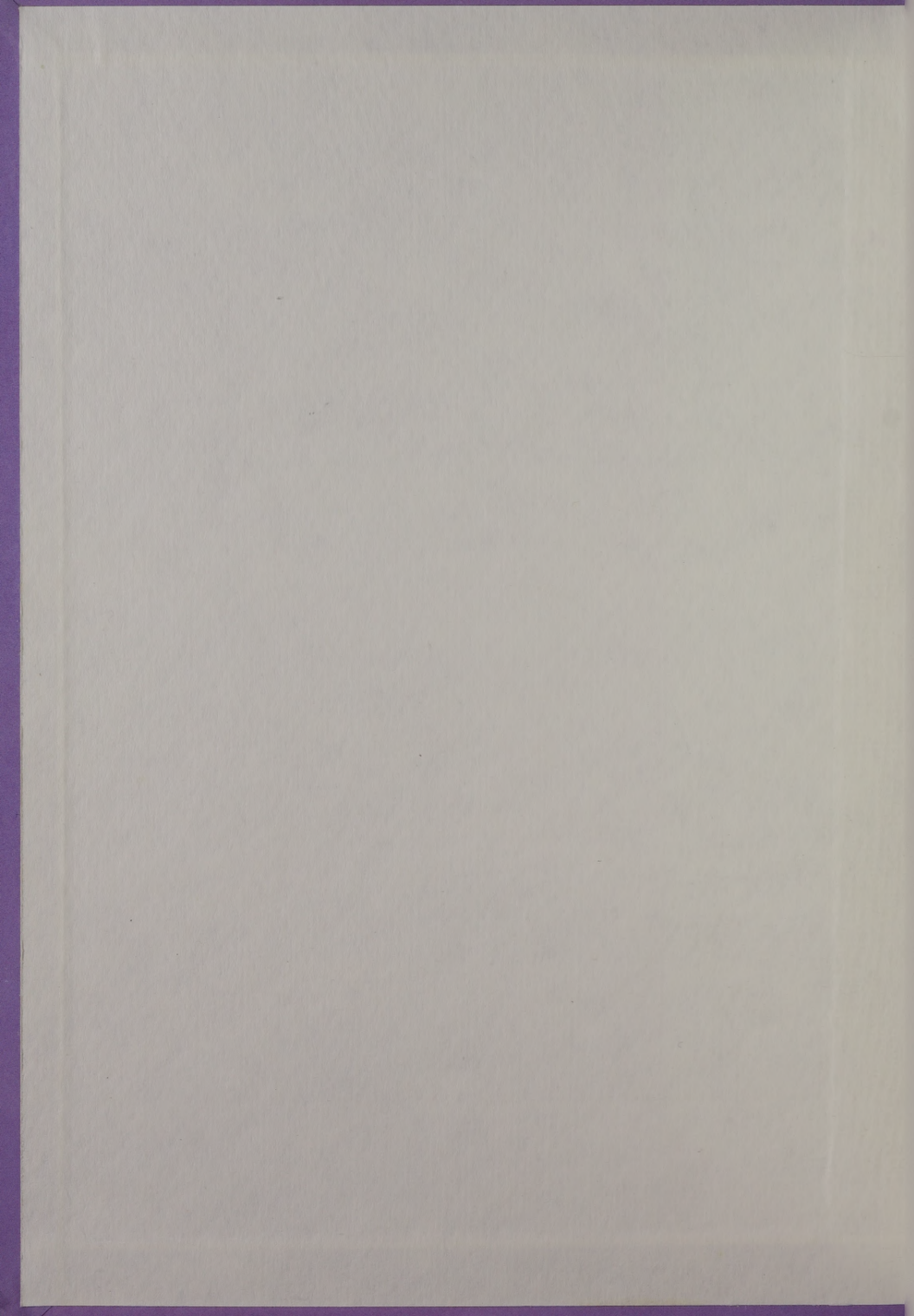


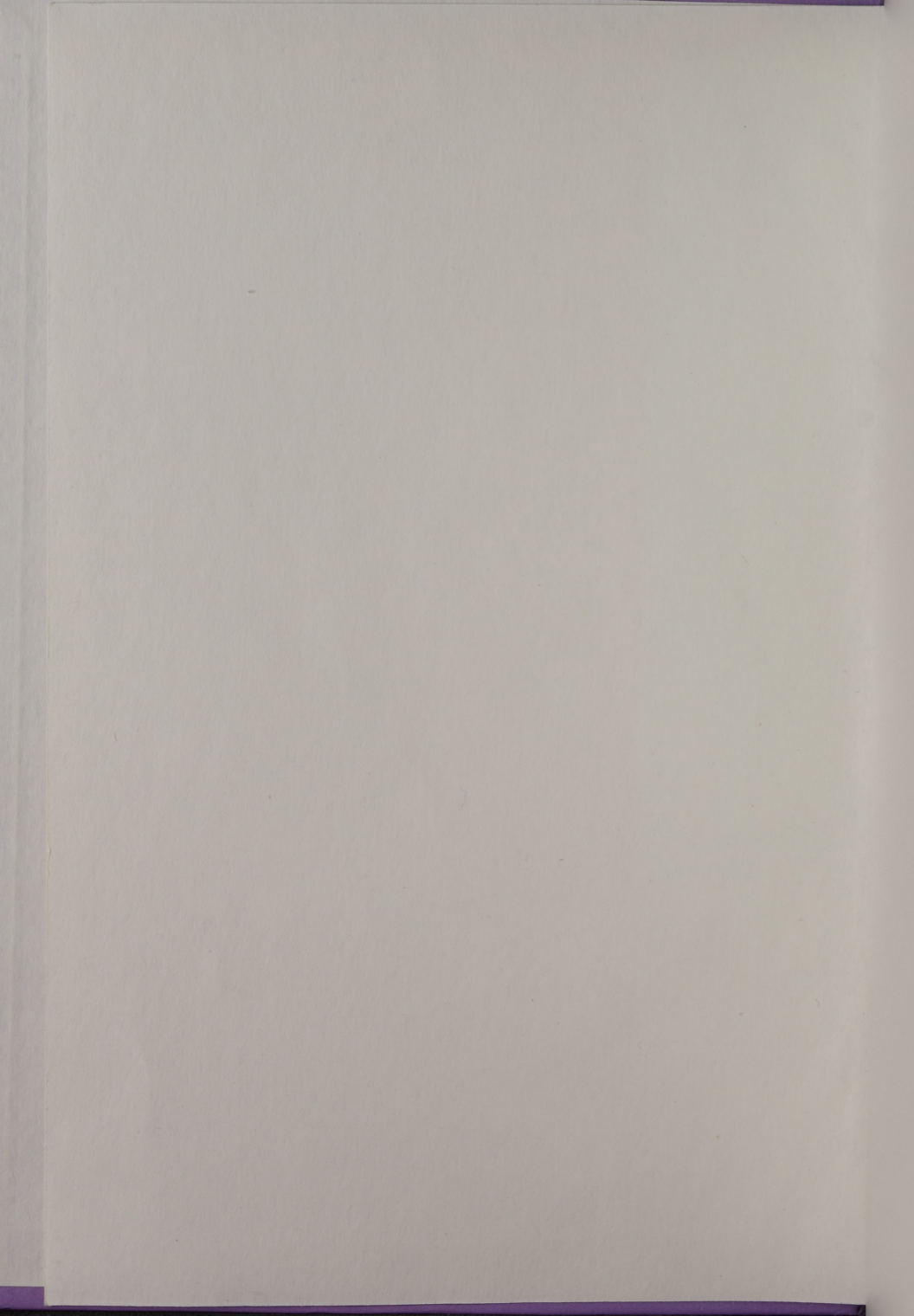
**MĀRIS DETLAVS
KATRĪNA SATAKI**

ComputerLand
izdevums

**IESKATIES
DATORU PASAULĒ**



RESEARCH
CITY OF
PASADENA



**IESKATIES
DATORU
PASAUĻĒ**

IESKATIES
DATORU PASAUĻĒ

Kārlis Vīķis, Ilmārs Jāns

Latvian Computer Center
1997

IESKATIES
DATORU
PASAULĖ

98-3
- 255

L
6

**MĀRIS DETLAVS
KATRĪNA SATAKI**

IESKATIES DATORU PASAULĒ

Romana Vitkovska ilustrācijas

Datorzinību Centrs
1997

Māris Detlavs, Katrīna Sataki. IESKATIES DATORU PASAULĒ.
Romana Vitkovska ilustrācijas. Rīga: Datorzinību Centrs, SIA. 1997. -
288 lpp.

Populārzinātniskā veidā lasītājs tiek ievadīts aizraujošā datoru pasaulē,
izsekojot skaitļotāju vēsturei, apgūstot jēdzienus un terminus,
iepazīstot pielietojumprogrammu iespējas. Sniedz lielisku informāciju
par mums zināmiem vai nepazīstamiem datoriem.

Recenzenti:

S.Bāliņa, Datorzinību Centra direktore, LU doktorande, Mg. math.;

R.Vitkovskis, Latvijas Izglītības fonda valdes loceklis, Ekspertīžu un
informācijas centra prezidents.

Saturs

Priekšvārds	7
Skaitļotāju priekšvēsture	11
Informācijas mērvienības datoros	31
Datoru sastāvdaļas	39
Daži ar datoriem saistīti jēdzieni	59
Algoritms	59
Konteksts	61
Vide	63
Vadība	63
Draiveris, Dzinis	66
Instalācija	66
Situāciju apstrāde	68
Pārtraukumi	69
“Uzkāršanās”	69
Vīrusi	70
Protokols	75
Interfeiss	77
Emulācija	78
Cash	81
Procesora un atmiņas darbības pamatprincipi	85
Pirmais tuvinājums	85

Otrais tuvinājums	87
Procesora parametri	89
Atmiņas parametri	97
Videoatmiņa	99
Operētājsistēmas	103
Operētājsistēmas uzdevumi	105
Datu uzglabāšana	113
Programmēšana	127
Datortīkli	143
Lokālais tīkls	146
Plaša mēroga tīkli	158
Dažas smalkākas lietas	161
Datu drošība un kriptogrāfija	161
Mākslīgais intelekts, ekspertsistēmas un ...	169
Teksta redaktori	185
Elektroniskās tabulas	199
Datu bāzes	207
Uzskaites sistēmas	227
Personālie organizatori	237
Grafiskās apstrādes redaktori un paketes	247
Internet un tā pielietojuma iespējas	263

Priekšvārds

Laipni lūgti datoru pasaulē! Datori veido veselu pasauli, kas ir komplicēta, idejām pārpilna, interesanta un pat skaista. Mēs gribam atvieglot jums šīs pasaules iepazīšanu ar analogijām no dažādām dzīves jomām, ilustrācijām un jokiem.

Šodien datori ir ienākuši daudzu jo daudzu cilvēku dzīvē. Ja jūsu dzīvē to vēl nav, nesteidzieties mierināt sevi ar domu, ka jums nekad nebūs vajadzības apgūt darbu ar datoru. Paskatieties acīs realitātei - cilvēce jūsu dēļ nespers soli atpakaļ. Jums agri vai vēlu būs jāsastop savā ceļā datori. Paraugoties darba piedāvājumu sludinājumos, līdz ar valodām tiek prasīta arī prasme rīkoties ar datoru. Nedomājiet, ka no jums tiek prasīts neiespējamais. Jums pašiem nav jāveido programmas, bet tikai jāprot tās izmantot.

Ne visi ir apguvuši datorus kā profesionāli datorspeciālisti, lielākai daļai cilvēku dziļas un pamatīgas zināšanas par datoru nemaz nav vajadzīgas, taču, peldot pa milzīgo datoru un programmu okeānu, mums ceļā var rasties tik daudz dažādu zemūdens klinšu, ka visas aprakstīt nemaz nav iespējams. Lai izvairītos vai pārvarētu šādu "klinti", ir vajadzīgas iemaņas. Dažs burā ar "šiku" - tiek galā ar problēmām pāris minūtēs, bet

citiem ir nepieciešamas vairākas stundas vai pat dienas (ja pietiek pacietības), lai tiktu galā ar kādu izveidojušos situāciju. Kur tad ir atšķirība? Tā ir zināšanās un praktiskajā pieredzē, kas ir uzkrātas, strādājot ar datoru. Mēs to saucam par datoru inteligenci. Ja katram cilvēkam jaunu zināšanu jomā būtu jānoiet tas pats ceļš, ko pirms viņa jau ir nogājuši daudzie celmlauži, tad nebūtu progresa.

Šī grāmata ir divu datorspeciālistu gadiem krātā pieredze, pasniegta vairāk vai mazāk vienkāršotā veidā. Mēs centāmiem dalīties ar jums savā pieredzē, atvieglojot jums datora iepazīšanas ceļu. Šī grāmata nav domāta konkrētu lietu apgūšanai. Mēs esam centušies vienkāršoti apskatīt visus, cik vien varējām iedomāties, ar datoriem saistītos jautājumus no sava subjektīvā viedokļa. Lai nevajadzīgi nesarežģītu izklāstu, vietām speciāli neesam to detalizējuši, bet dažviet šī paša iemesla dēļ esam bijuši pat neprecīzi. Grāmatas mērķis ir radīt iesācējam priekšstatu par datoru pasauli - pasniegt kaut ko līdzīgu krāsojamai grāmatai, kurā pēc tam katrs var precīzi izkrāsot to lappusi, kas viņam vajadzīga.

Pieredze rāda, ka nespeciālistiem pāiet ilgs laiks, līdz viņi spēj salikt dažādus datoru lietošanas aspektus kopā. Sastopot cilvēkus, kas lieliski veic savu darbu ar datoru, atklājas, ka viņi nezina pat elementāras lietas, kuru zināšana būtu stipri atvieglojusi, paātrinājusi un pat uzlabojusi viņu darbu.

Grāmatā mēs vēlējāmiem sniegt vispārīgu priekšstatu par datoriem un ar tiem saistītām lietām, lai pēc tās

izlasīšanas anekdoti par datoru vīrusa noglabāšanu uz disketes un disketes svinīgu sadedzināšanu visi uztvertu kā joku.

Vienlaikus mēs centāties atdzīvināt grāmatu ar jokiem, laužot daudzos cilvēkos iesakņojušos uzskatu, ka datorspeciālisti ir sevī noslēgti un drūmi ļautiņi, kas to vien dara, kā sēž pie datora un neuzticīgi skatās uz visu pasauli.

Šī grāmata noderēs visiem: skolēniem, kas apgūst datorus, cilvēkiem, kuri strādā ar datoru, kuriem liekas, ka viņu darbu varētu krietni atvieglot dators, priekšniekiem, kuriem ir aizdomas, ka padotie tos "tin" ap pirkstu, jebkuram, kam ir kaut mazākā interese par datoriem. Jā, un arī mūsu kolēģiem datorspeciālistiem, kuriem varētu būt interesanti salīdzināt mūsu sniegto skatu uz datorpasauli ar savējo. Parādiet šo grāmatu tiem, kuri uzdod tik daudz jautājumu, ka jums pietrūkst laika atbildēt.

Mēs esam centušies aptvert visas galvenās personālo datoru lietošanas sfēras. Tas ir milzīgs lietojumu apjoms, tādēļ esiet iecietīgi pret nenožīmīgām neprecizitātēm. Īpaši mēs gribētu atvainoties par terminoloģiju, mēs nespējam visur būt konsekventi, tāpat nespējam visur izpēīt valodnieku norādījumus, kas bieži mainās un kuru vēl biežāk vienkārši nav. Mēs izmantojām tos jēdzienus, kurus cilvēki reāli lieto, tiesa, dažreiz mēģinājām palīdzēt ieviest kādu latvisku terminu. Mums nav apzinātas valodas politikas, mēs nespējam uzņemties šādu atbildību. Pirmkārt, svarīgi ir celt vispārējo datoru

inteliģenci, un, cerams, ka arī terminoloģijas jautājumi tiks sakārtoti.

Grāmatai ir patiesībā ir divas daļas. Pirmās desmit nodaļas ir teorijas un jēdzienu skaidrojumi. Pēdējās septiņas nodaļas izklāsta svarīgāko pielietojumprogrammu iespējas.

Šī grāmata ir mūsu kopdarbs un tā arī jāuztver. Mēs paši bieži nezinām kurš kuru ideju ir sācis rakstīt. Tamdēļ abi uzskatām grāmatu par simtprocentīgi savu, vienlaikus neaizmirstot, ka ir otrs autors ar precīzi tādām pašām domām un tiesībām.

Grāmatas tapšanas procesa beigās parādījās Romans, kura bildes ir vēl viens skats uz datoriem. Interesanti, ka viņa attēlotās analogijas bieži vien nāk no pavisam citas sfēras, nekā mēs esam aprakstījuši tekstā. Tas dos lasītājam iespēju meklēt šīs analogijas un mazlietiņ pafantazēt.

Ceram, ka jums patiks mūsu idejas un humors. To ka mūsu zināšanas jums noderēs mēs jau zinām. Lasiet, skatiet, baudiet!

Skaitļotāju priekšvēsture

*Paši skaitļošanas
pirmsākumi ir
attiecināmi uz laika
periodu vēl pirms
mūsu ēras, kad senie
cilvēki skaitīšanai
izgudroja skaitāmos
kauliņus, iezīmes uz
akmens vai uz
sienas. Parasti šāda
veida skaitīšanu
veica burvi un
dažādu reliģisko
ceremoniju
izpildītāji, lai
noteiktu laika
intervālus starp
ceremonijām vai
ražas novākšanu
rudeņos un sējas
darbiem pavasaros.*



Skaitļotāju priekšvēsture

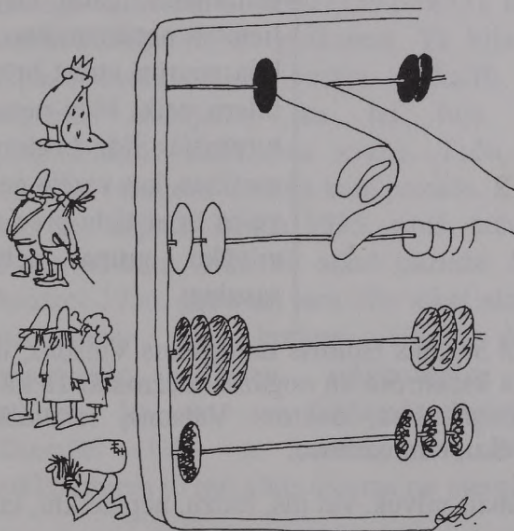
Akmeņpauris sēdēja uz nosūnojušā akmens, kas vienmēr bija aizšķērsojis ceļu uz viņa alu. Parasti kāds noteikti pret to paklupa, bet nevienam nekad nebija ienācis prātā pavelt to malā. Arī Akmeņpaurim nē, viņš bija, kā mēs šodien teiktu, senais cilvēks ar visai aprobežotu intelektu. Todien viņš ar nemiera pilnu sirdi gaidīja, kad kaimiņiene Vāle nāks ziņot par viņa kārtējā mazuļa piedzimšanu. Laiks vilkās lēni, un Akmeņpaurim, vērojot savu bērnu rotaļāšanos, kļuva pavisam garlaicīgi. Bērni savos raibajos zvērādu apģērbos ņirbēja viņam gar acīm kā mošķu bars. “Viņu ir daudz!”, nodomāja Akmeņpauris (tas bija

pirmais loģiskais secinājums cilvēces vēsturē). Un Akmeņpauris, paņēmis savu akmens cirvi, iegreba iezīmes (katram bērnam vienu) akmenī, uz kura sēdēja. Drīz vien arī visu viņa kaimiņu alu priekšas rotāja līdzīgi akmeņi, uz kuriem paskatoties, uzreiz varēja novērtēt alā dzīvojošās ģimenes lielumu ...

Apmēram tā mēs varētu iedomāties seno cilvēku pirmo mēģinājumu skaitīt. Pēc pirmā mēģinājuma sekoja nākamie, līdz beidzot šodien mēs skaitīšanai un skaitļošanai izmantojam precīzus mehānismus un nenovērtējamas mašīnas. Palūkosim, kā noritēja attīstība.

Austrumos, kur ķīnieši slēdza biznesa darījumus ar japāņiem, indiešiem un korejiešiem, kā jau katrā kārtīgā biznesā, radās nepieciešamība rēķināt ienākumus un izdevumus. Attīstoties biznesam un pieaugot apgrozījumam, rēķināšana kļuva sarežģītāka, un viltīgie ķīnieši vēl pirms 2000 gadiem izgudroja skaitīkļus. Tie darbojās pēc vietas vērtības principa, t.i., skaitli noteica nevis pēc kauliņu skaita, bet gan pēc to vērtības atkarībā no atrašanās vietas. Šādā veidā lielus skaitļus varēja apstrādāt ar nelielu kauliņu daudzumu. Vērtības palielināšanu pārvietojot kauliņus vienā virzienā, bet samazināšanu

*Skaitīkļi ir pirmais
patiesais
skaitļošanas mašīnu
un kompjūteru
priekštecis.
Tie darbojās tā:
katra kauliņa
vērtību nosaka nevis
tā ārējais izskats, bet
gan atrašanās
pozīcija attiecībā
pret pārējiem
kauliņiem.*



*Skaitīkļi kalpoja kā atmiņas ierīce cilvēkam, kas galvā
veica skaitļošanas darbības.*

Vēl vairāk nekā tūkstoš gadu pēc tam, kad ķīnieši rēķināšanā ieviesa skaitīkļus, skaitļošanas automatizācija un matemātika praktiski vispār neprogresēja.

Senie zinātnes entuziasti gan bija izstrādājuši neskaitāmas matemātiskas formulas un teorēmas, bet visi pierādījumi, aprēķini un kalkulācija bija jāveic ar roku. Visbiežāk vesela matemātiķu grupa sēdēja vienā istabā un nedēļām vai mēnešiem

strādāja pie vienas problēmas. Lielākā daļa integrāļu, logaritmu un trigonometrisku vērtību tabulu, kas šādā veidā bija izstrādātas, gadu no gada aprēķinos tika lietotas nepārbaudītas, droši vien tādēļ, ka bija pagrūti atrast brīvprātīgos, kas būtu ar mieru veikt šādu nepateicīgu darbu. Tā tas turpinājās līdz brīdim, kad tika izveidotas mašīnas, kas varēja ģenerēt šādas tabulas un veica to ar tādu precizitāti un ātrumu, kādu cilvēku grupa nekad nebūtu spējusi sasniegt.

Reiz Šerloks Holms un doktors Vatsons, lidodami gaisa balonā, cieta katastrofā un nogāzās mežmalā. Tā kā abi nezināja, kur viņi ir nogāzušies, doktors Vatsons, ieraudzījis tuvojamies kādu cilvēku, tam uzsauca:

- Labais cilvēk, vai jūs, lūdzu, nepateiktu, kur mēs atrodamies?

Uzrunātais īsu brītiņu padomāja un atbildēja:

- Jūs atrodaties gaisa balona grozā.

Cilvēks turpināja savu ceļu, bet doktors Vatsons, manāmi apjucis, mirkšķināja acis.

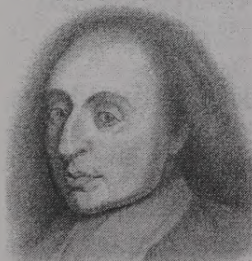
- Ak, šie matemātiķi! - pikti norūca Holms.

- Bet, Holms! Kā jūs uzminējāt, ka tas bija matemātiķis? – pārsteigti iesaucās Vatsons.

- Elementāri, mans dārgais draugs! Pirmkārt, pirms atbildēt, viņš padomāja. Otrkārt, viņa atbilde bija absolūti precīza. Un, treškārt, tā atbilde nekam nav derīga.

Neskaitāmi cilvēces vēstures momenti ir izzuduši laika gaitā. Diemžēl pasaule vairs gandrīz neatceras astronoma Keplera draugu Vilhelmu Šikardu (1592-1635), kurš uzbūvēja skaitļojošo pulksteni. Tā bija 6-ciparu mašīna, kas varēja saskaitīt un atņemt. Iespējams, ka tai bija arī pārpildīšanās indikatora zvans. Taču šīs mašīnas rasējumi tā laika karā pazuda. Kāds vīrs tos nejauši atrada 1935. gadā, taču II Pasaules kara laikā tie atkal pazuda, līdz beidzot 1956. gadā tas pats vīrs atkal atrada rasējumus, pēc kuriem 1960. gadā skaitļojošo pulksteni rekonstruēja un atklāja, ka tas nevainojami darbojas. Diemžēl līdz ar daudziem citiem atklājumiem to reti kāds atceras un piemin.

*Blēzs Paskāls
(1623-1662)*



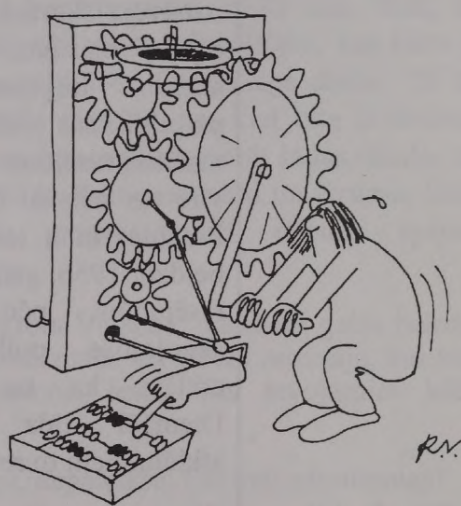
*Paskāla māte mira,
kad viņš bija vēl
mazs. Paskāla tēvs
Etjēns izlēma
pārcelties uz Parīzi*

Cilvēki mocījās, skaitļojot galvā, līdz pazīstamais zinātnieks un matemātiķis Blēzs Paskāls uzbūvēja pirmo mehānisko skaitļošanas mašīnu, lai atvieglotu sava tēva, nodokļu iekasētāja, darbu. Viņš izmantoja, papildināja un modificēja ideju, kas aprakstīta Aleksandrijas Herona darbos.

Mehāniskā skaitļošanas mašīna nenoliedzami strādāja precīzāk nekā cilvēks (darbības ātrumus salīdzināt ir pagrūti, jo nav zināms, cik ilgi bija jāgriež zobrati), taču Paskāls bija labāks zinātnieks nekā biznesmenis. Viņam izdevās pārdot tikai

kopā ar bērniem,
 Blēzu un viņa māsu.
 Mazo dēlu Etjēns
 mācīja pats, bet tikai
 viņam zināmu
 iemeslu dēļ viņš
 nolēma, ka Paskāls
 nemācīsies
 matemātiku līdz pat
 15 gadu vecumam.
 Paskāls, kuru ļoti
 saintrīgēja šāds
 neizprotams
 aizliegums, 12 gadu
 vecumā sāka
 patstāvīgi apgūt
 ģeometriju. Vēlāk
 Paskālu vairs nebija
 iespējams apturēt,
 un, kad dēls bija
 atklājis, ka trijstūra
 iekšējo leņķu summa
 ir vienāda ar 180° ,
 viņa tēvs padevās.
 Vēlāk Paskāls
 saskārās arī ar
 garīgām lietām,
 kurām pievērsās ar
 lielu interesi. Viņš ir
 sarakstījis daudzus
 reliģiska satura
 darbus.
 Vienā no saviem
 darbiem viņš ar

kādus 10-15 kalkulatorus, ja neskaita
 vairākas nelicencētas pirātu kopijas, kuras
 uzbūvēt varēja tikai retie gudrinieki.



Mehāniskā skaitļošanas mašīna

Turklāt papildus iedarbojās Mērfija likuma
 speciālgadījums: viss, kas var salūzt, arī
 salūst. Arī paskalīns lūza, bet, ja reiz tas
 salūza, neviens, izņemot Paskālu, nebija ar
 mieru uzņemties labošanu. Turklāt šāds
 kalkulators bija neizdevīgs finansiālu
 apsvērumu dēļ, jo tas izmaksāja dārgāk
 nekā cilvēki, ko tas aizvietoja. Turklāt
 parādījās arī daudzie kalkulatora ienaidnieki

*matemātiķim
piemītošu
racionalitāti pamato
ticību Dievam ar
šādu argumentu:
"Ja Dieva nav, tad
cilvēks nekā
nezaudē, tam
ticēdams, turpretī, ja
Dievs ir, tad cilvēks,
tam neticēdams,
zaudē visu."*

*Savas dzīves
vēlākajos gados
Paskāls studēja
ģeometriju,
hidrodinamiku, bet
hidrostatikas un
atmosfēras spiediena
pētīšanas rezultātā
tika radīta
hidrauliskā spiede
un Paskāla
spiediena likums.*

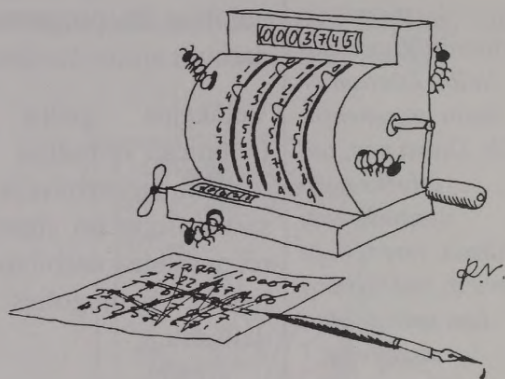
ar tehnofobijas pazīmēm: matemātiķi nobijās, ka progresa dēļ viņi var zaudēt darbu!

Vēlākajos gados vācu matemātiķis Leibnics, izstudējis Paskāla piezīmes un rasējumus, uzbūvēja mašīnu, kas līdz ar saskaitīšanu un atņemšanu varēja veikt arī reizināšanas darbības. Šī mašīna tika plaši lietota, neraugoties uz to, ka neveica dalīšanu.

1820. gadā francūzis Šarls Ksavjērs Tomas de Kalmārs izstrādāja mašīnu, kas izpildīja četras aritmētiskās pamatoperācijas: saskaitīšanu, atņemšanu, reizināšanu un dalīšanu. Gadu gaitā klāt nāca daudzi uzlabojumi, un vēlāk šis kalkulators jau varēja apstrādāt daļskaitļus, saglabāt un automātiski atjaunot pēdējo rezultātu (kaut kas līdzīgs atmiņai), un izdrukāt rezultātu. Te gan ir jāpiebilst, ka visi šie uzlabojumi tika veikti, lai gūtu lielāku peļņu, nevis lai virzītu zinātnes attīstību.

Kalmāra mehāniskais kalkulators jeb aritmometrs tika intensīvi lietots līdz pat I Pasaules karam un kļuva par pirmo vairuma pārdošanai ražoto kalkulatoru.

Arī Latvijā pirms-
kalkulatoru laikmetā
varēja sastapt
mehāniskas ierīces,
kuras firmas
izgatavotāja vārdā
sauca par
“Fēlikiem”.



Fēlikss

Čārlzs Bebidžs



Bieži tiek saukts par
“skaitļošanas tēvu”.
Bebidža bērnībā nav
nekā neparasta. Viņš
bija zinātkārs un
tiecās apgūt ne vien
matemātiku, bet arī
filozofiju, politiku un
mehāniku.

Kaut arī viņš ir
pazīstams
galvenokārt ar

Paskālu, Leibnicu un Kalmāru ar pilnām
tiesībām var uzskatīt par mehāniskās
skaitļošanas pamatlicējiem, taču mūsdienu
datoru patiesie pirmsākumi tiek saistīti ar
angļu matemātiķa Čārlza Bebidža (*Charles
Babbage*) vārdu. Kamēr Kalmārs darbojās
ar savu aritmometru, Bebidžs, Karaliskās
Astronomijas biedrības aprēķinos saskāries
ar milzīgu kļūdu daudzumu, iesaucās: “Es
lūdzu Dievu, kaut šos aprēķinus veiktu
tvaiks!” Tajos laikos tvaika mašīna bija ļoti
populāra. Un, lai cik dīvaini tas arī neliktos,
šie vārdi uzskatāmi par automātiskās
skaitļošanas sākumu.

Bebidžs izvirzīja mērķi — izmantot
mašīnas iespējas matemātikas vajadzībām.
Bebidža pirmais mēģinājums atrisināt šo

*saviem darbiem
skaitļošanas
attīstības jomā, viņš
ir arī veiksmīgi
ieviesis dažus
izgudrojumus,
standartizējis pasta
pakalpojumu
izmaksas, Grīnvičas
laika signālus u.c.*

problēmu 1822. gadā ir saistīts ar viņa izgudroto mašīnu diferenciālvienādojumu risināšanai.

Bebidžs bija pamanījis, ka daudzi darbietilpīgi aprēķini (īpaši tie, kuros jāizmanto matemātiskās tabulas) patiesībā ir darbības, kas pastāvīgi atkārtojas. Tad nu viņam radās aizdomas, ka tās varētu automatizēt. Tā Bebidžs izveidoja jau minēto diferenču mašīnu.

Tā bija darbināma ar tvaiku un izmēros varēja sacensties ar lokomotīvi. Mašīnā glabājās fiksēta norādījumu programma, kas veica aprēķinus un automātiski izvadīja rezultātus.

Pēc 10 gadu ilga darba ar šo mašīnu Bebidžu pārņēma doma strādāt ar vispārīgu skaitļošanas mašīnu, kuru viņš nosauca par analītisko dzinēju.

Izstrādātajos projektos redzams, ka šai mašīnai bija nepieciešams decimāls skaitļotājs, kas darbotos ar 50 decimāliem skaitļiem (vai vārdiem), un kam būtu nepieciešama atmiņa 1000 šādiem skaitļiem

Jaunā mašīna saturēja daudzas iebūvētas funkcijas, piemēram, nosacījumu kontroli, kas ļāva norādīt izpildāmo komandu secību atkarībā no nosacījuma. Norādījumi mašīnai

Ada Lavleisa



*Neraugoties uz to,
ka viņa tika
uzskatīta par
diezgan
vieglprātīgu, viņu
nekad neaizmirsīs.
Un nevis tāpēc, ka
viņa bija lorda
Bairona meita, bet
gan tāpēc, ka lēdija
Ada bija pirmā
programmētāja,
Bebidža asistente,
kas ieviesusi
programmēšanas
pamatelementus,
kuri tiek izmantoti
arī mūsdienu
programmētāju
darbos: sazarošanās
operators (pārbauda*

tika glabāti uz perforētām (sacaurumotām) kartītēm, kas tika “iebarotas” mašīnai caur vairākām nolasīšanas iekārtām. Dzinējs darbojās patstāvīgi. Apkalpei bija vajadzīgs tikai viens cilvēks.

Bebidža asistente Augusta Ada Kinga Lavleisa (angļu dzejnieka lorda Bairona meita), kas labi pārzināja mašīnas uzbūvi un darbības pamatprincipus, palīdzēja izstrādāt darbības plānu, kā arī iegūt finansējumu no Britu valdības, izskaidrot analītiskās mašīnas būtību publikai, veidoja arī dzinējam saprotamas instrukcijas, kas tika ievadītas skaitļotājā.

Tāpēc — gaviļējiet, sievietes, kodiet pirkstos, vīrieši, — daudzi lēdiju Adu Lavleisu uzskata par pirmo programmētāju pasaules vēsturē. Viņai par godu ASV Aizsardzības departaments vienu no programmēšanas valodām nosauca ADA.

Diemžēl Bebidža kompjuāteri nekad netika pilnībā pabeigti. Lai attaisnotu šo neveiksmi, tika minēti dažādi iemesli. Vispopulārākais ir apstāklis, ka tajā laikā vēl nebija izstrādātas precīzas tehnoloģijas. Citi min to, ka Bebidžs mēģināja risināt problēmas, kuru atrisināšana 1840. gadā būtu interesējusi tikai nelielu lietpratēju saujiņu.

nosacījumu un atkarībā no rezultāta izpilda vienu vai otru darbību), cikls (vienu un to pašu darbību cikliska atkārtošana) u.c.



Norādījumi mašīnai tika glabāti uz perforētām (sacaurumotām) kartītēm

Pēc Bebidža nāves iestājās pagaidu intereses zudums, un praktiski nenotika būtiska virzība kompjuārizācijas procesa attīstībā. Valdības komisija, izmeklējusi iespējas analītiskā dzinēja pabeigšanai, 1879. gadā nāca pie slēdziena, ka tā nav iespējama Bebidža nāves dēļ.

Kādu laiku nekādi būtiski pavērsieni nenotika, bet 1885. gadā Dorrs Felts (1862-1930) Čikāgā izgudroja komptometru, kas bija pirmais kalkulators, kurā skaitļi tika ievadīti nevis ar griešanu, bet gan vienkārši

spiežot taustiņus. Bet 1889. gadā Felts izgudroja pirmo drukājošo galda kalkulatoru.

Laika posmā no 1850. līdz 1900. gadam lielu soli uz priekšu spēra matemātiskā fizika, atklājot, ka lielāko daļu dinamisko fenomenu var aprakstīt ar diferenciālvienādojumiem. Tas izraisīja dabas parādību pētīšanas bumu, līdz ar kuru radās vajadzība pēc kādas ierīces, kas atvieglotu aprēķinu veikšanu.

"Viss, ko bija iespējams izgudrot, jau ir ticis izgudrots." — Čārlzs H. Duells, ASV Patentu birojs, 1899.

Attīstījās arī inženierbūve. Dzelzceļš, tvaikoņi, tekstilrūpniecības mašīnas, tilti — tiem visiem bija nepieciešami smalki aprēķini. Pat mazākā kļūda varēja izraisīt traģiskas sekas. Cilvēkiem bija nepieciešama spēcīga mašīna, kas varētu ātri veikt dažādus aprēķinus.

Un te 1890. gadā Hermans Hollerits un Džeimss Pauers izdomāja skaitļotājos izmantot perfokartes — vienota izmēra kartītes ar noteiktā secībā atbilstoši programmai izveidotiem caurumiem. Starp citu, viņi šo ideju nenožaga no Bebidža. Tā radās, vērojot kontrolieri izsītam caurumus dzelzceļa biļetēs. Viņi bija tie, kas



Cilvēkiem bija nepieciešama spēcīga mašīna, kas varētu ātri veikt dažādus aprēķinus.

izstrādāja iekārtas informācijas nolasīšanai no perfokartēm. Caurumi perfokartēs tika izveidoti automātiski bez cilvēku palīdzības. Līdz ar to nolasīšanas kļūdu daudzums strauji samazinājās, bet skaitļošanas procesa ātrums strauji palielinājās. Turklāt izmantojamo perfokaršu daudzums bija praktiski neierobežots. Tas deva iespēju glabāt gandrīz neierobežotu informācijas apjomu. Dažādu problēmu risinājumu perfokaršu pakas varēja glabāt un pielietot, kad vien vajadzīgs.

Visas šīs priekšrocības tika atzinīgi novērtētas. Nelielās komercfīrmas, tādi mūsdienu giganti kā IBM (*International Business Machines*), kuru dibināja pats Hollerits, Remingtons, Burrougs, uzreiz sajuta peļņas smaržu. Tirgū ātri vien

parādījās uzlabotas konstrukcijas skaitļotāji, kas izmantoja perfokartes.

Šajos skaitļotājos izmantoja elektro-mehāniskas iekārtas, kur elektrība nodrošināja mehānisku kustību, t.i., cilvēka vietā grieza aritmometra kloķi un zobratus.

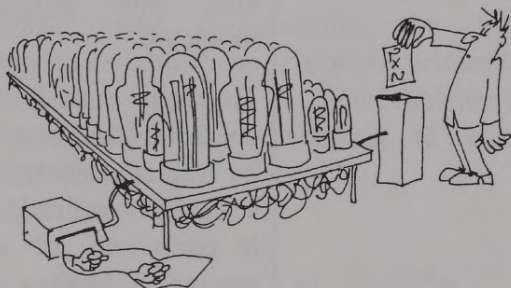
Uzlabotie skaitļotāji automātiski "iesūca" noteiktu perfokaršu daudzumu. Tie varēja ne vien saskaitīt un reizināt, bet arī kārtot. Iegūto rezultātu tie automātiski iecaurumoja izejas kartēs.

Kaut arī, salīdzinot ar mūsdienu kompjuāteriem, tā laika mašīnas bija milzīgas, lēnas un vispār neizteiksmīgas, tās veica aprēķinus biznesā un zinātnē vairāk nekā 50 gadus.

Sākoties II Pasaules karam, militāristi iedomājās, ka būtu labi izmantot kompjuāteru iespējas savā labā. Radās jauni, sarežģītākas konstrukcijas ieroči, kuriem bija jāpastāda lādiņu lidošanas trajektoriju tabulas un jāveic dažādi citi aprēķini.

1942. gadā divi Džoni, Ekerts un Močlijs, kopā ar saviem kolēģiem no Penna štata Elektriskās Inženierijas Mūra skolas, kuru vārdi vēsturei nav zināmi, nolēma uzkonstruēt ātrdarbīgu elektronisku skaitļotāju, kas varētu veikt darbu cilvēku

vietā. Viņiem tas izdevās — uzbūvētā mašina ienāca kompjuķteru vēsturē ar nosaukumu ENIAC (*Electrical Numerical Integrator And Calculator*).



ķv.

... nolēma uzkonstruēt ātrdarbīgu elektronisku skaitļotāju...

ENIAC darbojās ar skaitļiem, kuru garums varēja būt līdz 10 decimāliem cipariem. Tas varēja sareizināt divus tādus skaitļus ar ātrumu 300 vienības sekundē, atrodot katru rezultātu atmiņā noglabātajā reizinājumu tabulā. Lietojot tādu špikeri, jaunais skaitļotājs strādāja 1000 reizes ātrāk nekā iepriekšējās paaudzes skaitļotāji.

ENIAC gan nevarētu nosaukt par programmējamu, bet tas ļoti labi strādāja ar programmu, kuras veikšanai tika veidots. ENIAC mūžs ilga no 1946. līdz 1955. gadam.

Taču nevar runāt par skaitļošanas tehnikas attīstību, neminot britu matemātiķi Alanu Tjūringu (*Alan M. Turing*). 1936. gadā viņš publicēja rakstu par hipotētisku ierīci, kas vēlāk tika nosaukta par Tjūringa mašīnu, kura bija programmējama skaitļotāja imitācijai.

Tjūringa mašīna ir paredzēta loģisku operāciju veikšanai. Tā var nolasīt, uzrakstīt un izdzēst simbolu, kas ierakstīts lauciņā uz bezgalīgas papīra lentas. Šāda veida mašīna ir pazīstama arī ar nosaukumu “galīgs automāts”. Mašīna šo nosaukumu ieguvusi tādēļ, ka katrā skaitļošanas solī, mašīnas nākamā darbība atbilst kādam no galīgā sarakstā iepriekš norādītiem stāvokļiem.

Tjūringa mērķis nebija ieviest skaitļotāju, bet gan aprakstīt problēmas, kuras var loģiski atrisināt.

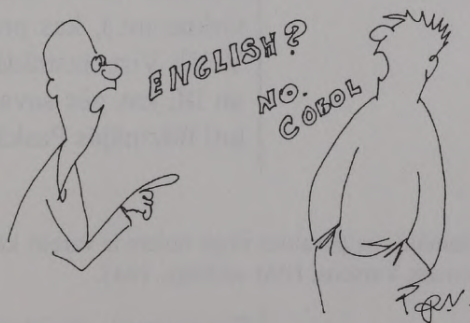
1948. gadā tika izgudrots tranzistors. Tas būtiski ietekmēja skaitļošanas mašīnu attīstību.

Tranzistori nomainīja lielās, masīvās un neskaitāmās vakuuma lampas. Līdz ar to elektronisko aparātu izmēri samazinājās.

“Nākotnē skaitļotāji svērs ne vairāk kā pusotras tonnas.”— Žurnāls “Popular Mechanics”, 1949

Tranzistoru ieviešana ievadīja otrās paaudzes skaitļotāju attīstību.

Jau 60.-to gadu sākumā liels skaits otrās paaudzes skaitļotāju tika izmantoti biznesā, universitātēs un valdībā. Tie izmantoja arī gandrīz visas komponentes, ko mēs izmantojam mūsdienų datoros: printerus, ārējo atmiņu, operētājsistēmu, programmatūru. Universālākais tā laika skaitļotājs bija IBM 1401.



Otrās paaudzes skaitļotāju laikā radās programmēšanas valodas COBOL un FORTRAN

Skaitļotāja izmantošanu patiesi efektīvu un produktīvu padarīja programmu un programmēšanas valodu ieviešana. Otrās paaudzes skaitļotāju laikā radās programmēšanas valodas COBOL (*Common Business-Oriented Language*) un FORTRAN (*Formula Translator*), kas tiek

lietotas arī šodien. Šīs valodas aizvietoja mašīnkodus ar cilvēku valodas vārdiem, teikumiem, matemātiskām formulām, atvieglojot skaitļotāja programmēšanas darbu. 60.-to gadu beigās šveiciešu zinātnieks Niklauss Virts (*Niklaus Wirth*) izstrādāja pirmo no savām programmēšanas valodām — Paskālu.

Šīs valodas prasība bija strukturēti un loģiski formulēt veicamās darbības. Turklāt tajā izmantoti dažādi datu veidi (vesels skaitlis, reāls skaitlis, simbols, simbolu virkne utt.), kas programmējot jāņem vērā. Vēlāk Virts izstrādāja arī valodas Modula-II un III, kas pēc savas struktūras un sintakses ļoti līdzinājās Paskālam.

“Es domāju, ka pasaules tirgū noiets ir varbūt kādiem pieciem skaitļotājiem.”
— Tomass Vatsons, IBM vadītājs, 1943.

Tranzistori skaitļotājos arī nebija ideālais atrisinājums, jo tie sakarsa, kas izraisīja nevēlamus blakusefektus. Džeks Kilbijs, inženieris no “Texas Instruments” uzņēmuma, 1958. gadā izstrādāja integrālo shēmu, kas uz maza, no kvarca veidota silīcija diska (kartes) apvienoja trīs elektroniskas komponentes. Vēlāk speciālisti iemanījās uz vienas kartes ierīkot vēl vairāk komponentu.

Integrālo shēmu ieviešana padarīja skaitļotājus vēl mazākus un ievadīja trešās paaudzes skaitļotāju attīstību. Tajos izmantoja operētājsistēmas, kas ļāva vienlaicīgi darbināt vairākas dažādas programmas. Viena centrālā programma (operētājsistēma) koordinēja atmiņas izmantošanu.

...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...

...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...
...the ... of ...

Informācijas mērvienības datoros

*Datori ir veidoti
operēšanai ar
diviem cipariem — 1
un 0. Kā tas ietekmē
parastu datora
lietotāju?*

Mūsdienu datori ir elektroniskas iekārtas, kas darbojas ar elektrību.

Elektroniski tiek atšķirti spriegumi - mazs spriegums vai liels spriegums; izmantojot pieņemtos apzīmējumus, 0 vai 1. Visa informācija tiek kodēta ar ciparu 1 un 0 virknēm. Šādu paņēmieni, kad informācijas pierakstā tiek izmantoti tikai divi cipari, sauc par divnieku aritmētiku jeb divnieku sistēmu pretstatā ikdienā pielietotajai desmitnieku sistēmai, kuras pierakstā izmantojam desmit ciparus no 0 līdz 9, kur pirmais cipars no beigām skaita vienus, otrais cipars — desmitus, trešais – simtus utt.

Taču desmitnieku skaitīšanas sistēma ir nekas cits kā pieradums, kas nāk vēl no tiem senajiem laikiem, kad cilvēkiem ienāca prātā sākt skaitīt. Toreiz cīņā ar pārējām ķermeņa daļām, pateicoties savam skaitliskajam pārsvaram, uzvarēja pirksti, un tā kā to ir tieši desmit, tad nu ir skaidrs, kāpēc mēs skaitām desmitniekos. Taču, ja evolūcija būtu noritējusi nedaudz savādāk un skaitīšanai būtu izmantotas, piemēram,

ausis, tad tagad mēs skaitītu divnieku sistēmā.



Visa informācija tiek kodēta ar ciparu 1 un 0 virknēm.

Ja desmitnieku sistēmā mēs skaitām 0, 1, 2, 3, 4, 5, 6, 7 utt., tad divnieku sistēmā mēs skaitām savādāk — 0, 1, 10, 11, 100, 101, 110, 111. Turklāt 100 divnieku sistēmā atbilst ciparam 4 desmitnieku sistēmā, bet desmitnieku sistēmas 100 divnieku sistēmā pierakstāms kā 110010.

Līdz ar desmitnieku pierakstu var būt arī savādāki pozicionālie pieraksti, kad mēs par skaitīšanas sistēmas bāzi izmantojam citu skaitli nevis desmit. Pozicionālais pieraksts balstās uz faktu, ka, saskaitot kāda skaitļa (bāzes) pakāpes reizinājumus ar skaitļiem, kas mazāki par bāzi, jebkuru skaitli varam iegūt tikai vienā vienīgā veidā. Tādējādi šo skaitli var pierakstīt kā polinoma reizinātājus:

$$b^n * a_n + b^{n-1} * a_{n-1} + b^{n-2} * a_{n-2} + \dots + b^2 * a_2 + b^1 * a_1 + b^0 * a_0 = c$$

kur b - bāze, a_i - reizinātāji, c - iegūtais skaitlis.

Skaitli c skaitīšanas sistēmā b pieraksta šādi:

$$a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0 \text{ [b]}$$

$$10^2 * 2 + 10^1 * 5 + 10^0 * 6 = 256_{[10]}$$

$$2^8 * 1 + 2^7 * 0 + 2^6 * 0 + 2^5 * 0 + 2^4 * 0 + 2^3 * 0 + 2^2 * 0 + 2^1 * 0 + 2^0 * 0 = 100000000_{[2]} = 256_{[10]}$$

$$16^2 * 1 + 16^1 * 0 + 16^0 * 0 = 100_{[16]} = 256_{[10]} = 100000000_{[2]}$$

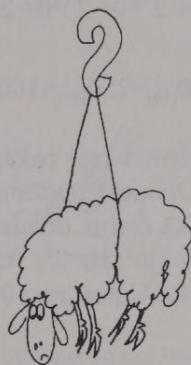
Kā var redzēt no piemēriem, bāzes pakāpes ir “apaļi” skaitļi konkrētajā 2 skaitīšanas sistēmā. Divnieku sistēmā reāli nākas darboties ar ļoti daudziem cipariem. Tā kā četrus divnieka ciparus var aizstāt ar vienu sešpadsmitnieku ciparu, jo $16=2^4$, tad programmētāji plaši lieto sešpadsmitnieku pierakstu. Cipari no 10 līdz 15 tiek rakstīti ar burtiem no A līdz F.

$0000_{[2]}=0_{[16]}=0_{[10]}$;	$0001_{[2]}=1_{[16]}=1_{[10]}$;
$0010_{[2]}=2_{[16]}=2_{[10]}$;	$0011_{[2]}=3_{[16]}=3_{[10]}$;
$0100_{[2]}=4_{[16]}=0_{[10]}$;	$0101_{[2]}=5_{[16]}=5_{[10]}$;
$0110_{[2]}=6_{[16]}=6_{[10]}$;	$0111_{[2]}=7_{[16]}=7_{[10]}$;
$1000_{[2]}=8_{[16]}=8_{[10]}$;	$1001_{[2]}=9_{[16]}=9_{[10]}$;
$1010_{[2]}=A_{[16]}=10_{[10]}$;	$1011_{[2]}=B_{[16]}=11_{[10]}$;
$1100_{[2]}=C_{[16]}=12_{[10]}$;	$1101_{[2]}=D_{[16]}=13_{[10]}$;
$1110_{[2]}=E_{[16]}=14_{[10]}$;	$1111_{[2]}=F_{[16]}=15_{[10]}$;

Katrā skaitīšanas sistēmā ir savi “desmitnieki”. Visas pozicionālā pieraksta ērtības, reizinot vai dalot ar desmit, saglabājas, tikai “desmit” ir vienāds ar skaitīšanas sistēmas bāzi. Cilvēkiem pierastā desmitnieka sistēma nav ar neko labāka vai ērtāka par citām skaitīšanas sistēmām, bet tā ir mums pierasta. Rēķināšana sešpadsmitniekos ir labs prāta vingrinājums, pamēģiniet!

Kodējot ar 0 un 1, pieraksts sanāk garš un nepārskatāms, tāpēc gadījumos, kad programmētājam jāsaprot datora valoda

(mašīnkodi), bieži tiek izmantots sešpadsmitnieku aritmētikas pieraksts.



— Tavā ganāmpulkā ir 3128 aitas!

Uz ceļa satikās gans, kas ar suņu palīdzību dzina lielu aitū ganāmpulku, un programmētājs. Programmētājs saka vecajam ganam:

— Ja es uzminēšu, cik tavā ganāmpulkā aitū, vai tu man dosi vienu?

Vecais pasmīn bārdā un saka:

— Nu, ja uzminēsi, kāpēc tad nedot.

— Tavā ganāmpulkā ir 3128 aitas!

Gans pārsteigumā noelšas, jo pareizi vien ir. Jāatdot viena aita. Gans ļauj programmētājam izvēlēties, kāda viņam patīk. Puisis izvēlas vienu, pasit padusē un grib jau iet prom, bet gans viņu aiztur:

— Klau, ja es uzminēšu, kas tu esi, vai varēšu dabūt savu aitū atpakaļ?

Puisis ir ar mieru, un veicais vīrs saka:

— Tu esi programmētājs!

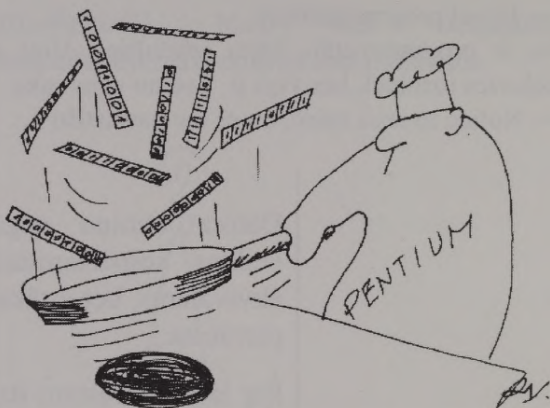
Nu ir programmētāja kārta izbrīnīties. Viņš sāk taujāt, kā vecajam izdevies uzminēt, kas viņš ir. Tad nu gans saka:

— Noliec to suni zemē, un es tev pastāstīšu ...

Datora atmiņu apgabalu izdrukā tiek veiktas sešpadsmitniekos, tas kalpo kā ekvivalents, bet ērtāks par divnieka sistēmu pieraksts.

Par laimi parastam datorā lietotājam par to nav jāuztraucas, taču daudzie skaitļi, ar kuriem saskaramies ikdienā pie datora, nāk no divnieka skaitīšanas sistēmas.

Elektroniskais signāls 0 vai 1 tiek saukts par bitu. Datorā tehniski mazākais bitu apvienojums ir astoņi. Šos astoņus bitus sauc par baitu. Reāli dators neapstrādā katru bitu atsevišķi, bet gan astoņus bitus (vienu baitu) reizē. Turklāt dators lieliski “zina”, kuri biti veido vienu baitu, un neķer tos pa labi un pa kreisi. Viens baits datoram nozīmē to pašu, ko cilvēkam nozīmē vārds, — tas tiek veidots no atsevišķiem burtiem, bet tik un tā katra vārda burti ir sakārtoti noteiktā secībā. Tā kā dators saprot tikai bitu veidotos vārdus, mēs visu ar datoriem saistīto informāciju mērām baitos.



Pentium apstrādā astoņus baitus vienlaicīgi

Pirmajiem personālajiem datoriem procesori apstrādāja ciparus pa vienam baitam. Tie bija astoņbitīgi procesori. Šodienas *Pentium* procesors ir 64-bitīgs procesors. Tas nozīmē, ka *Pentium* apstrādā astoņus baitus vienlaicīgi.

Tā kā n bitu garā 1 un 0 virknē (vārdā) var ierakstīt 2^n dažādus skaitļus (nedaudz no varbūtību teorijas - n pozīcijas, un katrā pozīcijā var atrasties viens no diviem cipariem) un datora iekārtas ir tehniski būvētas no šādām virknēm, tad datora dažādo iekārtu dažādo skaitļu ietilpība vienmēr ir kāda divnieka pakāpe. Datorā ir svarīgi, lai iekārtām būtu "apaļi" apjomi no divnieka aritmētikas viedokļa. Ērtības labad dažiem divnieka pakāpju skaitļiem, kas apzīmē baitu daudzumu, ir piešķirti

nosaukumi, kas arī ir dažādu datora atmiņas un citu ierīču apjoma mērvienības.

2^8 baiti = 1024 baiti = 1 Kb (viens kilobaits)
 2^{16} baiti = 1048576 baiti = $1024 * 1024$ baiti
 = 1024 Kb = 1 Mb (viens megabaits)
 2^{24} baiti = 1073741824 baiti
 = $1024 * 1024 * 1024$ baiti = 1024 Mb =
 1 Gb (viens gigabaits)

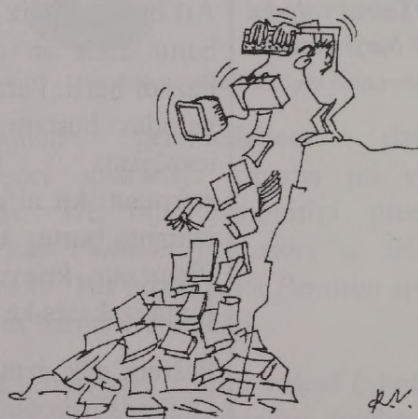
*Kāpēc katrai valodai
ir savi burti datorā?*

Arī burti pašreiz parasti tiek kodēti ar vienu baitu. Līdz ar to "alfabētā" ir 2^8 jeb 256 dažādi burti. Parasti ar to pietiek, lai katram valodas burtam piešķirtu kodu, bet nav iespējams izveidot visaptverošu starptautisku alfabētu. Tāpēc MS Windows izmanto burtus ar dažādām kodu sistēmām. Piemēram, krievu valodas teksti bez pareizā fonta izskatās kā ķeburu virtene.

Pašreiz sāk izmantot Unikodu (Unicode). Unikods ir kodu sistēma, kas izmanto divus baitus burtam ($2^{16} = 65536$ dažādi burti). Unikodā pietiek vietas visiem pasaules alfabētiem un dažādiem speciāliem simboliem, taču šajā kodā veidotie teksti aizņem divreiz vairāk vietas nekā vienbaitīgas kodēšanas gadījumā. Microsoft Windows 95 kopā ar Office 97 jau izmanto Unikodu. Tiesa, pašlaik tas rada zināmu sajukumu ar burtu kodējumiem.

*Ko nozīmē šie mega
un giga baiti
cilvēkam aptveramā
valodā?*

Mašīnraksta lapā ir apmēram divi tūkstoši burtu, tas ir apmēram 2 Kb. Uz parastās 1.44 Mb disketes mēs varam uzrakstīt apmēram 700 lappuses parasta teksta, uz 1Gb cietā diska varam uzrakstīt 524288 lappušu jeb 2621 grāmatas ar 200 lappusēm katrā. Datoru milzīgie atmiņas apjomi parastai tekstu un ciparu apstrādei nav vajadzīgi.



*Uz 1Gb cietā diska varam uzrakstīt 524288 lappušu jeb
2621 grāmatas ar 200 lappusēm katrā*

Milzīgi datu apjomi kļūst nepieciešami, darbojoties ar attēliem, skaņām un video.

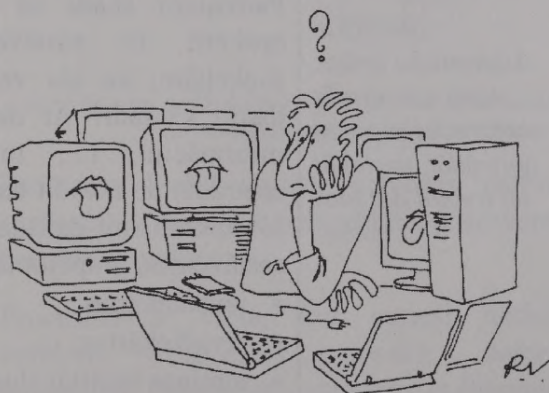
“Ar 640Kb jebkurai vajadzētu pietikt!” - Bills Geits, 1983.g.

Datoru sastāvdaļas

*Cik cilvēku, tik
atšķirīgu viedokļu.*

*Katram ar vārdu
"dators" asociējas
 kaut kas cits. Dažam
tā ir vienkārši
mašīna, citam -
iztikas vai peļņas
avots, vēl dažiem -
izklaides objekts.*

Runājot par datoriem, mēs domājam IBM
savietojamās personālos galda datorus. Ja
runa ir par cita veida datortehniku, tas tiek
atsevišķi norādīts.



Tas vienkārši par jums ņirgājas!

Ir novērots, ka liela daļa programmētāju un vienkāršu cilvēku, kas
ikdienā strādā ar datoru un kuriem it kā nav noslieces uz mistifikāciju,
tomēr pēc kāda laika, reizēm paši to pat tā īsti neapzinādamies,
personificē datoru, ar kuru strādā, kā dzīvu būtni. Liela daļa domās vai
pat skaļi sarunājas ar to, bieži vien lūdzot strādāt ātrāk vai izpildīt kādu
funkciju, ko dators šajā brīdī, šķiet, nemaz nevēlas darīt. Ar pilnu
atbildību varam paziņot, ka zinātniski fakts, ka dators būtu dzīva būtne,
ir absolūti nepamatots un nejēdzīgi aplams. Tāpēc nebrīnieties, ka
dators paliek kurls pret jūsu dedzīgajiem lūgumiem vai laiž gar ausīm
dusmu izvirdumus. *Tas vienkārši par jums ņirgājas!*

Kurā kastē tad ir tas dators? Kuras daļas datoram obligāti ir nepieciešamas?

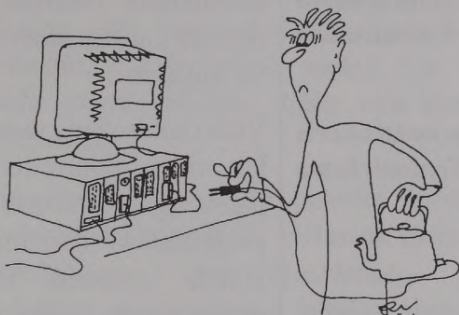
Dators ir elektroniska ierīce, kurā var ievadīt informāciju, tur to apstrādāt, uzglabāt un izvadīt dažādos veidos.

Parasts galda personālais dators sastāv no galvenā bloka, monitora, tastatūras un peles (ierīce, kas tā nosaukta savas ārējās līdzības dēļ ar oriģinālu), ak jā, un vadiem, kas visu to savieno. Savienošana ir nepieciešama, lai visas datora sastāvdaļas varētu funkcionēt saskaņoti.

Padziļinot skatu uz datoru, mēs varētu apskatīt tā sastāvdaļas atkarībā no funkcijām, ko tās veic. Bet ko tad veic datori kā tādi? Ar datoriem var apstrādāt informāciju, t.i., to ievadīt, uzglabāt, apstrādāt un izvadīt dažādos veidos. Līdz ar to apskatīsim dažādas iespējamās datora sastāvdaļas, grupējot tās kā

- ievadiekārtas;
- izvadiekārtas;
- atmiņas iekārtas ilgstošai datu glabāšanai;
- datu apmaiņas iekārtas un datu apstrādes iekārtas.

Dažas datora sastāvdaļas ir obligāti nepieciešamas tā darbam, un neviens dators bez šīm sastāvdaļām nestrādās. Savukārt citas sastāvdaļas ir nepieciešamas kādu vairāk vai mazāk specializētu funkciju veikšanai. Izņēmums varētu būt monitors, bez kā normāli ne ko jēdzīgu ar datoru darīt nevar, bet dators lieliski darbojas arī bez monitora.



Savukārt citas sastāvdaļas ir nepieciešamas kādu vairāk vai mazāk specializētu funkciju veikšanai.

Procesora bloks *Nepieciešams*

Procesora blokā ietilpst galvenās datora sastāvdaļas, tādas kā: pamatplate, procesors, barošanas bloks u.c.

Procesors *Nepieciešams*

Procesors ir vieta, kur patiesi notiek aprēķini. Procesors nosaka datora ātrdarbību. Pašreizējais standarts ir *Pentium* procesori. Vecāki procesori (286, 386) ir krietni vien lēnāki un nespēj pietiekoši ātri darbināt lielu daļu mūsdienu programmatūras. Arī 486 ir gandrīz "miris" procesors. Strauji attīstās jaunie *Pentium Pro* un *Pentium II*.

Operatīvā atmiņa *Nepieciešama*

Operatīvā atmiņa tiek izmantota īslaicīgai datu glabāšanai. Darba laikā procesors lasa instrukcijas un glabā starprezultātus atmiņā.

Tastatūras, peles, izejas printeru un

Mazas elektroniskas plates, kuras nodrošina iespēju pieslēgt datoram standarta un

**citu iekārtu
kontrolieri**

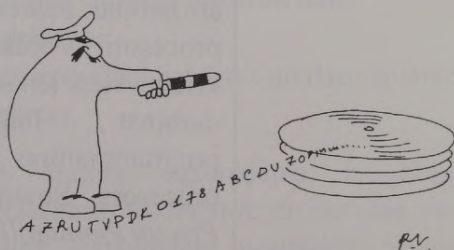
nestandarta iekārtas. Kontroliera kvalitāte dažreiz var būt nozīmīgāka nekā iekārtas, ko tas apkalpo.

**Video kontrolieris
*Nepieciešams***

Video kontrolieris vada monitora darbu. Uz video kontroliera atrodas daži datora patstāvīgās atmiņas apgabali, kas satur programmas darbam ar visām ievada un izvada iekārtām. Bez šīm programmām dators vispār nespēj strādāt. Tas nozīmē, ka videokontrolieris ir obligāti nepieciešams datora darbam, kaut gan principā dators var strādāt arī bez monitora.

Disku kontrolieris

Disku kontrolieris veic cieto un mīksto disku iekārtu vadību. Tas nodrošina ierakstošo, nolasošo galviņu pārbīdīšanu uz vajadzīgo vietu virs cietā diska virsmas.



Disku kontrolieris

Cietais disks

Cietais disks ir atmiņas iekārta. Vieta, kur pastāvīgi glabāt informāciju. Jo lielāks diska apjoms, jo vairāk informācijas (programmu, datu, tekstu utt.) jūs varēsiet

glabāt uz tā. Cietais disks ir kā grāmatu plaukts - jūs varat salikt tajā tikai tik daudz grāmatu, cik ir brīvas vietas. Lai ieliktu kādu jaunu grāmatu, kāda cita (veca) ir jāizmet ārā.

Mūsdienu programmu paketes bieži vien aizņem pāri par 10Mb vietas uz cietā diska. Spēcīgākas operētājsistēmas, teksta redaktori, elektroniskās tabulas, datu bāzes aizņem pat vairāk nekā 20Mb vietas. Standarta cietais disks šodien ir 1,2Gb. Tiem, kas strādāja ar personāliem datoriem pirms kādiem 5-7 gadiem, tas liekas fantastiski daudz, jo toreiz cietā diska apjoms svārstījās starp 20Mb un 40Mb. 80Mb un 120Mb jau tika uzskatīts par kolosāli lielu apjomu. Šodien neceriet tikt cauri ar tik niecīgiem disku atmiņas resursiem. Arī 1,2 Gb būs aizpildīti ļoti ātri. Cietajiem diskkiem un naudai ir viena kopīga īpašība: kā naudas, neatkarīgi no daudzuma, vienmēr trūkst, tā cietie diskki, neatkarīgi no apjoma, vienmēr ir pilni.

Mīksto disku iekārtas

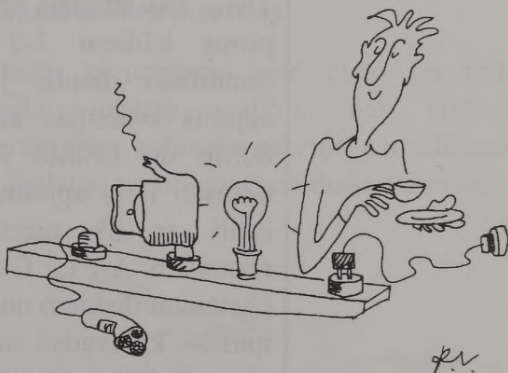
Mīkstie diskki ir atmiņas iekārtas. Atšķirībā no cietiem diskkiem mīksto disku iekārtās disketes var nomainīt, ar disketēm varam pārnest datus no datora uz datoru. Ir diva veida mīksto disku iekārtas: 3 collu un 5 collu disketēm. Pašreizējais standarts ir 3-collīgas disketes, tāpēc arvien retāk

sastopamas 5-collīgo mīksto disku iekārtas.

3-collīgās disketes ietilpība parasti ir 1,44Mb, bet 5-collīgajai 1,2Mb.

Dažas firmas pašreiz cenšas ieviest standartu 3-collīgajai disketei ar ietilpību 120Mb. Jaunas programmas agrāk tika izplatītas ar disketēm, šodien gan parasto diskešu apjomi ir pa maziem, daudzas programmas tiek izplatītas tikai ar CD-ROM jeb lāzerdisku datoru palīdzību.

Pamatplate Nepieciešama



Vieta kur uzlikt un savienot visu augstākminēto.

Vieta, kur uzlikt un savienot visu augstākminēto. Principā pamatplate sastāv no elektroniskiem celiņiem starp procesora, atmiņas un kontrolieru spraudņiem, un mikroshēmām, kas nodrošina datu apmaiņu starp šīm iekārtām.

Barošanas bloks*Nepieciešams*

Dators patērē strāvu, ko stabilizē un sadala barošanas bloks. Pieslēdzot jaunu datoru strāvas avotam, pārliecinieties, ka tas atbilst uz datora aizmugurējās sienas norādītajam. It īpaši tas attiecināms uz datortehniku, kas vesta no valstīm, kurās pieņemts no Eiropas standarta atšķirīgs spriegums.

CD-ROM iekārta

CD-ROM ir atmiņas iekārta, ar ko var nolasīt informāciju no lāzerdiskiem (~650Mb). Ir ieteicams to iegādāties, jo tagad liela daļa programmatūru instalāciju nāk tieši uz CD-ROM. Iekārta principā ir tas pats mūzikas lāzerdisku atskaņotājs, tikai mūzikas vietā var būt jebkura datoram saprotama informācija, ieskaitot arī mūziku. 6x, 8x, 12x un pat 24x CD-ROM iekārtas nozīmē, cik reižu šī iekārta strādā ātrāk par standarta mūzikas nolasītāju. Šie ātrumi lieti noder, izmantojot datu bāzes un video no lāzerdiskiem.

Rakstoša CD iekārta

Rakstošais CD ir atmiņas iekārta, kura var lasīt un rakstīt lāzerdiskus. Ierakstāmo lāzerdisku var ierakstīt tikai vienreiz, tiesa, to var darīt vairākas reizes, pakāpeniski pierakstot informāciju diskam klāt, līdz tas ir pilns.

Strīmeris

Atmiņas iekārta, kas līdzīgi magnetafonam kā datu nesēju izmanto magnētisko lenti.

Iekārtu izmanto lielu datu uzglabāšanai

kasetēs. Datu apstrādei ir nepiemērotas sava lēnuma dēļ, bet datu uzglabāšanai (drošībai un arhivācijai) tās ir noderīgas iekārtas. Standartietilpība ir 300Mb - 50Gb.

Monitors

Monitors ir izvadiekārta, kas nodrošina datora dialogu ar lietotāju. Ir dažādu veidu monitori. Tie atšķiras ar savu izšķirtspēju jeb attēlojamo punktu skaitu. Vēsturiski ir bijuši dažādi video standarti: CGA, EGA, VGA, SVGA u.c. Mūsdienās visizplatītākie ir SVGA (Super Video Graphic Adapter) monitori.



Jebkurš monitors izstaro radioaktīvo starojumu.

Jebkurš monitors izstaro radioaktīvo starojumu. Tāpēc ieteicams blakus datoriem novietot podiņus ar kaktusiem. Tikai jāuzmanās, kaktusus laistot, lai ūdens neiekļūst datortehnikā. Cita iespēja izvairīties no starojuma ir iegādāties zemas

radiācijas monitorus (*low radiation*). Lai acis mazāk nogurdinātu, var izmantot speciālus aizsargekrānus vai monitorus ar zemāku intensitāti (*low emission*). Ir arī ekrānu aizsargfiltri. Tiesa gan, likt kārtīgas firmas ražotam monitoram virsū Āzijā ražotu filtru par 5 Ls, ir tas pats, kas pēdējam Audi modelim pielikt klāt kabatas lukturīšus. Ekrāna ergonomika patiesībā ir svarīgākais parametrs, pērkot datoru biroja darbam. Datora reakcijas ātrums 20-50% robežās tekstu apstrādē nevienu šodien neinteresē, tas notiek ļoti ātri jebkurā gadījumā, bet konkrētā darbinieka veselība un spēja strādāt arī nākošgad ir svarīga.

Tastatūra

Tastatūra ir ievadiekārta. Bez tās lietotājs nevar ievadīt datorā informāciju (ja vien viņš neizmanto skeneri vai peli). Ir dažādas tastatūras, taču tām visām ir kopīgas iezīmes – vienādi izvietoti simboli (izvietojums praktiski neatšķiras no rakstāmmašīnu taustiņu izvietojuma). Simbolu izvietojums nav nejaušs. Tam ir stingri zinātnisks pamatojums. Ātrrakstīšanas tehnikā, kad darbojas visi desmit pirksti, ir svarīgi, lai veicamo piesitienu biežums dažādiem pirkstiem stipri neatšķirtos. Tāpēc, pamatojoties uz statistikas datiem par burta sastapšanās biežumu tekstā, tika izstrādāts atbilstošs

tastatūras izvietojuma variants. Protams, dažādām valodām viena un tā paša burtu parādīšanās biežums var atšķirties. Tāpēc, piemēram, franču tastatūrā atšķirībā no angļu varianta A un Q ir apmainīti vietām, vācu tastatūrā vietām apmainīti ir Z un Y, bet kirilicas tastatūra ir pilnīgi savādāka. Latviešu valodas tekstu ievadei angļu standarts gan nav īsti piemērots, taču tas ir populārākais Latvijā, kaut arī ir izstrādāta latviešu valodas ergonomiskā tastatūra, kurā ievērots latviešu valodas burtu biežums. Cilvēkiem, kas datorā vada lielus tekstu apjomus latviešu valodā, drukājot "akli", būtu vērts padomāt par ergonomiskā standarta izmantošanu.

Gandrīz visām tastatūrām uz diviem vidējiem taustiņiem (angļu standartā F un J) ir nelieli izcīlnīši, kas atvieglo atskaites taustiņu atrašanu, neskatoties uz pirkstiem.

Tastatūras ergonomika, kā tiek spiesti taustiņi, vai tie nospiežas vienmēr un lieki nedublējas, ir ļoti svarīga. Tiesa, apšaubāms ir fakts, ka, izmantojot sliktu klaviatūru, var sabojāt veselību. Pamēģinot firmas tastatūru un salīdzinot to ar lētu tastatūru, un ar pirkstiem sajusto atšķirību attiecinot arī uz datora iekšpusi, kļūst skaidrs, kamdēļ firmas datori maksā 20-30% dārgāk. Ar firmas tastatūru strādāt ir patīkamāk, taustiņus var

nospieš viegli, tie nav jāsit vairāk reižu. Smieklīgi sīkumi, bet ja cilvēks visu dienu “klabina”, tas ir svarīgi.

Pele



Visērtākā ir tā pele, pie kuras esi pieradis.

Pele ir ievadiekārta, kas atvieglo lietotāja “sazināšanos” ar datoru. Ar peles palīdzību var norādīt punktu vai apgabalu uz ekrāna. Lietojumprogrammas izmanto peli, lai veidotu dialogu ar lietotāju, piekārtojot darbību ekrāna apgabalā, ja uz tā atrodies peles norādītājam, tiek nospiesta peles poga. Ir divu un trīs pogu peles. Standarts ir divas pogas. Trīspogu peles vidējā poga tiek izmantota tikai ļoti retās programmās.

Mobilos piezīmjdatoros peles parasti ir bumbiņas, gumijas pulķīša vai jūtīgas plāksnītes veidā. Visērtākā ir tā pele, pie kuras esi pieradis.

Printeris

Printeris ir izvadiekārta. Iespējami visdažādākie printeri, sākot no adatu printeriem līdz krāsainiem lāzerprinteriem. Printeri drukā pa punktiņiem.

Adatu printeriem svarīga ir drukājošā galviņa un adatu skaits tajā. Jo vairāk adatu, jo kvalitatīvāka būs druka. Parasti ir 9 vai 24 adatu printeri. Pamatprincips, ko izmanto adatu printerī, līdzinās rakstāmmašīnas principam, jo adatu printerī simbolu “iekrāsošana” tāpat kā rakstāmmašīnā notiek ar speciālas lentas palīdzību.

Adatu printeri ir lēni un skaļi, un pašreiz vairs nav lētāki par tintes printeriem. Līdz ar to adatu printerus izmanto arvien retāk. Tie ir neatsverami reizēs, kad jādrukā caur paškopējošām formām, jo pārējo veidu printeri šo darbu nespēj veikt.

Viskvalitatīvāko izdruku var iegūt ar lāzerprinteri. Tas izmanto kaseti ar speciālu pulveri, kas tiek “uzkausēts” papīram. Kad pulveris beidzies, ir jānomaina kasete. Kaut arī šāda kasete ir dārga, tās pietiek ilgam laikam.

Tintes printeri ir lētāki nekā lāzerprinteri, un to izdruku kvalitāte bieži vien neatpaliek no lāzerizdrukas, taču tintes printeri drukāšanai izmanto ar speciālu tinti pildītus speciālus rezervuāriņus (pērkot jaunu, pievēršiet

uzmanību tam, lai nopirktu pareizo modeli), kas ātri vien iztukšojas. Līdz ar to tintes printera uzturēšana izmaksā dārgāk. Tintes printeri ļoti strauji progresē. Pašreiz praktiski visi jaunākie tintes printeru modeļi drukā arī krāsās. Daži no tiem gandrīz sasniedz foto kvalitāti. Domājams, ka, pateicoties tintes drukātājiem, mēs nākotnē drukāsim tikai krāsās.

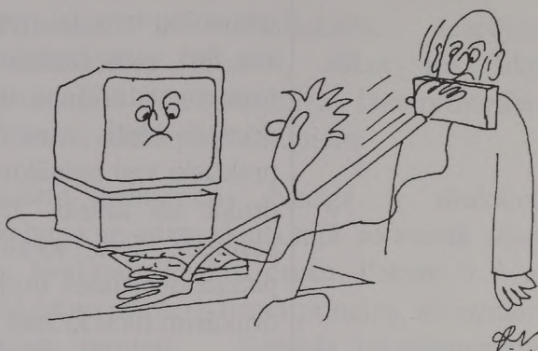
Ploteris

Ploteris ir izvadiekārta, kura velk līnijas. Mehāniska “roka” velk zīmuli pa papīra lapu vai arī papīra lapa tiek vilkta zem zīmuļa. Visbiežāk sastopams abu metožu apvienojums, papīru velk vienā virzienā, bet zīmuli — citā.

Ploteri parasti var zīmēt uz liela formāta papīriem. Tie ir vērtīgas iekārtas rasējumu izvadīšanai. Tiesa, šodien ploterus izstumj liela formāta tintes printeri, uzurpējot plotera vārdu.

Skeneris

Skeneris ir ievadiekārta, ar kuru ievada bildes datorā. Bildi novieto uz skenera tāpat kā uz kopētāja, vai arī ir mazāki skeneri, ar kuriem pārbrauc bildei pāri. Ievadītais attēls ir lielas matricas formātā, kur katrs tās elements apzīmē krāsu. Šādus attēlus var apstrādāt kopumā kā fotogrāfijas. Tos var palielināt, samazināt, izgriezt kādu daļu, bet nevar uztvert kā tekstu.



...vai arī ir mazāki skeneri, ar kuriem pārbauc bildi pāri

Burtu atpazīšanas programmas ieskanētu tekstu "atpazīst" un pārvērš to teksta redaktoram saprotamā formā. Līdz ar to tiek ietaupīts laiks un enerģija teksta ievadīšanai. Taču šāds burtu atpazīšanas process nebūt nav triviāls.

Pirms sāciat izmantot darbā tekstu atpazīšanu, pārbaudiet, vai procesa ātrums, kļūdu līmenis un procesa ērtība jūs apmierina. Skeneri var būt krāsaini un melnbalti. Iegādājoties skeneri, pievērsiet uzmanību tā izšķirtspējai - jo lielāka ir izšķirtspēja, jo kvalitatīvāks būs ieskenētais attēls.

Digitaizeris

Digitaizeris ir ievadiekārta karšu, shēmu u.c. attēlu ievadīšanai. No liela oriģināla var ievadīt precīzas koordinātes datorā. Spriežot pēc digitaizeru popularitātes, skaidrs, ka

karšu skenēšana nespēj aizstāt digitizēšanas roku darbu profesionālās aprindās.

Skaņas karte

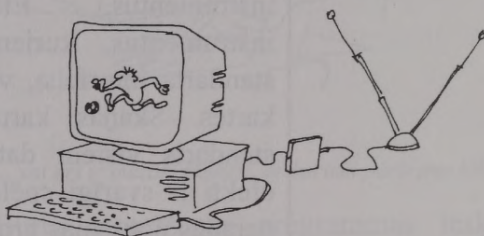
Skaņas karte ir iekārta kvalitatīvas skaņas izvadīšanai. Ar skaņas karti var klausīties un pat radīt mūziku, ja tai pieslēdz mūzikas instrumentus. Elektriskos mūzikas instrumentus, kuriem ir speciālā MIDI standarta interfeisi, var pieslēgt pie skaņas kartes. Skaņas karte visā drīzumā būs standarts visiem datoriem. Šodien skaņu efekti ir svarīgi spēlēs, taču arvien vairāk parādās nopietnas programmas, kas izmanto arī skaņu. Internetā var atrast daudz skaņu, ko bez skaņas kartes nedzirdēsim, kas liecina par skaņas nostiprināšanos datoros.

Video karte

Ir dažādas kartes video signāla ievadīšanai un izvadīšanai, kā arī video signāla kodēšanai un spiešanai. Video karte var aizstāt vai strādāt paralēli video kontrolierim. Pašreiz pasaulē valda MPEG video saspiešanas standarts. Ar MPEG standartu saspieests video aizņem daudzreiz mazāk vietas, līdz ar to reāla filma var tikt novietota uz CD-ROM, tāpat varam saņemt video klipus no Internet.

Lai noskatītos šādu video, nepieciešams atkodēt MPEG, ko pietiekami ātri dara tikai videokartes, kurām ir šim mērķim domāti speciāli procesori. Video kartes dod arī

iespēju ievadīt vai izvadīt no datora videoinformāciju videomagnetafonam, tāpat arī skatīties televizoru uz datora ekrāna. Dators ar videokarti var aizstāt TV montāžas studiju videoamatierim.



PN.

Dators ar videokarti var aizstāt TV...

Modēms, Faxmodēms

Faxmodēms un modēms ir gan ievadiekārtas, gan izvadiekārtas jeb drīzāk komunikāciju iekārtas. Ar telefona kanāla starpniecību ir iespējams pārraidīt un saņemt informāciju no cita datora vai faksmašīnas.

Viens no modēma svarīgākajiem raksturotājiem ir iespējamās pārraides ātrums, kura mērvienība ir bps (bit per second - biti sekundē). Jo lielāks ātrums, jo ātrāka ir datu pārraide. Pašreiz valdošie ir 28,8 bps, 33,3 bps ātrumi. Pateicoties Interneta bumam, tiek izdomātas arvien jaunas tehnoloģijas, lai palielinātu modemu ātrdarbību.

Tīkla karte

Tīkla karte ir komunikāciju iekārta. Ar tās palīdzību ir iespējams pievienoties lokālajam datu tīklam, lai apmainītos ar datiem un izmantotu tīkla printerus un citas tīkla iekārtas.

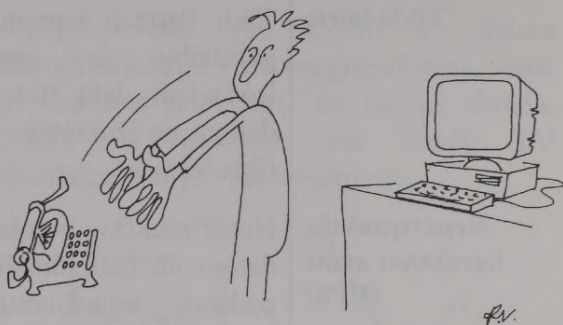
**Nepārtrauktās
barošanas avots
(UPS)**

Nepārtrauktās barošanas avots nodrošina datora un datu drošību elektrotīkla strāvas padeves traucējumu gadījumā. Tas ir neaizvietoājams serveru, komutācijas punktu un citu svarīgu tīkla vietu saslēgšanai. UPS dēļ mēs nepamanīsim mazus strāvas traucējumus, bet lielākas kļūmes gadījumā paspēsim normāli aizvērt pielietojamās paketes. Datu bāzu ieraksti, kas atstāti "pusratā", var radīt lielas problēmas un kļūmes, nākošreiz palaižot sistēmu. Vai tik mēs neskaitām naudu ar datoru!?

*Nepārciet
minimālas
konfigurācijas
datorus.*

*Pārciet "saprātīgas"
konfigurācijas
datorus.*

Datori tiek komplektēti atkarībā no tiem paredzētā pielietojuma. Principā dators var strādāt, ja tam ir pamatplate, procesors, atmiņa un videokontrolieris. Tāds dators gan neko lielu nevarētu padarīt, bet varētu būt specifiski pielietojumi, kur pietiek ar šādu datoru. Normāli biznesa datoram ir vēl cietais disks, mīksto disku iekārta, monitors, tastatūra un pele, kas arī būtu šodienas parasta datora konfigurācija. Datora iekārtu daudzums un ietilpība jānosaka pēc uzdevumiem, ko tas veiks.



Datora iekārtu daudzums un ietilpība jānosaka pēc uzdevumiem, ko tas veiks.

Apskatīsim piemērus, kas ilustrē datora izvēli. Datoru pirkšanā var būt divas radikālas stratēģijas:

*Pirmā datoru
pirkšanas stratēģija.
Ir zināma funkcija,
kas jāizpilda, un
neko citu mēs
negribam.*

Mums ir kāds konkrēts uzdevums, kas jāveic ar datora palīdzību, un neko citu mums nevajag. Šādā gadījumā jāpērk lētākais dators, kas veic vajadzīgo funkciju. Jāpaskatās tikai, lai būtu saprātīgas garantijas uz laiku, kad mēs datoru lietosim (jo vairāk, jo labāk), un lai firma, kurā dators nopirkts, nepazustu jau nākamajā mēnesī.

*Maksimālās
funkcijas stratēģija.
Mēs vienmēr gribam
būt progresa viļņa
virspusē.*

Mēs gribam turēties līdzī laimeta elpai un izmēģināt iespējas, ko sniedz jaunās programmas, kas ik pa brīdim parādās tirgū. Šajā gadījumā ir jāpērk dators par visu naudu, ko vien var atļauties izdot šādam mērķim, un jāiegādājas pēc iespējas jaunāks

un labāks dators. Tomēr ir jāņem vērā, ka datoru progress ir straujš un nežēlīgs, un tas, kas šodien ir jauns, pēc gada ir jau novecojis.

Principā visbiežāk, iegādājoties datorus, mēs vadāmies pēc abām stratēģijām, cenšoties atrast vispieņemamāko kompromisu.

Ja neesat tehnoloģijas "mīlis", tad jums droši vien datora izskats un saskaņa ar mēbelēm būs svarīgāka nekā cash izmērs.

Protams, svarīgākie datora parametri ir procesora ātrums, operatīvās atmiņas daudzums un citā diska izmērs. Šie ir arī visvieglāk saprotamie parametri, taču visi citi parametri arī ir svarīgi, un par to kvalitāti bieži vien ir vērts piemaksāt. Jāatceras, ka fiziski jūs skatīsieties uz ekrānu un darbosieties ar klaviatūru un peli. Ja jūs tāpat kā vairums citi datora lietotāju apstrādāsi tekstus, šie faktori būs svarīgāki nekā par 20% ātrāks procesors.

the latest thing. There is no such thing as a free lunch. The only way to get ahead is to work hard and to be honest. The only way to get ahead is to work hard and to be honest.

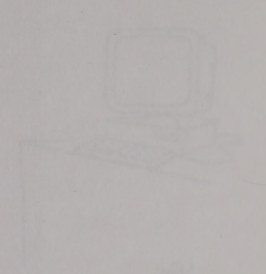
Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.



Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Principles of the American Medical Association. The American Medical Association is a non-profit organization. It is a voluntary association of medical practitioners. It is a non-profit organization. It is a voluntary association of medical practitioners.

Daži ar datoriem saistīti jēdzieni

Algoritms

Algoritms ir viennozīmīga instrukciju virkne, kas uzrakstīta izpildītājam saprotamā valodā.

Droši vien katrs ir dzirdējis vārdu “algoritms” un ir lieliski sapratis, kas ar to ir domāts, kaut arī nav pūlējies šo jēdzienu definēt. Algoritms ir precīzs priekšraksts kāda uzdevuma veikšanai. Citiem vārdiem sakot, algoritms pasaka izpildītājam, kā veikt uzdevumu, paredzot visas iespējamās situācijas un kļūdas, nepieļaujot nekādu novirzīšanos un pašdarbību no izpildītāja puses.

Datoru algoritmi ir precīzas instrukcijas datora valodā - mašīnkodos.

Labs algoritma piemērs ir mūzikas notis. Pēc notīm var izpildīt muzikālu skaņdarbu, jo notis viennozīmīgi nosaka, kādus taustiņus, kādā secībā un cik ilgi jāspiež uz klavierēm. Protams, dažādu klavieru skaņa var atšķirties, un katrs izpildītājs pieliek klāt arī kaut ko no sevis, taču tas netraucē klausītājam atpazīt skaņdarbu, jo notis ir viens no retajiem precīzajiem algoritmiem, kas uzrakstīti cilvēkiem.

Robotam viss būtu jānorāda precīzi, pat apraksts “pēc garšas”. Te gan rastos problēmas, jo “garša” no

Pavārgrāmatās sniegtie algoritmi gan ir jau stipri neviennozīmīgi. Bieži vien garšvielu daudzums tiek noteikts “pēc garšas”, un daudzie tehnoloģiskie procesi paliek neizklāstīti. Tāpat netiek norādīts, ko darīt

*“garšas” dažādiem
cilvēkiem var
atšķirties. Tad
laikam nākotnē,
pērkot virtuves
robotu, būtu jālūdz,
lai tas tiek
pārprogrammēts
atbilstoši ģimenes
garšas izjūtai.*

dažādās iespējamās, bet nestandarta situācijās. Tas tamdēļ, ka laba pavāra pieredze aizpilda šos robus. Arī blodiņas, katla un pannas izmēri netiek norādīti. Bet, ja mazs bērns, palīdzot mammai, interpretē to pēc savas saprašanas, tad viņam gan jāsaka: “To lielo pannu nē, padod to ar sarkano kātu, lūdzu!” Robotam visas situācijas jāapraksta, savādāk tas sajūk prātā.



Robotam visas situācijas jāapraksta, savādāk tas sajūk prātā.

Par algoritmiem mēdz teikt: labs vai slikts, ātrs vai lēns. Vērtējums, vai algoritms ir labs vai slikts, ir diezgan subjektīvs, tāpēc bieži vien algoritma kritērijs ir tā ātrdarbības novērtējums.

Uzrakstīt pietiekoši labu, precīzu un ātru algoritmu nemaz nav tik vienkārši. Tāpēc zinātnieki un pat veseli institūti nodarbojas ar algoritmu pētīšanu, uzlabošanu un izstrādi. Algoritmu teorija ir matemātikas nozare, bet algoritmu un programmu izstrāde ir informācijas tehnoloģijas lielākā daļa un faktiski ir tās “virzošais spēks”.

Konteksts

Virspusēji runājot, dators ir smiltis un metāls plastmasas kastē, un nekāda konteksta.

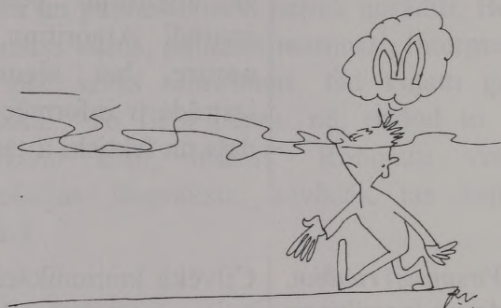
Cilvēku komunikācijā vienmēr ir konteksts (nejauciet ar zemtekstu). Konteksts — tā ir lietoto jēdzienu precīzās nozīmes izpratne. Vienīgais konteksts, ko saprot dators, ir mašīnkodu izpildīšana. Turpretī cilvēki reti kad viens otram dod pilnīgi viennozīmīgu informāciju. Iztrūkstošā informācija tiek iegūta no pagātnes pieredzes un konkrētās situācijas analīzes. Cilvēkiem ir saprotams situācijas konteksts. Tiesa, lielākā daļa konfliktu rodas tādēļ, ka diviem (vai vairākiem) indivīdiem ir dažādas konteksta izpratnes.

Daudzas lietotājprogrammas (Word, Excell uc.) patiesībā līdzīgi mašīnkodiem, kas domāti cilvēkiem, nodrošina augstāka līmeņa kontekstu,

Iedomājieties, ka jums jānorāda ceļš uz McDonalds ātras ēdināšanas restorānu. Vecmāmiņai no laukiem būtu saprotama norādījumu virkne: “Braul līdz Pareizticīgo katedrālei, ej garām Brīvības piemineklim, redzot māju ar bulciņām skatlogos un dzeltenu, līkkājainu burtu M, zini, ka esi

*radot virtuālu telpu,
kas ir saprotama
datoram un ērta
lietotājam.*

klāt.“ Tas tādēļ, ka vecmāmiņa labi zina, kas ir Brīvības piemineklis, bet pieredze ar McDonaldu viņai vēl nākotnē.



...jo amerikānim, labi zinot, kas ir McDonalds...

Bet, ja tas pats būtu jāstāsta kādam amerikānim, jūsu iepriekšējais apraksts izrādītos nederīgs, jo amerikānim, labi zinot, kas ir McDonalds, nav ne jausmas par to, kas ir Brīvības piemineklis. Tādēļ būtu tas jāapraksta: “Slaida, zaļa meitene ar trim zvaigznēm rokās augsta staba galiņā.”

Mēs jau runājām par pavāriem. Tiem problēmas rada precīza garšvielu nomērīšana un precīza laika, un temperatūras ievērošana procesa gaitā. Automātiem ar precīzu, ieprogrammētu algoritmu tā nebūtu problēma. Taču radīt sarežģītākus kontekstus, lai “cilvēcīgi” varētu robotam izstāstīt, kā jācep karbonāde, ir problēma.

Par kontekstu mēs varam pastrīdēties arī filozofiskā līmenī, bet šeit nav vietas to darīt. Konteksta veidošana ir savā ziņā vienīgais programēšanas uzdevums, tamdēļ svarīgi izprast jēdzienu un problēmu, ko pilnībā līdz galam nav iespējam izdarīt. Ir vai nav, un kas ir konteksts atstāsim sarunai pie vīna glāzes.

Vide

Vide parasti ir ekoloģisks termins. Arī ķīmijai šis vārds nav svešs.

Mēs varam nonākt draudzīgā, sausā vai draudīgā vidē. Vide ir kāds apstākļu kopums, kas ietekmē mūs kompleksi. Mēs varam nonākt vidē, kurā runā tikai spāniski. Līdzīgi arī datorprogramma var nodrošināt vidi citām programmām vai lietotājam, kurā tiek saprasta kāda valoda vai signāli. Principā programma realizē augstāka līmeņa kontekstu, radot iespēju strādāt ar sarežģītākiem jēdzieniem, kas veido šo jauno vidi.

Vadība

Vadības psiholoģiskais analogs ir apzinātā uzmanība. Tā īsti mēs spējam darīt tikai vienu darbu katrā laika sprīdī, jo mēs spējam

Vadība laikam ir mistiskākā lieta, par ko var runāt datoru sakarā. Tā īsti aizfilozofējoties, mēs nekādā jēgā netiksim, tamdēļ lēnāk. Datora procesors izpilda komandu pēc komandas no koda, kas ir ielikts tā atmiņā. Tā kā šis kods reaģē (kods reaģē, ha!!!) uz klaviatūru un peli, mēs

*koncentrēt savu
uzmanību tikai uz
vienu darbu.
Protams ir iespējami
dažādi "autopiloti",
piemēram, vadot
auto, mēs varam
plāpāt, klausīties
radio. Datoru
valodā tas jau būs
daudzuzdevumu
režīms ar uzdevumu
fonā.*

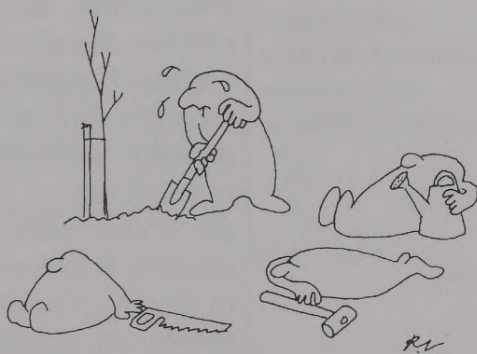
varam ietekmēt tā izpildes gaitu. Parasti pēc datora ieslēgšanas mēs palaizām programmu, ar kuru vēlamies strādāt. Šai programmai tiek nodota vadība, tā savā ziņā dzīvo. Procesors izpilda šīs programmas kodā pierakstītās darbības, un šī programma interpretē klaviatūras, peles, datortīkla un citus datoram uztveramus notikumus.

Iepriekšējā operētājsistēmā DOS, kas vēl šodien daudz kur sastopama, vadības nodošana un atdošana bija intuitīvi viegli saprotama. Uzsākot programmu, operētājsistēma tai nodeva vadību, programmai darbu beidzot, vadība atgriezās operētājsistēmai. Ja programma pati spēja "palaist" citas programmas, piemēram Norton Commander, tad vadība atgriezās pie tās pēc palaistās programmas darba beigšanas.

Windows vidē tas viss izskatās stipri nesaprotamāk, mums vienlaicīgi tiek palaistas vairākas programmas, un mēs varam lēkāt starp tām. Windows dažādās versijas, cita labāk, cita sliktāk, realizē daudzuzdevumu vidi. Tas nozīmē, ka tiek radīta sistēma, kad vairākas programmas var tikt izpildītas vienlaicīgi.

Vadība, protams, ir procesora "uzmanība", precīzāk - laiks, ko tas velta konkrētas

programmas kodam. *Windows* gadījumā šī vadība tiek sadalīta mazos gabaliņos laikā, tādējādi radot ilūziju, ka ir daudz vadību un daudz programmu tiek izpildītas vienlaicīgi, kaut gan īstenībā vienā brīdī tiek izpildīta tikai viena komanda.



... kaut gan īstenībā vienā brīdī tiek izpildīta tikai viena komanda

Vadība ir datora programmu dzīvības eliksīrs, tikai saņemot vadību, programma kaut ko var paveikt. Te nu beigsim atkal ar mazu filozofiju: programma neko neveic, procesors dara, izpildot tās kodu jeb drīzāk gan daudzu programmu kodus pa mazam gabaliņam. Ērtāk gan ir runāt, ka programmas strādā un dators ir tā vide, kur tās var strādāt.

Draiveris, Dzinis

Draiveri ir speciālisti programmu pasaulē.

Katrs draiveris specializējas uz savas iekārtas apkalpošanu, nodrošinot tai citām programmām pazīstamu interfeisu.

Iekāpjot taksometrā, mēs principā sākam to vadīt. Piemēram: "Man lūdzu uz staciju!" Taču šo mūsu vēlmi stūres pagriezienos un pedāļu spiedienos pārvērš taksometra vadītājs. Datora operētājsistēmā arī ir programmas, sauktas par draiveriem, kas pārtulko augstāka līmeņa prasības konkrētās iekārtu komandās. Piemēram, lietojuma programmām nav jāzina visu pasaules printeru komandas, tām ir jāprot veikt pieprasījums draiverim. Savukārt, pievienojot jaunu iekārtu datoram, ir arī jāpieliek tā draiveris, lai dators zinātu, kā ar to apieties. Visi printeru draiveri prot vienu valodu un izpildīs pieprasījumu cik labi vien spēs, protams, adatu drukas kvalitāte atšķirsies no lāzerdrukas. Arī rikšas vadītājs centīsies aizvest jūs līdz mērķim, un ir gadījumi, kad rikša ir vislabākais transporta veids. Svarīgi, lai rikšas vadītājs runātu, teiksim, angļiski.

Instalācija

Instalācija ir kārtējais angļu vārds ar latviešu valodas tulkojumiem "ievadīšana, ierīkošana, uzstādīšana,

Programmu instalācija patiesībā ir to uzstādīšana uz datora, un viss. Uzstādīšana gan tādā nozīmē, ka konkrētu programmu var lietot uz šī datora.

Jauna darbinieka ievadīšana firmas aprītē ir stipri līdzīga instalācijai. Mums ir

iekārta". Tā vietā, lai lietotu kādu no šiem tulkojumiem, mēs pārvēršam šo vārdu par terminu ar specifisku nozīmi. Tā mākslinieks sasit divus vecus dēļa galus un galā uzliek pusizdzertu šnabja pudeli, un nosauc to par instalāciju vai hapeningu, vai vēl kaut kā.

jāiepazīstina jaunais darbinieks ar viņa pienākumiem un arī ar firmas tradīcijām, instrukcijām un procedūrām. Arī firmas darbinieki jāiepazīstina ar jauno darbinieku un viņa pienākumiem.



Arī firmas darbinieki jāiepazīstina ar jauno darbinieku un viņa pienākumiem.

Šodien programmas reti kad ir viens .exe vai .com fails kā kādreiz, tās drīzāk ir daudzu koda un datu failu kongloremāts. Turklāt *Windows* vide pieprasa programmas reģistrāciju dažādos *Windows* reģistros. Šie iepriekšminētie faktori neļauj daudz maz sarežģītu *Windows* programmu vienkārši uzkopēt uz datora diska.

Programmas uzstādīšana ir kļuvusi par sarežģītu procesu, kas bieži vien jāveic profesionālim, īpaši, ja programmas uzstādīšana saistīta ar kādām izmaiņām

visas sistēmas darbībā. Šinī gadījumā arī dators, iespējams, būs vairākas reizes jāpārstartē. Diemžēl šodien programmas ir tik cieši saintegrētas un tik daudz sadarbojas savā starpā, ka kļūdas un dažādi konflikti ir praktiski neizbēgami. Līdz ar to kādu jaunu programmu uzstādīšana var beigties ar daudzu citu programmu nepareizu darbību, kas nozīmē visas sistēmas pārinstalāciju, kas, savukārt var nozīmēt vairāku dienu darbu profesionālim, ja uzstādīto programmu ir daudz un tās ir sarežģītas. Ja nevēlaties piedzīvojumus, strādājiet ar pārbaudītām instalācijām.

Situāciju apstrāde

Datora programmām jāspēj apstrādāt visas situācijas— gan parastās, gan negaidītās.

Ja pavārs pacietīgi gaidītu olu izcepšanos, līdz olas baltums paliek balts, tad viņš arī paliktu gaidīt uz mūžiem tajā gadījumā, ja būtu uzlicis pannu uz saplīsušas plīts. Stāstot kādu rīcības plānu cilvēkiem, mums nav jāapraksta visas iespējamās situācijas. Ar programmām ir savādāk. Ja programmā iespējams ievadīt tādus datus vai programmas darba gaitā tā saspaidīt taustiņus, ka programma “uzkaras”, t.i., neizrāda nekādas dzīvības pazīmes, tad saka, ka šī programma nav “mulķu droša”. Tas, protams, nav jāuztver personiski, ja jums ir izdevies “uzkārt” programmu.

Programmējot lielāko laiku aizņem dažādu blakus situāciju aprakstīšana, nevis programmas pamatfunkcijas izstrāde.

Pārtraukumi

Dažādas gaidītas un negaidītas situācijas datori apstrādā ar pārtraukumu palīdzību.

Zvanot telefonam, mēs droši varam uz brīdi pārtraukt kartupeļu mizošanu un pēc brīža to atkal turpināt. Olai piecepoties, mēs uz brīdi pārtrauksim telefona sarunu.



Zvanot telefonam, mēs droši varam uz brīdi pārtraukt kartupeļu mizošanu un pēc brīža to atkal turpināt.

Līdzīgi tas notiek arī datoru pasaulē. Dators pārtauc programmu, lai apstrādātu kādu situāciju. Datorpārtaukumam beidzoties, vadība vienmēr tiek atdota programmai pārtrauktajā vietā.

“Uzkāršanās”

Pēkšņi datora ekrāns sastingst un

“Uzkāršanās”, “nosprāgšana”, “ieiešana sevī” u.c. mīļvārdiņus izmanto

*nekādi nereaģē uz
taustiņiem.*

datorlietotājs, aprakstot situāciju, kad dators pārtrauc darbu, bieži vien pazaudējis vairāku stundu darba rezultātus. Šādas situācijas gadās visiem, kas mēģina strādāt ar kādu jaunu, neaprobētu programmu.

Uzkāršanās — tā ir negaidīta situācija programmai, ko tā nepareizi apstrādā. Principā šādos gadījumos dators strādā it kā mūžīgā ciklā, t.i., bezgalīgi turpina darīt vienu un to pašu, jo “dzelžiem” taču nav ne vainas. Pārlādējot datoru, tas atkal strādās, un bieži vien var gadīties tā, ka jums pat vairs neizdosies atkārtot situāciju, kas radītu uzkaršanās efektu.

Protams, šādu pašu efektu var panākt, sabojājot fiziski kādu datora komponenti strādāšanas laikā. Piemēram, uzliekot piezīmjdatoru uz spēcīga magnēta un tādējādi sabojājot tā cieta disku uz visiem laikiem. Šis gadījums nebūs “uzkaršanās”, bet vienkārša datora “nobeigšana”.

Vīrusi

*Nav vienota viedokļa
par to, vai vīrusi ir
dzīvas būtnes, vai
nav. Tiem piemīt,
daudzas dzīvu būtnu
īpašības, bet dažas
no tām trūkst. Savā*

Šajā grāmatā mums vīrusi interesē savas spējas vairoties dēļ, parazitējot uz nesējorganisma. Labvēlīgos apstākļos vīrusi spēj vairoties un izplatīties lavīnveidīgi.

Nezinām, vai pirmie datorvīrusi radās tamdēļ, ka to radītāji bija smēlušies

*ziņā vīrusi mākslīgā
intelekta literatūrā
figurē kā dabas
piemērs pārejas
posmam starp dzīvo
un nedzīvo dabu, kā
demonstrācijas
piemērs tam, ka
nekas nav "melns
vai balts" uz šīs
pasaules.*

iedvesmu no to bioloģiskajiem prototipiem, vai arī pēc tam, kad neatkarīgi radās pirmais datorvīruss. Vīrusam līdzīgās uzvedības dēļ to nosauca par vīrusu.

Datoru vīruss ir maza programma, kas 'pielīp' parastai programmai, "aizņemas" no tās vadību un nostrādā. Vīrusa nostrādāšanas laikā tam noteikti jācenšas izplatīties tālāk, tātad jācenšas pielipt kādai citai programmai.

Dzīvības ekvivalents datoru pasaulē ir vadība jeb spēja panākt, ka tiek izpildīts vīrusa programmas kods. Šeit atkal varam paveldzēties domās par brīvo gribu, un to, kas īsti kuru vada: vīruss procesoru vai procesors vīrusu. Vīruss neapšaubāmi ir tikai programma, atšķirībā no sava bioloģiskā prototipa, tā pavisam droši ir nedzīva parādība.

Programma pat nav lieta, programma ir informācija, kas savā ziņā dzīvo, darbojas, ja ir šī aktīvā vide, kura to darbina. Grāmata, fiziskais papīrs ir vide, uz kuras ar tinti var uzlikt burtus, kas veido informāciju. Kamēr burti 'gul' grāmatā, informācija ir nedzīva, kad kāds to sāk lasīt un interpretēt, tā sāk dzīvot - rasējumi pārvēršas kuģos un receptes ēdienos.

Tāpat varam uzskatīt, ka dzīvo tādas

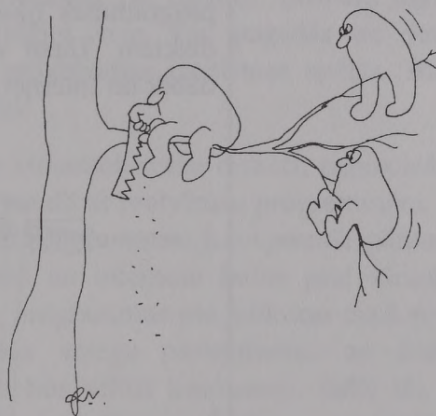
netveramas parādības kā slikts vai labs garastāvoklis. Skaists smaids rada nākošo smaidu, un tā tas turpinās. Kā tur bija ar to brīvo gribu!?

Atgriezīsimies pie programmu kodiem. Uz cietā diska mēs uzinstalējam programmas, lai kādreiz tās izmantotu. Mēs liksim operētājsistēmai ielikt programmas kodu atmiņā un padot tam vadību (uzstādīt procesora vadības reģistru uz programmu). Tālāk procesors izpildīs programmas kodu komandu pēc komandas. Vīrusam ir tikai jāiespraucas šajā komandu virknē ar savu personīgo kodu. Nākošreiz izpildot šo komandu virkni, izpildīsies arī vīrusa kods.

Ja šis iespraukšanās gājiens tiks izdarīts nekorekti, tad programma vairs nestrādās vispār vai strādās nepareizi, un visdrīzāk vairs netiks darbināta. Ja vīruss savā pirmajā darbības reizē arī izdarīs kādu draiskulību, piemēram, izdzēsīs visu informāciju uz diska kopā ar sevi, tam būs mazas iespējas plaši izplatīties. Tās slimības, kas ir momentāni nāvējošas, nevar būt lipīgas.

Līdz ar to vīrusiem ir latentais periods, kad vienīgas, ko tie dara, ir vairošanās. Tie meklē programmas uz diska - failus ar paplašinājumu .exe, .com, .dll. Arī diska un

diskešu vadības (boot) sektorā ir kods, kas tiek izpildīts, un tur var ieperināties vīruss.



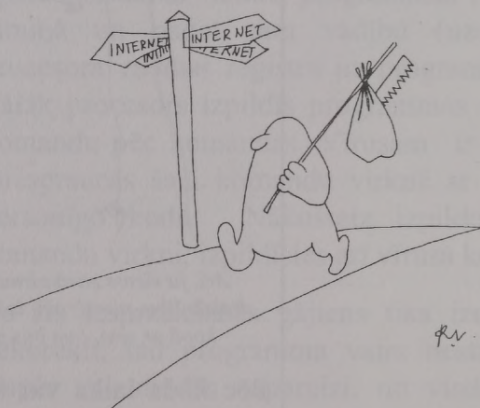
Arī, ja vīruss savā pirmajā darbības reizē izdarīs kādu draiskulību, piemēram, izdzēsīs visu informāciju uz diska kopā ar sevi, tam būs mazas iespējas plaši izplatīties.

Pēc kāda laika vai kādā konkrētā datumā, piemēram, Visu Svēto Dienā (Halloween) vīruss veiks savu netīro darbu.

Jāsaka gan, ka tādu īsti ļaunprātīgu vīrusu nav daudz un pret tiem cīnās nežēlīgi, taču pavisam nekaitīgi, humorīgi vīrusiņi klejo pa pasauli milzīgā daudzumā. Tas tikai liecina par cilvēku lielo vēlmi tēlot Dievu, un radīt vismaz nosacīti dzīvas būtnes.

Tātad vienīgais veids, kā dabūt vīrusu, ir atļaut tam darboties uz sava datora, ko savukārt panāk, darbinot vīrusainas

programmas. Vīrusainas programmas varam dabūt, instalējot programmas no nezināmām disketēm. Ieteicams būtu instalēt programmas tikai no oficiālajiem ražotāju diskietēm. Tāpat vīrusainas programmas var dabūt no Internet.



Tāpat vīrusainas programmas var dabūt no Internet.

Šodien daudzas pielietojuma programmas ļauj rakstīt makrokomandas, kas rada papildu vidi vīrusiem. Word un Excel makrovīrusi izplatās ar šo programmu failiem, proti, failiem ar paplašinājumu .doc un .xls. Makrovīrusi “dzīvo” pa šiem failiem un ar citiem vīrusiem “nekrustojas”. Internetā parādās aktīvais saturs: kustīgās bildes vai interaktīvas anketas, spēles, kas

arī ir programmas, kurās principā var ieperināties vīrusi.

Pret vīrusiem var cīnīties, pirmām kārtām, neinstalējot visu, kas pagadās uz datora, īpaši apšaubāmas izcelsmes spēles. Būsim izvēlīgi!

Pirms izmantot svešu disketi, nepieciešams to pārbaudīt ar pretvīrusu programmām. Var instalēt programmas, kas vienmēr pārbaudīs disketes un interneta failus pret vīrusiem. Šādas programmas pie jebkuras aizdomīgas darbības sniegs paziņojumu, no kuriem daudzi būs viltus trauksmes, tādēļ tās nav plaši izplatītas. Šādas programmas ieteicams lietot īpaši naidīgā vidē. Analogijas no dzīves mēs neminēsim.

Protokols

Procesus, kuri atkārtojas, mēdz aprakstīt, lai katru jaunu reizi nebūtu jāizdomā ritenis.



Ne vienmēr protokoli ir atbilstoši konkrētai situācijai, bet tie ievieš un uztur kārtību.

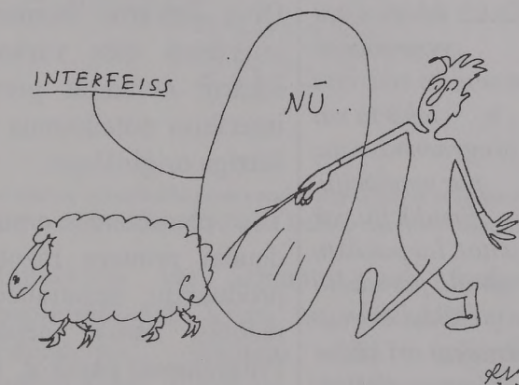
Karaļi viens otru ciemos aicina pēc protokola. Arī galda rīki tiek sakārtoti pēc protokola. Nazis pa labi un dakšiņa pa kreisi. Mēs zinām, kur tiem jābūt, un varam pateikt, vai protokols ievērots, vai ne. Ne vienmēr protokoli ir atbilstoši konkrētai situācijai, bet tie ievieš un uztur kārtību. Ja mēs neatšķirtu nazi no dakšiņas mēs zinātu, ka tas pa labi ir nazis. Tas, kurš pirmais nesa aizgājušā komunistu partijas ģenerālsekretāra zārku, bija nākošais ģenerālsekretārs. Datu pārraidē protokols ir datu formāts vai formātu secība, lai datu apmaiņas programmas viena otru saprastu. Piemēram, kāds teksta vēstījums pa datortīklu visdrīzāk tiktu sūtīts pa daļām. Sūtošā programma vispirms aizsūtītu ziņojumu, ka būs teksts 7 daļās, kas saturēs informāciju, garumā 777 katra. Pēc tam nāks ziņojums, kas saturēs informāciju, ka tas ir vēstījums Nr. 1, garumā 777, un tā visi vēstījumi. Saņemošā puse paziņo par vēstījumu saņemšanu, vajadzības gadījumā vēstījumi tiek atkārtoti. Šāds protokols nodrošina datu integritāti arī pie sliktiem sakariem.

Datoros tiek izmantoti ļoti daudz dažādu protokolu. Piemēram, Ethernet, ActiveX, TCP/IP ir protokolu nosaukumi, kas vēlāk var ieiet citos nosaukumos. Ethernet karte ir

datora tīkla karte, kas strādā pēc Ethernet protokola.

Interfeiss

Dabā mijiedarbojoties objektiem, tiek izmantota kāda signālu sistēma. Tie ir signāli, ko viens spēj pārraidīt un otrs uztvert. Šo signālu un to nodrošinājuma iekārtu sistēmu tehnikā sauc par interfeisu.



Tie ir signāli, ko viens spēj pārraidīt un otrs uztvert.

Runājot par datoriem, tiek minēti dažādu līmeņu interfeisi. Piemēram, *virtuālās interfeiss* ir savienojumu, vadu un datu pārraides protokolu sistēma, ko izmanto dažādas iekārtas, piemēram, pele un modems.

Lietotājiinterfeiss ir programmas komandu, ziņojumu, ekrānpogu un izvēlņu kopa, kas lietotājam dod iespēju komunicēt ar programmu tās kontekstā.

Programmu mijiedarbībai tiek definēti visdažādākie interfeisi. Tāpēc katrā

konkrētā gadījumā ir jāprecizē, kāds interfeiss tiks lietots.

Emulācija

Kādas iekārtas vai programmas tēlošanu, ko veic cita iekārta vai programma, sauc par emulāciju.

Emulāciju var izmantot, lai panāktu oriģinālā paredzēto efektu, bet var sasniegt arī kādus jaunus efektus.

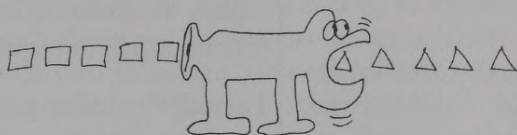
Divu iekārtu komunikācijā caur kādu interfeisu mēs varam vienu "oriģinālo" iekārtu nomainīt pret citu, kas ievērotu interfeisa noteikumus un visādi citādi būtu līdzīga oriģinālajai.

Tā, piemēram, emulējot HP LaserJet, daudzi printeru ražotāji var pārdot savu produkciju, nesatraucoties, ka programmu ražotāji nemaz neparedz šādu printeri. Programmai pasakot, ka tas ir HP LaserJet printeris, mums nebūs nekādu problēmu.

Precīza emulācija pieļauj iekārtu nomaiņu pa daļām. Mēs izveidojam procesoru, kas pilda visas iepriekšējā procesora komandas. Līdz ar to visas vecajam procesoram rakstītās komandas izpilda arī jaunais procesors. Ar emulāciju var panākt procesu izpildi lētāk, ātrāk un citādi labāk, vai arī pavisam savādāk.

Senie kurši dedzināja nepareizas krasta ugunis, lai piemānītu un pēc tam aplaupītu apmaldījušos vikingu ceļotājus, nepavisam neinteresējoties par interfeisa sākotnējo jēgu. To pašu var teikt arī par datorvīrusiem un "hakeriem", kas vajadzīgā brīdī patēlo

kādu citu, lai pēc tam, kad uzmanība ir iemidzināta, varētu sasniegt savus mērķus.



RV

Viens no vienkāršākajiem emulācijas veidiem ir oriģinālās informācijas filtrēšana, laišana cauri un nomaiņa...

Emulāciju var izmantot arī funkciju papildināšanai. Viens no vienkāršākajiem emulācijas veidiem ir oriģinālās informācijas filtrēšana, “laišana cauri” un nomaiņa tikai specifiskos gadījumos.

Tā darīja tas pastnieks, kas “iespraucās” starp diviem šaha lielmeistariem korespondētājmačā. Viņa “āķis” slēpās apstākļi, ka viņam kā vājākajam neizšķirta gadījumā pienākas atlīdzība, bet vinnesta summa ir lielāka nekā zaudējuma summa. Tā abi lielmeistari spēlēja viens ar otru, domādami, ka spēlē ar nevienam nepazīstamo pastnieku, un abi maksāja viņam.

Arī klaviatūras valodas draiveri “iespraucas” starp oriģinālo klaviatūras draiveri un tastatūru. Šī programma,

modificējot burtu plūsmu starp tastatūru un pielietojuma programmu, spēj padot no tastatūras neparedzētus simbolus. Tā mēs varam nodrošināt latviešu, igauņu, krievu un citu alfabētu burtus.

Latviešu valodas gadījumā draiveris "laiž cauri" visus burtus, izņemot kādu konkrētu tā saucamo "mirušo burtu" (parasti apostrofu vai tildi), ko dators "apēd" un atceras. Ja nākošajam nospiešajam burtam ir iespējama diakritiskā zīme, tad tas tiek nomainīts pret atbilstošo latviešu valodas burtu ar diakritisko zīmi, bet "apēdiena fakts" tiek aizmirsts. Tādējādi, piesitot apostrofu pirms latviešu specifiskajām rakstu zīmēm, varam rakstīt latviski.

Galējs emulācijas variants būtu perfekta "virtuālā realitāte". Dažādas virtuālās realitātes iekārtas cenšas emulēt mūsu maņām virtuālās telpas signālus, radot tās realitātes ilūziju.

Mazo, zaļo cilvēciņu izlūckuģis tuvojās planētai Zeme. Tā uzdevums bija veikt eksperimentu par Zemes iespējamo pārņemšanu, emulējot iezemiešus. - Par eksperimenta objektu izvēlējās Prātnieku. Viņu nolaupīja pa ceļam uz darbu, sadalīja atomos un viņa informatīvo nospiedumu ievietoja bibliotēkā. Uz Zemi Prātnieka vietā nosūtīja speciāli apmācītu mazo, zaļo cilvēciņu, kura uzdevums bija emulēt Prātnieku, lai neviens nepamanītu maiņu.

Prātnieka vāvuļošanu noimitēt nemaz nebija tik grūti, jo, pat ja viņš pa

retam bija arī ko prātīgu teicis, tad ne tiem, kas to saprot. Tāpēc viņa runas plūdus mierīgi varēja aizstāt ar pārgudru vārdu rasolu. Ēšana, dzeršana un citas fizioloģiskas darbības patiesībā ir tikai mehāniskas kustības, dažas no tām pat ļoti ritmiskas. Nelaime bija tā, ka Prātnieks publikas klātbūtnē ļoti daudz dzēra alu. Šo procesu nevarēja aizstāt ar masu hipnozi, un tāpēc bija vien jānoslogo aknu ekvivalents mazajā, zaļajā cilvēciņā. Visjaukākā Prātnieka nodarbošanās bija makšķerēšana. Galvenais, lai tik starp viļņiem būtu redzams makšķerkāts. Tad Prātnieka sieva jutās, ja ne gluži apmierināta, tad vismaz mierīga, bet mazais cilvēciņš varēja doties nosūtīt emulācijas darba atskaiti.

Eksperiments izdevās. Prātnieka sievai vīra jaunā versija tīri labi patika. Viņa pamanīja dažas jaukas pārmaiņas, kaut arī nenojauta, ka laulis nomainīts. Eksperiments gan nesniedz atbildi uz jautājumu, ko darīt, kad visi cilvēki tiks nomainīti pret mazajiem, zaļajiem cilvēciņiem. Vai turpināt emulāciju? Ja turpina, tad kāda tam visam vairs jēga? Katrai emulācijai vajag savu "āķi"!

Cash

*Iekārtu, kas emulē
lēnu iekārtu kādai
ātrākai iekārtai, lai
paātrinātu to kopīgo
darbu, sauc par
"cash".*



Nepareiza minējuma gadījumā pircējam vienalga ir jāgaida, kad saražos viņa kārto mantu.

Datoros tiek izmantoti dažādi *cash*. Diska *cash* ir programma, kas izmanto atmiņu, ielasa tajā diska saturu, lai nepieciešamības gadījumā varētu to ātrāk padot procesoram vai atmiņai. Procesora iekšējais *cash* ir ļoti ātra atmiņa, kas iebūvēta mikroshēmā un paātrina datu apmaiņu procesors - atmiņa. Lai *cash* varētu veiksmīgi strādāt, tam jāspēj pareizi uzminēt, ko ātrākā iekārta gribēs no lēnākās, un jāmaksā savlaicīgi nolasīt šo informāciju, lai nepieciešamības gadījumā spētu to ātri padot ātrākajai iekārtai.

Tirgotājs ir tāds pats *cash*, kas darbojas starp pircēju un ražotāju. Ražotājam ir grūti paredzēt, ko pirks, cik lielā apjomā pirks, jo tas nevar atļauties saražot par daudz nevajadzīgas preces. Tirgotājs atrodas tuvāk saviem pircējiem, tāpēc viņš var prognozēt preces noietu un iepirkt mantu noliktavā, lai ātri apmierinātu pircējus. Pareiza minējuma gadījumā visiem sanāk labs business. Nepareiza minējuma gadījumā pircējam vienalga ir jāgaida, kad saražos viņa kāroto mantu. Ja tirgotājs par daudz iepircis nepieprasītu mantu, tad viņam gaidāmi lieli zaudējumi. Datora gadījumā tā nav problēma, jo ar *cash* mēs kopējam informāciju, kas maksā tikai "sveces tiesu" — strāvas patēriņu datora darbam.

Philops 2005, taisot dārzeņu salātus, piecas reizes skrēja uz pagrabu. Pirmo reizi pēc burkāniem, pēc tam pēc kāpostiem, kartupeļiem, sīpoliem un bietēm, precīzi ievērojot pavārgrāmatas uzskaitījuma secību.

Philopam 2010 jau bija pastāstīts par *cash*. Ejot pēc burkāniem, viņš pa ceļam uznesa augšā visu, kas atradās blakus burkāniem, cik nu daudz viņš spēja panest. Izrādījās, ka pēc sīpoliem gan bija jāskrien vēlreiz, bet no uznestā lielākā daļa bija lieka, bet, ja dārzeņi ir virtuāli, tad nav ne vainas.

Procesora un atmiņas darbības pamatprincipi

Pirmais tuvinājums

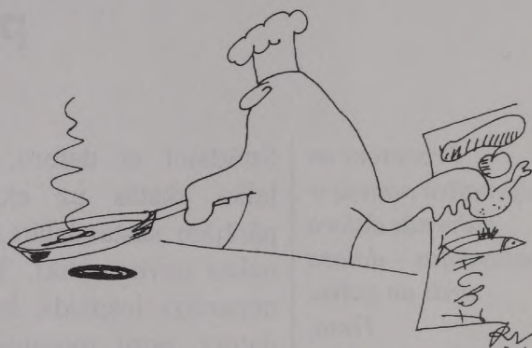
*Procesors un
operatīvā atmiņa ir
galvenās datora
sastāvdaļas – datora
sirds un galva.
Tiesa,
nav redzēts, ka kāds
cilvēks būtu izticis
bez vēdera.*

Strādājot ar datoru, cilvēks lielāko daļu laika skatās uz ekrānu, aizmirstot par pārējām sastāvdaļām, bez kurām patiesībā nekas nevar notikt. Tāpēc daudziem rodas nepareizs iespaids, ka viss darbs, ko veic dators, norit monitorā. Tā, skaišoties par neveiksmēm vai mudinot datoru strādāt ātrāk, nepacietīgie un nezinošie lietotāji sit pa monitoru, kas ir pavisam bez vainas, jo patiesībā būtu jāklapē procesora bloks (vispār mēs iesakām nepielietot fizisku spēku attiecībā ar datoru, tas var izraisīt neplānotus izdevumus).

Visa manipulācija ar datiem notiek procesorā. Procesors spēj strādāt tikai ar tiem datiem, kas atrodas atmiņā. Pārējās iekārtas (video, diski, printeri, tastatūra, pele) saņem datus no operatīvās atmiņas un nosūta datus uz operatīvo atmiņu, protams, procesora virsvadībā.

Tātad visus aprēķinus, izmaiņas veic procesors, operējot ar datiem, kas atrodas atmiņā. Vēl trakāk, procesors nolasa no atmiņas, kas tam būs jādara. Pēc tam izpilda

pasūtīto darbību un nolasa nākamo darbību no atmiņas.



Pašmodificējošās programmas mēdz lietot, lai noslēptu, ko programma īsti dara. Programmas koda sākuma daļa atkodē programmas tālāko kodu.

Neizpildot atkodēšanu, nav redzams, ko kods patiesībā dara. Varbūt formatizē disku - ļoti izplatīta vīrusu nodarbošanās.

Vēl trakāk, procesors nolasa no atmiņas, kas tam būs jādara. Pēc tam izpilda pasūtīto darbību un nolasa nākamo darbību no atmiņas.

Principā procesors savā kārtējā gājienā var izmainīt, kas tam būs jādara turpmāk. Programmas, kas realizē šādu procesora uzvedību, sauc par pašmodificējošajām programmām. Šādu programmu rakstīšana ir ļoti sarežģīta, grūti saprotama, un kļūdu “ķeršana” ir apgrūtināta. Reāli tā neviens nopietns programmu izstrādātājs nedara. Līdz ar to programmām parasti ir strikti nodalīts programmas (koda) apgabals un datu apgabals operatīvajā atmiņā.

Šeit būs maza atvairošanās. Mēs kā matemātiķi programētāji nevienam negrasāties izskaidrot, kā silīcija un vēl visa kā cita kristāls (sakausētas smiltis) spēj veikt saskaitīšanu vai reizināšanu. Ar šiem jautājumiem nodarbojas fizika un elektronika. Pirmā ir atklājusi

pusvadītāju īpašības, otrā — spējusi izveidot milzīgas, ļoti sarežģītas integrālās shēmas, kas realizē procesoru un atmiņu. Mums procesors un atmiņa ir kā abstrakti matemātiski objekti, kas manipulē ar 1 un 0. Stāsts “No tranzistora līdz procesoram” vairāk iederas grāmatā ar ievirzi elektronikā. Mēs centīsimies dot ieskatu tam, kā ar šīm samērā primitīvām 1 un 0 darbībām sanāk tās brīnumainās darbības un iespējas, ko dod šodienas dators.

Otra piezīme ir par lietas filozofisko pusi. Stingri ņemot, procesors katrā savā solī izpilda nākošo darbību no koda segmenta. Tam nav nekāda sakara ar to, ka tiek “izpildīta programma” vai ka “programma vada” procesoru. Programma ir precīzs un nekustīgs kods, kas vispār nekā nedara. Ja kāds ko dara, tad tas ir procesors, kas precīzi determinēti izpilda kodu. (Pat ne kodu, bet gan nākošo komandu no ‘atmiņas’, un tā ikreiz). Vienīgi jāņem vērā vēl kādi nedeterminēti dati, piemēram, cilvēka operatora komandas. Šeit var sākt debatēt par “brīvo gribu”, determinismu utt. Mēs to visu atstāsim citai reizei un teiksim, ka “programma strādā, aprēķina” un tā tālāk, kad vien gribēsies. Mēs galu galā dzīvojam, un mums nav daļas, vai dzīvei ir jēga, vai nav.

Otrais tuvinājums

*Procesoru varam
iedomāties kā robotu
ar aci – IP vai
koncentrētu
uzmanību (locus
concern) uz vienu
garas lentas –
atmiņas - šūnu.
Atkarībā no tā, ko
tas izlasa, tas izpilda
kādas darbības uz*

Procesors sastāv no daudziem reģistriem. Datu reģistros var izvietoties vieninieki un nulles divu, četru, astoņu baitu garumā. Atsevišķi apskatāms ir izpildāmās komandu norādes reģistrs (*IP instruction pointer*), kas pēc katras komandas izpildīšanas palielinās par šīs komandas izmēru. Tādējādi katrā komandas izpildes ciklā tiek izpildīta komanda, uz kuru norāda IP, un pēc tam tas norāda uz nākošo

*lentas un pārceļ aci
uz nākošo šūnu.*

*Brīžiem acs pārlec
uz priekšu vai
atpakaļ, brīžiem veic
cikliskas kustības.*

komandu. Ir komandas, kas izmaina IP saturu. Piemēram, salīdzināšanas komanda vienā gadījumā izmainīs IP vērtību, tādējādi radot programmai zarošanās un ciklošanās iespējas.

Tipiskākās procesora komandas ir:

- datu apmaiņas komandas starp atmiņu un reģistriem;
- aritmētiskās vai loģiskās darbības starp reģistriem, vai reģistriem un atmiņu, un rezultāta ierakstīšanu reģistrā vai atmiņā;
- reģistru un atmiņas salīdzināšana ar attiecīgu IP modificēšanu, kā rezultātā var tikt realizēta programmas zarošanās.

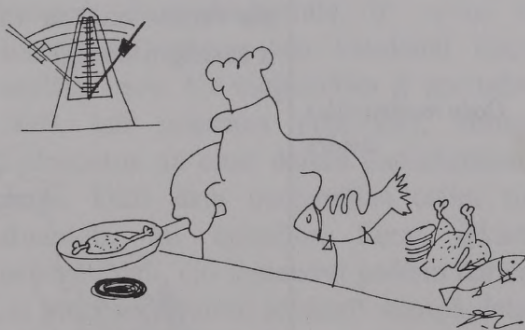
Konkrēti komandu formāti un komandas procesoram vai to saimei nosaka procesora arhitektūru. Vienas arhitektūras procesori var izpildīt vienu un to pašu kodu, neraugoties uz to, ka procesori var būt dažādi realizēti. Firms Intel procesori 8086, 80286, 80386, 80486, Pentium, Pentium Pro un to modifikācijas ir vienas ģimenes procesori. Visiem jaunākiem procesoriem ir tās pašas komandas, kas vecākiem procesoriem. Līdz ar to visas vecās programmas var izpildīt uz jaunāka procesora, cits jautājums ir, vai to vajag darīt.

Programmas, kas rakstītas citai arhitektūrai

(piemēram, PowerPc) nav iespējams izpildīt “pa taisno” uz Intel procesora. Protams, ir iespējami tulkošanas vai emulācijas varianti. Ar konkrētu procesora arhitektūru var iepazīties, mācoties zemākā līmeņa valodu šim procesoram – Assembleru.

Procesora parametri

Procesora ātrums



Katra procesora komanda tiek izpildīta vairākās taktīs.

Katrā procesorā ir iebūvēts kvarca pulkstenis, kas sinhronizē procesora darbu. Šī pulksteņa darbības (tikšķēšanas) ātrums nosaka procesora takts ātrumu. Katra procesora komanda tiek izpildīta vairākās taktīs. Dalīšanai vajadzīgas daudzreiz vairāk taktis nekā saskaitīšanai. Takts ir kā komandas svilpe, pēc kuras darbojas visas datora iekārtas. Dati ir kā paciņas, ko viena iekārta nodot otrai pa ķēdi, ja nebūs sinhronizācijas, paciņas kritīs zemē.

Pulksteņa ātrums tiek mērīts megahercos, 1 MHz ir miljons tikšņu sekundē. Jo ātrāk tikšņ procesora pulkstenis, jo ātrāk tiek izpildīta programma.

*16, 32, 64 bitu
procesors*

Procesora reģistra izmērs bitos nosaka, ar cik lieliem skaitļiem tas spēj operēt vienā ciklā. Gadījumā, ja jādarbojas ar lielākiem skaitļiem, operācijas ar tiem jāsadala tā, lai tās varētu veikt ar vairākām darbībām. Tas ir sarežģīti un ilgi.

*Datu maģistrāles
izmērs*



Vienā apmaiņas ciklā mēs varam pārsūtīt tik lielu datu daudzumu, kāds ir maģistrāles izmērs.

Datu apmaiņa starp procesoru un atmiņu notiek ar datu maģistrāles palīdzību. Datu maģistrāle atrodas uz pamatplates, un tai arī ir noteikts izmērs bitos. Vienā apmaiņas ciklā mēs varam pārsūtīt tik lielu datu daudzumu, kāds ir maģistrāles izmērs. Ja nepieciešams nosūtīt vairāk, datu pārsūtīšana jāveic ar vairākiem

piegājieniem. Skaidrs, ka arī pie vienādiem takts ātrumiem, pildot reālus uzdevumus, procesors, kas ir spējīgs apstrādāt lielākus datus vienlaicīgi, var strādāt daudzreiz ātrāk nekā tāds, kam datu maģistrāles izmērs ir mazāks.

Ar datu maģistrāles palīdzību datora sastāvdaļas savstarpēji apmainās ar datiem. Principā datu maģistrāle ir celiņi uz elektroniskās plates vai vienkārši daudz paralēlu vadu. Uz maģistrāles ir spraudnes (*Slot*), kur iespraust procesoru, atmiņu, kontrolierus un citas datora paplašināšanas kartes. Daži datu maģistrāles celiņi tiek izmantoti, lai norādītu, kurai iekārtai paredzēti dati, citi izmantoti pašiem datiem. Lai procesors varētu ierakstīt atmiņā datus, daži celiņi jāizmanto arī datu adreses norādīšanai.

Datu maģistrāle savā ziņā ir kā ziņojumu dēlis firmā, kas vienlaicīgi satur tikai vienu ziņojumu. Uz ziņojuma ir uzrakstīts tā adresāts. Visi darbinieki kontrolē ziņojumu dēli, vai tur nav tiem paredzētā ziņojuma. Nepieciešamības gadījumā darbinieks pats izliek ziņojumu citam darbiniekam. Ja šo ziņojumu dēli iedomāsimies kā skrejlošu uzrakstu virs VEF rūpnīcas, tad analogija būs vēl precīzāka. Drīzāk gan būtu jāiedomājas kādi 64 skrejloši ziņojumu dēļi, bet tikai ar vienu burtu katrs, un izmantoto alfabētu no diviem burtiem, tad tā jau ir datu maģistrāle.

Firmas Intel procesoriem ir šādi reģistru un datu maģistrāles izmēri:

8088	16 biti	8 biti
80286	16 biti	16 biti
80386SX	16 biti	16 biti
80386DX	32 biti	32 biti
80486SX	32 biti	32 biti
80486DX	32 biti	32 biti
Pentium	64 biti	64 biti
Pentium Pro	64 biti	64 biti

Koprocesors

Lai darbotos ar ļoti lieliem skaitļiem, noderīgs ir peldošā punkta procesors jeb koprocesors. Peldošā punkta procesoram ir savi reģistri un komandas. Peldošā punkta procesors darbojas ar jebkura saprātīga izmēra skaitļiem, tos noapaļojot līdz zīmju skaitam, ko tas spēj apstrādāt. Tādējādi nav jāmokās, speciāli programmējot darbības ar ļoti lieliem vai maziem skaitļiem, turklāt operācijas ar tiem notiek ļoti ātri. Var iedomāties, ka koprocesors darbojas ar skaitļiem zinātniskajā pierakstā, kur skaitlis pierakstīts kā decimāldaļas un desmitnieka pakāpes reizinājums. 286, 386 un 486SX procesoriem blakus varēja "piespraust" otru procesoru ar peldošā punkta operācijām. Visiem jaunākajiem procesoriem koprocesors ir jau iebūvēts.

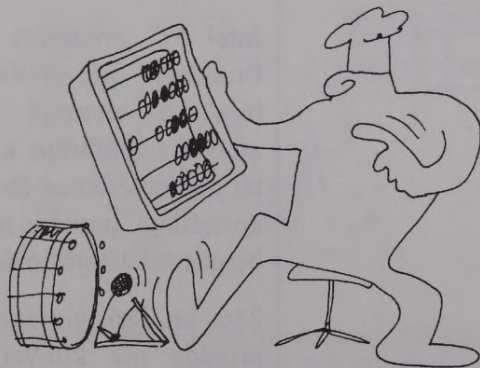
Takts ātrumu dubultošana, četrkāršošana, ...

Ir tādi procesori kā 486DX4/100. Četrinieks nozīmē, ka procesora takts ātrums ir četras reizes lielāks nekā datu maģistrālei. Tātad

šis procesors darbojas uz 25 MHz datu maģistrāles, ar šādu ātrumu apmainoties ar datiem. Bet iekšienē tas darbojas ar četrkāršu ātrumu, sasniedzot 100 MHz ātrumu.

Cash Lai lēnākais ātrums datu apmaiņai ar atmiņu nepalēninātu darbu, procesorā iekšā izveido datu buferi - *cash*, kurā ielasa lielākus datu apjomus nekā procesors pieprasa, mēģinot uzminēt nākošos pieprasījumus, tādējādi samazinot procesora atmiņas pieprasījumus.

Procesoru paaudzes



P.V.

...procesoru paaudzes atšķiras ar takts ātrumu.

Tātad procesoru paaudzes atšķiras ar takts ātrumu, ar datu izmēru, ko spēj apstrādāt un apmainīt ar atmiņu vienlaicīgi, ar kopprocesoru. Ar ko vēl?

Izrādās, ka saskaitīšanas komandu, ātrāko aritmētisko operāciju procesorā, katra nākošā procesora paaudze veic, izmantojot mazāku procesora taktu skaitu:

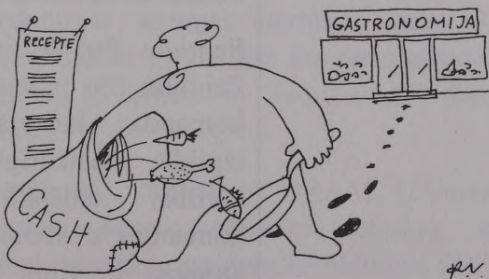
- 386. procesors divu skaitļu saskaitīšanai izmantoja minimums 6 taktis;
- 486. procesors to izdarīja jau divās taktīs;
- Pentium spēj saskaitīt divus skaitļus vienā taktī;
- Pentium Pro vienā taktī spēj saskaitīt trīs skaitļus.

Intel PC procesori sākās ar 86. ģimeni. Procesorā bija 30000 tranzistori. Procesors bija kā vientuļš strādnieks, kurš pats atkodēja konkrēto komandu, pats sagādāja tai nepieciešamos datus no atmiņas, pats tos apstrādāja, pats tos nogādāja pēc adreses un beigās sakārtojās nākamajam gājienam.

286. un 386. ģimene jau bija kā strādnieku brigāde pie konveijera. Viena procesora daļa atkodēja komandu, pēc tam cita procesora daļa sagādāja nepieciešamos datus, kamēr pirmā daļa jau varēja atkodēt nākošo komandu. 286 procesorā bija 100000 tranzistori, bet 386 jau 300000 tranzistori. 386 procesorā tika ieviestas dažas jaunas komandas, kas deva iespēju rakstīt ietilpīgākas programmas.

Jebkurā procesā ir "šaurā vieta". Procesora un atmiņas sadarbībā šaurā vieta ir atmiņa. Tās ātrums 70ns (nonasekundes) jeb 12Mhz ir ierobežojošs faktors, ja to nekādi nevar pārvarēt, tad 386. procesors ir pilnība.

486. ģimene, kurai jau bija ap miljons tranzistoru, sevī iekļāva iebūvētu kopprocesoru un līdz 8000 baitiem iekšējā *cash*. Šinī gadījumā *cash* ir ļoti ātra atmiņa, kurā tiek dublēts parastās atmiņas saturs.



Pateicoties kešatmiņai, procesoram tikai 10% gadījumā jāgriežas pie īstās operatīvās atmiņas.

Tā kā procesors bieži vien griežas pie blakus reģioniem atmiņā, tad kešatmiņā iespējams ar viltīgu algoritmu palīdzību turēt nepieciešamos datus. Pateicoties kešatmiņai, procesoram tikai 10% gadījumā jāgriežas pie īstās operatīvās atmiņas (DRAM). Tādējādi šaurā vieta tiek veikli pievarēta.

Pentium procesors ir nākošā procesoru paaudze ar 3 miljoniem tranzistoru. Pentium procesors patiesībā ir divi 486DX procesori, kas paralēli izpilda kodu. Ir parādījusies otra konveijera līnija, kas strādā sinhroni ar pirmo. Tiek apstrādātas divas, viena aiz otras sekojošas komandas vienlaicīgi. Protams, tas ir iespējams tikai tad, ja otrās komandas izpilde nav atkarīga no pirmās, kas izrādās nemaz nav tik bieži. Tādējādi daudzos gadījumos procesors izpilda divas komandas vienlaicīgi.

Pentium Pro šo ideju paceļ vēl augstāk. Pentium Pro procesors jau rūpīgāk analizē komandas, kas jāizpilda, un ir spējīgs izpildīt neatkarīgās operācijas jebkurā kārtībā, tādējādi neaizkavējot citas komandas ar komandu, kas gaida datus no atmiņas. Pentium Pro bez iekšējās (16Kb vai 32Kb) atmiņas ir arī ārējā (no 256Kb līdz 512Kb lielā) *cash* atmiņa.

MMX

Tikko parādījušies procesori, kuros iebūvētas papildu komandas skaņas un video ātrākai apstrādei. Šie procesori pašlaik ir nedaudz dārgāki par pārējiem, bet domājams, ka nākotnē visi Intel procesori būs ar MMX komandām.

Pentium II

Pentium II ir pēdējais Intel procesors, kurš, vienkāršoti izsakoties, ir Pentium Pro ar

iebūvētām MMX komandām.

Atmiņas parametri

Megahercs ir miljons procesora tikšņu sekundē. Bet cik ilgs laiks pāiet starp tikšņiem? 1 nanosekunde ir 1^{-9} sekundes. Tātad 25MHz atbilst 40nsec, 100MHz atbilst 10nsec.

Datora atmiņas mikroshēma saņem adresi pa vienām mikroshēmas kājiņām un pēc brīža atdod nolasītos datus no šīs adreses pa citām mikroshēmas kājiņām. Vienkāršākā gadījumā mikroshēma glabā 1 bitu katrā adresē, bet ir arī mikroshēmas, kas atdod 2, 4, vai 8 bitus uz atsevišķām kājiņām.

Brīdis starp atmiņas padošanu un datu atdošanu ir atmiņas darbības laiks, kas šodien tipiski ir 70 līdz 80 nanosekundes. Atmiņas ātrdarbība ir “šaurā vieta” datora darbībā.

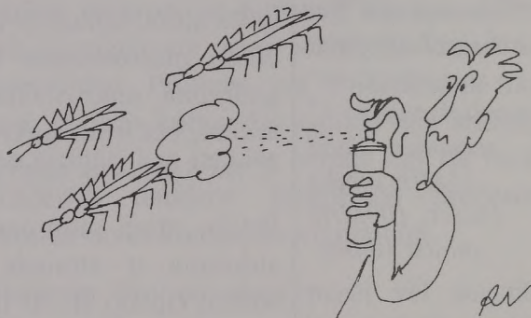
Parastā PC atmiņa ir DRAM (*Dynamic Random Access Memory*) atmiņa, kas parasti ir veidota no mazām platītēm SIMM (*Single Inline Memory Modules*), uz kurām ir izvietotas atmiņas mikroshēmas. Lai varētu izveidot SIMM, kas operē ar baitu vienlaicīgi, no atmiņas mikroshēmām, kuras operē ar vienu bitu, nepieciešams izvietot astoņas šādas mikroshēmas uz SIMM.

Pirmie plaši lietotie SIMM bija ar 30 kājiņām, šie SIMM spēja padot 1 baitu (8 kājiņas) uz katru atmiņas pieprasījumu, kas tiek veikts ar divu baitu adresāciju (16 kājiņas). Pārējās sešas kājiņas tiek

286 un 386 un arī
daži 486 PC izmanto
30 kāju SIMM. Šāda
atmiņa šodien var
būt grūti
nopērkama, jo visi
jaunie datori
izmanto 72 kājiņu
SIMM.

Kādēļ datora atmiņa
iespējama tikai
noteiktos
"mistiskos"
izmēros?

izmantotas vadībai un kontrolbitam. Ar kontrolbita palīdzību varam konstatēt atmiņas moduļa kļūdu, izmantojot paritātes pārbaudi.



Pirmie plaši lietotie SIMM bija ar 30 kājiņām...

286. un 386SX procesors pieprasīja atmiņu pa diviem baitiem, un SIMM vajadzēja likt pa pāriem. 486. procesors jau pieprasīja atmiņu pa četriem baitiem, četrus SIMM likt paralēli jau bija neērti.

Šodien visas jaunās pamatplates ir paredzētas darbam ar 72 kājiņu SIMM. Šī atmiņa spēj nodot 4 baitus katrā atmiņas pieprasījumā. Tā kā Pentium strādā ar 8 baitiem vienlaicīgi, tad Pentium sistēmās 72 kājiņu SIMM vienalga jāliek pa pāriem. Šodienas SIMM ir ar 4, 8, 16 un pat 32, un 64 megabaitu ietilpību.

Uz datora pamatplates atmiņa tiek ievietota speciālās spraudnēs, kas Pentium gadījumā būs izvietotas pa pāriem. Katrā pāri ir jāsprauž pilnīgi vienādi SIMM.

Pamatplatei var būt dažāds skaits atmiņas spraudņu. Ja mums trūkst vietas kur pievienot jaunus SIMM, lai palielinātu datora atmiņu, var gadīties, ka pēc ietilpības mazākie SIMM ir jānomaina pret lielākas ietilpības SIMM. Šajā gadījumā vecie SIMM visticamāk ir jāmet ārā, jo tādi nevienam nav vajadzīgi.

Līdz ar to datora atmiņa tiek veidota no SIMM, kuru ietilpība ir 4, 8, 16 ... megabaiti, kas apvienoti pāros. Ar šādiem noteikumiem mēs noteikti izveidosim datora atmiņu, kuras izmērs megabaitos dalās ar astoņi.

Videoatmiņa

Attēla veidošana ir samērā sarežģīts un resursus noslogojošs process, tādēļ visiem datoriem ir atsevišķa videoatmiņa, kas tiek izmantota attēla veidošanai.

Uz videokontroliera ir atmiņa, kas tiek izmantota attēla veidošanai. Monitors ģenerē krāsu, pamatojoties uz trīs ievadiem - spriegumiem jeb krāsu kanāliem, kas apzīmē trīs pamatkrāsu: sarkanās, zaļās un zilās (angļu saīsinājums RGB) daudzumu krāsā. Katru krāsu kanālu var ieslēgt 256 intensitātes līmeņos, kas teorētiski dod 256^3 jeb apmēram 16 miljonu krāsu. Šādu krāsu

režīmu sauc par *Truecolor*, šajā režīmā katram krāsu punktam uz ekrāna - pikselim - nepieciešami trīs baiti videoatmiņas, lai nokodētu krāsu uz ekrāna. Videokontrolieris ir iekārta, kas atkarībā no darbības režīma un savas atmiņas satura, kontrolē monitora krāsu kanālus. Tātad ar videokontrolieri, mainot videoatmiņas saturu, varam mainīt attēlu uz ekrāna.

Datora ekrāna režīmu raksturo iespējamais krāsu daudzums katram punktam jeb pikselim un punktu daudzums uz ekrāna jeb ekrāna izšķiršanas līmenis (*video resolution*). Šodien populārākais ekrāna izšķiršanas līmenis ir 800x600 pikseļi, kas ir SVGA (*Super Video Graphics Array*) standarts, kaut arī sastopami datori ar izšķiršanas spēju 640x480 pikseļi, kas ir VGA (*Video Graphics Array*) standarts, un lielākām izšķiršanas spējām - 1024x768 pikseļi vai 1280x1024 pikseļi.

Ko dod lielāka videoatmiņa? Kā aprēķināt tās nepieciešamo izmēru?

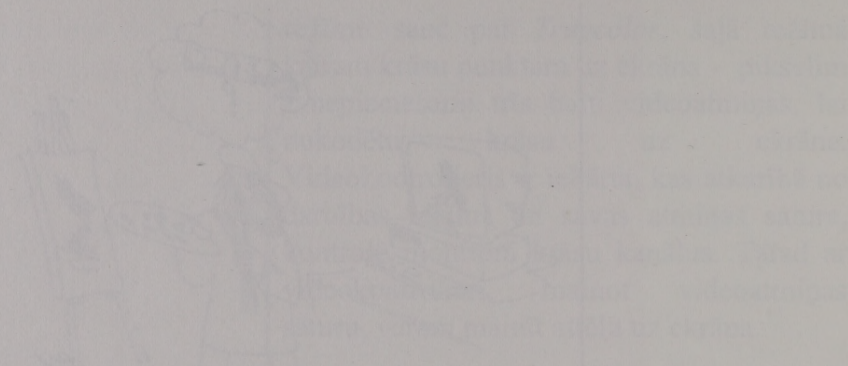
Atkarībā no iespējamā krāsu daudzuma katram pikselim jāatvēr videoatmiņas daudzums krāsas kodēšanai. Ar 4 bitiem var nokodēt 16 krāsas, ar 8 bitiem var nokodēt 256 krāsas, ar 16 bitiem var nokodēt 65,536 krāsas, ar 24 bitiem var nokodēt 16,777,216 krāsas.



Vairāk atmiņas nozīmē vairāk krāsu un punktu.

Zinot vēlamo punktu skaitu uz ekrānu un vēlamo krāsu daudzumu punktam, varam aprēķināt nepieciešamās videoatmiņas izmēru. Tā, piemēram, VGA kontroleris ar 256 Kb atmiņu var attēlot 640×480 pikseļus 16 krāsās. SVGA kontroleris ar 1 Mb videoatmiņas var uzturēt 800×600 pikseļus 65,536 krāsās, vai 1024×768 pikseļus 256 krāsās.

Vairāk atmiņas nozīmē vairāk krāsu un punktu.



[Illegible text block]

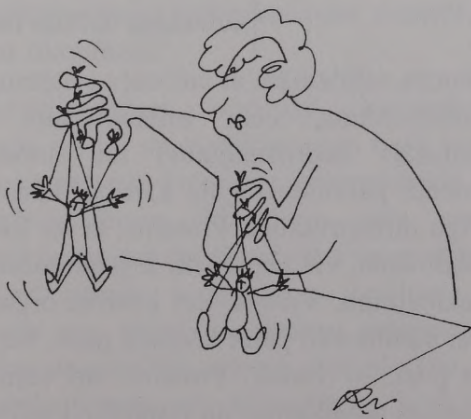
[Illegible text block]

Operētājsistēmas

Dators ar programmām ir kā slāņaina kūka. Katrs nākošais slānis nodrošina augstāka līmeņa interfeisu, kontekstu, vidi augšējā līmeņa, darbībai ar zemāko līmeni.

Operētājsistēma ir starpnieks starp datora "dzelžiem" un parastām programmām.

Datora procesors, atmiņa un datu nesēji veic samērā vienkāršas pamatdarbības. Lai no tām izveidotu ko sakarīgu, mums nepieciešams veikt daudz dažādu palīgdarbu. Ja katra programma veiks visus nepieciešamos palīgdarbus pati, tad uz cietā diska būs sajukums, un mēs reāli vienā laika momentā varēsim lietot tikai vienu programmu. Arī programmēt visus palīgdarbus ir grūti un sarežģīti. Tamdēļ jau no pašiem datoru pirmsākumiem tika radītas operētājsistēmas.



Operētājsistēma ir programma, precīzāk, programmu komplekss, kas darbojas ar programmām.

Operētājsistēma ir programma, precīzāk, programmu komplekss, kas darbojas ar programmām. Operētājsistēma apvieno sevī funkcijas, ko lieto vairākas programmas. Operētājsistēma nodrošina pielietojamo programmu darbu, izdalot tām nepieciešamos atmiņas resursus, ļaujot izmantot datora iekārtas, piemēram, cieto disku, printeri. Operētājsistēma nodrošina darbošanos ar failu sistēmu un daudz ko citu.

Attīstības gaitā operētājsistēmas ietver sevī arvien vairāk un vairāk funkciju. Rezultātā pielietojamo programmu izstrāde kļūst vienkāršāka. Tās ir arī unificētākas, jo izmanto standarta operāciju sistēmas līdzekļus dažādu mērķu sasniegšanai.

Ja datoru salīdzinām ar autoceļu sistēmu, tad autoceļu pārvalde ir operētājsistēma, ceļu infrastruktūra būtu datora "dzelzi" (tehniskais nodrošinājums) un automašīnas - programmas. Autoceļu pārvalde regulē kārtību, kādā jābrauc, un rūpējas par kopīgo infrastruktūru. Protams, ar to, ka pārvalde pastāv un visu to nodrošina, vēl nepietiek. Ir jābūt atbilstībai starp noteikumiem un ekipējumu. Valstīs, kur kustība organizēta pa ceļa labo pusi, stūrei automobīlī jābūt kreisajā pusē, bet tur, kur kustība norit pa labo pusi, — otrādi. Protams, arī ceļu stāvoklis valstī nosaka satiksmes noteikumus un vispārējo kustības organizācijas stilu.

Operētājsistēmas uzdevumi

*Jebkura
infrastruktūra pati
par sevi ir
bezzēdzīga, taču tai
var būt izšķiroša
nozīme
pielietojumos.
Bieži kopīga
infrastruktūra kļūst
par vienojošo
faktoru tās
lietotājiem, jo
uzspiež vienotu
darbības stilu.*



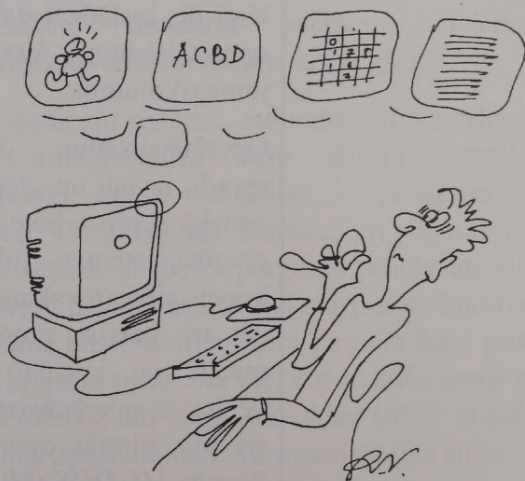
Operētājsistēmas organizē informācijas ievadi, izvadi un glabāšanu gan tiešā veidā, gan kā servisu lietojuma programmām.

Kad jūs ieslēdzat datoru, pirmā sāk strādāt operētājsistēma, kas gatavojas saņemt no jums rīkojumus.

Operētājsistēmas organizē informācijas ievadi, izvadi un glabāšanu gan tiešā veidā, gan kā servisu lietojuma programmām. Ar operētājsistēmas līdzekļiem mēs varam kopēt failus, tai skaitā arī citas programmas, kā arī palaist programmas izpildei. Pie ievada un izvada iekārtām programmas vēršas ar operētājsistēmas starpniecību. Šīs arī bija pirmās operētājsistēmas funkcijas. Pirmās MS-DOS (*Microsoft Disk Operating System*) operētājsistēmas neko īpaši vairāk sevī neietvēra. Katra nākošā versija strādāja

ātrāk, ļāva izmantot vairāk atmiņas utt., bet principā nodrošināja tikai šīs pamatfunkcijas.

Kvalitatīvs lēciens notika, parādoties *Microsoft® Windows™*. *Microsoft Windows* sevī ietvēra grafiskā interfeisa sistēmu. Sistēmas darbs tika organizēts ar grafisku logu palīdzību. Logos varēja attēlot ne vien tekstuālu, bet arī grafisku informāciju. Pati sistēma kļuva vieglāk saprotama un lietojama. Arī programmu ražotāji, veidojot pielietojamās programmas, izmantoja šo pašu grafisko interfeisu.



Microsoft Windows sevī ietvēra grafiskā interfeisa sistēmu.

Līdz ar to visas *Microsoft Windows* programmas ir izskatā ļoti līdzīgas, tādējādi atvieglojot gan programmu izstrādi, gan to apgūšanu.

Microsoft Windows nodrošina unificētu pieeju dažādām iekārtām. Piemēram, lietotājprogrammai ir tikai jānosūta izdrukas lapas izskats operētājsistēmai, kas tālāk to pārsūta uz konkrēto drukas iekārtu. *Microsoft Windows* un iekārtai pieinstalētie draiveri to tālāk “pārtulko” konkrētās iekārtas komandās, un mēs iegūstam rezultātu, kas atkarīgs no iekārtas kvalitātes. Mainot drukas iekārtas, mums tās tikai jāpieinstalē *Microsoft Windows* sistēmai. Agrāk katrai lietotājprogrammai bija jāzina, kāda ir drukas iekārta, un tas iekārtu maiņu gadījumā radīja problēmas. Katrai programmu ražotājfirmai bija atsevišķi jāieprogrammē visas pasaules drukas iekārtas, kas, protams, nebija iespējams, bet bija lieka laika un darba šķiešana.

Arī unificētā burtu garnitūru sistēma ir atvieglojums gan lietotājiem, gan programmu izstrādātājiem. Pateicoties *Microsoft Windows* fonu sistēmai (īpaši *TrueType* fontiem), mēs varam pievienot latviešu valodu *Microsoft Windows*, un visas mūsu programmas sāk “runāt latviski”. Agrāk bija atsevišķi jālatvisko

katra programma, katra drukas iekārta, un tas radīja liekus sarežģījumus.



R.V.

Tiesa, šāda tīkla administrēšana ir apgrūtināta, un tādēļ sarežģītu tīklu nevajadzētu veidot ar šo tehnoloģiju palīdzību.

Nākošās funkcijas, ko sevī ietvēra operētājsistēmas, bija datoru kopā saslēgšana tīklā. *Microsoft Windows for Workgroups* sevī ietver arī datoru tīkla iespējas. Mēs varam caur tīklu izmantot cita datora iekārtas. Šāds tīkls datoriem nodrošina “siāmas dvīņu” iespējas. Mēs varam lietot otra datora disku, printeri, CD iekārtu kā savu. Tiesa, šāda tīkla administrēšana ir apgrūtināta, un tādēļ sarežģītu tīklu nevajadzētu veidot ar šo tehnoloģiju palīdzību. *Microsoft Windows for Workgroups* mēs varam pieslēgt arī pie tīkla serveriem.

Parādoties tīkliem, operētājsistēmas tika sadalītas divās grupās. *Microsoft Windows for Workgroups*, *Microsoft Windows 95*, *Microsoft Windows NT Workstation* ir

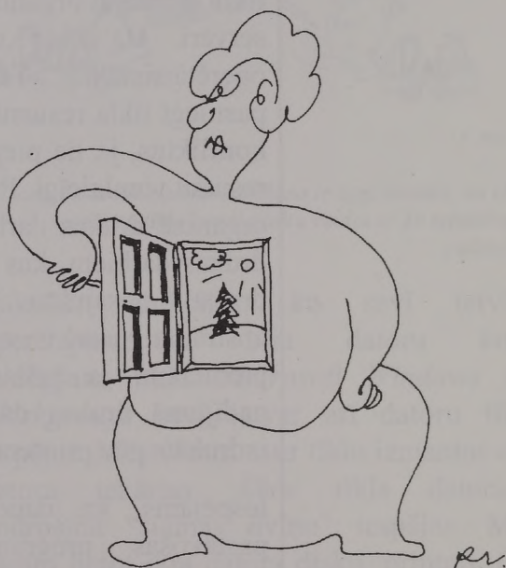
operētājsistēmas, kas nodrošina individuāla datora darbu un tā pieeju tīkla resursiem. No tīkla viedokļa tie ir klienti. Savukārt *Windows NT Server*, *Novell*, *Unix* nodrošina tīkla resursu organizēšanu — tie ir tīkla serveri. Uz tīkla servera darbojas tīkla operētājsistēma. Tās uzdevums ir ātri pasniegt tīkla resursus lietotājiem un risināt konfliktus, ja tie pieprasa vienu un to pašu resursu vienlaicīgi. Piemēram, print-serveris organizē drukas darbu rindu, lai dažādiem drukas darbiem, kas vienlaicīgi aizsūtīti uz druku, nesajuktu lappuses. Tas spēj apstrādāt arī negaidītas situācijas, piemēram, ja printeris ir izslēgts. Šajā gadījumā drukas darbs tiek uzglabāts un izdrukāts pēc printera ieslēgšanas.

Iespējams, ka daudzas šodien atsevišķi pastāvošas programmas nākotnē tiks ietvertas operētājsistēmās.

Ļoti svarīgi ir izvēlēties pareizo operētājsistēmu. Operētājsistēma nosaka programmu loku, ko varēs lietot uz datoru ar šo operētājsistēmu. Ja operētājsistēma nav populāra, tad tai rakstītu programmu nav daudz.

Šodien vispopulārākās operētājsistēmas ir dažādas *Microsoft Windows* versijas. Parastam lietotājam no pašreiz esošajām

operētājsistēmām visērtākā ir Microsoft Windows 95. Lietotājam ar paaugstinātām prasībām ieteicama ir Windows NT.

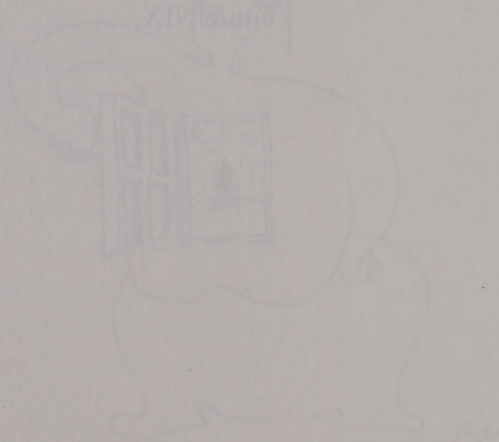


Šodien vispopulārākās operētājsistēmas ir dažādas Microsoft Windows versijas.

Kā tīkla operētājsistēma ļoti populārs kļūst *Windows NT* serveris. Tā ir viegli apkalpojama un administrējama tīkla operētājsistēma. Tradicionālā *Novell* tīkla operētājsistēma prasa profesionālāku pieeju, bet sistēmas labās īpašības tā īsti var novērtēt tikai ļoti lielos tīklos. Lielos tīklos, īpaši uz mašīnām ar RISC procesoriem, ir

populāri dažādi UNIX paveidi. UNIX datori pārsvarā veido Internet tīklu. Daudzas ar globāliem tīkliem saistītas lietas visērtāk izdarīt ar UNIX palīdzību. UNIX visvājākā vieta ir tas, ka nav vienas standarta UNIX operētājsistēmas, līdz ar to pielietojuma programmu pieejamība ir stipri ierobežota, jo viena UNIX programmas var nestrādāt ar citu UNIX.

The first part of the paper describes the development of the system. The second part describes the system architecture. The third part describes the system implementation. The fourth part describes the system evaluation. The fifth part describes the system conclusion.



The system is designed to be a general-purpose system. It is designed to be a general-purpose system. It is designed to be a general-purpose system. It is designed to be a general-purpose system. It is designed to be a general-purpose system.

Datu uzglabāšana

*Dators ir rīks
dažādu datu
ievadei, apstrādei,
uzglabāšanai un
izvadei.*

*Jebkurā datu
apstrādē izmanto arī
datu uzglabāšanu,
vismaz īslaicīgu. Tas
nozīmē, ka bez datu
uzglabāšanas
pārējās funkcijas ir
vai nu neiespējamās,
vai bezjēdzīgas.*



*Datori mums ļauj uzglabāt kolosālus informācijas
apjomus.*

To, ka datori kļuva tik populāri un izkonkurēja rakstāmmašīnas, lielā mērā noteica fakts, ka datori ļauj saglabāt paveiktā darba rezultātus, t.i., veltot daudz laika kāda dokumenta radīšanai vai programmas izstrādei, nākamreiz nav jāsāk viss no gala, bet gan var gluži vienkārši ielasīt datora atmiņā pagājušajā reizē saglabātos datus. Datori mums ļauj uzglabāt kolosālus informācijas apjomus.

Ir izdomātas daudzas datu saglabāšanas un nolasīšanas ierīces. Dažas no tām ļauj tikai nolasīt ierakstītos datus, bet neļauj rakstīt. Tāda ierīce ir, piemēram, CD nolasītājs.

Plašāk pazīstamie datu nesēji, uz kuriem glabā datus, ir datoru cietie diski, disketes un CD. Bez tiem ir vēl arī magnētiskās lentas, CD-ROM, ZIP diski u.c., kas ikdienas darbā vienkāršā birojā vēl netiek pārāk plaši izmantoti, kaut arī to pielietojuma diapazons pieaug ar katru dienu.

Visbiežāk dati tiek glabāti uz datora cietā diska. Principā parasti galda datora cietais disks glabā visu datora funkcionēšanai nepieciešamo informāciju un lietotāju saglabātos datus - operētājsistēmas ielādi, sistēmas konfigurāciju (visu parametru uzstādījumus), lietojumprogrammas, saglabātos dokumentus, izveidotās elektroniskās tabulas, attēlus, datu bāzes datus utt. Vārdu sakot, visu, ko lietotāji tajā ir ierakstījuši.

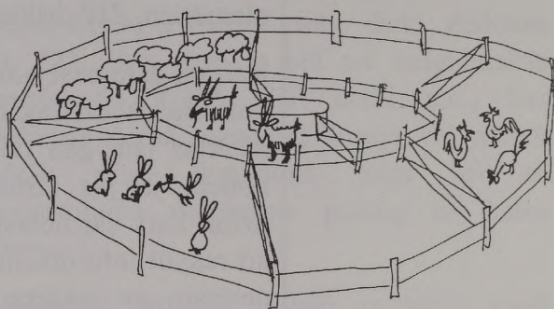
Taču mēdz gadīties tādas nepatīkamas situācijas, kad cietais disks vīrusa vai kāda lietotāja darbības rezultātā “nojūk” jeb nelasa datus. Tādēļ, lai dati neietu zudumā, ieteicams glabāt rezerves kopijas kaut kur citur. Ja vien jūsu datu apjoms nav pārāk liels, šim nolūkam labi kalpo parastās disketes. Uz tām tad arī nokopējiet svarīgāko informāciju no sava datora cietā diska. Ja failu izmēri krietni vien pārsniedz diskešu maksimāli pieļaujamo apjomu, tad

izmantojiet ietilpīgākus datu nesējus, piemēram, ZIP diskus.

Katram datu nesējam ir vajadzīga sava datu nolasīšanas un rakstīšanas iekārta. Parasti iekārtas ļauj gan rakstīt, gan lasīt. Tā tas notiek, ja jūs izmantojat disketes disku dzinī, kurš var nolasīt datus no disketes un arī rakstīt informāciju uz tās. Protams, datu nesējam un iekārtai jābūt savietojamiem. Jūs nevarēsiet nolasīt augsta blīvuma disketi ar diskiekārtu, kuras lasošā galviņa nespēj nolasīt datus no šādas disketes. Tādas diskiekārtas gan var gadīties tikai vecu modeļu datoriem, jo visi jaunākie datori ir aprīkoti ar augsta blīvuma diskešu lasītājiem.

Kā tad disku iekārtas glabā datus? Kā tās var tos nolasīt? Kāpēc nekas nesajūk? Pamēģināsim izpētīt šo jautājumu, aplūkojot vienkāršo 3.5'' (3,5 collu) disketi. Šo disketi veido cietā plastmasas apvalkā ievietots magnētisks disks. Apvalkā ir atvere, kas nodrošina diskiekārtas galviņas pieeju diskam. Šo atveri nosedz plāns metālisks aizsegs, kuru uzmanīgi pavelkot nost, varat apskatīt pašu disku.

Tajā var pamanīt sīkus koncentriskus riņķus, ko sauc par trekiem. Treki ir sadalīti gabalos, ko sauc par klāsteriem.



P.V.

Treki ir sadalīti gabalos, ko sauc par klāsteriem.

Klāsteros tad arī tiek rakstīti dati. Lai tie nesajuktu, ir izstrādāti stingri noteikumi šo datu struktūrai. Dati tiek sadalīti pa failiem. Katram failam ir savs nosaukums, bet visa failu sistēma tiek veidota pēc noteiktiem principiem, ko izvirza katra operētājsistēma. Visplašāk lietotā ir DOS operētājsistēmas failu sistēma, kurai gan ir daudzas nepilnības, piemēram, faila vārda garuma ierobežojums ar 8 simboliem faila nosaukumam un 3 simboliem tā paplašinājumam.

Dati datorā tiek ierakstīti, dzēsti, papildināti. Lai šajā haosā ieviestu kārtību, ir izveidotas samērā asprātīgas

Failu sistēmas pamatideja ir izveidot tabulu (FAT - *File Allocation Table*), ar kuras palīdzību var veidot patvaļīgas klāsteru virknes, glabājot informāciju par failu izvietojumu uz diska.

failu sistēmas.

Katram klāsterim atbilst viens ieraksts tabulā. Lai veidotos klāsteru virkne, klāsterim atbilstošajā tabulas ierakstā tiek norādīts nākošā klāstera numurs. Fails tiek izvietots vienā vai vairākos klāsteros, turklāt uz diska fiziski tie var būt izvietoti patvaļīgā secībā. Šī mehānisma rezultātā mēs varam ierakstīt un papildināt failus, nerezervējot tiem uz diska fiksēta apjoma vietu.

Pateicoties šādai struktūrai, mēs iegūstam it kā grāmatu, kurai varam pārkārtot lapas. Precīzāk gan būtu jāsaka, ka mums ir it kā viena grāmata, kuru var šķirstīt divos dažādos veidos. Pirmais ir fiziskais, kā grāmata patiesi izveidota, otrais virtuālais, kā loģiski lasīt informāciju šajā grāmatā, kur lappuses sajauktas haotiski. FAT ir tabula, kas norāda, kādā secībā patiesībā šī grāmata jālasa. Tiesa, pazaudējot FAT, mēs pazaudējam spēju apskatīt disku pēc būtības. Tamdēļ daudzu vīrusu uzbrukumu mērķis arī ir FAT.

Tehniskāks failu sistēmas izklāsts

DOS failu sistēma izvietojas tā saucamajā blokiekārtā, kas glabā informāciju vienāda izmēra sektoros. Visizplatītākais viena šāda sektora izmērs ir 512 baiti, kas arī ir parastās standarta disketes sektora izmērs. Bet citas ierīces, protams, var izmantot arī citus

sektoru izmērus, piemēram, 128, 256, 1024 vai pat vairāk.

Lai sektori nesajuktu, tie tiek numurēti, katram sektoram piešķirot savu numuru, sākot no 0 līdz sektoru skaits mīnus 1 (jo numurēšanu sāk no 0). Neviena numurs netiek izlaists.

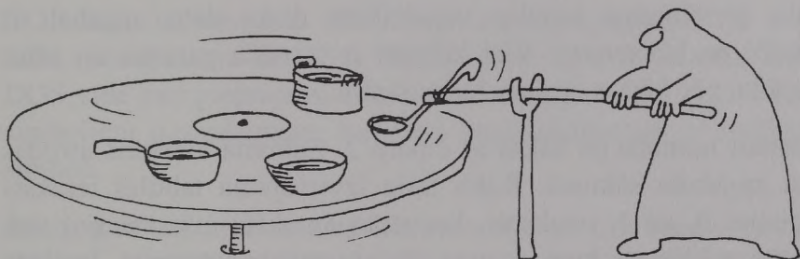
Kā jau minējām, uz diska sektori tiek izvietoti koncentrisku riņķu veidā, un šos riņķus sauc par trekiem. Tas, ka diskete ir divpusēja (*double sided*), nozīmē, ka treki uz tās ir izvietoti no abām pusēm. Cietais disks ir veidots no vairākiem diskietēm, kas griežas vienlaicīgi. Trekus numurē no 0 līdz treku skaits T mīnus 1. 0-tais treks atrodas ārpusē, bet T-1 treks - iekšpusē. Biežāk izmantoti ir ārējie treki, jo nolasīšana no tiem norit ātrāk, tādēļ ka sektoram atvēlētais treka apgabals diska ārpusē ir lielāks nekā iekšpusē.

Lai nolasītu kāda sektora saturu vai ierakstītu tajā datus, iekārta, ko sauc par diska kontrolieri, pārbīda galviņu, kas ir diskiekārtas lasīšanas un rakstīšanas iekārta, tā, lai tā norādītu uz šī sektora treku. Tad, griežot disku, atrod uz šī treka vajadzīgo sektoru un veic prasītās darbības, t.i., nolasa tā saturu vai ieraksta tajā datus.

Protams, lai veiktu visas šīs darbības, datoram ir "jāzina", uz kura treka atrodas pieprasītais sektors. Tādēļ sektorus vēlreiz sanumurē treka iekšienē. Katra treka sektorus numurē no 1 līdz N. Tā, zinot pieprasītā sektora numuru, sektoru skaitu trekā, treku skaitu diska virsmas vienā pusē un virsmu skaitu, dators sarēķina pieprasītā sektora treku.

Sākotnēji DOS operētājsistēma bija izveidota tā, ka darbojās ar 16 bitu sektoru numuriem. Tas nozīmē, ka sektoru skaits nevarēja pārsniegt 65535 sektorus, t.i., 32 megabaitus, ja viena sektora izmērs bija 512 baiti. Tādēļ DOS nebija spējīgs darboties ar diskietēm, kuru sektoru skaits pārsniedza 65535. Lai atrisinātu šo

problēmu, disks tika sadalīts divās vai vairākās daļās, ko sauc par partīcijām. Viena diska partīcija var saturēt ne vairāk par 65535 sektoriem, un katra partīcija tiek uzskatīta par atsevišķu disku.



q.v.

Tad, griežot disku, atrod uz šī treka vajadzīgo sektoru un veic prasītās darbības, t.i., nolasa tā saturu vai ieraksta tajā datus.

Jaunākās DOS versijas darbojas ar 32 bitu sektoru numuriem. Tas ļauj veidot lielākas partīcijas. DOS diska sektori tiek sadalīti grupās:

- rezervētie sektori;
- failu izvietojumu tabulas (*FAT - File Allocation table*);
- pamatdirektorija;
- datu apgabals;
- slēptie sektori.

Rezervētie sektori glabā informāciju par disku: disketes formatējuma veids (piemēram, DOS sistēmas versijai), sektora izmērs, failu izvietojuma tabulu kopiju skaits, sektoru kopskaits, sektoru skaits trekā, diska virsmu skaits u.c.

Nākamajos sektoros atrodas failu izvietojumu tabula FAT, kurā tiek glabāta pilnīga informācija par failu un direktoriju koku (parasti tiek glabātas arī divas šī koka kopijas). Operētājsistēma izmanto tikai pirmo tabulas kopiju, kaut arī pārējās kopijas tiek papildinātas tā, lai tās sakristu ar pirmo kopiju.

Failu izvietojuma tabulas vajadzībām diska datu apgabali ir sadalīti pa klāsteriem. Visi klāsteri ir vienāda garuma un satur vienādu, pēc kārtas esošu sektoru skaitu.

Klāsteru numerācija sākas ar ciparu 2. Sākuma klāsteris atrodas datu apgabala sākumā. Katrs failu izvietojuma tabulas ieraksts (izņemot 0. un 1. ierakstus, kas satur apraksta informāciju) tiek saistīts ar klāsteri, kura numurs atbilst ieraksta numuram. Ieraksta garums (12 vai 16 biti) ir atkarīgs no klāsteru skaita datu apgabalā. Ierakstā glabā šādu informāciju:

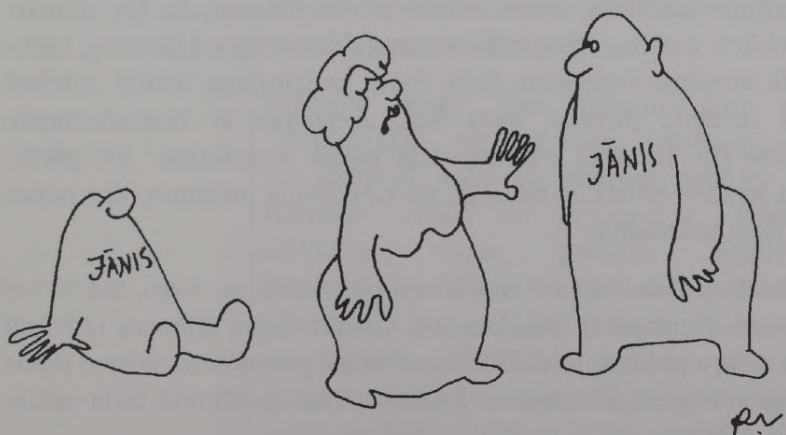
12 biti	16 biti	Paskaidrojums
000	0000	Klāsters nav izmantots.
001	0001	Kļūdainš ieraksts.
002- FEF	0002-EFEF	Klāsters ir izmantots. Saturiski šeit tiek glabāts tā paša faila vai apakšdirektorijas nākamā klāstera numurs.
FF0-FF6	FFF0-FFF6	Aizņemti.
FF7	FFF7	Klāsters satur bojātu sektoru.
FF8-FFF	FFF8-FFFF	Klāsters ir faila vai apakšdirektorijas pēdējais klāsters.

Katra apakšdirektorija, katrs fails tiek aprakstīts ar ierakstu virkni failu izvietojuma tabulā. Katrs šāds ieraksts atbilst vienam klāsterim. Pēdējā klāsterā ieraksts satur speciālu vērtību, pēc kuras var pateikt, ka šis klāsters ir pēdējais, ko aizņem dotais fails.

Pamatdirektorija satur 32 bitu garu ierakstu katram failam diska pamatdirektorijā. **Datu apgabals** glabā failu datus klāsteros atbilstoši failu izvietojuma tabulas ierakstiem. **Slēptie sektori** parasti ir tie pāri palikušie sektori, kas dažādu iemeslu dēļ netika iekļauti datu apgabalā.

DOS failu izvietojuma tabulas izmēra ierobežojumi, tai skaitā failu nosaukumam atvēlētās vietas ierobežojums, nosaka to, ka DOS vidē nav pieļaujami failu nosaukumi, kas būtu garāki par 8 simboliem nosaukumam, bet faila paplašinājumam ir atvēlēti 3 simboli.

Ikreiz, kad tiek izveidota apakšdirektorija, tai tiek rezervēts viens klāsters. Ja nav brīva klāstera, direktorija nevar tikt izveidota.



Vienas direktorijas ietvaros nevar tikt izveidoti divi faili ar vienādiem nosaukumiem.

Vienas direktorijas ietvaros nevar tikt izveidoti divi faili ar vienādiem nosaukumiem. Ja tomēr tiek radīts jauns fails ar tādu

pašu nosaukumu, kā kādam esošajam failam, un ja faila rakstīšanas režīms to pieļauj, tad jaunais fails tiek uzrakstīts vecajam pa virsu. Tas tā nevar gadīties nejauši, jo, šādai situācijai izveidojoties, sistēma vienmēr brīdina par briesmām zaudēt informāciju vecajā failā un tāpēc pārjautā, vai jūs patiešām vēlaties uzrakstīt jauno failu vecā vietā.

No visa iepriekšteiktā var izdarīt secinājumus, kāpēc gadās tā, ka diska vieta it kā pazūd. Jūs saskaitāt visu failu aizņemto vietu, tad pieskaitiet brīvu palikušo vietu un redzat, ka tik un tā iegūtais skaitlis ir krietni vien mazāks par diska kopējo apjomu. Tā tas mēdz gadīties, kad jums ir ļoti daudz mazu failiņu, jo tad veidojas situācija, ka paliek daudz līdz galam neaizpildītu klāsteru, respektīvi, iet zudumā diska vieta.

Izdzēšot kādu failu, netiek atbrīvoti visi klāsteri, ko tas aizņem. Vienkārši sistēma atļauj rakstīt jaunus datus tajos klāsteros, kurus agrāk aizņēma izdzēstais fails. Failu izvietojuma tabulā izdzēstā faila ieraksta pirmais baits tiek aizvietots ar heksadecimālo vērtību E5. Ieraksta atlikušā daļa paliek nemainīga. Arī pārējo failu ieraksti paliek nemainīgi, tie neaizpilda tukšumu, kas radies pēc faila izdzēšanas.

Ja izdzēstā faila dati vēl nav tikuši pārrakstīti pa virsu, tad to var atjaunot. Principā ir rīki, kas ļauj vismaz daļēji atjaunot izdzēstā faila saturu pat tad, ja daži klāsteri tikuši pārrakstīti, salasot datus no neskartajiem klāsteriem. Protams, šādā gadījumā faila saturs nebūs pilnīgs.

Tā rakstot jaunus failus, dzēšot ārā vecos un atkal rakstot pa virsu jaunus, veidojas stāvoklis, kad disks kļūst ļoti safragmentēts. Tas nozīmē, ka dažādu failu klāsteri ir galīgi sajaukušies. Šāds stāvoklis palēlina sistēmas darbu, jo sistēmai, lai nolasītu kādu

failu (vai saglabātu jaunu), ir jāveic vairākkārtīgas diskiekārtas galviņas pārbīdes un tā burtiski “maļ disku”.

Ir izveidoti rīki, kas novērš šādas situācijas. Piemēram, Windows 95 vidē ir tāds diska apkopes rīks *Disk Defragmenter*, kas sakārto disku.

*Ja failu sistēma ir
veids, kādā rakstīt
grāmatā tekstus, tad
disku tipi ir grāmatu
rakstīšanas rīki un
materiāli :
pildspalvas, spalvas,
cirtņi, papīri,
pāpīrusi, akmeņi,
alu sienas.*

Failu sistēma ir tikai loģisks veids, kā izmantot fizisku pieraksta vidi. Mūsdienās pastāv divas vadošās cieto disku tehnoloģijas - IDE (*Integrated Disk Electronics*) un SCSI (*Small Computer System Interface*). Pastāv vēl arī EIDE (*Enhanced IDE*) tehnoloģija. Patiesībā IDE, SCSI un EIDE ir interfeisi, kas tiek izmantoti, lai nodrošinātu datu apmaiņu starp datoru un disku iekārtu.

Starpība starp šīm tehnoloģijām nav “dramatiska”, un parastam lietotājam nav tās jāizprot, taču šie termini parādās arī ikdienas dzīvē, un protams ir īpašību atšķirības, kas pamato atšķirību tehnoloģijās. Datu apmaiņu nodrošina interfeisa karte, kas parasti jāiesprauž pamatplatē, pie kuras savukārt pievieno disku vai kādu citu iekārtu. Līdz ar to varam runāt par IDE, EIDE un SCSI kartēm.

EIDE atšķirībā no IDE tehnoloģijas, kas apkalpoja cietos diskus, kuru apjoms nepārsniedza 528 MB, var darboties ar lielāka apjoma diskiem, turklāt tā var

apkalpot četras iekārtas, ieskaitot diskus, magnētiskās lentas, CD-ROM.

IDE ir lētākā disku apkalpes tehnoloģija. Tā parasti ir iebūvēta datora pamatplatē vai ISA datu maģistrāles speciālā interfeisa platē. IDE disks ir savienots ar pamatplati vai interfeisa plati ar kabeļa palīdzību. Principā signāli IDE kabelī vienkārši dublē ISA datu maģistrāles darbības.

EIDE izskatās un uzvedas kā divas IDE kartes. Tā tiek pieslēgta ar divu kabeļu palīdzību. Tomēr EIDE disku ne vienmēr var pieslēgt vecajam PC modelim sistēmas ierobežojumu dēļ. Bet darbinot jaunās sistēmas Windows 95, Windows NT, OS/2, kuras pieļauj vairāku operāciju paralēlu izpildi, EIDE priekšrocība ir tā, ka tā var izmantot otru kabeli, kamēr pirmais ir nodarbināts.

Atšķirībā no IDE diska un ISA datu maģistrāles, kas ir IBM un Intel savietojamas, SCSI ir standartinterfeiss visiem datoru veidiem. SCSI izmanto Macintosh datori, RISC darba stacijas un pat daži lieldatori. SCSI tehnoloģija vienmēr ir atbalstījusi iespēju apvienot diskus, magnētiskās lentas, CD-ROM iekārtas. SCSI diski arī apjoma ziņā ir krietni lielāki par IDE diskus.

SCSI interfeiss nodrošina daudzu iekārtu pieslēgšanu pie vienas SCSI kartes. Pēdējā laikā ir parādījušās dažādas SCSI modifikācijas, kas palielina ātrdarbību vai datu drošumu.

Tomēr EIDE diski ir vairāk izplatīti nekā SCSI diski, jo SCSI diski ir gandrīz divreiz dārgāki nekā tāda paša apjoma EIDE diski. Tāpēc parastam galda datoram nav vērts tos iegādāties. Taču līdzekļu izdošana par SCSI disku ir tā vērts, ja nepieciešams veidot jaudīgu serveri.

Programmēšana

Programmēšanas uzdevums ir veidot datoram saprotamus priekšrakstus uzdevuma veikšanai.

Labai programmai ir jāparedz visas situācijas, kas darba procesā var rasties, jānorāda darbības katrā konkrētajā situācijā, jāprot reaģēt uz datora "ziņojumiem" utt.

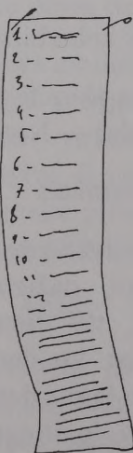
Patiesi labu un drošu programmu uzrakstīt nemaz nav tik vienkārši.

Pamēģiniet kaut vai detalizēti aprakstīt ikrīta mošanās un mazgāšanās procedūru tā, lai jebkurš, sekojot TIKAI jūsu aprakstam, ne reizi nekļūdītos.

Tuntuļa kunga saldo sapni nežēlīgi pārtrauca dzelžaina pavēle, ko līdz ar dunku sānos viņš saņēma no savas sievas:

— Mosties!

Tuntuļa kungs negribīgi atvēra acis un jau apgriezās uz otriem sāniem, lai vēl kādu brītiņu pagulētu, kad atskanēja nākamā pavēle:



Pamēģiniet kaut vai detalizēti aprakstīt ikrīta mošanās un mazgāšanās procedūru.

— Celies augšā, un ātri mazgāties!

Bija vien jāceļas. Negribīgi Tuntuļa kungs aizvilkās uz vannas istabu, nomazgājās un gribēja jau ķerties pie zobu tīrīšanas, bet no virtuves atskanēja viņa sievas balss:

— Neaiztiek manu zobbirstīti, tavējā ir sarkanā krāsā!

Tuntulis ielika sievas balto zobbirstīti atpakaļ glāzē, bet viņa sarkanā bija kaut kur pazudusi. Viņš pat nostājās četrāpus un paskatījās zem vannas, bet birstīte nekur nebija atrodama. Brīdi padomājis, viņš to ziņoja sievai:

— Manas zobu birstes nekur nav!

— Paņem skapītī jaunu! — atsaucās sieva un piebilda: — Noteikti noskujies!

Tuntulis nopūtās ...

Dažiem cilvēkiem nepārtraukti ir vajadzīgi norādījumi, dažiem nē, bet, runājot par datoriem, tad tiem visiem IR vajadzīgi norādījumi, ko darīt, kā darīt un cik ilgi darīt. Par laimi tiem absolūti nerūp, kāpēc darīt. Tas ir stingri jāzina cilvēkam, kas sagatavo datoram priekšrakstus jeb programmas. Šādus cilvēkus sauc par programmētājiem. Un būt par programmētāju ir kaut kas īpašs. Rakstīt

programmas ir, kā radīt kaut ko jaunu. Jaunu nevis materiālā ziņā, bet gan vairāk garīgā, jo pēc tam, kad programma ir izveidota un pierakstīta datoram saprotamā valodā, tā sāk darboties — dzīvot. Tāpēc nav brīnums, ka daži programmētāji neapzināti pielīdzina sevi dievībai un kļūst iedomīgi un nepieejami. Piedodiet viņiem un pacientieties netiesāt pārāk bargi.

Kas ir programmas?

Bet tagad atgriezīsimies pie programmām. Mēs jau minējām par algoritmiem. Programmas ir datoram saprotamā veidā pasniegti algoritmi. Dators dara tikai to, ko viņam liek programma. “Pats no sevis” tas neko nevar ne pielikt, ne atņemt.

*Kā dators “saprot”,
ko tam “saka”
programma?*

Datora atmiņā darbojas elektriski lādiņi jeb, kā jau runājām, loģiskie cipari 0 un 1. Tie veido virknes, kas arī nosaka datora darbību. 0 un 1 virknes ērtāk ir pierakstīt 16-nieku pierakstā. Tāpēc atmiņas izdrukās figurē 16-nieku skaitļi, piemēram,

AE0B 107A 93FF ...

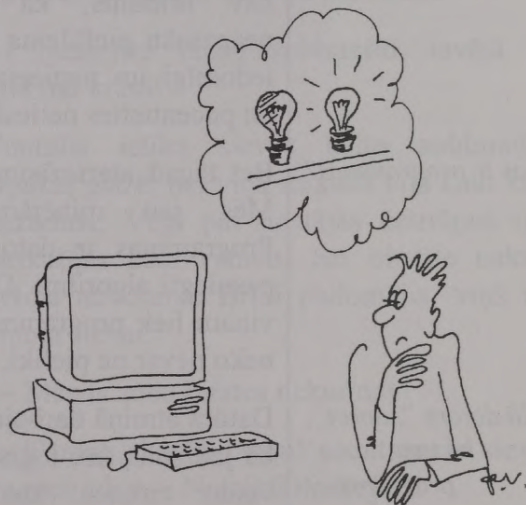
Inženieri, kas programmē zema līmeņa valodās, mēdz lasīt šādas izdrukas, bet tas, kā jūs saprotat, nav nemaz tik viegli un parocīgi.

Arī komandas un norādījumus datoram var dot ar 16-nieku skaitļiem. Šādu

Pašos pamatos visi datori darbojas tikai ar elektrību un saprot tikai sprieguma atšķirību.

Programmēšanas valodas ir domātas cilvēkiem, lai tiem būtu vieglāk izrikt datoru. Dažādās programmēšanas valodas nenozīmē, ka datori darbojas dažādi. Tas nozīmē, ka programmētājs uzdevuma risināšanai ir izvēlēties vienu vai citu programmēšanas valodu.

programmas pierakstu sauc par mašīnkodiem. Taču mašīnkodu programmēšana ir sarežģīta. Tāpēc cilvēki savas dzīves atvieglošanai izdomāja programmēšanas valodas.



...datori darbojas tikai ar elektrību un saprot tikai sprieguma atšķirību.

Pirmais solis bija programmēšanas valoda Asemblers (no angļu valodas *Assembler* varētu tulkot kā “kopā lasītājs”). Asemblera valodā konkrēti mašīnkodi, kas pilda noteiktu funkciju, tika aizstāti ar komandām — pilniem vārdiem vai saīsinājumiem. Tādējādi programmētājs pēc komandas

nosaukuma varēja noteikt, ko tā dara. Un nebija nepieciešamības iegaumēt ciparu un burtu virknes.

Procesora reģistriem Asemblera valodā tika piešķirti burtu indeksi, arī atmiņa tika iezīmēta simboliski. Programmējot Asemblerā, nav nepieciešamības veikt lielu rēķināšanu 16-niekos, jo konkrētās adreses, nobīdes utt., programmai izpildoties, tiek aprēķinātas automātiski.

Asemblerā rakstīta programma izskatās, piemēram, šādi:

A21:

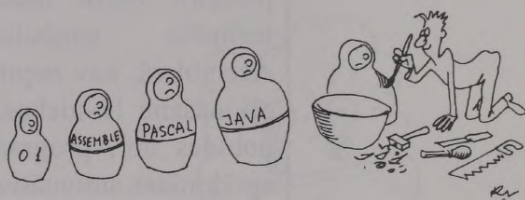
```
ADD  BX,AX ;Saskaita reģistrus
SHL  CX,1  ;Nobīda reģistru, pēc būtības pareizina ar 2
JMP  A20   ;Nodot vadību uz iezīmi A20
```

Asemblers pieraksta ziņā ir ērtāks nekā mašīnkodi, taču visa programmēšana notiek procesora komandu līmenī. Tas prasa no programmētāja procesora darbības principu pārzināšanu, atmiņas plānošanas un sadalīšanas prasmi u.c. sarežģītas lietas. Mūsdienās programmētāji griežas pēc Asemblera palīdzības tikai tajos gadījumos, kad, programmējot procesora līmenī, citas programmēšanas valodas palīdzēt nevar.

Programmēšanas valoda ir valoda, kādā cilvēks sazinās ar datoru, un tā ir domāta tikai cilvēkiem jeb, precīzāk,

Jā, bet kā tad dators,
kas darbojas tikai ar
0 un 1, saprot, ko
cilvēks tam raksta
Asemblerā vai kādā
citā
programmēšanas
valodā?

programmētājiem. Dators “nesaprot”
programmēšanas valodas: tam ir vajadzīgs
tulks. Par tulkiem datoram kalpo speciālas
programmas, ko sauc par kompilatoriem.



*Attīstoties kompilatora idejai, radās arvien jaunas un
jaunas programmēšanas valodas.*

Kompilators ir tas, kas pārtulko Asemblera
vai kādas citas valodas programmu
mašīnkodos. Attīstoties kompilatora idejai,
radās arvien jaunas un jaunas
programmēšanas valodas. Tajās komandas
bija arvien spēcīgākas. Līdz ar to
programmas teksti tika pierakstīti īsāk, un
kļuva iespējams veidot lielākas un
sarežģītākas programmas.

Līdzās kompilatoriem pastāv arī
interpretatori. Interpretatori arī ir datora
tulki, taču tie atšķiras no kompilatoriem.
Kompilators “pārtulko” visu programmu un
tikai tad dara to zināmu procesoram, bet
interpretators tulko pa komandai un katru
pārtulkoto programmu pārsūta procesoram.

Kompilators ir kā grāmatu tulkotājs, bet interpretators ir kā sinhronais tulkotājs. Nevar teikt, ka viena metode ir laba, bet otra slikta. Katrā konkrētā situācijā jāpiemeklē optimālākais varinats.

Kompilatora un interpretatora salīdzināšanai izveidosim tabulu:

KOMPILATORS	INTERPRETATORS
Kompilators pārveido programmu mašīnkodos vienreiz, radot īsu un efektīvu kodu vairākkārtīgai lietošanai. Izveidotais kods paliek nemainīgs, un tas ir aizsargāts. Mainot programmu, kods jāveido no jauna.	Interpretators pārveido tikai vajadzīgās programmas. Tas ļauj veidot programmas, kas pašas sevi modificē. Var pārtraukt programmas darbu jebkurā vietā, pamainīt un tad turpināt tās izpildi.
Iegūto kodu var darbināt uz jebkura datora (ja vien tā ātrdarbība, atmiņas resursi utt. atļauj). Tam nav vajadzīgas nekādas papildu programmas.	Lai programmu izpildītu uz cita datora, uz tā obligāti jābūt uzinstalētam attiecīgajam interpretatoram. Pretējā gadījumā dators programmu "nesaprot".

Dabā pastāv dažādu veidu hibrīdi. Tā, saliekot kopā interpretatoru ar interpretējamo programmu, iegūstam

hibrīdu, ko varam nosaukt par kompilētu programmas moduli. Šajā gadījumā interpretators vienmēr “nāk līdzi” programmai. Tas ļauj programmu darbināt uz dažādiem datoriem.

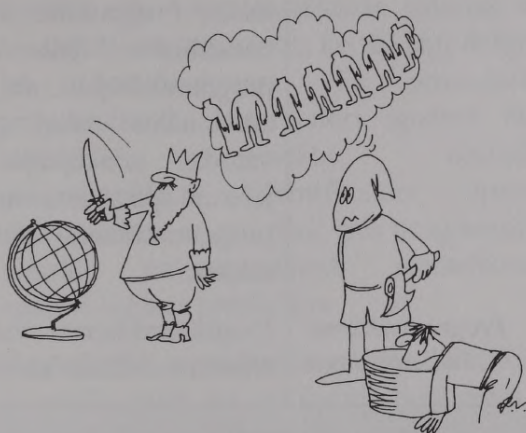


...interpretators ir kā sinhronais tulkotājs.

Visi faili ar paplašinājumu *exe*, *com* un daļēji *dll* ir kompilētas programmas. Lielākā daļa programmu ir kompilētas, tāpēc varētu šķist, ka interpretatoriem nav vietas, bet nekā. Basic valoda tradicionāli ir interpretēta, un Visual Basic nav izņēmums. Pašlaik plaši reklamētā un, iespējams, nākotnē ļoti plaši lietotā valoda Java tiek interpretēta, un to nekad nevarēs nokompilēt atšķirībā no Basic, kas radīs iespēju to izpildīt visur, kur ir Java valodas interpretators neatkarīgi no datora procesora vai operāciju sistēmas.

Liekas, ka esam
tikuši skaidrībā ar
to, kā savā starpā
sazinās dators un
programmētājs.

Tagad sīkāk
pievērsīsimies
programmēšanas
valodām un to
iedalījumam.



*Augsta līmeņa valodas operators bieži aizstāj pat simtiem
zema līmeņa valodas operatoru.*

Bieži vien dzirdam runājam par zema un augsta līmeņa programmēšanu. Te ir jāievēro viens princips: jo vairāk programmētājam, veidojot programmas, ir jāizprot un jāseko procesora darbībai, jo zemākā programmēšanas valodā viņš programmē. Tādējādi viszemākā programmēšana ir mašīnkodos un arī Asemblerā, kur programmēšana ir smags tehnisks process.

Augsta līmeņa valodas operators bieži aizstāj pat simtiem zema līmeņa valodas operatoru. Sakot, ka viena programma ir rakstīta zema, bet otra augsta līmeņa valodā, nekad netiek noniecināts programmētāja

darbs. Programmēt zemāka līmeņa valodā nenozīmē būt zemākas klases programmētājam, un otrādi! Šis sadalījums attiecināms tikai uz programmēšanas valodas koncepciju. Programmētājs kā jebkurš speciālists var būt labs vai slikts. Ar programmēšanas valodām tam nav nekāda sakara.

*Programmēšanas
valodu klases*

Programmēšanas valodas saturiski tiek iedalītas dažādās klasēs.

PROGRAMMĒŠANAS VALODU KLASES

Procedurālās
programmēšanas valodas

*PL/I, Fortran, Pascal, C,
Basic*

Loģiskās programmēšanas
valodas

Simula, LISP, Prolog

Objekt orientētās (OO)
programmēšanas valodas

C++, OO Pascal, Java+

Hibrīdidejas jeb vizuālās
programmēšanas sistēmas

Visual Basic, MS Access

Samērā labi programmēšanas valodas var kvalificēt pēc objektiem, ko tās ļauj definēt, jeb identifikatoriem. Tā procedurālā programēšanā identifikatori apzīmē datu struktūras, objektu orientētā programmēšanā identifikatori apzīmē objektus, kas ir gan dati, gan programmas, loģiskā programmēšanā identifikatori apzīmē loģikas predikātus.

*Procedurālā
programmēšana*

Procedurālā programmēšana ilgu laiku bija, un varētu teikt, ka vēl šodien lielā mērā ir, valdošā programmēšanas metode. Tās pamatā ir ideja veidot programmu no vairākām daļām: no daļas jeb procedūras konkrēta uzdevuma veikšanai un no galvenās daļas jeb programmas ķermeņa, kas manipulē ar pārējām daļām. Šādas pieejas rezultātā radās iespēja veidot lielākas programmas, izvairīties no lieka darba (vienu procedūru var pielietot vairākas reizes dažādiem datiem), veiksmīgāk “izķert” programmas kļūdas.

*Objektorientētā
programmēšana
(OOP)*

OOP ir liels solis uz priekšu programmēšanas attīstībā. Vienkāršoti OOP principu varētu aprakstīt tā: darbošanās notiek ar definētiem objektiem. Objekts ir vienlaicīgi gan programma, gan dati, kas paši zina, kā ar tiem jādarbojas. Objektiem tiek piekārtotas īpašības. Savas īpašības objekts var mantot no viena vai vairākiem

vecākiem, modificējot vai nemodificējot to oriģinālās īpašības.



Objekts ir vienlaicīgi gan programma, gan dati, kas paši zina, kā ar tiem jādarbojas.

Kad objekts ir nedefinēts, kādu no elementiem pasludinām par šī objekta pārstāvi, kam ir spēkā arī objekta īpašības. Skan samudžināti, vai ne?

Pamēģināsim apskatīt vienkāršotu piemēru. Definējam objektu mašīna: 4 riteņi, motors, stūre, gāze, sajūgs, bremze u.c. Tai piemīt īpašības: ātrums, benzīna patēriņš, ieskrējiens utt. Tagad ņemam konkrētu elementu MAZDA un definējam to kā objektu mašīna. Tagad varam darboties ar

elementu MAZDA, kam ir 4 riteņi, motors, stūre ... droši vien, ka tagad jūs varētu turpināt.

Pateicoties OOP, mēs varam izmantot citu programmētāju izveidotos objektus. Ja procedurālajā programmēšanā mēs varējām izmantot svešas apakšprogrammas, uz svešiem pamatiem it kā uzbūvējot savu māju, tad objektorientētajā programmēšanā mēs varam paņemt sveša bruņurupuča čaulu un tur iekšā ielikt savu bruņurupuci. Tādējādi OOP radikāli atvieglo programmu rakstīšanu, dodot mums iespēju svešos interfeisos ielikt savu jēgu.

Loģiskā programmēšana

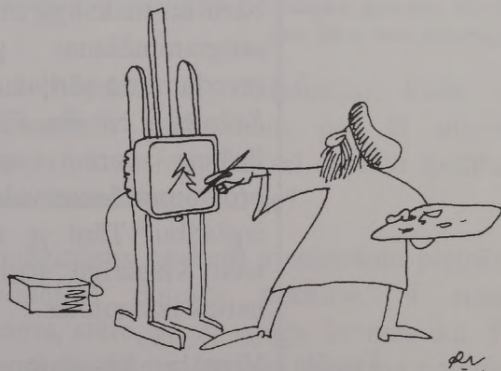
Loģiskā programmēšana attīstījās laikā, kad pasaulē bija liels ekspertsistēmu, zināšanu bāzu un mākslīgā intelekta bums. Loģiskās programmēšanas pamatā ir loģiskie izvedumi no sērijas: *Visi cilvēki ir mirstīgi. Sokrāts ir cilvēks. Tātad Sokrāts ir mirstīgs.* Pašreiz varam apgalvot, ka loģiskās programmēšanas valodas nav guvušas plašu izplatību. Tām ir specifiski pielietojumi, taču vispārīgu uzdevumu risināšanā tās netiek izmantotas.

Vizuāla programmēšana

Vizuāla programmēšana ir kaut kas oriģināls un īpašs. Tajās programmas vairs netiek rakstītas, bet gan, ja varētu tā teikt, dizainētas. Protams, nevaram apgalvot, ka

programmēšanas darbs nav vajadzīgs, tas tikai tiek atvirzīts it kā otrā plānā.

Varētu teikt, ka vizuālā programmēšana ir OOP atvieglots variants. Mums nav jāiedziļinās OOP smalkumos un viltībās, bet mēs varam izmantot jau definētus grafiskā interfeisa u.c. elementus. Šos elementus mēs "uzzīmējam" uz ekrāna līdzīgi kā zīmēšanas paketēs. Vēlāk mēs tiem varam paši pierakstīt uzvedību dažādās situācijās, iedod savu jēgu šiem elementiem, tādējādi ļoti ātri izveidojot glītu, profesionāla izskata programmu. Vizuālā programmēšana ir vienīgais programmēšanas veids, ko būtu vērts pamēģināt neprofesionālim



Šos elementus mēs "uzzīmējam" uz ekrāna līdzīgi kā zīmēšanas paketēs.

*Kādu
programmēšanas
valodu lietot?*

Programmēšanas valodas izvēlei jāņem vērā divi galvenie faktori. Pirmais ir projektā iesaistīto programmētāju pieredze un paradumi. Bieži vien tas ir pats svarīgākais, jo daudzas lietas var izdarīt ar jebkuru programmēšanas valodu. Otrs faktors ir uzdevuma specifiskās prasības. Tādēļ arī ir tik daudz dažādu programmēšanas valodu, jo katra no tām ir labākā kādu specifisku uzdevumu risināšanai.

Datortīkli

Tīkls kā termins parādās gandrīz katrā zinātnes un tehnikas nozarē. Datortīkli ir savā ziņā līdzīgi gan zvejnieku tīkliem, gan spiegu tīkliem.



R.V.

...ar kabeļiem saslēdzot datorus kopā, tādējādi izveidojot tā saucamo datoru tīklu.

Paplašinoties datoru pielietojumu jomai, rodas nepieciešamība apmainīties ar informāciju starp datoriem. To panāk, ar kabeļiem saslēdzot datorus kopā, tādējādi izveidojot tā saucamo datoru tīklu. Izšķir lokālos jeb neliela mēroga un globālos jeb liela mēroga tīklus. Lokālu tīklu veido, piemēram, vienas iestādes, organizācijas vai biroja datoru saslēgums, bet tiem jāatrodas vienā ēkā vai netālu blakus ēkās. Globālajā tīklā kopā varētu būt saslēgti datori pat vairākās valstīs.

Datoru tīklā informācija tiek pārraidīta pa kabeļiem. Ja jums jānosūta kāda grāmata kolēģim no otrā stāva, bet jūs atrodaties ceturtajā, tad šajā gadījumā viss ir skaidrs.

Jums ir vairāki varianti, kā to izdarīt. Piemēram, palūgt kolēģim uzkāpt pie jums vai nokāpt lejā pašam. Var izmantot arī viltīgākus paņēmienus, piemēram, piesiet grāmatu striķī un nolaist lejā pa logu. Bet kā notiek informācijas pārsūtīšana pa kabeļiem? Protams, elektroniskā formā. Pārraidēs pārsūtīšanu un saņemšanu nodrošina datoru tīkla pastnieki, ko sauc par protokoliem. Pārsūtīšanas ātrumu nodrošina tīkla ātrdarbība, kabeļu ietilpība u.c.

Lokālais tīkls atmaksājas birojā, kurā ir kaut vai tikai divi datori. Lokālais tīkls dod mums iespēju izmantot visu biroja datoru visas perifērijas iekārtas, lai arī kurā biroja vietā tās neatrastos. Parasti uz vairākiem datoriem ir viens lāzerprinteris. Ja tas ir pievienots datoram, kas ir saslēgts lokālā tīklā, tad uz šī printera var drukāt no jebkura lokālā tīkla datora. Arī CD iekārtas nav nepieciešamas vienmēr uz visiem datoriem.

Lielu failu apmaiņa starp diviem datoriem ir ļoti neērta ar disketēm. Tīkls šo problēmu atrisina ērtā veidā. Pateicoties lokālajam tīklam, iespējams organizēt grupas darbu. Glabājot dokumentus vienuviet, vairākiem cilvēkiem ir iespējams ar tiem vienlaicīgi strādāt dažādās vietās.

Globālie jeb liela mēroga tīkli ir tehniski daudz grūtāk realizējami nekā lokālie tīkli. Līdz ar to savādāks ir arī to pielietojums. Globālie tīkli neapšaubāmi galvenokārt ir komunikācijas līdzeklis. Izplatītākais pielietojums ir elektroniskais pasts (e-pasts).



...sniedzot iespēju katram lietotājam piekļūt pie globālās datu bāzes un saņemt aktuālāko informāciju.

Izmantojot globālā tīkla aparatūru, mēs varam saslēgt arī divus vai vairākus firmas birojus it kā vienā lielā lokālā tīklā. Būs pieejamas visas tās pašas funkcijas kā lokālajā tīklā, tikai ātrdarbība būs daudzreiz zemāka.

Globālos tīklus izmanto gan informācijas pārraidei no viena punkta daudziem

abonentiem, lai tie vienmēr būtu informēti par pēdējiem jaunumiem, gan sniedzot iespēju katram lietotājam piekļūt pie globālās datu bāzes un saņemt aktuālāko informāciju.

Pastāv arī hibrīdtehnoloģijas. Ja daudzi abonenti griežas pie vienas datu bāzes, tad informācijas iegūšanas process norit ļoti lēni, it īpaši, ja tas tiek veikts caur noslogotu tīklu. To var paātrināt, izmantojot bieži lietojamu datu bāžu lokālas replikas, kas tiek atjaunotas pēc vajadzības.

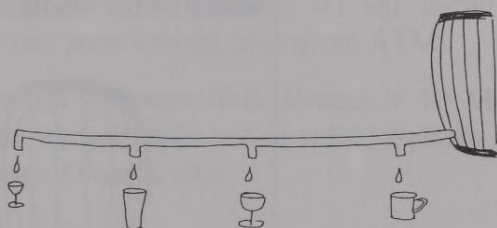
Lokālais tīkls

Ar Lokālo tīklu (*Local Area Network - LAN*) tiek saprasts tīkls vienas ēkas vai organizācijas ietvaros. Šādā tīklā ir vairāki datori, un var būt vairāki dažādas nozīmes serveri. Serveris ir dators, kas ļauj piekļūt informācijai uz tā un nodrošina noteiktu tīkla procesu darbību. Lokālie tīkli strādā ātri, jo nekur netiek izmantotas standarta telefonlīnijas, kur datu apmaiņas ātrums ir stipri zemāks.

Lokālais tīkls fiziski un ar programmām sastāv no septiņiem līmeņiem.

Pirmais līmenis ir fiziskā kabeļu sistēma. Var izmantot koaksiālo kabeli, kas ir līdzīgs televizora antenas vadam. Šādi tīkli bija ļoti

populāri pirms vairākiem gadiem. Datorus vienoja virknē, un mazā tīklā nebija nepieciešama nekāda cita aparatūra kā vien tīkla kartes katrā datorā. Tas bija ērti, vienkārši un lēti. Taču tīklu raksturo strauja izaugsme.



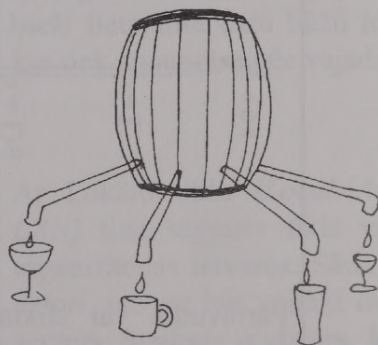
Es.

...datorus vienoja virknē...

Pārrāvuma vai slikta kontakta gadījumā šādā tīklā tiek traucēts viss tīkla darbs. Šādas kļūdas bieži vien ir grūti atrodamas. Ir dzirdēti stāsti par apkopējām, kas nejauši atvieno tīkla vadu, un firmas darbs ir paralizēts pusi dienas, kamēr speciālisti atrod triviālo kļūdu.

Šodien galvenokārt tiek vilkti vītā pāra tīkli. Šādā tīklā ir nepieciešams komutācijas punkts, kurā atrodas aktīvā komponente (*HUB vai Switch*), kas zvaigznveidīgi savieno visus datorus.

Katram datoram tīklā ir atsevišķs vads. Šāds risinājums ir daudz drošāks un vieglāk apkalpojams. Tiesa, šāds tīkls ir dārgāks, un tā izveide ir kārtīgi jāizplāno. Trīs datoru saslēgšanas gadījumā šāds risinājums ir stipri dārgs, taču, tīklam augot, tas atmaksājas un ir nenozīmīgi dārgāks. Zvaigznes slēgums šodien faktiski tiek uzskatīts par vienīgo nopietno slēgumu.



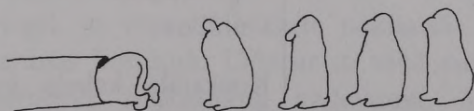
...atrodas aktīvā komponente (HUB vai Switch), kas zvaigznveidīgi savieno visus datorus.

Pašlaik pazīstami ir arī optiskie vadi. Optiskos vadus galvenokārt izmanto, lai savienotu vairākus komutācijas punktus, taču var arī visu tīklu būvēt tikai no optikas.

Nākošajā līmenī tiek noteikts veids, kā tīkla kartes atpazīst viena otru un nodod

informāciju. Pirmie divi līmeņi nosaka, kāds būs tīkla karšu protokols. Populārs ir *Ethernet* protokols. Rīgā visi pārsvarā izmanto *Ethernet* protokolu, līdz ar to visas tīkla kartes ir *Ethernet* kartes. Vēl ir *Token Ring* protokols, kas pasaulē ir pietiekami populārs. *Arcnet* protokols vairāk neattīstās un ir praktiski “miris”. Starp komutāciju punktiem datu pārraide vēl var notikt ar citiem protokoliem, piemēram ATM.

Parastā *Ethernet* tīkla ātrums ir 10 Mega biti sekundē, taču arvien vairāk izplatās 100 Mbps *Ethernet* tīkli.



22

LAN var būt tikai viens ziņojums laika brīdī.

Ethernet protokola darbības principu varam iedomāties kā garu kopmītnes koridoru. Katra tīklā ieslēgtā karte stāv pie atvērtām durvīm un klausās. Ja tā izdzird savu vārdu, tā noklausās ziņojumu un nodod to datoram (nākošā līmeņa protokolu). Kad tai jānodod ziņojums (saņemts uzdevums no augšējā protokola), tā gaida klusumu

koridorā. Klusuma brīdī tā izkļiedz adresāta vārdu un ziņojumu. LAN var būt tikai viens ziņojums laika brīdī. Ja vienā koridorā ir par daudz durvju, tad bieži gadās konflikti. Šādos gadījumos tīkla ātrums krītas, jo daudzi tīkla sūtījumi ir vairākkārtīgi jāatkārto.

Pārslodze tīklā parādās lavīnveidīgi, nevis pakāpeniski. Mēs pieslēdzam vēl vienu datoru tīklā, un tas ir kritiskais punkts. Daudzi tīkla sūtījumi ir jāatkārto, tāpēc tīkla noslodze vēl vairāk pieaug, un līdz ar to pieaug arī kolīziju skaits: "*Mazs cinītis gāz lielu vezumu*". Šādos gadījumos nepieciešams sadalīt vienu tīkla segmentu vairākos.

Koaksiālā kabeļa gadījumā jādala viena datoru virkne vairākās. Šādu situāciju gan grūti iedomāties, jo problēmas drīzāk būtu radušās no kabeļa, ne no tīkla noslodzes. Vītā pāra gadījumā viena zvaigzne jāsadala vairākās, uzliekot papildus koncentratorus un starp tiem ievietojot iekārtas, ko sauc par *Bridge*.

Šādi sasegmentējot tīklu, mēs samazinām tā noslodzi. Starp segmentiem *Bridge* izlaiž tikai to informāciju, kas tiešām nepieciešama citam segmentam. *Bridge* ir kā dežurants koridora vidū, kas izkļiedz

vēlreiz vienā koridora galā izkliegtu informāciju otrā koridora galā, bet tikai tad, ja adresāts tiešām ir otrā koridora galā.

Pašlaik *hubs* sāk izkonkurēt iekārtas, ko sauc par *switch*. *Switch* piegādā katrai kartei tikai tai paredzēto informāciju, tie ir stipri gudrāki un sarežģītāki, un tādējādi arī dārgāki par *hubs*. *Switch* šodien varētu būt svarīgi no datu drošības viedokļa. Piemēram, lielā bankā, kur datu apstrāde - naudas pārkaitīšana notiek ar datoru palīdzību, kāds "hakeris", ja tas atrodas vienā segmentā ar naudas operāciju pārvaldi, var viltot *Ethernet* informācijas pakas. Kārtīgam hakerim tas varētu būt viegli, jo viņa tīkla karte noklausās visas sarunas koridorā. Līdz ar to viņš zina, kā noformēt maksājuma uzdevumu. *Switch* gadījumā noklausīšanās nav iespējama.

Lokālā tīkla tehnoloģijas parasti neiekļauj kodēšanu, jo visa informācijas apmaiņa notiek vienas firmas ietvaros. Kodēšana varētu sarežģīt, sadārdzināt un palēnināt informācijas apmaiņas procesu.

Ar 100/10 Mbps kombinēto *switch* mēs varam sadalīt tīklu vairākos 10 Mbps segmentos, kur katram segmentam būs 10 Mbps pieeja serverim, jo serveris ir pieslēgts pie *switch* ar 100 Mbps.

Ethernet tīkls ir ideāls nevienmērīgi noslogotam tīklam. Katrs lietotājs lieto tīklu, cik un kad vajag. Parasti birojā ir tikai daži lietotāji, kas daudz lieto tīklu.

Tāda pielietojuma gadījumā, kur daudzi lietotāji vienmērīgi daudz lieto tīklu, *Ethernets* pārslodzes gadījumā kļūtu nelietojams pastāvīgo kolīziju dēļ.

Šādos gadījumos labāks būtu *Token Ring* (stafetes aplis). *Token Ring* gadījumā “runāt” drīkst tikai tas, kuram konkrētā brīdī ir atļauja. Tīkla karte norunā savu sakāmo un nodod atļauju runāt tālāk nākošai kartei (token – stafetes nūjiņa).



Token Ring gadījumā “runāt” drīkst tikai tas, kuram konkrētā brīdī ir atļauja.

Tīkla pārslodzes gadījumā tā ātrdarbība pakāpeniski krītas. Taču, ja tīklā ir daudz neaktīvu lietotāju, viņu kartēm lieki tiek nodota atļauja raidīt. *Token Ring* nav pārāk izplatīts, taču tīkla uzbūves principi tiek

izmantoti jaunos, ātrākos tīkla pārraides protokolos, piemēram, ATM.

Ethernet vai *Token Ring* tikai nodrošina datu pakešu pārraidi no vienas tīkla kartes otrai. Nākošie trīs līmeņi izmanto pirmos divus līmeņus, nodrošina globālu adresāciju un kontroli, lai informācija bez kļūdām nonāktu adresātam.

Nākošos trīs līmeņus realizē protokoli TCP/IP, SPX/IPX un Netbios. To funkcijas ir samērā vienādas. TCP/IP ir nācis no operācijas sistēmas UNIX un tiek izmantots arī Internet tīklā. SPX/IPX ir firmas *Novell* izstrādātais protokols, kas ir Netware vēsturiskais protokols. *Netbios* ir firmas *Microsoft* izstrādātais protokols, ko izmanto *Microsoft* tīklu sistēmas. *Netbios* darbojas tikai viena lokāla tīkla ietvaros, bet šajā gadījumā visefektīvāk.

Šodien aktuālās tīkla operāciju sistēmas UNIX paveidi, *Windows NT*, *Novell* var izmantot visus šos protokolus. Bieži vien normāli ir izmantot visus šos protokolus vienlaicīgi. Ja jūsu lokālais tīkls ir ieslēgts Internet tīklā, tad jums noteikti jādarbina TCP/IP. *Windows for Workgroups* vai *Windows 95* darba staciju resursu apmaiņa notiek ar *Netbios* palīdzību. Savukārt, *Novell* serveris izmanto SPX/IPX.

Sesto un septīto tīkla līmeni realizē tīkla operāciju sistēmas.

Tas viss ir ļoti sarežģīti, un ikdienā lietotājam nav jāzina, taču ar visiem šiem terminiem mēs saskaramies, pērkot datorus, kā arī tad, kad rodas problēmas tīklos.

Velkot analogijas ar transportu, mēs varētu teikt, ka pirmie divi līmeņi ir autoceļi un kuģu ceļi. Tīklu protokoli ir transporta kompānijas. Tīkla operāciju sistēma ir veikalu un noliktavu sistēma, kas apgādā pircējus ar precēm. Pircēji var izmantot dažādas veikalu grupas, vienā laika momentā atrodoties tikai vienā no tām. Veikali var izmantot vienu vai vairākas transporta kompānijas vienlaicīgi, un transporta kompānijas var izmantot dažādus transporta veidus. Taču jāievēro dažas īpatnības, piemēram, tas, ka kuģiem nav vajadzīgas sliedes, bet mašīnām ziemā ieteicams lietot ziemas riepas.

Jebkurā lielākā tīklā bez vadiem un tīkla kartēm ir aktīvās komponentes, kuras daļa četrās lielās grupās.

Tālākā tabulā attēlots tīkla aktīvo komponentu funkciju apraksts

<i>Repeater</i> (Atkārtotājs)	Pastiprina signālu. Darbojas pirmajā tīkla līmenī. Izmanto, lai pagarinātu tīkla segmentu. Parastie <i>Hub</i> ir <i>Repeater</i> .
<i>Bridge</i> (Tilts)	Savieno tīkla segmentus. Strādā pirmajos divos tīkla līmeņos. Informāciju, kas attiecas tikai uz vienu segmentu, nelaiž uz citu. Tādējādi <i>Bridge</i> var izmantot tīkla atslogošanai. Galējs <i>Bridge</i> variants ir <i>Switch</i> , kad katra tīkla mašīna ir savā segmentā un saņem tikai sev paredzēto informāciju.
<i>Router</i> (Maršrutētājs)	Iekārta strādā pa tīkla pirmajiem trim līmeņiem. <i>Router</i> uztur globāla mēroga tīkla adresāciju. <i>Router</i> ir nepieciešams, lai ieslēgtu lokālo tīklu kādā globālā tīklā.
<i>Gateway</i> (Vārti)	Iekārta, kas strādā pa visiem tīkla līmeņiem. Ar <i>Gateway</i> palīdzību mēs varam pārtulkot jebkura veida un līmeņa tīkla transportu uz cita veida tīkla transportu.

Tīkla operāciju sistēmas var būt būvētas kā serveris un daudz klientu, vai kā daudz vienādu klientu. Klienta - servera arhitektūras tīklu vieglāk uzturēt un apkalpot. Šinī gadījumā visa kopīgā informācija glabājas uz servera. Tīkla administrators vienkārši regulē pieejas

autorizāciju dažāda informācijai ar parolu palīdzību. Tāpat datu kopēšana drošības dēļ notiek no servera. Serveri mēs varam vieglāk nodrošināt arī pret nesankcionētu fizisku piekļūšanu, strāvas kritumiem utt. Nopietnos tīklos vienmēr ir izdalīti tīklu serveri.


fu.

Ar Gateway palīdzību mēs varam pārtulkot jebkura veida un līmeņa tīkla transportu uz cita veida tīkla transportu.

Peer to peer tīklos, kur visas mašīnas var ļaut citiem piekļūt pie saviem resursiem, administrēšana ir gandrīz neiespējama. Taču šodien modernās operāciju sistēmas uz katra datora (*Windows for Workgroups*, *Windows 95*, *Windows NT*) jau nodrošina šīs iespējas. Šādā veidā var ērti apmainīties ar informāciju vai ļaut citam izmantot kādu specifisku iekārtu, piemēram, krāsu printeri.

No tīkla operāciju sistēmām pašlaik populāras ir *Novell Netware*, *Windows NT* un *Unix*. *Novell Netware* ir ātrākā failu servera sistēma, vēsturiski tā ir ļoti

populāra. *Novell* instalācija un apkope ir samērā sarežģīta, un tam nepieciešama speciālista palīdzība. *Windows NT* ir jaunāka tīkla operāciju sistēma.



Tīkla administrators vienkārši regulē pieejas autorizāciju dažādaī informācijai ar paroli palīdzību.

Pamatadministrēšanu var veikt samērā vienkārši caur ierasto *Windows* grafisko interfeisu, taču sarežģītāku operāciju gadījumā vienalga nepieciešama kvalificēta profesionāla palīdzība. *Unix* kā operāciju un tai skaitā tīkla operāciju sistēma ir pati vecākā. *Unix* darbojas dažādos datoru tipos, līdz ar to tā ir populāra vide, kur nepieciešams kāds jaudīgāks dators, piemēram, ar *RISC* procesoru. Starp visiem tīkla operāciju sistēmu ražotājiem notiek nikna konkurences cīņa. Katra no sistēmām cenšas ietvert sevī citu sistēmu labās īpašības.

Plaša mēroga tīkli

Par plaša mēroga tīklu (*Wide Area Network WAN*) sāk runāt, kad tīkla ietvaros ir attālināta komunikācija, vai arī gadījumos, ja, saslēdzot daudzus datorus tīklā, tiek izmantota kāda specifiska aparatūra. Tehniski WAN ir kopā ar tīkla aparatūru saslēgti LAN. Lai saslēgtu divus LAN kopā, būs nepieciešami *Router*.

Globālo tīklu gadījumā mēs savu firmas lokālo tīklu slēdzam klāt kādai globālo tīklu nodrošināšanas kompānijai. Parasti tas notiek pa telefona līniju. Ja mēs no telekompānijas nopērkam izdalīto līniju, tad mums ir pastāvīga pieeja globālam tīklam.

Šajā gadījumā ir mazākas telekomunikāciju izmaksas, un arī mēs varam izmantot ātrākas datu apmaiņas iekārtas. Ja mums tīkls vajadzīgs tikai pa retam, tad mēs varam izmantot iezvanīšanos. Mazas informācijas apmaiņas gadījumā tas būs ļoti lēti un vienkārši..

Jebkurā gadījumā, ja mēs izmantojam telefonu līnijas, ir nepieciešams modēms. Modēmu ātrdarbība ir no 2400Kbps līdz 56000Kbps, kas ir šodienas modēmu standarts. Ātrāki modēmu sakari prasa vēl citu aparatūru un labākas sakaru līnijas.

Viselementārākais WAN ir viena datora iezvanīšanās birojā. Tīkla dators tiek pieslēgts ar modēmu pie telefonlīnijas, un uz datora tiek palaista dial-up programmatūra. No jebkuras vietas var piezvanīt uz šo datoru ar citu datoru, kam, protams, ir modēms un attiecīgā programmatūra, tad, ja zina paroli, var piekļūt jūsu biroja lokālajam tīklam. Vienīgā atšķirība no darba lokālajā tīklā birojā būs ātrdarbība, ko nosaka telekomunikāciju kvalitāte un modēmu ātrdarbība.



Ja mums tīkls vajadzīgs tikai pa retam, tad mēs varam izmantot iezvanīšanos

Gadījumā, ja mums nepieciešams savienot divus firmas biļorus, izdevīgāk ņemt izdalīto līniju. Tad var uzlikt divus ātrdarbīgus modēmus un nodrošināt sakarus ar ātrumu 128 Kbps un pat vēl ātrākus. Divu lokālo tīklu savienošanas gadījumā nepieciešama iekārta *Router*. *Router* var būt kā atsevišķa iekārta vai kā programma uz

speciāli tam izdalītas mašīnas vai servera. Savienojot divus lokālos tīklus, mēs radām lietotājiem ilūziju, ka mums ir viens liels lokālais tīkls, protams, ar ātrdarbības ierobežojumiem.

Gadījumā, ja mūsu firmas biroji atrodas dažādās valstīs, iegūt izdalītās līnijas būs ļoti grūti un dārgi. Ja mūsu tīklam jāstrādā ļoti ātri, tad vērts izmantot pasaules X.25 tīklu. X.25 tīklam ir pieslēguma punkti visās pasaules lielākajās pilsētās. Jūsu birojiem ir jānodibina kontakti ar lokālo X.25 mezglu, un tālāk jūsu savienojumu nodrošinās X.25 kompānijas. Šis ir ļoti labs risinājums reālam ātrdarbīgam plaša mēroga firmas iekšējam tīklam. Ja ātrdarbība un savienojuma nepārtrauktība nav tik svarīga, par savienotāju var izmantot Internet. Izmantojot Internetu kā firmas iekšējā tīkla nesēju, jādomā par informācijas drošību. Visa informācija ir jāšifrē, jo Internetā nešifrētu informāciju var izlasīt jebkurš. Internets neapšaubāmi ir labākais globālais tīkls, lai publicētu firmas informāciju plašai publikai. Pašlaik pasaulē notiek lavīnveidīga Internetizācija. Internets ļoti īsā laikā kļūs par svarīgu masu un privātu komunikācijas līdzekli un ienāks katra cilvēka dzīvē, spēcīgi to izmainot.

Dažas smalkākas lietas

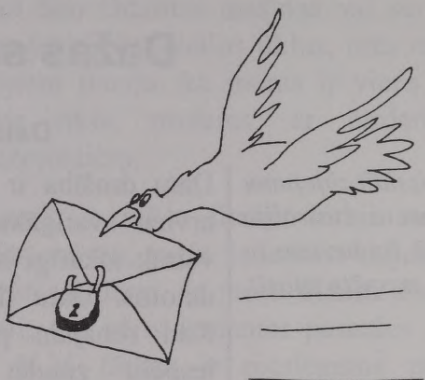
Datu drošība un kriptogrāfija

*Kā aizsūtīt ziņojumu
šefam uz Austrāliju
tā, lai neviens to
nespētu izlasīt.*

Datu drošība ir liela problēma, kas kļūst arvien svarīgāka, jo mēs kļūstam arvien vairāk atkarīgi no datiem, kas tiek glabāti datoros. Datu drošībai ir daudz aspektu. Kad runājam par diskkiem, jādomā par iespēju zaudēt informāciju aparātūras kļūmes dēļ, runājot par vīrusiem, jāuzmanās, lai tie nesabojā mūsu datus, par Internet jādomā, kā nosargāt pieeju mūsu datiem. Datu drošība ir jāņem vērā, domājot par jebkuru datoru pasaules sastāvdaļu.

Šoreiz pievērsīsim uzmanību tam, kā noglabāt konfidenciālu informāciju no liekām acīm. Mēs gribam, lai mūsu datoru komunikācija noritētu droši un nekādas liekas acis neredzētu mūsu privāto informāciju, biznesa plānus un atskaites. Bieži ir izdevīgi datu pārraides sistēmai izmantot vidi, kas pieejama daudziem, piemēram, Internet. Tāpat mums ir dokumenti, kurus vēlamies aizsargāt no liekām acīm, bet tomēr tie jāglabā uz biroja servera, jo ar tiem strādā vairāki cilvēki vienlaicīgi.

Ja esam nopietni
nobažījušies par
drošību kādā sfērā,
mēs noteikti
painteresēsīsimies par
mūsu pieņemto
drošības pasākumu
efektivitāti.
Ja durvju drošumu
katrs spēs kaut cik
objektīvi novērtēt
pēc lietotā materiāla
un tā biezuma, tad
ar datu šifrēšanu ir
sarežģītāk.



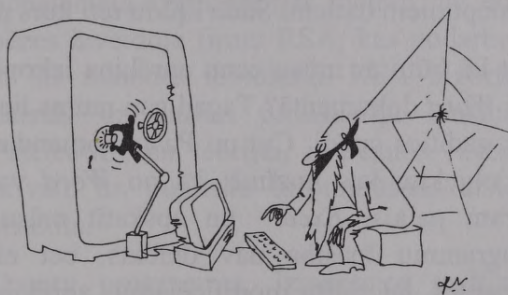
J.V.

Mēs gribam, lai mūsu datoru komunikācija noritētu droši
un nekādas liekas acis neredzētu mūsu privāto
informāciju.

Standarta iespēja, ko piedāvā dažādas sistēmas, ir datu aizsargāšana ar parolēm. Ar paroli var aizsargāt *Word*, *Excel* dokumentus vai *Microsoft Exchange* direktorijas, vai *Lotus Notes* datu bāzes.

Cik drošs ir šis mehānisms? Vienkāršākās sistēmās parole ir tikai parametrs, ko sistēma izmantos un mūs bez paroles pie datiem nelaidīs, bet apkārtceļi var būt visai primitīvi. Ar kādu no disku apskates programmām, piemēram, *Norton Commander*, varam apskatīt datus. Vārtiņu

atslēga var būt ļoti smalka, taču žodziņš var izrādīties smieklīgi zems.



Seifa ražotāja un kramplauža cīkstēšanās vērojama arī datoru sfērā.

Mazliet smalkāks mehānisms ir datu šifrēšana, izmantojot paroli, kas nodrošina, ka bez paroles zināšanas datus nevar atšifrēt un pati parole nekur datus netiek uzglabāta.

Tā mēs esam nonākuši līdz datu šifrēšanai vai kriptogrāfijai. Mēs varam izmantot dažādus kriptogrāfijas paņēmienus, tomēr vienmēr pastāvēs iespēja, ka kāds nozags mūsu noslēpumus. Seifa ražotāja un kramplauža cīkstēšanās vērojama arī datoru sfērā.

Sistēmu drošības un citu īpašību neizpratne var novest mūs līdz visai kuriozām, vai pat izmisīgām situācijām.

Piemēram, *Word* un *Excel* ļauj aizsargāt failus pret rakstīšanu un modificēšanu. Nevajag iedomāties, ka mēs varam izsūtīt cenu lapas saviem klientiem un noslēpt cenu aprēķina mehānismu. Viņiem ir iespēja noglabāt failu ar citu vārdu un tikt pie visiem noslēptajiem datiem. Šādu kļūdu reti kurš pieļaus.

Bet kā būtu ar mūsu cenu aprēķina iekopēšanu e-pasta sūtījumā vai *Word* dokumentā? Tagad gan mums liekas, ka esam droši, bet var gadīties citādi. Cut un Paste komandas mūsu aprēķinu ieliks kā objektu, kas nozīmē, ka no *Word* vai e-pasta programmas varam palaist *Excel* un apskatīt mūsu aprēķinu pilnībā. Šī programmu īpašība nav defekts, bet efekts, ko mēs varam izmantot, lai sūtītu modificējamus aprēķinus. Ja vēlamies izsūtīt tikai redzamo aprēķinu, jāizmanto *Paste special* komanda un aprēķins jāiekopē kā teksts.

Mūsu šausmu stāsts tuvojas kulminācijai. Kā būtu ar konfidencialas informācijas izdzēšanu no faila, pirms to kādam sūtam? Izdzēstam jābūt izdzēstam, taču jāņem vērā, ka gan *Word*, gan *Excel*, gan citu programmu faili vismaz mazliet atdarina failu sistēmu. Un tas nozīmē, ka kāda teksta gabala izdzēšana nozīmē tikai tā noslēpšanu. Kā gan savādāk būtu iespējama komanda *Undo* vai versiju kontrole? Uzmanību, informācijas izdzēšana no faila nenozīmē tās momentānu fizisku izdzēšanu. Gudrs programmētājs bieži spēs to atjaunot un bieži vien to varbūt varēs izdarīt ar prastu *Undo* komandu vai ko citu visiem saprotamu. Klīst baumas, ka kāda banka šādā veidā izplatīja savu klientu informāciju.

Šādas nianSES jāsaprot, ja mūsu rīcībā ir konfidenciala informācija. Visas šifrēšanas metodes būs bezjēdzīgas, ja paši rūpīgi glabātos datus pēc tam izdalīsim.

Populārākā metode, kas ir daudzu protokolu standarts un daudz kur vēl tiek ieviesta, ir RSA algoritms. Amerikāņu matemātiķi Rivests, Šamirs un Adlemanis izgudroja un nopatentēja kriptogrāfijas metodi, uz kuras bāzes izveidoja firmu RSA, kas nodarbojas ar šīs metodes ieviešanu. RSA algoritms balstās uz dažām pirmskaitļu īpašībām. Pateicoties šai teorijai, iespējams izveidot privāto un publisko kriptogrāfijas atslēgu sistēmu.

*Ja zaglis nespēj
uzlauzt durvis, tas
var mēģināt viltot
atslēgu.*

Datoru programma, izmantojot gadījuma datus, piemēram, pašreizējo laiku, jūsu iedotus datus un gadījuma skaitļa ģenerators izveidotus skaitļus, izveido privātās un publiskās atslēgas komplektu. Starp citu, tā ir ļoti svarīga lieta, jo nav jēgas ielikt automobili ļoti dārgu un sarežģītu signalizāciju, ja visām šī veida signalizācijām ir tikai 10 dažādas tālvadības pultis, zaglim jāsakrāj tikai komplekts, lai atvērtu jebkuru signalizāciju. Atslēgu izveides procesam jābūt unikālam un neatkārtojamam.

*Publiskā un privātā
atslēga patiesībā ir
divas ciparu virknes,
ko izmanto RSA
kriptogrāfijas
algoritms.*

Teorija liecina, ka nav pietiekoši labas metodes, kā no publiskās atslēgas reālā laikā atrast privāto atslēgu. Precīzāk, neviens līdz šim nav atradis šādu metodi. Nereāls laiks nozīmē spēcīgāko pasaules datoru izmantošanu gadu desmitiem, lai

*Durvis mēs gan
aizslēdzam, gan
atslēdzam ar vienu
un to pašu atslēgu.
RSA metode paredz
vienu atslēgu
aizslēgšanai, otrā
atslēgšanai, turklāt
vienreiz var slēgt ar
vienu otrreiz ar otru
atslēgu, taču atslēgt
var tikai ar pretējo
atslēgu.*

*Vai tiešām iespējams
neviltojams
elektroniskais
paraksts? Mūsu
paraksta datorizēts
attēls droši nederēs.
Bet kā būs ar RSA
metodi?*

“atlauztu” pietiekami garu atslēgu. Tas nozīmē, ka šādu iespēju var neņemt vērā. Toties šifrēšanas process notiek ātri, pat nemanāmi, izmantojot personālo datoru.

Kā mēs to varam izmantot? Jau pēc nosaukuma varam spriest, ka publiskā atslēga ir paredzēta publicēšanai. Mēs izsūtām savu publisko atslēgu visiem saviem korespondentiem, vai pat nopublicējam Internetā. Toties privāto atslēgu mēs glabājam. Fiziskā līmenī to turam uz mums vien pieejama datora vai disketes un aizsargājam ar parolēm atslēgas lietošanas procesu. Privātā un publiskā atslēga kalpo kā viens atslēgu komplekts, ko nošifrē ar privāto atslēgu, bet to var atšifrēt ar publisko atslēgu, un otrādi.

Ja mēs gribam kādai auditorijai vai visai pasaulei izsūtīt elektronisku uzsaukumu, bet gribam nodrošināties, ka ne viens nespēs izmainīt oriģinālo paziņojuma tekstu, mēs šifrējam paziņojumu ar savu privāto atslēgu. Paziņojumu var izlasīt visi, jo publiskā atslēga nav noslēpums, bet viltot mūsu paziņojumu nevar neviens, jo privātā atslēga ir tikai mums.

Protams, visi, kas var izlasīt, var arī nokopēt mūsu paziņojumu, bet neviens nevar izdarīt izmaiņas mūsu izsūtītajā

oriģinālā. Pēc būtības esam ieguvuši elektronisko parakstu.

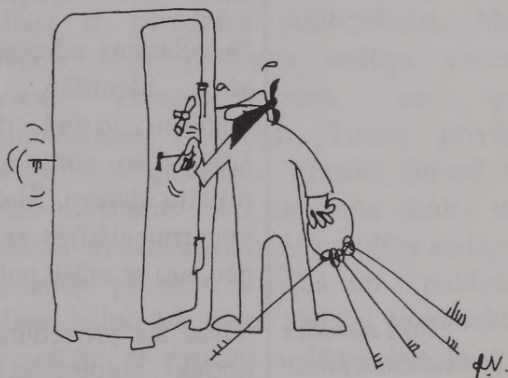
Savukārt, ja vēlamies kādam konfidenciali izsūtīt paziņojumu, tad nodrošināsimies, lai paziņojumu izlasītu tikai pareizais adresāts, nošifrējot ar viņa publisko atslēgu.

Ja vēlamies adresātam radīt pārliecību par savu identitāti, tad vispirms (labāk vispirms, jo tad citi pat neuzzina to, kurš šifrējis šo sūtījumu) nošifrējam ar savu privāto atslēgu. Šinī gadījumā mūsu draugs vispirms atšifrēs ar savu privāto atslēgu un pēc tam ar mūsu publisko.

*RSA metodes
izmantošana paver
plašas iespējas
elektroniskai datu
apmaiņai,
elektroniskai
tirdzniecībai un
tamlīdzīgi.
Tāpat šo metodi var
izmantot arī
starptautiskas
noziedznieku
organizācijas.*

Visas šīs procedūras dažādās programmās notiek automātiski un pat nemanāmi. Protams, vispirms ir jāizveido atslēgu komplekts un jāapmainās ar tā publisko daļu ar citiem korespondentiem. Izmantojot pietiekami garas atslēgas, tās ir neuzlaužamas rupjā veidā. Interesanti, ka ASV valdība cenšas uzlikt ierobežojumus šāda veida šifrēšanai. Viņi grib, lai jebkura veida publiskas komunikācijas būtu iespējams noklausīties, ja ir izdota prokurora sankcijas. Arī Lotus Notes starptautiskai versijai ir īsāka atslēga nekā ASV versijai, kas ļauj to uzlauzt reālā laikā ar superdatora palīdzību. Taču šim progresam matemātikā neko nepadarīsi,

katrs programmētājs iesācējs var izveidot programmu pēc mācību grāmatu algoritma, kas šifrēs ar pietiekoši garu atslēgu, lai neviena supervalsts ar superdatoru neatšifrētu viņa slepeno korespondenci.



Izmantojot pietiekami garas atslēgas, tās ir neuzlaužamas rupjā veidā.

Cerams, ka lielvalstu valdības visā drīzumā tiks galā ar apmulsumu, ko rada šīs metodes, un tiks ieviesti standarti šifrēšanā, kas pavērs iespēju izmantot elektroniskos norēķinus un dokumentus.

Protams, ja runājam par šāda veida drošību, tad jāatceras, ka mašīnas bieži tiek nozagtas tādēļ, ka tām ir atstāta atslēga aizdedzē, un nauda no seifa tiek izzagta tamdēļ, ka tas ir bija atstāts neaizslēgts.

Publiskā un privātā atslēga ir garas skaitļu virknes, kas jāglabā uz datora vai disketes. Kādā veidā aizsargāt datoru un disketi? Pirmkārt, fiziski, otrkārt, ar vienkāršāku paroļu mehānismu. Ja mūsu parole ir personīgais dzimšanas datums vai iemīļotā mašīnu marka, tad nevajag iedomāties, ka esam ļoti oriģināli.

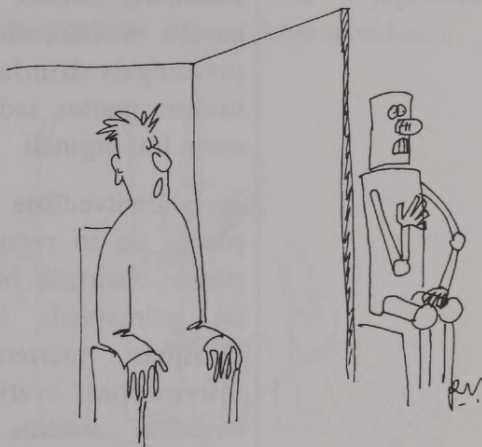
Ja grāmatvedības datoram būs sarežģīta parole un to regulāri mainīs, tad aktuālā parole visdrīzāk būs pielīmēta pie ekrāna, lai grāmatvede to varētu ērti ievadīt. Drošības mēriem jābūt atbilstošiem informācijas svarīgumam un slepenībai, savādāk sistēma degradēsies. Jebkuri drošības mēri ir pretrunā ērtam darbam, tie sarežģī procedūras, un rezultātā par tiem ir šādā vai tādā veidā jāmaksā, tādēļ pirms tos ievieš, jāpadomā, vai tie nepieciešami.

Mākslīgais intelekts, ekspertsistēmas un ...

Ko tad īsti saprot ar vārdiem “mākslīgais intelekts”? Klasisks ir Tjuringa tests (1950), kuru īsumā varam aprakstīt šādi. “Ja kādā no apkārtējās vides izolētā istabā nosēdina cilvēku un caur datoru ļauj tam sazināties ar citā istabā esošu datoru vai cilvēku, tad, ja šis cilvēks sazināšanās procesā nespēj

Tā nu ir sanācis, ka līdz ar datoru parādīšanos cilvēkam ir radusies iespēja realizēt savu gadsimtiem loloto sapni radīt rīkus, kas varētu domāt viņa vietā. Un tā cilvēks ar lielu entuziasmu ķērās pie mākslīgā intelekta radīšanas uz datora bāzes.

atšķirt cilvēku no datora, saka, ka uz datora uzstādītā sistēma (programmu kopums) ir mākslīgais intelekts.”



Klasisks ir Tjuringa tests

Citiem vārdiem sakot, mākslīgais intelekts nedrīkst atšķirties no sava dabiskā prototipa, kaut gan šajā vietā varētu diskutēt par to, kādu dabisko intelektu ņemt par etalonu, jo dažādu cilvēku intelekts var stipri vien atšķirties.

Vienkārši runājot, par mākslīgo intelektu parasti sauc programmas, kas dara lietas, par kurām cilvēki iedomājas, ka tie ir vienīgie uz tām spējīgie. Runas un attēlu atpazīšanas sistēmas, runas sintēze, orientēšanās telpā, konteksta veidošana utt., tiek uzskatīts par mākslīgo intelektu. Tiesa

gan, kad cilvēki pierod pie tā, ko dara mašīnas, to vairs nesauc par mākslīgo intelektu. Gadsimta sākumā jebkurš kalkulators varētu būt mākslīgais intelekts, šodien burtu atpazīšana vairs nav nekas.



...mākslīgais intelekts nedrīkst atšķirties no sava dabiskā prototipa...

Vai šāds mākslīgais intelekts vispār ir vajadzīgs? Droši vien ir, ja mēs vēlamies, lai kādreiz mūs apkalpotu tādi roboti, kādi redzami labākajās zinātniski fantastiskajās filmās. No otras puses, progress ir process pats par sevi, ja kas ir izgudrojams, tad to kāds izdomās, gribam mēs to vai ne. Nākotnē mēs sastapsim arvien saprātīgākas un gudrākas mašīnas.



Kas ko vada, vai prāts ir ķermeņa izpausme, vai arī prāts vada ķermeni?

Bet vai cilvēks var mākslīgi radīt, citiem vārdiem sakot, programmēt savam intelektam līdzīgu mākslīgu analogu? Pirms vairākiem gadiem daudzi jo daudzi zinātnieki, programmētāji apgalvoja, ka tūlīt jau būs, taču šodien lielai daļai ir nolaidušās rokas. Mākslīgā intelekta nozarē ir daži lieli strīdus akmeņi, kas, iespējams, pat teorētiski nav atrisināmi.

Pirmais ir filozofiskais jautājums par prātu un ķermeni. Kas ko vada, vai prāts ir ķermeņa izpausme, vai arī prāts vada ķermeni (*Mind-Body Problem*)? Šeit jau no filozofijas vēstures mēs sastopamies ar

ideālistu, materiālistu un duālistu teorijām. Principā nav zinātniskas atbildes uz šiem jautājumiem, un mūsu nostāja šeit var būt drīzāk reliģiska nekā jebkāda cita.

Paši zinātnieki savā starpā strīdas par dažādām nostādnēm MI nozarē. Mēs varam mēģināt pieiet MI no divām dažādām pusēm. Mēģināt imitēt šauru un dziļu daļu kādai sarežģītai cilvēka zināšanu sfērai, kā to dara simboliskā MI lauka kopēji, vai arī atrast, ar kādiem primitīviem mehānismiem, tos masīvi paralēli darbinot, kādus efektus no dzīvās dabas un saprāta mēs varam imitēt. Pirmā pieeja ir sintezēšana no augšas, otra — no apakšas. Eksperta sistēmas ir tipisks simboliskais MI, kuras arī ir visplašāk zināmais pielietojamais MI piemērs. No otras nometnes visvairāk zināmie ir pētījumi par neironu tīkliem.

No mākslīgā intelekta nozares attīstījusies un kā patstāvīga daļa ir kļuvusi zinātne par ekspertsisēmām. Ekspertsisēmās ir zināšanu bāzes, kas darbojas līdzīgi pieredzējušam profesionālim ekspertam. Cilvēkam ekspertam, protams, piemīt gadiem krātā pieredze, kā arī spējas rīkoties nestandarta situācijās. Nestandarta situācijas ekspertsisēma nespēs apstrādāt, toties tajās var apvienot tūkstoš ekspertu zināšanas. Turklāt atšķirībā no cilvēka eksperta

ekspertsistēmas palīdzību var saņemt daudz lietotāju vienlaicīgi.

Ekspertsistēmas patiešām tiek pielietotas. Tā, piemēram, ir sastopamas medicīniskās ekspertsistēmas. Iedomājieties, kā jūs, sajūtot slimības pirmos simptomus, iedarbināt ekspertsistēmu, ievadiet tai šos simptomus un beigās dabūjat zināt, kas jums kaiš (te gan nevajag aizrauties, atceroties Džeroma K. Džeroma varoņa bēdīgi slaveno pieredzi).



Nestandarta situācijas ekspertsistēma nespēs apstrādāt, toties tajās var apvienot tūkstoš ekspertu zināšanas.

Nopietni runājot, ekspertsistēmas patiešām ar lieliem panākumiem var pielietot dažādās nozarēs, piemēram, ģeoloģiskos pētījumos un meklējumos, telefonu tīkla iespējamo bojājumu uzmeklēšanā.

Kā tad īsti darbojas ekspertsistēmas? Šķiet, ka nav nemaz tik grūti: apkopo zināšanas, tad saņem no lietotāja informāciju un

sniedz atbildi. Taču diemžēl tas nemaz nav tik vienkārši. Pirmkārt, zināšanu apkopošana ir milzīgs darbs. Turklāt ir jāatlasa vajadzīgās zināšanas, atsijājot visu lieko un nevajadzīgo. Bet arī tas vēl nav viss, jo ekspertsistēmai ir jāprot pareizi iztaujāt lietotājs, saņemot **visu** vajadzīgo informāciju un neievācot **nekā** lieka. Un visbeidzot, apstrādājot visu no lietotāja saņemto informāciju, pamatojoties uz apkopotajām zināšanām, ir jāizdara secinājumi un jāpaziņo tie lietotājam saprotamā formā.

Varbūt vienkārši aplūkosim kādu piemēru. Mēģināsim izveidot ekspertsistēmu, kas noteiktu, vai aprakstītā dzīvā būtne ir putns, zivs vai dzīvnieks. Šim nolūkam izveidosim stipri vienkāršotu zināšanu bāzi.

Zivs:

<i>Vairošanās</i>	Ikri
<i>Segums</i>	Zvīņas
<i>Ekstremitātes</i>	Spuras
<i>Elpošana</i>	Žaunas

Putns:

<i>Vairošanās</i>	Olas
<i>Segums</i>	Spalvas
<i>Ekstremitātes</i>	Divi spārni, divas kājas
<i>Elpošana</i>	Plaušas

Dzīvnieks:

<i>Vairošanās</i>	Dzīvi
<i>Segums</i>	Spalvas
<i>Ekstremitātes</i>	Četras kājas
<i>Elpošana</i>	Plaušas

Kā redzat, šī zināšanu bāze ir ļoti primitīva, bet demonstrācijai tā būs laba diezgan.

Tagad ir jāizlemj, kādus jautājumus uzdot. Acīm redzams, ka nepietiek pajautāt, kā norit elpošana, jo gan putni, gan dzīvnieki elpo ar plaušām. Tāpēc vislabāk jautājumos iekļausim visus iespējamus laukus, jo mūsu piemērā šo lauku nemaz nav tik daudz:

Kā norit vairošanās?

Kas klāj būtni?

Kādas un cik ekstremitātes tai ir?

Kā tā elpo?

Kad lietotājs uz šiem jautājumiem ir atbildējis, varam izdarīt secinājumus:

Ja dzīvā būtne vairojas ar ikriem, ir klāta ar zvīnām, tai ir spuras un žaunas, tad tā ir zivs.

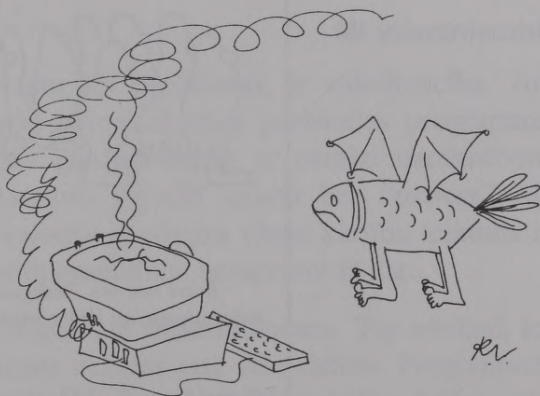
Ja dzīvā būtne vairojas ar olām, ir klāta ar spalvām, tai ir spārni, kājas un plaušas, tad tas ir putns.

Ja dzīvā būtne dzimst dzīva, ir klāta ar spalvām, tai ir kājas un plaušas, tad tas ir dzīvnieks.

Sarežģītas ekspertu sistēmas ir praktiski vienīgais, kas no MI lauka dzīvo ārpus laboratorijām un "mīlētāju" datoriem.

Lielākie pārmetumi simboliskā MI lauciņa kopējiem ir saistīti ar to, ka tie nemaz neinteresējās par intelektu, ka tas, ko viņi rada, patiesībā ir tikai smalkākas datu bāzu sistēmas. Šīs sistēmas spēj veikli vai mazāk veikli operēt ar mīlīgajiem datu apjomiem, bet nevar runāt ne par kādu patiesu izpratni

vai zināšanām. Protams, ka šādas ekspertsistēmas var radīt arī visai kuriozas atbildes. Šeit nav jāsatraucās par intelektu, kas izspiedīs cilvēku no balto apkaklīšu darba vietām. Kaut gan ir nopietni zinātnieki kas domā, ka intelektu var sasniegt ar vienkāršu simbolu manipulāciju.



Protams, ka šādas ekspertsistēmas var radīt arī visai kuriozas atbildes.

Otrs virziens, sintēze no apakšas, visplašāk pazīstams no pētījumiem par neironu tīkliem. Neironu tīkli imitē it kā nervu sistēmu, kur ļoti daudz neironu ir viens ar otru savienoti un viens otram sūta signālus. Katrs neirons “iedegas”, saņemis sev pietiekoši lielu kairinājumu, un izsūta kairinājumu blakus esošiem neironiem. Tā šis kairinājums izplatās no ievadneironiem līdz izvadneironiem šādu sarežģītu sistēmu

neprogrammē, bet apmāca ar apbalvojumu un sodīšanas metodi. Sistēmas var lieliski atpazīt burtus, no punktu matricas pārvēršot tos simboliskā formā. Pastāv uzskats, ka šādos neironu tīklos var spontāni rasties augstākas koncepcijas.



Dr.

Katrs neirons "iedegas", saņemis sev pietiekoši lielu kairinājumu, un izsūta kairinājumu uz sev blakus esošiem neironiem.

Viens no šī MI novirziena trūkumiem ir nespēja demonstrēt rezultātus, kas būtu saprotami nespeciālistiem. Pagaidām tās ir tikai programmas, kas imitē kādu mazu cilvēka domāšanas procesa daļu. Virtuāli "intelektiņi", kas dzīvo laboratoriju datoros.

Paralēli un daudzkārtīgi atkārtoti algoritmi uzrāda interesantas īpašības, par piemēru var minēt kaut vai fraktālus, kurus daudzi ir redzējuši. Primitīvas funkcijas, daudzkārtīgi atkārtotas, pašas uz sava rezultāta (rekursīvi) sāk veidot apbrīnojami sarežģītas struktūras. Nākotnē šajā virzienā

noteikti būs ļoti interesanti atklājumi. Jāpiekrīt dažiem zinātniekiem, kas izsaka nožēlu, ka ASV kongress akceptē milzu naudu dzīvības pēdu atrašanai uz Marsa, nevis MI pētījumiem. Pēdējie varētu būt daudz perspektīvāki arī militārā jomā.

MI valodniecībā

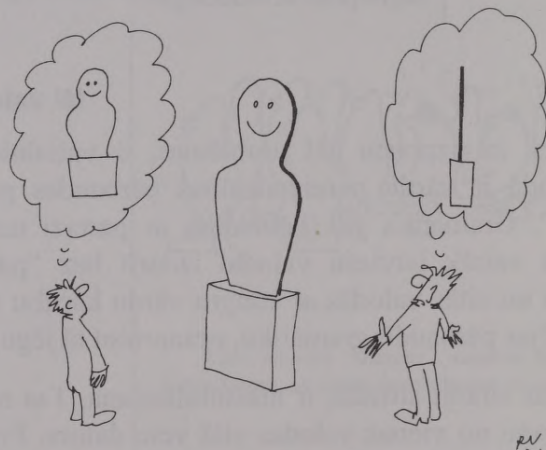
Labs piemērs, lai izprastu MI problēmas, ir valodniecība. Arī latviešu valodā ir vārdu pareizrakstības pārbaudes programma "Gramatiķis". Gramatika gan nebeidzas ar pareizi uzrakstītiem vārdiem, bet vairāk latviešu valodai izdarīt bez "patiesa" MI nevar. Angļu un citās valodās ar stingru vārdu kārtību teikumā ir lielākas iespējas pārbaudīt gramatiku, nesaprotot tā jēgu.

Cita joma, kas strauji attīstās, ir mašīntulkošana. Tas nozīmē, ka teksta tulkošanu no vienas valodas citā veic dators. Programmēt tulkošanas procesu ir ļoti grūti, it īpaši tādām valodām kā latviešu valoda, kurai ir raksturīga diezgan brīva vārdu kārtība teikumā. Piemēram, vācu vai zviedru valodā, kur teikumā otrajā vietā vienmēr atrodas darbības vārds, teikuma locekļu atpazīšanas process ir vienkāršāks, bet arī tur nenoliedzami ir savas problēmas.

Mašīntulkošanas procesa sarežģītība ir tā, ka dators nesaprot vārda kontekstu. Ja, piemēram, cilvēks saka:

"Esmu izstrādājis darbības plānu" vai: *"Man šķiet, ka tā maizes šķēle ir par plānu,"*

mēs ļoti labi saprotam, ka vienā gadījumā runa ir par veicamajiem pasākumiem, bet otrā — par biežumu. Taču datoram abi šie gadījumi ir absolūti vienādi. Vēl jāņem vērā, ka mēs esam spējīgi uz lietām skatīties no dažādiem skatu punktiem, saskatīt lietu dažādās izpausmes...



Vēl jāņem vērā, ka mēs esam spējīgi uz lietām skatīties no dažādiem skatu punktiem...

Tāpēc tulkojot var sanākt dažādas nepatīkamas lietas. Tas nozīmē, ka programmētājam ir jāpasaka, kā datoram jārikojas katrā atsevišķā gadījumā. Kā to var izdarīt? Pastāv dažādi varianti. Var, piemēram, noteikt vārdu šķiras, teikuma locekļus utt. Var izmantot statistikas datus par to, kādi vārdu savienojumi teikumā figurē biežāk.

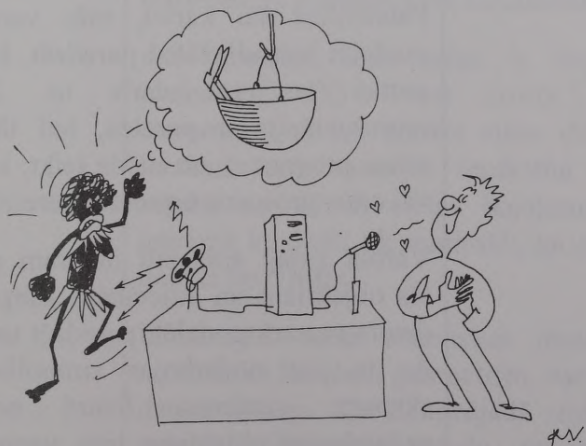
Konkrētajā gadījumā vārdu savienojums “darbība-plāns”, kur “plāns” ir pasākumu kopums, būs sastopams daudz biežāk nekā “darbība-plāns”, kur “plāns” nozīmē biežumu. Taču arī statistika ne vienmēr var atrisināt problēmu. Piemēram, teikumā:

“Basketbola spēles laikā uzsprāga nezināmu personu uzstādīta bumba”,

tulkojot vārdu “bumba”, to var interpretēt gan kā spēlei domātu rīku, gan kā spridzekli (kā tas arī ir šajā gadījumā). Skatoties uz statistikas datiem, vārdu savienojums “basketbols-bumba”, kur bumba ir spēles elements, nenoliedzami būs krietni vien biežāk sastopams nekā spridzeklis. Līdz ar to tulkojums būs aplams.

Jā, un tā kā datoram absolūti trūkt humora izjūtas, arī jokus un anekdotes, īpaši, ja tiek izmantotas vārdu spēles, tas nespēs pārtulkot. Ja vien līdz ar mākslīgo intelektu netiks ieprogrammēta arī humora izjūta.

Pašreizējā mašīntulkošana ir jāuztver kā tulka darbarīks, kas bez cilvēka autonomi nevar darboties. Savādāk no pārpratumiem neizbēgt.



Savādāk no pārpratumiem neizbēgt.

Īsi sakot, īsta mašīntulkošana un gramatikas pārbaude būs tad, kad datoram būs sajēga, par ko teksts rakstīts, nekāda statistika un citi triki neglābs. Pašlaik dators apguvis tikai sintaksi-formu, kad tas sapratīs semantiku-saturu, tad tās lietas sāksies.

Ko nozīmē saprast, apzināties utt., to neviens nezina, mēs atgriežamies pie filozofēšanas. Tiesa, katrs, kurš mazliet padomājis par paša domāšanas procesiem, būs piefiksējis, ka mēs galvās turam pašu subjektīvo pasauli.

Ja esam materiālisti, tad domājam, ka mūsu subjektīvā pasaule ir attēls, karte, reprezentācija ārējai objektīvajai pasaulei (subjektīvie ideālisti domā otrādi). Pateicoties šai kartei, mēs varam galvā modelēt pasauli, tādā paredzēt, kas būs, ja notiks tas vai darīs to. Ja mūsu reprezentācija ir pareiza, tad tāda ir arī mūsu prognoze, un mēdz teikt, ka mēs tos vai citus procesus saprotam pareizi.

Varbūt vajag aprakstīt datoram pasauli ar tās objektiem un attiecībām starp tiem, lai dators var arī modelēt, paredzēt un saprast. Ar to jau nodarbojas simboliskais MI, ekspertu sistēmas. Šaurā nozarē ar nedaudziem objektiem mēs varam pasauli aprakstīt, bet reālā pasaule ir ļoti sarežģīta.

Šeit jāmin arī ir cits viedoklis no otrās MI

nometnes, kur valda uzskats, ka zināšanas par pasauli var ierakstīt procedurāli, ka patiesam MI nemaz nevajag reprezentācijas, ka tās pašas spontāni veidojas un mainās dzīves laikā.

Medicīna ir pierādījusi, ka cilvēka smadzenes bieži uzvedas līdzīgi hologrammai, kur katrā bides punktā ir ietverts viss attēls. Sabojājot daudzus smadzeņu apgabalus, īslaicīgi pasliktinās kāda smadzeņu funkcija, kas vēlāk atjaunojas. Tas līdzinās mušai uz fotoobjektīva, visa bilde mazliet tumšāka, bet citādi nekas. Tas viss vedina uz pārdomām, ka cilvēka smadzenēs informācija glabājas kaut kā izkliedēti.

Arī tehniski tādas lietas ir iespējamās, piemēram, CD var izurbt septiņus caurumus, un tas skanēs visās vietās. CD ierakstā katra nots ir ierakstīta daudzās vietās, un vienas vietas bojājums varbūt ietekmē kvalitāti tik minimāli, ka ar ausi to nemaz nedzirdēs.

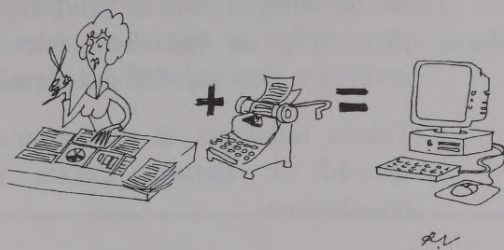
Izmantojot tādas eksotiskas matemātikas nozares kā Boolean ģeometriju, var izveidot datora atmiņu, kas daudzējādā ziņā līdzīga mūsējai. Nogriežot gabalu, tai zūd tikai statistiska kvalitāte, pārpildoties tā sāk aizmirst vecas lietas, no nepilnas

informācijas tā var mēģināt atcerēties pilno ainu.

Apskatu beidzot, jāatzīst, ka MI pētnieku vidū nav vienprātības par tā teorētiskajām iespējām. Ir daži, kas uzskata, ka cilvēka saprāts ir unikāls un principiāli neatdarināms, iespējams tikai imitēt vienkāršākās funkcijas, kā to darām jau šodien. Šajā virzienā ir iespējams progress, bet cilvēks vienmēr paliks nepārspēts.

Savukārt ir zinātnieki, kas uzskata pilnīgi pretēji. Piemēram, slavenais amerikāņu zinātnieks Minskis (Minsky) uzskata, ka cilvēki ir tikai pirmā saprāta evolūcijas pakāpe, ka mēs radīsim mašīnas, kas būs gudrākas par mums un pārņems progresa virzību no cilvēkiem. Ne pārāk glaimojoši mūsu pasaules valdnieku pasaules uztverei, bet varbūt nemaz nav tik slikti.

MI ir ļoti perspektīva un interesanta zinātnes nozare. Tiesa, pēc neveiksmīgajiem pagātnes pareģojumiem nopietni zinātnieki neizsaka prognozes. Bet mēs uzdrošināsimies teikt, ka MI datoros, mūsu darbā un atpūtā pieteiks sevi visai drīz, noteikti ātrāk, nekā mēs domājam.



Teksta redaktori

Šodien jau var teikt, ka rakstāmmašīnas pieder vakardienai. Šur tur tās, protams, vēl var atrast, taču reizēs, kad jāuzraksta vēstule, jāizveido atskaite vai kāds cits dokuments, arvien skaļāk sevi piesaka datori. Un ne jau tāpēc, ka tas būtu pēdējais modes pīkstiens, bet tādēļ, ka to priekšrocības salīdzinājumā ar rakstāmmašīnām ir acīmredzamas:

- viegli un ātri veicamas korekcijas jebkādas sarežģītības dokumentā;
- dokumentus var glabāt un jebkurā brīdī minūtes laikā izdrukāt uz papīra;
- pat liela apjoma tekstu, piemēram, grāmatu, var izvietot mazā disketē un pārnēsāt tā, ka neviens to pat nepamana;
- tekstu var glīti noformēt un papildināt ar attēliem, neizmantojot citus līdzekļus utt.

Šo uzskaitījumu varam turpināt vēl un vēl. Un, ja reiz jūsu rīcībā ir pietiekoši labs dators, tad atliek tikai izvēlēties jūsu vajadzībām piemērotu teksta redaktoru.

Teksta redaktors ir programma, kas nodrošina tekstu apstrādi ar datoru. Teksta redaktoram obligāti ir jānodrošina:

- teksta ievads no tastatūras;
- teksta izvads uz printeri;
- vismaz elementāra teksta rediģēšana;
- un, protams, tekstu uzglabāšana.

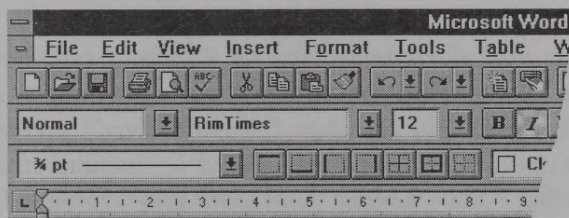
Šīs ir minimālās prasības, lai programmu varētu izmantot kā teksta redaktoru. Pašreizējo teksta redaktoru iespēju klāsts ir daudzreiz plašāks. Ar tiem var veidot ne vien nelielus dokumentus, bet arī plaša apjoma tekstus, kas papildināti ar grafiku, glītu noformējumu, automātiski ģenerējamiem paskaidrojumiem, atsauču un satura rādītājiem.

Taču, veidojot teksta failus ar kādu konkrētu redaktoru, ir jāņem vērā tas, ka formatējumi, ko ļauj veikt šis redaktors, var netikt pareizi atspoguļoti citos teksta redaktoros. Kādēļ tas tā notiek? Varētu teikt, ka katram redaktoram ir savs, lietotājam neredzams veids, kā tas pieraksta visus dokumenta noformējuma elementus, piemēram, fontus, to izmērus un efektus, atkāpes, tabulas. Šo pieraksta veidu redaktors pats lieliski "saprot", bet citi redaktori to nespēj apstrādāt.

Tāpēc ir ieteicams strādāt ar vienu un to pašu redaktoru. Principā gan visi nopietnākie teksta redaktori viens otru

samērā labi saprot, taču teksts kaut minimāli var atšķirties, bildes nobīdīties utt., un ir jāatkārto lielākā daļa noformēšanas darbu.

Piemērs



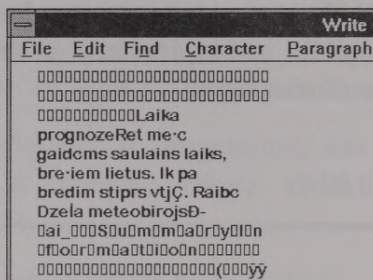
Laika prognoze

Rīt mežā gaidāms **saulains laiks**, brīžiem **lietus**.

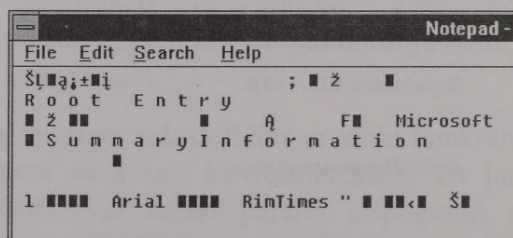
Ik pa brīdim **stiprs vējš**.

Raibā Dzeņa meteobirojs

Lūk, tā izskatās ar Microsoft Word veidots dokuments, kas atvērts tajā pašā Microsoft Word redaktorā. Dokumenta noformējums šādā gadījumā nemainās. Var mainīties tā atspoguļošanas forma, piemēram, uz jūsu datora dokuments tiek rādīts 100% izmērā, bet uz cita datora Microsoft Word programmai uzstādīts 75% samazinājums vai 120% palielinājums, respektīvi, lapa izskatās atbilstoši lielāka vai mazāka.



Šis ir tas pats dokuments, atvērts Windows redaktorā Write. Tas "saprot" dokumenta tekstu, bet nevar apstrādāt tā noformējumu, un tāpēc vietās, kur tam bija "neskaidrības", piemēram, latviešu valodas specsimboli, pēdiņas, Word iezīmējumi tika aizvietoti ar kvadrātiņu, kas nozīmē nesaprotamu simbolu. Kā redzat, neskaidrību ir daudz, un tas apgrūtina teksta uztveršanu.



Ar vienkāršo Notepad redaktoru dokuments izskatās vēl nesaprotamāks.

Vienā redaktorā izveidota faila ielasīšanu ar citu redaktoru sauc par importēšanu. Importēšana šajā

gadījumā ir tāds kā tulks, kas redaktoram “paskaidro”, kādam jāizskatās ar citu redaktoru veidotajam dokumentam.



Importēšana ir kā tulks.

Lai varētu vienkārši noteikt, ar kādu redaktoru ir veidoti faili, pietiek palūkoties uz faila paplašinājumu, piemēram, .doc, .wri. Visparastākais teksta faila paplašinājums ir .txt. Failos ar šādu paplašinājumu parasti tiek glabāts neformatēts teksts, t.i., vienkāršs teksts līdzīgi kā no rakstāmmašīnas.

Laika prognoze

Rīt mežā gaidāms saulains laiks.
Brižiem lietus.

Ik pa brīdim stiprs vējš.

Raibā Dzeņa meteobirojs

Teksta redaktoru izvēle ir pietiekoši plaša. Atkarībā no vajadzībām un datora iespējām varat izvēlēties ļoti vienkāršus redaktorus, piemēram, *Norton Editor* (DOS

vidē) vai *Notepad* (*Windows* vidē), vai tādus teksta redaktoru šedevrus kā, piemēram, *Microsoft® Word* vai *Word Perfect*.

Kāpēc ir jāņem vērā datora iespējas? Daudziem moderniem un patiešām spēcīgiem teksta redaktoriem ar milzīgām iespējām ir nepieciešami lieli atmiņas (gan diska, gan operatīvās atmiņas) resursi. Strādājot ar šādu redaktoru uz nepietiekoši spēcīga datora, no datora lietotāja varat kļūt par datora upuri, jo programma labākajā gadījumā strādās ļoti lēni. Tad jūs nepamatoti dusmosieties uz programmu un tās veidotājiem, kaut arī programma varbūt ir pat ļoti laba, ja vien ar to strādā uz piemērota datora.

Bieži vien cilvēki, dzenoties pēc modes, izmanto pārāk spēcīgus teksta redaktorus, kas strādā ļoti lēni, jo tie prasa lielus datora resursus. Vienkāršāki redaktori nodrošinātu visas nepieciešamās funkcijas un strādātu daudz ātrāk.

Viena no jaunāko teksta redaktoru labajām īpašībām ir iespēja tajos ievietot datus un rezultātus no citām programmām. Turklāt tos var ielikt "dzīvus" - grafiks vai tabula pēc pamainīšanas tabulu procesorā izmainās arī teksta redaktorā.

Teksta redaktorā var uzglabāt un organizēt informāciju, izmantojot dokumenta *Outline* skata sniegtās iespējas, kas ļauj noslēpt un atklāt teksta detaļas, piemēram, dokumenta virsrakstu hierarhiju, kas normālajā *Normal* un lappuses izklājuma *Page Layout* skatos nav redzama.

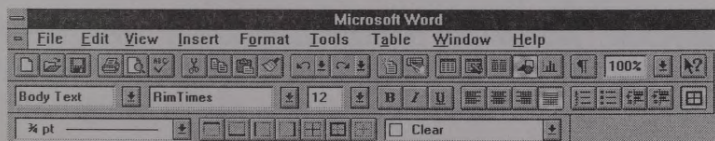
Ir svarīgi pareizi sakārtot redaktoru. Ar stilu palīdzību visiem firmas dokumentiem mēs varam piešķirt vienādu un konsekventu izskatu, kura izstrādi var uzticēt dizaineram. Firmas dokumenti būs glīti noformēti, bet darbinieki netērēs laiku domājot, kādus fontus lietot, radot briesmīgi saraibinātus un grūti lasāmus dokumentus.



... darbinieki netērēs laiku domājot, kādus fontus lietot.

Viens no populārākajiem teksta redaktoriem, ja ne pats populārākais (vismaz Latvijā), ir *Microsoft Word*. Tam analogais *WordPad* "saprot" lielu daļu *Word* vidē iespējamo formatējumu (izņemot ar *Word 2.0* veidotos dokumentus), taču ir stipri vājāks, bet līdz ar to prasa mazākus atmiņas resursus.

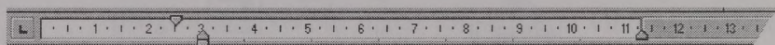
Microsoft Word lietotājam piedāvā viegli apgūstamu un ērti lietojamu rīku rindu, kas ir daļa no interfeisa un ir pazīstama katram, kurš lieto *Windows* programmas.



Word nodrošinātās darba ērtības nosaka redaktora pielietojamības plašumu.

Microsoft Word ļauj veidot dokumentus ar dažādiem fontiem un fontu parametriem, ar dažādiem rindkopu formatējumiem, ar atkāpēm, ar izcēlumiem utt.

Microsoft Word piedāvā strādāt vairākos dokumenta projektēšanas skatos: normālajā (*Normal*), izceltajā (*Outline*), lapas izvērsumā (*Page Layout*) un galvenajā dokumenta skatā.



Mīlais Trusīt!

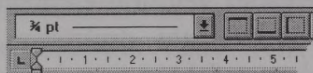
Ar šo daru Tev zināmu, ka šā gada šā mēneša piektajā dienā, skaitot no vakardienas, es svinēšu savu dzimšanas dienu. Programmā paredzētas šādas izklaides:

- izēšanās
- dejošana
- ālēšanās u.c. labumi.

Ļoti ceru, ka Tu neieradīsies.

Lapsa (Kūmiņš)

Normālais skats ir parasts dokumenta veidošanas režīms, kas domāts teksta sagatavošanai un formēšanai. **Izceltajā skatā** ir dota iespēja caurskatīt un formēt virsrakstu līmeņus. Lūk, viens dokuments, kas sastāv no triju līmeņu virsrakstiem, normālajā un izceltajā skatā:



Virsraksts

I n o d a ļ a

I n o d a ļ a s 1. a p a k š n o d a ļ a

I n o d a ļ a s 2. a p a k š n o d a ļ a

II n o d a ļ a

II n o d a ļ a s v i e n ī g ā a p a k š n o d a ļ a

BEIGAS



Virsraksts

I n o d a ļ a

I n o d a ļ a s 1. a p a k š n o d a ļ a

I n o d a ļ a s 2. a p a k š n o d a ļ a

II n o d a ļ a

II n o d a ļ a s v i e n ī g ā a p a k š n o d a ļ a

BEIGAS

Lapas izvērsuma skats paredzēts dokumenta lapu noformēšanai, bet **galvenā dokumenta skats** - liela apjoma dokumenta daļu apvienošanai vienotā formātā.

Ar to vien *Word* iespēju uzskaitījumu, protams, nevar beigt. *Word* vidē līdz ar tekstuālu informāciju var iekļaut gan parastus attēlus (zīmējumus, fotogrāfijas), gan citu programmu veidotos grafikus un elektroniskās tabulas (*Microsoft Excel*), gan specifiskas matemātiskas formulas (*Microsoft Equation*), gan datubāzes utt. Un pat vēl vairāk — dokumentu var izveidot tā, ka pamainot tabulu vai grafiku oriģinālajās programmās, arī jūsu *Word* dokumentā tas izmainīsies.

Word piedāvā iespēju automātiski veidot satura rādītājus, indeksus, atsaucis ... Vienīgais, ko vēl var piebilst: “Pamēģiniet strādāt ar *Microsoft Word*!”

Lai arī teksta redaktori galvenokārt veic ļoti elementāru funkciju – vēstuļu, teksta dokumentu drukāšanu, ir ieteicams pamatīgāk apgūt teksta redaktorus un izmantot tos pareizi. Ja dokumenti ir izveidoti pareizi, mums būs iespējams to vieglāk modificēt, pamainīt formātējumu, iedot citiem lietotājiem ievietot lielākā dokumentā.

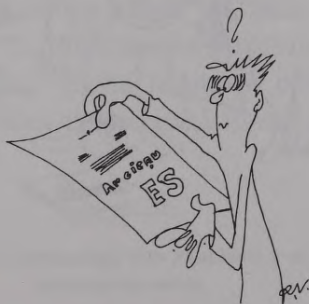
Uzmanību - PADOMS!

Tipiska iesācēju kļūda ir tukšumu izmantošana, veidojot atkāpes un tabulas. Tas nevar būt pareizi, jo izlīdzināta teksta gadījumā katrā rindīnā tukšumam ir cits platums, ko aprēķina teksta redaktors, lai burti būtu izlīdzināti gan gar labo, gan gar kreiso malu.

Atkāpes un tabulas jāveido ar *Tab* palīdzību, vai vēl pareizāk būtu izmantot rindkopu uzstādījumus stilos un tabulu veidotājus. Gadījumā, ja reiz ir iegūta ar tukšumiem veidota tabula, tai var pielietot tādu fontu, kur visi burti ir vienādā platumā, piemēram, *Courier*, un neizmanto izlīdzināšanu, citādi dažādie burtu un ciparu platumi radīs nepatīkamu skatu.

Svarīgi ir pareizi sakārtot teksta redaktoru visā uzņēmumā. Ja gribam, lai nebūtu problēmu ar dokumentu apmaiņu, jāpanāk, lai visi firmas darbinieki izmantotu vienus un tos pašus fontus, nemaz nerunājot par burtu kodēšanas sistēmām. Pretējā gadījumā regulāri tiks zaudēts laiks, mainot fontus. Ai, kā tas viss redzēts, piedzīvots!

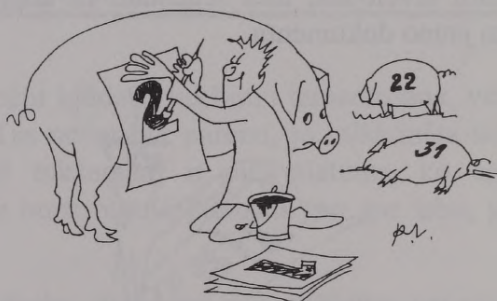
Bieži redzēts, ka viens un tas pats uzņēmums, pat viens un tas pats indivīds katru reizi rada pēc skata pilnīgi atšķirīgus dokumentus, bieži vien lietojot nepareizi sabalansētus fontus, kas rada sliktu iespaidu. Viena uzņēmuma dokumentiem, vēstulēm un līgumiem tomēr vajadzētu izskatīties vienādiem pēc izkārtojuma, stila, fontiem un dizaina elementiem. Tamdēļ teksta redaktoros ir izveidoti šabloni un stili. Piemēram, *Microsoft Word* komplektā ir daudz šablonu, un mēs varam veidot paši savus. Šablons ir dokuments, kuru izvēloties, mēs iegūstam tā kopiju, uz kuras vēlāk veidojam jauno dokumentu.



... lietojot nepareizi sabalansētus fontus.

Šablonā nepieciešams iekļaut tādus uzņēmuma dizaina elementus kā emblēmu, dekoratīvas līnijas, kā arī tekstus, kas nemainās, piemēram, telefona, faksa numurus, pasta un e-pasta adreses. Šablonā jāizveido arī stili. Stils ir formatējumu paraugs, ko parasti varam piemērot visai rindkopai. Stilā var iekļaut fonta veidu un izmēru, atkāpes pirmai rindai un visai rindkopai no visām pusēm, var mainīt arī attālumus starp rindām, vārdiem un burtiem, izlīdzināšanas un centrēšanas parametrus, teksta krāsu utt.

Katram rindkopas veidam ieteicams izveidot savu stilu, un to konsekventi lietot. Tas noderēs, ja, piemēram, iesniedzot garu rakstu konferencei, jums palūdz pamainīt virsrakstu un apakšvirsrakstu noformējumu. Mēs vienkārši pamainām stila parametrus, un lieta darīta. Latvijas apstākļos nevaram tieši izmantot standarta stilus, jo tie gan pēc valodas, gan pēc pielietojuma atbilst amerikāniskai videi, tamdēļ ieteicams tos izveidot pašiem.



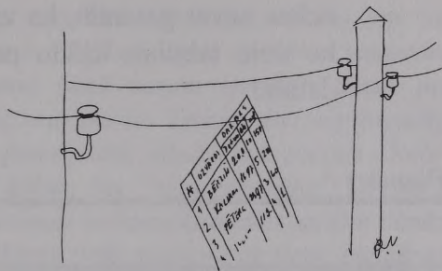
Šablons ir dokuments, kuru izvēloties, mēs iegūstam tā kopiju, uz kuras vēlāk veidojam jauno dokumentu.

Jau vidējam uzņēmumam varētu būt izdevīgi pasūtīt profesionālam dizaineram visu uzņēmuma galveno dokumentu veidu šablonus ar stiliem. Rezultātā visiem uzņēmuma dokumentiem būs vienāds, konsekvents un mākslinieciski profesionāls izskats. Dokumenti būs glīti noformēti, un nebūs problēmu ar nepareizi uzrakstītiem telefona numuriem un līdzīgiem pārrakstiem.

Ja mēs radām daudz dokumentu, tad tiešām jāpadomā par tehnoloģiju, kā mēs to darām.

Protams, modernie teksta redaktori dod mums daudz dažādu iespēju.

Jaunākie teksta redaktori praktiski izpilda daudzas no tekstu aplaušanas programmu funkcijām. Ar teksta redaktora palīdzību var pabeigt kā tipogrāfijai nododamu maketu ne tikai grāmatu, bet arī samērā sarežģītu žurnālu ar slejām, tabulām, bildēm utt. Tie nodrošina iespēju veidot Internet WWW lappuses .html formātā, kā arī aplūkot tīklā izveidotos dokumentus. To piedāvā *Microsoft Office* '97 komplekts. Tas viss vēlreiz apliecina, ka pasaule virzās ne tikai uz informācijas izveidi, bet arī uz informācijas glabāšanu un apskati tikai datoros. Nākotnes teksta redaktori būs vairāk orientēti veidot dokumentus elektroniskā formā, nevis izdrukai.



Elektroniskās tabulas

Termins “elektroniskās tabulas” ir angļu vārda *spreadsheets* ne gluži tulkojums, bet gan drīzāk saturiskais atspoguļojums. Kāpēc tās ir elektroniskas, būtu skaidrs, bet kā ir ar tabulu? Patiešām, darbība norit tabulas šūnās, attiecīgās kolonnas rindīnā ierakstot kādu simbolu virkni, skaitli vai formulu.

Kādam nolūkam ir paredzētas elektroniskās tabulas? Vienkāršoti uz šo jautājumu varētu atbildēt tā: “Visa veida aprēķinu veikšanai.” Ar elektronisko tabulu palīdzību var sastādīt banketa izdevumu pārskatu, ģimenes budžetu, izdarīt tirgus pētījumu analīzi utt. To visu, protams, var veikt ar roku uz papīra, taču katrs, kas ir veicis kādu uzskaiti, zina, ko nozīmē viena skaitļa izmainīšana vai nepareizi sarēķināta summa, kad visi aprēķini jāveic katru reizi no jauna, meklējot kādu viltīgi iezagušos kļūdu. Protams, elektronisko tabulu lietošana vien negarantē, ka kļūdu nebūs. Cilvēks arī tad, ja viņš strādā ar elektroniskām tabulām, ir tikai cilvēks, un

nekas nevar garantēt, ka viņš neklūdīsies, bet aprēķinos, ko veic tabulas, kļūdu patiešām nebūs. Un tas jau ir daudz!

Piemērs

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2				0,09					0,25			
3	Nr	Vārds	Alga	Soc.n.	Atviegl.	Apliek.	Neapliek.	Ienākums	Ieturēts	Ieturēt.	Ieturēts	Izmaksai
4				9%	par apg.	ienāk.	minimums	nodokļu apr.	algas nod.	alimentu	kopā	
5	1	Trusis	200,00	18,00	15,00	167,00	21,00	146,00	36,50		54,50	145,50
6	2	Bebrs	50,00	4,50	7,50	38,00	21,00	17,00	4,25		8,75	41,25
7	3	Dzenis	70,00	6,30		63,70		64,00	16,00		22,30	47,70
8	4	Lapsa (Kūmiņš)	160,00	14,40		145,60		146,00	36,50	40,00	90,90	69,10
9		Kopā:	480,00	43,20	22,50	414,30	42,00	373,00	93,25	40,00	176,45	303,55
10												
11												
12												

Trusis algu izmaksas sarakstu pēc lielās Meža uzkopšanas nolēma sastādīt ar elektronisko tabulu Excel palīdzību. Viņš rūpīgi pierakstīja visus datus un formulas. Ierakstīja atvieglojumus par diviem nepilngadīgiem bērniem sev un vienu Bebram. Ar zināmu prieku Trusis savā sarakstā iekļāva Lapsas Kūmiņa atvilkumu alimentiem. Šādi izskatās gatavs saraksts pēc visu aprēķinu veikšanas, bet ir iespējams apskatīties arī šūnu saturu.

	A	B	C	D	E	F	G	H
1								
2				0,09				
3	Nr	Vārds	Alga	Soc.n.	Atviegl.	Apliek.	Neapliek.	Ienākums
4				0,09	par apg.	ienāk.	minimums	nodokļu apr.
5	1	Trusis	200	=ROUND(C5*\$15		=C5-D5-E5	21	=ROUND((C5-D5-E5-G5
6	2	Bebrs	50	=ROUND(C6*\$7,5		=C6-D6-E6	21	=ROUND((C6-D6-E6-G6
7	3	Dzenis	70	=ROUND(C7*\$		=C7-D7-E7		=ROUND((C7-D7-E7-G7
8	4	Lapsa (Kūmiņš)	160	=ROUND(C8*\$		=C8-D8-E8		=ROUND((C8-D8-E8-G8
9		Kopā:	=SUM(C5:C8)	=SUM(D5:D8)	=SUM(E5:E8)	=SUM(F5:F8)	=SUM(G5:G8)	=SUM(H5:H8)
10								
11								

Lūk, šādas formulas nodrošina rezultātu. Protams, tabulu var vēl papildināt un papildināt, tuvinot to ideālam algas aprēķina maketam. Piemēram, uz ekrāna neredzamā šūnā varam ierakstīt datus par darbiniekiem, piemēram, apgādājamo skaitu: Trusim divi nepilngadīgi bērni, bet Bebram viens, vai arī pierēģistrēt atlaižu kategorijas - Bebrs bijis represēts un nosūtīts gāzt kokus, bet "ziņu pienesējs" Dzenis ir pensionārs. Veicot aprēķinus, vienmēr varam atsaukties uz šīm šūnām un pārbaudīt to saturu, un aprēķinus veikt saskaņā ar tiem, piemēram ja šūnas saturs ir 1, tad atvieglojumu atņemt, ja 0 — neatņemt.

Elektroniskā tabulā katrai šūnai ir savs identifikators-adrese, piemēram, *Microsoft Access* katrai tabulas rindai piešķir numuru 1,2, 3 ..., bet katrai kolonnai — burtu A, B, C ... Tas nozīmē, ka tabulas rindu un kolonnu krustpunkta šūnām tiek piešķirti nosaukumi C1, B5, A3 utt.



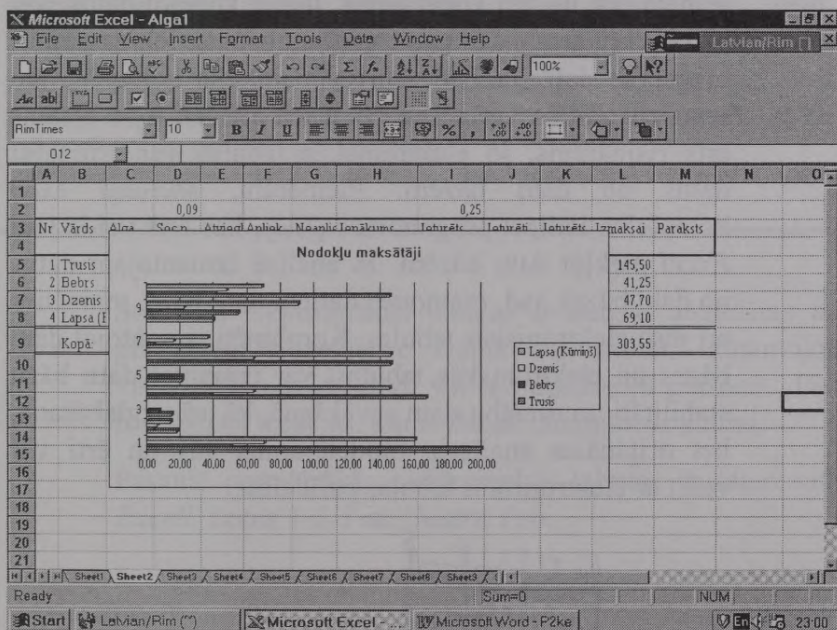
Tabulā katrai šūnai ir savs identifikators-adrese.

Lai arī kā mēs piešķiram numurus, nepastāv divas šūnas ar vienādiem nosaukumiem. Tas mums dod rīcības brīvību, jo:

1. runājot par kādu šūnu, mēs vienmēr zinām, par kuru šūnu tieši ir runa;
 2. elektroniskā tabula pēc nosaukuma atrod šūnu un nolasa tās saturu;
 3. mainoties šūnas saturam, šūnas nosaukums nemainās.
- Šūnā var rakstīt ne vien skaitļus un paskaidrojošu tekstu. Daudz svarīgāk ir tas, ka šūnās var rakstīt formulas, kas apstrādā skaitļus no norādītajām šūnām. Tā var veidot noteiktu šūnu summu, starpību un krietni vien sarežģītāku aprēķinu formulas.

Tas, ka ar elektronisko tabulu var veikt jebkādas aprēķinus, gan nenožīmē, ka aprēķinu veikšanai vienmēr tās ir jāizmanto. Elektroniskās tabulas labākā īpašība ir tā, ka dati un formulas atrodas vienuviet. Ar elektroniskajām tabulām ir vienkārši un ērti strādāt pat iesācējam, Katrs, kurš ieguvis elektroniskās tabulas failu, var pārbaudīt datus un formulas, kā arī tos mainīt un uzlabot. Elektroniskās tabulas ir neaizstājamas vietās, kur datus apstrādā radoši. Sastādot un modelējot biznesa plānu, izstrādājot budžetus, pētot tirgus stāvokli, elektroniskās tabulas sniedz atbildes uz jautājumiem: "Kas būtu, ja...?" Elektroniskās tabulas ietver daudz rīku, formulu, kas atvieglo, piemēram, uzzināt, kādam jābūt preču uzcenojumam, lai sasniegtu plānoto peļņas normu. Tātad elektroniskās tabulas spēj veikt aprēķinus arī "atpakaļ". Tajās ir iekļautas arī statistikas, finansu un citas funkcijas. Mēs varam paredzēt, kādi būs nākotnes finansu vai citi rādītāji pēc iepriekšējiem datiem. Protams, tabulas satur arī dažādus rīkus, kas uzskatāmi pasniedz informāciju.

Piemērs



Pēc daudzkārtējiem pārmetumiem par pārāk lielo algo, ko Trusis sev esot piešķīris, viņš nolēma ar piemēru parādīt, ka lielās algas saņēmējs maksā arī vislielākos nodokļus.

Tātad ar elektroniskās tabulas palīdzību varam ātri, ērti un vienkārši modelēt datus vai izveidot atskaiti. Bet, ja mums nepieciešams katru mēnesi veidot vienu un to pašu atskaiti, ērtāka būs datu bāze. Elektroniskās tabulas nav ērtas datu savākšanai un uzglabāšanai. Reizēm ar

vieniem un tiem pašiem datiem nepieciešams veidot dažādas atskaites. Protams, elektroniskās tabulas dod iespēju ērti un ātri kopēt datus, tomēr kopējot datus, var rasties kļūdas, īpaši tajos gadījumos, ja tiek izmainīti oriģinālie dati. Ja dati, kuri jāmaina, ir nokopēti, tad izmaiņas jāveic visās kopijās. Taču arī šeit ir iespējams ērts risinājums, jo elektroniskās tabulās var izmantot datus no datu bāzēm. Piemēram, *Microsoft Excel* komplektā ietilpst programmas *Query*, kas ērtā veidā ļauj *Excel* piekļūt datu bāzēm. Ja analīzē izmantojam datus no datu bāzes, tad, mainoties datiem datu bāzē, mainīsies arī jūsu elektroniskās tabulas. Kombinēti izmantojot datu bāzes un elektroniskās tabulas, var izmantot datu bāzu stabilitāti un drošību datu savākšanā, kā arī uzglabāšanā, bet oriģinālas analīzes, modelējumus ātri un ērti var veikt ar elektroniskās tabulas palīdzību.



pv.

... mainoties datiem datu bāzē, mainīsies arī elektroniskās tabulas

Uzmanību - PADOMS!

Ar elektroniskās tabulas palīdzību nebūtu nopietni veikt grāmatvedības datu uzskaiti pat mazā uzņēmumā, jo par dažiem

simtiem latu ir pieejamas programmas, kas šo uzdevumu veiks labāk, ātrāk, drošāk, un rezultātā tas būs lētāk. Protams, ja vien grāmatvede nav jūsu sievasmāte, kas strādā par velti. Elektroniskās tabulas gadījumā pašiem būtu jāizdomā, kā veidot grāmatvedību tabulās. Taču neviena grāmatvedības uzskaites sistēma nepiedāvās visa veida modeļus, atskaites un analīzes, ko var veikt ar elektroniskām tabulām.

Šodienas elektroniskās tabulas ir lieliski apgādātas ar grafiskiem līdzekļiem. Datus var attēlot, izmantojot visdažādāko veidu divu vai trīs dimensiju grafikus. Tās ir milzīgas iespējas, kuras jāizmanto ar gaumi.

Pasaulē populārākās elektroniskās tabulas ir *Microsoft Excell*, *Lotus 1-2-3* un *Quatro Pro*.



Datu bāzes

Katrs noteikti ne reizi vien ir dzirdējis jēdzienu “dati”, bet varbūt nav īpaši pedomājis, ko tas īsti nozīmē. Lietojot vārdu “dati”, divi dažādi cilvēki var saprast dažādas lietas. Vispārināti uzskatīsim, ka *dati* ir jebkāda veida saturiska informācija, ar ko var darboties, to apstrādājot, atlasot, kārtojot, caurskatot utt. Kā datu piemēru var minēt uzņēmuma darbinieku vārdus, uzvārdus, adreses, telefonus, dzimšanas datumus. Ja mums ir savākti visi šie dati, mēs no tiem varam iegūt visdažādāko informāciju, piemēram, visu darbinieku sarakstu alfabētiskā kārtībā, kāda konkrēta darbinieka telefona numuru vai dzimšanas datumu, visus darbiniekus, kas ir dzimuši februārī...

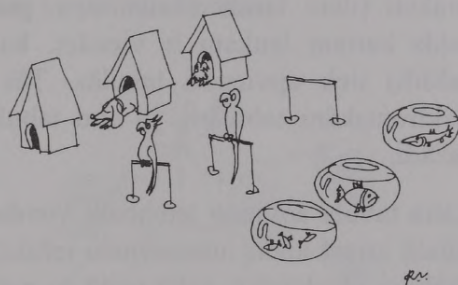
Pēc vienota standarta elektroniskā formā apkopotus datus sauc par datu bāzi. Tā patiešām ir kā bāze, kurā stingrā kārtībā ir sagrupētas un rūpīgi uzskaitītas visas nepieciešamās lietas, ko speciāli nesēji pēc pieprasījuma var piegādāt. Lai datu bāze nebūtu tikai nekam nederīgs

datu apkopojums, bet no tās varētu iegūt vajadzīgo informāciju vēlamā formā, ir izstrādātas speciālas datu bāzu vadības sistēmas (DBVS). To uzdevums ir padarīt jūsu darbu ar dažāda veida datiem vienkāršāku, ātrāku un, galvenais, ērtāku.

Lai izprastu, kas ir datu bāze, iedomājieties visparastāko kartotēku kaut vai bibliotēkas kartīšu veidā. Šādās kartītēs var būt arī ziņas par uzņēmuma vai iestādes darbiniekiem. Ja esat redzējis šāda veida kartītes, tad pamanījāt, ka tās visas ir aizpildītas pēc vienota parauga, kaut arī dati uz tām var pilnīgi atšķirties. Veidojot datu bāzi ar informāciju, ko satur kartītes, katra šīs kartītes aile atbilst datu laukam datu bāzē. Katram datu laukam ir savs nosaukums, pēc kura var iegūt atbilstošos datus no jebkuras kartītes.

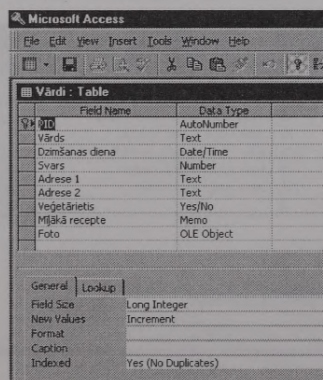
Informāciju, kas laukā tiek glabāta, raksturo datu lauku tips. Tas nosaka, kāda veida informācija tiek glabāta laukā un kāds darbības ir atļauta šī lauka datiem. Lauki var būt paredzēti darbībām ar skaitļiem, simbolu virknēm, datumiem, naudas izteiksmēm, loģiskām izteiksmēm u.c. Protams, pastāv iespēja pārveidot lauka datus no viena tipa uz kādu citu tipu. Piemēram, pārveidot skaitli simbolu virknē un otrādi. Veidojot datu bāzes struktūru, jau pašā sākumā pareizi jānovērtē, kādi lauki ir vajadzīgi un kādiem būtu jābūt to tipiem. Tāluņu numurus, kaut arī tā ir cipariska informācija, nebūtu jāklasificē kā skaitliska informācija, jo nekādas aritmētiskas operācijas jums ar tiem nebūs jāveic. Un, ja vien DBVS piedāvā tādu iespēju, tad datumu gan būtu jāievieto datumu, nevis simbolu tipa laukos, jo pretējā

gadījumā, lai sarēķinātu laika intervālu starp diviem notikumiem, jums būs jāveido sarežģītu aprēķinu programmas.



Informāciju, kas laukā tiek glabāta, raksturo datu lauku tips.

Piemērs



Microsoft Access tabulu veidošanas logs, kurā norāda datu bāzes lauku tipu, izmēru un citus parametrus.

Katru kartotēkas (datu bāzes) kartīti ar vienu vai vairākām ailēm (datu laukiem) sauc par datu ierakstu. Visi viena datu ieraksta lauki ir savā starpā saistīti un tiek uzskatīti par vienotu veselumu. Visi vienāda formāta ieraksti (datu lauku skaits, tips, garums, informācijas veids katram laukam ir vienādi, kaut arī paši dati ir dažādi) tiek apvienoti tabulās. Tās gan nav līdzīgas elektroniskām tabulām, jo abu tabulu pamatprincipi ir dažādi.

Katra tabula apvieno tematiski vienādos ierakstus. Šādā tabulā atrast kādu interesējošu ierakstu ir nesalīdzināmi ātrāk un vienkāršāk, nekā veikt šo pašu darbību parastajā katalogā.

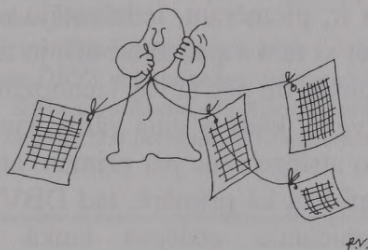
Piemērs

Vārdi : Table								
PID	Vārds	Dzimšana	Svars	Adrese 1	Adrese 2	Veģetārie	Mijāha recepte	Foto
1	Lācis Kepamis	91.03.03.	500	Lielais mežs	Zem celma pie upītes	No	Medus no dobuma	
2	Brālītis Truslītis	95.05.05.	7	Lielais mežs	Alina pie avota	Yes	Burkāni un zāķāpost	Bitmap Image
3	Lapsa Kūmīnš	93.03.14.	15	Lielais mežs	Ala zem vecās gobas	No	Svaigas peles	
7	Pelēkais Vilks	89.06.27.	55	Lielais mežs	Turnšā brikšņa	No	Ozādi medījumi	
8	Dzenis Raibais	91.05.15.	0.25	Lielais mežs	Sausais stumbrs	No	Tārpi un kāpuri	
9	Bebris Garzobis	93.05.18.	20	Lielais mežs	Mitrajā slapšnā	No	Zivis un vārdes	
10	Meža ruksis	92.07.03.	150	Lielais mežs	Kur ir ko ēst	Yes	Kartupeli, kāposti	
*	beir		0			No		

Ar Microsoft Access izveidota tabula

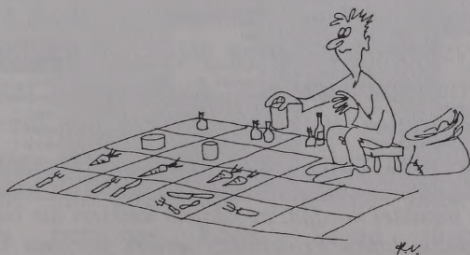
Tagad, kad esam izpratuši, kas ir tabulas un kā tās apvieno datu ierakstus, varam paplašināt datu bāzes jēdzienu: datu bāze apvieno savā starpā saistītas datu

tabulas. Tātad datu bāze elektroniskā formā glabā savstarpēji saistītus datus, nodrošina ātru un ērtu piekļūšanu šiem datiem, to papildināšanu, mainīšanu un cita veida speciālu apstrādi. DBVS ir programmu kopums, kas to visu nodrošina.



Datu bāze apvieno savā starpā saistītas datu tabulas.

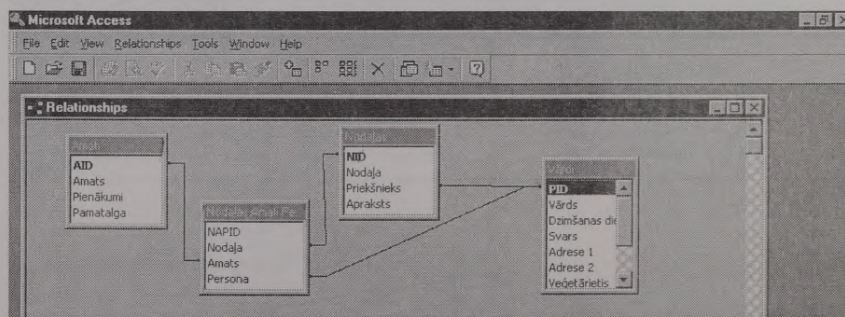
Lai atrastu kādu ierakstu datu bāzē, nav nepieciešams caurskatīt visas tabulas. Daudz vienkāršāk ir uzticēt caurskatīšanu DBVS rīkiem. Bet, lai šāda meklēšana noritētu veiksmīgi, ir pareizi jādefinē atslēgas jeb jāveic indeksācija.



Katra tabula apvieno tematiski vienādos ierakstus.

Katrai tabulai var definēt vismaz vienu atslēgu, ar kuras palīdzību DBVS atšķir tabulas ierakstus un atbilstoši tos sagrupē. Par atslēgu sauc speciāli iezīmētu datu lauku tabulā. Atslēgas vērtība ir lauka dati. Ja ieraksta atslēga ir unikāla, tā viennozīmīgi raksturo visu ierakstu. Tas nozīmē, ka diviem ierakstiem nav vienādu atslēgu. Tāda atslēga ir, piemēram, iedzīvotāju reģistra personas kods (vismaz tā tam vajadzētu būt), jo tas katram cilvēkam ir individuāls, un pēc tā var viennozīmīgi uzzināt, kas tas ir par cilvēku, kāds ir viņa vārds, kur viņš dzīvo utt. Šādu unikālo atslēgu sauc par primāro atslēgu. Ja kāda atslēga tiek definēta kā primārā, tad DBVS pati rūpējas par to, lai nepieļautu atslēgas laukā esošās informācijas atkārtotošanos.

Piemērs



Saites starp tabulām norāda, ar kādiem laukiem tās būs saistītas. Piemēram, tabulas Nodaļas lauks Priekšnieks ietvers personu identifikatorus un būs saistīts ar tabulas Vārds lauku identifikatoru PID, un tādējādi mēs nodaļas datiem varam viennozīmīgi piesaistīt tās priekšnieka personiģos datus. Savstarpēji

saistītas tabulas ļauj veidot patvaļīga izmēra struktūras, kas savādāk nebūtu iespējams.

Tabula Nodaļa-Amati-Personas veic pārējo trīs tabulu rakstu sasaistīšanu. Katrs raksts šajā tabulā veido vienu konkrētu atbilstību starp pārējo tabulu rakstiem, bet šādu atbilstību var būt patvaļīgi daudz. Nodaļā var būt patvaļīgs skaits viena vai dažādu amata vietu, kuras aizpildām ar konkrētiem darbiniekiem no *Personu* tabulas. Šī struktūra netraucētu vienam darbiniekam strādāt vairākās nodaļās vienlaicīgi. Papīrs pacieš visu, tāpat kā dators, šis ir tikai datu struktūras elastības piemērs.

Šādu saišu shēmai jābūt izstrādātai pirms tiek veidotas datu tabulas. Daudzos gadījumos saišu uzrādīšana DBVS nebūtu nemaz nepieciešama, jo saites tālāk tiek veidotas vaicājumos. Saišu uzrādīšana MS Access palīdz nodrošināt datu integritāti, mēs varam aizliegt izdzēst personu, ja tā kādā nodaļā ir uzrādīta kā priekšnieks.

Kā jau minējām, datu bāzē tabulas ir savstarpēji saistītas, t.i., starp tabulām tiek nodibinātas saites, kas nosaka saistību starp tabulām. Ja, piemēram, ir trīs atsevišķas tabulas: viena satur datus par visiem uzņēmuma darbiniekiem (personas kods, vārds, uzvārds, adrese), otra - par uzņēmuma nodaļām (nodaļas kods, nosaukums, darbības sfēra), trešā - par nodaļās nodarbināto darbinieku pienākumiem (nodaļas kods, darbinieka kods, pienākums), tad, lai uzzinātu kāda pienākuma veicēja mājas adresi, pietiek izveidot saiti starp pirmo tabulu, kurā tiek glabāta informācija par darbiniekiem, un trešo tabulu, kas satur pienākumus. Šādas saites izveidošana tiek veikta caur abu tabulu kopīgajiem laukiem, šajā gadījumā tāds lauks ir darbinieka personas kods. Tagad jāatrod mūs interesējošais pienākuma ieraksts, no kura iegūstam veicēja personas kodu. Zinot šo kodu, pirmajā tabulā uzmeklējam darbinieka mājas adresi, un viss ir kārtībā.



Saites izveidošana tiek veikta caur tabulu kopīgajiem laukiem.

Šādu saišu veidošana palīdz:

1. izvairīties no informācijas dublēšanas. Piemēram, tabulā par darbinieku veiktajiem pienākumiem nav nepieciešams atkārtoti glabāt visu pilno informāciju par katru darbinieku, jo šī informācija jau tiek glabāta darbinieku tabulā;
2. veikt datu vairākkārtīgas izmaiņas, jo tā kā informācija nav dublēta, šīs izmaiņas ir jāveic tikai vienu reizi;
3. samazināt nepieciešamo datora atmiņas apjomu.

Tagad sīkāk pakavēsimies pie tabulām. Tabula tiek veidota pēc šādiem principiem:

- katra tabulas kolonna atbilst datu laukam;
- katra tabulas rinda atbilst datu ierakstam.

Datu bāzē datu atlasīšana notiek nesalīdzināmi ātrāk nekā kartīšu kartotēkā, kur tas ir jāveic ar roku. Turklāt, ja vien ir pareizi norādīts atlases kritērijs un nav kļūdainu datu ievadu, tad atlases procesā nav kļūdu, t.i.,

tiek atlasīti tikai un vienīgi vajadzīgie ieraksti, respektīvi, nav arī liekas informācijas.



24.

... tiek atlasīti tikai un vienīgi vajadzīgie ieraksti.

Bieži šādus pēc noteikta kritērija atlasītus datus sauc par izvilkumu (*Dynaset*), bet pašu kritēriju (vai kritērijus), pēc kura (kuriem) tiek veikta datu atlase, sauc par vaicājumu (*Query*). Vaicājumu, izmantojot saites, var attiecināt arī uz vairākām tabulām uzreiz.



25.

... pēc noteikta kritērija atlasītus datus sauc par izvilkumu.

Principā saites tiek veidotas viena vaicājuma ietvaros, definējot saistības starp tabulu laukiem. Lai tas būtu iespējams, veidojot tabulas, jāparedz, kuri lauki būs

savstarpēji saistīti, un tādās programmās kā *Microsoft Access* iepriekšēja saikņu definēšana dos labākus ātrdarbības rezultātus.

Piemērs

Nodaras priekšnieki : Select Query

Nodaras

PID

Vārds
Dzimšanas di.
Svars

Field: Nodaras.* Vārds
Table: Nodaras Vārds
Sort:
Show: ☒ ☒

Nodaras priekšnieki : Select Query

	NID	Nodara	Apraksts	Priekšni	Vārds
▶	1	Celtniecība	Būvē alas	1	Lācis Keparnis
	2	Grāmatvedība	Skaita un pārskaita	2	Brālītis Truslītis
	3	Lauksaimniecība	Ievāc ogas, zuglus, saknes	10	Meža rukšis

#umber/

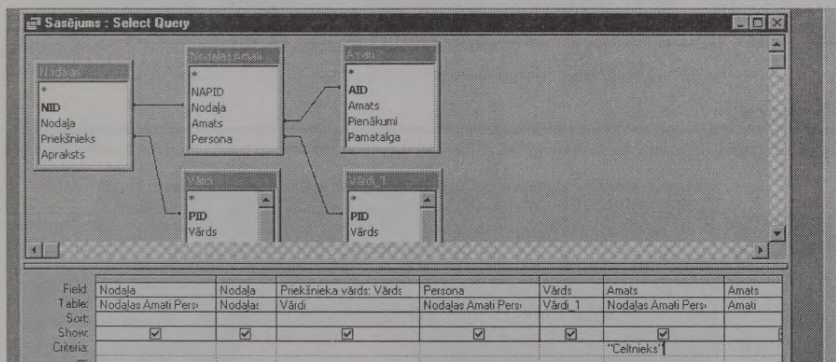
Vienkāršu vaicājuma veidošanas un darba logi. Tabula Vārdi saistīta ar tabulu Nodaļas ar identifikatora PID palīdzību. Tabulas Nodaļas laukā Priekšnieks mēs atzīmējam konkrētās nodaļas priekšnieka personas identifikatoru PID, kas mums dod viennozīmīgu saikni ar Vārdu tabulu, kurā ir visa informācija par personām. Vaicājumā mēs skatām visus laukus no tabulas Nodaļas un tikai lauku Vārds no tabulas Vārdi. Šādu vaicājumu var izmantot gan datu apskatei un atlasei, gan arī datu ievadam.

Vaicājuma būtiskākā īpašība ir tā, ka tas vienmēr atbilst tabulu pašreizējam stāvoklim, jo vaicājums pēc katra pieprasījuma tiek formēts no jauna un, atšķirībā no tabulām, netiek glabāts uz diska. Mainot datus vaicājumā, automātiski tiek izmainīti dati izejas tabulā. Parasts vaicājums neveido jaunus datus, tas tikai dod ērtākas iespējas strādāt ar esošajiem datiem no tabulām.

Tātad datu bāzes pamatā ir indeksētas, savā starpā saistītas tabulas ar datiem. Lai datu ievade būtu ērtāka,

drošāka un vienkāršāka, tiek izveidoti speciāli logi, ko sauc par formām, kurās izvietoti datu laukiem atbilstoši lodziņi, kur var izvadīt informāciju no datu bāzes vai, tieši otrādi, ievadīt jaunus datus.

Piemērs



Šis vaicājums uzrādīs mums visas personas, kas kādā nodaļā saņem pamatalgu kā amata Celtnieks izpildītājs, redzēsīm arī šīs nodaļas nosaukumu un nodaļas priekšnieka vārdu.

Formas ļauj apskatīt, rediģēt un papildināt rakstus no tabulām vai to izvilkumiem. Arī datu apskatei un izdrukai tiek izmantotas formas.



...lodziņi, kuros var izvadīt informāciju no datu bāzes vai, tieši otrādi, ievadīt jaunus datus.

Piemērs

N/D.	Nodaļa	Priekšnieks	Apraksts	Vārds
1	Celtniecība	1	Būvē alas	Lācis Kepānis
2	Grāmatniecība	2	Skaita un pārskaita	Brālītis Trucītis
3	Lauksaimniecība	10	Ievāc ogas, augļus, saknes	Meža rukas
*	[AutoNumber]			

Records: 14 of 3

Nodaļa	Priekšnieks
2 Administrācija	
1 Celtniecība	
3 Lauksaimniecība	

Vienkārša forma, kas viedota uz vaicājuma Nodaļas priekšnieki bāzes. Laukiem formā varam atvēlēt tik daudz vietas, cik nepieciešams, kā arī izmantot dažādus dizaina elementus. Strādājot ar formu, varam izmantot ikonas no rīku joslas, lai izmainītu šķirošanas kārtību. Piemērs ir sašķirts alfabētiskā kārtībā pēc lauka Nodaļa. Ir pieejami arī papildu filtrēšanas un datu meklēšanas rīki.

Formas pamatā ir jāizmanto tabula vai vaicājums, kas ļauj strādāt ar to laukiem, taču formu atverot, mēs varam izmantot parametrus, kas atlasa mums vajadzīgos datus.

Strādājot ar formu, mēs varam atlasīt un šķirot datus. Pēc tam mēs varam arī šķirot datus formā pēc tā lauka, kurš šoreiz mums ir svarīgāks. Formas, kas bāzētas uz vaicājumiem, ir vienīgais ērtais veids, kā ievadīt un pārskatīt datus savstarpēji saistītās tabulās.

Piemērs

Nodaļas priekšnieki

Nodaļa ar darbinieku sarakstu

NID Nodaļa	Apraksts	Priekšnieks
1	Celtniecība	Būvē alas, ligzdas, dambjus
1		Lācis Kēpains

Nodaļa	Amats	Pienākumi	Pamatalga
Vārds	Dzimšanas diena	5 vairs	Adrese 1
Adrese 2	Veģetābils		
1	Būvdarbu vadītājs	Uzraudzīt būvdarbu norisi. Rūpēties par materiālu savlaicīgu pi	100,00 Ls
1	Meža rūķis	92.07.03.	150 Lielais mežs
			Kur ir ko ēst
1	Celtnieks	Būvēt, mūrēt, rakt	75,00 Ls
1	Bēbris Garzobis	93.05.18.	20 Lielais mežs
			Mitrajā slapstā
1	Būvdarbu priekšnieks	Visu būvdarbu koordinācija	120,00 Ls
1	Lācis Kēpains	91.03.03.	500 Lielais mežs
			Zem celma pie upītes
*			
	Vidējais svarts:	223,33	
			Algas Kopā:
			295,00 Ls

Record: 1 of 3

Record: 1 of 3

Sarežģīta forma ar apakšformu, kas izmanto iepriekšaplūkotos vaicājumus. Augšējā forma ir vienkārša forma, kas uzrāda datus no tabulas Nodaļa un Personas, taču tajā ir ievietota sarežģītāka apakšforma, kas izmanto datu saisti ar starptabulu. Ievietojot apakšformu, tiek uzrādīti saistošie lauki, šajā gadījumā tas ir nodaļas identifikators NID. Rezultātā apakšformā mēs redzam tikai tos amatus un tās personas, kas saistītas ar konkrēto nodaļu.

Nodaļa	Amats
1. Būvniecība	D
2. Būvniecības vadītājs	
3. Finanšu direktors	
4. Celtnieks	
5. Mežsaimnieks	
6. Prezidents	
7. Sekretārs	
8. Būvniecības priekšnieks	
9. Īpašnieks	

Augšējā formā pārejot uz nākošo ierakstu, pilnībā mainītos apakšformas saturs. Apakšformā tiek izmantotas arī agregātfunkcijas, aprēķinot kopīgo algu fondu un vidējo darbinieka svaru. Šajā formā mēs varam ievadīt jaunus rakstus un labot rakstus no visām apskatītajām tabulām.

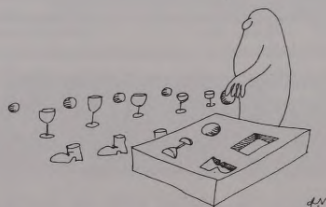
Iespēja jau esošos rakstus izvēlēties no saraksta atvieglo datu sasaisti. Izvēloties amatu no saraksta, mēs konkrētai personai piešķiram amatu no esošās tabulas Amati, un visi tabulas Amati lauki vaicājumā automātiski tiek aizpildīti. Rakstot no jauna amatu, mēs izveidojam jaunu ierakstu tabulā Amati.

NID	Nodaļa	Apraksts	Priekšnieks
1	Celtniecība	Būvē alas, ligzdas, dambus	1. Lācis Ķepains

Nodaļa	Vārds
1	Būvniecība
2	Meža ruksis
3	Celtnieks

Find in field: 'Nodaļa'	
Find What:	Administrācija
Search:	All
Match:	Whole Field
Match Case	☐
Search Fields As Formatted	☐
Search Only Current Field	☒
Find First	Find Next
Close	

Ja dati nav viegli atrodami, varam izmantot meklētāju, konkrētajā gadījumā tiek meklēta nodaļa "Administrācija". Šī forma, iespējams, ir pārāk sarežģīta un nepārskatāma reālam darbam, taču tā lieliski ieskicē datu bāzu sistēmu iespējas.



Forma atlasa mums vajadzīgos datus.

Formās var arī iekļaut formulas, ko izmantojam aprēķinātajos laukos, bet vērtību var aprēķināt no citu

lauku vērtības. Forma tiek iedalīta detaļu sekcijā, galvenē (*header*) un pakājē (*footer*). Formas pakājē var ievietot agregātfunkciju, kas saskaita kādu kolonnu vai aprēķina vidējo vērtību. Modernas DBVS atļaus formā ievietot grafiku, kas veidots uz formā atlasīto datu pamata.

Piemērs

Darbinieku algas

Nodaļa	Apraksts	Priekšnieks
Administrācija	Skaitļa un pārskaita	Brālītis Trusītis
Lapsa Kūmiņš	Sekretārs	0,00 Ls
Pelēkais Vilks	Prezidents	0,00 Ls
Brālītis Trusītis	Finansu direktors	200,00 Ls
Administrācija		200,00 Ls
Celtniecība	Būvē alas, ilgčas, dambļus	Lācis Ķepainis
Lācis Ķepainis	Būvdarbu priekšnieks	120,00 Ls
Bebris Garzobis	Celtnieks	75,00 Ls
Meža rūķis	Būvdarbu vadītājs	100,00 Ls
Celtniecība		295,00 Ls
Kopā		495,00 Ls

Atskaite ar vienu grupēšanas līmeni. Katrai nodaļai ir algu summa, kā arī visai atskaitēi ir kopīgā algu summa.

Arī atskaites spēj nodrošināt to pašu, ko formas. Atšķirībā no formām atskaitēs nav iespējams labot datus, un atskaites ir vairāk domātas izmantošanai drukātā

veidā. Taču šie ierobežojumi rada tehniskas iespējas veikt citas manipulācijas. Atskaitēs mēs varam vairākkārtīgi grupēt datus, veikt summēšanu, procentu aprēķinus pēc grupām, apakšgrupām, protams, arī izmantot grafiskus līdzekļus.

Modernas DBVS atļauj izmantot *Microsoft Windows* fontus, radot iespēju skaisti noformēt atskaiti ar dažādu izmēru, dažādu fontu burtiem. Lai vēl vairāk atvieglotu darbu ar datu bāzi un izveidotu vēl jaudīgākas iespējas, DBVS sistēmas ļauj rakstīt makrokomandas un iekļaut programmas datu manipulācijai.

Uzmanību - PADOMS!

Mūsdienu DBVS ir ērtas lietošanā, ar tām ļoti ātri var izveidot samērā sarežģītas datu bāzes, tās var skaisti profesionāli noformēt. Tomēr jāatceras, ka mēs nemaz neesam tik unikāli un pasaule ir pilna ar gatavām datu bāzu sistēmām, kas paredzētas grāmatvedībai, adrese sistēmām, personīgo finanšu uzskaiti utt. Autoru bēdīgā pieredze, kad ļoti labi izstrādāta un pārbaudīta datu bāze jāmet ārā, jo parādījusies vēl labāks firmas izstrādājums, ir spilgts apstiprinājums iepriekšteiktajam.

Ja tomēr situācija ir unikāla un ja pašiem jāveido sava datu bāzu sistēma, ieteicams to uzticēt profesionālim. Ātri izveidot sarežģītu, izskaistinātu datu bāzi ir viens, cita lieta ir nodrošināt tās darbu ar lieliem, augošiem datu apjomiem daudzlietotāju vidē. Turklāt formu un atskaišu izveidošana ir samērā ķimperīgs, ilgs process, kam nepieciešama "uzsista roka". Vienīgā datu bāzu

sistēma, ar ko neprofesionālis varētu pamēģināt veidot datu bāzi, ir *Microsoft Access*.

Būtu ieteicams, lai datu bāzi veidotu profesionāls programmētājs pēc klienta prasībām. Starp citu, izveidot projektu programmai un pēc tam pārbaudīt programmas atbilstību projektam ir problēma, ar ko nodarbojas pat zinātnieki teorētiskā līmenī, projektu reāli izstrādāt ir spējīgs profesionālis, pamatojoties uz interviju ar klientu.



pv.

Ja tomēr situācija ir unikāla un ja pašiem jāveido sava datu bāzu sistēma, ieteicams to uzticēt profesionālim.

Protams, lielākā daļa datu bāzu ir būvētas pēc “Ķirbīša metodes”. Ķirbītis katru dienu portfeli atnesa trīs ķieģeļus savam namiņam, kurus uzreiz arī iekāpināja namiņā. Pēc sociālisma terminoloģijas Ķirbītis bija “nesējs”. Šādi veidotas datu bāzes vienā brīdī sabrūk, jo to izveides pamati nav pārdomāti. Ar lieliem datu apjomiem tās vairs nestrādā gandrīz nemaz, vai arī darba procesā tīklā parādās kļūdas. Protams, iepriekšminētais attiecināms uz jebkura veida programmēšanu un ne tikai. Programmēšanā visspilgtāk izpaužas fenomens, ka “1+1 nav 2”. Katrs kārtīgs programmētājs spēj uzrakstīt 1000 operatoru programmu, teiksim nedēļā, tātad divas 1000 operatoru programmas divās nedēļās. Taču 2000 operatoru programmu šis programmētājs varbūt vispār nav spējīgs uzrakstīt. Par programmu šajā gadījumā uzskatīsim to algoritmiskas informācijas daudzumu, kas datorā veic ko jēdzīgu.

Datu bāzes formām un atskaitēm ir jāietver visas standarta atskaites un analīzes. Datu bāze ir jāsakārto tā, lai pie datiem varētu ērti tikt klāt ar elektronisko tabulu. Tādējādi ar *Microsoft Excel*, piemēram, var veidot dažādas nestandarta atskaites un datu analīzes, kuras katru reizi nepieciešamas savādākas un kuras neatmaksājas pasūtīt.

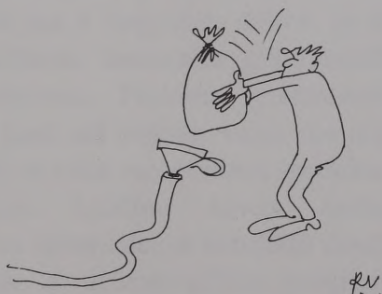
Lielas datu bāzes tīklā bieži vien tiek sadalītas pa funkcionālām daļām. Tiek izmantota klienta servera arhitektūra. Uz datora servera strādā programma DBVS serveris, kas nodrošina datu padevi pēc pieprasījuma DBVS klientiem, kas strādā ar datoriem tīklā. Visbiežāk pieprasījumi tiek veikti ar standarta pieprasījumu valodas palīdzību SQL (*Structured Query Language*). Tādēļ bieži vien datu bāzes serveris tiek saukts par SQL serveri.



Programma DBVS serveris nodrošina datu padevi pēc pieprasījuma DBVS klientiem.

Kamdēļ nepieciešami šādi sarežģījumi? Gadījumā, ja neizmantojam SQL serveri, bet lietojam datu bāzi tīklā, katrs pieprasījums no lielas tabulas nozīmē visas tabulas

pārsūtīšanu pa tīklu. Tikai pēc tam, kad visa apskatāmā tabula pārsūtīta, uz lietotāja datora sākas datu atlase, šķirošana utt. Tabula var būt vairākus Mb liela, un lietotāju skaits sasniegt vairākus desmitus. Šādā gadījumā tīkls būs pārslogots. Bet patiesībā lietotājs ir gribējis tikai viena mēneša datus par konkrētu klientu, kas kopumā ir mazāk par 1% no visas datu bāzes apjoma. SQL servera gadījumā datu bāzes klients sūta SQL pieprasījumu datu bāzes serverim. Pēc pieprasījuma serveris atlasa datus un sūta tos pa tīklu. Daudzlietotāju režīmā šī ir vienīgā iespējamā datu bāzu ideoloģija, kas strādā ar lieliem datu apjomiem.



... strādā ar lieliem datu apjomiem.

Šobrīd ir aktuāla datu publicēšana vai vākšana Internet tīklā. Šādus rīkus jau piedāvā *Microsoft Access 7.0*. Mēs varam ērti gan publicēt statiskus datus WWW lappusē, gan servēt datus no datu bāzes, pamatojoties uz pieprasījumu, kas noformēts iepriekšējā WWW lapā.

Šajā apskatā mēs esam aprakstījuši relāciju datu bāzes, kas ir populārākais datu bāzu veids. Relāciju datu bāzes

izmanto fiksēta formāta savstarpēji saistītas tabulas. Iespējams, ka nākotnē populāras būs objektu orientētas datu bāzes (OODB). OODB katra nākošā raksta laukā var būt cits objekts, tādējādi radot ļoti elastīgu datu struktūru. Šodienas datori ir par lēnu, lai reālā laikā spētu apkalpot šādas datu bāzes. Tādējādi OODB nekur tālāk par laboratorijām un žurnālu rakstiem pagaidām netiek.



Uzskaites sistēmas

Grāmatvedība ieņem stabilu vietu līdzās citām dzīves jomām, kurās jau ir iespiedies dators, un kopš personālā datora parādīšanās datorizēta grāmatvedība ir radījusi patiesu apvērsumu. Protams, grāmatveži par to nav sajūsmā, jo kaut arī viņiem vairs nav stundām un pat dienām jāsež, ar roku veicot visus aprēķinus vai meklējot kāduniecīgu kļūdiņu savos aprēķinos, tomēr grāmatvedības datorizācijas rezultātā daudzi ir zaudējuši darbu, jo līdz ar grāmatvedības programmu ieviešanu vidēji liela uzņēmuma grāmatvedības uzskaitē vairs nav nepieciešams milzīgs grāmatvežu skaits.

Kas tad ir grāmatvedības programmas? Tās ir programmas, kas ir izveidotas, lai atvieglotu grāmatvedības uzskaiti firmā, iestādē, uzņēmumā vai vienalga kādā organizācijā, kurā notiek naudas plūsma. Turklāt, plašas grāmatvedības programmas ne vien veic visas operācijas ar naudas līdzekļiem, bet arī uzskaita materiālās vērtības, produktus un citas lietas, par kurām uzņēmumam vienmēr būtu jābūtu skaidrībā.

Patiesībā būtu gan jārunā par uzskaites sistēmām. Uzskaitē jāveic praktiski it visur, bet ne visur to var saukt par grāmatvedību. Vienkāršākās personīgo līdzekļu uzskaites sistēmas ļauj tikt skaidrībā ar personīgajiem līdzekļiem un mantu. Sekojot līdzi investīcijām biržā, izmantojot Internet, varam pārrēķināt mūsu akciju vērtību konkrētā brīdī. Tas gan Latvijā vēl maz kam ir aktuāli.

Piemērs

Quicken 6 for Windows - Trusis - [Maciņš - Naudīga maciņš: Cash]						
File Edit Features Lists Reports Window Help						
Delete	Find...	Transfer...	Update Balance...		Edit Account...	Report View Close
Date	Ref	Payee / Category / Memo	Spend	Dr	Receive	Balance
11.08.97		Intermežs Šampaniets Trusītei Mīmītei	3.75			210.95
11.08.97		Karaliskā Meža Eļļas bode Gliemežvāks Auto Benzīns Pilna bāka	17.00			193.95
11.08.97		Karaliskā Meža Eļļas bode Gliemežvāks Higiēna	0.45			193.50
12.08.97		Dresszoo Drēbes Jaunas apakšbikses, savādāks kauns	3.00			190.50
12.08.97		Alus Krūmi Prezenlācījas Biznesa pusdienas ar Āhikas Kāni	12.00			178.50
12.08.97		Naktisklubs Underwood Izklaides Dejas ar Mīmīti	7.00			171.50
13.08.97		Mežpegro Aļņš uz pohām	0.80			170.70
14.08.97		Meža uzņēmumi Prēmijas No Vilka dzimšanas dienā			50.00	220.70
14.08.97		Kukū kūkas Ēdieni Bukāntorte kolāģiem	8.00			212.70
14.08.97		Karaliskā Meža Eļļas bode Gliemežvāks Zeļa Vindara vīns	7.00			205.70
14.08.97		Sports un Skastums Sporta rīki Skritulslidas	45.00			160.70
14.08.97		Restorāns Nokitusi Varde Vakarījas ar Mīmīti	18.00			142.70
15.08.97		Viesnīca Burkānsons SDS Viesnīcas Romantiskā dandusa	75.00			67.70
Ending Balance:						67.70
Maciņš Kredītkarte Aktīvi Saistības						
Start Contacts - Microsoft Out... Welcome to Zi On-Line... Microsoft Word - Pae.doc Quicken 6 for Window... 15:34						

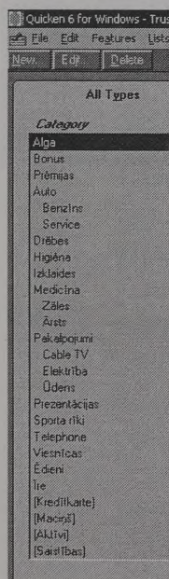
Trusis izmanto Quicken 6, lai vestu personīgo izdevumu-ieņēmumu uzskaiti. Visiem ieņēmumiem un izdevumiem jāuzrāda laiks un vieta, kur tie veikti, kā

arī tie ir sakategorizēti. Trusis savus izdevumus komentē, līdz ar to veidojas savdabīga dienasgrāmata, ko gan labāk svešiem nerādīt.

Trusim ir četri "balances" konti - maciņš, kredītkarte, aktīvi un saistības. Ar šo kontu palīdzību zināmā mērā var veidot "īstu grāmatvedību".

Saistību kontā iespējams iegrāmatot aizņēmumu no bankas dzīvokļa iegādei un sekot tā atmaksai. Savukārt, aktīvu kontā varam atzīmēt tādu vērtīgu lietu kā dzīvokļa, automobiļa iegādi. Nopērkot automobili, mēs neiztērējam 3000Ls, mūsu īpašums tikai maina formu. Quicken var atbildēt arī uz jautājumu par personīgo bagātību.

Aizejot uz restorānu, mēs iztērējam tos 18Ls, un viss. Šādus tēriņus Quicken grupē kategorijās un apakškategorijās. Attēlā var redzēt Truša kategoriju sarakstu.



Pateicoties kategoriju sarakstam, ir iespējamās dažādas atskaites un grafiki. Ja informācija ir ievadīta sistēmā pareizi, tad iespējami visdažādākie informācijas apskates un analīzes veidi.

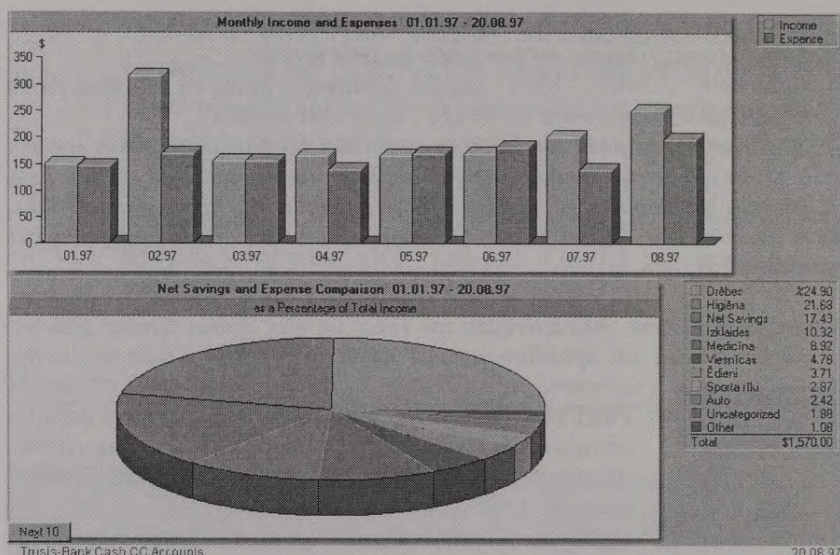
Lielākā problēma ir saprast, kā pareizi kategorizēt datus, tā ir prasme, kas nāk ar laiku. Izpētot tādas sistēmas kā Quicken, var uzzināt ļoti daudz par finansēm, budžetu, akcijām utt.

Augšējā grafikā redzama Truša ienākumu-izdevumu dinamika gada laikā. Izskatās, ka februārī bijusi trīspadsmitā alga. Apakšējā grafikā redzams, par ko ir tērēta Truša naudiņa šajā laikā.

Quicken priekšrocība ir tā, ka visdažādākie grafiki un atskaites ir jau sagatavotas, atliek tikai izvēlēties, kuras jums ir nepieciešamas, un izmantot tās.

Protams, personālās uzskaites sistēmas lietošana ir jēdzīga tikai tad, ja naudas operācijas ir sarežģītas.

Trusis, protams, ir pieredzējis naudas tērētājs, kam jāseko līdz kredītkartei un dzīvokļa pirkuma aizņēmumam.



Comparison Report

01.07.97 Through 20.08.97

20.08.97

Category Description	01.08.97- 20.08.97	01.07.97- 31.07.97	Amount Difference	% Difference
INFLOWS				
Aiga	200.00	200.00	0.00	0%
Prēmijas	50.00	0.00	- 50.00	- 100.00%
TOTAL INFLOWS	250.00	200.00	- 50.00	- 20.00%
OUTFLOWS				
Auto	17.00	21.00	- 4.00	- 23.53%
Drēbes	3.00	83.00	- 80.00	- 2000.00%
Higiēna	0.45	0.00	0.45	100.00%
Izkādes	7.00	0.00	7.00	100.00%
Prezentācijas	12.00	5.00	7.00	58.33%
Sporta rīki	45.00	0.00	45.00	100.00%
Vienības	75.00	0.00	75.00	100.00%
Ēdieni	8.30	50.00	- 41.70	- 502.41%
TOTAL OUTFLOWS	167.75	139.00	28.75	17.14%
OVERALL TOTAL	82.25	61.00	- 21.25	- 25.84%

Quicken ir pilns ar dažnedažādām datu analīzēm un atskaitēm. Trusim ir visas iespējas salīdzināt savus tēriņus un ieņēmumus dažādos laika periodos. Šī atskaite ir lielisks piemērs ārzemju uzskaites sistēmas priekšrocībām un trūkumiem, jo tā demonstrē analīžu daudzveidību un izstrādātības pakāpi,

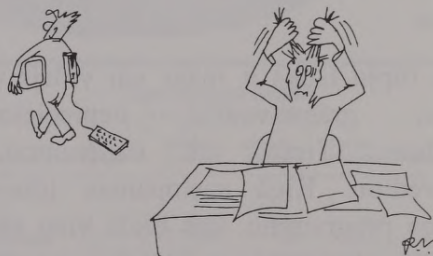
pašam nekas nav jāizdomā, jāiemācās izmantot to, kas ir, bet, savukārt, nav nekādas iespējas tikt vaļā no angliskiem uzrakstiem.

Programma *Quicken*, kas ir līderis personīgo uzskaites sistēmu jomā, spēj veikt operācijas, kas atbilstu maza uzņēmuma grāmatvedībai. *Quicken* izmanto arī daudzas bankas, atļaujot klientiem, ar *Quicken* caur telefonu tīklu iezvanoties, kontrolēt savus rēķinus. Ir iespējams apskatīt savu rēķinu, izdarīt komunālos un citus maksājumus. Kārtīgi vadot datus uzskaites sistēmā un tos pareizi organizējot, iespējams iegūt visdažādākās atskaites.

Ja jūsu rūpju lauks ir mazs vai vidējs uzņēmums, tad, protams, grāmatvedībā nepieciešams speciāla programma. Tiesa, cik uzņēmumu, tik dažādu grāmatvedību. Rīgā sastopamas ļoti daudz vietēja ražojuma programmu, kas bieži vien sākušās no viena uzņēmuma datorizācijas. Daudzas no tām ir aprobētas daudzos uzņēmumos ilgā laikā. Izvēloties vietējo programmu, vajadzētu novērtēt ražotāja spēju sniegt konsultācijas un uzlabojumus ilgā laika periodā. Šī spēja un vēlme parasti var tikt labi novērtēta ar programmas pašreizējo izplatības daudzumu. Populāru programmu vienmēr kāds gribēs uzturēt. Programmētājs vientuļnieks var sniegt jums lielisku servisu un būt ļoti lēts, jo viņam nav jāuztur birojs, bet vienā brīdī viņam var apnikt šī programma, un ko tad? Visas vietējās programmas parasti veido izdrukas, kādas grāmatvedībai jāiesniedz dažādās kontroles institūcijās. Protams, ka ārzemēs

ražotas programmas tiešā veidā neveido Latvijā nepieciešamās atskaites, tiesa, visi nepieciešamie dati ir iegūstami. Toties ārzemēs ražotajās programmās ir ļoti daudz standarta atskaišu, kas atvieglo uzņēmuma vadību. Bieži vien datus var analizēt grafiski.

Izmantojot ārzemēs ražotu programmu, mēs saskaramies ar pasaules pieredzi uzskaites veidošanā un varam ļoti daudz ko iemācīties. Jautājums, vai grāmatvedība ir tam, lai atrakstītos Valsts Ieņēmumu Dienestam, vai lai mēs paši spētu efektīgi vadīt uzņēmumu? Ideālai programmai būtu jāsniedz visas šīs iespējas ērtā veidā.



*Programmētājs vientuļnieks var sniegt jums lielisku servisu,
bet vienā brīdī viņam var apnikt šī programma, un ko tad?*

Vislielākā datorizētās uzskaites problēma ir to pareizi izmantot. Pašas programmas var izmantot ļoti dažādi, un ieguvums būs atkarīgs no tā, kā būs izveidotas datu struktūras un datu plūsma.

Ieviešot lielas grāmatvedību sistēmas, uzņēmumi, bieži vien izmanto auditoru firmu pakalpojumus, lai pareizi parametrizētu sistēmu un definētu datu plūsmas. Jau vidējā uzņēmumā profesionāla padomi šai jomā var ļoti labi atmaksāties.

Ko tad veic grāmatvedības programmas? Tās parasti ir sadalītas moduļos. Protams, grāmatvedības pamatā ir galvenā grāmata ar kontējumiem un iespēju no tās izveidot bilanci, peļņas un zaudējumu aprēķinu un naudas plūsmas atskaiti.

Noliktavas modulis uzskaitīs materiālu plūsmu, preces noliktavā, to ienākšanu, iziešanu. Kaut kur starp galveno grāmatu un noliktavu jābūt modulim, kas uzskata debitoru un kreditoru parādus. Katrā sistēmā tas var būt izveidots savādāk. Protams, jābūt modulim, kas uzskaita pamatlīdzekļus un aprēķina to amortizāciju. Tāpat ir nepieciešams algu un ar to saistīto nodokļu aprēķināšanas modulis. Šīm daļām praktiski jābūt jebkurā grāmatvedībā.

Vēl iespējami daudz un dažādi moduļi, piemēram, servisa uzskaites vai ražošanas uzskaites moduļi. Šādi moduļi jau atspoguļo uzņēmuma darbības profilu. Labai sistēmai jāspēj visas šīs lietas ērti integrēt kopā tā, lai vienreiz ievadīti dati otrreiz vairs nav jāievada.

Uzmanību - PADOMS!

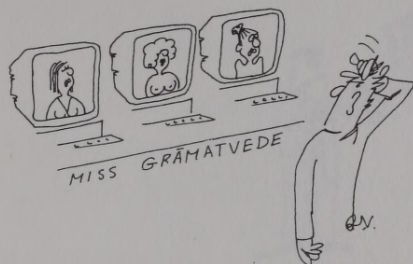
Grāmatvedības programmu tirgus ir ļoti plašs, taču izvēlēties īsto nav nemaz tik vienkārši. Pirms tērēt līdzekļus kādas grāmatvedības programmas iegādei, ir jāizvērtē šādi apstākļi:

- jūsu uzņēmuma apjoms (gan darbinieku, gan apgrozījuma ziņā);

- jūsu vajadzības (cik plašs piedāvāto iespēju spektrs jums patiešām ir nepieciešams);
- jūsu iespējas.

Tas viss jums ir jāzina par savu uzņēmumu, taču ir vēl vesela virkne ar prasībām, kuras jāizvirza programmatūrai, kuru jūs izvēlēsities grāmatvedības uzskaitēi:

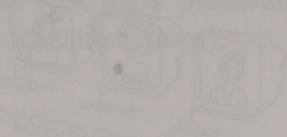
- pārliecinieties, ka jūsu noskatītā programma atbilst jūsu uzņēmuma prasībām;
- pārbaudiet, vai tā spēj apkalpot jūsu uzņēmuma grāmatvedības apjomu un vai tā piedāvā visas jums nepieciešamās funkcijas;
- noskaidrojiet, kādi ir nepieciešamie datora parametri, lai programma varētu sekmīgi funkcionēt: operētājsistēma (MS-DOS, Windows, MAC, OS/S u.c.), datora ātrdarbība, operatīvās atmiņas apjoms, cietā diska apjoms. Ja jums ir lokālais tīkls, pārliecinieties, ka programma nodrošina darbību tajā;
- pārbaudiet, vai tā nodrošina izmaiņu veikšanu. Ir ļoti svarīgi, lai vienkāršas nodokļu procentuālo likmju izmaiņas neradītu neprognozējamu problēmu sēriju;
- noskaidrojiet, cik ilgi programmas piegādātājs “uzturēs” programmu (tas gan vairāk attiecas uz mūsu vietējo programmu veidotāju produktiem);
- pārliecinieties, ka veidotās atskaites ir vajadzīgajā formā, un vai nepieciešamības gadījumā tās var pietiekoši ērti pamainīt.



... izvēlēties īsto nav nemaz tik vienkārši.

Ja jums nav līdzekļu pilna programmas moduļu komplekta iegādei, painteresējieties par iespējām nopirkt izvēlēto pamatkomplektu, bet pārējās komponentes piepirkt vēlāk pēc vajadzības.

• The first part of the report is a summary of the work done during the last year. It is a very short and simple summary, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a summary report.



The second part of the report is a more detailed account of the work done. It is a very long and detailed account, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a detailed report.

• The third part of the report is a summary of the work done during the last year. It is a very short and simple summary, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a summary report.

• The fourth part of the report is a more detailed account of the work done. It is a very long and detailed account, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a detailed report.

• The fifth part of the report is a summary of the work done during the last year. It is a very short and simple summary, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a summary report.

• The sixth part of the report is a more detailed account of the work done. It is a very long and detailed account, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a detailed report.

• The seventh part of the report is a summary of the work done during the last year. It is a very short and simple summary, but it is a good one. It gives a clear and concise picture of the work done, and it is easy to read. It is a good example of a summary report.



Personālie organizatori

Savā attīstībā cilvēks ir izgājis cauri vairākām pašdisciplinēšanās stadijām. Ir izdomāti visdažādākie kalendāri un piezīmju blociņi, lai tikai cilvēks varētu pēc iespējas ērtāk plānot savu dienu. Protams, arī datori ne vien neatpaliek, bet piedāvāto iespēju ziņā pat pārspēj savus papīra analogus.

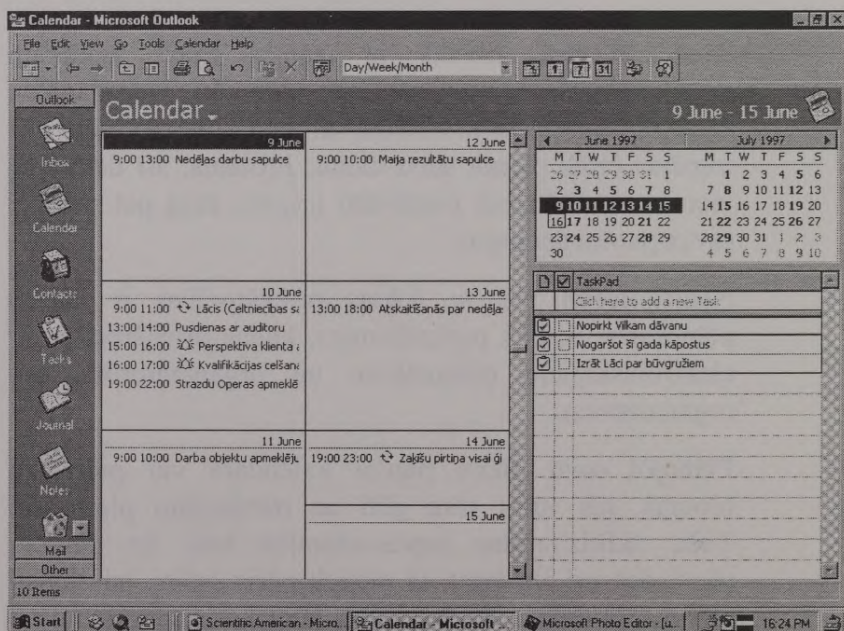
Nediskutēsim par to, kādas priekšrocības ir dienas plānošanai. Labāk paskatīsimies, kādas priekšrocības ir elektroniskajiem plānotājiem un organizatoriem jeb organaizeriem.

Plānojot savu dienu papīra kalendārā vai pierakstu blociņā, jūs varat visu glīti un pārskatāmi pierakstīt. Taču tiklīdz rodas nepieciešamība kaut ko pārcelt, pārveidot vai izsvītrot, tā pietiek pāris švīku, lai skaisti pierakstītā diena pārvērstos par kaut ko nesaprotamu un nesalasāmu. Izmantojot datoru, var veikt visdažādākās

modifikācijas, tomēr saplānotā diena, kā bija, tā paliks perfekti izskatīga un pārskatāma.

Pat vislabāk izstrādātie kalendāri un piezīmju bločiņi nevar sevī apvienot visas plānošanas iespējas. Pat ja tie satur vairākas iespējas, tad pārslēgšanās starp tiem ir apgrūtināta. Datora plānotajā pārslēgšanos starp dažādiem plānojumiem var izdarīt ar vienkāršu peles klikšķi.

Piemērs



Trusijs sava laika plānošanai izmanto Microsoft Outlook. Ikonas kreisajā pusē ir ērts veids, kā pārlekt no kalendāra uz E-pastu, adresu grāmatu vai darbu sarakstu. Mazajos kalendāriņos, apzīmējot dienas, varam mainīt apskatāmo

dienu skaitu lielajā kalendārā. Labajā apakšējā stūrī ir palikusi vieta darbu sarakstam, kas vispār ir cits Outlook saraksts, bet saīsinātā formā pieejams arī šajā skatā.

Jā, protams, jūs nevarat pasist savu galda datoru padusē un doties uz tikšanos ar darba kolēģi vai priekšnieku. Taču tāpēc nav vērts stingri turēties pie pierakstu blociņa, jo visu, ko esat saplānojis savā datorā, jūs varat izdrukāt uz papīra un nest līdzī, kurp vien vēlaties.

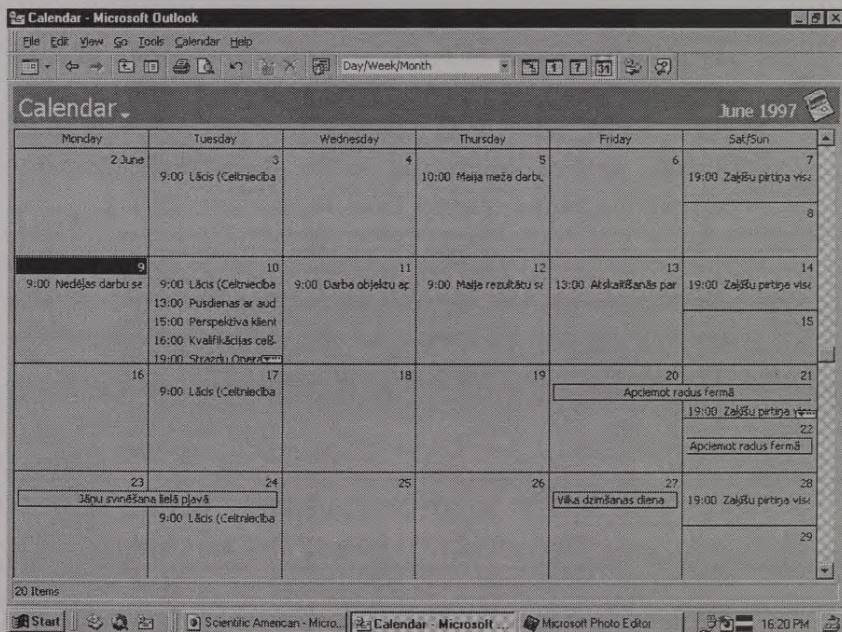
Pateicoties datoru plašai pieejamībai, darba organizācijas programmatūra sāk ļoti ātri attīstīties. Šodien visplašāk pazīstamā programmatūra ir *Microsoft Schedule+* un *Lotus Organizer*. Pašreiz pieejama ir jau arī jaunāka programmatūra, proti, *Microsoft Outlook* no *Office 97* komplekta un *Lotus Notes 4.0* kalendāra iespējas.

Abi pēdējie programmprodukti pilnībā savu funkcionalitāti nodrošinās, ja birojā uz servera darbosies *Microsoft Exchange* serveris vai *Lotus Notes* serveris. Tālāk apskatītās tīkla funkcijas darbosies tikai pēdējām aprakstītajām konfigurācijām.

Visas šīs sistēmas nodrošina kalendāru. Mēs varam plānot savu laiku ar sapulcēm, tikšanās reizēm, atkārtotiem notikumiem. Šo plānojumu varam apskatīt kā vienas dienas izvērsumā, tā visu nedēļu un pat mēnesi kopumā. Varam to visu izdrukāt.

Varam publicēt un veidot kalendārus lokālā un globālā tīklā, tādējādi publicējot plānotos notikumus, vai veidot to grafiku kopīgi ar kolēģiem.

Piemērs



Truša kalendārs mēneša skatā. Te var ievērot ciklisko notikumu priekšrocības. Tie jāievada tikai vienu reizi, norādot uz to cikliskumu, piemēram, reizi nedēļā vai reizi gadā. Kalendārā var ievadīt konkrētas tikšanās, kā arī atzīmēt dienas notikumus.

Organizējot sapulces, varam apskatīt kolēģu laika izplānojumu, iekārtojot tikšanos tiem brīvā laikā. Turklāt sistēma automātiski izsūtīs viņiem ielūgumus uz sapulci. Ja kolēģis pieņem ielūgumu, ieraksts par sapulci

automātiski parādīsies viņa plānā. Vēl trakāk, mēs ar šīm sistēmām varam rezervēt koferenču telpas un pieejamās iekārtas.

Protams, mēs varam palūgt sistēmu atgādināt par notikumiem, kas tuvojas, piemēram, 15 minūtes pirms sapulces pie šefa vai nedēļu pirms drauga dzimšanas dienas.



Mēs varam palūgt atgādināt par notikumiem, kas tuvojas

Nākošā iespēja ir darbu sarakstu izveide. Mēs varam sekot darbu izpildes termiņiem, iepriekš tos nosakot sev un citiem. Sistēma ziņos par laikā neizdarītiem darbiem. Mēs varam apskatīt šonedēļ darāmos darbus, darbus, kas pabeigti pagājušā mēnesī utt.

Tas viss nav iespējams bez informācijas par darba grupu un citiem visapkārt. Šajās sistēmās tiek iekļauti spēcīgi kontaktu uzskaites līdzekļi. Mēs varam uzkrāt informāciju par draugiem, kolēģiem, klientiem, konkurentiem utt. Mēs varam dažādi grupēt un izdrukāt informāciju, piemēram, varam izdrukāt adreses, lai

izsūtītu Ziemassvētku apsveikumus vai paziņotu par mūsu pēdējo produktu. Protams, mēs varam kolēģiem pārsūtīt informāciju par klientiem pa e-pastu.

Microsoft Outlook un *Lotus Notes* visas šīs funkcijas saista kopā un balsta uz e-pastu. Mums ir iespējas sūtīt elektroniskus ziņojumus, saņemt tos, atbildēt uz tiem, veidot grupu diskusiju. Domājams, ka šī ir vispareizākā pieeja un nākotnē visi laika plānošanas rīki būs integrēti ar e-pastu. Pašreiz notiek darbs pie atklātu standartu izstrādi kalendāriem, lai dažādu ražotāju programmas saprastu viena otru un mēs varētu publicēt savus kalendārus globālos tīklos.

Lai šādas sistēmas kļūtu par ikdienas realitāti, tām ir jāpārvar zināmas problēmas. Pirmkārt, mūsu darba grupai ir jābūt datorizētai un saslēgtai tīklā, kaut arī tā būtu izkaisīta pa visu pasauli. Otrkārt, lai visai šai plānošanai būtu kāda jēga, sistēma ir jālieto konsekventi visur, vienmēr un visiem. Tas nav nemaz tik viegli izpildāms. Pierakstīt uz papīra bieži vien ir daudz ērtāk un ātrāk, biznesa tikšanās bieži vien notiek patīkamākos apstākļos nekā kabinetī. Interesanti būtu vērot, kā restorānos bez mobilo telefonu zvaniem pie galda parādītos arī portatīvie datori. Smieklīgi! Šādu tehnoloģiju saprātīgs pielietojums būtu sistēmas lietošana nepastarpināti ar datoru un izdrukām uz papīra. Darba kārtība bieži vien ir ērtāk plānojama un apspriežama, kad tā ir izdrukāta uz papīra. Arī telefona grāmatu izdrukas ir daudz ērtākas nekā datu bāze. Protams, visa uz papīra pierakstītā svarīgā informācija

savlaicīgi ir jāievada datorā, pretējā gadījumā mēs varam par to aizmirst pilnībā.



... restorānos bez mobilo telefonu zvaniem pie galda parādītos arī portatīvie datori.

Piemērs

Contacts							
	Full Name	Company	E-mail	Business Phone	Business Fax	Home Ph...	Mobile Phone
	Click here to add a...						
	Bebriš Garzobis						
	Hawk Grey	Confidential	hawk@comando...	001 255 1234567			
	Lācis Ķepainis	Meža uzņēmumi SIA	lancis@lmezs.lv	7000001			9000001
	Lapsa Kūmiņš						
	Big Lion	African Fruits Ltd	lion@Afruits.sa				
	Dzenis Raibais						
	Meža Rukšis		Rukšis@lmezs.lv				
	Brālītis Trusītis	Meža uzņēmumi SIA					
	Peļēkais Vāks						

Būdam ļoti organizēts, Trusītis savus biznesa kontaktus un ne tikai tos vien ir sarakstījis Outlook Contacts mapē. Outlook ir jau paredzējis visvisādus iespējamus laukus, ko varētu izmantot informācijas ievadam par personām, ar kurām mums ir darīšana. Protams, var veidot savus laukus, taču vieglāk lietot gatavu modeli. Datus varam ievadīt un aplūkot dažādos skatos. Tabulārā skatā iespējams pievienot, noņemt redzamās kolonnas un mainīt to platumus. Datu ievada skats mums piedāvā četras lapas plašu informāciju par vienu personu. Attēlā redzama pirmā lapa, kuru arī ir iespējams modificēt.

General | Details | Journal | All Fields

Full Name...: Meža Ruksis Job title: Racējs

Company: File as: Ruksis, Meža

Address...: Kur ir ko ēst Lielais mežs Phone: Business Home Business Fax Pager 7777

☒ This is the mailing address

E-mail: Ruksis@lmezs.lv Web page:

Kartupeļi, kāposti Meža Ruksis's Birthday

Categories...

Car Company Home Home 2 Home Fax ISDN Mobile Other Other Fax Pager Primary

Private ☐

Contacts

Garzobis, Bebris
Mitrāajā slapšajā
Lielais mežs

Grey, Hawk
Business: 001 255 1234567
E-mail: hawk@comandos.mil

Keipainis, Lācis
Lielais mežs
Zem celm pie, upītes
Business: 7000001
Mobile: 9000001
E-mail: lacis@lmezs.lv

Kūniņš, Lapsa
Ala zem vecās gobas
Lielais mežs

Lion, Big
E-mail: lion@lfruits.sa

Raibais, Dzenis
Sausais stumbris
Lielais mežs

Ruksis, Meža
Kur ir ko ēst
Lielais mežs
E-mail: Ruksis@lmezs.lv

Trusis, Brālītis
Alpa pie avota
Lielais mežs

Vilks, Pelēkais
Tumšā brīdīņa
Lielais mežs

Piemēram, četros "telefona numuram" līdzīgos laukos pašam iespējams izvēlēties no saraksta attēlojamo lauku, ko piedāvā izvēles poga, kas attēlā ir atvērtā stāvoklī. Komentāru laukā var arī var būt ievietots citas programmas fails, piemēram, Word dokuments. Trusis pagaidām neizmanto kategorizāciju, kas vēlāk var kļūt ļoti būtiska, lai organizētu kontaktpersonas grupās, šķirotu un skatītos tās filtrētos skatos, jo strādājot ar lieliem informācijas daudzumiem, tas kļūst nepieciešams. Izmantojot šādu

informāciju tīklā, daloties tajā ar citiem darba kolēģiem, aktuāls kļūst jautājums par privātās informācijas noslēpšanu no citiem., Protams, arī tas ir paredzēts Outlook programmā. Tiesa, Trusis par Ruksi neko pārāk konfidencialu neraksta, tādēļ kastīte Private nav atzīmēta. Visu šo informāciju varam aplūkot arī pavisam citos skatos. Piemēram, adresu kartīšu skats imitē kartotēku, izdrukājot šo skatu, tas izskatās ļoti jauki.

k

Kūmjis, LapsaAla zem vecās gobas
Lielas mēls

l

Lion, Big

E-mail: lion@fruits.sa

r

Raibais, DzenisSausais stumbrs
Lielas mēls**Rukšis, Meža**Kur ir ko ēst
Lielas mēls
E-mail: Rukšis@mezs.lv

Tasks					
☐	Subject	Status	Due Date	% Complete	Categories
	Click here to add a new Task				
☒	Categories : Business				
☒	Nogatšot šī gada kāpostus	Not Started	None	0%	Business
☒	Barot bērnus par šī gada izdevumiem	Completed	None	100%	Business
☒	Categories : Personīgi				
☒	Nopirkt Vilkam dāvanu	In Progress	Wed 02-Jul-97	50%	Personīgi
☒	Categories : Sabiedriskie p.				

Trusis, protams, izmanto arī uzdevumu mapi Tasks. Konkrētajā skatā ir redzams, kā izmantot kategorijas, lai saorganizētu dažādos ierakstus. Klikšķinot ar peli pa + vai - pogām kategorijas rindā, varam atvērt vai aizvērt šai kategorijai piederošo ierakstu sarakstu. Uzdevumiem ir sava specīfika, kas attēlota to laukos, piemēram, izpildes datums, prioritāte. Uzdevumus iespējams nodot citiem, un tas tiek nodrošināts ar e-pasta palīdzību.

Darba organizācija ar datoru ir realitāte, kas diez vai samazinās stresu. Tā nenoliedzami paaugstinās darba ražīgumu (lasi “ekspluatāciju”). Patīk vai nē, tā ir realitāte, pirmais iedīglis dziļai nākotnes cilvēku un mašīnu integritātei. Iedomājieties mazo datorīņu, kas visu laiku pieslēgts Internet tīklam caur mobilo telefonu ar LCD ekrānu uz kontaktlēcas. Peli klikšķinām, piemiedzot ar aci, hā!



Darba organizācija ar datoru ir realitāte, kas diez vai samazinās stresu.



Grafiskās apstrādes redaktori un paketes

Modernā datortehnika un programmatūra sniedz praktiski neierobežotas iespējas grafisku attēlu veidošanai un apstrādei.

Lietotājs var:

- veidot savus attēlus;
- izmantojot skeneri, ieskenēt plakanus vai pat telpiskus attēlus un apstrādāt tos;
- izmantot gatavus attēlus no pieejamām esošām kolekcijām;
- pārveidot izveidotos attēlus utt.

Izvēloties programmproduktu savu ieceru realizācijai, jums vispirms ir jāuzstāda sev jautājums: kādiem mērķiem man tas būs vajadzīgs? Ja jūs vēlaties strādāt profesionāli, veidojot sarežģītus attēlus, tad izvēlētajai programmai jābūt pietiekoši komplicētai, lai atbilstu jūsu

vajadzībām un prasībām. Turpretī, ja jūsu prasības pret gala produktu nav tik lielas, tad jūs mierīgi varat samierināties ar ne tik spēcīgiem rīkiem. Sīkāk aplūkosim vispirms otro variantu.

Tātad jūs nevēlaties strādāt kā profesionāls mākslinieks vai dizainers, bet jums šad tad nepieciešams bagātināt savu ikdienu ar kādu jauku bildīti. Tad pirms ķerties pie grūtā darba, visvienkāršākā izeja, radot jaunu šedevru, ir apskatīties, vai kādā no piedāvātajām attēlu kolekcijām nav pieejams attēls, kas jums derētu.

Kas ir attēlu kolekcijas? Dažiem programmproduktiem līdzī ir mākslinieku izstrādātu attēlu kolekcija, kurā atrodami attēli daudziem dzīves gadījumiem. Tā, piemēram, *Windows*, *MS Word*, *Visual Basic* atsevišķā direktorijā *Clipart* jūs atradīsiet (bieži vien vienādus, bet) ļoti jaukus attēlus. To lietošanai netiek noteikti autortiesību aizsardzības aizliegumi, un jūs jebkuru no tiem attēliem varat brīvi izmantot. Taču te ir viens bet! Neaizmirstiet, ka *Windows* produkciju izmanto miljoniem cilvēku. Diemžēl pasaule ir pārāk maziņa, un agri vai vēlu kāds noteikti pamanīs, ka esat izmantojis vispārpieejamu kolekciju. Ja tas aizskar jūsu pašcieņu, tad izmantojiet kādu no grafiskajiem redaktoriem sava attēla radīšanai.

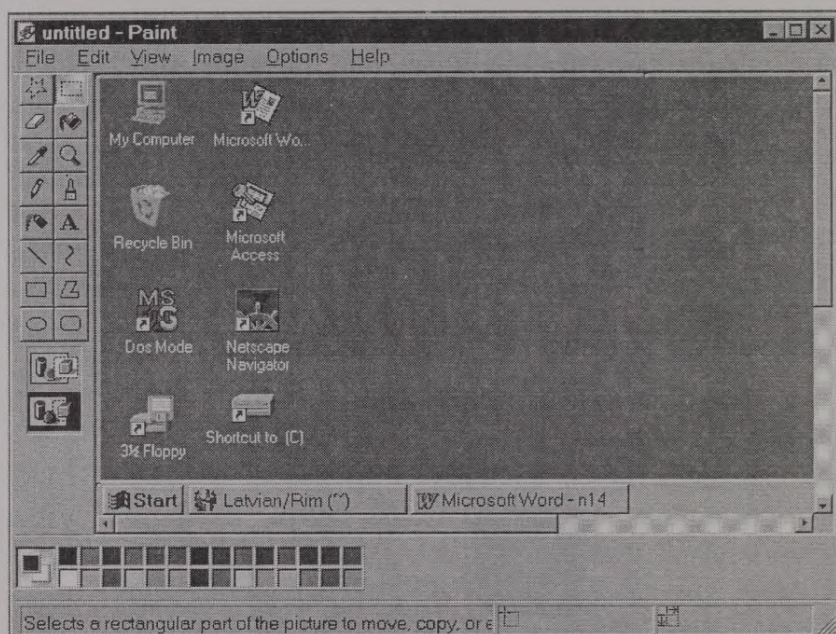
Grafisko redaktoru ir ļoti daudz, taču tikai nedaudzi ir pazīstami un plaši lietoti. Slavenākās un labākās grafisko redaktoru ražotājfirmas ir *Corel*, *Adobe*. Katrā *Windows* versijā ir pieejams arī tāds redaktoriņš *Paintbrush* (nejauciet ar daudz spēcīgāko MS-DOS variantu). Tas ir

Ļoti vienkāršs un diezgan primitīvs, bet labi pielietojams dažu darbu izpildei. Piemēram, ja jums vajag "noķert" displeja ekrānu, t.i., izveidot grafisku attēlu, kurā būtu iemūžināts kāds jūsu ekrāna stāvoklis, tīri labi varat izmantot *Paintbrush* programmiņu. Iegūstiet uz ekrāna to stāvokli, kuru vēlaties saglabāt, un nospiediet tastatūras pogu *Print Screen*. Tagad dators atmiņā ievieto ekrāna stāvokli. Atveriet programmu *Paintbrush* un nospiediet pogas *Ctrl+V*. Datora atmiņas saturs parādīsies *Paintbrush* darba logā. Pēc tam to var rediģēt, saglabāt un izmantot kā attēlu.



... ja jums vajag "noķert" displeja ekrānu ...

Diemžēl, vecākās *Paintbrush* versijas (*Windows 3.1+* vidēs) spēj apstrādāt tikai to ar *Ctrl+V* iekopēto attēla daļu, kas neiziet ārpus *Paintbrush* loga.



No konkrētas programmas *Paintbrush* aplūkošanas atgriezīsimies pie vispārīgākām lietām - attēlu formātiem.

Kad jūs redzat uz ekrāna kādu glītu (vai neglītu) attēlu, jums varbūt ienāk prātā jautājums: kā to visu var saglabāt failā tā, lai, atverot failu, atkal iegūtu to uz ekrāna tieši tādā pašā izskatā, jo katrs attēls sastāv no neskaitāmiem punktiem, un katram punktam ir sava krāsa un konkrēta atrašanās vieta. Divu punktu samainīšana vietām varbūt arī pārāk stipri neizkropļos

visu attēlu, bet ja šādi sajuku visi punkti, tad no attēla nekas daudz pāri nepalikis vis.

Ir izdomāti dažādi attēlu saglabāšanas veidi - formāti. Katram formātam ir savs nosaukums, kuru lietotājam un zināmā mērā arī grafiskajam redaktoram norāda faila paplašinājums. Piemēram, *Paintbrush* failu formāts ir Bitmap (.BMP).

Daži no pazīstamākajiem grafisko failu formātiem ir:

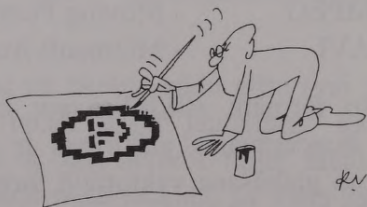
JPEG (JPG)	- JPEG File Interchange Format
GIF	- Graphics Interchange Format
TIFF (TIF)	- Tag Image File Format
CDR	- CorelDRAW
BMP	- Windows Bitmap
WMF	- Windows Meta File
PSD	- PhotoShop Document
AI	- Adobe Illustrator
EPS	- Encapsulated PostScript
CGM	- Computer Graphics Metafile
3DS	- 3D Studio
MPEG	- Moving Picture Expert Group
AVI	- Microsoft Audio Video Interleave

Visu šo formātu atšķirības var būt pat ļoti būtiskas. Divas konceptuālās atšķirības ir glabāšana punktuālā formā un glabāšana vektoriālā formā. Pirmajā gadījumā var veidot un glabāt ļoti izsmalcinātus attēlus. Otrajā gadījumā attēli nav tik smalki, bet toties tos var veidot ar lielāku precizitāti. Turklāt vektoriālā formā apstrādājamus attēlus var viegli samazināt un palielināt. No tā praktiski necieš attēla kvalitāte, turpretī pirmajā

gadījumā kvalitātes zudumi ir būtiski. Tāpēc, izvēloties redaktoru, ir jānovērtē veicamā darba nepieciešamā kvalitāte un tā pielietojums. Veidojot vektorgrafikas attēlus, jārēķinās ar to, ka lielāko daļu mākslas ir jāveic pašam, turpretī punktu grafikas redaktoriem ir izstrādāti tā sauktie filtri, kas piešķir attēlam specifisku efektu. Aplūkosim trīs piemērus:

Trusītis Trusītis Trusītis

Pirmajā piemērā ir izmantots parasts fonts. Otrajā gadījumā tam tika uzstādīti parametri, lai pazustu robainās maliņas. Grafiskais redaktors pats pieliek nepieciešamos pustoņus tā, lai robainums izzustu.



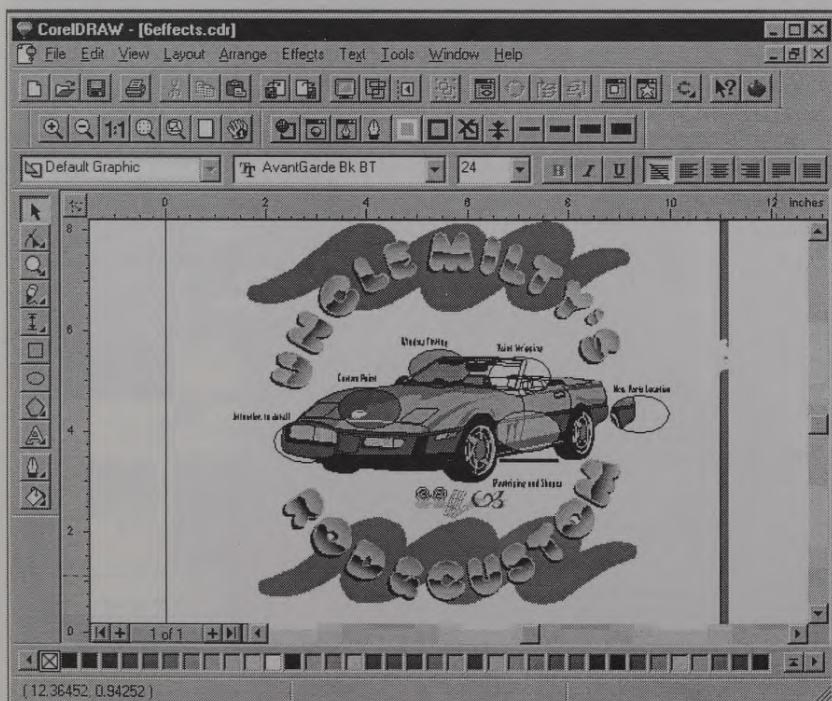
Grafiskais redaktors pats pieliek nepieciešamos pustoņus.

Trešajā piemērā parastie fonti tika apstrādāti ar diviem filtriem: pirmais padarīja tekstu telpisku, bet otrais pielika ēnu, tādējādi pastiprinot telpiskuma ilūziju.

Lai noteiktu, vai programma glabā attēlus punktu vai vektoru formātā, pietiek paskatīties uz grafiskās programmu paketes interfeisu.

Piemēram, aplūkosim divus grafiskos redaktorus: vienu no populārākajiem vektorgrafikas redaktoriem *Corel Draw* un vienu no populārākajiem (ja ne vispopulārāko) punktu grafikas redaktoriem *Adobe PhotoShop*.

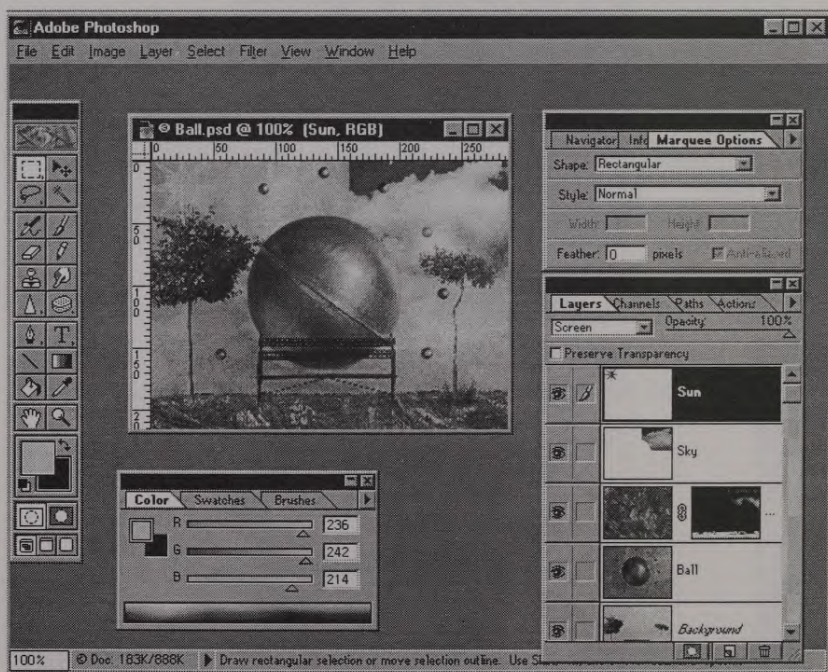
Corel Draw logs varētu izskatīties šādi:



Kā redzat, tā rīki jau paši par sevi izskatās ģeometriski — dažādas ģeometriskas figūras, telpiskuma elementi,

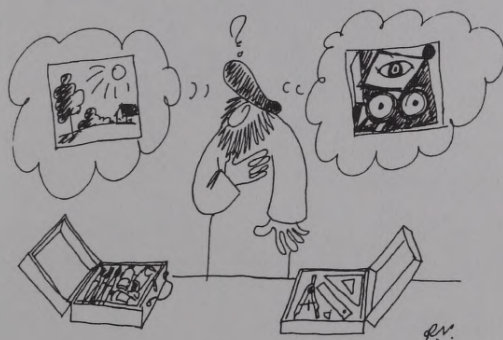
transformēšanas rīki utt. Arī veidotais zīmējums ir bez īpašiem izskaistinājumiem, kaut gan arī ar *Corel Draw* var izveidot smalkus un efektīgus attēlus. *Corel Draw* attēlu bibliotēka, kas nāk līdzī produktam, ir ļoti plaša. Tajā var atrast grafiskus attēlus praktiski katram dzīves gadījumam.

PhotoShop programmas rīki drīzāk izskatās kā parasti mākslinieku piederumi — otas, zīmuļi, dzēšgumijas.



Arī veidotie attēli ir mākslinieciski smalki. Tos iespējams bagātināt ar dažādiem grafiskiem efektiem un elementiem.

PhotoShop programmai līdzīgi nenāk attēlu bibliotēka, toties tā ir aprīkota ar bagātīgu dažādu filtru klāstu, kas var tikt papildināti ar citu firmu veidotiem filtriem.

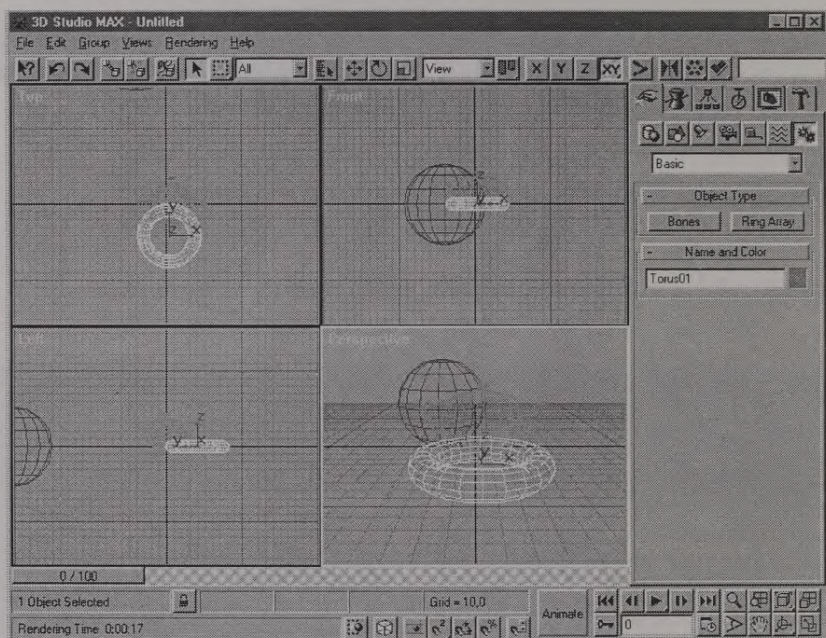


Lai noteiktu, vai programma glabā attēlus punktu vai vektoru formātā, pietiek paskatīties uz tās interfeisu.

Pavisam savādāk izskatās telpiskie redaktori, piemēram, *3D Studio* (gan *MS-DOS*, gan *Windows* videi).

Telpiskajā redaktorā veidoto attēlu var apskatīt no dažādām pusēm, kā arī telpā (perspektīvā). Ar kameru palīdzību attēlus var apskatīt no visām pusēm plūstoši, radot objekta rotēšanas vai pārvietošanās efektus.

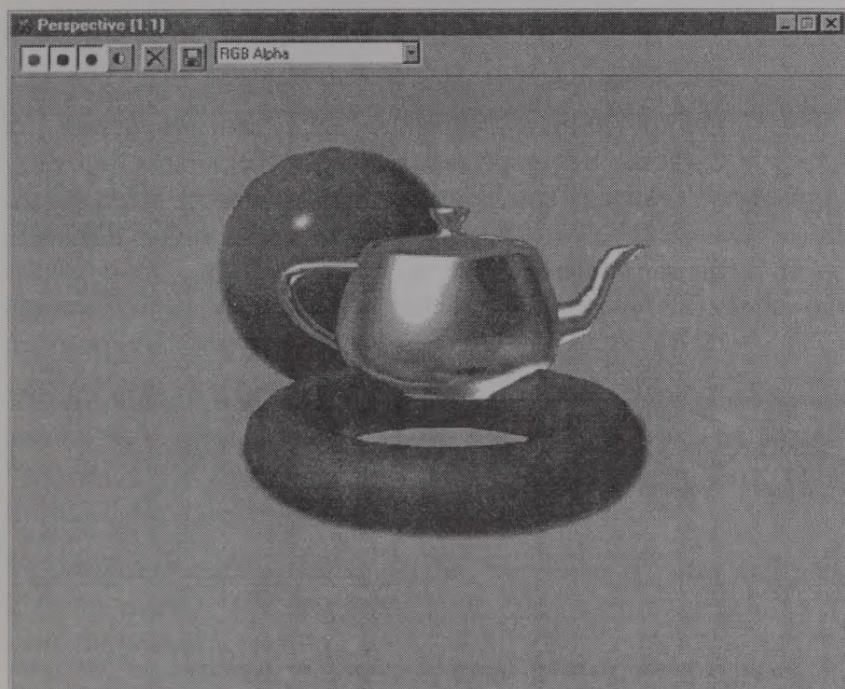
Telpiskie redaktori operē ar telpiskiem objektiem un par katru objektu veidojošu punktu glabā informāciju ar tā koordinātēm telpā.



Objektu virsmas ir iespējams gan iekrāsot, gan apspīdināt no dažādiem telpas punktiem, gan piešķirt tām kāda materiāla, piemēram, granīta, metāla vai koka izskatu.

Protams, darbošanās ar šādu redaktoru prasa zināmas iemaņas un telpisko domāšanu.

Piemērā izveidotais attēls mums pierastākā izskatā ir šāds:



Vienu un to pašu attēlu saglabājot vienā vai citā formātā, var tikt izmainīts gan faila izmērs, gan kvalitāte, gan krāsu palete, gan pielietojuma varianti. Izvēloties formātu attēla saglabāšanai, jāzina, kādam nolūkam jūs vēlaties to vēlāk izmantot. Piemēram, jūs vēlaties to apstrādāt ar kādu citu grafisko redaktoru, tad jāzina, kādus formātus "pieņem" tas redaktors. Ja pieņemamie formāti nav norādīti *Open* logā, tad meklējiet, kādus formātus tajā var importēt (*Import*, parasti *File* izvēles

kartē). Tāpat no redaktora var eksportēt arī attēlus citos formātos.

Punktu formātā saglabātie attēli izskatās dažādi uz dažādas izšķirtspējas iekārtām. Piemēram, 600x800 SVGA pilna ekrāna attēlu, izdrukājot to uz lāzerprintera ar izšķirtspēju 600 punkti kvadrātcollā, sanāks tikai tāda maza pastmarka vien.



Telpiskie redaktori operē ar telpiskiem objektiem un par katru objektu veidojošu punktu glabā informāciju ar tā koordinātēm telpā.

Palielinot punktu attēlu, tas kļūst “robains”, jo attēla palielināšanas algoritmi vienkārši dubulto vai četrkāršo punktus. Slīpas līnijas kļūst līdzīgas trepēm. Turpretī, samazinot punktu attēlu, tiek “nesaprātīgi” mestas ārā rindiņas un kolonnas ar punktiem, tā ka atsevišķas detaļas mēs varam pazaudēt pavisam.

Mainot punktu attēla izmērus, attēlā parādās optiskie efekti, piemēram, dažādi riņķi vai viļņi. Taču punktu attēli ir ērti izmantojami vienkāršos gadījumos, jo ar tiem ir viegli strādāt, to apstrādes programmas ir pieejamas un lētas.

Uzmanību - PADOMS!

Profesionāli punktu formātu attēli jāizmanto, apstrādājot fotogrāfijas. Lai varētu veidot kvalitatīvus liela izmēra attēlus, ir nepieciešams ļoti ātrs dators ar lielu atmiņas un cietā diska apjomu. Tāpat vēlams arī optiskais disks, lai attēlu varētu tur aiztransportēt.

Vektorformas attēlus ieteicams izmantot, ja attēls veidots ar datoru. Šāds attēls nezaudēs detaļas pēc samazināšanas un nekļūs "robains" pēc palielināšanas. Vektorformu attēls būs tādā kvalitātē, kādu pieļauj iekārta, kas to attēlo.

Vektorformu attēls ir kā aplikācija, kur viens uz otra uzlīmēti krāsaini papīriši, bet mēs jebkuru no tiem varam pamainīt, pat izmainīt krāsu vai tekstūru.

Ar vektorgrafikas paketi var ļoti ātri izveidot jaukus attēlus. Tā ir patīkama rotaļa gan lieliem, gan maziem, taču, lai attēls izskatītos profesionāli, ir vajadzīgs profesionāls mākslinieks, kas prot strādāt ar datoru.

Strādājot ar krāsainiem attēliem, svarīga ir izvēlēta krāsu palete. Aplūkosim dažas no tām:

1) **melnbaltā palete** (*Bi-tone*) - izmanto tikai melno un balto krāsu, nekādu nokrāsu, nekādu pustoņu;

2) **pelēkās skalas palete** (*Greyscale Palette*) - tiek saglabāta visa krāsainā attēla krāsu atšķirību intensitāte, tikai viss tiek attēlots ar pelēkās krāsas palīdzību - no baltās līdz melnajai;

3) **indeksētās krāsas** (*Indexed Colors*) - attēlā izmantoto krāsu palete sastāv no 256 krāsām. Samazinot krāsu skaitu, būtiski tiek ietekmēts faila apjoms;

4) **RGB** (*True color*) - visu pieejamo krāsu palete;

5) **CMYK** - krāsu palete, ko izmanto poligrāfijā.

Plašāk izmantotie grafiskie redaktori:

Paintbrush (Windows vidē), **punktu redaktors**

Vājš redaktors bez spēcīgiem rīkiem attēla apstrādāšanai. Ērti izmantojams neliela apjoma vienkāršu attēlu veidošanai, kā arī displeja attēla "ķeršanai". Neļauj izvērsties ar krāsu lietošanu, jo izmanto tikai 16 pamatkrāsas.

Paintbrush IV Plus (MS-DOS videi), **punktu redaktors ar vektorgrafikas iezīmēm**

Parādījies apmēram 1988. - 1989. gadā. Pašlaik praktiski netiek izmantots, jo skaitās novecojis, kaut gan tā izmantošanas iespējas bija pietiekoši plašas.

CorelDraw, **vektorgrafikas redaktors**

Ļoti spēcīgs redaktors. Ļoti parocīgs ģeometriski precīzu attēlu, piemēram, logotipu, veidošanai. Nav ērts smalku attēlu veidošanai. Strādā ar atsevišķiem objektiem.

Corel PhotoPaint, **punktu redaktors**

Nav paredzēts precīzu ģeometrisku attēlu veidošanai, bet ir pietiekoši ērts smalkākiem attēliem. Ir apgādāts ar noderīgiem filtriem.

Adobe PhotoShop, **punktu redaktors**

Nav piemērots ģeometrisku attēlu apstrādei. Ļoti ērts smalku attēlu veidošanai. Apgādāts ar lielisku filtru kolekciju, kas var tikt papildināta. Strādā ar daudzām paletēm. Parocīgs ieskenētu fotogrāfiju apstrādāšanai

Adobe Illustrator, **vektorgrafikas redaktors**

Līdzīgs CorelDraw redaktoram, ir paredzēts ģeometrisku objektu modelēšanai un apstrādei.

3D Studio, **telpiskais redaktors**

Redaktors, kas paredzēts telpisku objektu modelēšanai. Plašas iespējas objektu rotācijas veidošanai.

3D Studio MAX, **telpiskais redaktors**

Pēc savas būtības tā ir *3D Studio* redaktora versija *Windows* videi. Tādēļ šis redaktors līdz ar *3D Studio* plašajām (un uzlabotajām) iespējām ir ieguvis *Windows* vides programmām raksturīgo ērto interfeisu.

The first of these is the fact that the
economy has been growing at a rapid rate
and that the government has been able to
maintain a high level of employment.

The second is the fact that the
government has been able to maintain a
high level of employment and that the
economy has been growing at a rapid rate.

The third is the fact that the
government has been able to maintain a
high level of employment and that the
economy has been growing at a rapid rate.

The fourth is the fact that the
government has been able to maintain a
high level of employment and that the
economy has been growing at a rapid rate.

The fifth is the fact that the
government has been able to maintain a
high level of employment and that the
economy has been growing at a rapid rate.

The sixth is the fact that the
government has been able to maintain a
high level of employment and that the
economy has been growing at a rapid rate.



Internet un tā pielietojuma iespējas

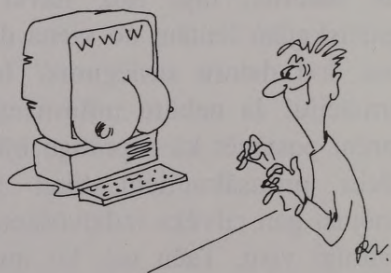
Droši vien cilvēkam, kas pirmais iedomājās saslēgt kopā divus datorus, bija līdz nāvei apnicis skraidīt ar magnētiskajām lentām no viena datora pie otra. Ko tad mums dod datoru saslēgums? Iespēju apmainīties ar informāciju! Ja nebūtu informācijas apmaiņas, cilvēks nevarētu pastāvēt kā saprātīga būtne. Jau kopš pašiem cilvēces pirmsākumiem tikai iespēja sazināties ir sekmējusi gan cilvēka izdzīvošanu, gan attīstību, gan ... jā, pilnīgi visu. Taču tas, ko mums sniedz datortīkli, pārspēj visfantastiskākos sapņus, kas radušies pēc informācijas izslāpušo cilvēku domās vēl pirms kādiem desmit gadiem.

Kā jau minējām nodaļā *Datoru tīkli*, ir divu veidu datortīkli: LAN (*Local Area Network*) - lokālais jeb vietējais tīkls un WAN (*Wide Area Network*) - globālais tīkls, kas apvieno datorus plašā teritorijā.

Pazīstamākais globālais datortīkls ir Internet. Tas radās 80.-to gadu vidū ASV, un, protams, savu roku tur ir

pielikušas militārās struktūras, kuras sākotnēji šo datortīklu veidoja kā slēgtu tīklu savām vajadzībām. Toreiz gan tas netika saukts par Internet, taču ideja attīstījās un pārtapa par globālu datortīklu - visdemokrātiskāko iekārtu pasaulē, jo:

1. Internet nevienam nepieder;
2. kaut arī Internet atsevišķas daļas uztur kādas iestādes, faktiski tam nav konkrēta apmaksātāja, kas varētu diktēt noteikumus;
3. Internet izmanto gan jauni, gan veci, gan biznesmeņi, gan studenti, gan ticīgie, gan ķecerī, gan ...

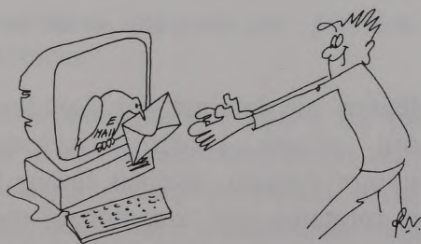


Tāpēc Internetā jūs starp absolūti nevajadzīgas informācijas plūdiem atradīsiet arī bezgala noderīgas lietas

Ja neskaita visas pārējās iekārtas, demokrātija ir vissliktākā iekārta, ko cilvēki ir radījuši. Cilvēce savu tūkstošu gadu attīstībā kopš Noasa laikiem nemaz nav mainījusies, tai vēl aizvien ir raksturīgas galējības. Tāpēc Internetā jūs starp absolūti nevajadzīgas informācijas plūdiem atradīsiet arī bezgala noderīgas lietas. Taisnības

labad jāpiebilst, ka nevajadzīgās informācijas būs krietni vien vairāk, taču, ja vien prot meklēt, parasti izdodas atrast vajadzīgo informāciju.

Daudzi cilvēki, izdzirdot pieminam Internet, asociē to ar elektronisko pastu jeb *e-mail* (angliskā izruna "ī-meil"). Latviski to mēdz saukt par e-pastu. Šāda asociācija reizē gan ir, gan nav korekta. Patiešām, e-pasta sakari var tikt nodibināti un bieži vien arī tiek nodibināti caur Internet, bet Internet ir kaut kas daudz plašāks nekā vienkārši e-pasts. Vienlaikus e-pasta tīkls, ja varam to tā nosaukt, ir daudz plašāks nekā Internet. Skan absurdi, vai ne? Taču tā nu tas ir, jo Internet tīklam ir arī citi pielietojumi ne vien e-pasts, bet daudzi jo daudzi lietotāji izmanto vienīgi e-pastu un par citām Internet iespējām nemaz nav dzirdējuši.



Daudzi cilvēki Internet asociē ar elektronisko pastu.

Kā jau minējām, Internet ir globāls datortīkls. Tiem, kam šis jēdziens ir jauns un neizprotams, salīdzinājumam var minēt pasaules telefonu sakarus. Normālā gadījumā ir nodrošināts, ka no jebkuras vietas, kur ir telefonu tīklam pievienots telefons, var piezvanīt jebkuram citam telefona tīkla abonentam. Telefona aparāti, kabeļi u.c.

iekārtas un ierīces var arī atšķirties, taču informācijas apmaiņai ir jānotiek. Tāpat notiek informācijas apmaiņa arī starp diviem datoriem, kuri ir pieslēgti Internet. Dati elektronisku signālu vai viļņu veidā pa parastiem telefonu kabeļiem, pa optiskiem kabeļiem, pa satelītkanāliem vai citādi tiek pārraidīti no viena datora uz otru un atpakaļ.



Pieslēgties pie Internet var dažādos veidos, visur, kur piejami sakaru kanāli.

Pieslēgties pie Internet var dažādos veidos, visur, kur piejami sakaru kanāli. Visvienkāršāk to izdarīt, izmantojot parasto telefona līniju, kādas ir mājās vai birojā (protams, var izmantot arī mobilo telefonu, taču tas šo procesu sadārdzina). Lai pa telefona līniju pieslēgtos pie Internet, ir vajadzīgs dators, telefona līnija, modems un Internet pakalpojumu sniedzējs (IPS), kurš jūs konsultēs, piešķirs identifikācijas vārdu tīklā un apgādās ar nepieciešamajām programmām. Caur modemu savienojoties ar IPS, jūs pieslēdzaties pie Internet. Uz savienojuma brīdi jūsu dators kļūst par Internet tīkla sastāvdaļu, taču atslēdzoties tas atkal ir

vienkāršs dators. Tas notiek tādēļ, ka jūs savienojumam izmantojat telefona sakarus. Lai jūs savu datoru varētu uzskatīt par pastāvīgu Internet sastāvdaļu, tas ir jāpieslēdz tīklam ar kabeli vai radiosavienojumu, taču tas ir krietni vien dārgāk, un parasts lietotājs bez tā var gluži labi iztikt.

Atgriezīsimies pie e-pasta, kas ļoti noder gan darījumu cilvēkiem, gan vadītājiem, gan studentiem, gan zinātniekiem. Ja esat pieslēdzies pie Internet un saņēmis savu identifikācijas vārdu, kāds nav nevienam citam visā plašajā pasaulē, un ko sauc par e-pasta adresi, tad, zinot cita cilvēka e-pasta adresi, varat nosūtīt viņam vēstuli.

Informācijas pārsūtīšana pa tīklu nav droša, jo pietiekoši ziņkārīgs un prasmīgs cilvēks var pārtvert un izlasīt jūsu vēstījumu. Ar to ir jāreķinās, tad vai nu ir jāsamierinās, vai arī slepenākas informācijas pārraidei jāizmanto šifrēšanas metodes.

Kāpēc elektroniskais pasts ir kļuvis tik ļoti populārs? Vienkārši tas ir vislabākais kompromiss starp ātrumu un cenu. Informācijas apmaiņa pa e-pastu ir lēnāka nekā ar telefona starpniecību, taču nesalīdzināmi lētāka (it īpaši sarakstoties ar ārzemēm).

Salīdzinājumā ar parasto pastu, e-pasts kaut arī sanāk dārgāks, tomēr ir nenoliedzami ātrāks. Vienīgā parastā pasta priekšrocība ir tā, ka pa to var nosūtīt, piemēram, parakstītu un apzīmogotu dokumentu, ko arī atzīs kā oriģinālu dokumentu it visur. To pagaidām nav iespējams nosūtīt elektroniskā veidā, kaut arī zināmi soļi uz priekšu ir sperti arī šajā lauciņā. Šodien ir iespējams šifrēt

vēstules un likt elektroniskus parakstus firmas iekšienē, kā arī darījumos ar zināmiem partneriem. Tas noteikti ir drošāk nekā parastā pasta pakalpojumi. Domājams, ka drīzumā dzirdēsim par elektroniskajiem notāriem, kas mums dos iespēju veidot publisku apmaiņu ar oriģināliem dokumentiem.

Savas vēstules varat sūtīt arī uz diskusiju grupām jeb vēstkopām, kurās cilvēki apspriež dažādus jautājumus (jautājumu loku var noteikt pēc diskusiju grupas nosaukuma). Šajā gadījumā jūsu vēstuli saņems visi tie, kas ir pierakstījušies uz diskusiju grupu, kurai jūs sūtāt savu vēstuli. Latvijā pirmā un viena no populārākajām vēstkopām ir **sveiks.latnet.lv**.

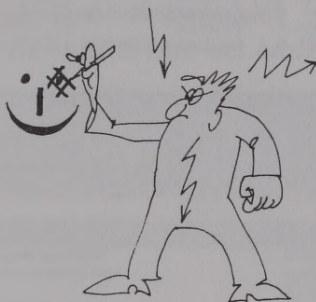
Iesaistoties sarakstē pa e-pastu, jāņem vērā, ka šajā vidē ir izveidojušās savas tradīcijas, apzīmējumi un stils. Visvājākā vieta iesācējiem ir tīklā pieņemto apzīmējumu lietošana. To nepārvaldot, reizēm var pat pārprast rakstītā jēgu.

Visbiežāk lietotie apzīmējumi savas attieksmes paušanai ir:

:-) un :-(

Pieliecot galvu pie kreisā pleca, jūs redzēsiet, ka tie izskatās kā priecīga un noskumuša cilvēciņa ģīmīši. Tā tas arī ir. Rakstot par nenopietnām lietām vai jokojo, nespējot paust to ar savu intonāciju, rakstītājs liek zīmi :-)/iespējami varianti ;-):); :-|/. Rakstot par nopietnām lietām vai jūtot skumjas, rakstītājs izmanto zīmi :-(/iespējami varianti ;-(:(;(/ Emocionālam

pastiprinājumam var izmantot atkārtojumus :-)))))) vai :-
 ((((((



Visvājākā vieta iesācējiem ir tīklā pieņemto apzīmējumu lietošana.

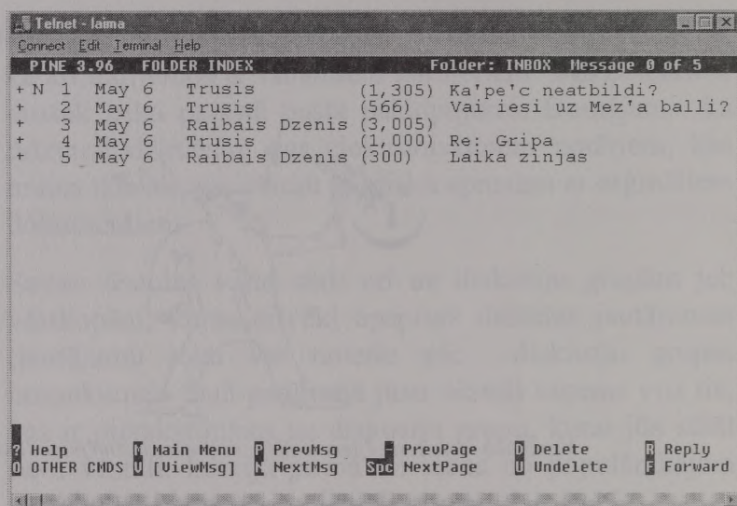
Plaši tiek lietoti saīsinājumi no angļu valodas:

BTW - By the Way (starp citu)

ASAP - As Soon As Possible (cik vien ātri iespējams)

IMHO - In My Humble Opinion (es domāju)

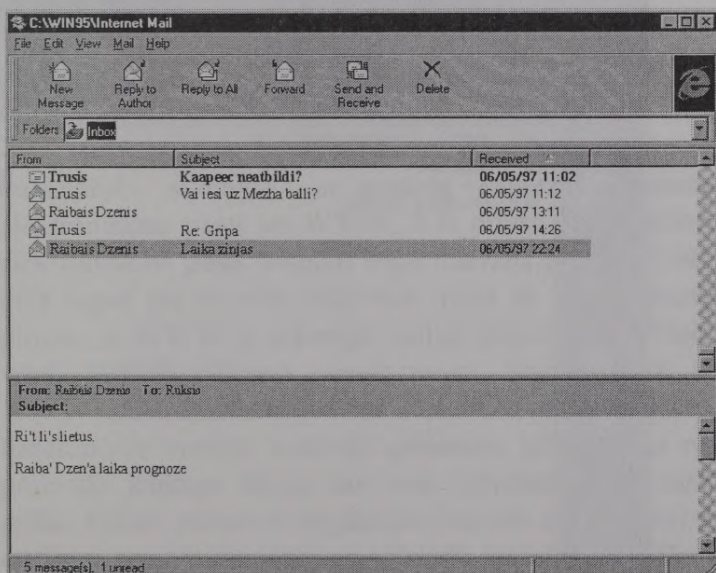
Elektroniskā pasta lasīšanai ir izstrādātas speciālas programmas. Pieslēdzoties pie sava konta ar telnet komandu, tiek nodrošināta interaktīva datu apmaiņa. Šajā režīmā pastu var lasīt ar komandu mail vai ar daudz ērtāku programmu pine.



Taču telnet pieslēgumam ir liels mīnuss - visā darba procesā ir aktīvs pieslēgums tīklam. Tas nozīmē, ka par katru darba minūti šajā režīmā ir jāmaksā. Tāpēc labāk izmantot vēstuļu savākšanas programmas, kas pieslēdzas jūsu pastkastei un “pārvelk” jaunpienākušās vēstules uz jūsu datora cietā diska. Tad, lieki netērējot savus līdzekļus, jūs tās varat lasīt, rakstīt uz tām atbildes (tad atkal pieslēgties IPS, lai atbildes nosūtītu), veikt izdrukas utt. Ir izplatītas dažādas brīvi pieejamas e-pasta ziņojumu lasīšanas/sūtīšanas programmas, piemēram, *Eudora*, *Pegasus Mail*. Tās darbojas *Windows 3.+* vidē. Parastas DOS programmas praktiski neizmanto elektroniskajam pastam to neērtā interfeisa dēļ. *Windows 95* vidē ir iebūvēta telnet un ftp aktivizēšana.

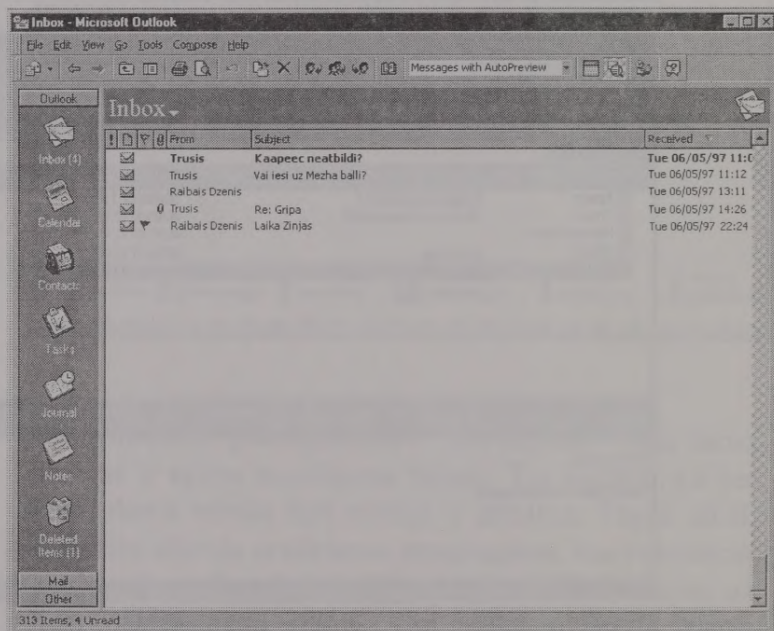
Protams, var izmantot *Microsoft Exchange* vai *Lotus Notes* ar to plašajām papildu iespējām.

Populāra programma elektroniskā pasta ziņojumu lasīšanai ir arī *Microsoft® Internet Mail*.



Lielākie pielietojamprogrammu izstrādātāji ir sapratuši, ka programmu paketēs ir jāiekļauj arī Internet iespējas. Tā, piemēram, *Microsoft® Office 97* līdzekļi piedāvā *Outlook* komponenti ar rūpīgi izstrādātu elektroniskā pasta ziņojumu sūtīšanas sistēmu. *Outlook* ir uzlabota *Microsoft Scheduler+* laika plānošanas un kontaktinformācijas glabāšanas sistēma ar elektroniskā pasta iespējām.

Outlook programma saņem un nosūta e-pasta ziņojumus, ļauj nosūtīt tos adresu grāmatā ierakstītajiem kolēģiem un draugiem, sakārtot ienākušos ziņojumus, meklēt vajadzīgās vēstules utt.



Pieslēdzoties Internet tīklam, jāņem vērā, ka jūsu adrese var iekļūt masveida e-pasta ziņojumu izsūtītāju sarakstos (no tiem gan var atrakstīties, aizsūtot atbilstošu ziņojumu). Tas nozīmē, ka jūsu pastkastīte ik dienas var tikt pārplūdināta ar dažādām vēstulēm. Labas e-pasta programmas sašķiros ienākušos ziņojumus. Protams, lai tās spētu to izdarīt, jums jānorāda, kādā veidā šķirošanu veikt.



Labas e-pasta programmas sašķiros ienākušos ziņojumus.

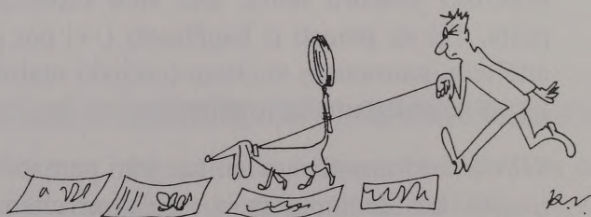
Pašreizējā situācijā ar vārdu Internet saistās elektroniskais pasts un WWW. Vēl pirms pāris gadiem elektroniskais pasts noteikti būtu minams pirmajā vietā, taču tagad jau ir grūti izšķirties, kuru no abiem minēt pirmo, jo WWW ir iekarojis milzu popularitāti. Vienā vārdā WWW rakturot nemaz nevar. Varētu teikt, ka WWW (saīsinājums no *World Wide Web* jeb vispasaules tīmeklis) ir vienotā sistēmā apvienota informācija par praktiski jebkuru tēmu, kas vien cilvēkam var ienākt prātā. Teksts parasti ir bagātināts (vai pat pārsātināts) ar attēliem, animētiem attēliem (cikliski mainīgām bildēm), kā arī ar video un audio efektiem.

WWW informatīvā vienība jeb pamatšūniņa ir lapa (*page*), kas ar norādēm ir saistīta ar citām lapām. Pašu galveno lapu, no kuras aiziet saites uz otrā līmeņa lapām, sauc par mājas lapu (*Home Page*). Internet tīklā šādas lapas ir atrodamas visās pasaules malās. Lai tās varētu caurskatīt un iegūt informāciju, ir jāapgūst divas lietas: darbošanās ar hipertekstu un ar programmu, kas ļauj pieslēgties un skatīties lapas, un kuru sauc par pārlūkprogrammu vai vienkārši par pārlūku. Gan vienu,

gan otru apgūt ir ļoti vienkārši, un droši vien ar to arī ir izskaidrojama WWW lielā popularitāte dažādu profesiju pārstāvju vidū, jo "braukāt" pa WWW un ielūkoties pasaules visnomaļākajos kaktos (kur vien ir iespiedies Internet, protams) ir vienkārši, ērti un, galvenais, interesanti.

Katru mēnesi pasaulē klāt nāk aptuveni miljons jaunu lietotāju un parādās tūkstošiem jaunu lapu. Arī Latvijā cilvēki sāk apjaust Internet un WWW milzīgās iespējas.

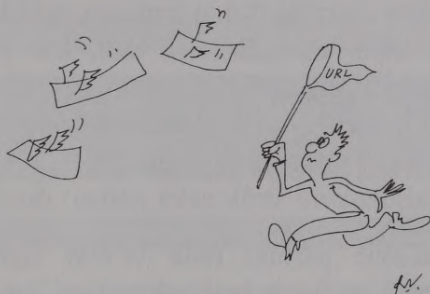
Navigāciju pa WWW nodrošina tā saucamais hiperteksts. Hiperteksta dokuments ir vienkāršs teksts, kas iezīmēts ar speciāliem marķējumiem tā, ka, nostādot uz tā peles norādi, pēc klikšķa atveras kāda cita lapa. Hiperteksts ir pazīstams katram, kas *MS Windows* vidē ir lietojis palīginformāciju no *Help* failiem.



Lai tās varētu caurskatīt un iegūt informāciju, ir jāapgūst darbošanās ar hipertekstu un ar programmu, ko sauc par pārlūku.

Katrai WWW lapai Internet tīklā ir sava adrese, pēc kuras jebkurš, kam ir pieeja Internet un atbilstošā programmatūra WWW caurskatei, ko sauc par pārlūkprogrammu jeb vienkārši par pārlūku, šo lapu var

viennozīmīgi atrast. Šādu adresi sauc par URL (*Uniform Resource Location*). Nav divu dažādu lapu ar vienu adresi.



Katrai WWW lapai ir sava adrese, ko sauc par URL,

Adresi veido WWW servera IP adrese, kurai var būt piesaistīts domēna vārds, plus pilns ceļš līdz šai lapai. Par adresēm mēs jau runājām nodaļā *Datoru tīkli*. Iegaumēt IP adreses principā nav sarežģīti, it īpaši pēc tam, kad ir skaidrs, pēc kāda principa šīs adreses tiek iedalītas, tomēr cilvēkam daudz vieglāk ir iegaumēt IP adresei piekārtoto vārdisko nosaukumu. Nu, jā, normālam cilvēkam adreses 159.148.108.129 vietā vieglāk atcerēties www.latnet.lv.

WWW informācijas kodēšanu hiperteksta veidā veic ar iezīmju valodas HTML (*HyperText Mark-up Language*) palīdzību. Kodēšana ir jāveic tādēļ, lai pārlūki izveidoto lapu varētu parādīt atbilstoši tās veidotāja vēlmēm. Cilvēkam, kas grib paklejt pa WWW resursiem, HTML kodēšanas zināšanas absolūti nav vajadzīgas.

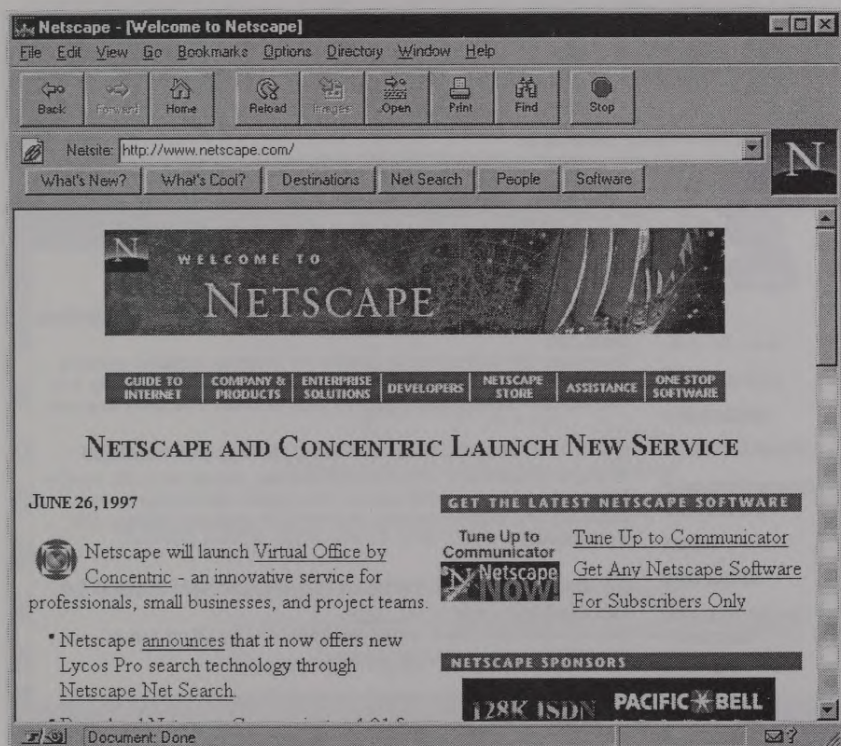
Ar HTML marķējumiem iezīmētu tekstu pa Internet pārsūta ar HTTP (*Hypertext Transfer Protocol*) protokola starpniecību.

WWW caurskatīšanai izmanto dažādus pārlūkus. Tie var būt tikai tekstuāli, t.i., tādi, kas ļauj caurskatīt tikai tekstuālu informāciju un nav spējīgi rādīt attēlus. Tie ir labi izmantojami informācijas ieguvei, jo ļauj ekonomēt naudas līdzekļus, tādēļ ka teksts tiek pārsūtīts nesalīdzināmi ātrāk nekā jebkuri dizaina efekti.

Grafiskie pārlūki rāda WWW lapas pilnībā un dod iespēju ne vien izbaudīt patiesi augstas klases WWW šedevrus, bet arī izjust pilnīgu bezgaumību dažu lapu noformējumā.

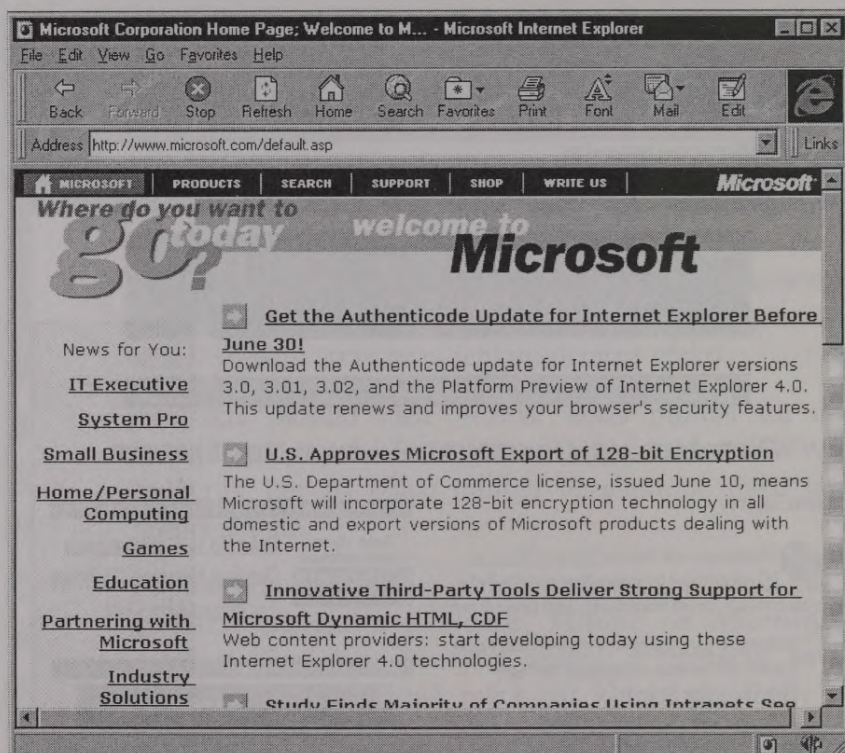
Populārākais tekstuālais pārlūks ir *Lynx*, bet populārākie grafiskie pārlūki ir *Netscape*, *Mosaic* un *Microsoft® Internet Explorer*. Pēdējais strauji ir panācis savus konkurentus un zināmā mērā pat kļuvis par līderi un noteicēju.

Šo pārlūku pamatprincipi pamatā ir vienādi. Pat to logu izskats ir ļoti līdzīgs:



Šis ir *Netscape Navigator* pārlūka logs, kurā ir redzama *Netscape* kompānijas mājas lapa, kurā tā informē par saviem produktiem, piedāvājot to izmēģinājuma versijas ielādei. Kā redzams lauciņā *Address*, šīs lapas adrese Internet tīklā ir *www.netscape.com*. Burti pirms šīs adreses norāda, kāds protokols izmantots datu pārsūtei.

Microsoft® Internet Explorer logs ir ļoti līdzīgs *Netscape* logam:



Šajā piemērā redzama *Microsoft®* mājas lapa, kuras adrese ir *www.microsoft.com*.

Kā jau minējām, informācija lapās ir kodēta, lietojot HTML (Hypertext Mark-up Language) līdzekļus. Šādā veidā sakodēti faili ir vienkārši teksta faili, kurus pārlūki pārveido lietotājam skatāmā formā. Kaut arī HTML iezīmēšanai ir stingri izstrādāti standarti, tomēr katrs pārlūks interpretē tos nedaudz savādāk. Tāpēc nevajag

brīnīties, ka, izmantojot dažādus pārlūkus, rezultāts var atšķirties. Parasti atšķirības ir noformējumā, un tas informācijas kvalitāti neietekmē.



Pārlūka logs, kurā ir redzama mājas lapa.

Atrast kādu lapu tīklā var trijos veidos:

- zinot tās URL un uzdodot to pārlūkam;
- ar klikšķināšanu uz hiperteksta norādēm atnākot no kādas citas lapas;
- uzdodot atslēgas vārdu speciāli WWW izstrādātiem meklētājiem.

Kā var zināt lapas URL? Pirmkārt, lapu adreses var diezgan viegli iegaumēt, jo lielākoties tās ir veidotas pēc principa *www.nnnn.kk*, kur *nnnn* ir kāds konkrēts vārds (ar punktu atdalīti vārdi) vai saīsinājums, bet *kk* ir divburtīgs valsts kods (Latvijā izvietotajām lapām tas ir *lv*), kurā šī lapa atrodas. Šajā principā, protams, var būt, un arī ir savi izņēmumi. Piemēram, Savienotās valstis kā jau Internet dzimtene savām adresēm nepievieno valsts kodu.

WWW adresēs figurē arī dažādi saīsinājumi, kuri var sniegt papildu informāciju par lapu. Lūk, daži vispārpieņemti saīsinājumi:

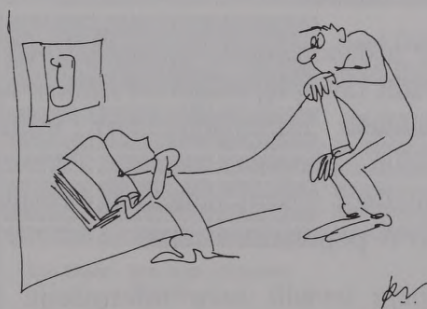
- com (*company*) - komercinformācijas lapa;
- edu (*education*) - izglītības vai zinātniskas iestādes lapa;
- org (*organization*) - organizācija;
- gov (*government*) - valdības struktūras lapa;
- net (*network*) - tīkla organizācija;
- mil (*military*) - militāras vienības lapa.

Pašreiz tiek ieviesti arī par citi domēni. Tie ir paredzēti, lai kaut cik atslogotu milzīgo .com domēnu Amerikas Savienotajās Valstīs.

Šie paši saīsinājumi figurē arī elektroniskā pasta adresēs. Tagad zinot, kā veido WWW lapu adreses, varam pamēģināt piekļūt tām. Piemēram, apskatīsimies pazīstamās automobiļu ražotājkompanijas *Ford* lapas. Tās noteikti ir komerciālas un atrodas Amerikā (lielākā daļa pasauleslaveni firmu, kas neatrodas ASV, reģistrē tās Amerikas .com domēnā). Tāpēc varam pamēģināt šādu URL: www.ford.com. Un patiešām, tas labi nostrādā. Pēc analogijas varam veidot arī citus URL: www.volkswagen.de (Vācijā), www.volvo.se (Zviedrijā) utt. Taču tādā veidā strādājot, var arī nokļūdieties, jo, piemēram, kompānijas MAZDA servera adrese ir www.mazda.co.jp, bet kompānijas Volvo lapas ASV adrese ir www.volvocars.com.

Tāpēc, ja nezināt lapas URL, bet vispār meklējat kādu informāciju WWW, tad vislabāk ir izmantot speciālus

WWW meklētājus. Speciāli atvēlētos lodziņos norādot atslēgas vārdus, pēc kuriem veikt meklēšanu, jums ātri tiks uzmeklētas lapas, kurās ir sastopami jūsu norādītie vārdi. Ja šaubāties par atslēgas vārdu izvēli, tad izmantojiet meklētājos piedāvāto lapu klasifikāciju pēc kategorijām.



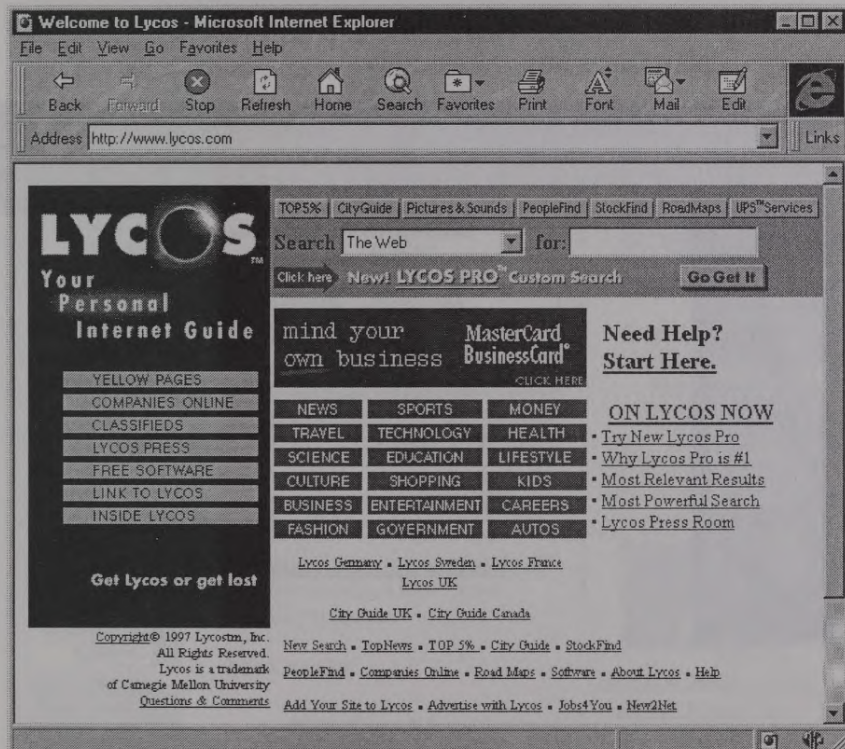
Izmantojiet lapu klasifikāciju pēc kategorijām.

Populārākie WWW resursu meklētāji ir Yahoo! <http://www.yahoo.com>, Lycos <http://www.lycos.com> un AltaVista <http://digital.altavista.com>. Rodas jautājums, kādā veidā šāds meklētājs var sameklēt vajadzīgo informāciju visā plašajā Internet tīklā? Patiesībā pēc jūsu veiktā pieprasījuma meklētājs nebūt neceļo pa pasauli, meklēdams lapas, kurās būtu norādītie atslēgas vārdi. To meklētājs jau ir izdarījis pirms tam, apstaigājot pasaules WWW atrodamos resursus un izveidojot to indeksu, kas ir pietiekoši informatīvs, apjomīgs un sarežģīts. Šis indekss tiek glabāts uz diska, tādēļ katru reizi saņemot pieprasījumu, meklētājs griežas pie šī indeksa un veikli sameklē vajadzīgo informāciju.

Šajā shēmā gan ir iespējamās dažādas variācijas. Piemēram, *Yahoo!* neapstaigā pasaules WWW resursus. Tam ir pašam savs katalogs, kurā pierēģistrēta informācija par dažādām lapām, ko meklētājs arī sakārto pa kategorijām un ieliek indeksā. Neatradis pieprasīto informāciju savā katalogā, *Yahoo!* nodod pieprasījumu *AltaVista* meklētājam, kurš, savukārt, veic meklēšanu savā indeksā.

Šeit nedaudz jāpakavējas pie tā, kādā veidā lietotājs var ievadīt savu pieprasījumu. Kamēr WWW piedāvāja tikai vienkāršu informāciju bez mijiedarbības iespējām, izjūtām pietrūka asuma. Tikai parādoties iespējai lietotājam pašam noteikt informācijas plūsmas saturu, WWW popularitāte ceļas.

Iespēju ievadīt savu informāciju HTML dokumentos nodrošina formas, kuru saturs jānosūta serverim vai pārlūkam, kas to apstrādā un izdod rezultātu.



Piemēram, *Lycos* mājas lapā lietotāja informācijas ievadei tiek piedāvāta viena izvēles karte, kurā norādīt meklēšanas lauku, un viens lodziņš, kurā izmantojot tastatūru, lietotājs var ievadīt meklējamās atslēgas vārdus. Noklikšķinot *Go Get It* pogu, var iedarbināt programmu, kas apstrādā ievadītos datus.

Protams, tas pats *Lycos* piedāvā arī sarežģītākas struktūras formas.

Latvijā populārākais informācijas meklētājs ir katalogs Zaļās Lapas <http://www.zl.lv>

Tajā, kā redzat, varat norādīt vairāk informācijas, pēc kuras veikt meklēšanu, ko aktivizē, noklikšķinot pogu "Aiziet!"

WWW var izmantot arī kā pēdējo jaunumu ieguves avotu. Slavenajam CNN ir sava mājas lapa <http://www.cnn.com>, kurā atspoguļotas pasaules norises.

CNN lapas struktūra (līdzīgi daudzu citu infomatīvu lapu struktūrai) veidota pēc principa, kurā informatīvās sadaļas tiek atspoguļotas vienā malā šaurā joslā, bet

būtiskāko lapas daļu aizņem aktuālo notikumu anotācijas.

CNN
interactive
CNN.com

custom news

CNN Plus

WORLD
U.S.
LOCAL
WEATHER
SPORTS
SCI-TECH
TRAVEL
STYLE
SHOWBIZ
HEALTH
EARTH

CNN
allpolitics

CNN

SITE SOURCES
CONTENTS
HELP
FEEDBACK
SEARCH
CNN NETWORKS

SPECIALS
QUICK NEWS
ALMANAC
VIDEO DRIFT
NEWS QUIZ

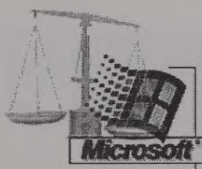
CNNfn

AOL sues another spammer

October 21, 1997 -- Updated 11:32 a.m. EDT (1532 GMT)

TEXT - ONLY VERSION
MORE NEWS

Microsoft faces EC probe



Microsoft Corp. may be fighting a battle on both sides of the Atlantic as the European Commission is conducting its own investigation of the software company's practices.

FULL STORY

- [CNNfn: Microsoft Special Section](#)
- [Microsoft rivals say browser war is really about Windows](#)
- [Today on CNN: Talkback Live: Microsoft Anti-Trust Fine 3 p.m. ET](#)



(CNN/file)

[Sources: Final deal reached in U.S.-Japanese trade dispute](#)

[Yeltsin concessions may scrap no-confidence vote](#)



(AP Photo)

[Mother of Marlins pitcher allowed to enter U.S. from Cuba](#)

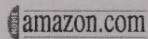
Internet tīklā arvien plašāk tiek piedāvāta iepirkšanās iespēja. Jūs līdzīgi kā veikalā varat aplūkot dažādas preces, izvēlēties tās, kuras jums patīk, un iegādāties tās savā īpašumā. Vienīgā atšķirība no reāla veikala ir tā, ka preces jūs neiegūstat uzreiz, bet ir jāpagaida, līdz tās

pienāk pa pastu. Norēķināšanās lielākoties notiek ar kredītkaršu palīdzību. Te ir jāievēro, ka ne visās vietās ir nodrošināta pārraidītās ionformācijas drošība pret ziņkārīgiem cilvēkiem. Pirms sūtīt pa tīklu datus, kurus jūs nevēlaties darīt zināmus plašākai publikai, pārliecinieties, ka tiek veikta droša datu pārraide, piemēram, ar droša protokola *https* (*http secure*) starpniecību.

Viena no plaši apmeklētām iepirkšanās vietā ir grāmatu veikals *amazon.com*, kurā var iegādāties dažādas grāmatas:

Order today and win a cool \$10,000 in our one-millionth customer celebration!

[Search](#)
[Browse Subjects](#)
[Bestsellers](#)
[Recommendation Center](#)
[Award Winners](#)
[Reviewed in the Media](#)
[Shopping Cart](#)

 **amazon.com**

[Text-Only](#)

Search by:

► Author, Title, Subject

[Quick Search](#)
[ISBN](#)
[Publication Date](#)
[Power Search](#)
[Search Tips](#)

Enter Author and/or Title in the Fields Below
 You do not need to fill in both fields.

Author:

☐ Exact Name
 ☒ Last, First Name (or Initials)
 ☐ Start of Last Name

Title:

☐ Exact Start of Title
 ☒ Title Word(s)
 ☐ Start(s) of Title Word(s)

[Author Search Tips / Title Search Tips](#)

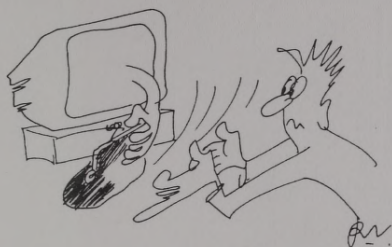
Search by Subject

Or you can [browse our most popular titles](#), subject by subject.

Subject:

☐ Exact Subject
 ☐ Start of Subject
 ☒ Subject Word(s)
 ☐ Start(s) of Subject Word(s)

[Subject Search Tips](#)



Internet tīklā arvien plašāk tiek piedāvāta iepirkšanās iespēja.

Internet un WWW pasaule ir ļoti aizraujošas, tomēr jābūt uzmanīgiem neieslīgstot tajā, zaudējot laika un reālās dzīves sajūtu. Fantastikas filmās rādītie datorfanāti, kuri ir atslēgušies no reālās dzīves un dzīvo tikai savā fantāziju jeb virtuālā pasaulē, patiešām ir realitāte. Patiesajā dzīvē ir savi jaukumi, ko virtuālā pasaule nespēs jums sniegt.

Māris Detlavs, Katrīna Sataki
IESKATIES DATORU PASAULĒ
Romana Vitkovska ilustrācijas

Mākslinieks: I.Pūgule
Redaktors: E.Milčonoka

Izdevējs: Datorzinību centrs, SIA. Jauniela 24, Rīga, LV-1050.
Iespiests AS Paraugtipogrāfija. A.Puškina 12, Rīga.

ISBN 9984 - 9108 - 4 - 9

Walter D. (1904-1984) School
1934-1935 DASHING RASBIE
Rogers, Virginia, 1935-1936

Walter D. (1904-1984)
Rogers, Virginia, 1935-1936

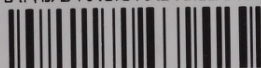
Walter D. (1904-1984) School, 1934-1935
Rogers, Virginia, 1935-1936

1934-1935 - 1935-1936

OBLIGĀTAIS
EKSEMPLĀRS

3.45

LATVIJAS NACIONĀLA BIBLIOTĒKA



0302054268

98-3
1
-255



Laipni lūgti datoru pasaulē! Datori veido veselu pasauli, kas ir komplicēta, idejām pārpilna, interesanta un pat skaista. Mēs gribam atvieglot jums šīs pasaules iepazīšanu ar analogijām no dažādām dzīves jomām, ilustrācijām un jokiem.