu pirmuju raidziu abecele ir t.t. Nuo to duomenys nesikeicia. Butina turti galimybe viena eilute atskirti nuo kitos. Tam reikia viena is atributu nurodyti kaip rakto atributa ir jo reiksmes tampa eiluciu raktais, t.y. identifikatoriais. Jeigu del kokios nors klaidos atsitinka taip, kad lenteleje yra dvi ar daugiau to pacio rakto ir vienodu kitu atributu reiksmiu eilutes, tai visos eilutes isbraukiamos is lenteles paliekant tik viena is ju.

Tokia klaida surasti ir istaisyti tik siu duomenu savininkas ir yra butina kreiptis i pirmini duomenu saltini.

Duomenys, kuriu schema ir jos atributai turi realias atrib. reiksmes, gali buti identifikuojami.

Jeigu duomenu lenteles identifikavimui pakanka tik kodo a, tai reiskia, kad tos lenteles duomenys nepriklauso konkrečiam subjektui ir nepriklauso nuo laiko faktoriaus. Tai a yra sudaruta is ivairiu konstantu ir/arba normatyviniu duomenu, nes normatyviniai duomenys yra salyginai pastovus.

14. Algoritmai ir programos.

Duomenys apie duomenis yra metaduomenys.

Nuorodu ir taisykliu visuma, kuri apibrezia duomenu apdoravimo procesa taip, kad is pradiniu duomenu butu gaunami duomenys - apdoravimo rezultatai, vadinama algoritmu. Algoritmas netures jo- kios itakos duomenu apdoravimui, kol jis nebus isreikstas programa, o pastaroji nebus pateikta dvejetainine forma

ir irasyta i komp. atminti. Taigi kompo atmintyje gali buti tik 2 is esmes besiskirianciu duomenu rusys:

-duomenys apie realius objektus;

-duomenys apie duomenis arba programas, igalinancios duomenis perdaryti, apdoroti, ivesti, isvesti is kompo atminties ir t.t.

Programa vadinami duomenys, kuire skirti duomenu apdoravimo komponentams valdyti ir algoritmui ivykdyti.

Duomenu apdoravimo programoje turi buti nuorodos i kitas, bendras daugeliui duomenu apdoravimo uzdaviniu programas,

kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomenu ivestis, isvestis,

duomenu ir programiniu veiksmu korektiskumo kontrole, prog. ir duomenu apsauga nuo istrinimo ir t.t.

Kad bendro pobudzio darbai ruosiant programas atkristu, yra sudarytos programines bendrojo pobudzio sistemos,

kuriu aptarnauja taikomojo pobudzio programas. Tos sistemos yra operacines sistemos.

15. Atminties adresacija ir duomenu strukturos.

Kompiuterio atminti pakanka suvokti, kaip dvejetainiu skilciu ar ju grupiu vektoriui. 8 skilciu grupe sudaro maziausia adresuojama atminties lastele- baita.

Duomenų baito, iraso, lauko, failo, masyvo ar kitos struktūros turini galima nustatyti tik tokiu atveju, jeigu žinomas struktūros pradžios adresas atmintyje.

Duomenų irasų laikomas duomenų sankaupos (failo, lauko, masyvo) elementas, kurio vidiniu daliu duomenų tipai gali būti skirtingi.

Duomenų failu (byla, rinkmena) laikoma identifikuota logiskai ir/ar fiziskai susietu irasų aibe.

Duomenų masyvu laikoma ivardinta vienodo tipo irasų aibe, kur irasai identifikuojami indeksais.

Duomenų lauku laikoma logiskai ir/ar fiziskai susietu failo ar masyvu irasų ivardinta dalis, arba irasams skirta bet dar neužpildyta atminties dalis.

Jeigu duomenų lauka sudaro tam tikra iš duomenų sankaupos išrinkta irasų aibe, bet pateikiama ne tokia eilė, kokia buvo minėtose duomenų struktūrose, tai tokia aibe vadinama duomenų rinkiniu.

Kai nėra svarbu, ar nagrinėjami duomenys yra failas, masyvas, laukas ar rinkinys, toks ivardintas duomenų derinys - duomenų sankaupa.

Bet kokios duomenų sankaupos gali būti surastos ir pateiktos apdorojimui, jei žinomi tu duomenų struktūrų adresai.

Paprastai tai būna tu struktūrų pirmųjų baitų komp. atminties adresai.

Absoliutus adresas yra failų adresai, kurie yra dvejetainiai komp. atminties adresai.

Programoms pateikti absoliučius adresus naudojam tam tikras komp. programos, kurios tiesiogiai nedalyvauja

vartotojo uždavinio sprendime, bet aptarnauja kompiuterinį to sprendimo procesą.

16. Irasų formatai.

Viena iš tų patų duomenų failų ar masyvų gali sudaryti arba pastovaus, arba kintamo ilgio, skaičiuojant simboliškai ar baitais, irasai.

Tam tikri duomenys, sudaryti iš irasų apie realius objektus, turi vienodą simboliu skaičių.

Praktikoje, dažniausiai, kad irasai turėtų pastovų ilgį, toks ilgis padaromas dirbtinai.

Kad būtų gautas max duomenų ilgis, jo pradžioje arba pabaigoje papildomai irasomi specialūs duomenų pailginimo sim-

boliniai - protarpiai (). Pvz.:

Anita_
_123456

Mantas_

Kintamojo ilgio irasas paprastai nėra papildomas jokiais simboliškai, bet dažniausiai reikalauja tarnybinių ženklų,

kurie atskirtų irasų dalis nuo kitos dalies. Pvz.: 123|123|234|ts_|kak|234

Pastovaus ilgio irasus lengviau ir greičiau galima rasti, ir suradus klaidingą irasą, jį lengva pakeisti teisingu. Tačiau pastovaus ilgio irasai neracionaliai panaudoja komp. atmintį. Bet yra galimybė, kad ilgesni laika juos naudojant,

juo max ilgis gali pasikeisti.

Kintamojo ilgio irasų naudojimas, pastovaus ilgio irasams būdingu trukumu neturi. Paminint neigiamus aspektus, tokie irasai

privalo turėti skiriamąjį ženklą tarp kitų irasų (/). Taisant klaidą, jeigu naujojo (teisingo) irasų ilgis mažesnis už

klaidingo irasų ilgį, tai duomenų sankaupoje susidaro "skylių", užpildytos bereiksmiais simboliškai. Kitu atveju būtinas

duomenų, einančių už šio irasų, postumų, kad teisingas irasas tilptų.

Naudojant kintamo ilgio irasus prireikia duomenų sankaupas persiuntinėti iš vienos atminties vietos į kitą.

Kitas neigiamas

kintamo ilgio irasu aspektas yra tas, kad kiekviena karta ieskant naujo iraso, jo ilgi reikia nustatyti programiniu budu.

Nuo atminties strukturos priklauso keletas svarbiu duomenu apdoravimo charakteristiku: duomenu suradimo greitis, ju apdoravimo laikas ir kt.

17. Operatyvioji ir isorine kompiuterio atmintis.

Bendra ju paskirtis yra isiminti duomenis.

Operatyviojoj atminty esanti informacija, isjungus komp., dingsta, o isorineje issaugoma. Tad tesiant darba, tai yra toliau

apdorojant duomenis, tarp operatyviosios ir isorines atminties vyksta keitimasis duomenimis.

Reikalingos specialios programines pastangos, kad galima butu apdoroti duomenis, kai jie visi i operatyviaja atminti netelpa.

Duomenu sankaupos nuolat dideja. Vis didesne operatyviosios atminties dali uzima operacines sistemos, bei kitos sistemines programines priemones.

Operatyviosios ir isorines atminties savybes:

Paskirtis, struktura, talpa, duomenu isiminimo greitis, irasu issidestymas, irasu paieska.

Operatyviosios atm. paskirtis - tiesiogiai dalyvauti duomenu apdoravimo procese. Bet kurie pakitimai ivykstantys apdoravimo

metu duomenyse fiksuojami operatyv. atm., kad butu galima toliau keisti, toliau juos apdoroti. Duomenu apdoravimo procese

dalyvauja ir isorine atmintis. Jos paskirtis - issaugoti duomenis, kai del ivairiausių priezasciu duomenu apdoravimo procesas

pertraukiamas.

Operatyviosios atminties struktura pakanka suvokti, kaip vientisa duomenu vektoriu. Ji sudaro simboliai, skaičiai, raidės,

irasyti i atminties baitus, kurie adresuojami naturniais skaiciais. Nera svarbu, kokio pobudzio ar turinio duomenys irasyti

i atminties baitus.

Isorines atm. struktura nera vientisa. Ji turi tam tikrus atminties vienetus. Pvz., diskines atminties vienetu gali buti

sektorius su fiksuotu - vienodu talpumu (256 baitai).

Oper. atm. talpa mazesne uz isorine atminti. Duomenu isiminimo greitis isorineje atminty yra mazesnis uz operat. atm.

greiti.

Duomenu irasu isdestymui isorineje atminty ypac svarbu vaidmeni vaidina irasu ir isorines atm. vieneto ilgio (talpos)

santykis. Isorines atm. vienetai identifikuojami naturniais skaiciais. Todel iraso ilgis ir isor. atm. vieneto ilgis su

tampa. Siuo atveju irasu paieska yra lengva. Dazniausiai paieska atliekama isorines atminties masyvus ar failus pernesus

i operatyv. atm. pagal duomenu irasu turini.

18. Duomenu sankaupu sudarymo budai ir irasu paieska.

Duomenu sankaupu sudarymo budai reiskia duomenu irasu sankaupose isdestyma taip, kad duomenis apdorojanciu uzdaviniu

programos reikiamus irasus galetu ef. surasti. Tai reiskia, kad reikiami irasai butu randami tokiu greiciu, kad galetu

issprestu uždavinį vartotojui reikiamu laiku ir sprendžiamas uždavinys netrukdytu kitu uždaviniu savalaikio sprendimo.

Duomenų saukaupų sudarymo metodai ir būdai priklauso nuo daugelio ir įvairių faktorių: pagrindinis - kiek ir kokie užd.

sprendžiami, naudojant vieną ir tą patį duomenų saukaupą. Jei duomenų apdorojimui iš anksto nėra nustatytas uždavinys

kiekis, be to tas kiekis gali keistis, tai duomenų organizavimo būdai ir metodai labai skiriasi nuo ankščiau paminėtų, o

tokiu duomenų saukaupa vadinama duomenų baze.

Metodus bei būdus, kurie yra naudojami duomenų saukaupos struktūrai sudaryti, lemia duomenų apdorojimo algoritmas

asociacijos su pradiniu duomenų įrašais. Jeigu algoritmas toks, kad apdorojami įrašai visi ir paeiliui, tai saukaupos

struktūros sudarymo būdas trivialus. Iš problemines sritis imami įrašai ir be jokių papildomų veiksmų išimami visi iš

eilės išvardintoje atm. vietoje. Šiuo atveju įrašai paprastai apdorojami ten, kur jie įrašyti. Analogiškai elgiamasi, jei

vienos saukaupos duomenis naudoja dviejų ar kelių uždavinių spr. programos. Bet yra atveju, kai formuojant duomenų saukaupa

racionali duomenis suskirstyti saukaupoje į iš eilės einančius įrašus, kiekvienam uždaviniui atskirai.

Vieni duomenų saukaupos įrašai naudojami tik konkrečiam uždaviniui spresti, o kiti keliems uždaviniams spresti. Be to tie

įrašai duomenų saukaupoje išimami tokia eile, kokia jie buvo gauti. Įrašai gali būti atskirti vienas nuo kito tik pagal

savo turinį, t.y. jie gali būti identifikuojami pagal pareikštą veiksnio, požymio ar papildinių kodus, duomenų lentelė

identifikatorius arba duomenų lentelės kortezų raktus. Tai priklauso nuo to, kas iš duomenų strukt. konkrečių atveju

laikomas įrašas: pareikštis, lentelė ar kortezas.

Jei duomenų saukaupos pakankamai didelės ir gali trikdyti uždavinio sprendimą, tai būtinas vienoks ar kitoks metodas

vienokiems ar kitokiems paieškos adresų laukams sudaryti.

Formuojant duomenų saukaupa, kiekvienas įrašas, kuris įrašomas į saukaupai skirtą atminties lauką, įgyja savo adresą. Tas

adresas paprastai būna įrašo pirmojo simbolio atm. baito adresas. Tolesniame duomenų apdorojimo procese jau galima operuoti

tik su įrašų adresais. Prireikus įrašų, imami jų adresai ir iš karto randamas eilinis reikiamas įrašas. Jei iš anksto pagal

įrašo identifikatorius žinoma, kokiems uždaviniams kokie įrašai reikalingi, tai įrašų adresai surasomi iš atskiriems

uždaviniams skirtus įrašų adresų laukus. Šiuose laukuose įrašų adresai gali ir kartotis. Tai yra vienas ir tas pats įrašas

panaudojamas ne vieno uždavinio sprendime, o ir tuose sprendime, kur jo adresas adresu lauke kartojasi.

Jei kai kuriems uždaviniams spresti reikalingi įrašai turi būti surusiuoti pagal koki nors požymį, pvz, pagal identifikatoriaus didejimą, tai iš adresų lauką galima surašyti įrašus kartu su rusiavimo požymiais ir tuos

adresus

surusiuoti. Šitokio duomenų saukaupos paruosimo dažniausiai pakanka įvairiems duomenų apdorojimo uždaviniams spresti, kai

vienas ar keli užd. turi jiems skirtą pradinių duomenų saukaupa. Adresų pasikeitimas reiškia, kad tam tikra saukaupos įrašų

dalys perkeliama į kita atm. vieta. Suprantam, kad turi keistis ir uždaviniams skirtu irasu adresu lauku turinys.

Pastarieji

duomenų surinkimo organizavimo būdai taikomi tik tada, kai adresu surinkimo yra salyginai didelės.

Schematiškai adresu

pasikeitimus galima išreikšti taip, kaip paprastumo dėlei imama tik vieno iraso korekcija ji pailginant.

Esant daugeliui korekcijų ir didelėms duomenų surinkimo duomenų apdorojimo užd. spr. laikas gali neleistinai pailget ir

trukti sistemos darbui. Tad racionaliausia duomenų surinkimo naudoti tik vieno užd. sprendimui, o ta surinkimo sudaroma tik

tam užd. ir to pacio užd. programomis arba, jei tai tikslinga, naudoti duomenų bazę, bei jos duomenų organizavimo būdais.

19. Duomenų bazės.

Duomenų bazė tai tarp savęs susietų duomenų rinkinys, kuriuo gali naudoti daugelis programų įvairiems duomenų apdorojimo

užd. sprendimams. Iš duomenų bazės apibr. matyti, kad:

- duomenų bazėje kaupiant duomenis kontroliuojamas jų teisingumas, pilnumas, atliekamos korekcijos, nepriklausomai nuo

duomenų apdorojimo uždavinių ir jų programų;

- atsiradus naujiems užd., kai jų spr. truksta duomenų, duomenų bazė papildoma naujais duomenimis irgi tiesiogiai

nepriklausomai nuo naujų užd., bet taip, kad korektiškai duomenų uždaviniams sprendimui.

Duomenų bazės tikslai - išvengti tu negatyviu duomenų apdorojimo reiškiniumi, apdorojant duomenų surinkimas:

- duomenų bazėje žemiau lengviau išvengti duomenų dubliavimo, nes duomenys vienaip ar kitaip identifikuojami, todėl tuos

pacius identifikatorius ir reikšmes turintys duomenys į bazę nebeirasomi;

- duomenų apdorojimo uždaviniams programoms nereikia rūpintis duomenų kaupimo problemomis, bet reikia pagal konkrečios

duomenų bazės reikalavimus paruošti kaupti į duomenų bazę.

Duomenų bazių sistema turi negatyviu dalyku: dalis duomenų retai naudojami, todėl visa duomenų bazė reikia nuolat

palaikyti aktualiame busenoje. Antra, koreguojama arba aktualinama duomenų bazė ir ta dalis, kuri retai naudojama,

neįmanoma prognozuoti ar ji iš viso bus kada nors ateityje naudojama.

Duomenų bazių valdymo sistemų sukurta labai daug.

Duomenų irasų gali būti duomenų lentelės, lentelės kortelės, atskira pareikštis, arba joms adekvacijos struktūros.

Svarbiausia, kad irasas būtų vienareikšmiškai identifiкуotas su išreikštais paieškos požymiais, t.y. iraso duomenų savybių

kodais. Tik tokiu atveju galima surasti reikiamus irasus, turinčius vienokias ar kitokias taikomajam uždaviniui reikalingas

savybes.

Kiekvienas irasas, patekęs į duomenų bazę, turi savo vietos joje adresą. Tad toliau tas irasas identifiкуojamas jo adreso

skaitine ar simboliu reikšme.

Duomenų bazių org. principas - duomenų kaupimo kompiuteryje atskyrimas nuo duomenų apdorojimo užd. bei jų programų.

Prie duomenų bazės reorganizavimo uždavinių priskirtini:

- duomenų koregavimo užd.;

- duom. dalinio dubliavimo išvengimo užd.;
- duom. bazės peradresavimo ir racionalizavimo uždaviniai.

Tie taikomieji uždaviniai turi paprastą kreipties struktūrą į duomenų bazę. Pagal konkrečios duomenų bazės valdymo sistemą

reikalavimus, nurodomas duomenų pozicijos ir to pakanka kreiptis realizuoti. Žymiai sudėtingiau kreiptis į duomenis, kada kai

kuriu pozicijomis ne visi įrašai reikalingi taikomajam uždaviniui. Šiuo atveju išskiriami visi duotojo pozicijos įrašai, o

reikiamus įrašus turi pasirinkti pačios taikomąji uždavinį realizuojančios programos.

Įrašų koregavimo procedūras atlieka duomenų bazės valdymo sistema. Aptikus klaidingą įrašą atliekama procedūra, analogiška

duomenų įvesties į bazę procedūrai.

Kad išvengtų įrašų dubliavimo, įrašų adresu einančius to paties pozicijos adresu grandinėje turiniai

sulyginami su naujojo įrašo turiniu. Visiško duomenų dubliavimo išvengti praktiškai labai sunku. Visiškai išvengiama dubliavimo, jeigu įrašų

laikome pareikštis, arba vieno identifikuoto atributo reikšmę.

Praėjus tam tikram duomenų bazės eksploatacijos laikui, neracionaliai didėja tų vietų apimtys.

20. Duomenų ir programų sąsajos duomenų bazėse.

Ryšius tarp duomenų ir programų galima suskirstyti į formalius ir priešastinius arba semantinius ryšius.

Formaliems

ryšiams tarp duomenų ir programų priskiriami tokie ryšiai, kurie tiesiogiai nepriklauso nuo programų atskirų komandų bei jų

rinkinių, o taip pat nuo duomenų struktūros bei jų turinio.

Priežastiniai ryšiai tai tokie ryšiai, kurių realizacijos tikslas yra duomenų apdorojimas ir kitu - rezultatinis duomenų

gavimas, tai taikomojo pobūdžio uždavinių sprendimas.

Sprendžiant daugelį uždavinių būtinas sprendimų duomenų išiminių tarpusavio derinimas, kad duomenys vienai kita

neuztrintų ar apskritai uždaviniai netrukdytų vienas kito sprendimui. Taikomajai programai paliekama tik tikslinės uždavinio

funkcijos realizacija. Galutinio vartotojo programine iranga gali būti suskirstyta į tipinę irangą ir specialiąją programine

iranga.

21. Kompiuterio komandos.

Kompiuterio komandos struktūra tiesiogiai priklauso nuo oper. atminties struktūros. Kompiuter. atmintis sudaryta iš baitų,

kurių yra mažiausios adresuojamos atminties ląstelės. Todėl komanda gali sudaryti skirtingas, priklausomai nuo jos

paskirties, greta esančių baitų skaičius, o jos algoritma apsprendžia tu baitų turinys, t.y. skiltyse esančių nulių ir

vienetų kombinacijos. Komandos vieta atmintyje nustato operacijos kodas. Nuo jo priklauso operandų skaičius, jų turinys,

komandos užimamų baitų skaičius, ir kt.

Operandais vadinamos komandos dalys, nurodančios komandos operacijos veiksmo objektus. Registras laikomi oper. atminties

dalys, dažniausiai esančios to kompiuterio masinio žodžio ilgio.

Masininiu kompo komandu gali buti ivairiu. Jas galima suskirstyti i tokias grupes: aritmetines, logines, postumio, palyginimo, perdavimo, ikrovio ir kt. Aritmetiniu komandu pagrinda sudaro sudities ir atimties komandos. Siu komandu kombinacijos leidzia atlikti daugybos, dalybos ir kitokius veiksmus su skaiciais. Logines komandos leidzia isskirti is duomenu teksta, skaiciu arba reikiama ju dali. Postumio operacijos igalina sujungti arba isskirti duomenis i bendras arba atskiras ju dalis. Valdymo perdavimo komandos suprantomos kaip nuoseklaus komandu vykdymo, nutraukimo bei perdavimo kitai komandai komandos. Ikrovio komandos atlieka duomenu perkeliama is vienos oper. atminties i kita.

22. Komandu operandai ir formatai.

Zemiau yra pateikiami operandu elementai bei ju raidiniai zymejimai:

B - bazinis registras, kuris nurodomas visose komandose, kurios susietos su tam tikru oper. atminties adresu.

D - dydis, nurodantis slinki. Slinkis sudetas su bazinio registro turiniu duoda tikraji (masinini) komandos adresa oper.

atmintyje.

I - simbolis arba skaicius, kuris komandoje nurodomas toesiogiai.

L - duomenu ilgi nurodantis dydis.

M - kauke.

R - bendrasis registras.

X - indeksinis registras, kurio turiniu naudojasi indeksines komandos.

Komandu operandu israiskose pateikti skaiciai 1,2 ir 3 reiskia operando elemento arba paties operando numeri komandoje.

Jeigu operandas neturi elementu, tai tada tokio skaiciumi zymimas jis pats.

Visos komandos skirstomos i 5 tipus:

RR - sio tipo komandose atliekama operacija registruose. RR komandu vykdymo pvz gali buti duomenu persiuntimo is registro i registra operacija.

RS - atliekama operacija oper. atminty ir registruose. Siame formate atkreiptinas demesys i 2 registru i atmnities lasteles

panaudojimo. Taip pat sis formatas igalina operacijos rezultata atsiminti treciajame operande nurodytu adresu.

RX - atlieka operacija oper. atminty ir registruose. Siame komandu formate galimas atminties adreso formavimas. Veliau,

visuose kituose formatuose atminties adresas bus formuojamas paprasciau, nes kituose formatuose nebus indeksinio operando X.

Indeksines komandos, turincios RX formata, yra ypac daznai naudojamos.

SI - atlieka operacija oper. atminty, panaudojant tos pacios komandos simboli ar skaiciu. I- komandoje uzduota konstanta,

dazniausia uzima viena atminties baita pacioje komandoje.

SS - operacija atliekama dviejose oper. atminties laukuose. Sioje komandoje yra pazymetini tiesiogiai uzrasyti komandoje

duomenu atminty nurodytuju masininiu adresu duomenu lauku ilgiai. Ilgiu nuorodos supaprastina manipuliavima duomenimis.

23. Formalus duomenu ir programu rysiai.

Formalus duomenu ir programu rysiai pirmiausiai nagrinetini tokiais aspektais:

- duomenu bei programu isdestymo ivairove vienas kito atzvilgiu kompiuterio atmintyje;

- adresiniai duomenų ir programų ryšiai;
- duomenų kodavimai ir perkodavimai;
- fiziniai diskretūs kompiuterio elementai bei dvejetainiai kodai.

Kai komp. atmintyje sprendžiamas vienas uždavinys ir ta sprendima realizuoja viena programa, tai programos fragmentai ir

apdorojamų duomenų laukai gali būti išdestyti vienas kito atžvilgiu labai įvairiai. Sudėtingiau yra išdestyti tarpinius uz-

davinio sprendimo duomenis bei rezultatinus duomenis. Po programų ir pradinių duomenų skiriama pakankamo dydžio atminties vieta tarpiniams rezultatiniams duomenims. Nežiūrint į didelę siuolaikine op. atm., jos trukumas visada išlieka, nes visu pirma tobulėja ir didėja bendrujų ir sisteminių programų kiekis, antra, daugėja uzdavinių, kurie užima ypač didelius atm. laukus. Pritrūkus op. atm., programa lengvai išsiverčia dalį savo pacios fragmentų ir duomenų laukų laikydama išorinėje atmintyje.

Situacija op. atm. darosi sudėtingesnė, jeigu sprendžiamas ne vienas, o keli užd. vienas po kito arba dalimis, t.y. lygiagrečiai. Visais atvejais ryšiai tarp programų ir duomenų yra formalūs ir tiesiogiai nesietini su duomenų semantika.

Adresiniai duomenų ir programų formalūs ryšiai sudėtingi, nes fiziskai, kiekvienam duomenų apdorojimui tomis

paciomis programomis ir programos ir duomenys privalo gauti būti vis kitoje op. atm. vietoje. Adresinis ryšys tarp programų ir duomenų yra kiekvienam duomenų apdorojimui programų ir jų komandų buvimo atmintyje vietos

naujų adresų nustatymas. Šis duomenų ir programų adresų nustatymas dažniausiai vadinamas saitu redagavimu.

Prie formalių duomenų ir programų ryšių priskirtini ir duomenų kodavimai ir perkodavimai į tokias duomenų kodavimo

sistemas, kad programos gautu atpažinti įvairius simbolius, bei skaitmenis, kuriais išreiškiami duomenys.

Vienu ir

tuos pačius duomenų simbolių ir skaitmenų dvejetainiai kodai gali būti įvairūs ir skirtingi. Svarbu, kad diskreciškai komp.

elementų vidiniai kodai sutaptų su duomenų kodais. Viename ir tame pačiame komp. skirtingi įrenginiai gali turėti nevienodus

vidinius duomenų kodus. Todėl akivaizdu, kad duomenų kodai tik formaliai išreiškia duomenis ir tik formaliai pagal kodus

programa atpažįsta duomenis.

Formalus duomenų perkodavimai gali būti labai įvairūs. Bet jų mechanizmai nėra sudėtingi. Kompiuterijoje yra labai dažnai

naudojamos dvi kodavimo sistemos: ASCII (American Standard Code for Information Interchange) ir EBCDIC (Expanded Binary

Code Decimal Interchange Code). Perkoduojant iš ASCII į EBCDIC, ASCII sesioliktainiai kodai traktuojami kaip perkodavimo

lenteles baitų eilės numeriai, o EBCDIC kodai įrašomi į lenteles vieta, atitinkanti tą patį simbolių ar skaičių, esanti

nurodytoje lenteles vietoje.

24. Priezastiniai duomenų ir programų ryšiai.

Priezastiniai ryšiai keičia vienus kitais duomenis tarp programų.

Priezastiniai - jų realizacija leidžia apdoroti duomenis arba spresti uzdavinius.

Priezastinis ryšys prasideda tada, kai perkoduotieji duomenys išdestomi tam tikra griežta tvarka (eile) ir griežtai tam

tikri, t.y. vienoki, o ne kitokie.

Nežiurint į tai, kokios sudėtingos programines sistemos aplinkoje ta programa veiktu, pateikiamos priezastiniu ryšiu grupės išlieka. Abipusiu bei griztamuju priezastiniu ryšiu savybės sisteminei programinei irangai yra būdingesnės, nei taikomojo pobūdžio uždaviniams.

Vienpusiu priezastiniu ryšiu programa - duomenys esmė yra ta, kad paprastai dauguma programiniu proceduru nereikalauja įvertinti daugumos apdorotų duomenų rezultatu. Sudėtingesnėse duomenų apdorojimo programose dažnai yra ir abipusis programos ir duomenų ryšys. Kintantys duomenų apdorojimo rezultatai yra papildomai analizuojami, arba dar prieš apdorojimą analizuojami pradiniai duomenys ir priklausomai nuo tos analizės rezultatu parenkama vienokia ar kitokia programos tolesnio vykdymo atsaka.

Griztamas priezastinis ryšys tarp duomenų ir programos nustatomas tada, kai priklausomai nuo duomenų tenka keisti pacia programą. Pavyzdžiui, esamas duomenų kiekis netelpa į programoje numatytus ir rezervuotus duomenų laukus. Tada pati programa priversta rezervuotųjų laukų apimti keisti.

Priezastiniai ryšiai realizuojami naudojantis formaliais ryšiais, nes vienoks panaudojimas gali išpresti duomenų apdorojimo uždavini, o kitoks negali. Priezastiniu ir formaliu ryšiu tarp duomenų ir programų visuma, naudojama vienu metu, atskleidžia programavimo, kaip tokio esmė, nes paaiskina, kaip duomenys apie realų pasaulį paversti dvejetainėmis išraiškomis, bei duomenys apie duomenis taip pat paversti dvejetainėmis išraiškomis (programomis) tam tikrais grįžtamais nustatytais formaliais ryšiais veikdamos keičia duomenis į apdorojimo rezultatus ir vėl pateikia iš dvejetainės formos galutiniam vartotojui suprantama išraiška.

25. Dvejetainis programavimas.

Programavimas - tai istisas kompleksas programų kurimo, derinimo, realizavimo ir netgi užduociu analizės etapu.

Yra trys virtualios dvejetainių komandų sistemos.

Pirmas principas taikomas tuo atveju kai, skirtingu komandų ilgiai (baitais) yra nevienodi, nes tokios komandos yra analogiskos assemblerinėms komandoms.

Antras principas tai komandos turincios operacijos koda bei tris atminties adresus.

Trecias, pagrindinis principas, tai istisa dvejetainiu komandų sistema, turinti fiksuota komandų ilgi ir geriausiai atspindi

zemiausio dvejetainio programavimo lygmeniu sąsajas su fiziniiais komp. elementais. Minetas dvejetainis programavimo lygmuo atskleidžia gilumine programavimo sampratą, nes visi suolaikiniai komp. bei jų istobulinta prog. iranga iš tikrujų dirba tik

dvejetainiame lygmenyje.

26. Skaiciai ir simboliai.

Tegu virtualus procesorius "supranta" oper. atm. struktura, kurios vienetas sudarytas iš sesiu baitu. Virtualia kompiuterio

atm. struktura lemia tai, kaip atm. vienetė, siuo atveju 48 bitu lasteleje (atm. lauke), pateikiami duomenys ir programu ko-
mandos. Bet koks tekstas, sudarytas is raidziu, specialiu zenklu, bei skaitmenu, i atminties lasteles uzrasomas taip: i viena
baita uzrasomas vienas simbolis. Simboliais laikomi: raides, desimtainiai ir sesioliktainiai skaiciai ir specialus
zenklai.

27. Skaiciu pateikimo formos.

Skaiciai komp. atm. pateikiami labai ivairiai. Bet paprastai skaiciu komp. procesorius apdoroja, kai jis yra
tam tikroje

vienoje dvejetaineje formoje. Skaiciai gali tureti komp. atm. tokias pateikimo formas:

- a) desimtainiai skaiciai pateikiami tekstu;
- b) desimtainiai supakuoti skaiciai;
- c) desimtainiai skaiciai su slankiuoju tasku;
- d) sesioliktainiai skaiciai, kai vienas skaitmuo uzima viena baita;
- e) sesioliktainiai supakuoti skaiciai;
- f) sesioliktainiai ir dvejetainiai skaiciai su slankiuoju tasku.

a - desimtainiai skaiciai irasomi i atm. baitus vienas po kito visiskai neatsizvelgiant i atm. lasteles.
Akivaizdu, kad viena nuo kito gretimuose baituose surasytus skacius skiria ju zenklai.

b - dvigubai kompaktiskese yra supakuotu skainiu duomenu uzrasymo forma. Ji atmeta pirmaja kiekvieno
baito tetrada, palikdama tik antraja tetrada. Visiskai mechaniskai to padaryti negalima, nes kliudo tai, kad
skaciaus desimtainis taskas bei zenklai (+-) uzima visa baita. Tiesiog atmetus pirmaja tasko, pliuso ar minuso
kodo tetradas liks antroji tetrada. Siu kodu neimanoma butu atskirti nuo desimtainiu skaciu kodu.

c - kai atliekami komerciniai skaciavimai, ir tu skaciavimu pagrindas yra piniginis vienetas, tai del
desimtainio tasko vietos nekyla abejoniui nei pradiniuose duomenyse, nei apdorojimo rezultatuose.
Desimtainis taskas skiria du desimtainius zenklus nuo sveikosios sk. dalies. Atliekant mokslinius ar
inzinerinius skaciavimus, daznai desimtainio tasko vieta sunkiai nusakoma, nes ivairiu tarpiniu skaciavimu
rezultatai gali skirtis tukstanciais ir net daugiau kartu. Slankaus desimtainio tasko desimtainiu skaciu
vaizdavimo komp. atm. forma skatina keli svarbus veiksniai: pirma, galimybe didelius skacius uzrasyti
kompaktiskai, antra, atsisakyti ivairiu lygiu vienos ir tos pacios dimensijos, ja isreiskiant skirtingai desimties
laipsniu rodikliais, trecia, galimybe is anksto ivairiai isreiksti desimtainio tasko vieta. Uzrasant skacius su
slankiuoju tasku paprastai naudojamas exponentes pavidalas.

d - sesioliktainiai skaitmenys uzrasomi i antraja baito tetrada, o pirmoje tetradoje visur uzrasomas HEX 'F'
(BIN '1111'). Kai yra tarnybiniai simboliai (+- ir kt.) pirmoji baito tetrada neturi buti uzpildyta.

e - dvejetaines tetrados kombinacijos nuo HEX '0' iki HEX 'F' uzima vien tik sesioliktainiu skaitmenu
kombinacijas. Todel nelieka HEX ar BIN kombinaciju net tarnybiniams simboliams pazymeti. Negelbsti
dvieju ar net keliu is anksto zinomu simboliu irasymas i greta esancias tetradas. Iseitis, kai HEX skaciu
negalima laikyti atmintyje supakuotame pavidale randame gana paprastai. Tai pozicinis HEX skaciu
vaizdavimo atmintyje budas. Tegu zinoma HEX skaciu masyvo pradzia. Kiekvienam skaciuui jo pradžioje
irasomas baitas, nustatantis to skaciaus struktura. Jeigu septintame bite nurodytas nulis - skaciusteigiamas,
jeigu vienetas - skacius neigiamas.

Pastebetina, kad viena skaitmeni turintis skacius atm. uzima 1,5 baito arba tris tetradas. Skaiciai gali buti l.
dideli (HEX skacius turi 15 skaitmenu) ir dar septyni skaitmenys trupmeninej daly. Jeigu tokiu skaciu
didumo nepakanka, galima naudoti skacius su slankiuoju tasku.

f - sesioliktainis skacius, isreikstas trupmena, kuri dauginama is skaciaus sesiolikto laipsnio rodiklio, t.y.
HEX skacius su slankiuoju tasku. HEX skaciaus su slank. tasku analogiskas DEC skaciuui su slank. tasku.
Didziajai daugumai skaciavimu pakanka 8 zenklu HEX skaciaus rodancio reiksmines skaciaus skiltis. HEX
skaciaus su slank. tasku struktura adekvati BIN skaciaus sandara.

Dvejetainio skaciaus su slank. tasku bendriausia forma galima uzrasyti taip:

$$x = \pm m \cdot 2^p$$

Cia m yra skaiciaus mantise, p - skaiciaus eile.

Mantise yra skaiciaus dalis, parodo to skaiciaus reiksmines israiskas, naudojamas skaiciavimaams. Normalaus mantises pavidalo auksciausioji skiltis negali buti nulis.

Skaiciaus eile yra toks laipsnio rodiklis, kurio pakeltas skaiciavimo sistemos pagrindas, siuo atveju - 2, kad jis padaugintas is mantises teiktu tikraja skaiciaus skaitine verte.

Slankiojo tasko skaiciu vaizdavimo budas yra ypac naudingas ivairiausiu skaiciavimu programoms sudaryti. Programines priemones arba ivairiu skaiciavimu aparatine realizacija slankiojo tasko bet koki atveji priima vienodai. Tuo tarpu fiksuoto tasko kiekviena galima vieta bei prie kiekvienos is vietu skirtingas sveikosios skaiciaus dalies ilgis suprantamas vis kitaip ir reikalauja atskiros programines analizes ir tam tikru koregavimo veiksnu. Del to apribojimo veiksnu programavimas yra sudetingeja.

28. Dvejetainiu komandu sandara.

Komp. procesorius programa pradeda vykdyti analizuodamas ieigos i programa adrese esancio baito turini. Pirmas - maziausias atm. lauko, kuriame yra programa adresas nebūtinai yra programos ieigos adresas. Analizuojant ieigos adreso turini, pirmiausia nustatomas komandos kodas, o pagal ji nustatomas tos komandos ilgis. T.y. kur prasideda sekanti komanda pagal ju isdestyma atm. lauke. Jau pati programos ieigos adrese esanti komanda programos vykdyma gali nukreipti kitur, t.y. i kita programos adresa. Bet kuri komanda laikoma turinti vienoda ilgi, t.y. kiek komanda uzima saligine atm. lastele, kuri sudaryta is sesiu is eiles einanciu baitu. Komandos lasteles nulinis baitas skirtas operacijos kodui (OK) uzrasyti. OK reiksme paprastai laikoma operacijos nebuvimu sioje lasteleje. Pirmasis numeris lasteles baite yra komandos indeksinis baitas (I). Pirmieji 3 sio baito bitai skirti I pazymeti. Likusieji penki bitai yra komp. atm. bloko indeksai arba ju numeriai. Antras, trecias, ketvirtas ir penktas komandos lasteles baitai yra skirti atm. adresams nurodyti tame atm. bloke, kurio numeris uzrasytas i baito 3-7 bitus. I lasteles (registrai) atmintyje sudaro atskira atm. bloka. Registro numeris 000 reiskia, kad registro panaudojimas konkreccioje komandoje nereikalingas. Uz registro baitu seka pagrindines atm. blokai. Kiekvieno atm. bloko viduje baitai numeruojami nuo HEX '0000' iki HEX 'FFFF', nes viename bloke yra DEC '65536' baitu. Is viso bloku yra 32. Tai yra pagrindine atmintis.

Registrai:

nulinis - komandose nenaudojamas;

pirmas - komandoje naudojamo adreso modifikacijos registras;

antras - komandoje naudojamo kito adreso modif. registras;

trecias - pries vykdoma komanda vykdytas komandos operacijos rezultato registras;

ketvirtas - komandu adresu skaitiklis;

penktas - pagrindines komandos realizav. registras;

sestas ir septintas - bendrieji registrai.

Nulinis ir nuo ketvirto iki septino registrai naudojami kaip bendro pobudzio registrai, galintys padeti vykdyti ivairiu tipu komandas.

Dvejetainiu komandu bendrosios sandaros nustatymas leidzia pereiti prie dvejetainiu komandu sistemos pateikimo.

29. Dvejetainiu komandu sistema.

Dvejetainiu komandu sistema sudaro bendri siu komandu algoritmu atskiru komandu grupiu bei konkreciu komandu algoritmai. Bendri komandu Algoritmai - veiksmai, kurie leidzia nustatyti duomenu vykdomuosius adresus, nustatyti komandu adresus bei atlikti ju keitimus operat. atmintyje. Komandas salyginai galima suskirstyti i aritmetiniu operaciju, kodo keitimo operaciju, valdymo operaciju ir isoriniu irenginiu valdymo komandu grupes.

30. Bendrieji dvejetainiu komandu algoritmai.

Bendrieji BIN komandu algoritmai susieja su simboliu ir skaiciu israiskomis, komando sandara, atm. struktūromis ir kt. Bet kurios komandos formali israiska yra:

b/OK, I, a1, a2

kur b - komandos adresas operat. atm.;

OK - operacijos kodas;

I - indeksas;

a1, a2 - duomenis nustatantys komandos komponentai.

Indeksas: $I=R/n$, kur R - registro kodas (nuo 0000 iki 0111), n - atm. bloko kodas (nuo 00000 iki 11111).

31. Formalios operatyviosios atminties adresu israiskos.

Oper. atm. sudaro registrai ir oper. atm. baitai. Formaliai visi oper. atm. adresai uzrasomi dviem tvarkiosiomis aibemis: registru tvarka aibeje ir pagrindines atm. adresu tvarka aibeje. Adresu isreikstu skaiciais is viso nera, yra tik registrai ir atm. blokai. Tusti registrai bei operat. atm. uzpildomi specialiais uzpildymo simboliais. Svarbu zinoti adresa, kuriame yra reikiama komanda arba reikiami duomenys, bet ne maziau svarbu ir adreso turijys, bei jo formali israiska.

32. Formalios adresu turinio israiskos.

Adresu turiniu gali buti ne tik duomenys apie realaus pasaulio objektus, bet ir duomenys apie duomenis, t.y. programu komandos, ju adresai, arba metaduomenys. Formalioje adreso turinio israiskoje, kur reikalingas adreso turinys, adreso zenklas uzrasomas paprastuose skliausteliuose.

33. Adresu modifikacijos komandose.

Musu minetos komandos adresas b gali buti ne tik atm. adresu, o ir registro adresu, isreiskiamu jo numeriu, t.y.

komanda, uzrasyta i registra, jame ir vykdoma. Koks registro numeris naudojamas komandoje, priklauso vienom komandom tik nuo OK (operacijos kodo), o kitoms registro numeri parenka programuotojas pagal jo vaidmeni bendrajam komandos vykdymo algoritme.

34. Adresu turinio kaitos algoritmas.

Sis algoritmas yra pagrindinis is bendruju algoritmu ir pagrindinai liecia registru turinio kaita. Algoritma sudaro keturi pagrindiniai jovykdymo zingsniai:

pirma - pirmosios komandos adreso nustatymas ir jo irasymas i komandu adresu skaitikli.

antra - komandos, kurios adresas komandu adresu skaitiklyje ikelimas i komandu vykdymo registra.

trecia - komandos vykdymas.

ketvirta - ivykdytos komandos rezultato irasymas i registra.

35-36. Aritmetiniu operaciju komandos. Pagrindines aritmetines komandos.

Pagrindines aritmetines komandos:

sudetis - 1M;

atimtis - 2M;

daugyba - 3M;

dalyba - 4M, kur 1, 2, 3, 4 - aritmetiniu komandu grupes pogrupiai, M - komandu modifikacijos, kurios gali tureti tokias reiksmes:

M=0 - plus, minus, daugyba, dalyba;

M=1 - sudeties, atimties, daugybos, dalybos modifikacija su dvigubu duomenu ilgiu baitais;

M=2 - "-" modifikacija, kai duomenys operacijai pateikti su slank. tasku.

37. Pagalbines aritmetines komandos.

Jos yra:

skaiciaus pervedimas i forma su slank. tasku - '50';

skaiciaus pervedimas i forma su fiksuotu tasku - '51';

norminimo komanda - '52';

sandaugos maziausiuju skilciu nustatymo komanda - '53';

dalybos liekanos nustatymo komanda - '54';
skaiciaus apvalinimo komanda - '55';
skaiciaus apvalinimo blokavimo komanda - '56';
BIN vienetu kiekio nustatymo komanda - '57'

Bendras aritmetiniu (pagalbiniu!) kodas - 'S', kur S yra nuo 0 iki 7.

38. Kodu keitimo operaciju komandos.

Kodu keitimo grupes komandos yra komandos, kurios tiesiogiai apdoroja duomenis, juos keisdamos ir gaudamos naujus duomenis. Kodu keitimo komandu rezultatai - kokybiniai pakitimai. Visas kodu keitimo komandas skirstome i tokiu operaciju atlikimo komandas:

- logines, kurioms dar priskirtinos komandos:

- o palyginimo - '60';
- o logines daugybos - '61';
- o logines sudeties - '62';
- o ciklines sudeties - '63'.

- kodu perdavimo (persiuntimo), kurioms dar priskirtinos komandos:

- o persiuntimo - '64';
- o persiuntimo su skaiciaus zenklo pakeitimu - '65';
- o skaiciaus modulio persiuntimo - '66';
- o adreso persiuntimo is registro - '67'.

- Priskyrimo, kurioms dar priskirtinos komandos:

- o kodo peremimo is klaviaturos - '68';
- o skaiciaus zenlo priskyrimo - '69'.

- Postumio, kurioms dar priskirtinos komandos:

- o loginis postumis - '6A';
- o aritmetinis postumis - '6B'.

39. Valdymo operaciju komandos.

Daznai programose tenka atlikti suolius, t.y. persukti dali paeiliui einanciu komandu, pasukti komandu vykdyma tam tikra programos saka, pereiti paprograminio komandu atlikima, sugrizontant i ten, kur iseita, i tolesni komandu vykdyma paeiliui ir t.t.

- pagal valdymo komandas gali buti atliktos operacijos:

- o programos stabdos - '70';
- o pertraukties drausties/leidimo - '71';
- o nesaligines pereigos - '7M';
- o pereigos i paprogrami - '78'
- o ciklo - '79';

Cia M gali buti lygi 3, 4, 5, 6, 7, t.y. komandos: 73, 74, 75, 76, 77.

Operacija 7M vykdoma, kai anksčiau ivykdyta operacija tenkina salyga:

- teigiamas arba neigiamas rezultatas - '73';
- rezultatas nulis - '74';
- lasteles perpildymas - '75';
- rezultatu sutapimas - '76';
- rezultatu nesutapimas - '77';.

pabaiga.

1. Skaičiavimo sistemos.

Skaičiavimo sistema yra visuma būdų ir priemonių, leidžianti užrašyti ar kitaip pateikti skaičius.

Naudojamos yra kelios skaičiavimų sistemos: dvejetainė, astutainė, dešimtainė, sešioliktainė. Dvejetainė – turi du simbolius, yra elektroniniu sch. pagremdas, nes tik tokia schem. leidžia skaitmeninis reikšminės nustatyti (p. schem. būv). Pvz.: kruvio buvimas gali reikšti 1, o nebuvimas 0 (ir atvirkščiai).

Dešimtainė – yra pagrindinė, visuotiniai pirmta sk. sistema (turi 10 skirtingų simbolių).

Sešioliktainė – turi 16 simbolių, dažniausiai naudojama skaičių pervedimui tur dvejetainės ir dešimtainės sk. sistemu. Sešiolika simboliu iki devinto yra parnsomi skaičiumi, o toliau raidė (A, B, C, D, E, F).

Astutainė – yra analogiška sešioliktainei, gali būti naudojama pervedimui dvejetainio skaičiaus ir dešimtainės sistema, tačiau yra 8 simboliai, o pervedėjanti je BINI i DEC yra naudojamos triados, bet ne tetrados.

Bet kurioje skaičiavimo sist. gali būti atliktami aritmetiniai veiksmai (sudetis, atimtis, dalyba, daugyba ir t.t.), ir bet kurioje skaičiavimo sist. tas pats skaičius gali būti užrašomas skirtingai, tačiau skaičiaus dydis lieka tas pats.

2. Skaičių pervedimas iš vienos skaičiavimo sist. į kita.

Norint bet kurioje skaičiavimo sistemoje, bet koki skaičių pervesti į kita skai. sist., naudinga pirmiausia jį pasivertinti į dvejetainį, o tik paskui į norimą skai. sistemos skaičių.

Verta žinoti, kai iš dešimtainės skaičiavimo sistemos pasivertinti dvejetainiu. Reikia atlikti paprastą dalybos veiksmą su dviųju. Pvz:

```
DEC '12' = BIN '1100' -> 12=6 ir liekana 0      12 | 0 -> BIN '1100'
6=2+3 ir liekana 0          6 | 0
3=2+1 ir liekana 1         3 | 1
1                            1 | 1
```

Dvejetainis skaičius užrašomas nuo galo.

Skaičiai ir astutainės ir sešioliktainės skai. sist. yra pervedami atitinkamai triadomis ir tetradomis. Pvz:

```
HEX '1F3' -> 1 F 3
BIN '0001111100011' -> 0001 1111 0001
```

OCT '123' -> 1 2 3

BIN '0010100011' -> 001 010 001 011

Pervadant iš vienos sk. sistemos į kita skaičius, kuriuos sudaro sveikųjų ir trupmeninės dalys, ir sveikųjų ir trupmeninės dalyse pervedamos atskirai. Visada galima būdas, jog is bet kokios sistemos reikia pasivertinti ir BIN, o paskui į norimą skai. sist.

Tegiu duota trupmena DEC '0.257'
0257 x 16 = 4.112 = 4 + .112
112 x 16 = 1.792 = 1 + .792
792 x 16 = 12.672 = C + .672
672 x 16 = 10.752 = A + .752
752 x 16 = 12.032 = C + .032

DEC '257' = HEX '41CAC.' = BIN '100 0001 1100 1010 1100'

Džiauniai tampa apvalinti trupmenas, nes pervedimas gali testis neribotai.

3. Dvejetainė aritmetika.

Dvejetainės aritmetikos pagrindas – sudėtis, atimtis, daugyba.

Dvejetainiu sk. daugybos ir dalybos veiksmus yra analogiškas dešimtainiu sk. daugybos ir dalybos veiksmams.

Sudėstant skaičius vienetų pernesimas vyksta į didesniajį (į kaire) tada, kai sudedama 1+1=10.

Skaičius atimant, prireikus vyesniosios skaities vienetą pernesti į desimtiną skiltį (pasiskolinti), pastarojoje skiltyje atsiranda du vienetai (t. y. 10), o tarpinė nulnėsis skiltyje – po viena vienetą, nes kitas vienetis keliauja iki mažiausios skilties i kura skolinamasi. Todėl pastarojoje skiltyje prieš atimant, atsiranda du dvejetainiai vienetai.

4. Logines operacijos.

Pagrindines logines operacijos yra tiesa, kuri žumima skaičiumi 1, ir netiesa – skaičiumi 0.

Logines operacijos vadinsimos ARBA (logine sanda, disjunkcija), IR (logine daugyba, konjunkcija), IŠSKIRTINE ARBA (neekvivalentumas), NE (loginis neigynis, inversija).

Logines operacijos žymimos simboliais: ARBA – v, IR – ^, IŠSKIRTINE ARBA – (-), NE – ~.

Logines operacijos rezultatas gali būti tik tiesa 1, arba netiesa 0.

Logine operacijoje ARBA, būtų palyginimo rezultatas visais atvejais yra 1, išskyrus tu atvejį, kai yra palyginami nuliai, tada rezultatas bus 0. Logines operacijos IR, visais būtų palygino atvejais rezultatas yra 0, išskyrus tu atvejį, kai palyginami yra vienetai, tada rezultatas bus 1.

Logine operacijoje IŠSKIRTINE ARBA lyginami skirtingi baitai, turi rezultata 1, o vienodi 0.

Loginis neigimo NE rezultato analogiskose vietose vienetai yra pakeičiami nuliais, o nuliai vienetais.

Kompiuterio atmintyje esančių dvejetainiu strukturu tiesiogiai vizualiai matyti yra neįmanoma, tai yra daroma operacijoje pateikiant sabloną, t. y. anksčiau žinoma dvejetainė struktura. Paskui pagal logines operacijos rezultata, spresti, kas ten buvo.

5. Logiūnų operacijų pavyzdžiai:

```
A ARBA IR (-)IŠSKIRTINE ARBA (-)NE
00001111 10010010 01111011 01101011
10000000 00000000 11111111 -----
----- 10010100
00011111 11111001 10000010
```

6. Dvejetainės operacijos diskreicose kompiuteriu elementuose.

Kompiuteriu elementu nuroditi kita, tobulėja, o dvejetainė aritmetika lieka ta pati. Diskreicinis elementas vadinuojamas schema, kuri sudaro elektrone grandinę, susidedanti iš magnetinio (a,b) ir elektromagnetinio taisy (A,B), kontaktų ir elektrone stroves saltninio, bei sejinimo, operacijos įėjimo tuskio C.

7. Operacijos ARBA, IR schema. ARBA.

Dvejetainė operacijos schemai pateikiama elektrone stroves buvimu arba nebuvimu A ir B taksuose. Itampos buvimas taksuose A, B, C reiskia 1, o nebuvimas 0.

7. Operacijos ARBA, IR schema. ARBA.

IR

8. Informacija.

Informacija - tai žinių apie pasauli visuma. Ta visuma sudaro žinios apie įvairianuosius objektus, nuo elementariausių iki labai sudėtingų. Informacija gali būti skirstoma ir nusijauoma pagal įvairius požymius.

Pagal apdoravimo periodus, ji gali būti pradine, tarpine, rezultantine, kuri skirta galutinai jos naudojimui. Pagal sprendžiamų uždavinių tipus, inf. gali būti operatyvi, uždara, komercinė, mokslinė, ekonominė, filosofinė ir t. t. Pagal reikšmę - teisinga, klaidinga, pilna, nepilna, prisotangta, nereikalinga ir kt.

Vieno ir to paties pavadinimo inf. gali atitikti ne vienam, o keliems susijviam kriterijams.

Galutinai vartotojui laikomas asmuo arba organizacija, turintis savo inf., kuri reikia, būtina ar tikslinga apdoroti kompiuteriu. Tokia inf. negali būti pilnai apdorojama esama programne sisteme, taikomąja, tipine ar kitokia iranga.

Tokios inf. apdorojimui reikalinga sukurti specialia prog. iranga, kuri savarankiškai išspresintu mietos inf. apdorojimo uždavinius.

Informacijos pateikimoforma yra lentelės. Jomis pateikiama informacija nesunku transformuoti į matem. konstrukcija - relacinės aibės, relacinės

aibės iraisiaka turi didesni technines, technologines, ekonomines, mokslines ir kt. inf. dideles.

9. Informacijos apdorojimo lygiu samprata.

Informacijos apdorojimo lygiu laikytini tokie atvejai: 1) kompiuterinio informacijos apdorjimo, kuris inf. nekciaia, naujos informacijos negaunama, komp. naudojamas, kaip fizinis irankis zm. darbu tobulinti.

2) Vartotojui pateikiama kito vartotojo informacija, apdorojant ją i kitam vartotojui reikiama forma, eile, apdorojimo trukme ir pan.

3) Inf. apdorjimo rezultatas yra nauja inf., kuri išreiskinama skaičiais, zodiomis, simboliais, kodais, kurie yra žinomu ir naujai atributu reikšmes. Informacijos apdorjimo lygiu yra salyginiai. Kompiuteris yra tik irankis, kuris įvairiais aspektais palengvina zm. darbą. Kompiuterine sistema jokius naujos inf., siuo atveju, sukurti negali.

10-11. Duomenys, pareikštys.

Duomenys laikoma pareikstis, kuri pateikiama logiskai savarankisku, daugiau nedalomu skėniu. Toks sakėnys turi logini veiksmi ir pagrinda, bei paziminius ir papildinius.

Pavyzdys: "INSTITUCIJOJE b d METU PRADZIOJE DIRBO c DARBOTOJUI, KURIE SIOJE INSTITUCIJOJE TURI e METU DARBOTI AZA".

Cia loginis veiksnys yra zodiis DARBUTOJUI, o pagrindas - c. Pateiktis pareikstis yra logiskai savarankiska ir negali prarasti ar igyti nauju pozymiu ar papildiniu.

12-13. Duomeni identifikavimas, duomeni lentelės.

Jeigu turimas ne vienas duomenis, o keletas ar keliolika duomeni apie institucijas bei ju darbuotojus darbo staža, tai galima sudaryti tokiu duomeni hipotezinę lentelę. Sudarant tokią lentelę, pirmiausia yra išskiriami bendri visų duomeni pozymiai. Tokiu pozymiu galetu koki meta momentas (metu pradzia), Sis pozymis yra uzrasomas lentelės priskyje ir taramaji.

lentės identifikatoriaus sudetine dalimi. Kita lentės identifikatoriaus sudetine dalimi galetu būti subjektas, kurio duomenims skir- tios esoties.

Tai pat instituciju visumos tam tikra dalis: kurio tai verslo visos institucijos. Po to nustatomi ir koduojami tam tikra eile surasyti atributai.

Paigi identifikuoja lentės sudetine dalis rasyti identifikatoriaus atributu skėniu ir pacio lentės schema.

Identifikatoriaus schema: a - pacios lentės, kaip savarankisko duomenio darinio, kodas; B – jos pacios atributas – suvaidinti, kad buta pateikiu, kokia veiklos sriti اپاia institucijos, kuri duomenys pateikiama lentelėje.

D - laiko faktoriaus kodas, siuo atveju reiskiantis metu pradzia, o pidant lentelę nurodoma koki meta pradzes duomenys yra lentelės.

Duomeni lentelėje eilutes gali būti isdetytos ivairia tvarka: pagal kurio nors atributu reikšmiu didejima arba mažejima, pagal tekstinu atributu reikšmiu pirmaji radimo abecbe ir t. t. Nuo to duomenys nesiskicia. Butina turi galimybę viena eilute atskirti nuo kitos.

Tam reikiama vienas ir atributu nurodyti, jog koks atributa ir jo reikšmiu reikšmiu reikiama, t. y. identifikatoriaus. Jedu del kokios nors klaidos atstinkta taip, kad lentelėje yra dvi ar daugiau to pacio rakio ir vienedu kitu atributu reikšmiu eilutes, kurios eilutes išbraukiamas is lentelės paleikant tik viena is ju.

Tokia klaida surasti ir istatyti tik siu duomeniu savininkas ir yra butina kreiptis i pirmi duomeniu salini.

Duomenys, kuriu schema ir jos atributai turi realias atrib. reikšmes, gali būti identifikuojami.

Jeigu duomeni lentelės identifikavimui pakanka tik kodo a, tai reiskia, kad tos lentelės duomenys taip kad is pradiniu duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Duomeniu apdorjimo programoje turi būti naurodosi i kitas, bendras daugeliu duomeniu apdorjimo uždaviniu programas, kad atliktu pagalbinius darbus ir algoritmo tam tikrus realizavimo momentus, tokius kaip duomeniu vestis, isvestis.

Kad butu gautas mas duomenis ilgio, ju pradzioje arba pabaigoje papildomi iraisini specialius duomenis pagalinimo sim- boliai - protarpiai (). Pvz.:

Anita

123456

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Kintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

Mintamojo ilgio iraisini paprastai duomenis pagalinimo sim- boliais - protarpiai (). Pvz.:

