

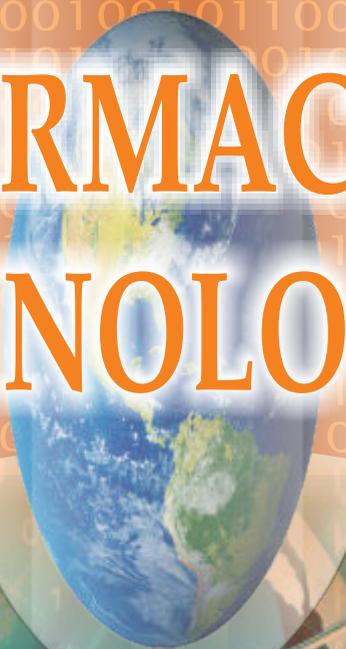
SINIŠA G. MINIĆ - INFORMACIONE TEHNOLOGIJE

EVROPSKI UNIVERZITET
BRČKO DISTRIKT
BOSNA I HERCEGOVINA



EUROPEAN UNIVERSITY
BRCKO DISTRIKT
BOSNIA AND HERZEGOVINA

INFORMACIONE TEHNOLOGIJE



SINIŠA G. MINIĆ



INFORMACIONE TEHNOLOGIJE

Prof. dr Siniša G. Minić, Informacione tehnologije

UREDNIK:

Dr Nedeljko Stanković

RECENZENTI:

Prof. dr Gradimir Milovanović,

Prof. dr Mihajlo Stefanović,

IZDAVAČ:

EVROPSKI UNIVERZITET BRČKO DISTRIKTA

Tel. 049 490 905

<http://www.evropskiuniverzitet-brcko.com/>

Odlukom Senata Evropskog univerziteta u Brčkom, broj: 827-0D-02/2012, od 15. 09. 2012. godine, knjiga «Informacione tehnologije» autora prof. dr Siniše G. Minića prihvaćena ja kao udžbenička literatura.

ŠTAMPA:

Markos, Banja Luka

TIRAŽ:

200.

ISBN 978-99955-750-6-9

SINIŠA G. MINIĆ

**INFORMACIONE
TEHNOLOGIJE**

**EVROPSKI UNIVERZITET
BRČKO, 2012.**

Predgovor

Tokom poslednjih decenija došlo je do povezivanja informatike, računarske tehnike i telekomunikacija, što je dovelo do dubokih, može se slobodno reći revolucionarnih promena u tehnologiji obrade i prenosa informacija.

Sve je počelo pred drugi svetski rat, tačnije 1938. godine kada je patentiran postupak koji omogućava transformaciju kontinualnih signala u digitalne. Pokazano je da se svaki kontinualni signal, čiji je spektar ograničen, može diskretizovati po vremenu i po trenutnim vrednostima, a svaki ovako kvantovani odmerak predstavi odgovarajućom brojnom vrednošću tj. grupom cifara (kodovanje). Na ovaj način se dalja obrada signala (odnosno informacija) svodi na niz aritmetičko-logičkih operacija koje treba obaviti nad diskretnim vrednostima podataka. Pošto se svaka informacija, bilo da je to govor, slika, ili neki drugi podatak, transformiše u skup brojeva to u principu nema razlike u obradi i prenosu različitih tipova informacija. Problem je, međutim, što je broj numeričkih podataka kojim se u procesu digitalizovanja predstavljaju informacije, veoma veliki.

Nagli porast broja međusobno povezanih računara (Internet) iznedrilo je potpuno nove zahteve u pogledu prenosa informacija koje će zadovoljiti u pogledu kapaciteta, brzine prenosa i pouzdanosti. Zbog toga je potrebno izgraditi moćne mreže za prenos podataka kako bi se omogućio nesmetan protok sve većih količina informacija. Može se zaključiti da je informaciona tehnologija postala osnovna infrastruktura svake zemlje i glavno sredstvo za povezivanje sa svetom. Zbog toga je neophodno izučavanje informatike u našem obrazovnom sistemu radi formiranja novih predstava o ulozi informatike u savremenom društvu.

Udžbenik INFORMACIONE TEHNOLOGIJE je prvenstveno namenjen studentima tehničkih fakulteta, a nastao je na bazi predavanja i vežbanja koje je autor držao studentima iz istoimenog predmeta.

Pri određivanju sadržaja i obima izlaganja, autor je uglavnom sledio program predmeta **INFORMACIONE TEHNOLOGIJE**. Na taj način je pružena mogućnost studentima da se, po sopstvenom nađenju, detaljnije upoznaju sa informacionim tehnologijama.

U uvodnom poglavlju je najpre ukratko opisan razvoj informatike i objašnjen pojam informacije. Zatim je dat prikaz komponenata informacionog sistema: hardvera i softvera.

U drugom poglavlju je dat prikaz digitalnih podataka u raznim brojnim sistemima i kodovima, kao i osnovne računске operacije nad njima.

Treće poglavlje sadrži arhitekturu računara. Najpre je dat kraći opis računara, a zatim je pažnja posvećena najvažnijem delu računara - procesoru, koji i određuje arhitekturu računara. Izložena je arhitektura procesora i skupa instrukcija procesora. Na kraju ovog poglavlja su nabrojane neophodne komponente računarskog sistema.

Četvrto i peto poglavlje se bave memorijskim sistemom računara i ulazno-izlaznim uređajima. Prikazan je memorijski sistem računara, kao i komponente za masovno skladištenje podataka. Opisan je princip rada najčešće korišćenih ulazno-izlaznih uređaja. Posebna pažnja je posvećena monitoru, koji može biti kako izlazni tako i ulazni uređaj. Opisan je fizički proces generisanja boje na ekranu monitora.

Šesto poglavlje koje nosi naziv "Softver računara" odnosi se pre svega na operativne sisteme. Pored opisa Windows operativnog sistema, dati su i kraći opisi Unix operativnog sistema i njegove varijante Linux za personalne računare. Ako se ima u vidu da je XP Windows System koji radi pod operativnim sistemom Linux besplatan, treba očekivati da će se u budućnosti sve više koristiti.

Sedmo poglavlje obuhvata računarske komunikacije, i odnosi se pre svega na globalnu računarsku mrežu - Internet. Pored pristupa globalnoj računarskoj mreži, ovo poglavlje se bavi i HTML jezikom. Prikazan je postupak generisanja Veb prezentacija, kako direktnim korišćenjem HTML jezika, tako i primenom programskog paketa FrontPage koji radi pod operativnim sistemom Windows.

Sadržaj

Predgovor	3
1 Uvod	15
1.1 Razvoj informatike	17
1.2 Informacije i njihovo predstavljanje	20
1.3 Definisanje pojma informacija	22
1.4 Komponente informacionog sistema	30
1.4.1 Hardver	30
1.4.2 Programi	34
1.4.3 Podaci	37
1.4.4 Kadrovi	37
1.4.5 Procedure	38
1.5 Pitanja za proveru znanja	39
2 Predstavljanje podataka	41
2.1 Brojni sistemi	41
2.1.1 Binarni kodovi	45
2.1.2 Binarno kodovani decimalni brojevi	47
2.2 Predstavljanje numeričkih podataka	49
2.2.1 Celobrojni podaci.....	49
2.2.2 Format pokretne tačke.....	57
2.3 Predstavljanje nenumeričkih podataka	61
2.4 Pitanja za proveru znanja	63

3	Računarski sistemi	65
3.1	Istorijski razvoj računara	66
3.2	Informacione tehnologije.....	69
3.3	Obrada podataka.....	70
3.3.1	Operacije nad podacima	71
3.3.2	Proces obrade podataka	71
3.4	Struktura računara.....	72
3.4.1	Ulazna jedinica	73
3.4.2	Izlazna jedinica	74
3.4.3	Centralni procesor	75
3.4.4	Operativna memorija	76
3.4.5	Stek memorija	79
3.5	Procesor	79
3.5.1	Funkcije procesora.....	79
3.5.2	Arhitektura procesora	83
3.5.3	Registri procesora	90
3.5.4	Instrukcije	93
3.5.5	Izvršenje programa.....	98
3.6	Konfiguracija personalnih računara	99
3.6.1	Osnovna ploča	100
3.6.2	Ulazno-izlazna ploča	101
3.6.3	Kontroler diskova i disketa	101
3.6.4	Video-adapter	101
3.7	Pitanja za proveru znanja	102
4	Memorijski sistem	105
4.1	Memorijski medijumi	107
4.1.1	Memorijski sistem računara	108
4.1.2	Klasifikacija memorija	109
4.1.3	Struktura memorijskog sistema	111
4.1.4	Hijerarhija memorija	112

4.2	Operativna memorija	115
4.3	Spoljna memorija ..	116
4.3.1	Disket.....	116
4.3.2	Tvrđi magnetni disk.....	118
4.3.3	Optički disk (kompakt disk)	121
4.4	Pitanja za proveru znanja	124
5	Ulazno izlazni uređaji	125
5.1	Ulazni uređaji.....	125
5.1.1	Tastatura ..	127
5.1.2	Miš	128
5.1.3	Grafička tabla digitalizator	130
5.1.4	Skener	130
5.1.5	Čitači prugastog koda	131
5.2	Izlazni uređaji	132
5.2.1	Monitor ..	134
5.2.2	Štampači.....	142
5.3	Pitanja za proveru znanja	147
6	Softver računara	149
6.1	Sistemske softver	150
6.1.1	Operativni sistem	152
6.2	Režimi rada računara	163
6.2.1	Obrada sa podelom vremena	167
6.2.2	Rad u realnom vremenu	169
6.2.3	Daljinska obrada	169
6.2.4	Distribuirana obrada.....	170
6.3	Programski sistem	172
6.3.1	Programski prevodioci	172
6.3.2	Uslužni programi	174
6.3.3	Sredstva za razvoj softvera	177

6.4	Operativni sistem UNIH	177
6.4.1	Upravljanje zadacima	179
6.4.2	Upravljanje memorijom	181
6.4.3	Upravljanje datotekama	181
6.4.4	Upravljanje ulazam-izlazom	183
6.4.5	Pomoćni sistemski programi alati	184
6.5	Operativni sistem WINDOWS	185
6.5.1	Uslužni softver	189
6.6	Korisnički interfejs	193
6.6.1	Karakteristike korisničkog interfejsa	194
6.6.2	Direktna manipulacija	196
6.6.3	Modeli interfejsa	197
6.6.4	Sistemi sa menijima	199
6.6.5	Komandni interfejsi	200
6.6.6	Vođenje korisnika (help sistem).....	202
6.6.7	Poruke o greškama	202
6.7	Operativni sistem Linux.....	202
6.7.1	Istorijat Linuxa	204
6.7.2	Pregled Linuxa	205
6.7.3	Pristupanje Linux sistemu	207
6.7.4	Upravljanje datotekama i arhivama.....	207
6.8	Pitanja za proveru znanja	211
7	Računarske komunikacije	213
7.1	Računarske mreže	213
7.1.1	Prenos podataka	218
7.2	Internet-globalna računarska mreža	223
7.2.1	Protokoli Interneta	225
7.2.2	Adresiranje na Internetu.....	225
7.2.3	Pristup Internetu	226
7.2.4	Klijent-server model	227

7.3	Usluge i servisi Interneta	228
7.4	Veb-multimedijalni servis internet	231
7.4.1	Pristup Vebu	233
7.4.2	Veb čitači.....	235
7.5	Kreiranje HTML dokumenta.....	235
7.5.1	HTML - jezik Veb dokumenata	237
7.5.2	Struktura HTML dokumenta	237
7.5.3	HTML kategorije	242
7.6	FrontPage	250
7.6.1	Kreiranje Veb stranice	250
7.6.2	Editovanje stranice u FrontPage-u	252
7.7	Mreže za integrisane usluge	254
7.7.1	Digitalni prenos - ISDN	254
7.7.2	Specijalizovani telekomunikacioni servisi	255
7.8	Mrežne alatke	258
7.8.1	Informacije o mreži	258
7.8.2	Mrežni klijenti	260
7.9	Pitanja za proveru znanja	261

Glava 1

Uvod

Značajne promene u industrijskom društvu desile su se onda kada su se razvile informacione tehnologije kao što su: mikroelektronika, računari, telekomunikacije, robotika i tehnologija novih materijala. Uvođenjem ovih tehnologija uslovalo je promene u društvenim odnosima, proizvodnji, životu ljudi i njihovom obrazovanju. U rastu produktivnosti sve značajnije mesto zauzima znanje u odnosu na kapital i rad.

Naziv informatika je nastao konstukcijom dve franciske reči "INFORMation" i "automATIQUE" koje je spojio francuski inženjer Philippe Dreyfus 1962. godine, u prevodu: informacija i automatika (automatska obrada podataka). Pandan ovom nazivu u engleskom jeziku je "Computer Science", u prevodu nauka o računarima. Međitim, problem određenja pojma informatike je dosta složen jer se informatika ili informaciona nauka različito shvata i tumači u pojedinim zemljama.

Kao samostalna naučna disciplina razvija se 60-tih godina prošlog veka, najpre u SAD i Velikoj Vritaniji, a zatim i u drugim tehničkim razvijenim zemljama. Iako se pojam informatike ne može isključivo vezivati za automatsku obradu podataka, činjenica je da je razvoj informatike u direktnoj vezi sa razvojem informacione tehnologije, drugim rečima, za razvoj kompjuterske tehnike.

Informatika je interdisciplinarna naučna disciplina, njen razvoj je uslovljen razvojem znanja, koncepata, metoda i principa raznih nauka i njihovom simbiozom (matematike, elektronike,

logike, teorije informacija, semiotike, kibernetike i dr.).

U definisanju informatike, zbog predhodno rečenog, ne postoji saglasnost teoretičara i praktičara u ovoj oblasti. Definicije informatike, kao i sam predmet dobijaju sve šira obeležja, sa ili bez kompjuterske podrške.

Francuska akademija nauka 1966. godine definiše informatiku kao: „*nauku o racionalnoj obradi informacija, prv svega pomoću automatskih mašina, timv da se informacija smatra nosiocem ljudskih znanja i komunikacija u oblasti tehnike, ekonomije i drugih društvenih nauka*“.

Neke od definicija informatike su:

- Naučna disciplina koja proučava: strukturu i svojstva informacije, zakonitosti informacionih delatnosti, te njenu teoriju, istoriju, metodologiju i organizaciju informacionih delatnosti.
- Naučna disciplina čiji je predmet istraživanja sastav, funkcija, organizacija, razvoj i funkcionisanje informacionih sistema sa računarskom podrškom.

Kao naučna disciplina, informatika se bavi metodama algoritamskog rešavanja problema, metodama strukturnog programiranja i razvijanja programskih jezika višeg reda. Takođe, informatika razvija optimalne metode i sredstva za prihvatanje, memorisanje, prenos, obradu, distribuciju i korišćenje informacija.

U definisanju informatike prisutan je zahtev više autora da se ona definiše: „*Kao integralna nauka o informacijama u najširem smislu reči, koja proučava zakonitosti u informaciono-dokumentacionoj delatnosti, a razvija primenom tehničkih sredstava za obradu podataka u prvom redu kompjutera*“.

Sa razvojem savremenih pristupa u organizovanju informacionih sistema, onih koji su orijentisani ka korisniku i čine podršku odlučivanju na svim nivoima upravljanja sistema, onih koji su distribuirani i bave se prognozom budućih ponašanja sistema, informatika dobija sve više osobine interdisciplinarnosti. Ona se vezuje za rezultate niza drugih disciplina: opšte teorije sistema, nauke o organizaciji, teorije upravljanja, kibernetike, teorije odlučivanja, raznih društvenih nauka, operacionih istraživanja i drugih.

Tako informatika dobija inetrdisciplinarna obeležja i u odnosu sa drugim naukama i u odnosu sa drugim praktičnim delatnostima.

Na području metoda i postupaka organizivanja informacionih sistema, informatika se najbliže vezuje i za područja računarske tehnike, telekomunikacija, veštačke inteligencije, programskih jezika i područja informacione tehnologije i informacione metodologije, najšire shvaćenih. Tako se područje informatike širi i razjedinjuje na manje i specifične oblasti i discipline koje se dalje međusobno prožimaju itd., što je već poznati način razvoja naučnih oblasti i disciplina.

„Posebna područja informatike, odnosno posebne relativno zaokružene naučne celine ove discipline jesu:

- *organizacija i arhitektura rač unarskih sistema;*
- *organizacija podataka;*
- *razvoj i funkcionisanje informacionih sistema;*
- *programiranje i teorija algoritama.”*

Cilj informatike bi mogao da se odredi kao:

1. Razvijanje i implementacija metoda i sredstava za optimalno organizovanje informacionih sistema;
2. Razvijanje i impelmentacija metoda i sredstava za optimalno prikupljanje, skladištene, obrađivanje, distribuiranje i tumačenje informacija.

1.1 Razvoj informatike

Naučno-tehnološki razvoj u poslednjih nekoliko decenija karakteriše pojava i ekspanzija novih tehnologija pa i informacionih tehnologija.

Sredinom prošlog veka veka susreću se svi faktori i uslovi za razvoj informatike. Tada se razvija društvena potreba za primenom novih tehnologija u oblasti informacija, ali i mogućnost zadovoljenja tih potreba kroz razvoj tih tehnologija koje "industrijsko" društvo uvode u "informatičko" društvo.

Niz tehničkih i tehnoloških dostignuća, kao: razvoj novih materijala, novih izvora energije, savremene infrastrukture i komunikacija i drugo, omogućavaju kompleksniji proces automatizacije čovekovog rada. Javlja se automatizacija, sa svojim poznatim karakteristikama:

- potpuna mehanizacija u rukovanju materijalom – integracija rada pojedinih mašina;
- kontinuiranost proizvodnih procesa - racionalizacija i optimizacija proizvodnje;
- automatska kontrola i upravljanje;
- kompletno izmenjena uloga čovek/mašina.

Automatizacija proizvodnje dovodi do takvog povećanja kompleksnosti, međuzavisnosti, osetljivosti i brzine proizvodnih procesa, da tradicionalan način upravljanja više nije mogao biti uspešan. Ovaj problem još više dolazi do izražaja kada se automatski upravlja ne samo proizvodnjom materije, energije, već i informacija.

Pronalaskom vakuumske elektronske cevi omogućena je zamena mehaničkih uređaja za kontrolu i regulaciju, elektromehaničkim i elektronskim uređajima. Elektronska cev je omogućila konstruisanje mašina koje same kontrolišu svoj rad, dakle, eru automatizacije. Elektronska cev počinje brzo da se primenjuje u raznim područjima telekomunikacija i prenosa informacija, što izaziva pravu revoluciju na području nauke i prakse informacionih sistema.

Od primene u konstrukciji prvih digitalnih računara zasnovanih na elektronskim cevima pa do tranzistora, integrisanih kola LSI (Large Scale Integration), integracije VLSI (Very Large Scale Integration), pa do današnjih neuro i bio čipova, nastaje fantastičan razvoj i primena raznih pronalazaka informacione tehnologije.

Složenost i dinamičnosti pojava i procesa u prirodi i društvu, kao i nemogućnost njihovim upravljanjem na efikasan način, nametnuo je potrebu stvaranja i razvoja novih naučnih principa upravljanja dinamičkim sistemima. Razvoj informacione tehnologije omogućuje brzu i efikasnu obradu podataka, menja i uspostavlja nove metode poslovanja i odlučivanja u

složenim sistemima. Tako su stvoreni upravljački orijentisani informacioni sistemi, čiji je cilj proizvodnja informacija za potrebe odlučivanja u poslovnom sistemu. Tendencija je u razvijanju sistema za podršku odlučivanju u nestruktuiranim upravljačkim problemima, ekspertnih sistema i uopšte, sistema zasnovanih na znanju.

Sve veći značaj informacione tehnologije u proizvodnji kvalitetnih informacija za odlučivanje, pooštavaju odnos između postojećih rešenja i potreba za novim obrazovnim softverima. Izlaz se traži u sasvim novom konceptu u proizvodnji softvera. Jedna od mogućnosti sa velikom perspektivom je baza podataka.

Slično kao i hardver, i softver je krenuo putevima automatizacije, specijalizacije i standardizacije. Dosadašnja metodologija u izradi softvera za prikupljanje, unos, generisanje izveštaja i obezbeđenje informacija za potrebe odlučivanja, morala je da pretrpi transformacije.

S druge strane, stalna je deficitarnost informatičkih kadrova, pogotovu u proizvodnji softvera. Vrednost softvera je u porastu, zahvaljujući sve vrednijem visokostručnom radu. Cena softvera pomera se sa 70% : 30%, na 90% : 10%, u odnosu na ukupnu cenu uređaja, u korist softvera.

Sa razvojem informacione tehnologije a i potreba, razvoj komunikacionog hardvera i softvera se takođe nameće kao jedan od trendova razvoja u ovoj oblasti. Sa razvojem računarskih mreža razvija se i komunikacioni softver koji povezuje više korisnika u jednu mrežu, ili više mreža, i obezbeđuje podršku različitih usluga korisnika. Povezanost hardverskih uređaja predstavlja i osnovu funkcionalne povezanosti informacionih resursa organizacija. Da bi se ostvarila potreba korisnika za komunikacijama treba ostvariti široku vezu hardverskih uređaja, softverskih proizvoda i informacionih resursa.

Uticaj informacione tehnologije na aktivnosti sistema se, međutim, posmatra i sa mnogo šireg aspekta. Informaciona tehnologija i informacioni sistemi prestaju da budu pretežno i samo podrška procesa, koji se realizuje u nekom sistemu, već ih i radikalno transformišu.

Informaciona tehnologija utiče na procese sistema na

sledeći način:

- Transformiše nestruktuirane procese u rutinske.
- Omogućava brzi prenos informacija na veliku distancu.
- Smanjuje ili u potpunosti zamenjuje ljudski rad u procesima.
- Obezbeđuje kompleksne analitičke metode za analizu procesa.
- Obezbeđuje dovoljnu količinu detaljnih informacija o procesu.
- Izvršava više zadataka u procesu simultano.
- Čuva i širi znanja i ekspertize za unapređenje procesa.
- Detaljno prati status zadataka, ulaza i izlaza.
- Neposredno komunicira učesnike u procesu.

Informacione tehnologije nisu samo sredstvo za automatizaciju poslovanja, već i alat za fundamentalnu promenu procesa sistema. Pri samom projektovanju procesa sistema treba razmotriti mogućnosti koje informaciona tehnologija pruža u oblasti smanjenja troškova koordinacije i komunikacije i organizacione transformacije uopšte.

1.2 Informacije i njihovo predstavljanje

Polazeći od najopštijeg razmatranja informacionog sistema kao posebne vrste sistema, može se reći da je informacioni sistem, sistem u kome se veze između elemenata i interakcija sa okolinom ostvaruje razmenom informacija. Informacioni sistem je osnova upravljanja, odnosno upravljačkog sistema, pa se sa tog stanovišta može govoriti i o upravljačkim informacionim sistemima, kao sistemima koji proizvode informacije za upravljanje.

Informacioni sistemi se definišu kao sistemi koji obezbeđuju prikupljanje, generisanje i čuvanje informacija za odlučivanje kod planiranja, izvršavanja, kontrolisanja i adaptacije sistema. Sa tog stanovišta svaki informacioni sistem je deo sistema upravljanja nekog realnog sistema.

U svakom slučaju, informacioni sistem nas upućuje na povezivanje informacionog sistema sa upravljanjem. Sistemi upravljanja, su sistemi u kojima se ostvaruje upravljanje ponašanjem sistema i koji imaju jednu od svojih osnovnih komponenti informacije.

Informacija se javlja kao bitna komponenta svih organizovanih sistema. Kako ovi čine mnoštvo i prirodno i veštački organizovanih sistema, to je moguća široka primena i (upravljačkih) informacionih sistema.

Informacioni sistem predstavlja tip dinamičkog sistema u kome se ostvaruje transformacija podataka u informacije za upravljanje. On se može izučavati kao poseban tip dinamičkog sistema, koji se karakteriše promenom stanja, kao i uslova i načina stvaranja tih promena. Iz tih razloga, prilikom izučavanja (upravljačkih) informacionih sistema, mora se stalno imati na umu kinematika sistema, kretanje i promene stanja sistema.

Za primenu sistemskog pristupa u upravljanju sistemima neophodan uslov predstavljaju informacioni procesi i informacioni sistemi koji ih realizuju.

Informacioni sistem je kompleksan, dinamičan, organizacioni sistem. Svaki realan sistem ima svoj informacioni sistem. Ovaj je više ili manje razvijen, a treba da zadovolji informacione zahteve brojnih korisnika.

Na bazi podataka o odstupanju stvarnog stanja sistema od željenog stanja (iskazanog u definisanim ciljevima sistema), kao i podataka iz okoline, informacioni sistem proizvodi informacije potrebne za upravljanje, koje sistem, i pored poremećajnih dejstava, vode ka željenom stanju.

U strukturi svakog informacionog sistema mogu se definisati četiri osnovne aktivnosti u vezi sa podacima i informacijama.

To su:

- Prikupljanje podataka.
- Obrada podataka.
- Memorisanje (pamćenje) podataka.
- Dostavljanje podataka.

U prvoj aktivnosti se vrši prikupljanje podataka na mestu njihovog nastajanja (na primer podaci o uspehu sudenata). Osim samog prikupljanja podataka ovde se vrši i pripremanje podataka za njihovo unošenje u kompjuter i obradu.

Obradom se prikupljeni podaci transformišu u skladu sa principima i logikom obrade, i na osnovu toga utvrđenim pro-

gramskim postupcima. Tek obrađeni podaci primaju karakter informacija. Obradeni i/ili delimično obrađeni podaci, takođe i izvorni, memorišu se u tzv. "skladištima" podataka na raznim memorijskim medijumima. Ovi podacii se dalje koriste u postupcima obrade ili kao informaciona podloga za odlučivanje.

U postupku dostavljanja, podaci se organizovano dostavljaju korisnicima u vidu standardnih izveštaja ili na trenutni zahtev - upit.

Svaki informacioni sistem zahteva visok stepen automatizovanosti procesa, počev od obrade informacija do planiranja i upravljanja.

1.3 Definisanje pojma informacija

Pojam informacija se veoma mnogo koristi u svakodnevnom govoru i najčešće se identifikuje sa pojmom podatak. Precizna i opšte prihvaćena definicija informacija ne postoji, no to ne predstavlja prepreku za njeno proučavanje i korišćenje.

Informacija je količina neophodnog sadržaja podataka za određivanje, odnosno, shvatanje kompleksnosti nekog sistema. To je količina obaveštenja neophodnih da se sistem prati u dinamici, odnosno u svojoj evoluciji, povezano sa njegovim položajem u okruženju, u njegovom odnosu na druge sisteme i povezano s promenama u njegovoj strukturi i funkciji.

Stavovi N. Vinnera osnivača kibernetike, da se informacijom i organizacijom može suprotstaviti entropiji, su u osnovi formiranja svakog sistema. Po njemu: "mašine, fizički, hemijski i duševni procesi života predstavljaju lokalne antientropijske procese".

N. Vinner pod informacijom podrazumeva sadržaj onog što razmenjujemo sa spoljašnjim svetom, dok mu se prilagođavamo i dok na njega utičemo svojim prilagođavanjem.

U matematičko-statističkoj teoriji informacija Shannona i drugih naučnika, posmatra se kvantitativni aspekt informacija i komunikacija. Prema osnovnim stavovima ove teorije, *količina informacija je obrnuto proporcionalna verovatnoći nastajanja nekog događaja, čiju informaciju merimo*. Drugim rečima, što je nastu-

panje događaja verovatnije to je količina informacija koju ovaj nosi manja. Ako, na primer, u novinama pročitamo da je fudbalska utakmica između državnog prvaka i trećerazrednog provincijskog tima završena pobedom prvaka, onda nam takva vest ne donosi "mnogo informacija", jer ne sadrži ništa što bi za nas bilo neočekivano. Nasuprot tome, ako pročitamo ili čujemo prognozu da će prvak na svom terenu biti poražen od gostujućeg provincijskog tima - onda će takva poruka sadržati "mnogo informacija", jer je za nas neočekivana. Prema tome, u saglasnosti sa našim uobičajenim shvatanjem, svaka poruka koju očekujemo sadrži malo informacija, i obratno - neočekivana poruka sadrži mnogo informacija.

U opštem slučaju, kao mera za količinu informacija u nekoj poruci može, dakle, da posluži veličina promene verovatnoće događaja pod uticajem date poruke. Na taj način zamnjujemo maglovit pojam "informacije" preciznom definicijom koja će omogućiti uvođenje brojčane mere za količinu informacija. Tako ćemo biti u stanju da kvantitavno upoređujemo različite informacije i različite sisteme za prenos i obradu informacija. *Izvor informacija* je osoba ili uređaj koji generiše poruku. Poruka može biti: govor, muzika, pisani tekst, slika, računarski, merni ili neki drugi podaci.

Posmatrajmo događaj koji predstavlja jedno od mogućih stanja stacionarnog¹, diskretnog slučajnog procesa $\xi(t)$. Potencijalni korisnik informacije poznaje prirodu slučajnog procesa i funkciju raspodele verovatnoća njegovih stanja. Predpostavićemo da je korisnik zainteresovan samo za jedno određeno diskretno stanje slučajnog procesa, koje ćemo obeležiti sa s_i , i živi u očekivanju informacije da je stanje s_i nastupilo. Ukoliko je verovatnoća datog diskretnog stanja $P_{\xi}(s_i)$ manja, utoliko je veća neizvesnost u kojoj se nalazi korisnik informacije. Međutim, u trenutku prijema informacije da je stanje s_i nastupilo, stepen neizvesnosti korisnika informacije se naglo smanjuje, ili, u ekstremnom slučaju potpuno pouzdane informacije, neizvesnost prelazi u izvesnost. Prema tome, količina informacije $Q(s_i)$ koju sadrži događaj o stanju s_i i verovatnoća datog stanja $P_{\xi}(s_i)$ su obrnuto

¹Pod stacionarnim procesom se podrazumeva proces koji ne menja svoje karakteristike tokom vremena. Na primer, bacanje kocke je stacionarni, diskretni slučajni proces koji sadrži šest porika.

proporcionalne veličine, što se matematički može izraziti na sledeći način

$$Q(s_i) \Leftrightarrow \frac{1}{P_\xi(s_i)}$$

Uobičajeno je da se količina informacija definiše kao logaritam recipročne vrednosti verovatnoće, pa je

$$Q(s_i) = \log_b \frac{1}{P_\xi(s_i)}$$

Da bi se količina informacija mogla brojčano izraziti potrebno je definisati osnovu logaritma b . Najčešće se usvaja binarna (tj. dvojna) osnova logaritma pa se ta jedinica za meru informacija naziva "Shannon)", u čast Clauda Shannona, koju ćemo skraćeno obeležavati skraćnicom "Sh"

$$Q(s_i) = \log_2 \frac{1}{P_\xi(s_i)} [Sh]$$

Kada je izvor informacija slučajni proces sa svega dva diskretna stanja s_1 i s_2 , i kada su oba stanja jednako verovatna, tj. $Q(s_i) = 0.5$ za $i = 1, 2$, tada je

$$Q(s_1) = Q(s_2) = \log_2 2 [Sh]$$

Prema tome, Šanon je količina informacija koju sadrži događaj koji ima dva moguća stanja, uz uslov da su oba stanja jednako verovatna. Na primer, saopštenje o tome da je metalni novčić prilikom bacanja u vis pao na određenu stranu nosi informaciju od jednog Šanona. Slično tome, ako je naš poznanik dobio dete i mi ga pitamo da li je sin ili kćer, onda njegov odgovor: "kćer", sadrži takođe jedan Šanon informacija. Najzad, ako se u memoriji računara za predstavljanje podataka koriste se binarni brojevi koji koriste samo cifre 0 i 1, i ako je verovatnoća pojavljivanja nula i jedinica jednaka, onda svaka cifra nosi informaciju od jednog Šanona. Međudim, ako verovatnoće pojavljivanja nula i jedinica nisu međusobno jednake, onda jedna cifra nosi neku drugu količinu informacija različitu od jednog Šanona. U ovim razmatranjima, koja se odnose na binarno predstavljanje brojeva, treba striktno razdvojiti pojam "binarni digit" skraćeno "bit" od pojma šanon. Bit je samo nosilac informacija i fizički je

predstavljen impulsom koji može da ima dva stanja. U zavisnosti od verovatnoće pojavljivanja stanja, bit može da sadrži veću ili manju količinu informacija.

U daljem tekstu ćemo smatrati da su verovatnoće pojavljivanja dva stanja kod bita jednake, i da jedan bit nosi informaciju od jednog Šanona ($1 \text{ bit} = 1 \text{ Sh}$).

Sada nije teško razumeti zašto se za definiciju količine informacija koristi logaritam sa osnovom $b = 2$. Očigledno je da su u prirodi najrasprostranjeniji slučajni procesi binarnog tipa i da elementarna informacija "da" ili "ne", kada su "da" i "ne" podjednako verovatni, odgovara logičkoj jedinici količine informacija.

Može se zaključiti da definicija da mera za količinu informacija počiva na čisto statističkim osobinama pojedinih stanja odgovarajućih informacija, i ne vodeći računa ni o sadržaju ni o vrednosti tih poruka u odnosu na korisnika kao ljudsko biće. Na primer, količina informacija koju nose dva slova s_i i s_j određena je samo verovatnoćom pojavljivanja tih znakova u tekstu, a ne zavisi od toga da li je tekst napisao Njegoš ili neki novinar, da li je tekst na srpskom ili engleskom jeziku, itd. Usvojena definicija, prema tome ne pravi nikakvu razliku između korisne i beskorisne informacije i potpuno ignoriše vrednost informacije. Ako je informacija malo verovatna smatra se da ona sadrži dosta informacija bez obzira kakav je odnos čoveka korisnika prema toj informaciji.

U opštem slučaju, svaki slučajni proces može da predstavlja izvor informacija. Na izlazu izvora informacija pojavljuje se, naime, u nekom prostornom ili vremenskom rasporedu, saopštenje u vidu simbola koji predstavlja moguće stanje procesa. Pošto u pogledu pojavljivanja nekog stanja ne postoji izvesnost, to simbol kojim se saopštava da je nastupilo neko stanje donosi korisniku izvesnu količinu informacija. U većini slučajeva simboli se pojavljuju na izlazu izvora informacija u vremenskom nizu, pri čemu vremenski intervali između pojavljivanja simbola mogu biti konstantni ili promenljivi. Ukoliko je posmatrani proces diskretnog tipa i ukoliko mu je skup mogućih stanja konačan - govorimo o diskretnom izvoru informacija, s konačnom listom simbola. Ukoliko je pojavljivanje simbola na izlazu izvora infor

macija u statističkom smislu potpuno nezavisno, onda imamo na-jprostiji tip izvora informacija.

Posmatrajmo, dakle, diskretni izvor informacija koji je u potpunosti okarakterisan svojom listom simbola $S = s_1, s_2, \dots, s_q$ i prostim verovatnoćama pojavljivanja tih simbola $P(s_1), P(s_2), \dots, P(s_q)$. Kada se na izlazu izvora informacija pojavi neki simbol s_i , tada taj događaj nosi količinu informacija

$$Q(s_i) = \log_2 \frac{1}{P(s_i)} [b]$$

Međutim, u praktičnim situacijama nas obično ne interesuje koliko informaciju nosi jedan određeni simbol, jer on ne mora da bude tipičan predstavnik svih simbola koje generiše posmatrani izvor informacija. Na primer, ako fiksiramo simbol koji se vrlo retko pojavljuje, onda, i pored činjenice da takav simbol nosi veliku količinu informacija, njegov doprinos statističkoj vrednosti informacija neće biti značajan. Da bismo mogli objektivnije da procenimo informacionu moć izvora informacija, potrebo je, dakle, da odredimo statističku srednju količinu informacija po jednom simbolu izvora

$$\sum_{i=1}^q P(s_i) Q(s_i) = \sum_{i=1}^q P(s_i) \log_2 \frac{1}{P(s_i)}$$

gde je $P(s_i)$ verovatnoća diskretnog stanja s_i .

Izraz na desnoj strani predhodne jednačine formalno je identičan entropiji u statističkoj termodinamici, pa je zato usvojeno da se u Teoriji informacija koristi pojam entropije izvora kao merilo prosečne neizvesnosti o pojavljivanju nekog simbola sa liste S posmatranog izvora informacija. Prema definiciji, pišemo da je entropija izvora informacija

$$H(S) = \sum_{i=1}^q P(s_i) \log_2 \frac{1}{P(s_i)} [b/\text{simb.}]$$

pri čemu veličina entropije numerički označava prosečan broj bita po jednom simbolu izvora. Pošto je entropija funkcija verovatnoće pojavljivanja simbola, to raspodela verovatnoće utiče na vrednost entropije.

Primer 1.1 *Posmatra se srpski jezik kao izvor informacija a slova kao moguće poruke. Predpostavlja se da su verovatnoće pojavljivanja svih suglasnika jednake P_1 a samoglasnika P_2 . Takođe važi jednakost da je verovatnoća pojavljivanja samoglasnika pet puta veća od verovatnoće pojavljivanja suglasnika $P_2 = 5 P_1$. Potrebno je odrediti:*

- a) *količinu informacija za jedan suglasnik i jedan samoglasnik,*
- b) *entropiju srpskog jezika.*

Za izračunavanje logaritma sa osnovom dva može se koristiti sledeć formula $\log_2 x = \log_{10} x / \log_{10} 2$.

Srpski jezik ima dvadesetpet suglasnika i pet samoglasnika. Zbir verovatnoća mora biti jednak jedinici $25 P_1 + 5 P_2 = 1$. Pošto je $P_2 = 5 P_1$ lako se izračunava $P_1 = 0.02$ i $R = 0.1$.

- a) *Količina informacija za jedan suglasnik je $Q_1 = -\log_2(P_1) = 5.64$ b, a za jedan samoglasnik $Q_2 = -\log_2(P_2) = 3.32$ b.*
- b) *Entropija srpskog jezika iznosi $H = 25 P_1 Q_1 + 5 P_2 Q_2 = 4.48$ b/simb.*

Kao što je količina informacije u datom sistemu mera za stepen organizovanosti, tako je entropija datog sistema, mera za stepen njegove dezorganizovanosti - jedno je jednostavno obrnuto od drugog. Glavni stavovi matematičko statističke teorije informacija su:

- Organizovanost sistema i njegova entropija stoje jedna naspram druge kao antiteze. Povećanje stepena organizovanosti sistema dovodi do smanjenja entropije i obrnuto.
- Organizovanost je moguća i povećava se pomoću informacija.
- Organizovanost sistema je funkcija optimizacije količine informacija u sistemu.
- Neodređenost sistema - njegovu entropiju u celosti pobeđujemo samo potpunom informacijom. - Entropiju umanjuje ne bilo koje saznanje o sistemu već samo određeno i nazivamo ga informacijom.
- Mera entropije je mera odnosa i uzajamne povezanosti pojava i procesa.
- Entropija zavisnih događaja uvek je manja od entropije nezavisnih događaja.

- Količina informacija koja se sadrži u nekom događaju mora da bude tolika kolika je njegova entropija da bi se obezbedio minimum egzistencije i razvoja

W. R. Ashby pojam informacije vezuje za pojam raznovrsnosti. Količina informacije izražava količinu, broj raznovrsnosti. "Sistem koji ostvaruje svrsishodan izbor na višem stepenu od slučajnog izbora, postiže to kao rezultat informacija".

Ne ulazeći u analizu navedenih definicija informacija primenjujemo njihove zajedničke komponente. Prema većini autora, informacija je sadržaj neke poruke, zatim informacija se prenosi među elementima sistema, sistema i okoline i kao supstrat upravljanja usmerava ponašanje sistema, informacija umanjuje neodređenost u ponašanju sistema.

Mere količine informacija su:

- Jedinična količina informacija.
- Srednja količina informacija.
- Maksimalna količina informacija.
- Redundansa informacija koja može biti apsolutna i relativna.

Kao što je napomenuto, jedinični nosilac informacije je - bit, i on sadrži jediničnu količinu informacija. Srednja količina informacije je količina informacija dobijena izborom (događanjem) jednog iz više mogućih događaja sa različitim vrednostima verovatnoća. Maksimalna količina informacija, jednaka je količini informacija nastaloj događanjem jednog iz više mogućih događaja sa podjednakim vrednostima verovatnoća. Redundansa je jednaka razlici maksimalne i srednje količine informacija - apsolutna redundansa, i toj razlici prema jedinici maksimalne količine - relativna redundansa. Jedinice mere količine informacija su bit (b), bajt² (B), kilobajt (kB), mega bajt (MB), giga bajt (GB), terabajt (TB), i petabajt (PB). Jedan bajt=8 bita predstavlja najmanju jedinicu za merenje kapaciteta memorije računara. U računarskoj tehnici je uobičajeno da se koristi binarna aritmetika pa je 1 kB veći od 1000 bajta. U Tabeli 1.1 je dat broj bitova koje sadrže kilobajt, megabajt, gigabajt, terabajt i petabajt.

²Bajt je grupa od osam bitova; zato se ponekad naziva i oktet.

Tabela 1.1: Jedinice mere količine informacija

1 kB	2^{10}	=1 024	bajta
1 MB	2^{20}	=1 048 576	bajta
1 GB	2^{30}	=1.074x10 ⁹	bajta
1 TB	2^{40}	=1.099x10 ¹²	bajta
1 PB	2^{50}	=1.126x10 ¹⁵	bajta

Od pojma informacije treba razlikovati pojam podatka. Razlika se sastoji u sledećem:

- Informacije nisu bilo koji podatak već onaj koji umanjuje neodređenost u ponašanju sistema.
- Informacije se mogu ali i ne moraju sadržati u podacima.
- Podaci su još nevrednovane činjenice o nekoj pojavi, dok su informacije znanja iz organizovanih i analiziranih podataka.
- Informacija je više nego sam podatak. Informacija uključuje i ocenu o pojavi za koju postoji podatak.
- Informacija je uvek usmerena prema korisniku i zavisno od količine novog znanja koju on dobija porukom i od mere uma-njenja neizvesnosti koja je prethodno postojala u vezi sa događajem, on je primio veću ili manju količinu informacije.
- Informacija se u materijalno-energetskom obliku prenosi signalima: svetlosnim, zvučnim, magnetnim, električnim, hemijskim, biohemijskim i drugim. Dakle, podatke i informacije sadržane u njima prenosimo prirodnim ili veštačkim komunikacijama uz pomoć signala.

Ove karakteristike su date, uglavnom, prema stavovima matematičko-statističke teorije informacija. Ona se nije bavila semantičkim aspektom (smislom) prenosa podataka, već isključivo mehanizmom komuniciranja. Ona je dala doprinos upravo u oblasti kvantifikacije informacija, kvantifikacije njihovog prenosa, pronalaženja univerzalnih mera prenosa informacija kroz prostor i vreme. Sam razvoj društva, opšte civilizacione, poslovne i druge potrebe uslovljavaju sve brži, jednostavniji, masovniji i raznovrsniji prenos informacija. Razvoj informacione tehnologije omogućava na ovom području zadovoljenje tih potreba.

1.4 Komponente informacionog sistema

Osnovne komponente informacionog sistema su: hardver, programi, podaci, procedure i kadrovi. Iako, striktno uzevši, informacioni sistem ne podrazumeva obavezno podršku računara, ovde ćemo govoriti samo o takvim informacionim sistemima, dakle o informacionim sistemima baziranim na računarskoj podršci: hardveru i odgovarajućim programima.

Današnji računari se uglavnom baziraju na fon Nojmanovoj (von Neumann) koncepciji računara koju je ovaj naučnik predložio još 1946. godine i koju karakterišu sledeće četiri osnovne karakteristike:

- programsko upravljanje sa programima smeštenim u operativnoj memoriji,
- definisanje adresa operanda, a ne samih operanada,
- sekvencijalno izvršavanje naredbi, i
- naredbe i podaci su istog oblika tj., naredbe se ne razlikuju od podataka.

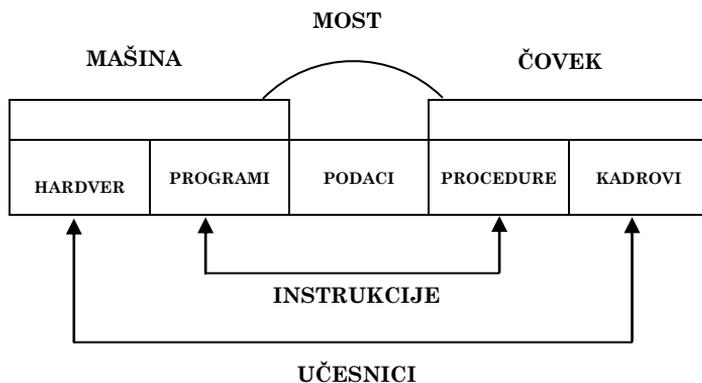
Računari sa kojima se srećemo u svakodnevnom životu su projektovani upravo na ova četiri principa.

Nabrojane komponente informacionog sistema su zajedničke svim tipovima informacionih sistema. Specifičnosti svake strukture biće posebno obrađene. Računari i čovek su opšti entiteti svakog informacionog sistema. Vezu između njih čine podaci. I računar i čovek (kadrovi) izvršavaju specifične instrukcije. Računarske instrukcije uzimaju formu programa, dok čovekove instrukcije uzimaju formu procedura. Povezanost ovih komponenti najbolje ilustruje Slika 1.1.

1.4.1 Hardver

Osnovu za izradu hardvera računarskog sistema predstavljaju digitalna logička kola koja operišu sa samo dve cifre: 0 i 1. Sve informacije unutar informacionog sistema prikazane su u obliku nizova binarnih brojeva³. Ovi brojevi se organizuju u veće celine

³Binarni brojni sistem, koji za predstavljanje brojeva koristi samo dve cifre: 0 i 1, potiče iz 1679. godine i pripada nemačkom matematičaru Lajbnicu.



Sl. 1.1: Odnosi komponenata informacionog sistema

radi lakšeg pamćenja i obrade.

Hardver odnosno računar u fizičkom smislu reči, se može kategorisati uglavnom po kriterijumu funkcije podataka (ulazni, obrađeni, memorisani i izlazni) na:

- hardver za unos podataka (Input hardware);
- hardver za obradu podataka (Processing hardware)
- hardver za memorisanje podataka;
- hardver za izlaz (Output hardware).

Za povezivanje IBM personalnih računara sa hardverom za unos, obradu, memorisanje i štampanje podataka, može se koristiti USB magistrala koju je razvila firma Intel. USB magistrala obezbeđuje priključivanje do 127 uređaja kao sto su CD-ROM čitači i pisači, štampači, modemi, idr.

Hardver za unos podataka

Hardver za unos podataka ima zadatak da sve podatke pretvori u električne signale (električni ekvivalent podatka) i da ih prenese do hardvera za obradu (processing hardware) i hardvera za memorisanje. Najčešće jedinice hardvera za unos su kombinacija tastature i ekrana. Ekran služi za verifikaciju podataka koji se unose preko tastature.

Jedinice hardvera za unos su u bliskoj komunikaciji sa ostalim jedinicama hardvera, naročito sa procesing hardverom. Koristi se takođe analogni uređaj za unos podataka, zvani miš. Veoma je upotrebljiv u grafički orijentisanim aplikacijama.

Skener je takođe deo hardvera za unos. On se koristi za transformisanje tekstualnih podataka ili grafičke slike u, za računar, osetljive signale.

Hardver za unos obuhvata i ostale uređaje kao što je modem kojim se povezuje računar telefonskom linijom za drugi računar, itd.

Hardver za obradu podataka

Hardver za obradu podataka obuhvata uređaje za računanje, upoređivanje i izvršavanje instrukcija.

Centralna jedinica računarskog sistema za obradu podataka - CPU (Central Processing Unit) sadrži upravljačku jedinicu, aritmetičko-logičku jedinicu - ALU (Aritmetic logic Unit), glavnu ili operativnu memoriju (Main Memory) kao i druge komponente.

Kao podrška personalnih informacionih sistema koriste se mikro računari. Takođe, mikro računari se koriste kao podrška i ostalim tipovima informacionih sistema, kao informacionim sistemima radne grupe i informacionim sistema organizacije. U ovim informacionim sistemima mikro računari su međusobno povezani preko lokalnih računarskih mreža LAN (Local Area Network) radi razmene podataka.

Kako poluprovodnička i ostala informaciona tehnologija proizvodi komponente mikro računara velikih mogućnosti, tako se mikro računari približavaju ostalim tipovima računara, kao što su su mini računari (Mini Computers) i veliki računari (Main frame Computers).

Hardver za memorisanje podataka

Hardver za memorisanje podataka su različiti mediji za prihvatanje, memorisanje i čuvanje podataka. Postoje više vrsta tih medija, najvažniji je disk koji se javlja kao ugrađen ili čvrst disk (Winchester Fixed Disk), zatim kao izmenjivi disk ili diskete (Floppy Disk) od 3.50 inča, gde dimenzija označava prečnik diska. Flopi disk je sačinjen od milara prekriven gvozdanim oksidom i zaštićen plasičnim omotačem. Za upisivanje i očitavanje podataka koristi se elektromehanički uređaj Floppy disk drive.

U zadnje vreme sve više se koristi CD (Compact Disk) disk, koji se zasniva na laserskoj tehnologiji kako pri čitanju diska tako i pri upisu podataka. On se sastoji od ispoliranog nemagnetnog metalnog diska sa plastičnom zaštitom tako da se može čitati optičkim uređajima, odnosno laserskim snopom svetlosti. Naziv "Laser" je akronim izraza "pojačanje svetlosti stimulisanim emitovanjem zračenja" (engl. Light Amplification by Stimulated Emission of Radiation) i označava usnopljeni zrak svetlosti. Takav snop svetlosti se može dobiti i kada Sunčevu svetlost usmerimo kroz optički sistem sočiva ili običnu lupu. Laserska svetlost se razlikuje od Sunčeve svetlosti po tome što emituje samo jednu svetlosnu frekvenciju, koja stvara fokusirani zrak po kome je laser prepoznatljiv, i ta svetlost je "koherentna".

Svetlost koju prima ljudsko oko prepoznajemo kao belu, bilo da je svetlost svetiljke ili Sunca. Međutim, mi ne primećujemo spektar boja koji čini belo, osim kroz prizmu kroz koju se svetlost može razložiti na komponente. Ovaj eksperiment pokazuje da se svetlost sastoji od mešavine zraka različitih frekvencija, gde svaka frekvencija odgovara jednoj boji koju naše oko registruje. Laserski uređaj iz te mešavine generiše samo svetlost jedne boje. Crveni laser proizvodi svetlost niže frekvencije, dok plavi laser proizvodi svetlost više frekvencije.

U današnje vreme najviše se koriste poluprovodnički laseri, uglavnom zbog manje snage, dobrog iskorišćenja uložene energije i malih dimenzija. Mogu da rade u kontinualnom ili prekiračkom režimu. Kompakt disk se sastoji od spiralno oblikovanog zapisa koga čini niz rupica i ravnih delova. Veličina jednog delića tog zapisa meri se nanometrима (milijarditim delovima metra). Laser služi da očita zapis i da ga dekodira.

Magneto-Optical Disk Drive, gde se takođe laserom čita i piše od nekoliko stotina miliona do biliona karaktera.

Hardver za izlaz

Hardver za izlaz čine razne vrste štampača, počev od serijskih, linijskih do savremenih laserskih, ploteri kao i jedinice za govorni izlaz. Elasočni izlaz iz računara se danas široko koristi u svakodnevnoj upotrebi kod dobijanja različitih telefonskih in-

formacija - odgovora od pošte, banke, biblioteke isl.

Uređaji za govorni izlaz iz računara su, za sada, dostigli veću primenu od uređaja za govorni ulaz u računar. U osnovi postoje dve metode za generisanje govornog izlaza.

Prva se sastoji u pretvaranju reči u u digitalni oblik koji se memoriše na magnetnom disku. Izlazna poruka se formira od potrebnih reči koje se čitaju sa diska, a zatim se ponovo pretvaraju u analogni signal, koji se pojačava i na kraju vodi u zvučnik. Ovaj metod zahteva veliki kapacitet memorije, ali može sa malim brojem reči da formira veliki broj poruka.

Kod druge metode govor se elektronskim postupkom generiše iz fonetskog teksta. Za svaku fonetsku jedinicu se izračunavaju parametru tona (jačina, visina, boja) koji se vode u sintisajzer; ovaj zatim proizvodi veštački glas koji odgovara fonetskom tekstu. Ovaj metod zahteva manji kapacitet memorije i usavršen je zahvaljujući računarima.

1.4.2 Programi

Programi informacionog sistema mogu se podeliti u dve osnovne grupe:

- Sistemski programi.
- Uslužni programi.

Sistemski programi sadrže programe za organizaciju i upravljanje radom računarskog sistema, automatizaciju procesa i održavanje programa. U ovoj grupi su operativni sistemi, programski sistemi, komunikacioni softver, sistem za upravljanje bazom podataka i drugi programi.

Operativni sistem

Operativni sistem (engl. Operating System) je skup programa koji kontroliše rad računarskog sistema. Drugim rečima, operativni sistem reguliše rad ulaznih i izlaznih uređaja, upravlja operativnom memorijom i informacijama (datotekama), kao i interakciju korisnika i računara. Zavisno od toga da li opslužuje jednog ili više korisnika operativni sistem može biti "Singleuser"

sitem za opsluživanje jednog korisnika i "Multiuser" - sistem za opsluživanje više korisnika.

Isto tako zavisno od broja istovremeno aktivnih programa, operativni sistem može biti jednoprocetni i višeprocetni.

Od jednokorisničkih programa poznat je već zastareli MSDOS, dok je višeprocetni i višekorisnički operativni sistem poznat UNIX. UNIX operativni sistem omogućava obavljanje više procesa na jednom centralnom procesoru po principu podele vremena tzv "time sharing". Vreme odvijanja jednog procesa je malo tako da se stvara utisak da su svi procesi istovremeno aktivni.

UNIX operativni sistem omogućava aktiviranje interaktivnih programa (onih koji zahtevaju ulazne podatke sa tastature sistema) i pozadinskih procesa (koji raspolažu od početka potrebnim podacima i ne zahtevaju direktno učestvovanje korisnika).

U principu računar može da radi i bez operativnog sistema. Prvi računari nisu posedovali operativni sistem. Međutim, rad na računaru bez operativnog sistema bio bi jako složen. Izvršavanje svih aplikativnih programa se odvija posredstvom operativnog sistema. Pored toga, zadatak operativnog sistema je da organizuje optimalno korišćenje različitih resursa računara. Operativni sistem je taj koji omogućava da rad na računaru bude jednostavno i svima dostupno, a što se pre svega ogleda u:

- Načinu unosa podataka u računar,
- korišćenje različitih tipova datoteka, njihovom kreiranju i preuređenju,
- mogućnost korišćenja različitih perifernih uređaja,
- zaštiti programa i podataka,
- otkrivanju grešaka, itd.,

S obzirom na to da operativni sistem obavlja različite funkcije, očigledno je da je on sastavljen od čitavog niza programa koji obavljaju strogo specificirane zadatke.

Operativni sistem ima modularnu slojevitou strukturu. Osnovni deo operativnog sistema se naziva jezgro ili nukleus. Na jezgro se logički nadograđuju ostali moduli. Svaki sloj realizuje funkcije koje zavise samo od unutrašnjih, donjih slojeva. Zato se kaže da operativni sistem ima hijerarhijsku strukturu.

Nukleus predstavlja osnovnu vezu između hardvera i ostalih modula operativnog sistema. Zbog toga je nukleus neprestano prisutan u memoriji računara.

Korisnik dolazi u kontakt sa spoljnim slojevima operativnog sistema, što znači da kontaktira sa računarom posredstvom operativnog sistema, a ne direktno. Tako korisnik komunicira sa hardverom računara posredstvom operativnog sistema.

Jedna hardverska konfiguracija računarskog sistema može raditi pod različitim operativnim sistemima. Operativni sistem omogućava da se za istu hardversku konfiguraciju računara dobiju, korišćenjem različitih operativnih sistema, različite karakteristike računara. Operativni sistemi se međusobno bitno razlikuju po satavu, funkcijama i karakteristikama. Međutim, u svim operativnim sistemima se rešavaju, praktično, istovetni problemi.

Operativni sistemi se stalno razvijaju obezbeđujući korisnicima sve konformniji rad i efikasniju eksploataciju računara.

Uslužni programi

Uslužni programi (engl. Utility Software) sadrži programe za rešavanje konkretnih zahteva korisnika za obuhvatanjem, obradom određenih podataka i dobijanje određenih izveštaja (rezultata). Uslužni programi se obično se deli u tri grupe programa:

- jezičke procesore,
- servisne programe,
- pomoćne programe i potprograme.

U jezičke procesore spadaju programi-prevodioci tj. *kompajleri* i *inturpretiri*. Kompajleri (engl. Compiler) i interpreteri (engl. Interpreter) su programi koji prevode izvorni program u objektni program, tj. u program koji je preveden na mašinski jezik. Procesi rada kompajlera i interpretera razlikuju se u tome što kompajler transformiše izvorni program u objektni program kao jedinsvenu celinu od početka programa do oznake koja simbolizuje da predhodni niz naredbi predstavlja jednu programsku celinu, pa se tek potom može dati program pustiti u rad, dok interpreter

prevodi i izvršava naredbu po naredbu. Zato su u principu programi, koji su napisani u jeziku koji koristi kompajler, brži od programa koji su napisani u jeziku koji koristi interpreter.

Prilikom prevođenja izvornog programa u izvršni program koriste se uslužni programa tzv. *linkeri* (engl. Linker) i *loaderi* (engl. Loader), čiji je zadatak da u izvršnom programu unesu potprograme ili rezultate rada potprograma koji su navedeni u izvornom programu.

U servisne programe spadaju, između ostalog, i tzv. *editori* (engl. Editor) koji omogućavaju pisanje programa, zatim programi za sortiranje podataka, programi za rukovanje datotekama, programi za razne vrste testiranja, programi za prenos podataka sa jednog medijuma na drugi, itd.

U pomoćne programe se svrstavaju programi koje korisnik može da pozove iz neke biblioteke programa (programi za izračunavanje vrednosti trigonometrijskih i logaritamskih funkcija, kvadratnog korena, apsolutne vrednosti, isl.) i koji pri likovanju ili bivaju inkorporirani u izvršni program, ili oni jednostavno izvrše funkciju koju korisnik traži, pa korisniku programa šalje samo rezultat.

1.4.3 Podaci

Obrada podataka preko obrade datoteke (File processing) i preko obrade baze podataka (Data base processing) su dva osnovna načina obrade podataka u računarskim sistemima.

Kroz obradu u datotekama podaci se memorišu, obrađuju u odvojenim datotekama. Obrada podataka baze je takva da se podaci memorišu tako da može da im se pristupi preko slogova i njihovih veza. Prema Američkom nacionalnom institutu standarda, jezik SQL iz četvrte generacije postaje neophodan za formalizaciju adhos zahteva za bazu podataka.

1.4.4 Kadrovi

Kadrovi (LifeWare) su aktivni članovi informacionog sistema. Može se uvesti više kategorija u strukturi kadrova, zavisno od tipa informacionog sistema. Uglavnom se razlikuju dve osnovne

kategorije:

- Korisnici sistema.
- Profesionalci sistema: operateri, sistem analitičari, programeri i sistem administratori.

Da bi računar obavio neki zadatak potrebno je da čovek sa njime uspostavi komunikaciju. Komunikacija, tj. razmena informacija između više učesnika je moguća pod uslovom da učesnici koriste isti jezik. Prema tome, da bi korisnici i profesionalci informacionog sistema mogli da komuniciraju sa računarom mora postojati jezik koji je razumljiv kako njima tako i računaru. Sa čovekove tačke gledišta najpogodnije bi bilo kada bi sa računarom moglo da se komunicira na prirodnom jeziku, tačnije na maternjem jeziku. Problem je što današnji računari zahtevaju u komunikaciji apsolutnu jednoznačnost iskaza/naredbe, a prirodni jezici nemaju tu osobinu. Za komunikaciju sa računarom razvijeni su *programski jezici* čija je osnovna osobina da u njihovim iskazima nema dvosmislenosti. U stvari, računar "razume" samo najelementarniji programski jezik - *mašinski jezik*. Svi programi napisani na nekom višem programskom jeziku se u računaru moraju predhodno da prevedu na mašinski jezik. Azbuka mašinskog jezika se sastoji samo od dve cifre: 0 i 1.

1.4.5 Procedure

Procedure (Orgware) su instrukcije za kadrove. Korisnici zahtevaju procedure kako bi obezbedili standardne forme, uputstva za izvršenje operacija na računaru. Tu spadaju procedure iz oblasti svakog tipa programa:

- Iz operativnih sistema, na primer: kako se startuje sistem, kako se inicijalizuju programi i sil.
- Iz oblasti aplikativnih programa: kako se postavljaju zahtevi, kako se izvršavaju pojedine funkcije, kako se interpretiraju poruke o greškama.

Procedure su osnovna prateća dokumentacija uz računarsku podršku informacionih sistema i ima najčešće naslove: Tutorials, Reference manual, Users manual, i druge.

1.5 Pitanja za proveru znanja

1. Koje su osnovne komponente informacionog sistema?
2. Kolika je cena softvera u odnosu na celokupnu cenu uređaja?
3. Šta radi operativni sistem? U zavisnosti broja korisnika koje opslužuje kakvi mogu biti operativni sistemi? Navesti primere.
4. Koji pronalazak je omogućio zamenu mehaničkih uređaja kontrole i regulacije, elektronskim uređajima?
5. Šta sve može biti izvor informacija?
6. Kako se izračunava količina informacija nekog događaja? Kolika je informacija da će bačena kocka pasti na određenu stranu?
7. Izračunati entropiju srpskog jezika pod pretpostavkom da je verovatnoća pojavljivanja svih slova jednaka.
8. Izvor informacija generiše dve moguće poruke sa verovatnoćama $P_1 = p < 1$ i $P_2 = 1 - p$. Za koju vrednost p se dobija maksimalna entropija izvora informacija⁴?
9. Izvor informacija daje poruke kombinovanjem p simbola 0 i 1. Sve poruke su jednakoverovatne.
 - (a) Pokazati da broj mogućih poruka iznosi $q = 2^n$ i da je entropija maksimalna i da iznosi $H = n \text{ b/simb}$.
 - (b) Ispisati sve moguće poruke za $n = 3$.
10. Kako se izražava količina informacija? Koliko bajta sadrži jedan terabajt?
11. Nabrojati osnovne karakteristike na kojima se zasniva fon Nojmanova koncepcija računara.
12. Od kojih komponenata se sastoji informacioni sistem?
13. Nabrojati jedinice hardvera koje se mogu koristiti za unos podatak u računar.
14. Zašto su za komunikaciju sa računarom razvijeni programski jezici? Da li se jezičkim procesorima vrši prevođenje programskog jezika na mašinski jezik?

⁴Entropija izvora informacija je:

$$H = p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p}$$

Maksimalna vrednost se dobija za $p = 0.5$ i iznosi 1 b/simb .

15. Koje delove sadrži centralna jedinica računarskog sistema za obradu podataka?
16. Kakva je razlika između kompajlera i interpretera? Od koliko znakova se sastoji azbuka mašinskog jezika?

Glava 2

Predstavljanje podataka

Za predstavljanje podataka u memoriji računara koriste se različiti formati zavisno od toga o kakvim se podacima radi i koje će se operacije u procesu obrade podatka nad njima izvršavati. Zajedničko za sve formate je da se podaci u memoriji računara registruju u obliku binarnih nizova za čije se generisanje u osnovi koriste dva opšta koncepta predstavljanja podataka: binarni brojni sistemi i binarni kodovi.

2.1 Brojni sistemi

Pod brojnim sistemom podrazumevamo skup pravila definisanih za predstavljanje brojnih vrednosti podataka. U praksi danas najširu primenu imaju pozicioni brojni sistemi u koje se svrstava i dekadni sistem brojeva koji koristimo u svakodnevnom radu sa brojevima. Zajedničko za pozicione brojne sisteme je da se za predstavljanje brojnih vrednosti koristi određeni skup cifara, pri čemu vrednost koju označava određena cifra zavisi od pozicije na kojoj se u zapisu brojeva nalazi. Na primer, u dekadnom brojnom sistemu koristimo cifre: 0,1,2,3,4,5,6,7,8 i 9 ali u broju 777.7 cifra 7 označava četiri različite vrednosti: u prvom sličaju je to sedam stotina (7×100), u drugom sedam desetih (7×10), u trećem sedam jedinica (7), a u četvrtom sedam desetih delova je-dinice (7×0.1).

Ako jedna cifra ima uvek istu vrednost, bez obzira na kom

se mestu u zapisu nalazi, onda je to nepozicioni brojni sistem. Tipičan primer je rimski brojni sistem sa ciframa: *I, V, X, S, E, M*.

U opštem slučaju zapisu broja u pozicionom brojnom sistemu sa osnovom N gde je N broj različitih cifara brojnog sistema, oblika

$$(\mathbf{X})_N = x_p x_{n-1} \dots x_o, x_{n-1} \dots x_m, \quad (2.1)$$

odgovara brojna vrednost određena relacijom

$$\begin{aligned} (X) &= \sum_{i=-m}^n x_i N^i \\ &= x_n N^n + x_{n-1} N^{n-1} + \dots + x_0 + x_{-1} N^{-1} + \dots + x_{-m} N^{-m} \end{aligned} \quad (2.2)$$

gde su sa H označene cifre brojnog sistema

U računarskoj tehnici značajnu primenu imaju različiti pozicioni brojni sistemi među kojima su najznačajniji: binarni, oktalni i heksadecimalni. U svim ovim sistemima koriste se ista pravila za predstavljanje brojeva ali se razlikuju skupovi cifara pomoću kojih se izražavaju brojne vrednosti. U binarnom brojnom sistemu koriste se samo cifre 0 i 1, u oktalnom skup cifara $\{0,1,2,3,4,5,6,7,8\}$, a u heksadecimalnom skup: $\{0,1,2,3,4,5,6,7,8,9,A,V,S,B,E,R\}$. U tabeli 2.1 dati su neki primeri dekadnih brojeva i odgovarajući ekvivalenti u binarnom, oktalnom i heksadecimalnom brojnom sistemu.

Tabela 2.1: Primeri brojeva u različitim brojnim sistemima

Dekadni	Binari	Oktalni	Heksadecimalni
1	1	1	1
7	111	7	7
8	1000	10	8
9	1001	11	9
15	1111	17	<i>P</i>
16	10000	20	10
17	10001	21	11

Predstavljanje brojeva u heksadecimalnom brojnom sistemu se vrlo široko koristi u računarskoj tehnici, naročito kada treba štampati ili prikazati binarne sadržaje, jer su zapisi kompaktniji i lakše se prate.

Pošto se u običnom životu najčešće koristi decimalni brojni sistem, potrebno je pri unošenju numeričkih podataka u računar pretvoriti decimalne brojeve u binarne. Takoće je potrebno po završetku obrade podataka u računaru izlazne rezultate iz binarnog pretvoriti u decimalni oblik.

U slučaju konverzije decimalnih brojeva u binarni oblik posmatraćemo odvojeno konverziju celih i razlomljenih brojeva. Konverzija celog decimalnog broja N u binarni broj može se primeniti primenom sledećeg algoritma:

1. Ispitati da li je N paran broj,
2. a) Ako je N neparan, zapisati 1 u rezultat i formirati novu vrednost $N = N - 1$. Skok na 3.
b) Ako je N paran, zapisat 0 u rezultat.
3. Naći novu vrednost deljenjem N iz koraka 2 sa 2.
4. a) Ako je $N > 1$, vratiti se na korak 1 i ponoviti postupak,
b) ako je $N = 1$ upisati 1 u rezultat.

Pri upisivanju jedinica i nula u rezultat prvo se upisuje *bit najmanje težine* (engl. Least Significant Bit - LSB) a ostali se upisuju levo od njega. Poslednji upisani bit je *bit najveće težine* (engl. Most Significant bit - MSB). Dobijeni broj je binarni ekvivalent decimalnog broja N .

Primer 2.1 Konverzija decimalnog broja 109_{10} u binarni broj. Postupak konverzije se može prikazati tabelarno na sledeći način

Broj	Aktivnost	Bit
109	Neparan - oduzmi 1	1
108	Podeli sa 2	
54	Paran - podeli sa 2	0
27	neparan - oduzmi 1	1
26	Podeli sa 2	
13	Neparan - oduzmi 1	1
12	Podeli sa 2	
6	Paran-podeli sa 2	0
3	Neparan - oduzmi 1	1
2	Podeli sa 2	
1	Kraj algoritma	1

Dakle, $109_{10} = 1101101_2$.

Pri konverziji pravog decimalnog razlomka koristi se sličan algoritam samo proces počinje određivanjem bita najveće težine, odnosno onog koji se nalazi neposredno iza tačke koja razdvaja celobrojni od razlomljenog dela broja. Algoritam za konverziju se sastoji samo od dva koraka:

1. Formirati novi broj N množenjem N sa 2.
2. a) Ako je $N > 1$, upisati 1 u rezultat i oduzeti 1 od N . Skok na korak 1.
 b) Ako je $N < 1$, upisati 0 u rezultat. Skok na korak 1

Postupak se završava kada se dobije $N = 0$.

Primer 2.2 *Konverzija pravog decimalnog razlomka 0.78125_{10} u binarni razlomak. Postupak konverzije se izvodi na sledeći način:*

Broj	Aktivnost	Bit	
0.78125	Pomnoži sa 2		
1.5625	Oduzmi 1	1	MZB
0.5625	Pomnož sa 2		
1.125	Oduzmi 1	1	
0.125	Pomnoži sa 2		↑
0.25	Pomnoži sa 2	0	
0.5	Pomnoži sa 2	0	
1.0	Oduzmi 1	1	LSB
0	Kraj algoritma		

Lakle, $0.78125_{10} = 0.11001_2$.

Treba primetiti da se osim ovakvog završetka procesa konverzije može desiti da se proces konverzije ne završi, jer se grupa bitova ponavlja. Kako svaki novi bit vredi polovinu predhodnog, konverzija se može prekinuti kada se proceni da je postignuta zadovoljavajuća tačnost.

Ako treba izvršiti konverziju decimalnog broja sa celobrojnim i razlomljenim delom, svaki deo se posebno konvertuje, a rezultat je binarni broj sa celobrojnim i razlomljenim delom.

Konverzija iz binarnog sistema u heksadecimalni se izvodi tako što se vrši grupisanje bitova binarnog broja u grupe po

4 bita počevši sa desne strane - LSB. Zatim se svaka grupa predstavi cifrom od 0 do F koja predstavlja vrednost grupe. Konverzija iz heksadecimalnog u binarni sistem vrši se tako što se svaka heksadecimalna cifra zameni grupom od četiri bita. Heksadecimalni brojni sistem se koristi za kompaktno predstavljanje podataka kod računarskih sistema, jer se svaki bajt predstavlja sa dve heksadecimalne cifre.

2.1.1 Binarni kodovi

Kao što je poznato, računar "razume" samo jezik cifara. Za predstavljanje podatka koristimo skup simbola $A_k = \{a_1, a_2, \dots, a_k\}$ i svakom podatku pridružujemo reč (niz simbola) nad azbukom A_k . Proces pridruživanja reči azbuke podacima naziva se kodiranje, a sam skup reči je skup podataka. Obrnuti proces u kome se osnovni skup reči datog koda odeđuje skup podataka naziva se raspoznavanje koda ili dekodiranje.

Kodiranje nalazi široku primenu u različitim oblastima ljudske delatnosti i predstavlja jedan od osnovnih koncepata predstavljanja podata u informatici. Ako se za predstavljanje podataka koriste reči iste dužine (sa istim brojem simbola) radi se o ravnomernim kodovima, dok u suprotnom govorimo o neravnomernim kodovima. Dekodiranje u slučaju ravnomernih kodova je jednostavnije pa se zati češće koriste.

Kodovi kod kojih se za predstavljanje podataka koriste kodne reči definisane nad azbukom koja sadrži samo dva simbola, na pr. $A_2 = \{0,1\}$, nazivaju se binarnim kodovima. Binarni kodovi nalaze široku primenu u informatici i računarskoj tehnici jer se jednostavno preslikavaju u odgovarajuće binarne memorijske sadržaje. Posebno značajani su binarno kodirani decimalni brojevi (BCB) kojima se predstavlja skup dekadnih cifara, ali i mnogi standardni kodovi (ISO 7, EBCDIC) kojima se predstavljaju znatno širi skupovi nenumeričkih podataka.

Binarno-decimalni kodovi u kojima se skup decimalnih cifara $\{0,1,2,3,4,5,6,7,8,9\}$ predstavlja određenim binarnim nizovima, formiraju se prema različitim principima. U tabeli 2.2 prikazani su neki od binarno decimalnih kodova. BCD kod 842 je poznat i pod nazivom prirodni binarno-decimalni kod, NBCD

kod, i ima najširu primenu u odnosu na sve ostale BCB kodove. Ovaj kod se svrstava u klasu težinskih kodova, kod kojih se kodiranje vrši tako što je svakoj od binarnih cifara u kodnoj reči pridružena neka težina. U slučaju ovog koda to su težine 8, 4, 2 i 1, tako da se vrednost (d) dekadne cifre koja odgovara kodnoj reči $b_3b_2b_1b_0$ dobija kao zbir proizvoda binarnih cifara i odgovarajućih težina. Odnosno važi sledeća relacija

$$d = 8b_3 + 4b_2 + 2b_1 + 1b_0 \quad (2.3)$$

Pored težinskih postoje i BCB kodovi koji se formiraju primenom nekih drugih pravila. Kod viška 3, je sličan prirodnom binarnom kodu i od njega se razlikuje samo po tome što se svakoj decimalnoj cifri prvo doda 3 pa se onda rezultat koduje NBCD **kodom**. Tako je nula kodirana kao cifra tri u prirodnom kodu, cifra jedan kao cifra četiri i tako redom.

Kod dva od pet sastoji se od reči dužine pet u kojima se uvek pojavljuju dve jedinice.

Tabela 2.2: Binarno dekadni kodovi

Cifre	8421	viška 3	2 од 5	Grejov	Hafmenov
0	0000	0011	00011	0000	0000000
1	0001	0100	00101	0001	1101001
2	0010	0101	00110	0011	0101010
3	0011	0110	01001	0010	1000011
4	0100	0111	01010	0110	1001100
5	0101	1000	01100	0111	0100101
6	0110	1001	10001	0101	1100110
7	0111	1010	10010	0100	0001111
8	1000	1011	10100	1100	1110000
9	1001	1100	11000	1000	0011001

Napomenimo da Grejov **kod** nije težinski **kod** te nije pogodan za izvođenje aritmetičkih operacija, ali je koristan kod analogno-digitalnih pretvarača, ulazno-izlaznih i drugih uređaja. Glavna karakteristika Grejovog koda je da se svaki kodni niz za susednu cifru, predhodnu ili sledeću, razlikuje samo u jednom bitu (jednoj poziciji). Grejov kod se može definisati za svih šesnaest binarnih nizova dužine 4 kao i za binarne nizove sa bilo kojom konstantnom dužinom.

Od binarno dekadnih kodova sa više od 4 bita posebno je interesantan 7-bitni Hafmenov kod, jer on spada u kodove sa otkrivanjem i ispravljanjem grešaka. Naime, ako pri prenosu kodnog niza neke dekadne cifre dođe do greške u jednoj binarnoj poziciji, prijemni uređaj može sam otkriti grešku i odmah je "automatski" ispraviti. Ako se greška javi u dve ili tri binarne pozicije istovremeno, greška će biti otkrivena ali se ne može ispraviti, o čemu uređaj signalizira. Označimo pozicije Hafmenovog koda sleva u desno u desno sa $AV8C421$, gde su A , B i C kontrolni bitovi, a 8, 4, 2 i 1 informacioni bitovi, kontrolni bitovi se dobijaju iz izraz

$$\begin{aligned}c_A &= c_8 \oplus c_4 \oplus c_1 \\c_B &= c_8 \oplus c_2 \oplus c_1 \\c_C &= c_4 \oplus c_2 \oplus c_1\end{aligned}\tag{2.4}$$

gde je znakom \oplus obeležena binarna *isključivo ili operacija*. O ovoj operaciji biće govora kasnije u delu o logičkim funkcijama.

2.1.2 Binarno kodovani decimalni brojevi

Kao što je poznato, decimalni brojni sistem se koristi u običnom životu dok je binarni brojni sistem pogodan za primenu u računarima. Binarno kodovani decimalni broj predstavlja kompromisno rešenje između ova dva sistema. Naime, svaka decimalna cifra višecifrenog decimalnog broja predstavlja se odgovarajućim binarnim brojem i tako se dobija *binarno kodovani decimalni broj* (BCB).

Prirodni binarni decimalni kod (NBCD) predstavlja decimalne cifre od 0 do 9 grupom od četiri binarne cifre. Konverzija decimalnog broja u NBCD broj je jednostavna i očigledna, kao i obrnuta konverzija. Međutim, izvođenje računskih operacija sa BCB brojevima je znatno komplikovanije nego sa pravim binarnim brojevima, pa se BCB brojni sistemi za predstavljanje binarnih brojeva malo koriste. Zadržali su se samo kod kalkulatora, uređaja sa numeričkim prikazivanjem i numeričkih tastatura. Kada se koristi BCB predstava broja potrebno je, radi izvođenja aritmetičkih operacija, pretvoriti takav broj u pravi binarni broj. Posle obrade binarni brojevi se, radi prikaza, mogu vratiti u BCB format.

Za konverziju binarno kodovanih decimalnih brojeva (NBCD) brojeva u prave binarne brojeve postoji više algoritama. Jedan od takvih algoritama je *algoritam sa sabiranjem*. Prema tom algoritmu se prvo svakoj binarnoj cifri NBCD broja pridruži odgovarajući težinski faktor. Idući sa desna u levo težinski faktori su: 1, 2, 4, 8, 10, 20, 40, 80, 100, 200, itd. Zatim se izvrši sabiranje težinskih faktora koji odgovaraju bitovima u NBCD broju čija je vrednost jedinica. U Tabeli 2.3 dati su težinski faktori u binarnom formatu.

Tabela 2.3: Težinski faktori za konverziju NBCD brojeva u binarne brojeve

Pozicija	Težinski faktor	Binarna vrednost
1	1	1
2	2	10
3	4	100
4	8	1000
5	10	1010
6	20	10100
7	40	101000
8	80	1010000
9	100	1100100
10	200	11001000
11	400	110010000
12	800	1100100000
13	1000	1111101000
14	2000	11111010000

Primer 2.3 *Kao primer, posmatrajmo proces konverzije broja 231_{10} u binarni broj pomoću opisanog algoritma sa sabiranjem. NBCD predstava broja 213_{10} je*

$$213_1 = 0010\ 0001\ 0011_{BCD}$$

Jedinice u NBCD broju su na pozicijama 1, 2, 5 i 10. Prema tabeli 2.3 težine koje odgovaraju ovim pozicijama su:

$$\begin{array}{r} 1 \\ 10 \\ 1010 \\ 110010 \end{array}$$

Sabiranjem navedenih težinskih faktora dobija se 11010101_2 što je binarni ekvivalent br ju 213_{10} .

Jedan od algoritama za konverziju binarnog u NBCD broj je *algoritam oduzimanja*. Koristeći Tabelu 2.3 nađe se najveći broj iz tabele koji je manji ili jednak od binarnog broja. Taj broj se oduzme od binarnog broja. Postupak se ponavlja sa ostatkom sve dok ostatak ne bude nula. Na svakoj poziciji gde je vršeno oduzimanje upiše se jedinica dok se ostale pozicije popune nulama. Rezultat je NBCD ekvivalent binarnog broja.

Primer 2.4 *Kao primer posmatrajmo konverziju binarnog broja 11010101_2 u NBCD broj. Postupak se može sprovesti na sledeći način:*

11010101	<i>Binarni broj</i>
11001000	<i>- 200 (Pozicija 10)</i>
00001101	<i>Ostatak</i>
00001010	<i>- 10 (Pozicija 5)</i>
00000011	<i>Ostatak</i>
00000010	<i>- 2 (Pozicija 2)</i>
00000001	<i>Ostatak</i>
00000001	<i>- 1 (Pozicija 1)</i>
00000000	<i>Kraj</i>

Dakle, konverzijom se dobija da važi jednakost $11010101_2 = 0010\ 0001\ 0011_{NBCD}$. Kao što se vidi, konverzijom je dobijen decimalni broj 213, što se i moglo očekivati s bzirom na konverziju u obrnutom smeru koji opisan u predhodnom primeru.

2.2 Predstavljanje numeričkih podataka

Format predstavljanja podataka je osnovno svojstvo na osnovu kojeg se mogu razlikovati različiti tipovi podataka. Svi podaci se prilikom registrovanja u memoriji računara svode na unapred definisane formate standardne dužine. Za standardne dužine formata podataka od 8, 16, 32 i 64 bita koriste se nazivi bajt, polureč, reč i dvostruka reč, respektivno.

2.2.1 Celobrojni podaci

Celi brojevi se obično definišu kao podaci tipa INTEGER. Za njihovo predstavljanje na u računaru koristi se notacija koja je uobičajena u matematici: niz cifara sa ili bez znaka. U memoriji računara za predstavljanje celih brojeva se koristi jedna reč

dužine 32 bita. Krajnje leva pozicija je pozicija znaka broja, dok se sve ostale koriste za predstavljanje apsolutne vrednosti broja. Pozitivni brojevi se predstavljaju tako što se direktno prevode u binarni brojni sistem i u poziciju znaka se upisuje cifra 0.

Potreba da se jedinstveni hardver koristi za operaciju sabiranja i oduzimanje dovela je do predstavljanja negativnih brojeva korišćenjem komplementa broja. Komplement broja može da bude potpun ili nepotpun. Nepotpuni komplement binarnog broja formira se tako što se svaka cifra zameni komplementarnom cifrom, odnosno nula se zameni jedinicom, a jedinica nulom. Potpuni komplement se dobija kada se nepotpunom komplementu na poziciji najmanje težine doda jedan. U slučaju i jednog i drugog komplementa broja na poziciji znaka broja nalazi se jedinica.

U standardnom formatu brojeva negativni brojevi se predstavljaju preko potpunog komplementa.

Primer 2.5 Standardni format broja 183

00000000000000000000000010110111

Standardni format broja - 183

111111111111111111111111101001001

Neka je X m -trocifreni binarni broj

$$X = \pm x_{m-1} x_{m-2} \dots x_1 x_0. \quad (2.5)$$

Njegov potpuni komplement je

$$Y = y_m y_{m-1} \dots y_1 y_0, \quad (2.6)$$

u kome je y_m cifra znaka. Broj X je određen sledećim izrazom

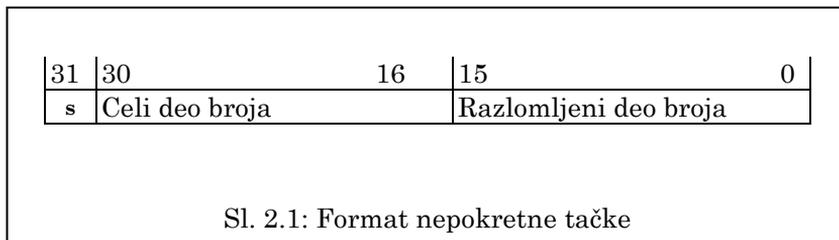
$$X = -y_m 2^m + y_{m-1} 2^{m-1} + y_{m-2} 2^{m-2} + \dots + y_1 2 + y_0 \quad (2.7)$$

Pored standardnog formata celih brojeva dužine jedne reči, koriste se i neki skraćeni formati od jedne polureči ili jednog bajta, ali i prošireni formati dvostruke reči. Princip registrovanja brojeva su isti kao kod standardnog celobrojnog formata.

Primer 2.6 *Dat je potpuni komplement $Y = 111010001$ nekog broja X . Naći broj X .*

$$X = -2^8 + 2^7 + 2^6 + 2^4 + 2^0 = -256 + (128 + 64 + 16 + 1) = -47$$

Za predstavljanje brojeva može se koristiti i format nepokretne tačke koji se sastoji u tome da se od jedne memorijske lokacije odedeni broj pozicija koristi za predstavljanje celobrojnog dela broja, a ostatak za razlomljeni deo. Pozicija najveće težine se odvajaju za znak broja. Na slici 2.1 prikazan je primer ovog formata.



Format nepokretne tačke ima ograničenu primenu zbog toga što znatno smanjuje opseg brojeva koji se mogu predstaviti u jednoj memorijskoj reči.

Pravila za aritmetičke operacije koja važe u dekadnom brojnom sistemu mogu se primeniti i u bilo kom drugom prirodnom brojnom sistemu sa osnovom q . Sabiranje dva broja vrši se sukcesivnim sabiranjem cifara počev od cifre najmanje težine. Pri tome je

$$a_i + a_j \begin{cases} a_k, \text{ gde je } a_k = a_i + a_j & \text{ako je } a_i + a_j < q \\ 1a_k, \text{ gde je } a_k = a_i + a_j - q & \text{ako je } a_i + a_j \geq q \end{cases} \quad (2.8)$$

gde su a_i , a_j i a_k cifre brojnog sistema sa osnovom q . Sabiranje dva višecifrena broja u pozicionom brojnom sistemu sa osnovom q vrši se prema sledećem algoritmu:

1. Sabrati dve cifre najmanje težine. Ako postoji jedinica prenosa zapamti se za sledću višu poziciju.
2. Sabrati dve cifre sledeće više pozicije i cifru prenosa iz predhodne niže pozicije.

3. Ako ima još pozicija u kojima nije izvršeno sabiranje ići na korak 2.
4. Ako je izvršeno sabiranje cifara u svim pozicijama i ne postoji prenos, postupak sabiranja je završen. Ako postoji prenos upisati 1 u sledeću višu poziciju, čime je postupak sabiranja završen.

Primer 2.7 Sabiranje binarnih brojeva 1100111 i 110110.

$\begin{array}{r} 1100111 \\ + 110110 \\ \hline 10011101 \end{array}$

Oduzimanje brojeva u pozicionom brojnom sistemu sa osnovom q vši se po pravilima koja važe u dekadnom brojnom sistemu. Pri tome se kod rada sa razlomljenim brojevima tačke razlomaka pišu u istoj poziciji po vertikali. Oduzimanje dva broja se vrši sukcesivno oduzimanjem cifara počev od najniže pozicije. Pri tome je

$$a_i - a_j \begin{cases} a_i - a_j & \text{ako je } a_i \geq a_j \\ q + a_i - a_j & \text{ako je } a_i < a_j \end{cases}$$

gde se za slučaj $a_i < a_j$ "pozajmljuje" jedna jedinica iz više pozicije koja sadrži c jedinica niže pozicije.

Primer 2.8 У бинарном бројном систему од бинарног броја 10101.11 одузети бинарни број 1001.001

$\begin{array}{r} 10101.110 \\ - 1001.001 \\ \hline 1100.101 \end{array}$

Predstavljanje negativnih brojeva u obliku komplementa omogućuje da se operacija oduzimanja obavi na sabiračima. Naime, operacija oduzimanja svodi se na operaciju sabiranja komplementa brojeva. Pri korišćenju potpunog komplementa pri sabiranju prenos iz pozicije znaka se gubi, dok se pri korišćenju nepotpunog komplementa prenos iz pozicije znaka naknadno dodaje sumi komplementa u poziciji najmanje težine. Pri sabiranju

komplemenata sabiraka može doći do prekoračenja kapaciteta korišćenog formata brojeva. Kriterijum za ustanovljenje prekoračenja je sledeći: ako je cifra prenosa iz pozicije znaka različita od cifre prenosa u poziciju znaka došlo je do prekoračenja. U tom slučaju dobijeni rezultat je pogrešan. Ispravan rezultat se može dobiti primenom dužeg formata brojeva.

Primer 2.9 *Primenom potpunog komplementa izračunati (a) razliku binarnih brojeva: $c = 111_2 - 101_2$ i (b) zbir negativnih binarnih brojeva $d = -100_2 - 110_2$. Za komplemente brojeva usvojiti dužinu od četiri bita. Vertikalna crta | odvajaja prenos iz pozicije znaka koji*

$\begin{array}{r} \text{prenos} = 1 111 \\ c = 1111 \\ + 1011 \\ \hline 0010 \end{array}$	$c = 7$ $- 5$ $+ 2$
---	---------------------------

(a)

$\begin{array}{r} \text{prenos} = 11000 \\ d = 1100 \\ + 1010 \\ \hline 0110 \end{array}$	$d = -4$ $- 6$ $- 10$
---	-----------------------------

(b)

se ignoriše jer se za predstavljanje brojeva koriste četiri bita.

Razlika brojeva je pozitivna sa vrednošću 10_2 , dok je zbir negativnih brojeva takođe pozitivan sa vrednošću 110_2 , što je pogrešan rezultat. Do očigledne greške je došlo zbog prekoraćenog kapaciteta korišćenog formata brojeva. Na to ukazuju različite vrednosti prenosa u poziciju znaka sume (0) i iz pozicije znaka sume (1). Produžavanjem formata brojeva na pet cifara dobija se ispravan rezultat:

$\begin{array}{r} \text{prenos} = 111000 \\ d = 11100 \\ + 11010 \\ \hline 10110 \end{array}$	$d = -4$ $- 6$ $- 10$
---	-----------------------------

Komplement je sada 10110 , što za sumu daje ispravnu vrednost -1010_2 .

U aritmetici koja koristi potpuni komplement, prekoračenje koje nastaje kao medjurezultat sabiranja više binarnih brojeva,

neće prouzrokovati grešku u konačnom rezultatu ukoliko je konačni rezultat u zadatom opsegu brojeva.

Primer 2.10 *Korišćenjem potpunog komplementa binarnih brojeva izračunati algbarski zbir brojeva $x = 27 + 16 - 18$. Komplemente brojeva predstavljati sa 6 binarne cifre. Sabiranjem brojeva 27 i 16*

$x = 011011$	$x = 27$
010000	<u>16</u>
<u>101011</u>	43
<u>101110</u>	<u>-18</u>
1 011001	25

došlo je do prekoračenja, jer je najveći broj koji se može predstaviti sa 6 binarne cifre je 31 (jedna cifra je rezervisana za znak). Konačni rezultat je ispravan.

Poznato pravilo za množenje koje se koristi u dekadnom brojnom sistemu primenjuje se i za množenje brojeva u binarnom brojnom sistemu. Razlika je jedino u tome što se pri množenju i sabiranju koriste pravila koja važe za binarni brojni sistem. Množenje višecifrenih binarnih brojeva vrši se tako što se množenik množi sa cifrom iz svake pozicije množioca, počev od cifre najmanje težine, parcijalni proizvodi se potpisuju jedan ispod drugog, pomereni za jedno mesto u levo, a zatim sabere

Primer 2.11 *Množenje brojeva 11001 i 110 u binarnom brojnom sistemu.*

<u>11001</u> × 110
0000000
00011001
<u>00011001</u>
10010110

Opisani postupak množenja binarnih brojeva pokazuje da se realizacija binarnog množenja u računaru svodi na višestruko binarno sabiranje i pomeranje za jednu binarnu poziciju. Dakle, za

množenje binarnih brojeva potrebno je da aritmetičko-logička jedinica računara sadrži samo sabirač i kola za pomeranje.

Množenje označenih binarnih brojeva, ako su negativni brojevi izraženi u potpunom kplementu, može se obaviti kao množenje neoznačenih brojeva, vodeći računa da bit znaka ima težinu -2^{n-1} . Iz tog razloga se parcijalnom proizvodu, koji potiče od bita najveće težine množioca, mora promeniti znak, odnosno poslednji parcijalni proizvod je potpuni komplement množenika. Pošto u označenom brojnom sistemu pozicija bita za znak mora da bude definisana, to se unapred određuje broj cifara proizvoda, koji mora biti veći ili jednak zbiru broja cifara množenika i množioca. Svim parcijalnim proizvodima treba povećati broj cifara na usvojeni broj cifara proizvoda, vodeći računa o znaku¹. Prilikom sabiranja ignoriše se eventualni prenos koji je posledica sabiranja bita znaka.

Navedena pravila su pokazana u sleđim primerima množenja četvorocifrenih binarnih brojeva, a usvojeno je da proizvod ima osam cifara.

Primer 2.12 *Množenje brojeva $0101_2 = 5_{10}$ i $1101_2 = -3_{10}$ u binarnom brojnom sistemu.*

$ \begin{array}{r} 0101 \times 1101 = 0\ 0000101 \\ 0000000 \\ 000101 \\ \underline{1101} \text{ potpuni komplement množenika} \\ 11110001 = -15 \end{array} $

Ako je množilac pozitivan, tada je poslednji parcijalni proizvod jednak nuli pa se može i izostaviti.

Primer 2.13 *Množenje brojeva $1101_2 = -3_{10}$ i $0101_2 = 5_{10}$ u binarnom brojnom sistemu*

¹Povećanje broja cifara kod pozitivnih brojeva se sasoji u dodavanju nula, dok se povećanje broja cifara kod negativnih brojeva sastoji u dodavanju jedinica. Na primer, povećanje broja cifara za pozitivan broj pet je $0101 = 00000101$, dok za povećanje broja cifara za negativan broj pet je $1011 = 1111011$. U oba slučaja bit najveće težine je znak broja.

$$\begin{array}{r}
 1101 \times 1101 = 11111101 \\
 0000000 \\
 \underline{111101} \\
 1 | 11110001 = -15 \\
 \uparrow \text{prenos koji se zanemaruje}
 \end{array}$$

Lako se može pokazati da je rezultat pozitivan ako su množenik i množilac negativni.

Primer 2.14 *Množenje brojeva $1101_2 = -3_{10}$ i $1011_2 = -5_{10}$ u binarnom brojnom sistemu.*

$$\begin{array}{r}
 1101 \times 1101 = 11111101 \\
 1111101 \\
 \underline{00011} \text{ potpuni kompsonent množenika} \\
 10 | 00001111 = 15 \\
 \uparrow \text{prenos koji se zanemaruje}
 \end{array}$$

Najjednostavnije pravilo za deljenje binarnih brojeva je isto kao i za deljenje decimalnih brojeva. Broj koji obrazuje cifre najveće težine deljenika se podeli sa deliocem, ostaku se pridružuje sledeća cifra neljenika i procedura se nastavlja sve dok ostatak ne bude manji od delioca. I u decimalnom i u binarnom sistemu deljenje može da se nastavi možeći ostatak osnovom brojnog sistema (pridružujući ostatku 0), a u količnici cela mesta treba odvojiti tačkom od razlomačkih.

Broj koraka deljenja ograničen je brojem pozicija za količnik, zbog čega se deljenje najčešće vrši zaokruživanjem, tj. približno. Ukažimo još i na to, da opisani postupak deljenja zahteva samo operaciju binarnog oduzimanja i pomeranje binarnih brojeva za jedno mesto u levo ili u desno.

Navedeno pravilo koje važi za neoznačene binarne brojeve, ilustrovano je u sledećem primeru.

Primer 2.15 *Deljenje broja $28_{10} = 100011001_2$ brojem $13_{10} = 1101_2$ u binarnom brojnom sistemu.*

Deljenje označenih binarnih brojeva je nešto složenije, međutim postoje pravila i algoritmi za deljenje brojeva čije su negativne vrednosti izražene u potpunom komplementu.

2.2.2 Format pokretne tačke

Za rad sa realnim brojevima u obradi podataka se najčešće koristi tip podataka REAL (pomekad se označava i sa FLOAT) koji se zasniva na predstavljanju podataka u formatu pokretne tačke. Format pokretne tačke može biti sa decimalnom tačkom ili u eksponencijalnom obliku.

Eksponencijalni oblik formata pokretne tačke² predstavlja se tako što se najpre prevede u binarni brojni sistem i zapis brojne vrednosti transformiše u eksponencijalni oblik tako da je

$$X = M \cdot 2^E, \quad (2.10)$$

gde je t mantisa, $1 \leq M = m.f < 2$, a E je eksponent broja. Broj se predstavlja tako što se u određenu memorijsku lokaciju upisuje cifra znaka broja, cifre mantise i cifre eksponenta broja. Eksponent se predstavlja u polarizovanom obliku $e = E + P$, gde je P polarizacija ili pomak veličine $P = 2^{k-1} - 1$, a k je broj binarnih cifara eksponenta. Ovom transformacijom se vrednosti eksponenta E iz opsega $[-2^{k-1} - 1, 2^{k-1}]$ pri polarizaciji $P = 2^{k-1} - 1$ pre-slikavaju u opseg vrednosti $[0, 2^k - 1]$ polarizovanog eksponenta e . Pošto celi deo normalizovane mantise mora biti jednak jedinici, moguće je izostaviti ovu jedinicu, i od mantise čuvati samo ra-zlomek. Zato se $m = 1$ u normalizovanoj mantisi $M = m.f$ naziva implicitna ili skrivena jedinica. Ovo rešenje koje je moguće samo samo za osnove eksponenta 2, omogućuje da se na račun skraćivanja polja mantise produži polje eksponenta za jedan bit, i time poveća opseg predstavljanja brojeva.

Za predstavljanje realnog broja u memoriji računara koristi se standardni format pokretne tačke dužine jedne memorijske reči,

²Standardi za aritmetiku sa pokretnom tačkom su: IEEE 754-85, IEC 559 i JIS N.R5.211

100011001:1101 = 10101 Količnik
<u>-1101</u>
1001
<u>-0000</u>
10010
<u>-1101</u>
1010
<u>-0000</u>
10101
<u>-1101</u>
1000 Celobrojni ostatak

kao što je prikazano na slici 2.2, sa poljima: znak, polarizovani eksponent i mantisa. Dužina ovih polja je 1, 8 i 23 bita.

31	30	23	22		0
8	Eksponent	Mantisa			

Sl. 2.2 Standardni format pokretne tačke.

Pored standardnog formata pokretne tačke koristi se i prošireni format koji se obično označava kao DOUBLE PRECISION. U programima za obradu podataka ovi podaci se uvek pred-stavljaju u eksponencijalnom obliku sa slovom B u zapisu: 1.009 D 01, 1.55 D —0.2.

Za predstavljanje jednog broja u ovom formatu koriste se dve memorijske reči tako da se formiraju binarni nizovi dužine 64 bita, kao što je prikazano na slici 2.3. Zapis broja se formira prema istim principima kao kod standardnog formata REAL s tom razlikom da su polja eksponenta i mantise šira. To omogućava da se predstave brojevi iz šireg opsega vrednosti i sa većom tačnošću.

Vrednosti eksponenta $e = 255$ kod standardnog formata pokretne tačke i $e = 2047$ kod proširenog formata pokretne tačke, je rezervisana za kodiranje beskonačne vrednosti ($\pm\infty$) i nebrojeva *NaN* (Not a Number).

63	62	52	51		0
8	EkspONENT	Mantisa			

Sl. 2.2 Prošireni format pokretne tačke

Pravila za interpretaciju standardnog formata pokretne tačka su sledeća:

- ako je $e = 255$ i $f \neq 0$ tada je $v = NaN$;
- ako je $e = 255$ i $f = 0$ tada je $v = (-1)^s \cdot \infty$;
- ako je $a < e < 255$ i tada je $v = (-1)^s \cdot 2^{e-127} \cdot 1.f$;
- ako je $e = 0$ i $f \neq 0$ tada je $v = (-1)^s \cdot 2^{-126} \cdot 0.f$
(denormalizovani broj);
- ako je $e = 0$ i $f = 0$ tada je $v = (-1)^s \cdot 0 = 0$.

Slična pravila se mogu formirati za interpretaciju proširenog formata pokretne tačke vodeći računa da polarizacija za ovaj format iznosi $P = 1023$.

Primer 2.16 *Dat je 32-bitni podatak*

1|10000001|0100000 00000000 00000000

Koji broj predstavlja prema standardu IEEE 754?

Iz datog podatka je $e = 1000001_2 = 129_{10}$; pa dati podatak predstavlja broj $v = 1.f \cdot 2^{e-127}$. Za polarizovani eksponent $e = 129$ nepolarizovani eksponent je $E = e - 127 = 129 - 127 = 2$. Takođe, iz datog podatka je $f = 0.01_2$ a $M = 1.f = 1.01_2$. Dakle, $v = M \cdot 2^E = 1.01 \cdot 2^2 = 101_2 = 5_{10} = 5$.

Sabiranje i oduzimanje brojeva sa pokretnom tačkom obavlja se svođenjem oba sabirka na isti eksponent. Pri tome veći sabirak mora ostati normalizovan pa se eksponent manjeg sabirka izjednačava sa eksponentom većeg sabirka. Naravno, to mora biti praćeno pomeranjem mantise manjeg sabirka u desno za broj binarnih pozicija određen razlikom eksponenta većeg i manjeg sabirka.

Ovim pomeranjem manji sabirak postaje denormalizovan. Dakle, za zbir brojeva $A = M_A \cdot 2^{M_A}$ i $B = M_B \cdot 2^{M_B}$ može se pisati

$$\begin{aligned} C &= MC \cdot 2^{e_c} = M_A \cdot 2^{e_A} \pm M_B \cdot 2^{e_B} \\ &= (M_A \cdot 2^{e_A - e} \pm M_B \cdot 2^{e_B - e}) \cdot 2^e \end{aligned}$$

gde je e eksponent većeg sabirka i u isto vreme eksponent zbira, $e_c = e$. Po potrebi se može normalizovati vrednost zbira S .

Prilikom množenja brojeva A i B treba najpre sabrati eksponente, a zatim oduzeti polarizaciju: $e_c = e_A + e_B - P$. U sledećem koraku se množe mantise i određuje znak proizvoda. Po potrebi normalizovati vrednost proizvoda, svesti vrednost mantise na 23 cifre iza binarne tačke. Oduzimanje polarizacije je neophodno da bi se eksponent zadžao u polarizovanom obliku.

Neka je posle normalizacije u nekom koraku aritmetičke operacije dobijena vrednost mantise $M = 1.f - 1 f - 2 f - 23 f - 24$ koju treba svesti na 23 binarne cifre. Svodenje razlomka sa 24 na 23 cifre može se izvesti na dva načina: odsecanjem i zaokruživanjem.

Odsecanje je postupak kojim se odbacuju dodatne cifre razlomka tako da se dobija $M = 1.f - 1 f - 2 \dots f - 23$. Maksimalna vrednost greške koja se tom prilikom čini je približno jednaka vrednosti jedinice u poziciji cifre $f - 23$ (2^{-23}). Za odsecanje kažemo da je polarizovano jer je greška uvek istog (pozitivnog) znaka.

Zaokruživanje je postupak kojim se mantisa oblika $M = 1.f - 1 f - 2 \dots f - 23 f - 24$ zamenjuje vrednošću dobijenom dodavanjem cifre $f - 24$ u poziciju cifre $f - 23$. Pri $f - 24 = 0$ zaokruživanje se svodi na odsecanje. U ovom slučaju zaokruživanjem se pravi maksimalna greska koja je približno jednaka polovini jedinice u poziciji cifre $f - 23$ (2^{-24}). Pri $f - 24 = 1$ zaokruživanjem se pravi greška koja približno iznosi $f - 23$ (2^{-24}). Dakle, greška koja se pravi zaokruživanjem nije polarizovana već je označena.

Aritmetičke operacije brojeva sa brojevima koji su predstavljeni pokretnom tačkom obavljaju se po posebnim pravilima. Njihova realizacija zahteva specifične postupke i aritmetičke blokove i složenija je za celobrojnu aritmetiku ili aritmetiku sa nepokretnom tačkom. Osim toga i vreme izvršenja aritmetičkih operacija sa pokretnom tačkom je duže.

2.3 Predstavljaje nenumeričkih podataka

Pored numeričkih računara može da obrađuje i podatke koji se mogu uslovno nazvati nenumerički podaci. To su podaci nad kojima se ne primenjuju klasične aritmetičke operacije. Pomenućemo dva najznačajnija: znakovni i logički tip podataka.

Znakovni tipovi podataka (Character) predstavljaju osnovu za rad sa podacima koji se predstavljaju rečima (nizovima znakova). Osnovu ovog tipa ovog čini skup znakova za čije se predstavljanje koriste unapred definisani standardni kodovi. Najšire primenjeni standardni kod za predstavljanje znakovnih podataka je kod ASCII – American Standard Code for Information Interchange koji predstavlja američku varijantu međunarodnog koda ISO 7. U Tabeli 2.4 prikazana je tablica ASCII koda u kojoj se svaki znak kodira binarnim nizom dužine 7 bitova. Kod registrovanja ovih podataka svaki znak zauzima jedan bajt s tim da je osmi bit bit parnosti (dopuna osnovnog koda do parnog broja jedinica). U Tabeli 2.4 su, uporedo sa ASCII kodom, date i izmene koje važe u našoj verziji ISO 7 koda definisane standardom JUSI.B1.002-1982.

Prve 32 kombinacije, kao i poslednja (*DEL*) predstavljaju kontrolne kodove za upravljanje periferijskim uređajima računarskih sistema i ne mogu se štampati. Brojevi su predstavljani kombinacijama od 30_{16} do 39_{16} , velika slova od 41_{16} do $5A_{16}$, a mala slova od 61_{16} do $7A_{16}$. Primećuje se da se kod za mala slova razlikuje od koda za velika slova samo po vrednosti bita b_5 tj. za 20_{16} . To olakšava konverziju velikih slova u mala slova i obrnuto. Ostale kombinacije predstavljaju znak za interpunkciju i specijalne znake.

Kao što se vidi, latinična slova specifična za našu abecedu ne mogu se predstaviti sedmobitnim ASCII kodom. Zbog toga je po Jugoslovenskom standardu (JUS) predviđeno da se 10 specijalnih znakova iz kodbne tabele 2.4 zameni sa 10 specifičnih slova naše abecede. Heksadecimalna kombinacija 40_{16} predstavlja slovo 2, kombinacije $5v_{16}$ do $5E_{16}$ predstavljaju mala slova š, đ, ć, i č respektivno. Na ovaj način zadžana je osobina sedmobitnog koda; da se kodovi za velika i mala slova razlikuju za 20_{16} .

Da bi se omogućilo predstavljanje posebnih slova važnijih evropskih jezika, grafičkih i drugih simbola uveden je prošireni

Tabela 2.4: Tablica ASCII koda.

	b ₇	0	0	0	0	1	1	1	1
	b ₆	0	0	1	1	0	0	1	1
	b ₅	0	1	0	1	0	1	0	1
b ₄ b ₃ b ₂ b ₁		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@ [Ž]	P	' [ž]	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	”	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	\$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	,	7	G	W	g	w
1 0 0 0	8	BS	CAN	(8	H	X	h	x
1 0 0 1	9	HT	EM)	9	I	Y	i	y
1 0 1 0	A	LF	SUB	*	:	J	Z	j	z
1 0 1 1	B	VT	ESC	+	;	K	[[Š]	κ	{ [š]
1 1 0 0	C	FF	FS	.	<	L	\ [Đ]	l	[đ]
1 1 0 1	D	CR	GS	-	=	M] [Ć]	m	} [ć]
1 1 1 0	E	SO	RS	.	>	N	^ [Č]	n	~ [č]
1 1 1 1	F	SI	US	/	?	O	_	o	DEL

osmobitni ASCII kod kod koga kodne reči 12816 do 25516, tj. one kod kojih je $b_7 = 1$, predstavljaju proširenje koda. Nažalost, ovo proširenje nije u potpunosti standardizovano. Problemi oko standarizacije su nastali zbog toga što su potrebe za specijalnim simbolima veće nego što ima mesta u kodnoj tabeli. U poslednje vreme se čine naponi za standardizaciju proširenog ASCII koda uvođenjem tzv. kodnih strana koje na jedinstveni način definišu proširenje koda. Sve kodne strane imaju identični sadržaj osnovnog dela kodne tabele; za kodne reči od 0 do 12710. Proširenje tabele, tj. kodne reči od 12810 do 255₁₀, definisano je prema potrebama pojedinih zemalja kojima je kodna strana namenjena. Na primer, pouklarni IBM PC računari koriste kodnu stranu sa oz-nakom 437 za potebe korisnika u SAD, dok je korisnicima u našoj zemlji nemenjena kodna strana 852 (tzv. Latin II).

U okviru aplikacija znakovni tip podataka se koristi za definisanje nizova, reči sastavljenih od osnovnih simbola iz ASCII tablice. Uobičajena notacija je da se u okviru programa ovi podaci označavaju kao reči ograničene navodnicama ili apos-

trofima. Na ovim podacima primenjuju se specifične operacije kao što su npr. spajanje nizova, izdvajanje dela niza, traženje zadatog dela podniza i druge.

Primer 2.17 *Na tastaturi je otkucana reč RADIO. Na izlazu tastature se generiše sledeć niz binarnih cifara:*

11010010 01000001 01000100 11001001 11001111

← *Prenos podataka u računar*

Razmak između svakog bajta unet je radi bplje preglednosti. Prenos podataka u računar se vrši sa leva u desno.

Uvođenje znakovni podataka omogućilo je pamćenje, obradu i štampanje u tekstualnom obliku podataka o licima ili objektima i njihovim osobinama, obradu teksta u kancelarijama. Značajno je i korišćenje znakovnih podataka za različita lingvistička istraživanja, npr. stvaranje rečnika, prevođenje tekstova, automatska sinteza govora, različite analize literalnih dela, itd.

Skup logičkih podataka LOGICAL obuhvata samo dve vrednosti TRUE i FALSE, koje se binarno obično predstavljaju tako što memorijska reč popunjena jedinicama označava FALSE, a svi drugi sadržaji TRUE. Nad logičkim podacima izvršavaju se osnovne operacije definisane u okviru matematičke logike.

2.4 Pitanja za proveru znanja

1. Kako se u računaru predstavljaju podaci?
2. Kako se dekadni brojevi konvertuju u
 - binarne,
 - oktalne i
 - heksadecimalne?
3. Kako se izračunava dekadna vrednost broja 10101101 ako je to:
 - binarni,
 - oktalni ili
 - heksadecimalni broj?
4. Odrediti komplement i potpuni komplement neoznačenog binarnog broja 01101101.
5. Izračunati zbir i razliku dva neoznačena binarna broja 10100010 i 11100111.

6. Kako se prikazuju binarni brojevi u formatu poretne tačke? Konvertovati decimalni broj 10 u binarni broj sa pokretnom tačkom.
7. Dekadni broj 1993 predstaviti NBCD kodom.
8. Pomnožiti u binarnom brojnom sistemu neoznačene binarne brojeve 1011.1 i 110.1.
9. U binarnom brojnom sistemu od neoznačenog binarnog broja 10101.11 oduzeti neoznačeni binarni broj 1001.001.
10. U binarnom brojnom sistemu neoznačeni binarni broj 1111101 podeliti neoznačenim binarnim brojem 101.
11. Predstaviti broj $X = 10_{10}$ u standardnom formatu pokretne tačke prema standardu IEEE 754.
12. Iskoristiti tabelu *ASCII* kodova za prikazivanje reči *COMPUTER* kao niz binarnih brojeva.

Glava 3

Računarski sistemi

Računari su elektronski uređaji koji su namenjeni za rad sa podacima, odnosno za obradu podataka. Oni spadaju u uređaje koji se nazivaju mašine ili automati.

U opštem slučaju termin mašina (automat) označava uređaj koji bez neposrednog učešća čoveka izvršava operacije dobijanja, pretvaranja (transformacije), prenosa i korišćenja energije, materijala ili informacija. Kada mašina funkcioniše prema unapred zadatom programu, ona se naziva programski upravljana mašina ili mašina sa programskim upravljanjem. Ovde termin program označava plan delovanja koji izvršava automatski uređaj. Program je sastavljen od konačnog skupa instrukcija ili naredbi, pri čemu svaka instrukcija opisuje neku elementarnu operaciju koju automatski uređaj može direktno da izvrši. Računar ili tačnije elektronski računar, predstavlja programski upravljanu mašinu, realizovanu digitalnim elektronskim kolima, koja nad podacima izvršava razne brojčane i nebrojčane operacije. Često se za računar koristi tačniji termin - računarski sistem ili sistem za obradu podataka, jer savremeni računar predstavlja kompleks uređaja koji zajedno, kao celina, obavljaju obradu podataka.

Proces intenziviranja proizvodnje i usavršavanje načina poslovanja, zajedno sa tehnološkim razvojem transporta i načina komuniciranja, značajno je povećao obim podataka i informacija i doveo je do tzv. informacione krize, odnosno do rekordnog porasta informacionih tokova. S druge strane, čovekove ograničene biološke sposobnosti za pamćenje, obradu i distribuciju podataka i

informacija i tradicionalna sredstva kojima je on raspolagao za rad sa njima, nisu bili u stanju da prate i reše nastali problem. Ovo je posebno važno ako se ima u vidu da nivo obezbeđenosti informacijama stručnjaka, naučnika i rukovodilaca predstavlja jedan od osnovnih faktora napretka društva.

Jedan od najvažnijih pronalazaka koji je omogućio nagli razvoj informatike bio je pronalazak nove vrste mašina (uređaja) koji su bili u stanju da pomognu čoveku u njegovom umnom radu. Time je čovekov rad na obradi podataka i informacija zamenjen radom mašine nazvane računarom, jer je njena prvobitna namena bilo mehaničko ("automatsko") izvršavanje računskih operacija. Danas su oblasti primene i mogućnosti računara znatno proširene, tako da "stari" termin računar više ne odražava u potpunosti suštinu njegovog korišćenja, već označava uređaj ili sistem koji prihvata podatke, izvršava nad njima operacije transformacije (razne numeričke ili nenumeričke operacije) i saopštava rezultate obrade.

Savremeni period u razvoju ljudskog društva često se naziva informaciona era ili era računara. Karakteriše se širokim korišćenjem sistema zasnovanih na računarima u kojima se prikupljaju, pamte (skladište), obrađuju, dostavljaju (distribuiraju) i koriste podaci i informacije. Nastao je kao posledica stalnog porasta brzine promena i višestrukog rasta svih društvenih funkcija i procesa, zajedno sa društvenim, tehnološkim i drugim vidovima razvoja.

Računari su posebno postali značajan činilac u obradi podataka i informacija u organizacijama kao sastavni deo sistema koji se nazivaju informacioni sistemi ili upravljački informacioni sistemi. Njihova je namena da podrže funkcije operativnog praćenja, rukovođenja i donošenju odluka.

3.1 Istorijski razvoj računara

Ljudi su od davnih vremena pokušavali da naprave uređaje za računanje. Najpoznatiji je abak (ili abakus) - računaljka u obliku sistema žica sa klizećim perlama. U XVII i XVIII veku bilo je više pokušaja izgradnje mehaničkih mašina za računanje, gde su se posebno isticali radovi Paskala, Lajbnica i Bebidža.

Smatra se da istorija razvoja računara počinje od elektronskih računara. Prvi elektronski računar ENIAC napravljen je 1946. godine. Te godine predložen je i model arhitekture računara sa upamćenim programom koja se i danas koristi a naziva fon Nojmanovom arhitekturom.

Dosadašnji razvoj računara opisuje se pomoću generacija računara - karakterističnih vremenskih perioda koji su obično određeni promenama u dominantnoj tehnologiji korišćenoj za realizaciju računara. Pri tome je svaka nova generacija računara omogućila uvođenje nove klase računara.

Opšta saglasnost o broju generacija računara uglavnom postoji, ali kod raznih autora postoje male razlike u početku i kraju trajanja svake generacije računara. Osim toga, period do elektronskih računara nije obuhvaćen generacijama razvoja računara, iako je bilo predloga za realizaciju računskih uređaja pomoću mehaničkih i elektromehaničkih komponenata.

Generacije računara su:

- Prvu generaciju (1946 -1955) karakteriše upotreba elektronskih cevi i kablovskih veza između komponenti što je uslovalo velike gabarite računarskih sistema, veliku potrošnju električne energije i česte kvarove komponenti računarskog sistema. Na primer, računar ENIAC je bio težak 30 tona, sastojao se sedamnaest hiljada elektronskih cevi i trošio je snagu od 174 kW, dok je za hlađenje elektronskih cevi bila potrebna isto tolika snaga. Programi su pisani na mašinskom jeziku što je zahtevalo specijalizovana programska znanja korisnika računarskog sistema.
- Druga generacija (1956 - 1963) se zasnivala na tranzistorima i štampanim kolima tako da su dimenzije znatno smanjene a povećala se sigurnost u radu. Povećani su i memorijski kapaciteti i usavršavani su ulazni uređaji. Softver se takođe usavršavao tako da se manje koristio mašinski jezik, a više simbolički jezici kao što su prve verzije programa Cobol i Fortran.
- U trećoj generaciji (1964 -1977) hardverske komponente su usavršene tako da se koriste integrisana kola malog¹ Small

¹Kola malog stepena integracije sadrže do 100 do komponenti na silicijumskoj podlozi

Scale Integration - SSI i srednjeg² Medium Scale Integration - MSI stepena integracije, što omogućuje još manje dimenzije računara, bržu obradu podataka, veći kapacitet memorije i veću pouzdanost u radu. Poboljšanje karakteristika osnovnih komponenti omogućilo je povezivanje više perifernih uređaja u računarski sistem. Usavršene komponente su omogućile multiprogramski rad, kao i komunikaciju između računara putem telefonskih linija. Za upravljanje i kontrolu računara razvijaju se operativni sistemi, a koriste se i viši programski jezici.

- U četvrtoj generaciji (1978 - 1989) tehnologija komponenti računarskog sistema se zasniva na LSI tehnologiji (Large Scale Integrated), tj. tehnologiji izrade elektronskih kola velike integracije³, i VLSI tehnologije (Very Large Scale Integration), tj. izrade elektronskih kola vrlo velike integracije⁴. Poboljšanje karakteristika i dimenzija hardverskih komponenti dovelo je do smanjenja dimenzija računara, povećanje kapaciteta operativne memorije i znatno povećane brzine obrade podataka. Operativni sistemi su fleksibilniji i jednostavniji za upotrebu širem krugu korisnika, a programski jezici i prevodioci pogodniji za kreiranje aplikativnih programa, što je dovelo do proizvodnje personalnih računara i radnih stanica.
- Peta generacija (od 1990.) zasnovana je na paralelnoj arhitekturi i višeprocorskim računarima.

Može se uočiti da su u svakoj novoj generaciji računara povećavani brzina rada procesora, kapacitet operativne memorije i sistemski softver, a korišćeni su sve bolji programski jezici. Od četvrte generacije računara počinje i sve masovnije povezivanje računara u računarske mreže.

Danas se radi na razvoju još jedne nove klase računara - neuroračunara, zasnovanih na korišćenju veštačkih neuronskih mreža

²Kola srednjeg stepena integracije sadrže od 100 do 1000 komponenta na slijumskoj ploči

³Kola velikog stepena integracije sadrže od 1000 do 10 000 komponenti na slijumskoj ploči

⁴Kola vrlo velikog stepena integracije sadrže preko 10 000 komponenti na slijumskoj ploči. Najsočepija savremena VLSI kola sadrže oko 125 000 000 komponenta na slijumskoj ploči.

sličnih neuronima mozga, koji će imati, verovatno, principijelno drugačiju arhitekturu od dosadašnje, a biće u stanju da znatno bolje rešavaju neke "teške" probleme. Naime, kognitivni (spoznajni) problemi kao što su prepoznavanje likova, učenje govora, razumevanje prirodnog jezika, pretraživanje tekstualne informacije u memoriji ili vođenje mehaničke ruke robota radi hvatanja predmeta su primeri problema koje brzo rešava ljudski mozak a koji su teški za rešavanje pomoću konvencionalnih računara.

3.2 Informacione tehnologije

Termin "informacione tehnologije" pojavio se krajem sedamdesetih godina. Uveden je sa ciljem istovremenog spajanja elektronike, računarske tehnike i telekomunikacija, oblasti koje su dovele do revolucionarnih promena u rukovanju, obradi i skladištenju podataka i informacija. Informacione tehnologije se mogu definisati kao: tehnička sredstva i metode za generisanje, skladištenje (čuvanje), prenos i korišćenje informacija. Na taj način informacione tehnologije označavaju skup metoda i načina za prikupljanje, unos, skladištenje (čuvanje), obradu, izdavanje (dostavljanje, diseminaciju) i korišćenje informacija.

Sve oblasti koje pripadaju informacionim tehnologijama mogu se prema specifičnosti primene uslovno svrstati u sledećih šest grupa:

- Sredstva i metode za obradu podataka - mikroelektronske poluprovodničke tehnologije, digitalna elektronska kola, druge vrste komponenata, memorijski medijumi, računarske arhitekture, lokalne računarske mreže, softversko inženjerstvo.
- Telematika (komunikacija podataka) - prenos podataka, mreže za prenos podataka, telekomunikacione mreže za integrisane usluge, optičke komunikacije, satelitske komunikacije.
- Uslužni telekomunikacioni servisi - baze podataka, elektronska pošta, teleteks sistemi, videoteks sistemi, informacioni servisi i sistemi za pretraživanje informacija.
- Birotika (automatizacija poslovanja) - obrada teksta, stono izdavaštvo, poslovni informacioni sistemi, audio i video

sistemi, telekonferencije, elektronski prenos dokumenata, mikrografija.

- Veštačka inteligencija - baze znanja, inteligentni sistemi, ekspertni sistemi, inteligentni roboti, komunikacija čovek-mašina, mašinski vid, učenje, neuronske mreže.
- Računarski integrisana proizvodnja - računarski integrisani proizvodni sistemi, računarska grafika, fleksibilni proizvodni sistemi, sistemi za automatizaciju projektovanja, računarsko upravljanje proizvodnjom, industrijski roboti.

Napomenimo da neki autori kao sastavnu komponentu informacionih tehnologija ne smatraju mikroelektroniku, već samo informatiku i telekomunikacije. Osim toga, ponekad se takođe sreću mišljenja da i telekomunikacije, iako se u velikoj meri zasnivaju na korišćenju računara, nisu sastavna komponenta informacionih tehnologija.

3.3 Obrada podataka

Savremeni život zahteva da se pamte podaci o mnogim promenama koje se dešavaju u našem okruženju. Broj različitih dokumenata koji se koriste udvostručava se svakih nekoliko godina.

Sistematska primena operacija nad podacima naziva se obrada podataka ili obrada informacija. Obradom se vrši transformacija (pretvaranje) podataka i informacija u oblik koji će moći da se koristi za određenu namenu.

Korišćenje računara počelo je obradom podataka, što je još uvek osnovna namena današnjih računara. Sa širenjem oblasti primene uloga računara pomera se od jednostavne obrade podataka na obradu informacija. Predviđa se da će se u budućnosti računari koristiti za obradu znanja.

Treba uočiti da su obrada podataka i obrada informacija definisani kao sinonimi, dok pojmove podaci i informacije treba razlikovati. Razlog je taj što se informacije dobijene obradom mogu podvrći novoj obradi radi dobijanja informacija "višeg" nivoa, tj. od jednih informacija dobijaju se obradom nove informacije.

U procesu obrade podataka vrši se prikupljanje, analiza, primena operacija (obrada), skladištenje (čuvanje, pamćenje) i dostavljanje (distribucija) podataka radi njihovog korišćenja.

Ako se obrada obavlja automackim sredstvima, pre svega računarima, tada se ona naziva automatska obrada podataka.

3.3.1 Operacije nad podacima

Pri obradi podataka nad njima se obavljaju raznovrsne operacije. Međutim, mogu se izdvojiti kao posebno karakteristične sledeće operacije nad podacima u toku njihove obrade:

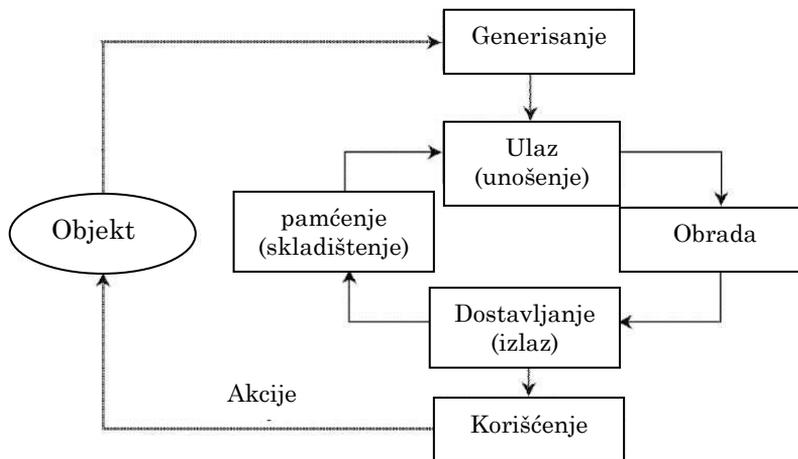
1. klasifikacija - ovom operacijom se izdvajaju (sređuju) podaci u određene klase koje za korisnika imaju neki smisao;
2. sortiranje (uređivanje) - predstavlja operaciju kojom se podaci smeštaju u rastući ili opadajući niz u odnosu na zadatak vrednost nekog atributa;
3. agregacija (sumiranje) - ovom operacijom formiraju se različiti sumarni podaci, npr. suma neto-zarada ili ukupan broj podataka koji zadovoljavaju neki uslov;
4. pretraživanje ovom operacijom traže se određeni podaci i obezbeđuje se pristup tim podacima;
5. izračunavanje (računanje) - ovom operacijom izvršavaju se razne aritmetičke ili logičke operacije nad podacima.

3.3.2 Proces obrade podataka

Proces obrade podataka odvija se globalno u sledeća četiri koraka (etape) koji se ponekad nazivaju ciklusom obrade podataka:

1. ulaz (unošenje),
2. obrada (transformacija),
3. dostavljanje (izlaz, komunikacija, distribucija),
4. pamćenje (memorisanje, skladištenje, čuvanje).

Ove četiri aktivnosti sukcesivno se ponavljaju na način prikazan na Slici 3.1, gde su prikazane i veze sa nastankom (generisanjem) podataka i njihovim korišćenjem.



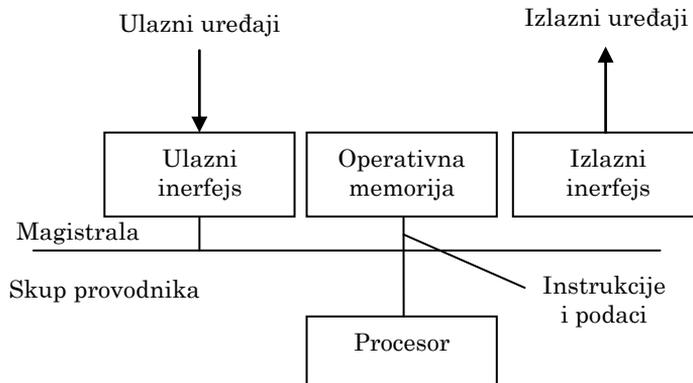
Sl. 3.1: Proces obrade podataka

Ulaz se sastoji u prikupljanju, konverziji i predstavljanju podataka na mestu nastanka i njihovom registrovanju u sistemu radi dalje obrade. Obrada ima osnovni cilj da ulazne podatke organizuje i transformiše, tako da se dobije informacija koja se može koristiti za predviđene namene. Etapa dostavljanja ili komunikacije sastoji se u pripremi čitljivih dokumenata (izveštaja, pregleda, tabela i sl.), njihovom prenosu i dostavljanju korisnicima. Uneti podaci, kao i podaci i informacije dobijeni obradom pamte se na masovnim memorijama radi daljeg korišćenja, arhiviranja, reprodukcije ili pripreme za kasnije nove obrade. Na osnovu primljene informacije primalac (čovjek ili mašina) donosi odluke i preuzima određene akcije. Odzivom objekta na preduzete akcije nastaje generisanje novih podataka koji se onda ponovo uključuju u proces obrade.

3.4 Struktura računara

Svaki računar se sastoji od pet osnovnih funkcionalnih celina koje se nazivaju jedinice računara. To su: ulazna jedinica, izlazna jedinica, memorija (sastoji se od osnovne i spoljne memorije), upravljačka (komandna) jedinica i aritmetičko-logička je-

dinica. Aritmetičko-logička i upravljačka jedinica često se razmatraju kao celina koja se naziva centralni procesor ili kratko procesor. Ulazni i izlazni uređaji često se nazivaju periferni uređaji. Na slici 3.2 prikazana je strukturna šema računara i tok podataka i upravljačkih signala.



Sl. 3.2: Strukturna šema računara

Podatke za obradu računar dobija od raznih ulaznih uređaja kao što su: tastatura, miš, jedinice magnetnih diskova, razni senzori (pretvarači), telekomunikacione linije, itd. Posle obrade u aritmetičko-logičkoj jedinici (procesoru) računar pamti rezultate obrade na spoljnoj memoriji ili ih prosleđuje na jedan ili više izlaznih uređaja, koji su takođe vrlo raznovrsni. To mogu biti: ekrani, razne vrste štampača, jedinice magnetnih diskova, telefonske linije, razni aktuatori, itd.

3.4.1 Ulazna jedinida

Preko ulazne jedinice unose se u memoriju dve osnovne vrste podataka za računar:

- (a) polazni podaci za obradu,
- (b) programi - uputstva računaru kakvu vrstu obrade i nad kojim podacima treba izvršiti.

Ulazni podaci za računar prethodno se pripremaju na nosiocima podataka, a unose se preko različitih ulaznih uređaja. Osim toga, postoji više različitih načina i uređaja za tzv. neposredno unošenje podataka bez njihove prethodne pripreme na posebnim posrednim nosiocima.

Za unošenje podataka i programa u računar (procesor, osnovnu memoriju ili spoljnu memoriju) koriste se ulazni uređaji. Prema načinu unošenja podataka, svi ulazni uređaji mogu se svrstati u dve grupe:

- uređaje za ručno unošenje,
- uređaje za automatsko unošenje.

Uređaji za ručno unošenje. - Postoji više načina za neposredno ručno unošenje podataka u računar bez korišćenja posebnih posrednih nosilaca. Pomoću uređaja za ručno unošenje podaci se unose relativno sporo (ne više od 10 znakova u sekundi). U grupu uređaja sa ručnim unošenjem svrstavaju se: upravljački pult računara, tastatura, ekran sa dodirrom, miš, svetlosna olovka, komandna palica (palica za igre, džojstik), grafička tabla (digitalizator) i dr. Ručno unošenje se koristi kada obim podataka nije veliki.

Uređaji za automatsko unošenje. Uređaji za automatsko unošenje podataka mogu se podeliti na uređaje sa neposrednim unošenjem i uređaje sa posrednim unošenjem. Pri neposrednom unošenju podaci se unose direktno u računar sa mašinski čitljivih dokumenata, kao što su razni obrasci, optički čitljivi dokumenti i dr.

Pri posrednom unošenju ulaznoj aktivnosti prethodi prikupljanje (obuhvatanje) podataka. Prikupljanje podataka se sastoji u zapisivanju podataka na posebne nosioce, najčešće na magnetne medijume.

3.4.2 Izlazna jedinica

Izlazna jedinica služi da se preko izlaznih uređaja izdaju iz računara rezultati obrade i prikažu u pogodnom obliku određenom korisniku, kao i da se oni zapamte (registruju) radi kasnijeg korišćenja.

Izlaz rezultata obrade, tj. izdavanje podataka iz računara može imati sledeće svrhe:

- (a) privremeno pamćenje na mašinski čitljivim nosiocima radi kasnije obrade;
- (b) saopštavanje korisniku u obliku izveštaja ili prikaz na ekranu;
- (c) neposredno korišćenje, kada se izlazni rezultati bez posredovanja čoveka koriste za daljinski prenos ili za automatsko preduzimanje akcija, npr. pri neposrednom upravljanju procesima, uključivanju i isključivanju određenih izvršnih uređaja i sl.

Izlazni uređaji mogu se podeliti u sledeće grupe:

1. uređaji za trajno pamćenje podataka na mašinski čitljivim nosiocima, npr. magnetne trake i diskovi;
2. uređaji za prikaz rezultata u obliku teksta ili grafika, gde spadaju štampači, ekrani, informacioni panoi, crtači itd.;
3. uređaji za izlaz podataka u okruženje radi njihovog neposrednog korišćenja, npr. za upravljanje tehnološkim procesima, daljinski prenos podataka preko telekomunikacionih linija i dr.

Zajedno sa tastaturom, mišem i svetlosnom olovkom ekran predstavlja danas najbolje sredstvo za neposrednu komunikaciju čoveka i računara. Na taj način ekran služi ne samo kao izlazni već i kao ulazni uređaj.

3.4.3 Centralni procesor

Centralna profesorska jedinica, centralni procesor ili, kratko, procesor predstavlja uređaj za upravljanje i obradu koji pod programskim upravljanjem vrši transformaciju podataka unutar računara, donosi odluke u procesu obrade, upravlja ostalim komponentama računara, obezbeđuje prenos podataka između komponentata računara, kao i razmenu podataka sa spoljnjim okruženjem. Kako je već rečeno, procesor se sastoji od aritmetičko-logičke i upravljačke jedinice.

Upravljačka jedinica upravlja tokom izvršenja operacija u računaru tako što na osnovu instrukcija (uputstva o potrebnoj obradi) programa određuje aktivnosti svih ostalih jedinica računara i prati njihovo obavljanje. Primeri tih aktivnosti su: ulaz podataka, pristup podacima radi pamćenja ili čitanja, redosled izvršenja operacija i izlaz rezultata.

Upravljačka jedinica obavlja dve osnovne funkcije:

- (a) određivanje redosleda instrukcija za izvršenje,
- (b) dekodiranje izabrane instrukcije, tumačenje pojedinih polja instrukcije i upravljanje izvršenjem operacije određene kodom operacije te instrukcije.

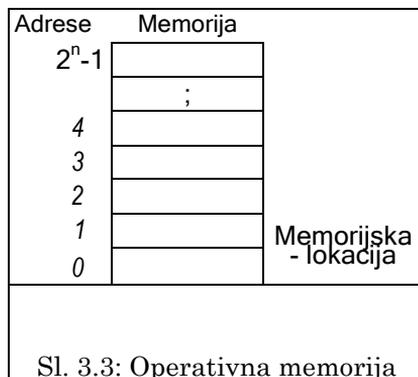
Bitnu komponentu procesora čini aritmetičko-logička jedinica namenjena za obradu podataka. Aritmetičko-logička jedinica je deo računara gde se izvršavaju operacije nad podacima. To su, pre svega, osnovne aritmetičke operacije: sabiranje, oduzimanje, množenje i deljenje, zatim logičke operacije, ali i operacije kao što je pomeranje binarnog sadržaja nekog registra levo ili desno i druge. O tome koja operacija i nad kojim podacima treba da se izvrši aritmetičko-logičkoj jedinici saopštava upravljačka jedinica.

3.4.4 Operativna memorija

Operativna memorija namenjena je za privremeno pamćenje podataka i programa, a sačinjena je od grupa bistabilnih memorijskih elemenata koje se pri pristupu memoriji (čitanju ili upisu) tretiraju kao neljdelive celine. Pored pamćenja podataka za obradu (ulazni podaci) i pamćenju podataka koji definišu postupak obrade (program), u operativnoj memoriji se takođe pamte međurezultati i konačni rezultati obrade.

Operativna memorija je adresno organizovana, pa se može posmatrati kao niz adresibilnih memorijskih lokacija, kao što je prikazano na slici 3.3. Princip adresibilnosti omogućava direktan pristup svakoj memorijskoj lokaciji. To znači da se određeni, unapred specificirani podatak može uneti (upisati) u određenu memorijsku lokaciju, kao i da se sadržaju svake memorijske lokacije može u svakom trenutku prići, tj. da se on može proći

tati. Na taj način je omogućeno prenošenje podatak iz jednog dela računara u druge njegove delove.

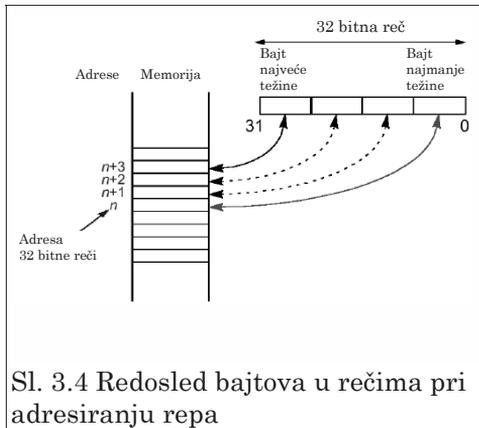


Osnovni element operativne memorije predstavlja memorijska lokacija. To je grupa memorijskih elemenata koja se pri pristupu memoriji (čitanju ili upisu) tretira kao nedeljiva celina. Adresa memorije predstavlja ceo pozitivan broj iz intervala $[0, 2^n - 1]$, gde je n dužina adrese, i njome se može adresirati 2^n memorijskih lokacija. Dužina memorijske lokacije iznosi 8 bita (1 bajt).

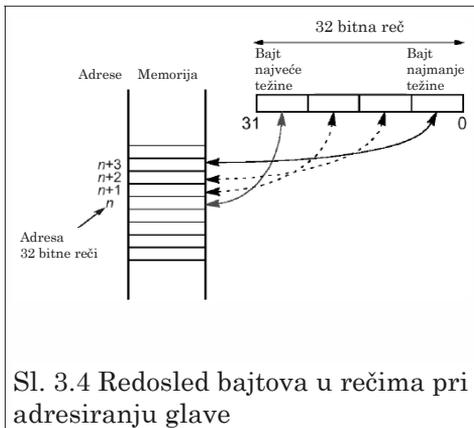
U računarima različitih proizvođača sreću se dva načina uređivanja bajtova unutar podataka sa dužinama većim od jednog bajta. *Pri adresiranju repa* (engl. Little endian byt order) adresa reči je i adresa bajta najmanje težine u reči. Ostali bajtovi sa rastućim težinama imaju rastuće adrese. *Pri adresiranju glave* (engl. Big Endian byt order) adresa reči je i adresa bajta najveće težine. Ostali bajtovi sa opadajućim težinama imaju rastuće adrese. Raspored bajtova u rečima za ova dva načina uređivanja bajtova prikazan je na slikama 3.4 i 3.5.

Važan kvantitativni parametar memorije je vreme pristupa, koje se definiše kao vremenski interval od prispeća zahteva za čitanje memorije do trenutka kada taj podatak postaje raspoloživ na izlazima podataka memorije. Pri zahtevu za upis podatak u memoriju, kraj ovog intervala određen je završetkom upisa.

Zavisno od tipa memorije mogu postojati ograničenja u pogledu



Sl. 3.4 Redosled bajtova u rečima pri adresiranju repa



Sl. 3.4 Redosled bajtova u rečima pri adresiranju glave

načina i redosleda pristupa lokacijama u memoriji. Kada nema nikakvih ograničenja imamo *memorije sa proizvoljnim pristupom* ili *slučajnim pristupom* (Eng. Random Access memory, RAM). Vreme pristupa kod ovih memorija je konstantno. Takve su memorije u poluprovodničkoj tehnologiji. Očuvanje podataka u ovim memorijama zahteva stalno napajanje elektručnom energijom, inače se sadržaj memorije gubi.

U okviru RAM memorija često se nalazi tzv. keš ili skrivena memorija, koja se nalazi između procesora i operativne memorije. To je brza memorija relativno malog kapaciteta u kojoj se čuvaju aktuelne instrukcije i podaci za aktivne programe. Obraćajući se memoriji procesor se najpre obraća keš memoriji. Ukoliko keš memorija ne sadrži podatak koji procesor zahteva, obraćanje procesora se prosleđuje do glavne memorije.

ROM memorija (engl. Read Only memory) je poluprovodnička memorija sa proizvoljnim pristupom čiji se predhodno upisani sadržaj može samo čitati. Sadržaj memorije se ne može se izbrisati ni u slučaju nestanka električne energije, kao u slučaju RAM memorije. Najčešće se koristi za trajno zapisivanje mikroprocesorskih programa, odnosno stalnih podataka koji su neophodni, napr., prilikom pokretanja računara.

Masovna (spoljna) memorija služi za pamćenje podataka velikog obima i njihovo arhiviranje, a znatno je sporija od operativne memorije. Medijume za ove memorije predstavljaju magnetni diskovi, optički diskovi (kompakt diskovi) i magnetne trake

(strimer trake). Za podatke manjeg obima koriste se diskete (izmenjivi diskovi). Masovna memorija služi za pamenje (skladištenje) podataka i programa koje treba sačuvati za duži period.

3.4.5 Stek memorija

Stek, magacin ili magacinska memorija predstavlja linearnu strukturu podataka kod koje se upis i čitanje elemenata podataka vrši po pravilu "poslednji upisan - prvi pročitano" (engl. LIFO – Last In First Out). Po ovom pravilu se slažu korpe u samousluzi, poslužavnici u ekspres restoranu ili meci u šaržeru (magacinu) automata odakle je i uzet termin magacin. Mesto gde se vrši upis ili čitanje elemenata podataka naziva se vrh steka.

3.5 Procesor

Po svojim funkcijama procesor predstavlja centralni uređaj računara koji neposredno upravlja procesom obrade i uzajamnom komunikacijom svih ostalih delova računara. Procesor izvršava instrukcije programa, ograničava obraćanje (pristup) osnovnoj memoriji, po potrebi inicira rad perifernih uređaja, prihvata i obrađuje zahteve koji dolaze od drugih jedinica računara i iz okruženja i dr.

3.5.1 Funkcije procesora

Procesor (centralni procesor, centralna jedinica) predstavlja programski upravljani digitalni uređaj koji obavlja sledeće funkcije:

- na osnovu instrukcija obrađuje podatke izvršavanjem nad njima određenih relativno prostih operacija mašinskih operacija;
- donosi odluke u procesu obrade o toku odvijanja izvršenja instrukcija programa;
- upravlja ostalim komponentama računara;
- obezbeđuje prenos podataka između komponenta računara, kao i razmenu podataka sa spoljnim okruženjem.

Procesor radi izvršavajući program smešten u operativnu memoriju, koji se sastoji od sledećih aktivnosti:

- prenos (pozivanje) svake instrukcije iz operativne memorije u upravljačku jedinicu;
- prenos podataka iz operativne memorije ili registara procesora u aritmetičko-logičku jedinicu;
- izvršenje (realizacija) operacije predviđene tom instrukcijom;
- pamćenje rezultata u operativnoj memoriji ili registrima.

Tako se izvršavanje programa odvija u neprekidnoj komunikaciji između operativne memorije, aritmetičko-logičke jedinice i upravljačke jedinice.

Instrukcijam programa procesoru se nalažu aktivnosti koje on treba da obavi. U slučaju operacije nad parom operanada procesoru se instrukcijom moraju dostaviti:

- informacija o operaciji koju treba da izvrši, u vidu koda operacije (*KOp*),
- oba izvorišna operanda ili informacije o mestima odakle ih može uzeti,
- informacija o mestu gde treba da pošalje rezultat (određišni operand), i
- adresa sledeće instrukcije programa.

Instrukcije koje sadrže sve nabrojane elemente bile bi četvoroadresne instrukcije (osim koda operacije imale bi četiri adresna polja). Umesto da se u svakoj instrukciji navodi adresa sledeće instrukcije, koristi se rešenje da se adresa sledeće instrukcije drži u posebnom registru nazvanom programski brojač (engl. program counter). Time se dobija troadresna instrukcija. Ovaj i još tri formata instrukcija prikazani su na slici 3.6

Dejstvo troadresne instrukcije pri izvršenju operacije \otimes , sa *Op2* i *Op3* kao izvorišnim operandima i *Or1* kao određišnim operandom je

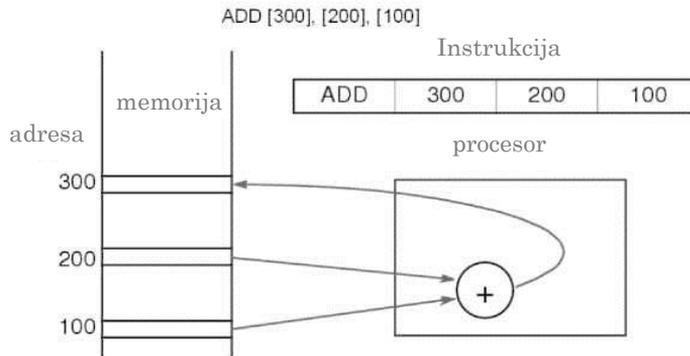
$$Op1 - Op2 \otimes Op3$$

Odlika ovog formata je da se pri izvršenju instrukcija ne uništava ni jedan od izvorišnih operanada.

F3	KOp	Op1	Op2	Op3
F2	KOp	Op1	Op2	
F1	KOp	Op		
F0	KOp			

Сл. 3.6: Formati troadresnih, dvoadresnih, jednoadresnih i nulaadresnih instrukcija

Primer 3.1 Na slici 3.7 je prikazano dejstvo troadresne instrukcije za sabiranje brojeva koji se nalaze na memorijskim lokacijama M[100] i M[200], a rezultat se smešta u memorijsku lokaciju M[300].



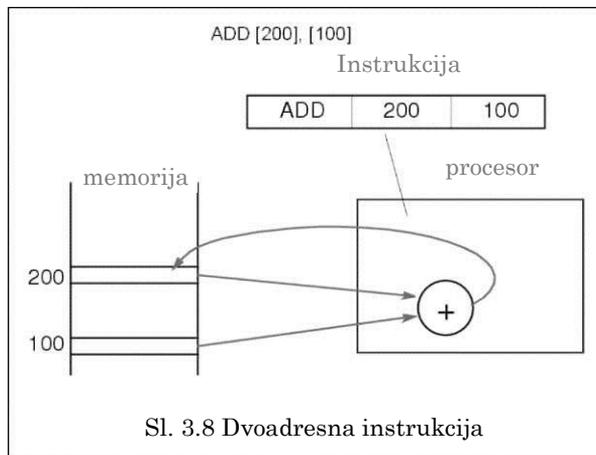
Sl. 3.7: Troadresna instrukcija

U dvoadresnom formatu (instrukcija F2 na Slici 3.6) jedno adresno polje adresira i izvršni i odredišni operand. Dejstvo ovakve instrukcije je

$$Op1 \leftarrow Op1 \otimes Op1$$

U toku izvršenja operacije, $Op1$ je izvorišni operand. Po formiranju, rezultat operacije se upisuje na mesto izvorišnog operanda $Op1$, pri čemu ga poništava. Alternativa je da odredišni operand zamenjuje drugi izvorišni operand.

Primer 3.2 Na slici 3.8 je prikazano dejstvo dvoadresne instrukcije za sabiranje brojeva koji se nalaze na memorijskim lokacijama M[100] i M[200], a rezultat se smešta u memorijsku lokaciju M[200].

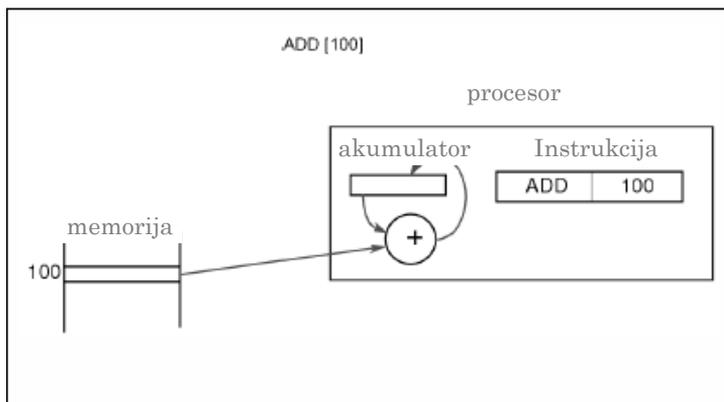


U jednoadresnom formatu instrukcija (F1 na Slici 3.6) adresa jednog izvorišnog operanda i rezultata se podrazumeva, implicitno je određena. Obično je implicitno imenovan akumulator (*ACC*), pa je dejstvo instrukcije

$$ACC \leftarrow ACC \otimes Op$$

Instrukcije ovog oblika se sreću kod računara sa akumulatorskom arihtekturom procesora.

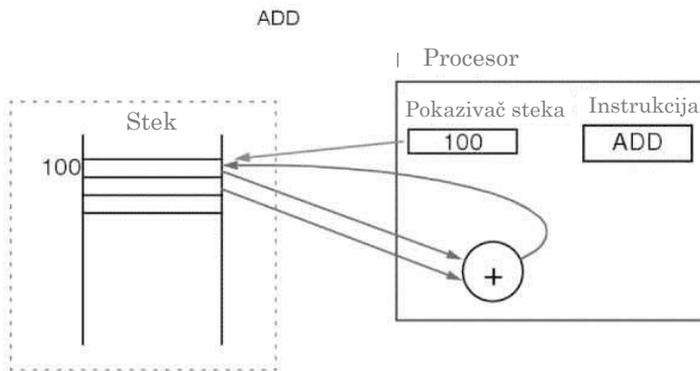
Primer 3.3 Na slici 3.9 je prikazano dejstvo jednoadresne instrukcije za sabiranje broja koji se nalazi na memorijskim lokacijama $M[100]$ i broja koji se nalazi u registru akumulatora, a rezultat se smešta u registar akumulatora.



Sl. 3.9: Jednoadresna instrukcija

U nulaadresnom formatu instrukcija (F0 na Slici 3.6) ne navodi se eksplicitno nijedan operand. Format instrukcije sadrži samo kod operacije instrukcije. Instrukcije ovog formata tipične su za stek arhitekturu procesora.

Primer 3.4 Na slici 3.10 je prikazano dejsvo nulaadesne instrukcije za sabiranje broja koji se nalazi na vrhu steka. Stek može biti organizovan kao deo operativne memorije, ili kao skup registara unutar procesora. Pokazivač steka je registar unutar procesora



Sl. 3.10: Nulaadresna instrukcija

koji pokazuje na vrh steka.

Instrukcije sa najvišom adresnošću određuju i adresnost računara. Za računar koji ima troadresne instrukcije kažemo da je troadresni računar. Naravno, to ne znači da su sve njegove instrukcije troadresne; u svom skupu instrukcija on može da ima i dvoadresne, jednoadresne pa čak i nulaadresne instrukcije. Različiti formati instrukcija izraz su težnje projektanata računara da postignu što veću gustinu instrukcija, odnosno da zauzmu što manje bitova u memoriji računara.

3.5.2 Arhitektura procesora

Arhitektura ili strukturna organizacija odražava sastav i uzajamne veze elemenata sistema. Struktura računarskog sistema predstavlja model računarskog sistema na osnovu strukturnih

(konstrukcionih) komponenata (elemenata), a odražava uzajamne veze elemenata u procesu funkcionisanja sistema.

Prema mestu privremenog čuvanja operanda, arhitekture procesora se dele na

- akumulatorsku arhitekturu,
- stek arhitekturu i
- arhitekturu sa registrima opšte namene.

Akumulatorska arhitektura tipična je za procesore u početnoj fazi razvoja računara. Odlikuje se postojanjem u procesoru jednog registra nazvanog akumulator, koji učestvuje u svim operacijama procesora. Neposredno pre izvršenja operacije akumulator sadrži prvi operand, a po izvešenju operacije sadrži rezultat. Tipična instrukcija ovakvog procesora je i njeno dejstvo je

$$ADD \text{ } adr \quad ACC \text{ --- } ACC + M[adr]$$

Procesori sa akumulatorskom strukturom spadaju u jednonoadresne procesore. U insrukcijama je neophodno navesti samo adresu drugog operanda u memoriji, dok je adresa prvog operanda i rezultata unapred poznata - to je akumulator. U ovakvim instrukcijama akumulator je implicitno adresiran. Prvi opreand se iz memorije prenosi u akumulator instrukcijom Load (LD)

$$LD \text{ } adr \quad ACC \text{ --- } M[adr],$$

a rezultat operacije se iz akumulatora šalje u memoriju instrukcijom Store (ST)

$$ST \text{ } adr \quad M[adr] \text{ --- } ACC$$

Osnovni nedostatak akumulatorske arhitekture procesora je postojanje samo jednog registra-akumulatora, koji učestvuje u obavljanju svih operacija procesora. Ovo uslovljava povećanje obima podataka koji se prenose između procesora i memorije

Stek arhitektura procesora kreirana je sa ciljem da smanji jaz u obimu aktivnosti iskazanih instrukcijama procesora i naredbama viših programskih jezika. Centralna memorijska struktura ove arhitekture je stek (engl. stak). Operandi koji učestvuju u aritmetičko-logičkim operacijama nalaze se na vrhu steka, tako

da nije potrebno eksplicitno ih adresirati. Operacijom se gube oba izvorišna operanda sa vrha steka, a rezultat postaje novi vrh steka. Tipična instrukcija za obavljanje aritmetičko logičkih operacija (napr. *ADD*) ima sledeće dejstvo

$$ADD \quad S[SP + 1] \leftarrow S[SP + 1] + S[SP], SP \leftarrow SP + 1$$

Ovde *S* označava stek organizovan u memoriji, koji raste u smeru opadanja memorijskih adresa. *SP* je registar pokazivač steka (engl. Stack Pointer) koji pokazuje na vrh steka. Može se zapaziti da se u instrukciji *ADD* eksplicitno ne navodi ni jedan operand. Zato se za procesor sa stek arhitekturom kaže da je nula adresni procesor. Drugi operand se nalazi na vrhu steka, a drugi operand ispod vrha steka. Po izračunavanju rezultata on zamenjuje prvi operand u steku. Implementacija steka često za dve lokacije sa vrha steka umesto memorijskih lokacija koristi registre procesora. Podatak se iz memorijske lokacije *M[adr]* prenosi instrukcijom *PUSH adr*, a iz stek prenosi u memorijsku lokaciju *M[adr]* instrukcijom *POP adr*. Dejstvo ovih insrukcija su

$$\begin{array}{ll} PUSH \text{ } adr: & POP \text{ } adr: \\ SP \leftarrow SP - 1 & M[adr] \leftarrow SSP \\ S[SP] \leftarrow M[adr] & SP \leftarrow SP + 1 \end{array}$$

Umesto u memorijskoj lokaciji *M[adr]*, operand u instrukcijama *PUSH* i *POP* može biti i u nekom registru. To je češći slučaj u arhitekturi procesora sa registrima opšte namene.

Veliki nedostatk stek arhitekture procesora su ograničenja u pristupima podacima u steku.

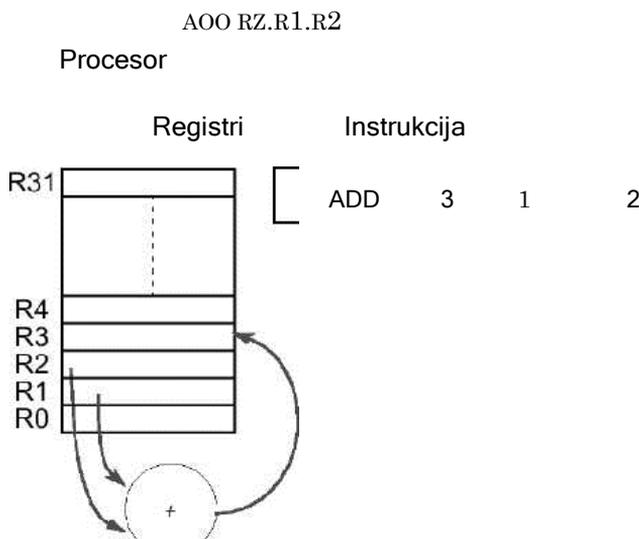
Arhitektura sa registrima opšte namene sarže grupu registara procesora koji se mogu koristiti u različite svrhe: za privermeno čuvanje operanda, za učešće u izvršavanju operacija, za adresiranje opearnda u memoriji, za prenos parametara pri pozivu procedura itd. Broj registara opšte namene u različitim procesorima koji se kreće u granicama od 8 do prko stotinu. Savremeni 32-bitni računari - 64-bitni procesori najčešće sadrže po 32 registara opšte namene. Ove arhitekture otklanjaju osnovni nedostatak akumulatorske arhitekture - jedinstveni akumulator kao usko grlo arhitekture procesora.

Prema mogućnostima kombinovanja korišćenja registara i memorija za čuvanje operanda instrukcija, arhitektura sa registrima opšte namene dele se na:

- registarsko-registarsku arhitekturu,
- registarsko-memorisku arhitektura i
- registarsku i memorijsku arhitekturu.

Registarsko-registarska arhitektura zahteva da svi operandi koji učestvuju u operacijama budu prisutni u registrima procesora. Instrukcije su troadresne. Procesor pristupa operandima u memoriji samo preko instrukcija Boas 3loga, kojima puni registre operandima i pamti rezultate u memoriji računara. Ova arhitektura je zastupljena u K18S procesorima.

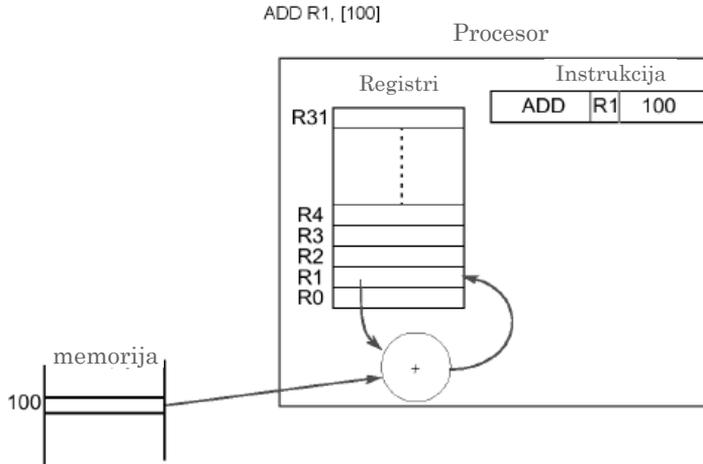
Primer 3.5 Na slici 3.11 prikazan je primer registarsko-registarske arhitekture procesora na primeru sabiranja brojeva koji se nalaze u registrima K1 i K2, a rezultat se smešta u registar K3.



Sl. 3.11: Registarsko-registarska arhitektura

Registarsko-memorijska arhitektura dopušta da jedan od dva operanda bude u memoriji. Instrukcije su dvoadresne, pri čemu rezultat zamenjuje jedan operand u registru. Mogućnost uzimanja operanda direktno iz memorije povoljna je kod jednokratnog korišćenja operanda.

Primer 3.6 Na slici 3.12 prikazana je registarsko-memorijske arhitektura procesora na primeru sabiranja brojeva koji se nalaze u registru R1 i memorijskoj lokaciji sa adresom 100, a rezultat se smešta u registar R1.



Sl. 3.12: Registarsko-memorijska arhitektura

Registarsko i memorijska arhitektura dopušta da operandi budu u registrima ili u memoriji. Moguće su sve kombinacije, počev od svih operanada u registrima do svih operanada u memoriji, sa svim međukombinacijama. Ovo je najopštiji tip arhitekture sa registrima opšte namene.

Primer 3.7 Za navedene tipove arhitekture napisati programe za izračunavanje vrednosti izraza

$$D = A + B \times S.$$

Za posanje programa za stek arhitekturom pogodno je dati izraz transformisati iz standardne matematičke notacije u takozvanu inverznu poljsku notaciju, odnosno u obliku zapisa u kome se operator navodi za oba operanda na koja se on primenjuje. Prelazak iz standardne matematičke notacije u inverznu poljsku notaciju može se obaviti postupno zamenjujući trojku *operand1_operator_operand2* transformisanom trojkom *operand1_operand2_operator*. Primenom na naš izraz to daje

$$D = A + B \times S \Rightarrow A + \{BC \times\} \Rightarrow ABC \times +$$

Na osnovu izraza u ovom obliku pisanje programa za stek organizaciju procesora je jednostavno: Kada se pri prolazu kroz izraz sa levo u desno naiđe na operand, u programu se unosi instrukcija *PUSH*, a kada se naiđe na operator, u program se unosi instrukcija koja realizuje taj operator. Instrukcijom *POP* rezultat iz steka se šalje u memoriju.

Programi za svih pet tipova arhitekura navedeni su u tabeli

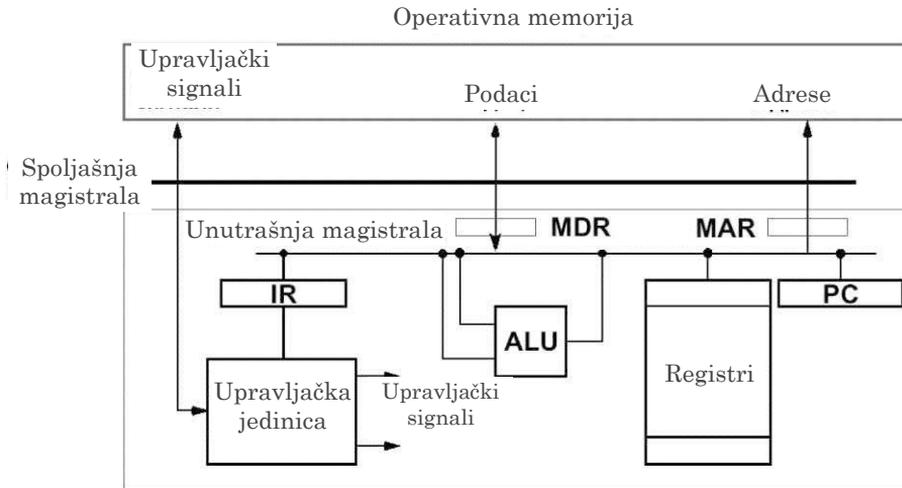
Tabela 3.1: Programi za izračunavanje izraza $B = A + B$

Stek Arhitektura	Akumulator arhitektura	Reg.-mem. arhitektura	Reg.-reg. arhitektura	Reg.+mem. arhitektura
<i>PUSH A</i>	<i>LD B</i>	<i>LD R1, B</i>	<i>LD R1, B</i>	<i>MOV D, B</i>
<i>PUSH B</i>	<i>MUL C</i>	<i>MUL R1, C</i>	<i>LD R2, C</i>	<i>MUL D, C</i>
<i>PUSH C</i>	<i>ADD A</i>	<i>ADD R1, A</i>	<i>MUL R1, R2</i>	<i>ADD D, A</i>
<i>MUL</i>	<i>ST D</i>	<i>ST D, R1</i>	<i>LD R2, A</i>	
<i>ADD</i>			<i>ADD R1, R2</i>	
<i>POP</i>			<i>ST D, R1</i>	
<i>D</i>				

Poredeći programe po broju instrukcija, vidimo da je registarsko-memorijska arhitektura najefikasnija. Ovo proističe iz fleksibilnosti ove arhitekture, koja omogućuje da se aktivnosti obuhvaćene programom iskažu najmanjim brojem instrukcija. Za arhitekturu koja sa malo instrukcija kodira program kažemo da ima veliku gustinu instrukcija. Programi za registarsko-registarsku i stek arhitekturu su najduži.

Uprošćena strukturna šema procesora prikazana je na slici 3.13. Na šemi su prikazani samo sledeće najvažnije komponente:

- upravljačka (komandna) jedinica,
- aritmetičko-logička jedinica (ALU),
- registar instrukcija (IR),
- adresni registar memorije (MAR),
- prihvatni registar memorije (MDR),
- brojač instrukcija (PC),
- skup registara opšte namene,
- spoljašnja i unutrašnje magistrale.



Sl. 3.13: Strukturna šema procesora

Sve komponente procesora povezuju se magistralama. Magistrala ili sabirnica predstavlja grupu provodnika (linija) kroz koje se signalima prenose podaci između registara ili pojedinih funkcionalnih celina u računaru. Svakom magistralom mogu se prenositi tri grupe signala: podaci, adrese i upravljački signali. Magistrale koje se koriste unutar procesora nazivaju se unutrašnje magistrale, magistrale koje se koriste van procesora nazivaju se spoljne magistrale. Unutrašnje magistrale povezuju sve komponente procesora u jednu celinu.

Upravljačka jedinica služi za prepoznavanje instrukcija (mašinskih naredbi) koje treba izvršiti, kao i za formiranje upravljačkih signala za izvršenje tih instrukcija.

Upravljački registri služe za privremeno čuvanje informacija koje se koriste za upravljanje radom procesora. Sadrži registre i brojače koji učestvuju u upravljanju procesom obrade: registre koji čuvaju podatke o stanju procesora, brojač instrukcija (programski brojač), registar instrukcije, registre za određene metode adresiranja i dr. Registri opšte namene imaju višestruku ulogu. Mogu se koristiti kao upravljački registri, ali je njihova osnovna namena privremeno čuvanje podataka i me

đurezultata pri izvršenju operacija.

Procesor obično koristi i skup programski adresivih registara opšte namene. Njihova se namena može definisati u programu. Ovi se registri često tretiraju kao adresiva lokalna registarska memorija koja je brža od osnovne (operativne) memorije. Interfejs (blok za spregu) sadrži sprežna kola kojima se organizuje razmena podataka između procesora i operativne memorije, kao i veza procesora sa perifernim uređajima, drugim računarima i sl. Važnu ulogu pri izvršenju operacija ima raspored upravljačkih signala u vremenu. Zato se govori o taktovanju ili sinhronizaciji rada procesora.

Osim navedenih, procesor može sadržati i druge komponente (funkcionalne blokove) kao što su: kesh-memorija, stek-memorija i niz blokova namenjenih organizaciji procesa obrade (blok prekida programa, blok zaštite memorije, blok za kontrolu ispravnosti rada i dijagnostiku procesora i dr.).

U procesor se dovode signali spoljnih zahteva kojima se od procesora traži neka usluga, preduzimanje akcije i sl. Ovi signali dolaze od blokova samog procesora, od drugih jedinica računara i iz spoljašnjeg okruženja. Izlazni signali procesora predstavljaju sinhronizacione (taktne) signale, razne upravljačke signale, signale koji saopštavaju stanje procesora i dr.

3.5.3 Registri procesora

Registar je elektronsko kolo koje služi za pamćenje binarnih podataka. Sastoji se od grupe bistabilnih kola koja se tretira kao celina. Registri procesora identifikuju se svojim adresama ili simboličkim imenima. U procesoru se mogu izdvojiti dve grupe registra:

- upravljački registri (radni registri) i
- registri opšte namene (lokalni registri).

Upravljački registri procesora. - Većina savremenih procesora konstruisana je tako da poseduje blok posebnih namenskih registara koji imaju unapred definisane upravljačke funkcije. Ovi se registri nazivaju upravljački ili radni registri procesora.

Blok upravljačkih registara namenjen je privremenom čuvanju informacija za upravljanje. Sadrži registre i brojače koji učestvuju u upravljanju procesom obrade:

- Adresni registar memorije **MAR** (engl. Memory Address Register, sadrži adresu memorijske lokacije kojoj se obraća procesor pri čitanju podataka iz memorije i upisu podataka u memoriju).
- Prihvatni registar memorije **MDR** (engl. Memory Data register), pri obraćanju procesora memoriji sadrži podatak pročitan iz memorije ili podatak koji treba upisati u memoriju. Prilikom upisa u memoriju predhodni sadržaj se briše, dok se pri čitanju vrši kopiranje sadržaja lokacije, tj. sadržaj lokacije se upisuje u prihvatni registar i zadržava se istovremeno u samoj lokaciji.

Neka je ad adresa memorijske lokacije čiji sadržaj treba pročitati. Proces čitanja tog sadržaja se u računaru odvija sledećim redosledom:

1. $(MAR) \leftarrow ad$ uzima se adresa ad i upisuje u adresni registar memorije.
2. Na osnovu sadržaja adresnog registra memorije nalazi se pomoću dekodera odgovarajuća adresa u memoriji.
3. $(MDR) \leftarrow M[ap]$. Čita se sadržaj izabrane memorijske lokacije i smesta u prihvatni registar memorije, pri tome se njegov sadržaj briše, a sadržaj izabrane lokacije iz koje je vršeno čitanje ne menja se.

Ako je R registar procesora čiji sadržaj treba upisati u memorijsku lokaciju sa adresom as postupak upisa u memoriju se odvija sledećim postupkom:

1. $(MAR) \leftarrow ad$. Uzima se adresa as i upisuje u adresni registar memorije.
 2. $(MDR) \leftarrow (R)$. U prihvatni registar memorije se upisuje sadržaj registra K .
 3. Na osnovu sadržaja adresnog registra nalazi se odgovarajuća lokacija u memoriji.
 4. $M[ad] \leftarrow (R)$. Upisuje se sadržaj prihvatnog registra u memorijsku lokaciju sa adresom ad . Pri tome se predhodni sadržaj memorije briše.
- Brojač instrukcija (programski brojač **PC**), sadrži memorijsku adresu instrukcije koju treba pribaviti.

- Registar instrukcija IR (engl. Instruction Registar), u koji se prenosi pribavljena instrukcija i u njemu drži do zavšetka izvršenja instrukcije. Pozivanje tekuće naredbe iz operativne memorije i njeno smeštanje u registar naredbi odvija se na sledeći način:
 1. $(AR) \leftarrow (PC)$. Adresa sledeće naredbe iz brojača instrukcija šalje se u adresni registar.
 2. Na osnovu sadržaja adresnog registra selektuje se lokacija u memoriji u kojoj se nalazi naredba.
 3. $(MDR) \leftarrow M[MAR]$. Čita se sadržaj izabrane lokacije i upisuje se u prihvatni registar memorije.
 4. $(IR) \leftarrow (MDR)$. Sadržaj prihvatnog registra memorije šalje se u registar instrukcija.
- Akumulator je poseban registar u kome se sve izvršavaju sve aritmetičke i logičke operacije, operacije pomeranja i mnoge druge mašinske operacije pri izvršenju neke operacije u njemu se uvek nalazi jedan od operanda, i u njega se uvek smešta rezultat dobijen po izvršenju te operacije. Pri tome se stari sadržaj akumulatora briše. Proces izvršenja mašinske operacije *op* kod jednoadresnih računara odvija se na sledeći način:
 1. Prvi operand se već nalazi u akumulatoru gde je smešten nekom ranije izvršenom naredbom.
 2. $(MBK) \leftarrow M[A1]$. Čita se drugi operand iz memorijske lokacije sa adresom A1 (koja je ranije smeštena u adresni registar) i smešta se u prihvatni registar.
 3. $(AK) \leftarrow (AK)op(MDR)$. Izvršava se operacija *op* koja je definisana sadržajem registra naredbi i dobijeni rezultat se smešta u akumulator.

Registri opšte namene. - Procesor obično koristi i skup programski adresivih registara koji mogu imati više različitih namena, zbog čega se nazivaju registri opšte namene ili lokalni registri. Njihov broj u procesorima se kreće od 8 do 64. Savremeni procesori imaju najčešće 32 registra opšte namene. Ovi registri su grupisani u polje registara sa zajedničkim ulaznim i izlaznim priključcima i pridruženom logikom, takozvanim portovima. Njihovo korišćenje može se definisati programskim instrukcijama i nazivaju se korisnički vidljivi registri. Ovi reg-

istri se često tretiraju kao adresiva lokalna registarska memorija brža od operativne memorije.

Registri opšte namene mogu se koristiti za privremeno čuvanje podataka ili međurezultata nad kojima procesor izvršava operacije. Kako je njihova brzina identična brzini rada procesora, operacije u procesoru se vrlo brzo izvršavaju.

3.5.4 Instrukcije

Računar je konstruisan tako da može direktno u procesoru izvršavati samo određeni broj prostih operacija koje se nazivaju instrukcije ili mašinske operacije. Operacija koja se izvršava odeđuje se kodom operacije. Kod operacije *KOp* mora da ima dovoljno različitih kodnih slogova da jednoznačno identifikuje svaku instrukciju iz skupa instrukcija procesora. Svaka instrukcija sadrži, pored koda operacije *KOp*, i modifikator adresa *M*, koji sadrži naznaku adresiranja svih operanada u instrukciji

Pored aritmetičkih i logičkih operacija, tipične operacije koje računar izvršava jesu prenos podataka iz jednog dela računara u drugi (npr. prenos iz centralnog procesora u operativnu memoriju ili obratno), učitavanje podataka sa ulaza ili slanje podataka na izlaz i dr.

Na primer, posmatrajmo računar u kome se koriste troadresne instrukcije i koji može da realizuje 30 različitih instrukcija, a čija operativna memorija ima kapacitet od 1kiloreči. Predpostavimo da treba da se izvrši instrukcija čiji je kod operacije u decimalnom prikazu 25, operandi se nalaze u memorijskim lokacijama čije su adrese u decimalnom prikazu 301 i 302, a rezultat treba smestiti u memorijsku lokaciju čija je adresa u decimalnom prikazu 1022. Pošto računar razlikuje 30 instrukcija to je dužina polja koda operacija 5 bitova. Da bi se mogla adresirati svaka memorijska lokacija od 1024 koliko ih sadrži operativna memorija, to adrese moraju biti dugačke 10 bitova. Prema tome troadresna instrukcija ima oblik

11001 | 0100101101 | 0100101111 | 1111111110

a dužina posmatrane instrukcije je 35 bitova.

Instrukcija predstavlja kod koji određuje operaciju računara i podatke koji učestvuju u toj operaciji, tj. sadrži informacije

potrebne za upravljanje izvršenjem jedne operacije. Izvršenje instrukcije može se podeliti na manje, jednostavnije etape koje se nazivaju mikroradnje ili mikrooperacije. Njima se u procesoru izvršavaju u jednom taktom intervalu određene elementarne radnje. Tipičan primer elementarne radnje koja se izvršava u procesoru predstavlja prenos sadržaja iz jednog registra procesora u drugi. Skup mikroinstrukcija koje se izvršavaju u jednom taktom intervalu procesora naziva se instrukcija. Sekvenca mikrooperacija kojom se realizuje operacija (tj. instrukcija) obrazuje mikroprogram operacije. Na taj način nastaje sledeća hijerarhija etapa izvršavanja programa u procesoru: program, instrukcija (mikroprogram), mikroinstrukcija.

Skup instrukcija

Sve instrukcije kojima su realizovane razne operacije u procesoru uslovno se mogu podeliti na nekoliko grupa. Najčešća je sledeća podela instrukcija:

1. instrukcije za prenos podataka,
2. aritmetičko-logičke instrukcije (instrukcije obrade),
3. instrukcije za upravljanje tokom izvršenja programa (upravljačke instrukcije),
4. ulazno-izlazne instrukcije i
5. ostale instrukcije.

Instrukcijama za prenos podataka poziva se sadržaj lokacija operativne memorije u akumulator ili neki drugi registar procesora, šalje sadržaj akumulatora ili drugog registra procesora u operativnu memoriju, prenosi ili premesta sadržaj jednog registra u drugi i sl. U sledećoj tabeli su date instrukcije za kopiranje sadržaja jedne lokacije u drugu lokaciju

Instrukcija	;	Komentar
MOV R1,R2	;	K1=K2
LD R3,[100]	;	R3=sadržaj memorijske lokacije čija je adresa 100.
ST [200],R4	;	Za 32 bitne registre to su 4 uzastopne memorijske lokacije, 100, 101, 102 i 103.
		Memorijska lokacija čija je adresa 200=K4, Za 32 bitne registre, to su 4 uzastopne memorijske lokacije, 200,201,202 i 203.

Instrukcije za prenos podataka nose naziv MOVE, LOAD, STORE, PUSH, POP, IN, OUT itd. Instrukcijom MOVE podatak se može preneti između para registara ili između registra i memorije. Instrukcijom LOAD podatak se prenosi iz memorije u registar a instrukcijom STORE iz registra u memoriju. PUSH i POP su instrukcije za razmenu podataka između registra procesora i steka. Instrukcijom IN prenosi se podatak iz ulaznog porta računara u registar procesora. Instrukcijom Out. prenosi se podatak iz registra procesora u izlazni port računara. Ulazni i izlazni portovi su delovi ulazno izlaznog sistema računara.

Aritmetičkim instrukcijama izvršavaju se osnovne aritmetičke operacije (sabiranje, oduzimanje, množenje i deljenje), zatim poređenje dva sadržaja, komplementiranje, povećanje ili smanjenje sadržaja za 1 i sl. Logičkim instrukcijama (And, Or, XOR itd.) izvršavaju se logičke operacije (operacije Bulove ili prekidačke algebre). Na primer, instrukcija

SUB R5, R4, 3

označava da od sadržaja registra R4 treba oduzeti 3 i rezultat smestiti u registar R5. U zavisnosti od asemblerske notacije ista instrukcija se može napisati i na sledeći način

SUB R5, R4, #3.

Instrukcijama za pomeranje pomera se binarni sadržaj nekog registra procesora, najčešće akumulatora, za određeni broj mesta levo ili desno. Pomeranje može biti "logičko" SHL i SHR tako što se prazno mesto, koje nastaje posle pomeranja, popunjava nulom. Kod "aritmetičkog" pomeranja SAL i SAR se svodi na množenje odnosno deljenje sa dva. Na primer, instrukcija za logičko pomeranje sadržaja registra R1 za jedno mesto u levo je

SHL R1, R1, 1

Instrukcije za upravljanje tokom programa sadrže instrukcije bezuslovnog i uslovnog grananja, skokove, pozive procedura (potprograma) i povratka iz procedura (potprograma). U ovim instrukcijama za odredište grananja odnosno skoka koristi se adresiranje relativno u odnosu na programski brojač. Otklon je označen ceo broj. Sve instrukcije (osim JR i JALR) su dužine 4

bajta, tako da je adresa instrukcije, koja u programu sledi upravljačku instrukciju, $PC + 4$. Otklon sabira se sa $PC + 4$, odeđujući određište skoka odnosno grananja, respektivno. Na primer, in-strukcija za skok piše se *JR ime*, a znači $PC \leftarrow PC + 4 + ime$.

Ulazno-izlazni prenos se upravlja prekidom koji se odvija na sledeći način. Procesor prenosi upravljačku informaciju kontroleru perifernog uređaja a zatim se od njega isključuje i nastavlja paralelno da izvršava neki drugi program. Kada periferni uređaj završi zahtevanu operaciju on šalje procesoru zahtev za prekid da bi saopštio o izvršenju predviđene radnje. Procesor prihvata prekid i registruje stanje perifernog uređaja.

Ciklus izvršenja instrukcije

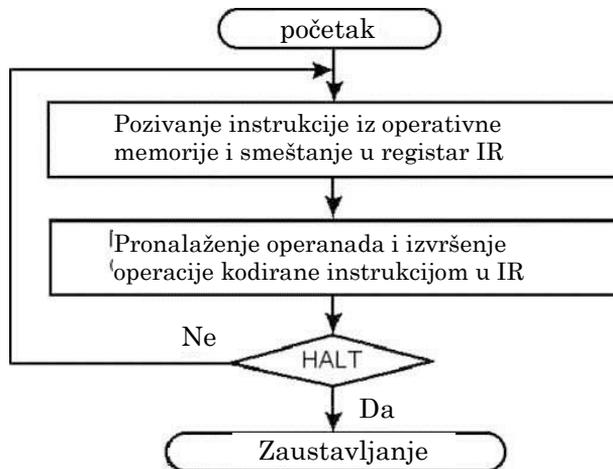
Pod pojmom izvršenje instrukcije podrazumeva se proces izvršenja operacije zadate tom instrukcijom. Instrukcija koju treba izvršiti smeštena je u operativnoj memoriji. Da bi ona mogla upravljati izvršenjem operacije, potrebno je da se ona uzme iz operativne memorije i smesti u registar instrukcija procesora.

Funkcionisanje svakog procesora uglavnom se sastoji od cikličnog ponavljanja izvršavanja instrukcija neprekidnim smenjivanjem sledece dve faze:

1. faze pripreme ili pozivanja instrukcije i
2. faze izvršenja operacije (instrukcije).

Faza pripreme prethodi fazi izvršenja i one se ciklično menjaju prema dijagramu prikazanom na Slici 3.14. U bloku HALT proverava se zaustavljanje izvršenja programa. U fazi pripreme učitava se instrukcija iz memorije, tj. dovodi se u sam procesor, a u fazi izvršenja obavlja se operacija određena pozvanom instrukcijom.

U opštem slučaju pod terminom naredba podrazumeva se izraz koji specificira operaciju i vrednosti ili mesta operanada nad kojima se ta operacija treba da obavi. Ako se radi o mašinskim operacijama tj. operacijama koje računar može raspoznati i direktno izvršiti koristi se termin instrukcija ili mašinska naredba.



Sl. 3.14: Ciklus izvršenja instrukcije

U jednom intervalu vremena računar može izvršiti samo jednu operaciju specificiranu odgovarajućom instrukcijom. Da bi se rešio neki zadatak mora se izvršiti celi niz operacija. Pod terminom izvršenje instrukcije podrazumeva se proces izvršenja operacije zadate tom instrukcijom.

Instrukcija (mašinska naredba) sastoji se iz više delova. Njena struktura zavisi od računara ali obično ima dva dela:

1. kod operacije, kojim se označava operacija koju treba izvršiti,
2. adresni deo, u kome se specificira način kako i odakle treba uzeti operande i gde smestiti rezultat.

Primer 3.8 . - Instrukcija

$$ADD\ rd,rs,rt \quad R|rd| \leftarrow R|rs| + R|rt|$$

ima sledeću značenje: vrednost specificiranu operandom $R|rs|$ sabrati sa vrednošću specificiranu operandom $R|rt|$ i rezultat smestiti na mesto specificirano operandom $R|rs|$. Sa $R|rd|$, $R|rs|$ i $R|rt|$ označeni su registri opšte namene sa adresama rd , rs i rt .

Instrukcijom

$$OR\ rd,rs,rt \quad R|rd| \leftarrow R|rs| \vee R|rt|$$

se izvršava logička operacija ili.

Plan kojim se zadaju dejstva koja treba izvršiti naziva se op-

štim imenom program. Program za računar, ili računarski program, predstavlja program izrađen u obliku pogodnom za izvršenje na računaru.

Rešavanje nekog problema pomoću računara sastoji se u primeni niza operacija koje se izvršavaju po predviđenom redosledu. Obično svakoj operaciji odgovara instrukcija programa. Program za računar, koji ćemo dalje nazivati masinski program (mašinski kod) ili kratko program, predstavlja niz (sekvencu) instrukcija koje saopštavaju računaru, korak po korak, kako da reši neki problem, uključujući i donošenje odluka o toku izvršenja samog programa.

Da bi neki zadatak mogao da bude rešen na računaru potrebno je da za njega postoji program. Postupak pripreme programa, koji se sastoji od projektovanja (sastavljanja), pisanja i testiranja programa naziva se programiranje.

Proces programiranja sastoji se od više faza među kojima su osnovne sledeće tri:

1. projektovanje programa - određivanje sekvence operacija neophodnih za rešenje problema,
2. pisanje programa - predstavljanje programa na način kojim se detaljno zadaju operacije koje računar treba da izvrši i provera pravilnosti rada programa,
3. testiranje programa - provera ispravnosti rada programa.

Umesto realizacije ciklusa izvršenja instrukcije u samo dve faze, savremeni procesori instrukciju izvršavaju obično u četiri osnovnih koraka sa vremenskim preklapanjem njihovog izvršenja u pet posebnih procesnih jedinica koje istovremeno obrađuju pojedine delove instrukcija (tzv. protočna obrada instrukcija). Protočna obrada se se u savremenim procesorima koristi i kod realizacije aritmetičkih operacija.

3.5.5 Izvršenje programa

U radu računara mogu se izdvojiti sledeće etape:

- unošenje i prevođenje programa,
- unošenje (ulaz) podataka,

- izvršenje programa.

Unošenje i prevodenje programa. Pripremljen (napisan) program unosi se u osnovnu memoriju pomoću editora (editora programa ili editora teksta). Uneti program se zatim pozivanjem programskog prevodioca prevodi u mašinski program u obliku tzv. objektnog modula koji se smešta na spoljnu memoriju kao datoteka. Danas se obično za rešenje nekog zadatka koristi skup programa i potprograma koji čine programski paket. Objedinjavanjem objektnih modula svih tih programa i potprograma postupkom koji se naziva povezivanje, pomoću posebnog sistemskog programa - povezivača, dobija se izvršni kod.

Unošenje podataka. - Preko ulazne jedinice podaci za obradu se ručno ili automacki prenose u osnovnu memoriju a zatim se skladište na spoljnoj memoriji u datoteke. Pri izvršenju programa ovi će podaci biti ponovo preneti u osnovnu memoriju, obrađeni saglasno aktivnom programu i ponovo uskladišteni u datoteke na disku ili direktno izdati korisniku preko neke izlazne jedinice.

Izvršenje programa. Program se izvršava na sledeći način: na osnovu komande operativnom sistemu računara unete sa tastature izvršni kod programa se ceo ili njegov deo prenosi u osnovnu memoriju i startuje se njegovo izvršenje. Podatke za obradu program uzima ili sa spoljne memorije, ili unete preko neke ulazne jedinice. Kod tzv. interaktivnih programa, a to su programi koji se izvršavaju u režimu dijaloga korisnika i računara, korisnik preko nekog ulaznog uređaja saopštava programu podatke za obradu ili komande šta treba da radi sa podacima (npr. kako da formira sliku, ili šta da radi sa slikom koja se obrauje).

3.6 Konfiguracija personalnih računara

Personalni računar je konstruisan tako da na njemu može raditi samo jedna osoba. Sadrži sledece karakteristične osnovne module:

- osnovnu ploču,
- tastaturu,

- miša,
- ekran (monitor),
- grafičku karticu (video-adapter),
kontroler jedinica disketa i diskova,
jedinice disketa,
- jedinicu tvrdog diska,
- CD ROM jedinicu,
- izvor napajanja.

Pored ovih osnovnih modula na personalni računar se mogu priključiti i mnogi drugi uređaji. Za te namene unutar kućišta postoji određen broj praznih utičnica ili slotova (engl. slots) u koje se umeću tzv. kartice/ploče elektronike koje omogućavaju povezivanje na magistralu personalnog računara drugih uređaja. U praksi se najčešće priključuju sledeći uređaji: štampač, skener, crtač (ploter), komandna palica (palica za igru), grafička tabla (digitalizator), zvučna kartica i dr. Pomoću modema personalni računar preko telekomunikacionih linija ima mogućnost pristupa računarskim mrežama.

3.6.1 Osnovna ploča

Na osnovnij (matičnoj) ploči personalnog računara nalaze se procesor, memorija, magistrala, skup kola koja kontrolišu rad računara i priključci za kontrolere.

Procesor definiše tip personalnog računara. Danas su aktuelni procesori firme INTEL i AMD, dok se raniji tipovi procesora (od 8086 do 80386) više ne koriste. Brzinu rada procesora određuje frekvencija radnog takta. Procesori 80486 su radili na frekvencijama 66, 100 i 120 MHz, a Pentium procesori na 100, 120, 133, 150, 200, 300 MN i više.

Memorija personalnog računara sastoji se RAM, ROM i keš memorija. RAM memorija predstavlja najveći deo memorije. Navedeni procesori zahtevaju najmanje 64 MB RAM memorije, ali je za normalan rad ipak potrebno od 128 MB, 512 MB i više. U ROM memoriju smešten je bios (Basic Input/output Control System - osnovni ulazno-ilazni upravljački sistem) koji pokreće personalni računar, a ima i još neke funkcije pomenute ranije. Keš memorija

je vrlo brza memorija u koju se iz operativne memorije prenose podaci koji se često pozivaju. Ima kapacitet od 128, 256 KB i više.

Priključci za kontrolere sastoje se od konektora na koje se priključuju kontroleri različitih jedinica personalnog računara predviđeni za određenu internu magistralu. Standardno se koristi ISA magistrala (engl. ISA – Internacional Standards Association), a pored nje još ili VESA (engl. Video Standards Association) ili PCI (engl. Peripherals Connect Interface) magistrala.

3.6.2 Ulazno-izlazna ploča

Za povezivanje personalnog računara sa spoljnim svetom koristi se obično posebna ulazno-izlazna ploča. Ona ima standardno dva serijska porta COM1, na koji se obično priključuje miš, COM2, za drugu serijsku vezu, na primer modem, i jedan paralelni port LPT1, na koji se obično priključuje štampač, kao i jedan port za priključak palice za igre. Za povezivanje računara sa perifernim uređajima na raspolaganju stoje USB priključci.

3.6.3 Kontroler diskova i disketa

Standardno kontroler diska isporučuje se na jednoj kartici (ploči elektronike). On može da upravlja radom najmanje dva diska i dve jedinice disketa. Kapacitet diskova stalno raste i kontroler diska je sastavni deo diska. Na tržištu se mogu naći čvrsti diskovi kapaciteta većeg od 120 GB

3.6.4 Video-adapter

Povezivanje računara sa monitorom, kao izlaznom jedinicom, ostvaruje se preko posebnog elektronskog sklopa - video-adaptera koji se još naziva grafička kartica. Video-adapter se u vidu specijalne ploče elektronike (video kartice) smešta u jednu od pozicija (slotova) za proširenje, ili se već nalazi na osnovnoj (glavnoj) ploči računara kao njen sastavni deo. Personalni računari najviše koriste SVGA adaptere (engl. SVGA – Super Video Graphic Array - super grafička video matrica). SVGA adapteri imaju najbolje karakteristike i u potpunosti su potisle ostale grafičke kartice.

Većina današnjih kartica omogućava priključivanje i drugog monitora. Ukoliko se želi snimanje TV programa, treba izabrati kartice sa integrisanim TV prijemnikom. Firma EVGA pravi grafičke kartice s TV prijemnikom zasnovanim na sklopu integrisanih kola Personal Cinema kompanije NVidia.

3.7 Pitanja za proveru znanja

1. Objasniti strukturu naredbe. Koji su elementi naredbe?
2. objasniti funkciju brojača naredbi.
3. Predpostavljajući da se koristi jednoadresni procesor, napisati instrukcije koje orenose sadržaj memorije na lokaciji 1000h u akumulator.
4. Identifikovati način adresiranja za sledeće hipotetične instrukcije:

- LD R2, [R3]
- LD R2, [100]
- LD R2, 100[R3],
- MOV R2, 6

5. Odrediti tip arhitekture i izraz koji se izračunava sledećim programima

(a)	(b)	(v)
<i>SUB</i> <i>Y, V</i>	<i>MOV</i> <i>Y, A</i>	<i>LOAD</i> <i>D</i>
<i>MUL</i> <i>T, V, E</i>	<i>SUB</i> <i>Y, B</i>	<i>MUL</i> <i>E</i>
<i>ADD</i> <i>T, T, C</i>	<i>MOV</i> <i>T, D</i>	<i>ADD</i> <i>C</i>
<i>DIV</i> <i>Y, Y, T</i>	<i>MUL</i> <i>T, E</i>	<i>STORE</i> <i>Y</i>
	<i>ADD</i> <i>T, C</i>	<i>LOAD</i> <i>A</i>
	<i>Y, T</i>	<i>SUB</i> <i>B</i>
		<i>DIV</i> <i>Y</i>
		<i>STORE</i> <i>Y</i>

6. Dat je izraz u inverznoj poljskoj notaciji

$$Y = AB - CDE \times +/.$$

Napisati isti izraz u standardnoj matematičkoj notaciji i program za izračunavanje tog izraza za računar sa nulaadresnim instrukcijama.

7. Pokazati da izraz

$$Y = \frac{AX + B}{CX - D}$$

ima sledeći oblik u inverznoj poljskoj notaciji

$$Y = AX \times B + CX \times D - /.$$

Napisati program za izračunavanje tog izraza za računar sa nulaadresnim instrukcija.

Glava 4

Memorijski sistem

Memorije koje se koriste u računarima sačinjavaju memoriski medijum, sposoban da ograničeno ili neograničeno dugo čuva prethodno unete podatke, i upravljački sistem memorije, koji omogućuje upisivanje i čitanje podataka iz memorije po nalogima procesora ili drugih elemenata računara.

Memorije je fizički ili logički izdvojena na blokove, čije su veličine određene dužinama osnovnih tipova podataka u računaru. Takve blokove memorije, kojima se pristupa odjednom, nose memorijske lokacije. One su karakteristične za glavne ili operativne memorije računara. Obično je dužina memorijskih lokacija jednaka dužini registra podatak u procesoru. Svaka memorijska lokacija ima jedinstvenu adresu, koja označava njeno mesto u okviru memorijskog medijuma.

Zavisno od tipa memorije mogu postojati ograničenja u pogledu načina i redosleda pristupa lokacijama u memoriji. Kada nema nikakvih ograničenja tog tipa, i kada su vremena pristupa lokacijama nezavisna od položaja lokacija u memoriji, imamo memorije sa proizvoljnim ili slučajnom pristupom (engl. Random Access Memory - ROM). Memorije čijim se lokacijama može pristupiti samo unapred određenim redosledom su memorije sa serijskim pristupom. Kod njih su vremena pristupa zavisna od položaja lokacija u nizu lokacija kojima se pristupa. Način rada kojim se direktno dolazi do adresiranog bloka memorije, a zatim se serijskim pristupom dolazi do određenog podatka u bloku, naziva se direktan pristup memoriji. Ovaj tip pristupa se koristi u

magnetnim i optičkim diskovima.

Čuvanje podataka prisutnih u memoriji može zahtevati stalno napajanje električnom energijom, inače se sadržaj memorije gubi. Takve memorije se nazivaju nestalne memorije. Postoje memorije koje posle upisa podataka u njih zadržavaju te podatke i po prekidu napajanja električnom energijom. Takve memorije se nazivaju stalne memorije. Ovde spadaju memorije sa magnetnim zapisom.

Važni kvantitativni parametri memorije su kapacitet memorije, vreme pristupa, vreme ciklusa memorije, propusnost memorije, snaga disipacije, specifična cena, pouzdanost, i drugi.

Kapacitet memorije se količinom podataka koje memorija može da prihvati i čuva. Izražava se u kilobajtima (kB), megabajtima (MB) ili gigabajtima (GB). Operativne memorije personalnih računara su kapaciteta do nekoliko stotina megabajta. Kapacitet krutih diskova je reda nekoliko desetina do nekoliko stotina gigabajta.

Vreme pristupa memoriji definiše se kao vremenski interval od prispeća zahteva za čitanje memorije do trenutka kada je taj podatak raspoloživ na izlazima memorije. Pri zahtevu za upis podatka u memoriju, kraj ovog intervala odeden je završetkom upisa. Kod memorija sa proizvoljnim pristupom vreme pristupa memoriji je konstantno.

Vreme ciklusa memorije definiše se kao najkraće vreme za koje memorija može uspešno da prihvati uzastopna obraćanja. Ovo vreme se navodi samo za memorije sa proizvoljnim pristupom.

Brzina sa kojom se podaci prenose u memoriju ili iz memorije naziva se propusnost memorije. Ona predstavlja odnos količine prenetih podataka i vremenskog intervala u kome se prenos obavlja. Izražava se u MB/s.

Snaga disipacije predstavlja snagu koju zahteva memorija za svoj rad. Snaga disipacije navodi se za operativni režim i za režim mirovanja. Ona se u memoriji pretvara u toplotu.

Specifična cena memorije izražava se odnosom cene memorije i njenog kapaciteta. Obično se iskazuje u dolarima po megabajtu ili u dolarima po gigabajtu.

Pouzdanost memorije meri se srednjim vremenom između otkaza.

Pouzdanost memorija koje ne sadrže pokretne delove je mnogo veća od pouzdanosti memorija sa pokretnim delovima, kao što su magnetni i optički diskovi. Povećanje gustine pakovanja komponenata i brzine prenosa podataka snižavaju pouzdanost memorijskih komponenata.

4.1 Memorijski medijumi

Za obradu podataka na računaru tj. za njihovo unošenje u računar ili izlaz iz računara, kao i za prenos na daljinu, podaci se pamte (registruju, skladište) na posebnim materijalnim nosiocima - (memorijskim) medijumima ili nosiocima podataka. Memorijski medijum predstavlja materijal, ili konfiguraciju načinjenu od njega, čija se neka promenljiva fizička veličina koristi za predstavljanje (registrovanje) podataka. Primeri medijuma su: magnetni diskovi, diskete, optički diskovi, papir koji se koristi za izlaz na štampaču itd.

U svim memorijskim sistemima koji se danas koriste na memorijskom medijumu se pamti (skladišti) energija u nekim diskretnim količinama kojima se predstavljaju binarne vrednosti "0" i "1", kao što su npr. magnetni moment, struja, naelektrisanje, provodni put i si. Da bi se neki medijum mogao koristiti za pamćenje podataka, treba da poseduje:

- a) dva stabilna energetska stanja odvojena visokom energetsom barijerom;
- b) mogućnost prelaska iz jednog u drugo stanje primenom spoljne energije neograničeni broj puta;
- v) mogućnost otkrivanja (ustanovljavanja) ta dva energetska stanja pomoću spoljnog izvora energije;
- g) gubitke energije pri upisu zbog pouzdanog pamćenja (skladištenja).

U tabeli 4.1 navedeni su najviše korišćeni fizički fenomeni koji zadovoljavaju gornja četiri zahteva a) - g) za pamćenje podataka.

Svi nosioci podataka se mogu svrstati u tri grupe: - nosioci ulaznih podataka - nosioci izlaznih podataka, - nosioci ulaznih

Tabela 4.1: Primeri fizičkih fenomena koji se koriste u memorijskim sistemima za skladištenje podataka (PET tranzistor se efektom polja, CCD element sa spregnutim naelektrisanjem)

Fizički fenomen	Vrsta memorijskog sistema
Magnetni moment	Magnetni disk
Količina elektriciteta	FET memorija, CCD elementi,
Električna struja	Bistabilna kola
Fizička struktura	Optički disk

i izlaznih podataka. Osim toga, postoji više načina i uređaja za tzv. neposredno unošenje podataka bez njihove prethodne pripreme na posebnim posrednim nosiocima.

4.1.1 Memorijski sistem računara

Memoriju ili memorijski sistem čine uređaji koji obezbeđuju zapisivanje (registrovanje) binarnih podataka nezavisno od njihovog sadržaja. Memorija je namenjena upisu (prijemu), pamćenju (skladištenju, čuvanju) i čitanju (izdavanju) podataka i programa. Operacije upisa i operacija čitanja nazivaju se još i pristup memoriji. Svaki memorijski sistem bilo kog tipa sadrži memorijski medijum na kome se pamte podaci, kao i upravljačke elektronske i programske blokove koji služe za traženje, upis ili čitanje podataka na memorijskom medijumu.

Memorija računara se sastoji od operativne memorije i spoljne memorije.

Operativna (osnovna, glavna) memorija je namenjena privremenom pamćenju podataka i instrukcija koji se neposredno koriste pri izvršenju programa, a sačinjena je od bistabilnih poluprovodničkih memorijskih elemenata. Pored podataka za obradu (ulazni podaci) i podataka koji definišu postupak obrade (instrukcije programa), u operativnoj memoriji se takođe pamte i međurezultati i konačni rezultati obrade.

Spoljna (masovna, svkundarna) memorija ima veliki kapacitet i služi za pamćenje podataka velikog obima i njihovo arhiviranje za duži vremenski period, a znatno je sporija od operativne memorije. Spoljna memorija služi za dugotrajno skladištenje gotovih programa i podataka, čuvanje podataka velikog obima, arhiviranje

podataka, čuvanje rezervnih kopija podataka idr.

4.1.2 Klasifikacija memorija

Memorije se klasifikuju po različitim kriterijumima. Jedan od važnih kriterijuma je fizičko svojstvo materijala koji se koristi za čuvanje (skladištenje, pamćenje) podataka. Koriste se sledeći fizički principi čuvanja podataka:

- elektronski,
- magnetni,
- optički i dr.

Elektronske memorije sadrže poluprovodnička bistabilna tranzistorska kola koja se mogu ulaznim upravljačkim signalima postaviti (prevesti) u dva različita stabilna stanja. Izlazna stanja zadržavaju se i posle završetka ulaznih upravljačkih signala.

Do sada najviše korišćeni fenomen za realizaciju memorijskih sistema je feromagnetizam. Feromagnetizam predstavlja svojstvo nekih materijala (tzv. magnetno tvrdih materijala) da zadrže namagnetisanost (magnetizaciju) u odsustvu spoljašnjeg polja.

Kod optičkih memorija upis i čitanje podataka vrši se pomoću poluprovodničkog lasera i optičkog sistema za generisanje vrlo uskog snopa svetlosti koji se fokusira na medijum radi pristupa svakom bitu podataka.

Prema načinu organizacije pristupa zapamćenim podacima, razlikuju se sledeće vrste memorija:

- 1) memorije sa neposrednim (proizvoljnim) pristupom,
- 2) memorije sa direktnim pristupom i
- 3) memorije sa sekvencijalnim pristupom.

Kod memorija sa neposrednim (proizvoljnim) pristupom (engl. RAM - Random Access Memory - memorija sa proizvoljnim pristupom) na osnovu adrese može se pristupiti bilo kojoj lokaciji. Kod ovih memorija vreme pristupa podacima ne zavisi od mesta u memoriji sa koga se čita ili u koje se vrši upis. Memorije sa neposrednim

pristupom obično se realizuju pomoću bistabilnih elektronskih kola. Broj bitova koji se istovremeno upisuje ili čita naziva se memorijska reč.

Posebnu kategoriju memorija sa neposrednim pristupom čine čitačke (fiksne, konstantne) memorije (engl. ROM - Read-Only Memory - memorija koja se može samo čitati). U ovim memorijama podaci su konstantni (stalno zapisani) na memorijskom medijumu.

Kod memorija sa direktnim pristupom i memorija sa sekvencijalnim pristupom memorijski medijum se na određeni način kreće u odnosu na sistem za čitanje i upis. Tu spadaju memorije sa magnetnim diskovima, disketama i memorije sa optičkim diskovima.

Kod memorija sa direktnim pristupom, kao što su npr. diskovi i diskete, nosilac podataka (disk) stalno rotira u odnosu na sistem za čitanje i upis, tako da se pristup nekom delu nosioca ciklički ponavlja. Srednje vreme pristupa ovim nosiocima je 5 — 15

Kod memorija sa sekvencijalnim pristupom, kao što je npr. magnetna traka, vrši se sukcesivna (redna) provera delova nosioca podataka dok traženi deo ne dođe u položaj da mu se može pristupiti. Vreme pristupa ovde može iznositi i nekoliko minuta.

Prema mogućnosti zadržavanja podataka po prestanku (isključenju) napajanja, razlikuju se:

- postojeane memorije i
- nepostojane memorije.

Kod postojanih memorija podaci se pri nestanku napajanja zadržavaju, što je slučaj npr. kod magnetnih diskova, disketa i magnetnih traka. Kod nepostojanih memorija po prestanku napajanja sadržaj se nepovratno gubi. Primer ovakvih memorija su poluprovodničke RAM memorije.

Za opis karakteristika memorije koristi se više različitih parametara od kojih su osnovni sledeći:

- 1) kapacitet memorije - broj bajtova ili bitova koji se mogu zapamtiti u memoriji;
- 2) vreme pristupa - vremenski interval potreban za čitanje ili upisivanje u memoriju.

Kod diskova i traka vreme pristupa je srednje vreme potrebno za pristup nekom mestu na medijumu radi čitanja ili upisa, a sastoji se od vremena traženja, potrebnog za postavljanje glave za upis čitanje iznad ćelije sa podacima i vremena prenosa podataka. Kako je vreme traženja znatno duže od vremena prenosa, spoljna memorija je organizovana u blokove kod traka i segmente kod diskova. To je osnovna jedinica prenosa između spoljne memorije i procesora ili spoljne memorije i operativne memorije.

4.1.3 Struktura memorijskog sistema

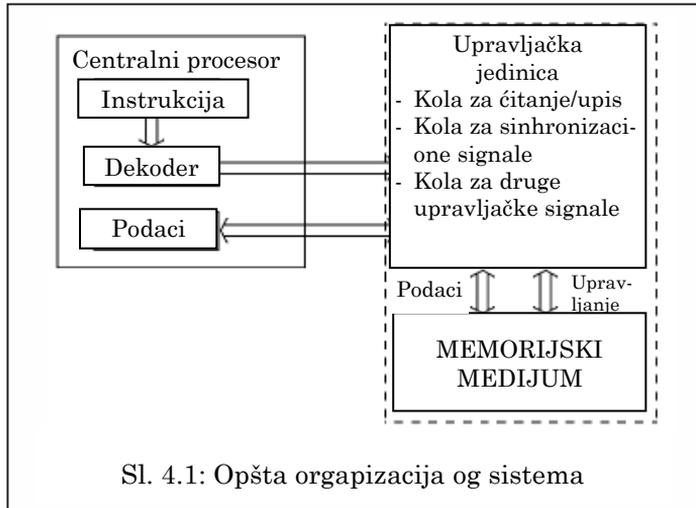
Osnovni cilj memorijskog sistema (memorije) računara jeste da opslužuje centralni procesor i druge delove za obradu podacima i programima. Dve osnovne operacije koje obezbeđuje memorijski sistem su:

- 1) pamćenje (upis) podataka,
- 2) pozivanje (čitanje) podataka.

Memorija je fizički i logički izdeljena na blokove, čije su veličine određene dužinama osnovnih tipova podataka u arhitekturi računara, ili njihovim celobrojnim umnošcima. Takvi blokovi memorije, kojima se pristupa odjednom, nazivaju se *memorijske lokacije*. One su karakteristične za operativne ili glavne memorije računara. Obično je dužina memorijskih lokacija jednaka dužini registara podataka u procesoru. Svaka memorijska lokacija ima jedinstvenu adresu, koja označava njeno mesto (poziciju) u okviru memorijskog sistema.

Za obezbeđenje osnovnih operacija memorijski sistem dobija određena uputstva od centralnog procesora, a pre svega koje funkcije treba da se izvrše (upis ili čitanje) i gde (adresa) treba da se izvrši upis ili čitanje. Pri tome centralni procesor obezbeđuje podatke koje treba zapamtiti pri operaciji upisa, odnosno mesto za podatke koji se dobijaju operacijom čitanja. Pristup memoriji na osnovu adrese, *adresni pristup*, podrazumeva da je poznata adresa podatka u memoriji i on je najčešće korišćen u memorijama.

Da bi memorija mogla izvršavati svoje funkcije, potrebno je obezbediti upravljanje, nezavisno od toga da li je reč o memoriji sa proizvoljnim ili nekim drugim pristupom. Zbog toga svaki memorijski sistem obično poseduje dva osnovna funkcionalna dela kao što je prikazano na Slici 4.1



- upravljačku jedinicu memorije i
- memorijski medijum.

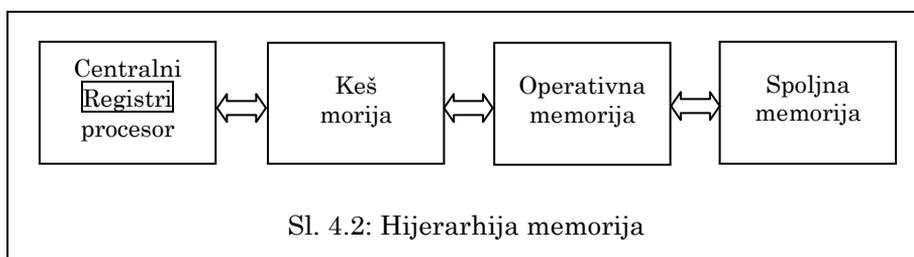
Upravljački sistem sadrži elektronske blokove koji služe za traženje, upis ili čitanje podataka na memorijskom medijumu. Memorijski medijum na kome se pamte podaci po pravilu sadrži skup jednakih memorijskih elemenata - lokacija (memorijskih ćelija), namenjenih za čuvanje binarnog koda veličine jedne memorijske reci.

4.1.4 Hijerarhija memorija

Većina računara radi mnogo efikasnije ako osim operativne (osnovne, glavne) memorije postoji i masovna (spoljna) memorija. Pri tome se u operativnoj memoriji nalaze podaci i programi koje procesor koristi u datom trenutku. Svi ostali podaci čuvaju se na spoljnoj memoriji i prenose se u operativnu memoriju kada su potrebni.

Ukupno raspoloživa memorija može se razmatrati kao hijerarhijski sistem komponenata koji se sastoji od svih uređaja za čuvanje podataka (memorijskih uređaja) koje koristi računarski sistem. Ovaj sistem sadrži sve memorije - od sporih, sa velikim kapacitetom, do relativno male ali brze operativne memorije i još manje i vrlo brze keš-memorije. Zbog toga se memorija računara organizuje u hijerarhijsku strukturu memorijskih uređaja koji na pojedinim nivoima imaju različite brzine i kapacitete (sl. 4.2).

U računaru se koriste sledeće memorije:



- registarska memorija,
- keš-memorija (ultrabrza memorija, priručna memorija),
- operativna memorija (osnovna memorija, glavna memorija),
- memorija sa direktnim pristupom (magnetni diskovi) i
- memorija sa sekvencijalnim pristupom (magnetne trake).

Registarska memorija predstavlja skup registara u procesoru koji se nazivaju registri opšte namene. Oni se koriste u različite svrhe, pre svega za privremeno pamćenje operanada, međurezultata, komponenata adresa i dr. Ovi registri rade brzinom koju ima centralni procesor tako da se operacije sa podacima koji su smešteni u registrima izvode brzinom rada procesora. Iz tog razloga svi procesori sadrže više desetina registara opšte namene.

Keš-memorija je mala, ultrabrza poluprovodnička memorija sa neposrednim pristupom. U njoj se čuvaju podaci i instrukcije iz operativne memorije koje procesor trenutno koristi. Keš-memorija služi za usaglašavanje brzine procesora i operativne memorije koja je sporija od procesora za red veličine (oko 10 puta). Ona omogućava povećanje brzine obrade, jer se u njoj nalaze

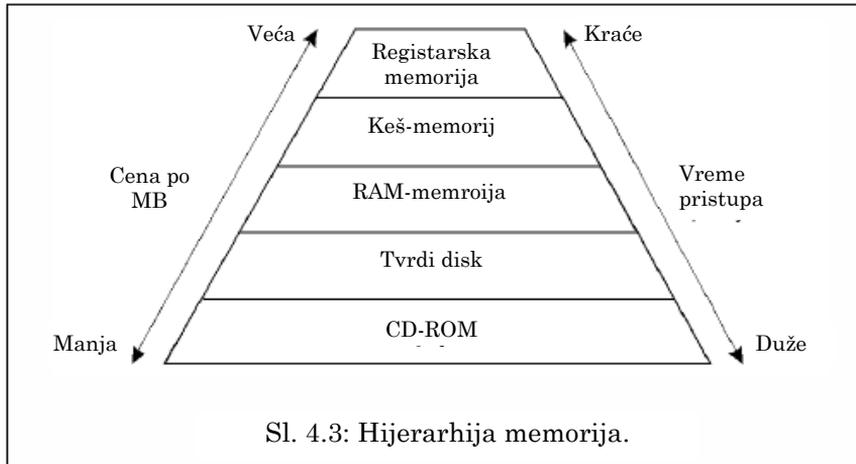
tekući podaci i tekuće instrukcije programa kojima procesor pristupa znatno brže, čime se povećava produktivnost rada procesora, odnosno vreme izvršenja programa. Navedeni redosled memorija odgovara povećanju nivoa u hijerarhiji, smanjenju brzine i cene po jednom bitu a povećanju kapaciteta. Svaki nivo može sadržati više modula (primeraka) odgovarajućih uređaja za dobijanje potrebnog kapaciteta datog nivoa memorije. Ovakva hijerarhijska struktura omogućava da se ekonomski efikasno usklade čuvanje (skladištenje) velikog broja podataka sa brzim pristupom podacima u procesu njihove obrade.

Na donjem nivou hijerarhije nalaze se relativno spore magnetne trake koje se koriste, pre svega, za arhiviranje podataka. Sledeći nivo čine magnetni diskovi koji se koriste kao spoljni memorijski uređaji za trajno čuvanje podataka i programa. Centralno mesto ima operativna memorija koja može neposredno razmenjivati podatke i sa centralnim procesorom i sa spoljnim uređajima (ponekad preko posebnog, tzv. ulazno-izlaznog procesora). Kada su procesoru potrebni podaci kojih nema u operativnoj memoriji, oni se unose iz spoljne memorije. Keš-memorija služi za povećanje brzine obrade pomoću koje tekuće instrukcije i tekući podaci mogu da se prenose u centralni procesor velikom brzinom. Time se kompenzuje razlika u brzini rada procesora i vremena pristupa operativnoj memoriji, jer je vreme pristupa keš-memorije približno jednako brzini rada procesora. Registarsku memoriju čini skup registara u procesoru ili posebna registarska polja (registarske matrice), a služi pre svega za privremeno pamćenje operanada i međurezultata.

Navedena hijerarhija se koristi kod optimalne organizacije sprege memorija različitog nivoa hijerarhije. Međutim, hijerarhija memorija može se uslovno posmatrati i za širu klasu mem-orija (Slika 4.3).

Razlozi za uvođenje hijerarhije memorija su pre svega ekonomski. Naime, cena proračunata na jedan bit obrnuto je proporcionalna nivou memorije u hijerarhiji. Zato bi bilo vrlo skupo sve programe i podatke čuvati u operativnoj memoriji, naročito ako nisu potrebni procesoru.

Deo operativnog sistema koji upravlja tokovima podataka između svih memorijskih uređaja naziva se sistem za upravljanje



memorijom. On raspodjeljuje programe i podatke među različitim nivoima hijerarhije memorije saglasno sa očekivanom učestanošću njihovog korišćenja.

4.2 Operativna memorija

Operativna memorija (osnovna memorija, glavna memorija) (engl. RAM - Random Access Memory memorija sa proizvoljnim pristupom) jeste memorija sa neposrednim (proizvoljnim) pristupom koja služi za čuvanje podataka koji se koriste u procesu izvršenja mašinskih operacija u aritmetičko-logičkoj i upravljačkoj jedinici procesora. To su polazni podaci za obradu, međurezultati i konačni rezultati obrade i instrukcije programa koji se izvršava.

U procesu obrade ostvaruje se stalna komunikacija između centralnog procesora i operativne memorije. Iz operativne memorije u procesor se prenose instrukcije programa i operandi nad kojima se izvršavaju operacije predviđene instrukcijama programa, a iz procesora u operativnu memoriju se šalju na čuvanje međurezultati i konačni rezultati obrade.

Karakteristike operativne memorije neposredno utiču na osnovne pokazatelje samog računarskog sistema, pre svega na njegovu brzinu.

Memorija sa neposrednim pristupom sastoji se od memorijskih lokacija realizovanih bistabilnim kolima i odgovarajućih elektronskih kola koja obezbeđuju upis podataka u memoriju i čitanje

podataka iz memorije. Kako se na osnovu adrese može pristupiti bilo kojoj lokaciji, koristi se termin memorija sa neposrednim pristupom ili memorija sa proizvoljnim pristupom.

U memoriji se binarni podaci čuvaju u obliku grupa bitova koje se nazivaju memorijske reči. Svaka reč čuva se u posebnoj lokaciji a čita se ili se upisuje kao celina. Svaka memorijska reč može predstavljati brojni podatak, alfanumerički podatak, kod instrukcije ili bilo koji drugi binarni kod.

Za realizaciju operativne memorije danas se gotovo isključivo kao memorijski elementi koriste poluprovodnička bistabilna kola.

4.3 Spoljna memorija

Spoljne ili sekundarne memorije su memorije velikog kapaciteta (za nekoliko redova veličina veće od operativne memorije). Za pamćenje podataka koristi se magnetno tvrdo materijal, obično feritni prah debljine oko 0.1 mm koji se nanosi na noseću površinu. Noseć površina plastična traka, plastična kartica, plasični disk (disketa) ili metalni aluminijumski disk.

Upis i čitanje se vrši pomoću magnetne glave koja se nalazi udaljena od medijuma samo nekoliko mikrometara.

4.3.1 Disketa

Funkciju spoljnih memorija danas najčešće obavljaju disk sistemi koji prevazilaze nedostatke medijuma sa sekvencijalnim pristupom. Prema principu upisivanja informacija, disk sistemi se mogu podeliti na magnetne i optičke. U zavisnosti od toga da li se magnetni nosilac informacije može menjati ili je fiksno ugrađen u računarski sistem, razlikuju se disk sistemi sa zamenljivim (izmenljivim) medijumom: disketa, zamenljivi tvrdo disk, zamenljivi paket diskova i fiksni tvrdo diskovi.

Za disketu se koristi i naziv flopi disk (engl. floppy - elastičan, savitljiv). Sastoje se od plastičnog materijala, prekrivenog magnetnim slojem. Nalazi se u omotaču kvadratnog oblika od plastike ili plastificiranog papira, koji služi za

zaštitu od fizičkih oštećenja, tako da je čitava magnetna površina zaštićena. Na omotaču postoji otvor za pristup disketi preko koga se realizuje čitanje i upis podataka;

- otvor u centru koji omogućava prilaz pogonskih motora jedinice diskete za okretanje; - indeksni (referentni) otvor koji omogućava da se detektuje početak i kraj staze, tj. da se razdvoje prvi i poslednji sektor; - četvrtasti izrez sa jedne strane omotača koji obezbeđuje zaštitu od neželjenog upisa (kod novijih modela disketa ovaj otvor se zamenjuje malim preklopnikom na samom omotaču).

Personalni računari danas koriste standardne diskete od 3.5 inča¹ (8.9 cm) i zapis podataka visoke gustine, - HD (high density).

Da bi se ostvario upis podataka na disketu, potrebno je prethodno formirati staze (trake) i sektore u kojima će se upisivati podaci i odrediti način organizacije podataka. Taj postupak se naziva formatizovanje diskete. Prilikom formatizovanja, disketa se deli na koncentrične krugove - staze, trake ili piste. Staze se zatim dele na sektore.

Sektor predstavlja osnovnu količinu memorije (jedinicu) koja se može čitati ili upisati na disketu, bez obzira na to koliko podataka imamo. Dve staze sa obe strane diskete na istom rastojanju od centra čine cilindar. Kako se postupkom formatizovanja određuje broj staza i veličina sektora, time se definiše i gustina upisa podataka. Formatizovanje može biti programsko i fiksno. Kada je veličinu sektora definisao unapred, mehaničkim putem, proizvođač onda je to fiksno formatizovanje. Za ovo formatizovanje je potrebno da indeksni otvor postoji za svaki sektor. Zbog svoje jednostavnosti i fleksibilnosti danas se uglavnom koristi programsko formatizovanje. To znači da se određenim programom prostor na disketi raspodeli na staze i sektore radi lakšeg pristupa. Kasnije se ta raspodela prostora može promeniti ukoliko se primeni drugačije formatizovanje. Diskete koje se programski formatizuju imaju samo jedan indeksni otvor koji razdvaja prvi i poslednji sektor svake trake.

Sektor na disketi imaju ulogu kao blokovi podataka. Opisaćemo organizaciju disketa koja se koristi za diskete kod personalnih računara. Svi sektori diskete mogu se podeliti u če-

¹1 inč=2.54 st

tiri vrste:

- 1) BOOT sektor (sektor za pokretanje računara) i uvek počinje prvim sektorom na disku,
- 2) DIR sektori (sektori za kataloge),
- 3) FAT (File Allocation Table) sektori (sektori sa podacima o rezervisanom prostoru za datoteke),
- 4) INFO sektori (sektori sa podacima)

Prve tri vrste sektora služe za specijalne namene (to su sistemski sektori), dok INFO sektori služe za smeštanje podataka.

Uređaj koji upisuje i čita podatke sa diskete naziva se jedinica diskete. To je elektromehanički uređaj koji sadrži četiri osnovna dela:

- upisno-čitajuća magnetna glava,
- elektromehanički deo,
- fotoelektronski sistem,
- upravljačka elektronika.

Kontroler diskete (RB kontroler) može biti ugrađen u osnovnu ploču računara, ali se najčešće priključuje u vidu kartice (ploče elektronike), zajedno sa kontrolerom tvrdog diska, na neki od slotova (pozicija) za proširenje računara.

4.3.2 Tvrđi magnetni disk

Memorijski medijum koji je u novije vreme dobio najvažniju ulogu među nosiocima podataka u računarskom sistemu svakako je tvrđi disk (engl. hard disc). On objedinjuje dobra svojstva ostalih nosilaca u jednu optimalnu celinu. Ovaj memorijski medijum funkcioniše na principu zapisa bitova podataka u malim ćelijama od feromagnetnih materijala sa dva stabilna magnetna stanja, koja predstavljaju logičke vrednosti 0 i 1. Ova stanja su određena veličinom ili smerom magnetnog fluksa ćelije. Električna struja menja i omogućuje prepoznavanje magnetnih stanja ćelija uz pomoć induktivne glave za upis i čitanje podataka. Induktivne glave su postavljene na vrlo malom rastojanju od magnetnih ćelija, u uzajamnom dejstvu sa njima mogu da menjaju magnetna

stanja ćelija pri upisu podataka, ili da generišu električne signale koji pokazuju stanje ćelija pri čitanju podataka.

Karakteristike tvrdog diska su:

- direktan pristup podacima,
- velika brzina pristupa,
- veliki memorijski kapacitet,
- niska cena po jedinici memorije,
- male dimenzije u odnosu na kapacitet,
- dugotrajno pamćenje podataka bez zahteva za dodatnom energijom,
- visoka pouzdanost.

Tehnologija proizvodnje tvrdih diskova je u stalnom usponu, tako da permanentno raste kapacitet, brzina, pouzdanost, a smanjuju se dimenzije, prosečno vreme pristupa i cena. Princip upisa i čitanja podataka kod diska je identičan kao i kod jedinica magnetne trake ili diskete. Koriste se ili pojedinačni diskovi ili paketi diskova. Mogu biti fiksni ili zamenljivi. Zamenljivi diskovi imaju kapacitet do nekoliko stotina GB.

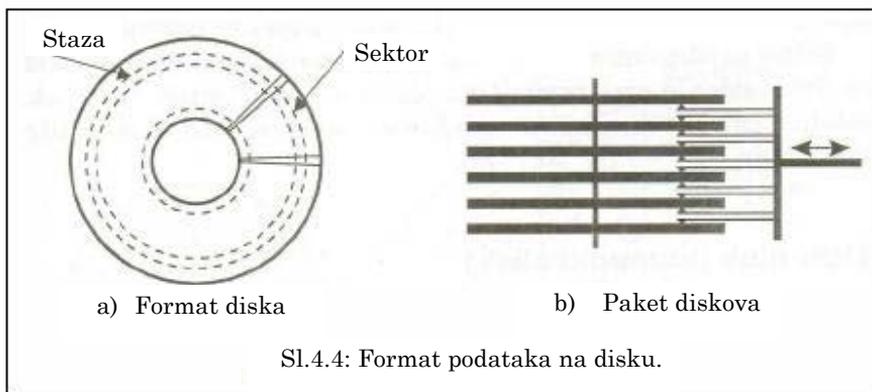
Na Sl. 4.4(a) prikazani su izgled diska i format podataka na disku. Disk je podeljen na staze - koncentrične kružne putanje, cilindre koje čine sve staze sa istim rednim brojem, i sektore, delove staza koji se kao celine čitaju ili upisuju na disk. Sektori se definišu formatizovanjem na jedan od sledećih dva načina:

1. softverski (programski), tzv. meko formatizovanje,
2. pomoću unapred unetih rupica, tzv. tvrdo formatizovanje.

Jedinice tvrdog diska su uređaji koji upisuju i čitaju podatke po magnetnoj površini diska i mogu biti:

- sa pojedinačnim diskovima smeštenim u kasetu - kasetnim diskovima,
- sa više disk-ploča objedinjenih fiksno oko iste osovine u paket - paketima diskova Sl. 4.4(b).

Diskovi isto tako mogu biti zamenljivi ili nezamenljivi (fiksni) u jedinici diska. Zamenljivi diskovi se stavljaju u jedinicu



Sl.4.4: Format podataka na disku.

samo u vreme upisa ili čitanja podataka. Nezamenljivi ili fiksni diskovi čine integralni deo uređaja tvrdog diska. Oni su najčešće zaštićeni od prašine i drugih čestica, pa su po pravilu većeg kapaciteta, brzine, pouzdanosti i mogu raditi u težim uslovima okruženja.

Danas prevladavaju nezamenljivi - fiksni diskovi sa više disk-ploča i pokretnim glavama Sl. 4.4(b). Između diskova se nalaze poluge - nosači koji na svom vrhu imaju po dve upisno-čitajuće glave, za gornju i za donju poršinu. Poluge su pričvršćene za pristupni mehanizam aktuator. Aktuator predstavlja uvlačiv sklop tako da se glave mogu radialno pomerati. Sve upisno-čitajuće glave se pomeraju istovremeno, što znači da se u datom trenutku svim stazama koje su na istom rastojanju od centra diska može pristupiti istovremeno.

Kontroler tvrdog diska ima ulogu da obezbedi sve neophodne funkcije za spregu između centralnog procesora računara i jedinice diska. On prevodi podatke i komande sa magistrale računara u kontrolne signale i zahtevani tok podataka. Za razliku od disketa, kod kojih, zbog prenosivosti medijuma, način kodiranja i zapisa podataka mora da bude standardan (jedinствен), postoji više vrsta disk kontrolera: MFM, RLL, AT/IDE, SCSI, ESDI, itd. Disk koji je formatizovan jednom vrstom kontrolera ne može koristiti drugačiji kontroler.

Tvrđi disk je podeljen na stranice, cilindre, staze i sektore. Pri ovakvoj organizaciji sektor predstavlja najmanju re

alnu količinu podataka koja se može pročitati ili upisati na disk. Određeni sektori se koriste za specijalne namene, a ostali za podatke. Sektori za specijalne namene su:

- 1) BOOT sektor (sektor za pokretanje računara),
- 2) DIR sektori (sektori za kataloge),
- 3) FATsektori (sektori o rezervisanom prostoru za datoteke),

Očigledno je da je osnovni princip organizacije podataka sličan disketnom.

Tvrđi disk se može podeliti na particije kojima se može pristupati kao zasebnim celinama. Operativni sistem particije prikazuje kao zasebne diskove.

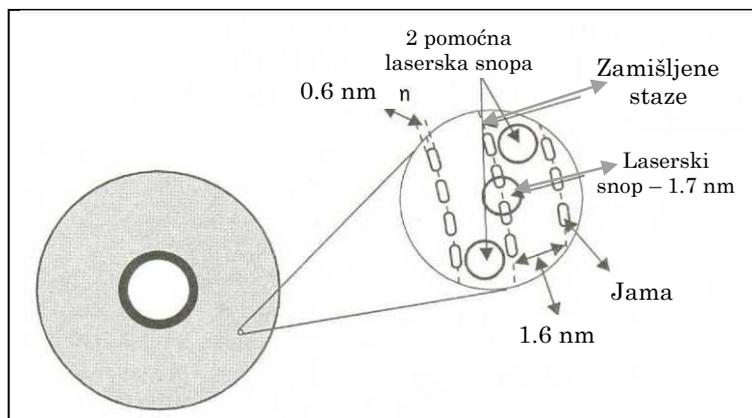
Sektor sa adresom: strana 0, cilindar 0, sektor 1, predstavlja prvi sektor na disku. Prvi sektor je uvek rezervisan i naziva se BOOT sektor. Ovaj sektor je najvažniji, jer ukoliko je on neispravan, oštećen, ceo disk može biti neupotrebljiv.

4.3.3 Optički disk (kompakt disk)

Optički disk predstavlja medijum u obliku diska prečnika 120 mmna kome su zapisani podaci u binarnom obliku (Sl. 4.5). Za upis i čitanje se koristi poluprovodnički laser i optički sistem koji generiše svetlosni snop prečnika oko jednog mikrometara fokusiran na medijumu za pristupanje svakom bitu podataka.

Pri upisu podataka snaga mlaza se bira tako je dovoljna da zagrevanjem osvetljene vrlo male površine medijuma (bicke ćelije) promeni povratno ili nepovratno njegove optičke karakteristike. Pri čitanju, snaga mlaza je znatno manja tako da ne izaziva nikakvu promenu na medijumu, svetlost koju medijum odbija detektuje se a njena jačina ili polarizacija se posmatraju da bi se utvrdilo da li je zapisana vrednost "0" ili "1".

Prva generacija "jednobrzinskih" uređaja za kompakt diskove bila je zasnovana na konstrukciji uređaja za zvučne kompakt diskove, što je, sa ispravljanjem grešaka iznosilo 150 kB/s. Kako ima više sektora na spoljašnjoj ivici kompakt diska nego u njegovom središtu, uređaj sadrži servo motor da uspori brzinu obrtanja ka spoljnim stazama, da bi održao konstantnu brzinu prenosa



Sl. 4.5: Optički disk.

podataka preko laserske glavza čitanje. Dok audio i video diskovi treba da se čitaju jedinstvenom brzinom, za očitavanje drugih podataka sa kompakt diska nema ograničenja. Zaista, potrebno je da se podaci očitavaju što brže. Sa napredovanjem tehnologije povećavala se i brzina očitavanja i već početkom 1998. godine uređaju su imali brzinu 32h, odnosno brzinu prenosa podataka od 4.8 MB/s.

Danas kompakt disk predstavlja najperspektivniji medijum za čuvanje podataka i standardni su deo računarske opreme. Najviše se koriste CD-ROM diskovi (od engl. Compact Disc Read-Only Memory - čitačka memorija na kompakt disku) - kompakt diskovi čija je prvobitna namena bila zapisivanje digitalizovanih zvučnih podataka, a kod računara služe za pamćenje fiksnih podataka koji se mogu samo čitati, kao što su razne baze podataka, programi, telefonski imenici, rečnici, enciklopedije i dr. Posebno su pogodni za pamćenje multimedijalnih digitalizovanih podataka: teksta, zvuka i slike istovremeno. Glavni nedostatak je nemogućnost upisivanja novih podataka.

Osnovne karakteristike optičkih diskova su:

- veliki memorijski kapacitet (150 MB - 10 GB),
- pouzdanost i trajnost,
- visoka gustina zapisivanja podataka (1010 bit/cm²),

- zamenljivi medijum,
- multimedijalni karakter.

Podaci na njemu su zapisani po spiralnoj putanji (za razliku od tvrdih diskova i disketa). Radi boljeg iskorišćenja memorijskog prostora disk se okreće konstantnom linearnom brzinom (1.3 m/s) u odnosu na optičku glavu. Time je omogućeno da gustina zapisivanja i dužina sektora bude ista po celoj površini medijuma, ali je ugaona brzina različita: za spoljnje staze je oko 200 o/min, a za unutrašnje oko 500 o/min.

Osnovne karakteristike zapisivanja na optičkim diskovima su sledeće. Veličina diska je 120 šš, staze su spiralne i ima ih oko 20000, rastojanje između staza je 1,6 mikrometara. Površinska gustina zapisivanja je oko 100 puta veća nego kod magnetnih diskova (10^{10} bit/cm² prema 108 bit/cm²). Linijska gustina zapisa je 25000 bpi (984 bit/mm). Kapacitet diska je 700 MB.

Optički diskovi imaju vreme pristupa 100 - 200 ms, što je još uvek znatno duže od vremena pristupa kod tvrdih diskova. Brzina prenosa podataka dostiže više od 1 MB/s. Materijali od kojih se proizvode mogu biti različiti (aluminijum, staklo, plastika, plemeniti metali). Svi oni imaju svojstvo da dobro reflektuju svetlost (efekat ogledala). Svaki disk, bez obzira na to od čega je načinjen, presvlači se tankim slojem tvrde plastike.

Upis i čitanje podataka se ostvaruje pomoću laserskog zraka. Poznato je da laserski zrak predstavlja paralelni snop svetlosti koji ima jednu talasnu dužinu u istoj fazi. Kada se upisuju podaci snaga laserskog mlaza se bira tako da bude dovoljna da zagrevanjem osvetljene vrlo male površine medijuma (bicke ćelije) promeni povratno ili nepovratno njegove optičke karakteristike. Tada se stvaraju udubljenja (jame) širine 0,6 mikrometara i dubine 1/4 talasne dužine laserskog zraka koji predstavljaju zapisane podatke.

Pri čitanju je snaga mlaza znatno manja, tako da ne izaziva nikakvu promenu medijuma. Svetlost koju medijum odbija se detektuje. Zatim se analiziraju jačina i polarizacija kako bi se utvrdilo da li je zapisana vrednost "0" ili "1". Ukoliko je laserski zrak naišao na udubljenje, dolazi do pomeranja faze za 180 stepeni, što prouzrokuje pojavu negativne interferencije i slabljenje svetlosti koja se vraća fotoćeliji. Pored glavnog

laserskog zraka, emituju se i dva pomoćna koji imaju ulogu da odrede položaj optike glave u odnosu na stazu. Za svaki pomoćni laser postoji i odgovarajuća fotoćelija. Ukoliko su pomoćne fotoćelije nejednako osvetljene šalju se signali za korekciju položaja glave. Glava se pomera posebnim motorom.

4.4 Pitanja za proveru znanja

1. Koje osobine treba da poseduje neki medijum da bi se mogao koristiti za pamćenje podataka? Navesti primere.
2. Na osnovu kog podatka se vrši pristup memoriji da bi iz nje uzeli ili u nju stavili podatak?
3. Da li vreme pristupa nekoj ćeliji operativne memorije zavisi od njene lokacije unutar operativne memorije?
4. Nabrojati osnovne kvantitativne parametre memorije.
5. Koja je najmanja jedinica podataka kojoj se može pristupiti na disku?
6. Koje sektore sadrži disketa?

Glava 5

Ulazno izlazni uređaji

Korisnicki interfejs čine ulazni i izlazni uređaji koji omogućuju korisniku da upravlja i prima odgovore od računarskog sistema. Najpoznatiji ulazni uređaji su tastatura džojstik, digitajzer, tableta miš, sistem za prepoznavanje govora itd.

Kao izlazni uređaji najčešće se koriste displeji koji se izvode u varijantama od prostih indikatora do velikih grafičkih displeja. Izlazni uređaji su takođe i štampači, generatori zvuka i dr.

5.1 Ulazni uređaji

Ulazni uređaji služe za unošenje podataka u računar (centralnu jedinicu, operativnu memoriju ili sekundarnu memoriju).

Prema načinu unošenja podataka, svi ulazni uređaji mogu se svrstati u dve grupe:

- a) uređaje za ručno unošenje podataka,
- b) uređaje za automatsko unošenje podataka.

U grupu uređaja sa ručnim unošenjem podataka u računar svrstavaju se: tastatura, miš, svetlosno pero, palica za igru i dr. Ručno unošenje je dosta sporo, a koristi se kada obim podataka nije veliki.

Tastatura predstavlja niz tastera koji, kada se pritisnu, proizvode u računaru binarni kod pritisnutog znaka. Tastaturom

se prosečno unose dva znaka u sekundi. Kod manjih računara tastatura predstavlja primarni ulazni uređaj za unošenje i programa i podataka.

Miš predstavlja tzv. pokazivački uređaj koji se rukom pokreće po ravnoj površini. Ovi pokreti se prenose u računar i uzrokuju kretanje kursora na ekranu. Miš se koristi, pre svega, za selekciju komandi iz menija ili za izbor ikone i aktiviranje određene komande.

Komandna palica (džhojstik) jeste ulazni uređaj u obliku ručice koja generiše signale za brzo kretanje kursora ili nekog drugog simbola po videoekranu. Koristi se najčešće za upravljanje raznim objektima u video igrama.

Uređaji za automacko neposredno unošenje mogu da čitaju podatke sa posebnih obrazaca (formulara), zatim štampani i rukom pisani tekst, grafičke podatke (crteže, slike) i prepoznati glas (govor). U ovu grupu uređaja svrstavaju se i analogno-digitalni konvertori, uređaji za prijem podataka sa telekomunikacionih linija i drugi.

Pri masovnim obradama, gde je obim podataka značajan, kreirani su obrasci koji mogu služiti i kao originalni dokumenti i istovremeno kao nosioci ulaznih podataka, sa kojih se unošenje može vršiti automacki pomoću posebnih uređaja za čitanje. Čitači markiranih obrazaca, ili optički čitači markera namenjeni su za detekciju prisustva ili odsustva markera (oznake) na svakoj dozvoljenoj poziciji na dokumentu. Oznake mogu biti pisane rukom, kucane na pisačkoj mašini, štampane i sl. Koriste se dva standardizovana pisma: OCR A (engl. Optical Character Recognition - optičko prepoznavanje znakova) i OCR B, čiji su znaci optimizovani za mašinsko prepoznavanje.

Uređaji za optičko čitanje prugastog (bar) koda koriste laserski zrak za čitanje oznaka sa prugastim kodom, a mogu biti u jednom od sledeća dva oblika:

- a) u obliku olovke kojom se čita oznaka,
- b) u obliku proreza iznad koga se prevlači oznaka.

Skener (ili skaner) predstavlja ulazni uređaj koji služi za optičko čitanje i diskretizaciju slika ili crteža i njihovo unošenje u memoriju računara. Koristi se pri prepoznavanju i

obradi slika pomoću računara.

Grafička tabla ili digitalizator predstavlja ravnu površinu koja zajedno sa posebnim perom služi za digitalizaciju i unošenje grafičkih podataka u računar, npr. mapa, tehničkih crteža i sl.

Treba napomenuti da ulazni dokumenti mogu biti zapisani i na nekom memorijskom medijumu. Primere predstavljaju prenosni terminali za unos podataka pri očitavanju strujomera, vodomera i sl.

5.1.1 Tastatura

Tastatura je osnovni uređaj za ručno unošenje podataka i programa u računar. Sastoji se od niza tastera (dirki) koji, kada se pritisnu, proizvode u računaru binarni kod pritisnutog znaka.

Raspored znakova na tastaturi je sličan kao i kod pisaće mašine i može da se usklađuje prema standardima države kojoj je tastatura namenjena. Broj dirki zavisi od vrste i namene tastature. Na tastaturi se mogu uočiti pet grupa dirki:

- Alfadecimalne dirke sadrže aladecimalne znakove i znakove interpunkcije. Ovo je najbrojnija grupa dirki, ima ih ukupno 56. Osim 26 tastera sa engleskim alfabetom, tu su i dirke sa posebnim znakovima i brojevima, kontrolne dirke isl.
- Numeričke dirke - posebna grupa tastera namenjena za brzo unošenje numeričkih podataka i znakova osnovnih arit-metičkih operacija. Ona se aktivira pritiskom na dirku NUM Lock, a ponovnim pritiskom na istu dirku isključuje se numerička tastatura.
- Tasteri za pomeranje kursora. Pritiskom na neku dirku sa strelicama dolazi do pomeranja kurzora u željenom pravcu.
- Funkcijski tasteri - njihova namena se uglavnom definiše konkretnim programom. Na primer, u većini programa dirka *F1* u većini programa znači "pomoć".
- Upravljačke dirke - koriste se za direktno pokretanje nekih aktivnosti računara, promenu načina rada itd. Na primer, u programu za obradu teksta Word dirka "Home" ima funkciju pomeranja kurzora na početak reda, ili stranice, pritisak

na dirku "Page UP" ima funkciju "skok" za jednu stranicu ispred tekuć i slično.

Na tastaturama se nalaze i svetlosne (LED) diode ispod kojih stoje oznake indikatora stanja: NUM, LOCK, CAPS LOCK i SCROLL LOCK. Ova stanja, kao i njihova indikacija, mogu se menjati ručno ili programom iz računara.

Tastaturu je moguće i zaključati. Na prednjoj strani centralne jedinice PC računara nalazi se bravica sa ključem, koja dozvoljava, ili zabranjuje komunikaciju tastature i računara.

Računar sadrži program za analizu tastature i centralni procesor treba često da generiše zahtev za prekid kako bi se aktivirao ovaj program radi detektovanja trenutka kada je određena dirka tastature pritisnuta, i kada je otpuštena. Interval najkraćeg pritiska dirke koji se mora detektovati je od 20 do 100 tz i zavisi od tipa tastature, dok se generisanje zahteva za prekid generiše jednom u svakoj milisekundi.

5.1.2 Miš

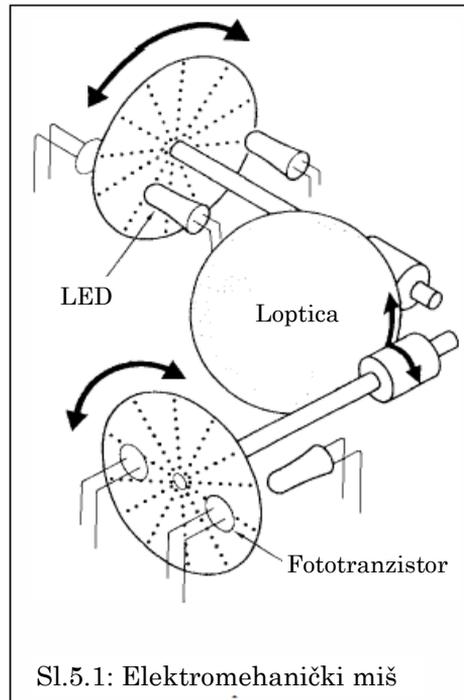
Za komunikaciju između čoveka i računara značajan je svetleći marker na ekranu monitora koji se naziva kursor. On ukazuje na poziciju gde će biti prikazan sledeći element teksta ili slike. Položaj kursora se može menjati pomoću kursorskih tastera na tastaturi. Danas većina programa podržava promenu položaja kursora pomoću specijalnog ulaznog uređaja - miša.

Miš je uređaj koji se postavlja na sto pored tastature i direktno pretvara pomeranje ruke u pomeranje kursora. Sastoji se od loptice sa mehanikom i prateće elektronike smeštene u plastično kućište. Na kućištu miša obično se nalaze, dva ili tri tastera koji omogućavaju izbor komandi iz menija programa ili fiksiranje neke tačke crteža na ekranu. Pomeranjem miša po ravnoj podlozi, loptica se okreće a prateće mehaničke i elektronske komponente to pomeranje pretvaraju u električne signale, pa se kursor kreće po ekranu u željenom smeru. Kada se se pokazivač miša dovede u željenu poziciju, akcija se vrši pritiskom i otpuštanjem tastera miša (klikom). Moguće su različite akcije mišem: jednostruki klik, pritisak na taster i njegovo držanje u pritisnutom položaju i slično. Na Slici 5.1 prikazana je struktura elektromehaničkog

miša.

Sinhronizaciju između kretanja miša i kursora na ekranu obezbeđuje odgovarajući softver. Pri aktiviranju miša računar prekida sve druge aktivnosti i pomera kursor na ekranu na osnovu dobijenih ulaznih signala.

Elektromečanički miševi su jevtini sklopovi, ali zbog toga što skupljaju nečistoće po površini po kojoj se pomeraju, posle dužeg vremena postaju nepouzdati.



Osim miša, postoje i drugi uređaji za ručno unošenje podataka i kontrolu kursora koji funkcionišu po sličnom principu. Među njima se najčešće koriste trekbol, koji je namenjen za korišćenje kod onih računara koji koriste relativno velika pomeranja, a pri tome zahtevaju veliku tačnost kao i veliku rezoluciju. Na žalost, trekbol sa visokim performansama je relativno skup te se zbog toga ređe koristi.

5.1.3 Grafička tabla digitalizator

Grafičke table se koriste za najpreciznije kreiranje objekata u programima za crtanje. Sastoji se iz dva dela:

- table,
- specijalnog miša.

Tabla je povezana na strujno kolo i može da magnetnim senzorom detektuje kretanje po svojoj površini. Pomeranjem specijalnog miša generišu se odgovarajući signali koji se šalju računam. Za razliku od običnog unosa mišem, ovde se kursor automacki postavlja na ono isto mesto na kojem je i miš na tabli. Dakle, prenose se apsolutne koordinate miša na grafičkoj tabli, dok je kod standardnog miša položaj kursora relativan u odnosu na poziciju na podlozi.

Postoje i digitalizatori čiji "miš" ima oblik olovke. Princip rada je isti i rad je često konformniji. Neki modeli digitalizatorskih tabli imaju i osetljivost na jačinu pritiska "olovke". Savršeniji grafički programi podržavaju upotrebu ovakvih tabli i daju deblju ili tanju liniju, svetliju ili tam-niju nijansu boje, simulirajući rad obične olovke.

5.1.4 Skener

Optički čitač je uređaj koji prihvata štampane, pisane ili cr-tane podatke sa originalnih dokumenata i prenosi ih u računar u obliku pogodnom za obradu. Služi za automacku pripremu podataka iz izvornih dokumenata. Posle unošenja podaci se obrađuju, a vrsta obrade zavisi od vrste znakova na originalnom dokumentu. Na primer, postoje optički čitač znakova (OCR - Optical Character Reader), optički čitač markera (OMR - Optical Magk Reader), čitač prugastog koda (engl. Bar Code Reader) itd.

Skener je ulazni uređaj kojim se realizuje prva faza u ovom procesu prenošenje slike dokumenta sa papira ili nekog drugog nosioca podataka u računar. Naziv potiče od engleske reči scanning - pregledanje, analiza.

U zavisnosti od oblika medijuma na kojem se izvorni dokument nalazi, postoje:

- stoni skeneri,
- ručni skeneri.

Glavna razlika između ovih tipova je u načinu kako čitaju dokument. Stoni skeneri su dovoljno velikih dimenzija da odrednom skeniraju dokument veličine A4 formata. Izgledom podsecaju na mašine za fotokopiranje. Ručni skeneri su po konstrukciji jednostavniji. Zahvatanje podataka se odvija tako što se skenerom pažljivo prelazi preko dokumenta koji se čita. Nemaju nikakve pokretne delove, i znatno su jeftiniji od stonih, ali su i manje pouzdani. Bez obzira na način zahvatanja podataka funkcionisanje oba tipa skenera je identično.

Postoje dva osnovna načina da se neki dokument skenira u boji. Pošto je poznato da se od tri osnovne boje (crvene, zelene i plave) mogu formirati ostale boje, onda je najjednostavnije tri puta skenirati isti dokument koristeći odgovarajući filter za svaku od ovih boja; dakle, skenirati u tri prolaza. Kasnije, specijalni program dobijene tri monohromacke slike udružuje u jednu sliku u boji.

Drugi način je da sam skener ima tri senzorska polja, za svaku od osnovnih boja po jedno. Ta polja imaju već ugrađene filtre za crvenu, zelenu ili plavu boju. Kod takve konstrukcije je moguće sliku skenirati u jednom prolazu.

5.1.5 Čitači prugastog koda

Poslednjih godina se prugasti (bar) kod (Sl. 5.2) sve više koristi za obeležavanje proizvoda, a njegova upotreba je definisana međunarodnim standardima. Sastoji se od debljih i užih linija i odgovarajućih razmaka između njih, pomoću kojih se kodiraju decimalne cifre.

U praksi se danas najviše koristi međunarodni sistem označavanja proizvoda (ranije značenje European Article Number -Evropski broj artikla) u dve verzije: EAN-13 trinaestocifreni broj (kod) proizvoda, i skraćena verzija EAN-8, i jedinstveni američki sistem šifriranja proizvoda UPC (Universal Product Code - univerzalni kod proizvoda) sa verzijom UPC-A dvanaestocifrenog koda, i verzijom UPC-D koja može imati od 14 do 32

cifre. Koristi se takođe i skraćena verzija UPC-E koda UPC-A sa šest cifara, pomoću kojih se posle rekonstrukcije dobijaju svih dvanaest cifara.

Zbog pogodnosti čitanja cifarski podaci se štampaju ispod oznaka prugastog koda. Oznake prugastog koda se jednostavno čitaju korišćenjem svetlosnog pera ili skenera.



Svetlosno pero je jedna vrsta skenera koji se sastoji od emitora svetlosti i detektora reflektovanog zraka. Prelaženjem svetlosnim perom preko oznaka prugastog koda detektuje se niz signala koji odgovaraju tamnim i belim vertikalnim crtama. Dekodiranjem ovih signala dobijaju se odgovarajući numerički podaci.

Svaka decimalna cifra se kodira sa dve tamne i dve svetle vertikalne linije koje mogu da budu različitih širina, ali sve četiri linije formiraju jedno polje uvek iste širine. Na sredini oznake prugastog koda nalaze se dve tamne vertikalne linije kao znak koji deli levu i desnu polovinu. Sve brojke u levoj polovini simbola počinju svetlom vertikalnom linijom i imaju neparan broj tamnih elemenata, dok u desnoj polovini počinju tamnom linijom i imaju paran broj tamnih elemenata. Time je omogućeno da se skeniranje može vršiti i sleva nadesno i zdesna nalevo.

5.2 Izlazni uređaji

Izlaz rezultata obrade, tj. izdavanje podataka iz računara može imati sledeće svrhe:

- a) privremeno pamćenje na mašinski čitljivim nosiocima,
- b) saopštavanje ili prikazivanje krajnjih rezultata obrade,
- v) neposredno korišćenje.

Privremeno zapamćeni podaci namenjeni su za neku drugu, obično vremenski kasniju obradu. Kao medijumi se koriste pre svega magnetni nosioci, dok su nekada korišćene bušene trake i kartice.

Krajnje rezultate obrade računar saopštava čoveku - korisniku pripremom raznih izveštaja u štampanom (trajnom) obliku, prikazom na ekranu u obliku teksta ili grafičkih podataka ili pripremom crteža na uređaju za crtanje (crtaju). Rezultati obrade su najčešće informacije koje služe da korisnik na osnovu njih donosi odluke.

Podaci za neposredno korišćenje su izlazni rezultati koji se bez posredovanja čoveka koriste za daljinski prenos, ili za automacko preduzimanje akcija, npr. pri neposrednom upravljanju procesima, uključivanju i isključivanju određenih izvršnih uređaja i si. U mnogim slučajevima neposrednog upravljanja predviđena je i mogućnost da čovek ima pristup izlaznim podacima, radi kontrole ili radi preduzimanja određenih hitnih akcija u slučaju havarije i dr.

Videoekran služi za prikazivanje teksta i, eventualno, grafičkih podataka. Zajedno sa tastaturom, mišem i svetlosnim perom ekran predstavlja danas najbolje sredstvo za neposrednu komunikaciju čoveka i računara. Na taj način ekran služi ne samo kao izlazni već i kao ulazni uređaj. Za znatno preciznije prikazivanje grafičkih podataka, npr. u sistemima za automacko projektovanje, koriste se grafički videoekrani. Na njima se slike formiraju od vektorski ili rasterski zapamćenih grafičkih podataka.

Matrični ili segmentni pano (informaciona tabla) predstavlja ekran u obliku ravne ploče namenjen za prikazivanje tekstualnih i grafičkih podataka većih dimenzija i za veći broj korisnika. Za formiranje znakova koriste se obično elektroluminescentni elementi, svetleći elementi ili svetleće diode.

Kod jedinica za govorni izlaz koje se korisniku obraćaju ljudskim glasom razlikuju se slučajevi kada ja ljudski glas prethodno

snimljen i slučajevi kada se na osnovu binarnog koda teksta vrši sinteza glasa radi saopštavanja korisniku poruka ili podataka.

Uređaji koji se koriste za pripremu izveštaja su najčešće štampači ali se mogu koristiti i mikrofilmski uređaji (naročito za arhiviranje), koordinatni crtači (ploteri) i drugi izlazni uređaji.

5.2.1 Monitor

Monitor, video-ekran ili displej predstavlja izlaznu jedinicu računara koja služi za vizuelno prikazivanje podataka u tekstualnom ili grafičkom obliku. Karakteriše ga trenutno prikazivanje bez memorisanja i danas, uz tastaturu, predstavlja najvažniji uređaj za neposrednu komunikaciju čoveka sa računarom. Kod većine monitora podaci se predstavljaju pomoću matrice tačaka, gde svaka tačka ima svoje koordinate, intenzitet svetlosti i boju.

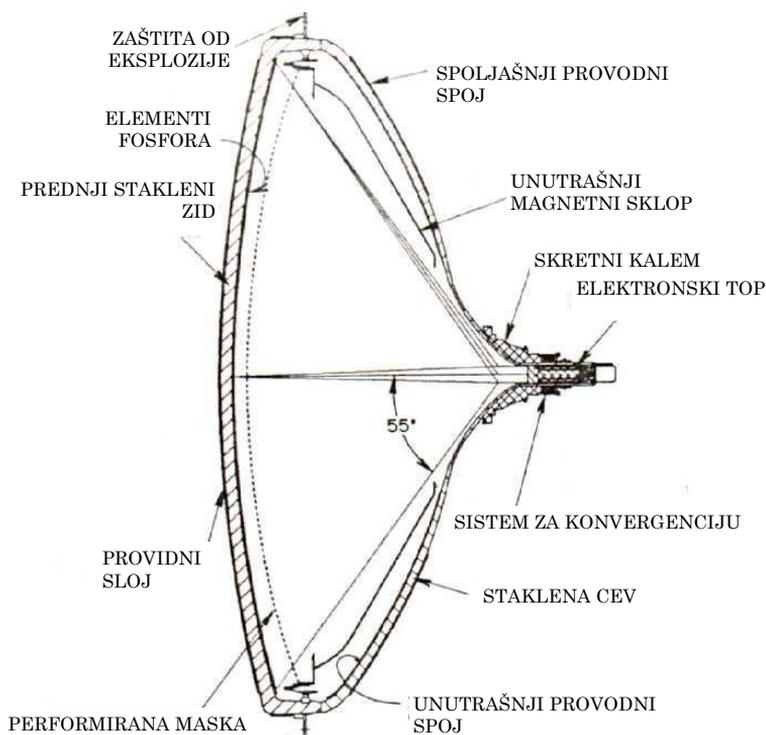
U zavisnosti od tehnologije prikaza koriste se:

- monitori sa katodnom cevi,
- monitori sa ravnim ekranom,
- monitori sa tečnim kristalima,
- elektroluminescentni monitori.

Monitori sa katodnom cevi (engl. CRT – Cathode Ray Tube) najviše su u upotrebi. Na Slici 5.3 prikazana je kolor katodna cev sa osnovnim komponentama.

Na mesto jednog elektronskog mlaza kao u crno-belim katodnim cevima, u kolor katodnim cevima postoje tri elektronska mlaza. Oni pobuđuju tri fosforna zrna, postavljena u vidu ravnostranog trougla (Slika 5.4), ili se nalaze u horizontalnoj ravni (Slika 5.5), koja čine jedan elemenat slike, piksel. Ova tri zrna predstavljaju svetlosne izvore, od kojih jedan emituje crveni, drugi zeleni, a treći plavi svetlosni fluks. Broj ovih grupa je vrlo veliki i iznosi preko 1 300 000. Pošto je površina elemenata dovoljno mala oko ne vidi pojedina zrna fosfora odvojeno, pa svetlosni fluks koji oko prima sa tri svetlosna izvora¹ meša aditivno.

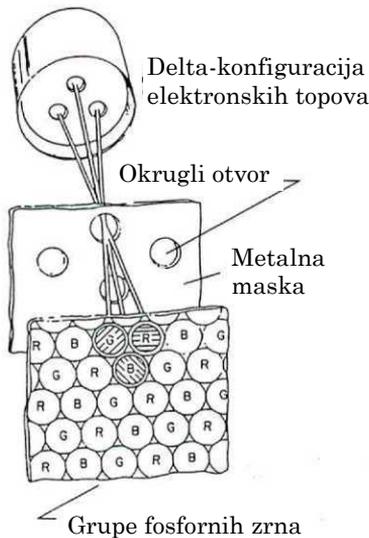
¹ Izabrani svetlosni izvori se nazivaju primarnim izvorima ili primarima.



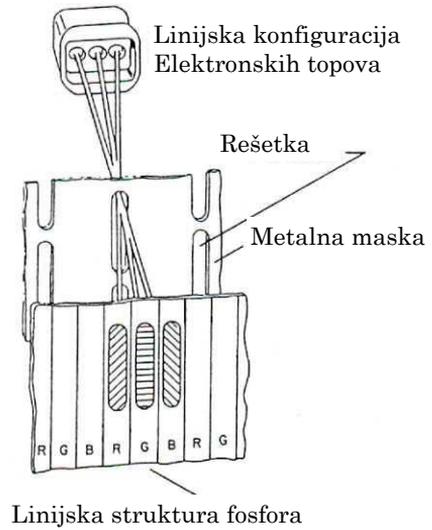
Sl. 5.3: Katodna cev sa perforiranom maskom

Ispred ekrana, na udaljenosti od 1 cm, nalazi se metalna maska debljine oko 0.15 mm, ko što se vidi na Slici 5.5. Na maski postoji po jedan mali otvor na spram svake pojedine grupe fosfornih zrna. Ova maska omogućava da, od tri elektronska mlaza svaki pobuđuje samo svoju vrstu fosfora u elementu. Na slikama 5.4 i 5.5 vidi se položaj tri elektronska mlaza pri eksitaciji jednog elementa slike, kako za katodnu cev sa delta topovima tako i za katodnu cev sa topovima u horizontalnoj ravni.

Pored uslova da pojedini mlazevi analiziraju samo samo po jednu vrstu fosfora, neophodno je da sva tri mlaza prolaze uvek kroz iste otvore, tj. da istovremeno pobuđuju fosforna zrna jednog elementa slike, piksela. Ovaj uslov, koji treba biti ispunjen za sve elemente slike, naziva se konvergencija. Idealnu konvergenciju nije moguće ostvariti u praksi, iako se za tu svrhu koriste prilično složeni postupci.



Sl. 5.4: Kolor-cev sa delta topovima



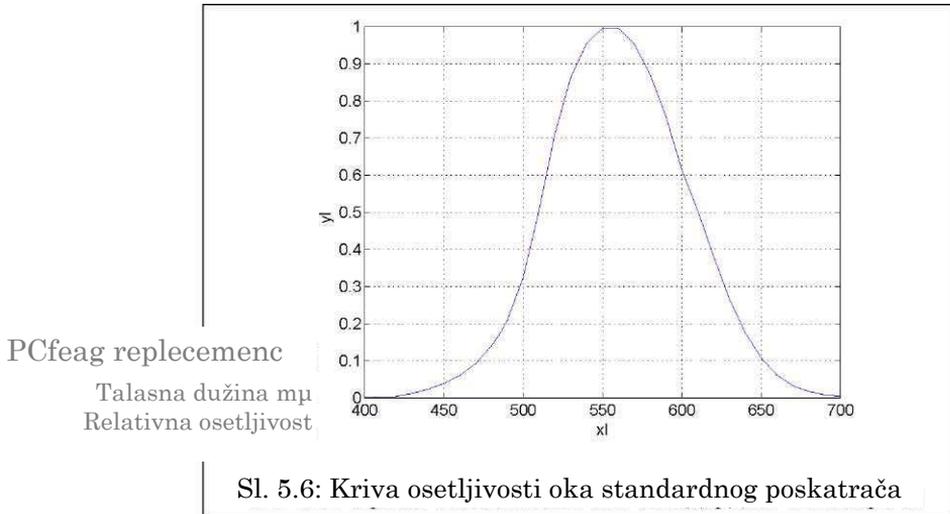
Sl. 5.5: Color-cev sa topovima u horizontalnoj ravni

Dakle, ova tri zrna predstavljaju primarne svetlosne izvore, od kojih jedan emituje crveni, drugi zeleni i treći plavi svetlosni fluks, a fluks elementa kao celine jednak je njihovoj aditivnoj smeši. Količine emitovanog svetlosnog fluksa pojedinih izvora zavise od veličine pobudnih signala dovedenih na ulaz katodne cevi (monitora). Kada se veličine ovih signala menjaju, a svetlosni fliksevi daitivno mešaju, dobija se svetlosni izvor čije se boje mogu menjati u širokim granicama.

Ako, pri tome, želimo da boja rezultatne svetlosti bude uvek jednaka željenoj boji, potrebno je znati kako treba menjati relativne jačine tri svetlosna izvora. Odgovor na ovo pitanje može se dobiti primenom jedne grane psihofizike, koja je poznata pod imenom kolorimetrija.

Pošto je svetlost fizička pojava inerpretirana psihološkim procesom, potrebno je poznavati mehanizam opažanja boja. Sistematskim merenjem je utvrđeno da, počevši od talasne dužine 400 nm, osetljivost oka raste i dostiže maksimum u koloni 555 nm, a zatim ponovo opada idući ka 700 nm. Normalizovanjem dobijenih rezultata ustanovljena je karakteristika relativne osetljivosti

oka u funkciji talasne dužine svetlosti prikazana na slici 5.6.



Što se pak tiče mehanizma opažanja boja postoji više teorija poznate po imenima istraživača koji su ih predložili i, izvesnoj meri, opravdali laboratorijskim eksperimentima. Najznačajnija teorija je Jang-Helmholcova.

Još početkom 19. veka engleski naučnik Tom Jang pretpostavio je da postoje tri vrste receptora u čovečijem oku od kojih je svaki osetljiv na širok ali različit opseg vidljivih radijacija. Osim toga pretpostavio je da su receptori maksimalno osetljivi na crvenu, zelenu i plavu boju. Kasnije je nemački fizičar Helmholtz prihvatio ovu hipotezu, modifikovao je i postavio na naučnu osnovu. Po njihovoj teoriji nervni impulsi sa izlaza receptora direktno odlaze u mozak noseći sa sobom informacije o kvalitetu svetlosti. Relativni odnos pobuda pojedinih receptorskih ćelija odeđuje boju, dok je osećaj sjanosti funkcija zbira pobuda.

Jedna osobina vida se odnosi na veličinu ugla pod kojim oko vidi obojenu površinu. Kada su obojene površine velike one obuhvataju veliki ugao. Na takvim površinama oko opaža sve fizički ostvarljive boje. U tom slučaju oko poseduje osobinu koja se naziva trihromatska osobina vida. Ako se površina smanji do veličine koja odgovara vidnom uglu od 10-15 minuta, vid gubi sposobnost opažanja nekih nijansi boja. Vid postaje dihromatičan. Za vrlo

male površine, ispod 10 minuta vidnog ugla, oko prestaje da oseća boje, dok nijanse sivog, veličine sjanosti površine i dalje oseća. Za vrlo male površine oko postaje oko postaje monohromatično.

Cilj kolorimetrije je da pruži prost i precizan način merenja i specifikacije svetlosti mereni iz izvora ili reflektovane sa nekog objekta. Ona ne određuje računskim putem vizuelni osećaj nego fizičku specifikaciju svetlosnog izvora koristeći trihromatske karakteristike vida²

Linearna priroda boja, koja je iskazana Grasmanovim zakonima, dopušta da se boje mogu predstavljati orjentisanim dužima, odnosno vektorima, i da se smeše boja mogu određivati primenom vektorske algebre.

U vektorskoj algebri uobičajeno je da se jedinični vektori označavaju simbolima j i k . Ako se za jedinične vektore svetlosnih primara usvoje oznake (R) , (G) i (B) , vektorska jednačina boje biće

$$S(5) = R(R) + G(G) + B(B). \quad (5.1)$$

Dakle, svetlost S je definisana količinama R , G i B odgovarajućih primara (R) , (G) i (B) , tj. smešom od R jedinica crvenog primara, S jedinica zelenog primara i V jedinica plavog primara. Smeša jednakih jedinica primara daje belu svetlost. Količina rezultantne svetlosti (5) je jednaka zbiru količina primarnih svetlosti tj.

$$S = R + G + B \quad (5.2)$$

Drugim rečima, ako sa TS označimo sjajnost boje C koju emituje katodna cev kolor-prijemnika, tada je

$$I_C = I_R + I_G + I_B \quad (5.3)$$

²Osnovne zakone kolorimetrije postavio je još pre stotvadeset godina, nemački profesor Hugo Grasman koristeći svoje i predhodne eksperimentalne i teorijske radove:

- Oko može razlikovati tri parametra svetlosti koji se mogu izraziti dominantnom talasnom dužinom, sjanošću i čistoćom.
- Boja dvokomponentne svetlosti se menja kontinualno ako se kontinualno menja jedna komponenta, a druga zadržava konstantnom.
- Svetlosti iste boje (tj. iste dominantne talasne dužine, čistoće i zasićenja) stvaraju identičan vizuelni efekat u smeši bez obzira na njihove spektralne karakteristike.

Što se tiče ukupne sjajnosti smeše ona je jednaka zbiru sjajnosti pojedinih izvora. na ovim zakonima je zasnovana trihromacka kolorimetrija.

gde je L_R sjajnost crvene boje, L_G sjajnost zelene boje i L_B sjajnost plave boje.

Na osnovu Grasmanovih zakona može se izvesti zaključak da je aritmetika u kolorimetrijskoj jednačini jednaka aritmetici običnih brojeva. Ako su, na primer, dve boje određene kolorimetrijskim jednačinama

$$C_1 = R_1(R) + G_1(S) + B_1(B)$$

i

$$C_2 = R_2(R) + G_2(G) + B_2(B)$$

rezultantna boja je

$$C = C_1 + C_2 = (R_1 + R_2)(R) + (G_1 + G_2)(G) + (B_1 + B_2)(B)$$

Komponente smeše dve boje jednake su zbiru komponenata pojedinih boja. Ova jednakost se ne menja ako se obe jednačine pomnože nekim konstantnim članom. To znači da se boja smeše neće menjati ako se sjanosti pojedinih izvora podjednako smanjuju ili povećavaju.

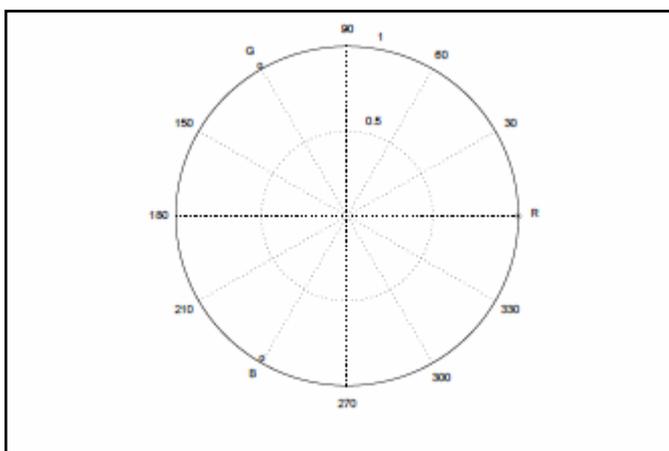
Za generisanje boja na ekranu monitora koristi se 8, 16 ili 24 bita za svaki piksel koje treba raspodeliti na primarne boje. Broj bita po pikselu određuje broj različitih boja, *kolor rezoluciju*, koji se može reprodukovati na ekranu monitora. Jedan 8-bitni monitor može reprodukovati isti broj boja kao i 24-to bitni displej, ali samo 256 boja istovremeno jer u tom slučaju koristi samo po dva bita za svaku primarnu boju. Sivo se reprodukuje u 256 nivoa. Kod 16-to bitnog displeja koristi se 5 bita za svaku primarnu boju, tako da se svaka boja može reprodukovati u 32 različita nivoa (2^5), što se kao konačni rezultat dobija 32 768 (2^{15}) različitih boja. Za reprodukciju sivog na raspolaganju stoje 32 nivoa. Konačno, 24-to bitni displeji reprodukuju crvenu, zelenu i plavu boju u 265 nivoa (2^8), te se na ekranu može reprodukovati 16 777 216 (2^{24}) različitih boja. Pošto se sivo dobija za $R = G = B$ to se i sivo reprodukuje u 256 nivoa.

Primer 5.1 *Aditivnim mešanjem crvene boje $FF0000_{16}$ i zelene $00FF00_{16}$ dobija se žuta boja $FFFF00_{16}$.*

Pored RGB kolor prostora za specifikaciju boja koristi se i HSL kolor prostor, koji je razvijen sa ciljem da aproksimiraju

način na koji ljudsko oko opaža boju. U HSL sistemu za specifikaciju boja, boje iste sjajnosti svrstavaju se u istu ravan u kojoj su njihova zasićenja i vrste boja prikazane u polarnim koordinatama, kao na Slici 5.7. Tačke na krugu predstavljaju razne vrste boja konstantnog zasićenja, a tačke na pravama označavaju uzorke konstantne boje ali raznog zasićenja. Zasićenje predstavljena je psihološkom veličinom koja pokazuje vizuelnu sposobnost procene relativne količine čiste hromatične vlićine o smeši boje.

U centru kruga nalazi se referentna bela svetlosat, a na obimu kruga nalaze se čiste hromatične boje, odnosno boje koje se nalaze u spektru sunčeve svetlostii.



Sl. 5.7: HSL koordinatni sistem. Tačke pa krugovima prikazuju boje jednakog zasićenja, a tačke na pravama boje jednake vrste.

Ukoliko su poznati *RGB* kolorimetrijski primari, parametri boje se u HSL kordinatnom sistemu se izračunavaju primenom sledećih jednačina.

Vrsta boje

$$H = \begin{cases} 60 \frac{G - B}{Max - Min} + 0 & \text{za } Max = R \\ 60 \frac{B - R}{Max - Min} + 120^\circ & \text{za } Max = G \\ 60 \frac{R - G}{Max - Min} + 240^\circ & \text{za } Max = B \end{cases} \quad (5.4)$$

Zasićenje boje

$$H = \begin{cases} 60 \frac{Max - Min}{Max + Min} & \text{za } L \leq \frac{1}{2} \\ 60 \frac{Max - Min}{Max - Min} & \text{za } L \geq \frac{1}{2} \end{cases} \quad (5.5)$$

Sjanost boje

$$L = \frac{1}{2}(Max + Min). \quad (5.6)$$

U izrazima (5.4), (5.5) i (5.6) Max i Min predstavljaju najveći, odnosno najmanji element u skupu $0 \leq R, G, B \leq 1$ kolorimetričkih primara.

Vrsta boje se izražava u stepenima $0 < H < 360^\circ$, dok se vrsta boje i zasićenje boje leže u opsegu $0 \leq S \leq 1$ i $0 \leq L \leq 1$ i mogu se izražavati u procentima. Međutim, za predstavljanje boja u računaru treba imati u vidu da je $0 \leq R, G, B \leq 255$, i da je usvojeno $0 \leq H, S, L \leq 240$. Veza između ova dva kolor prostora se može jednostavno uspostaviti množenjem S i L sa 240, a H sa količnikom 240/360.

Primer 5.2 Zelena boja je specificirana sledećim vektorom $[R \ G \ B] = [0.5 \ 1.0 \ 0.5]$. Primenom izraza (5.4), (5.5) i (5.6) dobijaju se sledeće vrednosti za vrstu, zasićenje i sjajnost zelene boje $[H \ S \ L] = [120^\circ \ 1.0 \ 0.75]$, jer je $Max = 1.0$ i $Min = 0.5$.

Ligitalni ekvivalent zelene boje specificirane u ovom primeru, u R, G, B kolorimetrijskom prostoru, je $[128, 255, 128]$. Odgovarajući digitalni ekvivalent ove boje u H, S, L prostoru je $[80, 240, 180]$

Princip rada ovih monitora je sličan kao kod kolor televizijskih prijemnika.

Monitori sa ravnim ekranom - plazmatički ekrani (engl. plasma screens - plazma, materija u jonizovanom stanju) sačinjeni su od mreže elektroda smeštenih u tanko stakleno kućište. Prostor u staklenom kućištu je ispunjen gasom. Ako se na pojedine elektrode dovede odgovarajući napon, doći će do jonizacije gasa na svim onim mestima gde se dve elektrode ukrštaju. Ionizacija vazduha proizvodi svetlost.

Monitori sa tečnim kristalom - LCD (engl. Liquid Crystal Display - displej sa tečnim kristalom) se karakterišu izuzetno malom

potrošnjom električne energije, malom težinom i proizvoljno malim dimenzijama. Najčešće se koriste kod prenosnih računara i kalkulatora.

Povezivanje računara sa monitorom, kao izlaznom jedinicom, ostvaruje se preko posebnog elektronskog sklopa - videoadaptera. Videoadapter se u vidu specijalne ploče elektronike (video kartice) smešta u jednu od pozicija (slotova) za proširenje ili se već nalazi na osnovnoj ploči računara.

5.2.2 Štampači

Periferni uređaj, koji se najčešće koristi, pored tastature i monitora, jeste štampač (engl. printer). On predstavlja standardnu izlaznu jedinicu koja izlazne podatke iz računara prikazuje u obliku najpogodnijem za čoveka - na papiru. U zavisnosti od tehnologije i načina rada, postoji više vrsta štampača.

Prema ciklusu štampanja, štampači mogu biti: - serijski - štampaju jedan znak u jednom ciklusu, - linijski - u jednom ciklusu štampaju jedan red, - stranični - pripremaju, a zatim štampaju celu stranicu ođednom.

Prema načinu štampanja, mogu se podeliti u dve grupe:

- elektromehanički ili udarni štampači (engl. impact printers) znakove utiskuju mehaničkim dodiranjem - udaranjem u papir preko trake natopljene specijalnom bojom. Linijski elektromehanički štampači se dele na štampače sa dobošem (valjkom) i štampače sa lancem. Serijski elektromehanički štampači mogu biti sa polugama (kao električne pisaće mašine), kuglom za štampanje, lepezom i matrični (igličasti);

- nemehanički ili bezudarni štampači (engl. nonimpact printers) rad zasnivaju na nemehaničkim principima: elektrostatičkom, termičkom, piezo-električnom, itd. Postoji više vrsta ovih štampača: fotoštampači, termički štampači, elektrostatički štampači, kserografski štampači, štampači sa ubrizgavanjem mastila i laserski štampači.

Brzina serijskih štampača kreće se od 30 do 300 znakova u minutu, a paralelnih - od 150 do 2 500 redova u minutu. Broj znakova u redu varira od 60 do 160, a najčešće je 60 - 70 znakova za A4 format papira i 120 - 136 znakova za papir širine 14 inča (35,6

cm). Brzina laserskih štampača iznosi 4-10 stranica u minutu. Papir se u štampač može uvoditi na dva načina: list po list, ili na tzv. beskrajnom formularu.

Matrični štampači

Matrični štampači predstavljaju uređaje iz grupe serijskih elektro-mehaničkih štampača koji se najviše koriste. Osnovni delovi matričnog štampača su: - glava za štampanje sa iglicama i elektromagnetima, - koračni motor za pokretanje papira, - koračni motor za pokretanje glave za štampanje, - mikroprocesor koji upravlja radom celog uređaja, - generator znakova u ROM ili EPROM memoriji, - interna prihvatna memorija - bafer, - upravljački programi smešteni u posebnoj memoriji, - traka natopljena bojom i smeštena u plastičnu kasetu.

Karakterističan deo predstavlja glava za štampanje u kojoj se nalazi 5, 9, 24 ili 48 iglica. Najviše su u upotrebi štampači sa 9 i 24 iglice. Iglice su postavljene na tačno određenom međusobnom rastojanju sa vrhovima prema papiru. Glava za štampanje se kreće po osovini postavljenoj pod pravim uglom u odnosu na kretanje papira i pomera se duž celog štampanog reda. Iglice glave se pomoću elektromagneta aktiviraju, udaraju u traku sa bojom i ostavljaju otisak na papiru.

Znakovi se formiraju pomoću tačaka raspoređenih u matrici. Sto je broj tačaka od kojih se formiraju znakovi veći, to je i kvalitet štampe veći. Pošto nemaju fizički formirane znakove, već se oni sastavljaju od određenog broja otisnutih tačaka, ovi štampači omogućavaju štampanje i teksta i grafike.

Karakteristike matričnih štampača su: - jednostavnost konstrukcije, - niska cena uređaja, - male dimenzije, - brzina štampanja od 50 do 300 znakova u sekundi, - niska cena po štampanoj stranici, - mogu da koriste različite vrste papira, - mogućnost formiranja nestandardnih znakova.

Glavni nedostaci matričnih štampača su: velika buka pri radu, mnogo mehaničkih delova i nizak kvalitet štampanja (jer znaci često nisu kompaktni). Zbog toga ih u novije vreme sve više potiskuju štampači sa ubrizgavanjem mastila i laserski štampači.

Linijski štampači

Linijski ili paralelni štampači štampaju ceo red znakova u jednom vremenskom ciklusu. Najčešće se koriste dva tipa ovih uređaja:

- štampači sa valjkom (dobošem) ili cilindrični štampači,
- štampači sa lancem.

Štampači sa valjkom se zasnivaju na principu neprestanog okretanja nosača znakova metalnog valjka. Znakovi su raspoređeni na omotaču, a jedan znak se ponavlja po celoj dužini valjka. Osnovni delovi linijskog štampača su:

- valjak (doboš) sa ugraviranim znacima (od 80 do 160 znakova u jednom redu);
- kodni disk koji prepoznaje koji se red znakova na valjku trenutno nalazi ispod udarnih čekića;
- komparator koji određuje trenutak kada treba odštampati znak;
- traka natopljena bojom - ribon;
- udarni elektromagnetni čekići koji preko trake otiskuju određeni znak na papir;
- mehanizam za transport papira;
- bafer-memorija za prihvatanje podataka iz računara;
- elektronski deo za pronalaženje i otiskivanje određenog znaka.

Valjak sa ugraviranim znacima se okreće velikom brzinom. Kada naiđe znak koji treba odštampati, on se u tom momentu otiskuje na svim potrebnim mestima u redu. Trenutak kada treba da se aktiviraju elektromagnetni čekići određuje se pomoću kodnog diska, optičkog sistema i komparatora znakova.

Štampači sa lancem rade na sličnom principu. Oni sadrže lanac sa ugraviranim simbolima koji se neprestano kreće velikom brzinom.

Linijski štampači se odlikuju velikom brzinom štampanja (do 2 500 redova u minutu), jasnim otiskom i robusnom konstrukcijom. Glavni nedostatak je nemogućnost prikazivanja grafike i otežano

menjanje skupa raspoloživih znakova, kao i visoka cena. Uglavnom se koriste u većim računskim centrima gde je potrebno za što kraće vreme odštampati ogromne količine podataka.

Štampači sa ubrizgavanjem mastila

Tehnologija štampanja sa ubrizgavanjem mastila (engl. Ink-Jet) se relativno dugo razvijala. Posle višegodišnjeg eksperimentalnog korišćenja, krajem osamdesetih godina, došlo je do masovne proizvodnje štampača zasnovanih na njoj. Štampanje sa ubrizgavanjem mastila se i danas neprestano usavršava, što donosi sve pouzdanije i jeftinije modele štampača.

Osnovni deo štampača sa ubrizgavanjem mastila predstavlja glava za pisanje. Na njoj se nalaze kapilarne cevčice kroz koje prolazi mastilo i ispisuje podatke na papiru. Glava se na svom nosaču pomera duž štampanog reda.

Postoje tri varijante ovih štampača u zavisnosti od toga kakvo mastilo koriste i kako ga prebacuju iz rezervoara na papir: štampači sa stalnim mlazom, štampači sa tečnim mastilom i štampači sa amorfnim mastilom.

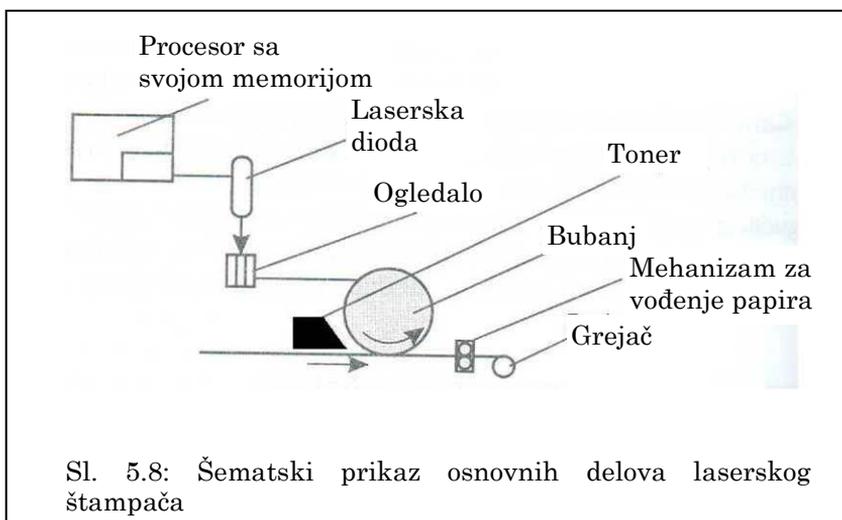
Karakteristike štampača sa ubrizgavanjem mastila su:

- dobar kvalitet štampe,
- relativno niska cena,
- bešumnost u radu,
- mogućnost prikazivanja grafike,
- jednostavnost konstrukcije,
- mala potrošnja energije,
- male dimenzije i težina,
- mogućnost korišćenja u prenosnim računarskim sistemima,
- jednostavnije štampanja u boji koje se ostvaruje "u jednom prolazu".

Nedostaci štampača sa ubrizgavanjem mastila su slaba izdržljivost pri intenzivnijem korišćenju i relativno visoka cena specijalnog mastila.

Laserski štampači

Najkvalitetniji i najsloženiji su laserski štampači (Sl. 5.8). Oni su i najskuplji. Pripadaju grupi straničnih nemehaničkih štampača. Postižu rezolucije od 300x300 do 1 200x1 200 tačaka po inču.



Princip rada laserskog štampača je sličan kao i kod aparata za fotokopiranje. Zasniva se na materijalu (selen ili neki drugi) koji, kada se osvetli, postaje naelektrisan. Ovako naelektrisan, ovaj materijal privlači toner koji se kasnije utiskuje na papir. Aluminijski valjak je širine papira na kome se štampa i presvučen je ovim materijalom. Laserski zrak je usmeren prema centru valjka. Ima ulogu da osvetli ona mesta na kojima treba da bude otisak. Šestougona prizma, koja stalno rotira, skreće laserski zrak po celoj dužini valjka. Jedna stranica prizme usmerava laserski zrak duž jednog reda (linije). Kada se nova stranica prizme nađe ispred zraka, usmerava ga na početak reda. Valjak pri obrtanju prolazi kroz toner koji se lepi za valjak na onim mestima koja su obrađena laserskim zrakom. Kada se valjak obrne za ceo krug, ispisu se sve linije i dobija se slika cele stranice.

Pored valjka, na kojem je formirana slika stranice, na vrlo malom rastojanju prolazi papir, ali ga ne dodiruje. Naelektrisani toner prelazi na papir formirajući sliku. Papir zatim prolazi kroz sistem za sušenje koji trajno učvršćuje toner za-

5.3. Pitanja za proveru znanja

grevajući ga do 200°C. Posle štampanja jedne stranice valjak se očisti i spreman je za novu stranicu.

Laserski štampači se odlikuju izuzetno visokim kvalitetom otiska, ne zahtevaju specijalni papir i štampaju relativno brzo (oko 4 strane u minutu). Loše strane su :

- relativno visoka cena,
- relativno velike dimanzije i težina,
- skupo održavanje.

Uglavnom se koriste na mestima gde se zahteva visok kvalitet štampe i pouzdanost, ali se zbog dimenzija i težine ne mogu koristiti u prenosnim računarskim sistemima.

5.3 Pitanja za proveru znanja

1. Kako se može proveriti koji ulazno-izlazni uređaji su priključeni, kao i da li su ispravno priključeni, na centralnu jedinicu računara? Podrazumeva se da je operativni sistem Windows.
2. Nabrojati ulazno-izlazne uređaje čije karakteristike se mogu podešavati korišćenjem Control Panel-a.
3. Dve boje, koje mešanjem stvaraju belu svetlost nazivaju se komplementarne boje. Odrediti boju koja je komplementarna plavoj boji (0000FF).
4. Da li su boje AA1155 i 55EEAA komplementarne?
5. Pokazati da se mešanjem crvene boje $K = RR$ sa zelenom bojom $G = FF$ (plava je 00) dobija žuta boja³. Sta će se desiti ako smeši boja dodamo i plavu boju $B = 80$?
6. specificirati vrstu boje, zasićenje i sjanost purpurne boje u HSV kolorimetrijskom prostoru ako je boja specificirana u RGB kolorimetrijskom prostoru.

³Za geperisanje boja koristiti program za crtanje Paint, a zatim u padajućem mepiju izbrati taster Colors, i pa kraju Define Custom Colors >>. Upisivanjem odgovarajućih pumeričkih vredposti u polja: Red, Green i Blue polje ColorSolids prikazuje geperisapu boju.

- a) $R = FF, G = 0$ i $V = FF$
- b) Da li će se vrsta boje promeniti ako purpurnoj boji dodamo zelenu boju $S = 40$?
7. Boje $A = [35, 80, 150]$ i $B = [135, 240, 60]$ su specificirane u HSL kolor prostoru. Odrediti smešu boja A i B u HSL kolor prostoru ako su koordinate boja A i B zadate u dekadnom ko-ordinatnom sistemu⁴.
8. Koje karakteristika miša se mogu podešavati sa korišćenjem **Control Panela** -a Windows operativnog sistema?

⁴Potrebno je boje A i V transformisati u RGB kolor prostor, a zatim sabrati odgovarajuće koordinate. Ukoliko je neka koordinata smeše veća od 255, potrebno je izvršiti normalizaciju tako da najveća koordinata ne prelazi 255. Za transformaciju boja iz RGB kolor prostora u RGB kolor prostor i obratno, može se koristiti već pomenuti program za crtanje Paint..

Glava 6

Softver računara

Računarski sistem sastoji se od dve osnovne komponente koje se nazivaju hardver i softver. Ove komponente međusobno su tesno povezane i ne mogu funkcionisati jedna bez druge.

Termin softver koristi se za one komponente računarskog sistema koje nisu fizičke. U najopštijem značenju termin softver ili programska podrška, nasuprot terminu hardver, označava sve programe koji se mogu koristiti na nekom računarskom sistemu. Ovi programi omogućavaju da hardver funkcioniše pravilno i efikasno.

Softver se deli u dve grupe: sistemski softver i aplikacioni softver. Sistemski softver čine svi programi koji na neki način pomažu korisnicima da upotrebljavaju računar. To su programi koji upravljaju radom raznih delova računarskog sistema i automatizuju proces razvoja i korišćenja programa. Aplikacioni softver sadrži programe namenjene za rešavanje konkretnih problema korisnika.

Sa stanovišta rešavanja konkretnih problema korisnika, odnosno tzv. aplikacionih problema, računarski sistem se može posmatrati kao hijerarhijski organizovan skup uređaja (hardvera) i programa (softvera) koji zajednički ostvaruju obradu podataka. Ovaj hijerarhijski skup komponenata predstavlja tzv. konceptualni model računarskog sistema.

Operativni sistem predstavlja skup programa kojima se organizuje rad računara, efikasno korišćenje svih resursa računara,

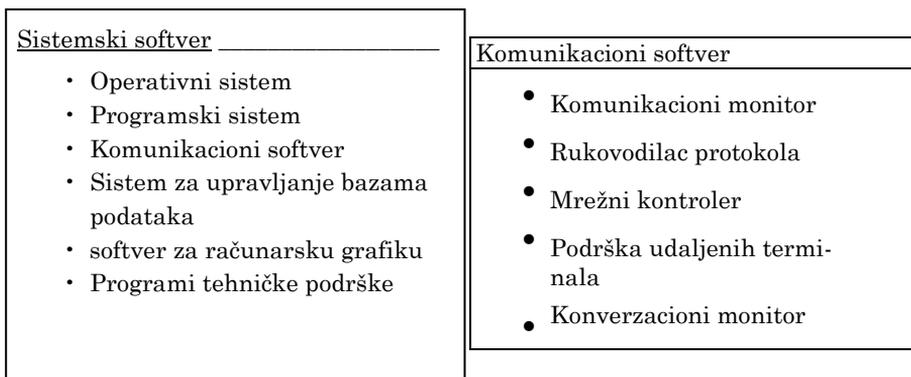
kao i upravljanje izvršavanjem programa. Osnovna jedinica operativnog sistema je posao, koji se sastoji od svih programa kojima se rešava neki problem.

Programski jezici i softverski alati služe korisnicima za pripremu i proveru ispravnosti aplikacionih programa. Aplikacioni programi rešavaju konkretne probleme korisnika. Korisnički interfejs čine programi, razni ulazno-izlazni uređaji i druga sredstva koja omogućavaju da korisnik na što lakši način komunicira sa računarom.

6.1 Sistemski softver

Sistemski softver sadrži programe koji se odnose na organizaciju i upravljanje radom računarskog sistema i automatizaciju procesa razvoja i održavanja programa. Ovi se programi nazivaju i upravljački programi.

Komponente koje ulaze u sistemski softver su; operativni sistem, programski sistem, komunikacioni softver, sistem za upravljanje bazama podataka i softver za računarsku grafiku, slika 6.1. Ponekad se ovde svrstavaju i programi tehničke podrške koji sadrže a) test programe za periodičnu preventivnu proveru pravilnog funkcionisanja blokova, jedinica, uređaja i računara u celini, i b) dijagnostičke programe kojima se lokalizuju mesta neispravnosti.



Sl. 6.1: Komponente sistemskog i komunikacionog softvera.

Programiranje koje obuhvata projektovanje, realizaciju i

Glava 6. Softver računara

održavanje sistemskih programa, tj. programa koji obezbeđuju uslove za korišćenje računara, naziva se sistemsko programiranje. Nasuprot sistemskom stoji aplikaciono programiranje čija je suština u projektovanju, realizaciji i održavanju programa koji služe za rešenje konkretnih problema korisnika.

Prve dve komponente sistemskog softvera, tj. operativni sistem i programski sistem biće kasnije detaljnije razmotrene.

Komunikacioni softver sadrži programe čija je funkcija upravljanje komunikacijom računara sa udaljenim terminalima ili drugim računarima.

Osnovne komponente komunikacionog softvera prikazane su na lici 6.1 Osnovni zadatak komunikacionog softvera je pre svega upravljanje funkcionisanjem i komunikacijom svih uređaja povezanih u računarsku mrežu, a posebno: upravljanje uzajamnim delovanjem procesa u mreži, upravljanje samom mrežom, upravljanje mrežnim službama. Prva i treća grupa zadataka upravljanja karakteristična je za bilo koje mreže, dok je druga grupa karakteristična za distribuirane sisteme.

Sistem za upravljanje bazama podataka obezbeđuje: kreiranje i vođenje baze podataka, centralizovano upravljanje podacima, smanjenje redundanse podataka, mogućnost otklanjanja protivrečnosti, celovitost baze podataka, zajedničko korišćenje podataka iz različitih baza podataka, pristup podacima iz različitih korisničkih programa, nezavisnost podataka. On takođe sadrži sredstva za definisanje šeme baze podataka i operacije koje se mogu koristiti za transformaciju baze podataka.

Sistemski softver koji obezbeđuje rad sa bazama podataka naziva se sistem za upravljanje bazama podataka. Nastao je kao rezultat razvoja sistema za rad sa datotekama kao dela svakog operativnog sistema koji je obezbeđivao skladištenje (memorisanje) datoteka na spoljnim nosiocima i pristup zapisima datoteka.

Imajući u vidu prirodu podataka, kao i njihovu količinu, najveća efikasnost u komuniciranju između čoveka i računara postiže se korišćenjem slike ili crteža kao nosioca informacija, odnosno korišćenjem računarske grafike. Ova efikasnost posebno dolazi do izražaja u projektovanju pomoću računara.

Računarska grafika je deo sistemskog softvera koji služi za

crtanje slika i prikazivanje grafičkih podataka. Sadrži metode za unos, obradu (transformaciju i editovanje) i prikaz grafičkih objekata (slika i crteža) pomoću računara.

Ako se grafička informacija može menjati kao odgovor na ulaze koji dolaze od operatera sa terminala govori se o interaktivnoj računarskoj grafici. Među pogodne ulazne uređaje za unos grafičkih podataka spadaju miš, grafička tabla (digitalizator) i svetlosno pero. Izlaz se može dati preko standardnog alfanumerickog ili grafičkog video terminala ili kao trajan zapis preko stampaća ili crtača.

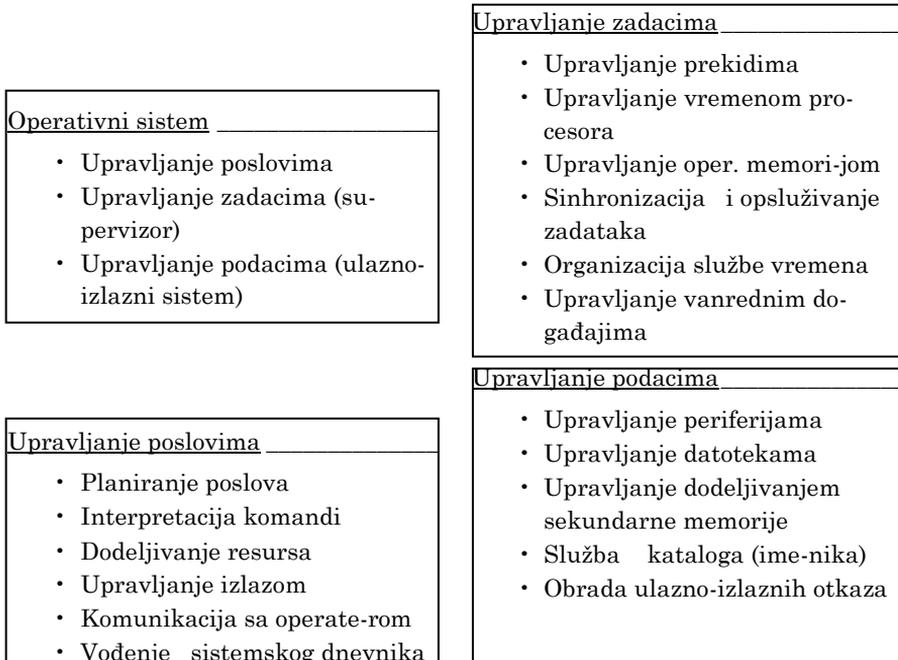
Osnovne funkcije programa za računarsku grafiku su: rad sa alfanumerickim ili grafičkim ekranima, grafičke metode pristupa, graficko programiranje, koriscenje ekrana kao konzole.

Računarska grafika se primenjuje u mnogim oblastima. Najčešće primene su: programiranje interfejsa sa korisnikom, priprema raznih crteza, konstruisanje u mašinstvu, konstruisanje integrisanih i štampanih kola, konstruisanje u građevinarstvu, izrada mapa i urbanistickih planova, izrada i kontrola upravljačkih traka za numerički upravljane masine, izdavaštvo (profesionalno i stono), i dr.

6.1.1 Operativni sistem

Operativni sistem predstavlja skup programa kojima se organizuje rad računara, efikasno korišćenje svih resursa računara, kao i upravljanje izvršenjem računarskih programa. Operativni sistem je deo sistemskog softvera koji je najbliži hardveru računara. On predstavlja osnovnu pomoć u organizaciji rada i efikasnom korišćenju hardvera. Svi programi operativnog sistema mogu se podeliti u tri relativno nezavisne celine. Dalje se svaka od ovih celina prema funkcijama deli kao što je prikazano na slici 6.2

Skup programa kojima se rešava neki problem korisnika organizuje se kao posebna celina i naziva se posao. Posao predstavlja osnovnu jedinicu sa kojom manipuliše operativni sistem. Pri planiranju operativni sistem od poslova obrazuje posebne manje radne celine - zadatke ili procese. Svaki zadatak može nezavisno konkurisati za dobijanje bilo kog resursa računara. O svakom zadatku operativni sistem poseduje i vodi određene upravljačke



Sl. 6.2: Komponente i funkcije operativnog sistema

informacije.

Postoji veliki broj operativnih sistema od kojih su posebno najrašireniji operativni sistemi za personalne računare i radne stanice, kao što su: DOS, WINDOWS, OS/2, UNIX i drugi.

Jedna od najvažnijih funkcija operativnog sistema jeste upravljanje resursima računarskog sistema. Upravljanje resursima ima uticaja na strukturu skoro svih komponenata operativnog sistema.

Računarski sistem poseduje pre svega sledeće resurse: vreme centralnog procesora, operativna memorija, ulazno-izlazni uređaji, datoteke i softver (programi). Upravljanje resursima sastoji se u dodeljivanju tih resursa programima koji konkurišu za njihovo dobijanje. Ovde ulaze raspoređivanje zadataka (procesa), dodeljivanje operativne memorije, dodeljivanje ulazno-izlaznih uređaja i upravljanje korišćenjem programskih resursa. Naime, potreba za upravljanjem javlja se zbog konkurencije za dobijanje

resursa. Programi konkurišu za dobijanje vremena centralnog procesora, za korišćenje pojedinih ulazno-izlaznih uređaja, a takođe za dobijanje operativne i spoljne (sekundarne) memorije. Kriterijum za ocenu kvaliteta upravljanja resursima je efikasnost korišćenja raspoloživih resursa, prioritet zadataka koji konkurišu, zahtevi za vremenom reakcije kod rada u realnom vremenu, kao i logička usaglašenost rešenja.

Vreme centralnog procesora raspodeljuje se među različitim suparničkim programima preključivanjem (prebacivanjem) zadataka prema određenim pravilima. Tako npr., kada je potrebno učitati nove podatke tekući zadatak se zaustavlja da sačeka ulazno-izlazni prenos, a procesor se dodeljuje drugom zadatku. Operativni sistem i u drugim situacijama može prekinuti tekući zadatak, npr. ako je isteklo dodeljeno vreme, i aktivirati drugi zadatak.

Potreba za upravljanjem operativnom memorijom javlja se zbog toga što obično nije moguće sve aktivne programe i sve podatke smestiti istovremeno. Upravljanje spoljnom memorijom posledica je potrebe da se rezerviše ili ažurira prostor za smeštanje datoteka i da se upravlja prenosom podataka.

Upravljanje ulazom-izlazom potrebno je zbog višeprogramskog rada kada mnogi programi zahtevaju korišćenje perifernih uređaja. Osim toga ulazno-izlazni prenos se odvija nezavisno i paralelno sa radom centralnog procesora, ali dodeljivanje uređaja, priprema informacija za kontroler, aktiviranje prenosa i obrada prekida po završetku prenosa zahteva upravljanje od strane operativnog sistema.

Upravljanje programima potrebno je zbog toga što svi programi koji su raspoloživi korisniku računarskog sistema takođe predstavljaju resurs. U svakoj aplikaciji mogu se pozivati određeni potprogrami iz određene biblioteke, uslužni programi ili drugi programi radi korišćenja u toj aplikaciji. Osim toga, neki programi mogu se koristiti za više zadataka. Operativni sistem čuva informacije o stanju svakog programa, dostupnosti, razmeštaju, korišćenju i dr.

Funkcija operativnog sistema može se posmatrati sa dve tačke gledišta:

1. korisničke, i

2. administrativne.

Sa korisničke tačke gledišta funkcija operativnog sistema je da olakša (ubrza) dobijanje rešenja problema koje interesuje korisnika, pružajući mu pritom raznovrsne usluge. S administrativne tačke gledišta, funkcija operativnog sistema je da obezbedi efikasno korišćenje resursa računara.

Još jedna moguća, tzv. hijerarhijska dekompozicija operativnog sistema je podela na sledeće delove koji predstavljaju slojeve (nivoje) operativnog sistema (od nižih ka višim): jezgro, upravljanje operativnom memorijom, upravljanje ulazno-izlaznim uređajima, upravljanje podacima (datotekama), planiranje i evidencija i interpretacija komandnog jezika.

Jezgro operativnog sistema obezbeđuje upravljanje sistemom prekida i obradu prekida, planiranje zadataka (procesa) operativnog sistema, manipulaciju sa zadacima (formiranje zadatka, završavanje zadatka i sl.) i komunikaciju između zadataka.

Upravljanje operativnom memorijom obavlja sledeće funkcije: realizacija određene strategije dodeljivanja memorije, samo dodeljivanje memorije i realizacija određene strategije oslobađanja memorije.

Na nivou upravljanja uređajima realizuju se sledeće funkcije: obezbeđenje nezavisnosti programa od tipa uređaja, obezbeđenje efikasnog rada uređaja, realizacija određene strategije dodeljivanja uređaja, samo dodeljivanje uređaja i realizacija određene strategije oslobađanja uređaja.

Upravljanje podacima treba da obezbedi softverska sredstva za organizovanje i pristupanje podacima na način koji odgovara korisniku računarskog sistema. Na ovom se nivou realizuju sledeće funkcije: formiranje i brisanje osnovnih struktura podataka (datoteka), čitanje iz datoteka i upis u datoteke, upravljanje sekundarnim memorijskim prostorom, obezbeđenje uslova za simboličko obraćanje datotekama, zaštita podataka od namernog ili nenamernog uništenja, zaštita podataka od neovlašćenog pristupa i korišćenja, deoba datoteka između više poslova (korisnika).

Planiranje se sastoji u uvođenju novih poslova u sistem i određivanju poretka u kojem će se oni izvršavati. Funkcije

koje se realizuju u okviru planiranja su: izbor novog posla za izvršenje, dodeljivanje prioriteta poslovima, realizacija strategije dodeljivanja resursa. U realizaciji evidencije i kontrole resursa osnovne su sledeće funkcije: ograničenje pristupa resursima, ograničenje pristupa sistemu (npr. nekim klasama korisnika može se uskratiti pristup ako zahtevaju mnogo resursa), vođenje računovodstvene evidencije za korisnike, ispostavljanje računa korisnicima za potrošene resurse i dr.

Komandni jezik omogućava uspostavljanje veze između korisnika i sistema i korišćenje resursa. Ova veza se ostvaruje pomoću interpretatora komandnog jezika operativnog sistema kod interaktivnih sistema ili jezika za upravljanje poslovima (engl. Job Control Language) kod sistema paketne obrade.

Upravljanje poslovima

Upravljanje poslovima podrazumeva odabiranje određenog posla i njegovo izvršavanje. Funkcije upravljanja poslovima su:

- planiranje poslova,
- interpretacija komandi,
- dodeljivanje resursa,
- upravljanje ulazom-izlazom,
- komunikacija sa operaterom i
- vođenje sistemskog dnevnika.

Planiranje poslova i dodela resursa se različito izvršavaju u zavisnosti od načina obrade podataka u računaru (režima rada):

- kod jednoprogamskog režima rada upravljanje poslovima se svodi na i formiranje redosleda izvršavanja poslova. Ukoliko se neki složeni program sastoji od više objektnih modula (delova), onda se određuje koji će modul biti u određenom trenutku u operativnoj memoriji, a koji će ostati u sekundarnoj.
- kod višeprogamskog režima rada prispeli poslovi formiraju red čekanja, koje, zatim, poseban program - planer poslova odabira po određenom, već zadatom kriterijumu.

Glava 6. Softver računara

Odabrani poslovi postaju aktivni. Kakav će prioritet dobiti određeni posao zavisi od cilja koji se želi postići. Ciljevi mogu biti različiti. Najčešće je potrebno postići što veću propusnu moć računara (izvršavanje što većeg broja operacija u što kraćem vremenskom periodu) ili ostvariti što kraće vreme odziva (minimizacija vremena od prihvatanja programa do dobijanja rezultata).

Upravljanje ulazom-izlazom je povereno posebnom modulu operativnog sistema koji obavlja sledeće funkcije:

- evidentira i nadgleda sve ulazno-izlazne uređaje bez obzira na to da li su trenutno aktivni ili ne. Operativni sistem u te svrhe stalno ažurira posebne - statusne datoteke koje mu omogućavaju kontrolu nad velikim brojem ulazno-izlaznih uređaja;
- upravlja dodeljivanjem ulazno-izlaznih uređaja pojedinim zadacima obrade;
- oslobađa uređaj po završenom zadatku (kad prestane potreba za njim).

Interpretacija komandi komandnog jezika podrazumeva:

- analizu svih operatora komandnog jezika,
- otkrivanje eventulanih grešaka,
- prihvatanje operatora i
- izvršavanje dejstva operatora.

Programi za dodeljivanje resursa ulaze u sastav sistema programa za upravljanje zadacima (supervizor), ali su na određeni način namenjeni upravljanju poslovima. Osnovne funkcije koje se ovde ostvaruju su:

- preraspodela perifernih uređaja koji se oslobađaju,
- preraspodela operativne memorije po završetku programa,
- preraspodela ključeva zaštite memorije,
- proveru raspodele operativne memorije pre početka programa,
- formiranje oblasti memorije,

- priprema za punjenje programa u operativnu memoriju i
- prenošenje upravljanja na program.

Funkcija operativnog sistema: komunikacija sa operaterom predstavlja osnovu za korisnički interfejs računara. Najčešće se komunikacija sa operaterom (korisnikom) ostvaruje na jedan od dva načina:

- zadavanjem komandi operativnog sistema sa komandne linije i
- korišćenjem grafičkog korisničkog interfejsa (pomoću menija i ikona).

Savremeni operativni sistemi sve više koriste ovaj način komunikacije, koristeći, pored tastature, mnoge druge uređaje za ručno unošenje podataka (miš, digitalizator, miš-olovka itd.), kao i monitore u boji za prikaz.

Upravljanje zadacima (supervizor)

Upravljanje zadacima predstavlja komponentu operativnog sistema namenjenu upravljanju hardverskim resursima računara. Sadrži sledeće funkcije:

- upravljanje prekidima,
- upravljanje vremenom procesora,
- upravljanje operativnom memorijom,
- sinhronizacija i opsluživanje zadataka,
- organizacija službe vremena,
- upravljanje vanrednim događajima.

Poznato je da procesor često radi sa prekidima. Prekid predstavlja odgovor na asinhroni ili sinhroni događaj u računaru. Prekidi se koriste radi povećanja efikasnosti, pouzdanosti i sinhronizacije elemenata procesora. Različiti načini rada operativnih sistema zahtevaju mehanizme za generisanje čitavog niza prekidnih signala da bi se mogli ostvariti. Isto tako, da bi se obezbedila pouzdanost sistema, ukoliko to nije regulisano softverski u okviru operativnog sistema, koristi se poseban hardver

Glava 6. Softver računara

koji ispituje da li su otkazali pojedini delovi, a ako se otkaz otkrije, procesor se o tome obaveštava posebnim prekidnim signalom.

Funkcija programa operativnog sistema za upravljanje prekidom sadrži sledeće aktivnosti:

- otkrivanje zahteva za prekid,
- određivanje prioriteta prekida,
- pripremu prekida i
- vraćanje u prekinuti program.

Prekidnim signalom se zahteva prelaz sa tekućeg programa na program za obradu prekida. Postoje različite vrste prekida koji imaju različite prioritete. Pošto svi prekidni signali ne moraju biti prihvaćeni, po prijemu signala za prekid vrši se analiza prioriteta. Po prijemu prekidnog signala, prekinuti program se privremeno napušta i kada se stvore uslovi, procesor mu se ponovo dodeljuje. Da bi program normalno nastavio da radi, potrebno je sačuvati zatečeni sadržaj nekih registara procesora (npr. akumulatora, brojača instrukcija, indeksnih i baznih registara i registra stanja). Kada se program ponovo aktivira, navedeni registri se pune zapamćenim sadržajem.

Upravljanje vremenom procesora se zasniva na dodeljivanju procesora datom zadatku prema unapred određenom algoritmu (strategiji planiranja) u slučajevima kada to zahteva više zadatka. Funkcija operativnog sistema za upravljanje vremenom procesora (supervizor procesora), najčešće podrazumeva sledeće aktivnosti:

- rešavanje problema podele vremena procesora,
- dodelu i analizu prioriteta zadacima i
- obezbeđivanje rada u realnom vremenu.

Savremeni računari sve češće poseduju više procesora (više-procesorski sistemi), zbog čega upravljanje vremenom procesora postaje složenije. U tom slučaju koriste se i različiti algoritmi za raspoređivanje zadataka kod višeprogramskih operativnih sistema, da bi se obezbedila što veća propusna moć sistema, tj. što je moguće više skratilo neproduktivno vreme procesora. U tom cilju se, kod operativnog sistema UNIX, na primer,

vreme procesora raspoređuje dodeljivanjem dinamičkog prioriteta zadacima, s tim što sistemski zadaci imaju viši prioritet od korisničkih.

Savremeni 32 i 64-bitni računari najčešće rade u višeprogramskom režimu, što znači da se više programa istovremeno nalazi u operativnoj memoriji. Javlja se potreba za podelom operativne memorije na delove u koje se smeštaju zadaci kojima je potrebno vreme centralnog procesora.

Glavni zadaci programa operativnog sistema za upravljanje memorijom jesu: vođenje evidencije o slobodnim i zauzetim delovima operativne memorije, odlučivanje o zahtevima zadataka ili poslova za dodelom operativne memorije, dodela operativne memorije zadatku i ponovna raspodela slobodog i zauzetog memorijskog prostora i oslobađanje operativne memorije posle njenog korišćenja.

Programi operativnog sistema za sinhronizaciju i opsluživanje zadataka vode računa o tome da se zaštiti konzistentnost podataka i smanji verovatnoća pojave grešaka pri međusobnom komuniciranju više zadataka. Najčešće greške nastaju kada više zadataka pristupi određenom resursu u isto vreme. Da bi se to izbeglo, programi za sinhronizaciju i opsluživanje zadataka moraju da poštuju sledeća pravila:

- u datom trenutku određeni resurs može koristiti samo jedan zadatak,
- ukoliko se pojavi više zahteva za određeni resurs, on se dodeljuje jednom od zadataka na određen (konačan) vremenski period,
- kada zadatak dobije resurs, mora ga predati u nekom konačnom vremenu i
- zadatak dok čeka dodelu resursa ne troši vreme procesora.

Osnovna funkcija organizacije službe vremena ima za cilj da spreči neki od zadataka da zadrži procesor neograničeno dugo i da prikaže realno vreme. Za te svrhe u računaru postoji sat realnog vremena koji uglavnom sadrži:

- kristalni oscilator,
- brojač i

Glava 6. Softver računara

- registar podataka.

Kad god program za raspodelu vremena centralnog procesora dodeli procesor nekom zadatku, brojač se inicijalizuje na vrednost dodeljenog vremena kopiranjem sadržaja registra podataka u brojač. Kristalni oscilator generiše impulse u određenom intervalu, impulsni signal se prenosi do brojača koji se dekrementuje (umanjuje za 1). Kada brojač dođe do nule, generiše se signal za programski prekid.

Upravljanje podacima

U savremenim računarima podaci koji se obrađuju čuvaju se u obliku datoteka i baza podataka na nekom od medijuma za pamćenje podataka: magnetnim ili optičkim. Operativni sistem vodi evidenciju o raspoloživim podacima, upravlja pristupom određenim podacima i njihovom dodeljivanju programima koji ih obrađuju. Najvažnije funkcije ove komponente operativnog sistema su:

- upravljanje periferijama,
- upravljanje datotekama,
- upravljanje dodeljivanjem sekundarne memorije,
- služba kataloga (direktorijuma ili imenika),
- obrada ulazno-izlaznih otkaza.

Operativni sistem u određenoj meri upravlja radom perifernih uređaja u vidu programske podrške. Zbog samih specifičnosti perifernih uređaja, koji su uglavnom elektromehaničke mašine (dakle prilično sporije od računara), upravljanje se ovde odvija u dva osnovna nivoa: logičkom i fizičkom.

Na logičkom nivou upravljanje periferijama treba da obezbedi:

- detekciju greške i eventualnu korekciju ili upozorenje,
- prilagođavanje perifernog uređaja centralnoj jedinici računara,
- sigurnost rada sa periferijom, što podrazumeva i podršku standarda za razmenu podataka.

Na fizičkom nivou upravljanje uređajima obezbeđuje:

- određivanje (ili dodeljivanje) adrese perifernom uređaju,
- formiranje niza kontrolnih znakova za uspostavljanje komunikacije,
- sinhronizaciju rada više perifernih uređaja (na primer: monitora i tastature),
- prilagođavanje brzine protoka podataka različitim perifernim uređajima.

Osnovna celovita forma za čuvanje podataka u računani su datoteke. Operativni sistem vodi evidenciju o datotekama tako što ima posebne tabele u kojima su opisane datoteke podataka. Ove tabele se nalaze u operativnoj memoriji i najčešće sadrže:

- ime, veličinu i vreme poslednje promene datoteke,
- podatke o njenoj organizaciji,
- način pristupa datoteci i
- status datoteke.

Upravljanje datotekama predstavlja, u stvari, spregu između korisničkih (aplikativnih) programa i funkcija operativnog sistema za upravljanje ulazom-izlazom (ulazno-izlazni supervizor). Pored vođenja evidencije o datotekama, funkcije upravljanja datotekama su:

- pretvaranje logičkog zahteva aplikativnog programa za operacijama nad zapisima datoteke u fizički zahtev za ulazno-izlaznom operacijom i predaja zahteva ulazno-izlaznom supervizoru;
- otvaranje datoteka, određivanje mesta gde se nalaze informacije o njoj, korišćenjem podataka iz kataloga datoteka;
- raspodela memorijskog prostora na sekundarnoj memoriji;
- regulisanje pravila pristupa datoteci;
- zatvaranje datoteke posle završetka rada - brisanje bafera i drugih radnih oblasti u operativnoj memoriji vezanih za rad sa datom datotekom.

6.2 Režimi rada računara

Programi se u procesoru računara mogu izvršavati na nekoliko različitih načina, kao što je prikazano na Slici 6.3, koje nazivamo režimi obrade ili režimi rada. Pre svega postoje jedno-programski i višeprogramski (multiprogramski) režim rada i paralelna obrada (multiprocesiranje).

Računarski sistem radi u jednoprogramskom ili monoprogramskom režimu ako procesor opslužuje samo jednog korisnika. Pri tome se u operativnoj memoriji računara nalazi samo jedan program koji se izvršava od početka do kraja. Računar u ovom režimu opslužuje korisnike na dva načina:

- (a) pojedinačnom obradom (individualno opsluživanje) i
- (b) paketnom obradom (indirektno opsluživanje).

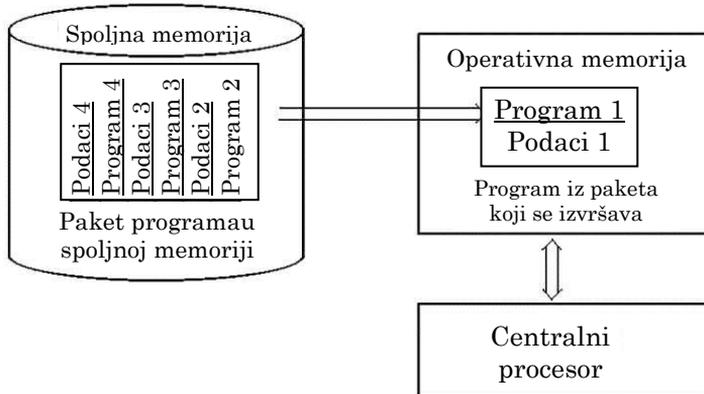
Kod pojedinačne obrade računar je u procesu rešavanja problema u potpunosti na raspolaganju jednom korisniku. Korisnik sam radi sa računarem, unosi program ili ga poziva sa spoljne memorije, startuje izvršenje programa, čime pristupa rešavanju svog problema i čeka rezultate. Ovakvo opsluživanje korisnika bilo je u prvoj generaciji i dobrim delom u drugoj generaciji računara, a danas se koristi kod personalnih računara.

<p>Osnovni režimi rada</p> <ul style="list-style-type: none"> • Jednoprogramska obrada (monoprogramska obrada) • Višeprogramska obrada (multiprogramska obrada) • Paralelna obrada (multiprocesiranje)
<p>Lednoprogramska obrada</p> <ul style="list-style-type: none"> • Pojedinačna obrada (individualno opslučivanje) • Paketna obrada (indirektno opslučivanje)
<p>Višeprogramska obrada</p> <ul style="list-style-type: none"> • Obrada sa podelom vremena • Rad u realnom vremenu • Daljinska obrada • Distribuirana obrada

Sl. 6.3: Režimi rada računarskih sistema

Paketna obrada, ili, tačnije rečeno, jednoprogramski paketni

režim rada, jeste način rada računara u kome se obrađuju paketi. Paket (engl. batch) predstavlja grupu programa i odgovarajućih ulaznih podataka koja se za obradu na računaru tretira kao jedna celina. Paket je smešten na spoljnoj memoriji kao red čekanja iz koga se programi čitaju i obrađuju jedan po jedan redosledom u kome se nalaze u paketu (sl. 6.4).



Sl. 6.4: Višeprogramski režim rada

Kod pakettne obrade korisnik nema neposredan pristup računarskom sistemu. Pripremljeni programi i podaci predaju se operateru računara koji ih zapisuje na spoljnu memoriju gde se formira paket ili korisnik sa svog terminala smešta programe i podatke u paket na disku (spoljnoj memoriji). Izvršavanjem programa kod pakettne obrade upravlja operativni sistem koji obezbeđuje prelaz sa jednog programa na drugi i kontroliše njihovo izvršavanje. Obrada programa je sekvencijalna (serijska). U jednom vremenskom trenutku obrađuje se samo jedan program, dok se ostali programi nalaze u stanju čekanja. Za vreme izvršenja jednog programa paketa zabranjeno je njegovo prekidanje. Prelaz na sledeći program moguć je samo ako je završen prethodni ili je on prekinut zbog neke izuzetne situacije, kao što je greška u programu i sl.

Rezultati obrade pamte se na disku da bi se prosledili korisniku u trenutku kada se oslobode izlazne jedinice, na primer

štampanje, ili kada se oslobodi terminal.

Veliki broj komercijalnih sistema koristio je paketnu obradu za tzv. obradu promena ili obradu transakcija. Ulazni zapisi, obično nazivani zapisi promena ili zapisi transakcija, prvo su prikupljeni na spoljnoj memoriji onako kako su oni pristizali, a onda su obrađivani svi zajedno. Svi ulazni zapisi formirani u toku određenog vremenskog perioda grupisani su u posebnu datoteku koja se naziva datoteka promena. Datoteka promena se zatim obrađuje radi ažuriranja podataka u osnovnoj - matičnoj datoteci.

Mana obrade promena je u tome što matična datoteka i njeni zapisi nikada nisu sasvim ažurni. Međutim, ova obrada se i dalje koristi, čak i na današnjim bržim i složenijim računarima, jer u mnogim slučajevima nema potrebe da se svakog momenta raspoláže najvažnijim informacijama, na primer pri obračunu potrošnje električne energije.

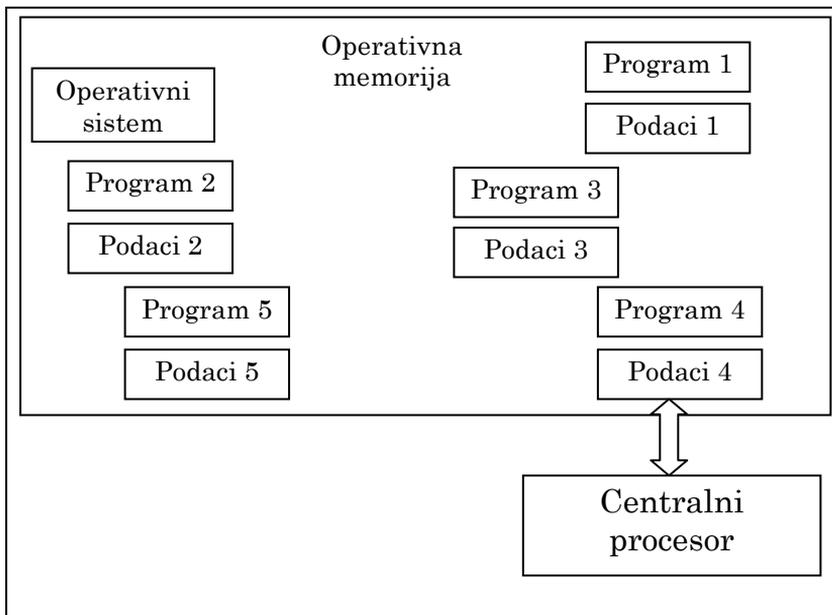
Stepen efikasnog korišćenja računara kod jednoprogamske i paketne obrade je bio nizak. Obrada se izvodila serijski, to jest trebalo je svaki postupak dovršiti da bi sledeći mogao da otpočne. Ako procesor mora da se bavi prvo čitanjem i pamćenjem podataka, zatim njihovom obradom i, na kraju, izdavanjem rezultata za, recimo, štampač, većina njegovog vremena otpada na ulazno-izlazne procese. Tako je udeo same obrade u ukupnom vremenu bio vrlo mali. To je posledica, s jedne strane, veoma malih brzina ulazno-izlaznog prenosa podataka i, s druge strane, izuzetno velike brzine njihove obrade u procesoru.

Sistem koji sadrži veći broj procesora i svi oni mogu učestvovati istovremeno u rešavanju jednog istog problema naziva se višeprocorski računarski sistem, multiprocesor ili paralelni računar. Pri tome se višeprocorskim sistemom ne smatra sistem čiji dodatni procesori služe samo za pomoćne, recimo ulazno-izlazne ili telekomunikacione aktivnosti. Obrada na ovakvim sistemima naziva se paralelna obrada ili multiprocesiranje.

Činilac koji ograničava stepen efikasnosti rada centralnog procesora je srazmerno mala brzina rada ulaznih i izlaznih uređaja. Ako u toku rada centralnog procesora na jednom programu ima dosta praznog hoda zbog prinudnog čekanja na ulazne podatke, može se u operativnu memoriju smestiti i aktivirati dva ili

više programa istovremeno. Kada procesor ne može da radi na jednom od njih, preći će na obradu drugog. Ovaj način rada naziva se višeprogramski način rada ili multiprogramiranje.

Računarski sistem radi u višeprogramskom načinu rada ako se više korisničkih programa nalazi u operativnoj memoriji, svi su oni aktivirani za izvršenje i izvršavaju se po određenom redosledu (Slika 6.5). Time je omogućena konkurentna ("paralelna", "istovremena") obrada više programa. Pri tome program koji se izvršava može biti prekinut radi prelaska na izvršenje nekog drugog programa, a naknadno se može vratiti na taj isti program. Muiultiprogramiranjem se poboljšava iskorišćavanje procesora, kao i operativne memorije.



Sl. 6.5: Višeprogramski režim rada.

Kao i u slučaju jednoprogramskog načina rada, i kod višeprogramskog načina rada procesor radi strogo serijski izvršavajući instrukciju po instrukciju. Kada se govori o konkurentnom izvršavanju programa, ima se u vidu da posle izvršavanja dela jednog programa procesor prelazi na izvršavanje dela drugog programa, zatim sledećeg itd., zadržavajući pri tome potrebne informacije za mogući povratak i nastavljanje obrade jednog od prethodnih programa. Ovaj način obrade obezbeđuje značajno povećanje sveukupne

efikasnosti procesora, mada je obrada pojedinačnih programa, po pravilu, duža nego kod jednoprogamskih računara.

Multiprogramiranje se može obavljati samo na računarima koji imaju složen operativni sistem koji je u stanju da donosi odluke o tome kako će se koji program pozivati. Drugi preduslov je dovoljan broj periferija u direktnoj vezi s računarom koje mogu istovremeno obavljati svoje poslove. Odluka o redosledu izvršavanja programa donosi se prema prioritetima saopštenim operativnom sistemu na početku rada.

Teoretski, muitiprogramiranjem je moguće postići efikasnost centralnog procesora od 100%. U praksi je malo verovatno da će sled naizmenične obrade programa teći bez prekida, pa procesor i dalje nije potpuno iskorišćen. Ipak, ovaj način obrade i te kako povećava njegovu efikasnost.

Paketna obrada se može organizovati i kod višeprogamskog načina rada. To su višeprogamski sistemi sa paketnom obradom.

Osnovni problem višeprogamskog načina rada je organizacija zaštite od uticaja jednog programa na drugi. Naročito je važna zaštita operativne memorije, koja se sastoji u tome da jedan program ne sme pristupati delovima operativne memorije dodeljenim drugim programima ili operativnom sistemu.

U okviru višeprogamskog načina rada razlikuje se nekoliko karakterističnih vrsta obrade, a to su:

- obrada sa podelom vremena,
- rad u realnom vremenu,
- daljinska obrada,
- distribuirana obrada.

6.2.1 Obrada sa podelom vremena

Suština načina rada sa podelom vremena (ili vremenskom podelom) jeste da se obezbedi da više korisnika gotovo istovremeno ima pristup računaru. Korisnici raspolažu udaljenim terminalima ili personalnim računarima koji mogu obavljati i ulogu terminala, povezanim telekomunikacionim vezama sa računarom. Svaki je terminal snabdeven vlastitom memorijom, u kojoj se podaci drže u toku ulaznih i izlaznih operacija. Ovde vreme

odziva, tj. vreme koje je potrebno računaru da započne slanje informacija terminalu, mora biti kratko. Ono je, u stvari, obično kraće od desetak sekundi.

Podela vremena predstavlja takvo raspoređivanje zadataka (procesa) gde je zadatku dozvoljeno da se izvršava tokom unapred definisanog perioda vremena, pre ponovnog raspoređivanja. U podeli vremena više korisnika istovremeno pristupa računaru.

U praksi se postupa tako što se svakom korisničkom terminalu dodeljuje unapred definisani period vremena u toku koga se njegov program stavlja na raspolaganje centralnom procesoru. Taj period vremena je kratak, nekoliko desetina milisekundi. Zbog relativno brzog odziva sistema svaki korisnik ima utisak da radi sam sa računarom. Ova istovremenost pretpostavlja da:

1. svaki korisnik poseduje sopstvene uređaje za pristup računaru;
2. pri istovremenom pristupu više korisnika računar mora da se ponaša, sa tačke gledišta korisnika, na isti način kao i pri individualnom opsluživanju.

Za opsluživanje u podeli vremena koriste se višeprogramski sistemi. Razlikuju se tri osnovna režima opsluživanja korisnika u podeli vremena:

- (a) režim dijaloga (interaktivni režim),
- (b) transakcioni režim i
- (v) univerzalni režim.

U režimu dijaloga korisnik sa svog terminala ima direktan pristup računarskom sistemu i može postaviti jedan ili više upita. Kada se na jednom od terminala generiše upit, operativni sistem prekida izvršenje tekućeg programa i u memoriju poziva program namenjen obradi upita. Kod transakcione obrade vrši se raspodela računarskih resursa između određenog broja udaljenih korisnika koji preko sopstvenih terminala direktno pristupaju računarskom sistemu. Za razliku od paketne obrade transakcija, ovde se ažuriranje matične datoteke obavlja odmah. Računarski sistem sa radom u univerzalnom režimu može raditi u bilo kom od opisanih režima. Primer sistema sa vremenskom podelom su

sistemi za rezervaciju avionskih karata: pošto se preko terminala unese zahtev za mesto u avionu na određenom letu, pojaviće se spisak svih slobodnih sedišta. Rezervisano mesto mora se odmah zapisati, to jest podaci ažurirati kako sledeći putnik ne bi dobio netačnu informaciju.

6.2.2 Rad u realnom vremenu

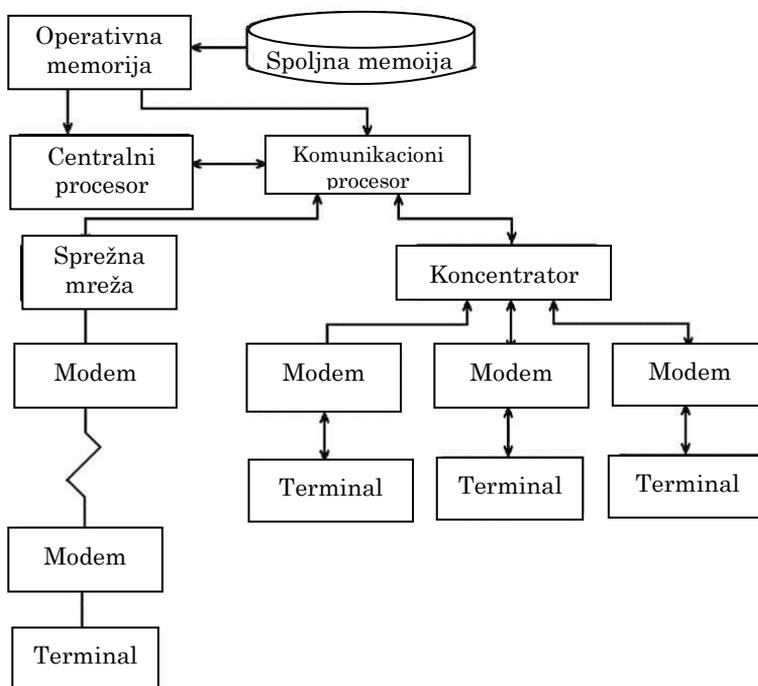
Računarski sistem koji proizvede izlaz za unapred zadato vreme naziva se sistem za rad u realnom vremenu ili sistem realnog vremena. To znači da je razmak između ulaznog vremena zahteva za obradom i izlaznog vremena te obrade značajan i mora biti dovoljno mali da odziv bude pravovremen. Primer rada u realnom vremenu čine sistemi za upravljanje tehnološkim procesima ili sistemi za upravljanje projektilima.

Najvažnije svojstvo sistema za rad u realnom vremenu je to što on mora da bude u stanju da prihvati i obradi podatke u vremenu kada je neka pojava u toku, i da rezultate obrade za modifikovanje nekih parametara te pojave blagovremeno povratno saopšti. Međutim, ovde vreme odziva, tj. vremenska razlika između nastajanja podataka i povratnog saopštavanja rezultata, zavisi od primene. Na primer, da bi se uticalo na ponašanje rakete u letu, odzivno vreme mora se kretati u okviru mikrosekundi. Kod pojedinih tehnoloških procesa vreme odziva može se meriti milisekundama.

Sistemima realnog vremena najčešće se smatraju oni sistemi kod kojih se vreme odziva meri milisekundama, dok se sistemima koji rade u režimu dijaloga smatraju sistemi u kojima se vreme odziva meri sekundama.

6.2.3 Daljinska obrada

Ukoliko se za pristup računaru koristi daljinski ulaz-izlaz tada se govori o daljinskoj obradi. Na procesor je preko telekomunikacionih linija vezan veći broj udaljenih terminala, personalnih računara ili radnih stanica koji se opslužuju u višeprogramskom režimu rada. Posredstvom modema ili koncentratora sa udaljenih uređaja šalju se u računar komande i podaci, a na osnovu primljenih komandi i podataka računar vraća izlazne rezultate (Slika 6.6).



Sl. 6.6: Povezivanje terminala kod daljinskog ulaza-izlaza

Primer daljinske obrade predstavlja sistem koji primenjuju banke. Prednost takvog sistema je ne samo u mogućnosti veoma brzog prenosa podataka nastalih u filijali do računara već i u činjenici da filijala ima pristup podacima koji se čuvaju u bazama podataka računara čime je obezbeđeno veoma brzo dolaženje do njih. U stvari, ovde se teži uvođenju moćnih centralnih računara koji su u mogućnosti da obrade podatke nastale na veoma širokom području. Korišćenjem terminala uvodi se još jedan princip, poznat kao višestruki pristup. To znači da više udaljenih ulazno-izlaznih punktova ima direktan i praktično istovremeni pristup računaru.

6.2.4 Distribuirana obrada

U opštem slučaju obrada se naziva distribuirana obrada ili raspodeljena obrada ako se vrši u sistemu u kome je povežno

više međusobno nezavisnih računara koji mogu da saraduju. To su tzv. distribuirani računarski sistemi. U svakom računaru distribuiranog sistema vrši se lokalna obrada, a preko telekomunikacionih linija međusobno se razmenjuju podaci koji su od uticaja za obradu na više računara. Distribuirana obrada zasniva se na činjenici da je neke funkcije obrade bolje obaviti lokalno, a druge centralno. Ona proširuje princip daljinske obrade postavljanjem računara na udaljene tačke i njihovim povezivanjem ne samo s centralnim računarom nego i međusobno. S pojavom miniračunara, radnih stanica i personalnih računara, kao i lokalnih računarskih mreža, ovaj način obrade postao je realno izvodljiv. Osnovna odlika distribuiranih sistema sastoji se u tome da različiti elementi sistema mogu biti raspodeljeni na odvojene delove u kojima je skoncentrisan strogo definisan poseban deo procedure obrade podataka i samih podataka.

Distribuirana obrada podrazumeva postavljanje računarske opreme za lokalne obrade u udaljene tačke. Ova oprema se koristi za obradu na mestu nastanka promena. Kada je to potrebno, moguć je pristup centralnim bazama podataka u glavnom računaru ili slanje sumarnih izveštaja. Uz to, međusobnim povezivanjem računara, na primer u slučaju računara postavljenih u fabrikama, omogućuje se koordinacija proizvodne aktivnosti, planiranje proizvodnje, održavanje optimalnih zaliha i slično.

Koncept distribuirane obrade ima više različitih varijanata. Ako se radi o jedinstvenoj konfiguraciji računarskog sistema, korišćenje više posebnih procesora ponekad se smatra jednom vrstom distribuirane obrade jer svi procesori u konfiguraciji paralelno (istovremeno) učestvuju u rešavanju jednog problema. U tom slučaju se kaže da su procesori "čvrsto spregnuti".

Drugi oblik distribuirane obrade jeste korišćenje procesnih elemenata koji su fizički odvojeni i slabije spregnuti. Tipičan primer ovakve obrade je korišćenje inteligentnih terminala i udaljenih podsistema za upravljanje u industriji. Ovde terminali za inteligentno prikupljanje podataka, podsistemi za upravljanje procesima i drugi procesni elementi mogu biti raspodeljeni po teritoriji fabrike. Određena ograničena obrada vrši se na udaljenim jedinicama, dok se sumarni rezultati prenose glavnom računaru putem prenosnih linija.

U opštem slučaju obrada se smatra distribuiranom ili raspodeljenom ako se vrši u sistemu u kome je povezano više međusobno nezavisnih računara koji mogu da saraduju. To su tzv. distribuirani računarski sistemi. U svakom računaru distribuiranog sistema vrši se lokalna obrada, a preko lokalnih računarskih mreža ili telekomunikacionih linija međusobno se razmenjuju podaci koji utiču na obrade na više računara.

Osnovna odlika distribuiranih sistema sastoji se u tome da različiti elementi sistema mogu biti raspodeljeni na odvojene delove u kojima je skoncentrisan strogo definisan poseban deo procedure obrade podataka i samih podataka. Konceptija se odlikuje sledećim karakteristikama distribuirane obrade:

- mnoštvo računarskih uređaja,
- celovitost (jedinstvo) sistema,
- električna povezanost sistema,
- visoki nivo uzajamnog delovanja između elemenata sistema.

Ako je veći broj terminala i računara povezan telekomunikacionim kanalima, onda oni čine sistem koji se naziva računarska mreža. Obrada u računarskoj mreži predstavlja proširenje koncepta distribuirane obrade. Korišćenje različitih konfiguracija lokalnih računarskih mreža vrlo je pogodno za distribuiranu obradu.

6.3 Programski sistem

Programski sistem sadrži programe koji služe za automatizaciju pripreme i testiranja korisničkih programa. To su programski jezici, programski prevodioci (asembleri, kompilatori i interpretatori), editori, biblioteke potprograma i uslužni programi. Osnovna podela programskog sistema je na programske prevodioce (jezičke procesore) i uslužne programe, (Slika 6.7).

6.3.1 Programski prevodioci

Termin programski prevodioci označava jezičke procesore koji prihvataju programe pisane na nekom od programskih jezika i

Programski sistem	
<ul style="list-style-type: none"> • Programski prevodioci (jezički procesori) • Uslužni procesori (servisni programi) • Sredstva za razvoj softvera (CASE alati) 	
Programski prevodioci	Uslužni programi
<ul style="list-style-type: none"> • Asembleri i makroassembleri • Kompilatori • Interpretatori • Pretprocesori 	<ul style="list-style-type: none"> • Editor • Povezivač • Sredstva za testiranje i otklanjanje grešaka • Pomoćni programi

Sl. 6.7: Komponente programskog sistema

generišu funkcionalno ekvivalentne programe na nekom drugom jeziku: Kompilator je prevodilac čiji je ulazni jezik viši programski jezik, a izlazni jezik je mašinski jezik. Interpretator je prevodilac čiji je ulazni jezik viši programski jezik a izlazni jezik je neka posredna forma koja se interpretira (izvršava). Za razliku od assemblera kompilator i interpretator, po pravilu, za jednu naredbu napisanu u višem programskom jeziku generišu više instrukcija mašinskog jezika. Kompilator i interpretator se razlikuju po tome što kompilator prevodi na interni mašinski jezik čitav program pa se tek onda pristupa njegovom izvršavanju, dok interpretator prevodi jednu naredbu iz višeg programskog jezika na niz instrukcija mašinskog jezika, pristupa izvršavanju ovih instrukcija, potom prelazi na prevođenje i izvršenje sledeće naredbe, itd.

Kompilatori

Kompilacija programa se može razložiti na dva glavna zadatka:

- (a) analizu - utvrđivanje strukture izvornog programa, i
- (b) sintezu - generisanje (konstrukciju) ciljnog programa ekvivalentnog izvornom programu.

U procesu analize izvorni program se konvertuje u apstraktnu formu koja sadrži sva bitna svojstva algoritma. Analiza obuhvata leksičku analizu i sintaksnu analizu. Leksičkom analizom

se razlazu ulazne jedinice određenog značenja, kao što su npr. imena, konstante, rezervisane reci, operatori. Semantička analiza, predstavlja proces odlučivanja da li je niz ulaznih simbola rečenica datog jezika i, ako jeste, određivanje sintaksne strukture tog niza.

Proces sinteze sadrži dva posebna dela: generisanje koda i asembliranje. Generisanje koda transformiše apstraktni izvorni program u ekvivalentni mašinski program. Asembliranjem se vrši konverzija mašinskih naredbi u odgovarajući izlazni format i formira se objektni modul prilagođen obradi od strane povezi-vača ciljne mašine.

Interpretatori

Interpretatorski način realizacije prevodioca najviše se primenjuje kod interaktivnih programskih jezika, kao što je npr. bejzik (engl. BASIC) i kod raznih komandnih jezika. Kad god stigne nova ulazna linija dalje akcije preuzima kontroler obrade koji poziva:

- (a) program homogenizator radi konverzije ulazne programske linije u neki standardni interni format, ili
- (b) sintaksni analizador i generator koda radi sintaksne provere i generisanja mašinskog koda, ili
- (v) procesor komandi radi obrade upravljačke (kontrolne) naredbe, npr. RUN, ali ne i njenog izvršenja.

Izvršenje prevedenog programa spada u nadležnost izvršioca. Realizuje se čitanjem odgovarajućeg mašinskog koda i njegovim izvršavanjem. Stoga je stvarno izvršavanje prevedenog programa mešavina interpretativnog i mašinskog koda, s tim što se većina naredbi izvršava preko mašinskog koda a kontrola prelaska s jedne na drugu naredbu podržava se interpretativno.

6.3.2 Uslužni programi

Editori

Priprema (unošenje) programa obavlja se u posebnoj fazi razvoja programa koja se naziva editovanje, a deo sistemskog softvera koji

realizuje tu fazu naziva se editor programa ili editor teksta. Namenjen je za unos i modifikaciju tekstualnih podataka, npr. programa na višim programskim jezicima, knjiga i drugih tekstualnih dokumenata. Rezultat editovanja programa je izvorni program.

Editori teksta čine bitan deo korisničkog interfejsa i najčešće su korišćeni delovi sistemskog softvera u interaktivnim računarskim sistemima. Za većinu korisnika oni predstavljaju glavnu spregu sa računarom. Mogu biti linijski editori, kada se tekst posmatra kao niz linija razdvojenih upravljačkim znakom za kraj linije, ili ekranski editori. Kod ekranskih editora sadržaj ekrana predstavlja prozor u tekst unutar kojeg se kursor može dovesti na mesto gde će se vršiti umetanje, brisanje i druge editorske funkcije.

Jedna od važnih karakteristika editora je skup editorskih funkcija koje su stavljene na raspolaganje korisniku. Taksonomija mogućnosti editora obuhvata potencijalno preko 200 funkcija koje se njime mogu izvršavati. Ovde ćemo opisati samo neke od najvažnijih i najčešće korišćenih funkcija nad tekstualnim objektima, gde se pod objektom podrazumeva: znak, reč, linija, rečenica, paragraf i sekcija, koji kao celina podležu određenoj operaciji editovanja:

- umetanje objekata u tekst,
- uklanjanje (brisanje) objekata iz teksta,
- zamena objekata u tekstu novim objektima,
- premeštanje objekata s jedne na drugu poziciju u tekstu,
- kopiranje objekata na razne pozicije u tekstu,
- razdvajanje i spajanje više objekata u tekstu na više delova,
- selekcija (označavanje) pojedinih delova teksta,
- traženje objekta u tekstu, itd.

Povezivači

Rezultat prevođenja programa pomoću kompilatora ne dobija se direktno u mašinskoj već u formi koja se naziva objektni modul. Da bi se program mogao izvršavati u računani neophodno je dovesti ga u izvršnu formu i smestiti ga u operativnu memoriju.

Izvršna forma programa se dobija povezivanjem objektnih modula u jednu celinu tzv. modul punjenja i njegovo svođenje na direktnu mašinsku formu (mašinski jezik). Celokupan posao oko generisanja izvršne forme i njenog smeštanja u operativnu memoriju obavlja se u posebnoj fazi razvoja programa koja se naziva povezivanje. Ova faza sledi posle procesa prevođenja. Deo sistemskog softvera koji realizuje funkciju povezivanja naziva se poveziivač ili editor veza. Postoje razne verzije poveziivača a ponekad oni vrše i punjenje izvršne forme programa u operativnu memoriju.

Programi za testiranje i otklanjanje grešaka

Otklanjanje grešaka ili prečišćavanje programa predstavlja proces identifikacije i odstranjivanja lokalizovanih grešaka iz programa. Nasuprot tome testiranje nastoji da ustanovi da li greške postoje, ali ih ne izoluje niti ih otklanja.

Otklanjanje grešaka u programu predstavlja dvodelni proces koji počinje indikacijom o postojanju greške kao rezultata uspešnog testiranja, i nastavlja sa sledecim aktivnostima:

- (a) precizno određivanje prirode uočene greške i
- (b) lokalizacija i otklanjanje greške.

Pomoćni programi

Pomoćni programi predstavljaju skup programa koji nisu fundamentalne prirode, ali se njima proširuju mogućnosti računarskog sistema i time poboljšava kvalitet usluga raspoloživih korisniku. Broj i vrsta ovih programa koji su zastupljeni u sistemskom softveru se razlikuje od sistema do sistema. Ovde ćemo navesti samo nekoliko tipova pomoćnih programa:

- formiranje programskih biblioteka,
- održavanje datoteka: brisanje, formiranje, spajanje, kopiranje, preimenovanje, itd,
- korišćenje datoteka: modifikacija, ažuriranje, čitanje, prikaz na ekranu, itd.
- rad sa katalozima (imenicima),
- konverzija datoteka iz jednog formata u drugi,

- štampanje sadržaja datoteka,
- upoređivanje sadržaja dve datoteke, itd.

6.3.3 Sredstva za razvoj softvera

U razvoju softvera se otišlo do primene softvera pete generacije. U petoj generaciji softvera ubraja se čitav skup metoda i tehnika automatizovanog projektovanja integrisanih informacionih sistema. Jednim imenom ih nazivamo, CASE (Computer Aided Software Engineering) alati.

Osnovne karakteristike CASE alata su u tome što definišu metodologiju i tehniku za automatizaciju softverskih rešenja.

Komponente CASE omogućavaju:

- sakupljanje, čuvanje, i održavanje svih informacija realnog sistema;
- definisanje strukture i relacija u bazi podataka;
- izgled ekrana i izlaznih izveštaja;
- definisanje programa za pristup bazi podataka, omogućavajući njenu zaštitu i integritet;
- interaktivni rad korisnika, manipulacije s podacima iz baze, simuliranje toka procesa i rešenja aplikacija;
- proizvodnju svih operativnih komponenti automatski, opise baze podataka, tehničku i korisničku dokumentaciju, datoteke sa "help" i "error" procedurama i sl.

Uz automatizaciju navedenih rešenja, dolazi još i standardizacija i konzistentnost programa, tako da CASE softveri, ne samo da unose velike novine u vidu informacionog inženjeringa, već i predstavljaju značajno povećanje produktivnosti i efikasnosti korišćenja hardverske i softverske opreme, kako od strane profesionalnih informatičara, tako i od strane krajnjih korisnika.

6.4 Operativni sistem UNIX

Ken Thompson, Dennis Ritchie i istraživači iz AT&T Bell laboratorija razvili su operativni sistem Unix 1969. godine,

uključivši mnoge mogućnosti istraživačkog projekta MULTICS (Multiplied Information and Computing Service). Sistem su oblikovali za potrebe istraživačkog okruženja, projektujući ga tako da se izvršava na mini-računarima. Od svog početka, Unix je bio efikasan višekorisnički operativni sistem koji je u stanju da obavlja više zadataka istovremeno.

Unix sistem postao je popularan u Bell laboratorijama budući da je sve više i više istraživača počinjalo da ga koristi. Dennis Ritchie je saradivao sa Kenom Thompsonom 1973. godine, ponovo napisao programski kod sistema Unix u programskom jeziku C. Unix je postepeno rastao, od projekta koji je oblikovala jedna osoba do standardnog softverskog proizvoda koji distribuiraju mnogi prodavci, kao što su Novell i IBM. U početku, Unix je smatran za istraživački projekt. Prve verzije Unixa distribuirane su besplatno katedrama za računarstvo mnogih poznatih univerziteta. Tokom sedamdesetih, Bell laboratorije počele su da izdaju zvanične verzije Unixa i da izdaju dozvole za sisteme različitim korisnicima. Jedan od ovih korisnika bila je katedra Computer Science kalifornijskog univerziteta Berkeley. Na Berkeley su sistemu dodane mnoge nove mogućnosti koje su kasnije postale standardne. Berkeley je 1975. godine izdao sopstvenu verziju Unixa, poznatu pod imenom Berkeley Software Distribution (BSD). Ova BSD verzija Unixa postala je glavni takmac verziji AT&T Bell laboratorija. U AT&T-u razvili su nekoliko istraživačkih verzija Unixa, a 1983. godine je AT&T izdao prvu komercijalnu verziju, nazvanu System 3. Za njom je kasnije sledio System V, verzija koja je postala podržani komercijalni softverski proizvod.

U isto vreme, BSD verzija Unixa razvijala se kroz nekoliko izdanja. Kasnih sedamdesetih, BSD Unix je postao osnova istraživačkog projekta koji je sprovela vladina agencija DARPA (Department of Defence's Advanced Research Projects Agency). Rezultat je bio taj da je 1983. godine Berkeley izdao moćnu verziju Unixa nazvanu BSD izdanje 4.2. Ovo izdanje se odlikovalo sofisticiranim upravljanjem datotekama kao i mogućnostima umrežavanja zasnovanim na Internet mrežnim protokolima - istim protokolima koji se sada koriste za Internet. BSD izdanje 4.2 bilo je široko distribuirano i usvojili su ga mnogi proizvođači, kao što je Sun Microsystems.

Sredinom osamdesetih izdvojila su se dva konkurentna stan

darda, jedan zasnovan na AT&T verziji Unixa i drugi zasnovan na BSD verziji. AT&T-ove laboratorije Unix System Laboratories razvile su System V izdanje 4. Nekoliko drugih kompanija, poput IBM-a i Hewlitt-Packarda, osnovale su organizaciju OSF (Open Software Foundation) kako bi razvile sopstvenu standardu verziju Unixa.

Razvojem izvorne verzije operativnog sistema, pojavljivale su se i nove verzije različitih proizvođača. Međutim, za sve verzije Unix-a bilo je zajedničko da je to modularni, višeprogramski operativni sistem koji omogućava rad sa podelom vremena. Sa druge strane, Unix je nezavisan od hardvera, pogodan za razvoj programa i lak za korišćenje.

Unix ima slojevitu strukturu klasičnih operativnih sistema, a nju formiraju:

- centralni deo
- jezgro,
- pomoćni sistemski programi - alati i
- programski prevodioci i korisnički programi.

Centralni deo operativnog sistema se naziva jezgro ili nukleus (engl. kernel). Ovaj deo operativnog sistema je zadužen za direktnu komunikaciju sa hardverom. Upravljanje hardverom Unix ostvaruje preko specijalne datoteke (SPECIAL FILE). Za komunikaciju jezgra sa gornjim slojevima koriste se tzv. sistemski pozivi.

Osnovne komponente jezgra su:

- sistem za upravljanje zadacima,
- sistem za vođenje evidencije o datotekama i
- sistem za upravljanje ulazom-izlazom.

6.4.1 Upravljanje zadacima

Sistem za upravljanje zadacima ostvaruje:

- komunikaciju između zadataka,
- sinhronizaciju između zadataka,

- planiranje zadataka,
- upravljanje memorijom.

Zadatak ili proces u Unix-u predstavlja program koji se izvršava. On može da se nađe u dva osnovna režima rada:

- u sistemskom režimu (engl. System state - povlašćenom režimu ili režimu jezgra, kada zadatak izvršava jezgro i kada pristupa sistemskim segmentima podataka;
- u režimu korisnika (engl. user state), kada zadatak izvodi korisničke programe i pristupa korisničkim segmentima podataka.

Prelazak iz režima korisnika u sistemski režim ostvaruje se preko sistemskog poziva, dok se prelazak iz sistemskog režima u režim korisnika dešava kada je sistemski usluga (pristupanje sistemskim segmentima podataka) - izvršena, ili kada se, ako usluga nije mogla biti izvršena, pošalje odgovarajuća poruka o greškama.

Kreiranje zadataka se ostvaruje posebnim sistemskim pozivom fork. Kada počne učitavanje programa jezgra sa diska, aktivira se zadatak "0" ili jezgro. Sve druge novije procese kreira neki od postojećih zadataka. Svaki zadatak dobija svoj identifikacioni broj. Tako, na primer, jezgro ima identifikacioni broj 0; sledeći zadatak, koji nadgleda kreiranje ostalih zadataka (init proces), kreira se od strane jezgra, sistemskim pozivom fork i on dobija identifikacioni broj 1. Ostali zadaci mogu imati različite identifikacione brojeve.

Komunikacija između zadataka i njihova međusobna sinhronizacija može se ostvariti na više načina:

- povezivanjem zadataka pomoću cevi (pipe) protok bajtova između procesa u jednom smeru,
- povezivanjem zadataka preko poruka,
- povezivanjem zadataka preko zajedničke memorije,
- povezivanjem zadataka preko semafora.

Planiranje zadataka predstavlja dodelu procesorskog vremena zadacima. Unix koristi algoritam kružnog planiranja sa redovima u više nivoa (RR-algoritam – round robin with multilevel feedback). Korisnički zadaci dobijaju dinamičke prioritete, zavisno

od količine već dobijenog procesorskog vremena. Zadaci koji su dobili više procesorskog vremena, dobijaju niže prioritete i obratno.

6.4.2 Upravljanje memorijom

Upravljanje memorijom se ostvaruje kreiranjem virtuelnog adresnog prostora. Memorija se zadacima dodeljuje u stranicama. Svaki zadatak dobija sliku memorije koja označava sadržaj koji je u memoriji za vreme izvođenja zadatka. Slika memorije dodeljene određenom zadatku može se prebacivati iz operativne memorije na disk (engl. swapping) ako neki od zadataka višeg prioriteta zatraži operativnu memoriju.

6.4.3 Upravljanje datotekama

Sistem za upravljanje datotekama obavlja sledeće poslove:

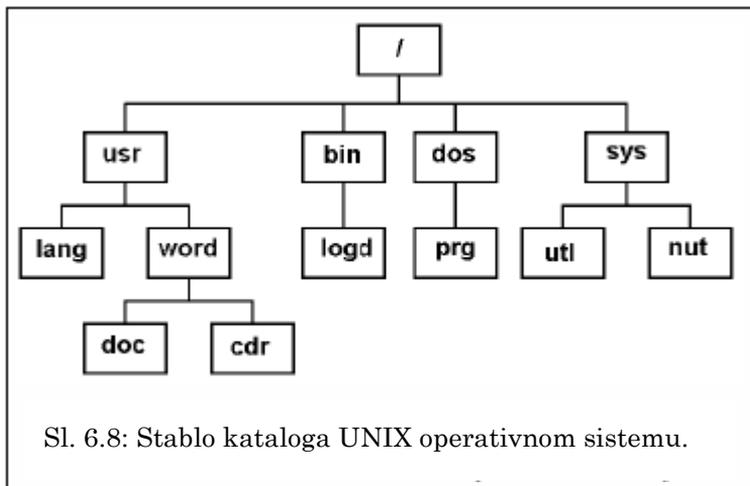
- dodeljuje prostor na spoljnjoj memoriji (diskovi, magnetne trake),
- vodi evidenciju o slobodnom prostoru,
- pronalazi podatke (pretražuje sadržaj datoteka) po zahtevu korisnika,
- vodi računa o pristupu datotekama.

Datoteke su organizovane na hijerarhijski način formirajući strukturu stabla kataloga (direktorijuma) i datoteka. Primer organizacije sistema datoteka u Unix-u je dat na slici 6.8.

Korisnički programi i podaci su u Unix operativnom sistemu smešteni u vidu nekoliko tipova datoteka. Postoji više tipova datoteka:

- obične datoteke,
- datoteke katalogi (imenici, adresari ili direktorijumi),
- FIFO datoteke
- specijalne datoteke.

Tipovi datoteka definišu njenu funkciju i način pristupa.



Obične datoteke sadrže podatke organizovane na određeni način. Ne postoji ograničenje na format običnih datoteka. Obične datoteke mogu biti binarni programi, programi zapisani u nekom izvornom programskom jeziku ili podaci u različitim oblicima.

Datoteke katalogi su čvorovi koji povezuju strukturu datoteka u celinu. Ove datoteke imaju već unapred definisani format zapisa i tretiraju se kao i ostale datoteke, osim što poseduju dodatnu zaštitu.

FIFO datoteke omogućavaju komunikaciju između zadataka koji nisu međusobno povezani. Najčešće se FIFO datoteke koriste kada je potrebno da se ostvari komunikacija više zadataka sa jednim određenim zadatkom.

Specijalne datoteke opisuju fizičke uređaje. Podatak upisan u specijalnu datoteku prosleđuje se odgovarajućoj perifernoj jedinici i obrnuto: kada se podaci čitaju sa perifernog uređaja, predaju se zadatku koji je tražio podatke. U stvari, specijalne datoteke predstavljaju samo pokazivače na adrese programa koji se nalaze u samom jezgru operativnog sistema i koji upravljaju radom perifernih jedinica.

Obične, FIFO i specijalne datoteke u strukturi stabla datoteka mogu se predstaviti kao listovi, dok datoteke katalogi (imenici) predstavljaju grane stabla.

6.4.4 Upravljanje ulazom-izlazom

Upravljanje ulazom-izlazom se ostvaruje preko ulazno-izlaznog sistema. Unix sistem je orijentisan na pojedinačni ulaz-izlaz koji se sastoji od bafer-memorije i veznih upravljačkih programa (drajvera).

Ulazni niz predstavlja niz znakova koji se završavaju nekom od oznaka:

- oznakom za kraj linije (RETURN),
- oznakom za kraj datoteke (ASCII EOF),
- oznakom za brisanje znaka (ASCII DEL).

Svaki od ovih znakova prouzrokuje prekid prijema ulaznog niza znakova i inicira obradu unetih podataka. Dužina ulaznog niza, očigledno, nije unapred definisana. Na taj način se ostvaruje veća nezavisnost perifernih uređaja od ostalih delova sistema i fleksibilnost u radu sa podacima, ali je potrebno da za svaki tip uređaja postoji odgovarajući vezni upravljački program. Razlikuju se tri tipa perifernih uređaja, pa samim tim i upravljačkih programa za njih:

- znakovno orijentisani upravljački programi: za štampače, terminale, monitor itd.;
- blok orijentisani upravljački programi: za diskove i magnetne trake;
- upravljački programi za mrežne interfejse.

Programi u Unix operativnom sistemu sve spoljnje uređaje vide kao datoteke. Analogno tome, ulazno-izlazni sistem poseduje četiri osnovna poziva:

1. open - otvaranje datoteke,
2. close - zatvaranje datoteke,
3. write - pisanje u datoteku i
4. seek - pretraživanje datoteke

6.4.5 Pomoćni sistemski programi alati

Oko jezgra operativnog sistema UNIX, nalaze se pomoćni sistemski programi. Najznačajniji među njima su:

- shell, sh - interpretatori komandnog jezika;
- ed, vi, ex - editori teksta;
- ld - punilac (engl. loader);
- as - asembler.

Najčešće se komunikacija korisnika sa računarom ostvaruje preko komandi koje interpretira shell. Prema ovom programu se UNIX odnosi kao prema svakom drugom korisničkom programu, što znači da shell nije rezidentan u memoriji računara (kao na primer jezgro), već se po potrebi učitava. To znači da shell predstavlja jedan od "običnih" programa koji svaki korisnik može menjati i prilagođavati sebi, a to opet znači da pored standardnih komandi korisnik može dodavati svoje. Komandna linija sadrži:

- ime komande,
- argumente koje komanda zahteva (oni se odvajaju blanko znacima).

Shell korisnički interfejs "prepoznaje":

- izvršavanje programa u pozadini,
- protok podataka između komandi (pipe),
- redirekciju (preusmeravanje) komandi izlazne vrednosti komande se mogu slati u neki određeni fajl ili neki od priključenih uređaja.

U tabeli 6.1 su date neke od komandi UNIX-a.

Kada se neka komanda ili datoteka (program) pokrene na izvršenje, shell mu dodeljuje tri nove datoteke:

- za ulaz,
- za izlaz,
- za greške.

Tabela 6.1: Neke komande UNIX operativnog sistema

Команда	Значење
cd	Promena tekućeg kataloga
chmod	Definisanje načina pristupa datoteci
chown	Promena vlasnika datoteke
cpm	Upredivanje sadržaja dve datoteke
cp	Kopiranje datoteka
date	Prikaz ili menjanje tekućeg datuma
find	Pretraživanje stabla kataloga
kill	Zatvaranje trenutno otvorenog dokumenta
ls	Prikazivanje sadržaja tekućeg dokumenta
mail	Slanje ili prijem elektronske pošte
mkdir	Kreiranje novog kataloga
passwd	Promena lozinke
pwd	Prikaz tekućeg kataloga
rm	Brisanje zadate datoteke
rmd	Brisanje tekućeg kataloga
stty	Prikaz ili promena opcija terminala
who	Ispis liste trenutnih korisnika

Ovakva koncepcija omogućava da se izlaz iz jedne komande iskoristi kao ulaz u drugu komandu. Takav način rada se naziva pipe (cev).

Novije verzije Unix operativnog sistema imaju grafički korisnički interfejs, što omogućava početnicima lakše upoznavanje i korišćenje. Grafičke radne stanice su obično priključene na lokalnu računarsku mrežu i, mada uglavnom poseduju sopstvene diskove i druge važnije resurse, oslanjaju se na jedan veći računar - server za datoteke. Standardni sistem za upravljanje prozorima u Unix-u, pod nazivom XWindows omogućava izvršavanje CAD programa i drugih aplikacija koje zahtevaju dodatne grafičke mogućnosti. Naravno, ispod sloja različitih aplikativnih grafičkih programa nalazi se ista - Unix osnova.

6.5 Operativni sistem Windows

Operativni sistem Windows (engl. windows - prozor) predstavlja proizvod firme Microsoft. Namenjen je različitim računarskim sistemima: od stonog PC računara, preko lokalnih i globalnih računarskih mreža, do superservera višeprosesorskih računara

baziranih na RISC procesorima.

MS Windows se kao operativni sistem pojavio osamdesetih godina. On je predstavljao platformu pod kojom su mogli da se izvršavaju mnogobrojni programi pisani za MS DOS, ali i da omogućiti:

- unificiran grafički korisnički interfejs,
- potpuno korišćenje mogućnosti savremenih procesora,
- efikasnije korišćenje operativne memorije,
- dinamičku razmenu podataka,
- virtuelni memorijski sistem,
- višeprogramske rad,
- podršku za multimediju itd.

Popularnost ovog operativnog sistema počinje od verzije Windows 3.0. Kasnije verzije: Windows 95, Windows 98 i Windows XP još više su učvrstile pozicije ovog operativnog sistema. Danas je Windows jedan od najprodavanijih operativnih sistema, pre svega za personalne računare. Za različite tipove računara postoje različite verzije Windows-a:

1. Windows 3.1 je namenjenim stonim PC računarima klase 386 i 486 i Pentium PC, kao i za razne prenosne računare;
2. Windows for Workgroups (prozori za radne grupe) ili Windows 3.11, predstavlja u stvari Windows 3.1, sa podrškom za lokalne mreže i integrisanu elektronsku poštu;
3. Windows 95 je naslednik Windows-a 3.1 x, predstavlja 32-bitnu verziju operativnog sistema namenjenu računarima sa nešto boljim karakteristikama i sa većim memorijskim zahtevima. Ova verzija predstavlja kompletan operativni sistem koji se definitivno odvaja od MS DOS-a, ali ga i dalje podržava; Windows NT je operativni sistem za snažne radne stanice i mrežne servere.

Programi za pružanje usluga operativnog sistema Windows su sadržani u tri modula:

- modul za opštu podršku
- jezgro (engl. kernel),
- grafički modul (GDI),

- korisnički modul - modul za upravljanje prozorima.

Windows 95 se oslanja na ovakvu strukturu, ali sadrži kompletniju podršku za rad sa 32-bitnim programima, omogućava veću dužinu imena datoteka, drugačiji, efikasniji pristup, optičkim diskovima i ostalim perifernim uređajima i dodatne mogućnosti za rad u mreži. Ipak, najveće mogućnosti pruža verzija Windows Nt.

Windows Nt (engl. N7 – New Tehnology nova tehnologija), predstavlja 32-bitni operativni sistem kreiran sa ambicijom da u budućnosti bude osnova za sve PC računare, radne stanice i servere. Objedinjuje sve dobre osobine koje bi trebalo da poseduje savremeni operativni sistem. Pored osobina prethodnih verzija, ovaj operativni sistem sadrži:

- (a) mogućnost rada na različitim hardverskim platformama (čak i na računarima sa različitim tipovima centralnih procesora);
- (b) podršku svim tipovima računarskih mreža i unapređene mogućnosti za distribuiranu obradu;
- (v) mogućnost korišćenja svih dosadašnjih DOS i Windows aplikacija;
- (g) veza sa UNIX aplikacijama preko standarda koji definiše potrebne funkcije i interfejse: POSIX (Portable Operating System Interface based on UNIX);
- (d) kompatibilnost sa savremenim operativnim sistemom OS/2, proizvodom firme IBM;
- (đ) visoku stabilnost sistema: greške u pojedinim aplikacijama ne mogu da prouzrokuju pad celog sistema;
- (e) podržava rad u višeprocensnim sistemima.

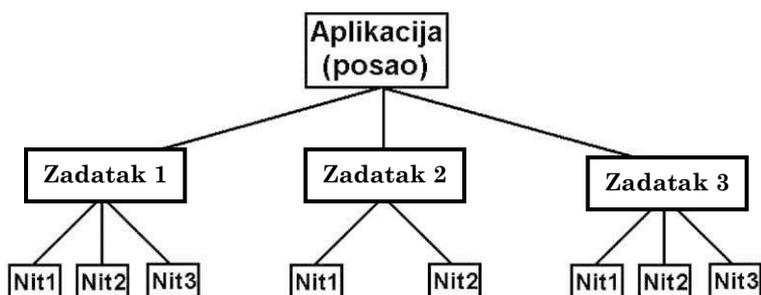
Kao u operativnom sistemu Unix, i u verzijama Windows 3.1 x, struktura Windows NT je slojevita: jezgro, upravljanje grafikom i korisnički nivo. Funkciju jezgra ima deo sistema pod nazivom NT Executive. Komponente jezgra su prikazane na slici 6.9.

Programi pod Windows NT mogu da se izvršavaju u dva osnovna režima: - režim jezgra - kad se izvršava neki sistemski program, - korisnički režim - kad se izvršavaju aplikacije korisnika.



Sl. 6.9: Komponente jezgra WINDOWS NT operativnog sistema.

Ovakva organizacija rada je slična kao i u UNIX operativnom sistemu. Posao koji treba da se obavi, deli se na zadatke ili procese. Na taj način se ostvaruje paralelno izvršavanje više delova programa ili više različitih programa. Zadaci se dele na niti (thread). Nit takođe predstavlja zadatak u okviru programa koji se izvršava paralelno sa svim ostalim zadacima u raspodeljenom vremenu (sl. 6.10).



Sl. 6.10: Podela poslova na zadatke i niti

Razlika između zadatka (proces) i niti sastoji se u sledećem: operativni sistem zabranjuje da jedan zadatak nekontrolisano ulazi u podatke drugog zadatka, već se razmena vrši isključivo preko operativnog sistema, dok je među nitima istog zadatka moguća interna razmena. Niti su segmenti jednog posla i oni dele podatke i ostale resurse. Zbog toga se nezavisne funkcije

programa, koje zahtevaju odvojene resurse, mogu realizovati samo kao poseban zadatak (proces). Svakom zadatku posebno se dodeljuje virtualni adresni prostor, jedna ili više niti i ostali resursi po zahtevu.

Programi u režimu jezgra imaju ulogu da:

- preraspodeljuju niti zadataka,
- organizuju komunikaciju između niti,
- upravljaju virtuelnom memorijom,
- upravljaju objektima,
- obezbeđuju zaštitu sistema,
- nadziru rad drugih upravljačkih programa.

U korisničkom režimu se obavlja ostali deo funkcija operativnog sistema kao što je ograničavanje sistema datoteka, formiranje veza prema različitim aplikacijama, podrška za mrežu itd.

6.5.1 Uslužni softver

Uslužni program je bilo koji program koji je specifičan za određenu primenu računara. Nasuprot tome sistemski programi su od suštinskog značaja za efikasno korišćenje računarskog sistema. Uslužne programe mogu da razvijaju korisnici računarskog sistema za rešavanje svojih problema. Zbog toga se ovi programi često nazivaju i korisnički programi.

Napomenimo još da se često u okviru aplikacionih programa mogu po potrebi da proširuju i određene funkcije operativnog sistema.

Aplikacioni programi se ponekad dele u sledeće vrste:

- pojedinačni aplikacioni programi,
- programski paketi,
- sistemi programa.

Programski paket sastoji se od više programa i predstavlja skup procedura za rešavanje problema. Sistem programa predstavlja skup programa čije su komponente usko povezane.

Windows Explorer

Windows Explorer, ili istraživač operativnog sistema je jedan od značajnih programa u Windows operativnom sistemu, a funkcija mu je upravljanje direktorijumima ili datotekama. Prozor programa je podeljen na dva dela kao što je prikazano na Slici 6.11.



Sl. 6.11: Windows Explorer

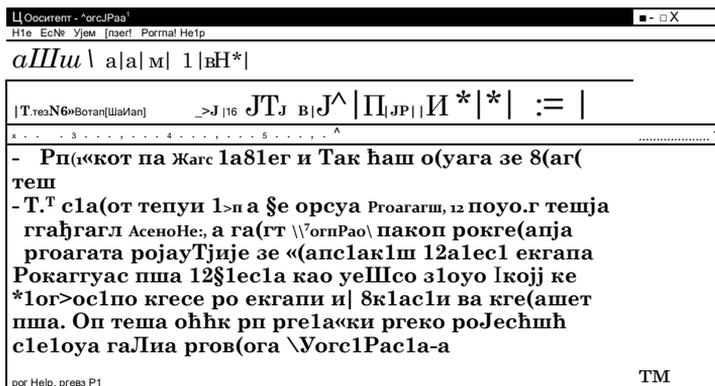
Levi panel sadrži stablo direktorijuma, tj. strukturu organizacije fajlova i podataka na memorijskom sistemu računara. Desni panel sadrži direktorijume i datoteke na disku izabranih na stablu direktorijuma.

Ispod naslovne linije koja sadrži ikonu za umanjivanje prozora, ikonu za uvećavanje prozora i ikonu za zatvaranje prozora, nalazi se linija sa padajućim menijima koji sadrže komande za upravljanje datotekama. Na primer padajući meni Edit omogućava sledeće operacije nad datotekama: otvaranje novih, brisanje starih (premeštanje u korpu za otpatke), promenu imena, kopiranje itd. O ninterfejsu sa menijima biće reči kasnije.

Kopiranje i premeštanje datoteka se može realizovati prevlačenjem datoteke, pritiskom na levi taster miša, na neku drugu datoteku ili disk. Ako se datoteka "prevuče" na drugu lokaciju ona će biti prenesena ali ne i kopirana. Premeštanje podrazumeva kopiranje na novu lokaciju i brisanje polazne datoteke na postojećoj lokaciji. Kopiranje datoteke treba naglasiti tako što se tokom prevlačenja drži pritisnut taster <Ctrl>. Datoteka koja sekopira prilikom prevlačenja ima ispisani znak + pored sebe.

Editor teksta WordPad

Korisnički program WordPad predstavlja je editor teksta koji omogućuje korisniku da izvrši formatiranje teksta: izbor fonta, podešavanje margina, prored između redova i slično. Standardno se isporučuje sa Windows operativnim sistemom. Pokretanje WordPad-a se vrši pritiskom na <Start> taster u Task baru, a zatim se biraju opcije Programs, Accessories i na kraju WordPad. Nakon pokretanja programa pojavljuje se izgled ekrana prikazan na Slici 6.12



Sl. 6.12: Radni prostor WordPad editora teksta.

Pokazivač miša izgleda kao veliko slovo I koji se slobodno

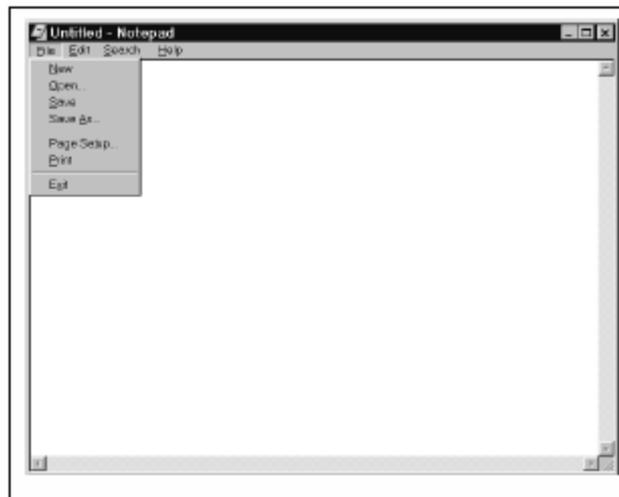
kreće po ekranu m skladu sa kretanjem miša. On menja oblik pri prelasku preko pojedinih delova radnog prostora WordPad-a.

Horizontaln lenjir je jedanod elemenata radnog prostora WordPad-a. Omogućava brzu promenu margina i uvlaka. Lenjir se može ukloniti otvaranjem <View> u glavnom meniju i uklanjnjem znaka \surd pored reči Ruler.

Pritiskom na funkcijsku tipku <F1> na ekranu se pojavljuje upustvo za rad sa tekst editorom WordPad.

Beležnica-Notepad

Tekst editor Notepad služi za unos i obrad jednostavnog teksta. Tekstovi koji se obrađuju ovim tekst editorom mogu da sadrže samo znake koji pripadaju ASCII kodnom rasporedu, bez ikakvih dodataka. To znači da u Notepad-u nije moguće pojedine delove teksta ukositi, podebljati, pisati različitom veličinom i bojom, poravnjavati i slično. Notepad se pokreće pritiskom na na <Start> taster u Task baru, a zatim se biraju opcije **Programs** , **Accessories** i na kraju **Notepad**, kao što je prikazano na Slici 6.13.



Sl. 6.13: Radni prostor WordPadeditora teksta.

Prozor nosi naziv Untitled - Notepad, što znači da tekstu koji se obrađuje nije dato ime, i rezervsan je za unos teksta. Kurzor je u obliku uspravne crte i označava mesto na kojem će se pojaviti tekst koji se unosi.

Pritiskom na funkcijsku tipku <F1> na ekranu se pojavljuje upustvo za rad sa tekst editorom WordPad.

Operativni sistem Windows sadrži memorijski deo, koji nosi naziv klipbord (Clipboard), u koji mogu da se kopiraju različiti podaci (tekst, slika, fajl isl.) iz bilo kog programa koji radi pod Windows operativnim sistemom. Takođe, podaci iz klipborda mogu se uključiti u svaki program koji radi pod Windows operativnim sistemom. U klipbord može da se čuva samo jedan podatak, a kopiranjem podataka u klipbord, predhodni podatak se briše.

Klipbord se najviše koristi pri radu sa beležnicom kao i za premeštanje podataka iz jednog programa u drugi. Komande iz Edit menija moguće je koristiti upotrebom kombinacije tipki:

- kopiranje podataka u klipbord <Ctrl> + <C>,
- kopiranje sadržaja klipborda na željeno mesto < Ctrl > + <V>
- premeštanje podata u klipbord < Ctrl > + <X>.

Navedene skraćenice su iste u svim programima koji rade pod Windows operativnim sistemom i lakše ih je upotrebljavati nego birati odgovarajuće komade u Edit padajućem meniju.

6.6 Korisnički interfejs

Korisnički interfejs računara često je kriterijum kojim se ocenjuje lakoća rada sa računarom. Interfejs koji je težak za korišćenje, u najboljem slučaju, imaće kao rezultat visoki nivo grešaka korisnika.

Korisnički interfejs sa računarom treba da reši na zadovoljavajući način sledeća dva problema:

- kako može informacija od korisnika da bude obezbeđena računam,
- kako može informacija iz računara biti prikazana korisniku.

Početakom 80-tih godina standardni uređaj za interfejs bio je "nem" alfanumerički terminal sa tastauroom i zelenim ili plavim znacima na crnoj pozadini ekrana. Korisnički interfejsi bili su tekstualni ili zasnovani na formularima (obscima, engl. formbased). Mada će interfejsi zasnovani na tekstu ostati u upotrebi još mnogo godina, korisnici sve više očekuju aplikacione sisteme koji imaju neku vrstu grafičkog interfejsa.

Prednosti grafičkog korisničkog interfejsa su sledeće:

- On se relativno lako uči i koristi. Korisnici bez praktičnog rada na računaru mogu da koriste interfejs posle kratke trening sesije;
- Korisnici imaju više prozora za interakciju sa sistemom. Prelaz od jednog zadatka na drugi moguć je bez gubljenja uvida na informaciju generisanu u toku prvog zadatka;
- Brza interakcija na celom ekranu moguća je sa neposrednim pristupom bilo gde na ekranu.

U tabeli 6.2 date su najvažnije karakteristike grafičkog korisničkog interfejsa.

Tabela 6.2: Karakteristike grafičkog korisničkog interfejsa.

Karakteristika	Opis
Prozori	Više prozora omogućava istovremeno prikazivanje na ekranu korisnika različitih informacija.
Ikone	Ikonomama se predstavljaju različiti tipovi informacija. Na nekim sistemima ikone predstavljaju datoteke, na drugim, ikone predstavljaju procese.
Meniji	Komande se selektuju iz menija umesto da se unose u komandnom jeziku
Pokazivači uređaja	Pokazivački uređaji, kao što je miš, koristi se za selekciju varijanti iz menija ili za indikaciju na ekranu elemenata koji su od interesa
Grafika	Grafički elementi mogu se kombinovati sa tekstualnim na istom displeju

6.6.1 Karakteristike korisničkog interfejsa

Korisnički interfejs mora uzeti u obzir fizička i mentalna ograničenja ljudi koji koriste računarske sisteme. Najvažnija

je, verovatno, potreba da se prepoznaju ograničenja obima kratkoročne memorije ljudi i da se izbegne prenatrpanost korisnika informacijama. Konzistentnost interfejsa označava da sistemske komande i meniji treba da imaju isti format, parametri treba da se prenose komandama na isti način, a interpunkcija komandi treba da bude slična.

Konzistentni interfejsi redukuju vreme učenja korisnika. Znanje stečeno u jednoj komandi ili aplikaciji primenljivo je na druge delove sistema. Važna je takođe konzistentnost interfejsa kroz podsisteme. Koliko god je moguće komande sa sličnim značenjem u različitim podsistemima treba da se izražavaju na isti način.

Korisnici neizbežno prave greške pri korišćenju računarskog sistema. Interfejs može minimizirati te greške (na primer, korišćenje menija znači da će biti izbegnute greške koje nastaju pri kucanju) ali greške se nikada ne mogu potpuno eliminisati. Interfejs treba da sadrži sredstva koja omogućavaju da korisnici izvrše oporavak posle svojih pogrešaka. Ovo se može ostvariti na jedan od sledeća dva načina:

1. potvrđivanje destruktivnih akcija. Ako korisnik zada akciju koja je potencijalno destruktivna, on ili ona će biti upitan da potvrdi da je to stvarno ono što je nameravano, pre nego što bilo kakva informacija bude uništena;
2. obezbeđenje undo sredstava. Komanda undo (poništi) vraća sistem u stanje pre nego što je akcija izvršena. Više nivoa undo komande su vrlo korisni jer korisnici uvek ne prepoznaju odmah da je učinjena greška. U praksi je ovo skupo za realizaciju. Zato većina sistema samo omogućava da poslednja komanda bude poništena.

Na kraju, interfejs treba da ima ugrađenu pomoć korisniku ili help sredstva. To treba da bude integrisano sa operativnim sistemom i treba da obezbedi različite nivoe pomoći i saveta. Nivoi treba da se satoje od osnovnih informacija kako početi rad sa sistemom, do potpunog opisa mogućnosti sistema. Sredstva za pomoć treba da budu strukturirana, korisnik ne sme biti "pregažen" informacijama kada on traži pomoć.

6.6.2 Direktna manipulacija

Modifikovanjem informacija prikazanih na njihovom ekranu i izdaju komande koristeći menije.

Korisnici interaguju sa prikazanim informacijama kroz direktne akcije kao što su zamena informacije, premeštanje informacije itd. Eksplicitne komande za modifikaciju informacija nisu neophodne. Modifikacije na prikazanom modelu neposredno menjaju osnovnu informaciju.

Čest primer interfejsa sa direktnom manipulacijom jeste interfejs koji obezbeđuje većina procesora reči ili ekranskih editora. U tom slučaju informacioni prostor se sastoji od niza paragrafa (odeljaka) koji su prikazani korisniku kao dokument. Za umetanje teksta kursor se postavlja na odgovarajuće mesto na ekranu i unosi se (kuca se) tekst. Tekst se neposredno pojavljuje pokazujući korisniku posledice njegovih akcija.

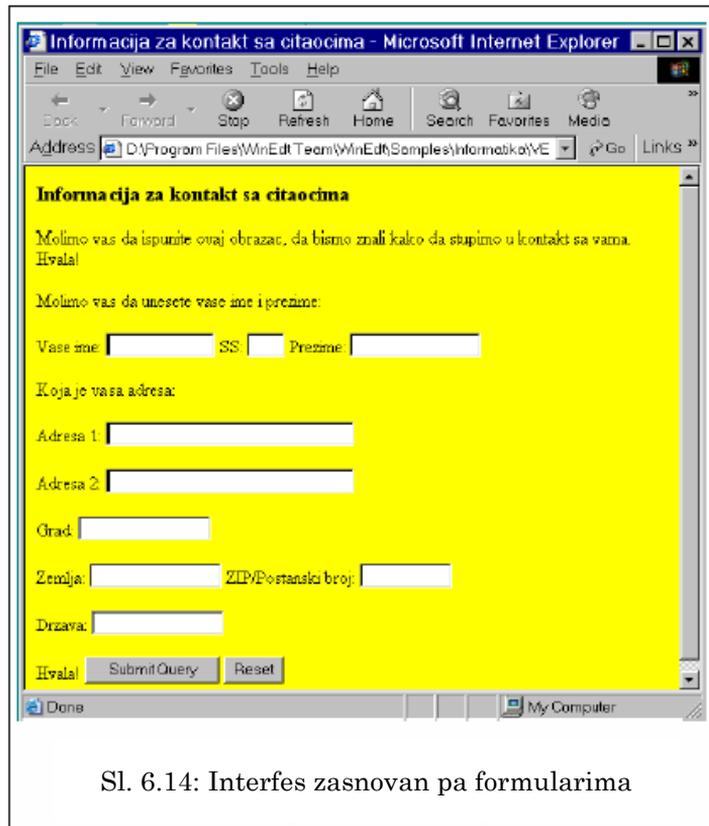
Drugi primer je grafički korisnički interfejs, gde se korisniku prikazuje lista imena datoteka. Da bi izmenio ime datoteke korisnik selektuje tekst na ekranu a zatim kuca novo ime datoteke.

Prednosti interfejsa za direktnu manipulaciju su:

- korisnici osećaju kontrolu računara i nisu njime zaplašeni,
- vreme obučavanja korisnika je relativno kratko,
- korisnici dobijaju neposrednu povratnu vezu i greške često mogu da se otkriju i isprave vrlo brzo,

Jedan od najprostijih i najlakši za razumevanje interfejsa sa direktnom manipulacijom jeste interfejs zasnovan na formularima (obascima, engl. forms). Korisniku se prikazuje formular koji treba da popuni. Polja u računaru su označena imenima koja ukazuju na informaciju koju treba obezbediti. Slika 6.14 je primer interfejsa zasnovanog na formularima koji se može koristiti za prikupljanje informacija za bibliotečki kontakt sa čitaocima.

Korisnik unosi informaciju koja se zahteva i prelazi na sledeće ulazno polje koristeći taster < Tab > ili birajući polje pomoću miša. Sa svakim poljem mogu biti povezane kontrole koje proveravaju da li su uneti podaci u korektnom obliku. Kada ima



Sl. 6.14: Interfes zasnovan pa formularima

relativno malo mogućih ulaza za neko polje, korisniku se može otvoriti meni sa mogućnostima izbora.

Interfejsi zasnovani na formularima pogodni su za primene gde korisnicima može da se da neka obuka. Oni realno nisu pogodni za povremene korisnike.

6.6.3 Modeli interfejsa

Jedan način da se ostvari konzistentnost korisničkog interfejsa je da se definiše konzistentni model ili metafora za interakciju (međusobno delovanje) korisnik-računar. Model korisničkog interfejsa treba da bude analogan nekom modelu realnog sveta koji korisnik razume.

Najpoznatija metafora je metafora radne oblasti (radne površine, radnog prostora, engl. desktop) gde korisnički ekran predstavlja radnu oblast. Entiteti sistema predstavljeni su na ekranu pomoću ikona.

Metafora radne oblasti nije pogodna za podršku kompleksnih sistema interakcije koje nude aplikacije kao što su editori, sistemi baza podataka itd. Ovi sistemi održavaju metaforu radne oblasti do izvesnog stepena, ali je dopunjuju dodavanjem upravljačkog panela (dijalog prozora, engl. control panel) koji je grafička reprezentacija sistema komandi. Ovaj upravljački panel može biti prosta pruga (linija) ikona, gde svaka ikona predstavlja određenu komandu. Pritisak (engl. clicking) na ikonu komande ekvivalentan je izboru komande iz menija ili unošenju (kucanju) njenog imena na tastaturi. Složeniji upravljački paneli mogu uključivati dodatne objekte korisničkog interfejsa kao što su polja za prikaz, klizači, indikatori itd.

Upravljački paneli mogu sadržati (uključivati) nekoliko vrsta entiteta za podršku interakcije sa korisnikom:

1. Tasteri (dugmad, engl. buttons). - Viranje tastera (pritisak na taster) ima za posledicu iniciranje (pokretanje jedne akcije);
2. Prekidači (engl. switches). - To može biti skup pozicija za konfigurisanje sistema ili prevođenje sistema iz jednog stanja u drugo;
3. Meniji (engl. menus). To je skup tastera ili prekidača koji se mogu učiniti vidljivim i mogu se birati (selektovati). Izbor u meniju nekog naziva izaziva pojavljivanje novog padajućeg menija;
4. Indikatori (engl. indicators). - Oni se aktiviraju da pokažu da je neka akcija u toku;
5. Displeji ili polja za prikaz (engl. displays). - To su oblasti na panelu gde se može prikazati tekstualna ili grafička informacija. Polja imaju ime i vrednost;
6. Klizači (engl. sliders). - To su ulazni uređaji koji se koriste za postavljanje ulazne vrednosti iz nekog opsega. Slični su klizačima koji kontolišu hardver, kao što je kontrola jačine zvuka na stereo sistemu. Korisnik povlači klizač duž skale (opsega) da bi postavio potrebnu vrednost.

6.6.4 Sistemi sa menijima

U meni interfejsu (interfejsu pomoću menija) korisnici biraju jednu od mogućnosti da bi izdali komandu računaru. Korisnici mogu kucati (uneti) ime ili identifikator izbora, mogu pokazati na nju pomoću miša ili nekog drugog pokazivačkog uređaja, mogu koristiti tastere za pomeranje kursora da bi postavili kursor na njih.

Sistemi zasnovani na menijima imaju nekoliko prednosti:

- Korisnici ne moraju da znaju imena komandi. One su uvek prikazane na listi važećih komandi. Imena komandi koja imaju značenje, kao što je "Save" sadrže u sebi, tj. impliciraju funkciju komande;
- Napor kucanja je minimalan. Ovo je posebno važno za povremene korisnike sistema koji ne mogu da brzo kucaju;
- Izbegavaju se neke vrste grešaka korisnika. Nevažne opcije u meniju može zabraniti sistem. Sintaksne greške u komandama nikada se ne čine;
- Može se obezbediti kontekstno zavisna pomoć. Sa sistemom menija lako se može čuvati trag korisnikovog konteksta i on se može povezati sa help sistemom (sistemom za pomoć);

Meniji mogu biti padajući (engl. pop-down) i pojavni (koji se javljaju, podizni, engl. pop-up). Padajući meniji prikazuju naziv menija. Njegovom selekcijom otvara se naniže meni sa komandama koje se mogu birati. Pojavni meniji pridruženi su entitetima (kao što je polje u formularu). Selekcijom entiteta a zatim pritiskom na taster miša izaziva se pojavljivanje menija.

Glavni problem sa meni interfejsom je potreba strukturiranja velikih menija. U nekim slučajevima može biti desetina, stotina ili čak hiljada mogućih izbora iz menija. Oni moraju biti organizovani tako da se prikazuju u razumnim porcijama (fragmentima).

Postoji nekoliko načina da se ovaj problem savlada (reši):

1. Pomični meniji, klizeći meniji (engl. scrolling menus). Kada izbor nije prikazan na meniju, on se pomera da bi se prikazao sledeći skup izbora. Ovo je nepraktično ako ima hiljadu izbora iz menija;

2. Hijerarhijski meniji (engl. hierarcical menus tepiz). Meniji se organizuju u hijerarhiju. Selekcija stavke menija izaziva (ima za posledicu) zamenu tekućeg menija drugim menijem koji predstavlja njegovo podstablo. Hijerarhijski meniji mogu upravljati vrlo velikim brojem izbora, ali oni su teški za navigaciju (vođenje) po izborima. Korisnici se takođe često gube u hijerarhiji menija.
3. Šetajući meniji (engl. walking menus). Šetajući meniji su vrsta hijerarhijskih menija koji su mogući ako je stablo plitko (male dubine). Kada je selektovana stavka menija, to izaziva prikazivanje sledećeg menija koji je susedan sa njim. Ovo je izvodljivo ako ima desetine umesto stotina izbora koje treba prikazati.
4. Pridruženi upravljački paneli (engl. associated control panels). Ovo je različit oblik hijerarhijskih menija gde su podmeniji predstavljeni kao upravljački paneli. Pritisak na taster za komandu menija izaziva pojavljivanje upravljačkog panela koji otvara nove opcije. Ovaj način se koristi u mnogim sistemima za obradu reči.

6.6.5 Komandni interfejsi

Komandni interfejsi (interfejsi preko komandne linje) zahtevaju od korisnika da unese (otkuca) sistemu tekst komande. Komanda može biti upit, zahtev za neku uslugu ili ona može biti sekvenca drugih komandi. To je bio prvi tip interfejsa koji su nudili interaktivni sistemi, jer je on mogao da se realizuje korišćenjem relativno jeftinog alfanumeričkog displeja i ograničene procesne moći. Vrlo veliki broj sistema razvijen je sa interfejsima komandne linije (engl. command-line interfaces) i nastavlja se njihovo korišćenje. Korisnici nekih operativnih sistema više vole interfejse komandne linije jer oni omogućavaju bržu interakciju nego grafički korisnički interfejsi.

Prednosti i nedostaci komandnih interfejsa su:

- Tehnike obrade jezika dobro su razvijene zbog radova obavljenih u tehnici kompilatora. Kreiranje komandnog jezika procesora je obično mnogo lakše nego realizacija grafičkog korisničkog interfejsa;

- Međutim, korisnici moraju učiti komandni jezik koji je ponekad vrlo složen. U nekim slučajevima (kao što je jezik Unix ljske) neki korisnici nikada ne nauče potpuni jezik;
- Komande skoro proizvoljne složenosti mogu se kreirati kombinovanjem pojedinačnih komandi. Moćna osobina Unix sistema je njegova sposobnost pisanja programa na komandnom jeziku;
- Međutim, korisnici neizbežno čine greške pri unošenju komandi. To zahteva sredstva za rukovanje greškama i generisanje poruka koja treba da budu uključena u procesor komandnog jezika;
- Interfejs se može uraditi konciznim sa malo napora pri kucanju od strane korisnika;
- Međutim, interakcija sa sistemom je preko tastature. Interfejs ne može da koristi u potpunosti pokazivačke uređaje kao što je miš.

Komandni interfejsi nisu pogodni za povremene i neiskusne korisnike. Vreme potrebno za učenje komandnog jezika je nesrazmerno vremenu koje se troši na interakciju sa računarom. Za takve korisnike interfejs zasnovan na menijima (ili, možda, interfejs na prirodnom jeziku) je jedini prihvatljiv stil interfejsa.

Iskusni, stalni korisnici računara, ponekad više vole komandni interfejs jer opšti troškovi uključeni u selekciju menija iritiraju ih i usporavaju njihov rad. Komandni interfejsi omogućavaju bržu interakciju sa računarom i uprošćavaju unos složenih zahteva. Iskusni korisnici takođe mogu po želji kombinovati komande u procedure i programe.

Naravno, komandni interfejsi (interfejsi komandnog jezika) i meni interfejsi (interfejsi zasnovani na menijima) nisu uzajamno isključivi. Mnogi veliki softverski sistemi moraju se prilagoditi raznim vrstama korisnika, od iskusnih računarskih profesionalaca do povremenih korisnika bez računarskog predznanja. Neki sistemi koji imaju standardni meni interfejs takođe obezbeđuju prost komandni jezik (ponekad se on naziva prečice sa tastature) koji omogućava da se komande unose bez njihovog biranja iz menija. Potpuna funkcionalnost komandnog jezika može se ponuditi stalnim iskusnim korisnicima, a prostiji meni interfejs ponuditi manje iskusnim i povremenim korisnicima.

6.6.6 Vođenje korisnika (help sistem)

Ne1r sistem (sistem pomoći) je jedan aspekt korisničkog interfejsa, naime priprema (obezbeđenje) vođenja korisnika koje pokriva dve oblasti: 1. poruke koje priprema sistem kao odgovor na korisnikove akcije, 2. on-line help sistem, Obično postoji razlika između obezbeđenja pomoći (koju zahteva korisnik) i izlaza poruka (asinhrono pripremanih sistemom). Međutim, oni su u suštini različiti aspekti jedinstvenog sistema vođenja korisnika.

6.6.7 Poruke o greškama

Prvi utisak koji korisnici mogu imati o softverskom sistemu su poruke sistema o greškama. Neiskusni korisnici mogu početi rad, činiti početnu grešku i neposredno moraju razumeti poruku o greški koja je posledica toga.

Poruke o greškama treba da uvek budu učtive, koncizne, konzistentne i konstruktivne. One ne smeju biti uvredljive i ne treba da imaju pridružene zvušne signale ili druge šumove koji mogu zbuniti korisnika. Gde god je moguće, poruka treba da sugeriše kako se greška može ispraviti. Poruka o greški treba da bude povezana sa kontekstno osetljivim on-line help sistemom.

6.7 Operativni sistem Linux

Operativni sistem je program koji upravlja hardverom i softverom računara za korisnika. Operativni sistemi su prvobitno projektovani tako da obavljaju hardverske zadatke koji se ponavljaju, što se uglavnom odnosilo na upravljanje datotekama, izvršavanje programa i primanje komandi od korisnika. Sa operativnim sistemom omogućuje da primi i protumači instrukcije koje je poslao korisnik. Treba samo da pošaljete naredbu operativnom sistemu da obavi zadatak, kao što je čitanje datoteke ili štampanje dokumenta. Korisnički interfejs operativnog sistema može biti tako jednostavan, kao što je unošenje komandi u liniji, ili tako složen, kao što je odabiranje menija i ikona na radnoj površini.

Operativni sistem takođe upravlja namenskim programima. Za

obavljanje različitih zadataka, kao što je uređivanje dokumenata ili vršenje proračuna, potrebne su specifične softverske aplikacije. Editor je primer softverske aplikacije koja omogućuje uređivanje dokumenta pravljjenjem izmena i dodavanjem novog teksta. Sam editor je program koji se sastoji od instrukcija koje računar treba da izvrši. Da bi se koristio, program se prvo mora učitati u memoriju računara i onda se njegove instrukcije izvršavaju. Operativni sistem kontroliše učitavanje i izvršavanje svih programa, uključujući sve softverske aplikacije. Kada se želi korišćenje editora, treba naložiti operativnom sistemu da učitava aplikaciju editora i da je izvrši.

Upravljanje datotekama, upravljanje programima i interakcija sa korisnikom predstavljaju tradicionalne mogućnosti zajedničke za sve operativne sisteme. Linux, kao i sve verzije Unixa, dodaje još dve mogućnosti. Linux je višekorisnički sistem koji ima mogućnost višeprocenane obrade (engl. multitasking system), odnosno može da obavlja više zadataka istovremeno. Dok se jedan zadatak izvršava, može se raditi na drugom. Na primer, može se menjati jedna datoteka dok se druga datoteka štampa. Ne treba čekati da se završi štampanje druge datoteke da bi uređivali prvu. Vudući da je Linux višekorisnički sistem, nekoliko korisnika se mogu prijaviti na sistem u isto vreme, pri čemu svaki ostvaruje interakciju sa sistemom preko sopstvenog terminala.

Kao verzija Unixa, Linux deli fleksibilnost tog sistema; fleksibilnost koja vuče korene u istraživačkim počecima Unixa. Unix sistem, koji je Ken Thompson razvio u AT&T Bell laboratorijama kasnih šezdesetih i ranih sedamdesetih godina dvadesetog veka, obuhvatio je mnoga nova razvojna dostignuća u projektovanju operativnih sistema. Prvobitno, Unix je projektovan kao operativni sistem za istraživače. Jedan od glavnih ciljeva bio je da se napravi sistem koji bi podržavao njihove promenljive zahteve. Da bi se ovo postiglo, Thompson je morao da oblikuje sistem koji bi mogao da izlazi na kraj sa mnogo različitih vrsta zadataka. Fleksibilnost je postala važnija od efikasnosti hardvera. Kao i ostale verzije Unixa, Linux ima tu prednost da je u stanju da se nosi sa mnoštvom zadataka sa kojima bi se bilo koji korisnik mogao suočiti. Korisnik nije osuđen na ograničene i krute interakcije sa operativnim sistemom. Umesto toga, operativni sistem se posmatra kao skup visoko efikasnih alatki koje su korisniku

stavljene na raspolaganje. Ova filozofija orijentisana na korisnika znači da se sistem može konfigurisati i programirati tako da zadovolji specifične potrebe. U slučaju Linuxa, operativni sistem postaje operativno okruženje.

Linux je trenutno zaštićen autorskim pravima u okviru javne licence GNU koju daje Free Software Foundation i često se pominje kao GNU softver. GNU softver se distribuira besplatno, pod uslovom da se i drugima distribuira besplatno. GNU softver se pokazao kao pouzdan i efikasan.

6.7.1 Istorijat Linuxa

Istorijat Linuxa, kao verzije Unixa, počinje prirodno sa Unixot. Priča počinje kasnih šezdesetih, kada se pojavilo jedno združeno nastojanje da se razviju nove tehnike operativnih sistema. Konzorcijum istraživača iz General Electrica, AT&TBell laboratorija i MIT-a Massachusetts Institute of Technology je 1968. godine sproveo poseban istraživački projekt u vezi sa operativnim sistemima nazvan MULTICS (engl. Multiplied Information and Computing ervice). MULTICS je obuhvatio mnoge nove koncepte u vezi sa istovremenim izvršavanjem više zadataka, upravljanjem datotekama i interakcijom sa korisnikom.

Prvobitno projektovan specijalno za personalne računare zasnovane na Intel-u, Linux je nastao kao lični projekt studenta računarstva pod imenom Linux Torvalds na helsinškom univerzitetu. U to vreme, studenti su upotrebljavali program pod imenom Minix, koji je sadržao različite mogućnosti Unixa. Minix je napravio profesor Andrew Tenenbaum i bio je široko distribuiran preko Interneta studentima širom sveta. Linusova namera bila je da se napravi efikasna PC verzija Unixa za korisnike Minxa. Ta verzija nazvana je Linux, a 1991. je Linus objavio verziju 0.11. Linux je široko distribuiran preko Interneta i sledećih godina su ga drugi programeri prečišćavali i dodavali mu delove, uključujući većinu aplikacija i mogućnosti koje danas nalazimo u standardnim Unix sistemima. Svi glavni moduli za upravljanje prozorima uneseni su u Linux. Linux ima sve alatke za umrežavanje, kao što su podrška za GTP transfer datoteka, veb pretraživači i ceo spektar mrežnih servisa, kao što su elektronska pošta, servis imena domena (DNS) i dinamička konfiguracija matičnog

servera, uz FTP i veb servere i servere za štampanje. On takođe poseduje potpun skup uslužnih modula za razvoj programa, kao što su C++ kompajleri i moduli za pronalaženje grešaka. I uza sve svoje mogućnosti, operativni sistem Linux ostaje mali, stabilan i brz. U svom najjednostavnijem formatu, Linux se efektivno može izvršavati na samo 2 MB memorije.

Iako se Linux razvio u slobodnom i otvorenom okruženju Interneta, on podleže zvaničnim Unix-ovim standardima. Usled umnožavanja verzija Unixa u prošlim decenijama, IEEE (Institute of Electrical and Electronics Engineers) je razvio nezavisan Unix standard prihvaćen od ANSI-ja (American National Standards Institute). Ovaj novi Unix sa ANSI standardom nazvan je POSIX (Portable Operating System Interface for Computer Environments). Standard definiše kako sistem zasnovan na Unixu treba da funkcioniše, određujući detalje poput sistemskih poziva i interfejsa. POSIX definiše univerzalni standard koji sve verzije Unixa moraju slediti. Najpopularnije verzije Unixa sada podležu POSIX-u. Linux je od početka razvijan u skladu sa POSIX standardom. Linux takođe podleže Linux FHS (file system hierarchy standard) standardu, koji određuje mesto datoteka i direktorijuma u Linux-ovoj strukturi datoteka.

6.7.2 Pregled Linuxa

Kao i Unix, Linux se uopšteno može podeliti na tri glavne komponente: jezgro, okruženje i strukturu datoteka. Jezgro je osnovni program koji izvršava programe i upravlja hardverskim uređajima, kao što su diskovi i štampači. Okruženje obezbeđuje korisnički interfejs. Ono prima komande od korisnika i šalje te komande jezgru na izvršavanje. Struktura datoteka određuje način na koji se datoteke čuvaju na uređaju za skladištenje, kao što je disk. Datoteke se organizuju u direktorijume. Svaki direktorijum može da sadrži bilo koji broj poddirektorijuma, od kojih svaki može da sadrži i datoteke. Jezgro, okruženje i struktura datoteka zajedno čine osnovnu strukturu operativnog sistema. Navedene tri komponente omogućavaju izvršavanje programa, upravljanje datotekama i ostvarivanje interakcije sa sistemom.

Okruženje obezbeđuje interfejs između jezgra i korisnika. Ono se može opisati kao tumač zahteva. Takav interfejs interpretira komande koje je uneo korisnik i šalje ih jezgru. Linux pruža neko

liko vrsta okruženja: radne površine, module za upravljanje prozorima i ljske za tekstualno zadavanje komandi. Svaki korisnik na Linux sistemu ima svoj sopstveni korisnički interfejs. Korisnici mogu da oblikuju svoja okruženja prema svojim posebnim potrebama, bilo da su to ljske, moduli za upravljanje prozorima ili radne površine. U ovom smislu, za korisnika operativni sistem funkcioniše više kao operativno okruženje, koje korisnik može da kontroliše.

Kod Linuxa su datoteke organizovane u direktorijume, kao kod Windows-a. Celokupan Linuxou sistem datoteka jeste jedan veliki skup međusobno povezanih direktorijuma, od kojih svaki sadrži datoteke. Neki od direktorijuma su standardni direktorijumi rezervisani za systemske potrebe. Mogu se praviti sopstveni direktorijumi za sopstvene datoteke, kao i da se lako premeštaju datoteke iz jednog direktorijuma u drugi. Mogu se čak premeštati celi direktorijumi i da se dele direktorijumi i datoteke sa drugim korisnicima na jednom sistemu. Kod Linuxa takođe mogu da postavljaju dozvole pristupa nad direktorijumima i datotekama, omogućujući drugima da im pristupe ili ograničavajući pristup tako da im samo jedan korisnik može pristupiti. Direktorijumi svakog korisnika su, u osnovi, povezani sa direktorijumima drugih korisnika. Direktorijumi su uređeni u hijerarhijsku strukturu stabla, čiji je početak koreni direktorijum. Svi drugi direktorijumi su, na kraju, izvedeni iz ovog prvog, korenog direktorijuma.

Sa okruženjima KDE (K Desktop Environment,) i GNOME (GNU Network Object Model Environment) Linux ima potpuno inegrirani GUI (Graphical User Interface) interfejs. Sve Linux operacije mogu se obavljati sa bilo kog od ova dva interfejsa. KDE i GNOME su potpuno operativne radne površine koje podržavaju operacije prevlačenja i spuštanja, omogućujući prevlačenje ikona na radnu površinu i postavljanje sopsvenih menija u panelu Applications. Oba se oslanjaju na X Windows System, što znači da dokle god su oba instalirana na sistemu, aplikacije sa jedne radne površine mogu se izvršavati na drugoj.

6.7.3 Pristupanje Linux sistemu

Operativni sistem Linux se može posmatrati kao sistem sa dva različita nivoa, pri čemu se jedan izvršava iznad drugog. Prvi nivo se odnosi na pokretanje Linux sistema, kada se sistem učitava sa diska u operativnu memoriju i izvršava. On ima kontrolu nad računarom i perifernim uređajima. Međutim, u tom trenutku ne postoji mogućnost za ostvarivanje interakcije sa njim. Tek nakon što se Linux uspešno pokrene, on prikazuje prozor za prijavljivanje čekajući da se korisnik prijavi na sistem i počne da ga koristi. Ne može se dobiti pristup Linux-u bez prijave.

Prijavljivanje na sistem i korišćenje Linuxa predstavlja sledeći nivo. Sada se mogu zadavati komande koje Linux-u nalažu da obavi određene zadatke. Mogu se koristiti uslužni moduli i programi, kao što su editori, kompajleri, ili video igre. U zavisnosti od izbora koji je napravljen prilikom instalacije, interakcija sa sistemom može da se ostvari ili preko jednostavnog interfejsa komandne linije ili preko radne površine. Dakle, prijavljivanje se može obaviti ili u tekstualnom režimu, preko odzivnika komandne linije, ili u grafičkom modu preko odgovarajućeg prozora.

Za obaranje Linux sistema treba odabrati meni System kako bi se prikazala opcija Reboot ili Shutdown. Za obaranje sistema treba odabrati stavku Shutdown. Alternativno, sistem se može oboriti prilikom odjavljivanja sa radne površine GNOME. GNOME će prikazati prozor za odjavljivanje sa opcijama Logout, Shutdown ili reboot.

6.7.4 Upravljanje datotekama i arhivama

Kod Linuxa su sve datoteke uređene u direktorijume, koji su, dalje, hijerarhijski povezani jedan za drugim u jednu sveukupnu strukturu datoteka. Datoteka se se referencira ne samo prema svom imenu, već i prema svom mestu u ovoj strukturi datoteka. Linux-ove komande za rad sa datotekama mogu da obave sofisticirane operacije, kao što je premeštanje ili kopiranje celokupnih direktorijuma sa njihovim poddirektorijumima. Za rad sa datotekama mogu se koristiti operacije nad datotekama, kao što su **find**, **cp**, **mv** i **ln**, za pronalaženje, kopiranje, premeštanje ili pravljenje

veze iz jednog direktorijuma u drugi. Ove komande za upravljanje datotekama koriste se u komandnoj liniji ljsuke. Međutim, bilo da se koristi komandna linija bilo grafički interfejs, osnovna struktura datoteka je ista.

Jedna od primarnih funkcija operativnog sistema je je upravljanje datotekama. Linux operativni sistem sadrži skup komandi koje obavljaju osnovne operacije upravljanja datotekama, kao što je prikazivanje spiska datoteka, prikazivanje njihovog sadržaja, kopiranje, preimenovanje i brisanje datoteka. Imena ovih komandi obično čine skraćeni oblici reči. Na primer komanda **ls** je skraćeni oblik engleske reči *list* i daje spisak datoteka u direktorijumu. Komande **cat**, **less** i **more** prikazuju sadržaj datoteka na ekranu.

Komada **cat**, je skraćenica engleske reči "concatenate".

Komanda

\$ cat mydata

daje kao izlaz celokupan tekst datoteke na ekranu, i to odjednom. Ovo predstavlja problem kod već datoteka, jer tekst brzo prođe ekranom. Komande **more** i **less** oblikovane su tako da se ovo ograničenje prevaziđe, prikazivanjme jednog po jednog ekrana teksta. Za prelazak na sledeći ekran treba pritisnuti taster **< F >** ili razmaknicu. Za povratak nazad služi taster **< B >**. Za napuštanje prikaza treba pritisnuti taster **< Q >**.

Uslužni moduli mtools

Linux sistem obezbeđuje skup uslužnih modula, poznatih pod imenom *mtools*, koji omogućavaju jednostavan pristup disketama, čvrstim diskovima formatiranim za MS-DOS sistem.

Komanda **mcopy** omogućava kopiranje datoteke sa MS-DOS diskete u disketnoj jedinici ili sa Windowsprtcije čvrstog diska na Linux sistem datoteka i obrnuto. Za navedena kopiranja nisu potrebne nikakve posebne operacije. Drugim rečima, uz module **mtools** nije potrebno montirati MS-DOS particiju da bi smo joj pristupili. U slučaju MS-DOS diskete, treba staviti disketu u disketnu jedinicu, a zatim upotrebiti **mtool** komande da bi smo pristupili ovim datotekama. Na primer, da bismo kopirali datoteku sa MS-DOS diskete na Linux sistem, treba upotrebiti ko

mandu **mcopy**. MS-DOS disketa se označava sa **a:** za jedinicu A. Za razliku od običnih DOS putanja, putanje koje se upotrebljavaju uz **mstools** komande koriste se kose crte unapred umesto kosih crta unazad backslash. Na primer, direktorijum **docs** na jedinici A bio bi refernciran putanjim **a:/docs**, a ne **a:\docs**. Za razliku od MS-DOS-a, kod koga je drugi argument porazumevao tekući di-rektorijum, kod Linux-a je uvek potrebno dati drugi argument za **mcopy**. U sledećim primerima datoteka **mydate** se kopira na MS-DOS disketu, a potom se datoteka **preface** kopira u tekući Linux direktorijum.

```
$ mcopy mydate a:
```

```
$ mcopy a:/preface . -4
```

S obzirom na razlike u načinu na koji DOS i Linux obrađuju znakove za novi red u tekstualnim datotekama, treba koristiti opciju **-4** kad kod se kopira DOS teksualna datoteka na Linux particiju.

Arhiviranje i komprimovanje datoteka

Arhive se koriste za pravljenje rezervnih kopija datoteka ili za smeštanje datoteka u paket, koji se potom može prenositi kao jedna datoteka preko Interneta. Standardni uslužni moduli za arhiviranje koji se koriste na Linux i Unix sistemima je **4ag**, za koji postoji nekoliko klijentskih grafičkih interfejsa. Pomoću programa **tar** mogu se arhivirati određene datoteke, da se ažuriraju unutar arhive i da im se dodaju nove datoteke. Mogu se arhivirati celokupni direktorijumi sa svim svojim datotekama i poddirektorijumima, koji se mogu povratiti iz arhive. Mogu se napraviti arhive na bilo kom uređaju, npr. disketi.

Sintaksa komande **tar**, sa opcijom **G** koja zadaje modulu **tar** da arhivira datoteke na određeni uređaj ili u određenu datoteku, je

```
$ tar orsje G ime-arhive.tar imena-direktorijuma-i-datoteka
```

Za pravljenje arhive treba koristiti opciju **s**. Kombinovana sa opcijom **G**, opcija **s** pravi arhivu u datoteci ili na uređaju. Ova opcija se navodi ispred opcije **G**. U sledećm primenu direktorijum **mydir** i svi njegovi poddirektorijumi se skladište u datoteku **myarch.tar**. Za prikazivanje prikazivanje svaki datoteke tokom

njenog arhiviranja koristi se opcija **v**. U ovom primeru direktorijum **mydir** ima dve datoteke **mymeeting** i **party**, kao i direktorijume pod nazivom **reports** koji ima tri datoteke **weather**, **monday** i **friday**

```
$ tar cvf myarch.tar mydir
```

```
mydir
```

```
mydir/reports/
```

```
tuSk/reports/weather
```

```
mydir/reports/monday
```

```
mydir/reports/friday
```

```
mydir/mymeeting
```

```
mydir/party
```

Korisnik kasnije može da izdvoji direktorijume sa arhivskog medijuma pomoću opcije **x**. Opcija **xf** vadi datoteke iz arhivske datoteke ili sa uređaja. Operacija ekstrakcije koju **tar** sprovodi generiše i sve poddirektorijume. U sledećem primeru opcija **xf** nalaže komandi **tar** da izvadi sve datoteke i poddirektorijume iz datoteke **myarch.tar**.

```
$ tar xvf myarch.tar
```

Operacija **tar** ne vrši kompresiju arhiviranih datoteka. Za kompresiju datoteka može se birati između nekoliko programa, uključujući GNU **zip**.

ZIP je uslužni modul za kompresiju i arhiviranje napravljen po uzoru na PKZIP, koji je prvobitno korišćen na DOS sistemima. ZIP je uslužni program za više platformi koji se koristi na Windows, Unix i Linux sistemima. Datoteka se komprimuje pomoću komade **zip**. Ona formira zip datoteku za sufiksom **.zip**. Ako se ne navede nijedna datoteka, **zip** šalje komprimovane komande na standardni izlaz. Takođe može da se koristi argument **-r** da bi komanda **zip** čitala podatke sa standardnog ulaza. Za kompresiju direktorijuma treba navesti opsiju **-r**. U prvom primeru datoteka se **mydata** se arhivira i komprimuje

```
$ zip mydata.zip mymeeting party
```

U sledećem primeru se komprimuje direktorijum **reports**:
zip -r reports

Za dekomprimovanje zip datoteke, koristi se komada **unzip**

\$ unzip mydata.zip

6.8 Pitanja za proveru znanja

1. Primeniti alate Disc cleanup, Chek disk i Disk Defragment za sređivanje fajlova na disketi.
2. Podešavanje Windows operativnog sistema vrši se sistemskom aplikacijom Control Panel. Primenom ove aplikacije podesiti vreme i datum na računaru.
3. Pomoću programa za obradu slika Paint, nacrtati kvadrat, trougao i krug. Obojiti ih različitim bojama i ispod njih napisati njihove nazive.
4. Objasniti šta se postiže sledećim opcijama iz Edit menija u programu za obradu slika Paint: Undo, Cut, Copy i Paste.
5. Klipbord je deo operativnog sistema Windowsu koji mogu da se prebace podaci (tekst, slika ili neki drugi fajl) iz bilo kog programa. Takođe podaci iz klipborda mogu se uključiti u svaki program koji radi pod operativnim sistemom Windows. Uz pomoć programa Notepad kreirati tekstualnu datoteku **dat.txt**, a zatim je kopirati u novu datoteku **dat-nova.txt**
Uputstvo: Kopirati datoteku **dat.txt** u klipbord kombinacijom tipki $\langle Ctrl \rangle + \langle C \rangle$, a zatim aktivirati novi prozor Untitled-Notepad-Notepad i preneti sadržaj klipborda kombinacijom tipki $\langle Ctrl \rangle + \langle V \rangle$. Sadržaj memorisati kao **datnova.txt**.
6. Windows Explorer jedan je od najvažnijih korisničkih programa u tom operativnom sistemu, a funkcija mu je upravljanje direktorijumima ili datotekama. Uz pomoć Windows Explorer-a kreirati na disketi stablo direktorijuma.
Uputstvo: U panelu Tree Explorera izabrati 3½Floppy (A:), a zatim redom: File, New, Folder i upisati ime novog direktorijuma, zatim pritisnuti taster $\langle Enter \rangle$.

7. Programom **zip** izvršiti kompresiju datoteka **dat1.txt**, **dat2.txt** i **dat3.txt** koje se nalaze na disketi. *Uputstvo:* Aktivirati program MS-DOS Prompt, dvostrukim klikom na ikonu. U komandnoj liniji upisati **A:**, a zatim **dir** da bi se videlo šta se nalazi na disketi. Kompresiju datoteka izvršiti komandom **zip**

A:\ zip dat.zip dat?.*

Glava 7

Računarske komunikacije

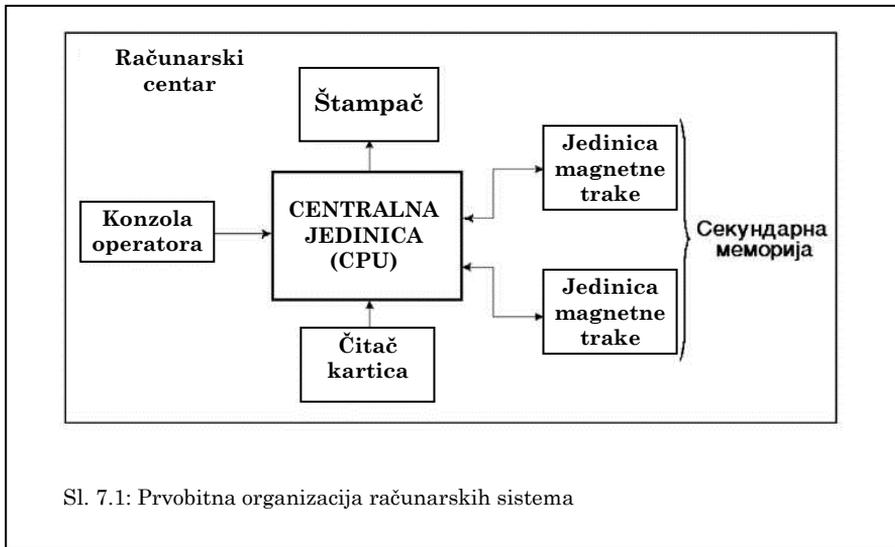
Nesporno je da su prošli vek obeležile komunikacije. Integrisanjem komunikacionih sistema u multimedijalne sisteme koje omogućavaju brzi prenos zvuka, dokumenata, podataka, digitalizovanih slika, animacija i televizijske slike, svet postaje globalno selo, jako blizu vizijama iz filmova i romana naučne fantastike.

Računari imaju bitnu ulogu u savremenim komunikacijama pa će stoga u ovom poglavlju biti reči o najznačajnijim pojmovima vezanim za mreže računara i Internet kao globalnu računarsku mrežu.

7.1 Računarske mreže

Za rani period korišćenja računara u okviru poslovnih sistema za obradu podataka karakteristično je da su to bili sistemi sa jako skupim hardverom i relativno primitivnim softverom. Tipični računski centar jedne radne organizacije bio je smešten u posebnoj prostoriji sa klimatizacijom i obuhvatao je centralnu jedinicu sa ograničenom operativnom memorijom (RAM), određeni broj jedinica sekundarne memorije (trake), štampač, čitač kartica i konzolu operatera, kao što je prikazano na Slici 7.1

Rad u ovim centrima bio je organizovan tako što su korisnici pripremali programe i podatke obično u prostorijama van ovog samog računskog centra (off-line), a kasnije je operater ubacivao te programe u računar. Ovaj koncept obrade poznat je i kao

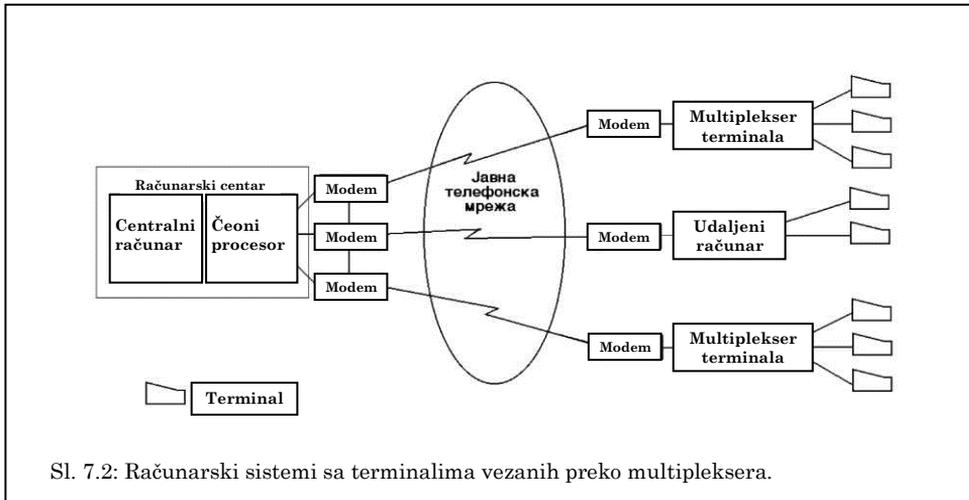


paketna obrada jer je obično više programa organizovano u pakete. Paketima su dodeljivani prioriteti na osnovu kojih je regulisan redosled njihovog izvršavanja.

Sa razvojem tehnologije rastu i mogućnosti računara, kako hardverske, tako i softverske. Pojavljuju se višekorisnički operativni sistemi koji omogućavaju da više korisnika istovremeno, u režimu multiprogramskog rada, koristi isti računarski sistem. Korisnici pristupaju centralnoj jedinici preko terminala. U početku su terminali obično bili smeštenu u okviru samog računskog centra. Kasnije se razvija koncept obrade u okviru koje su terminali udaljeni od računskog centra, a priključuju se na centralnu jedinicu računara preko telefonske mreže i modema (uređaja koji vrši konverziju digitalnog signala u analogni signal kakav se može prenesti preko mreže i obrnuto).

Kako u ovom periodu na cenu bitno utiče cena komunikacije preko telefonskih linija, da bi sistemi bili racionalnije korišćeni i ukupna cena obrade podataka manja, razvijaju se multiplek-serski uređaji i kontroleri koji omogućavaju povezivanje više terminala na jedan modem, odnosno jednu telefonsku liniju. Tipična organizacija računarske mreže prikazana je na slici 7.2.

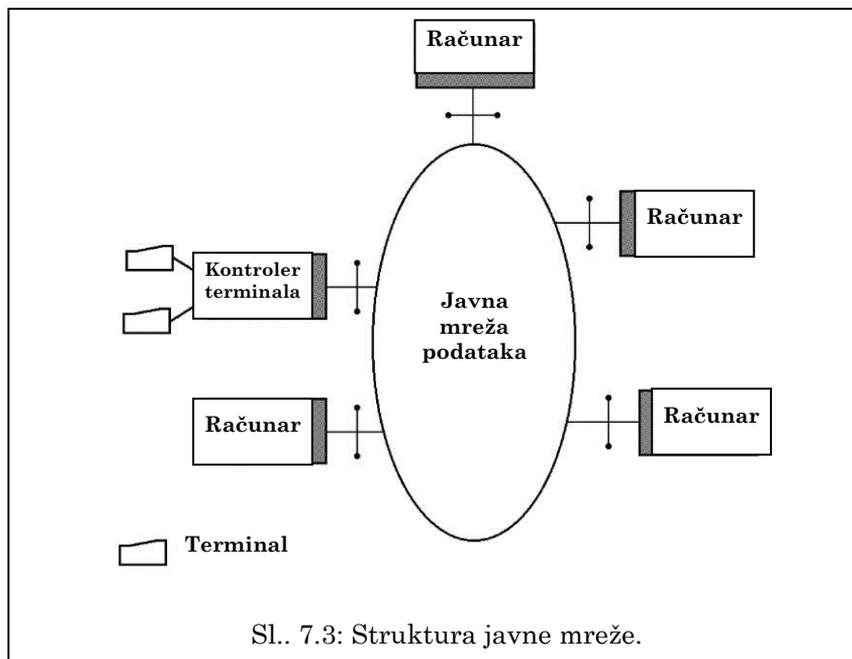
Da bi se centralni procesor rasteretio i proces obrade postao efikasniji, u ovakvim sistemima obično postoji poseban komunikacioni procesor (Čeoni procesor engl. Front-end-processor) koji upravlja samom komunikacijom sa terminalima. Računarski sistemi sa ovakvom organizacijom i danas imaju značajnu primenu,



posebno u velikim sistemima, kakve su banke, u kojima postoji potreba za transakcijama koje se višestruko ponavljaju na zajedničkoj bazi podataka.

Sa razvojem tehnologije mikroprocesora i pojavom PC računara i radnih stanica bitno se menja celokupna koncepcija korišćenja računara i organizacije računarskih sistema. Računari postaju dostupni širokom krugu korisnika, tako da terminale zamenjuju PC računari i radne stanice. Svaki korisnik u sistemu može deo obrade da izvrši na svom lokalnom računaru, ali u velikim sistemima ostaje potreba za njihovom komunikacijom. Ona se ostvaruje razmenom poruka između računara povezanih u mreže računara. Da bi komunikacija između više računara u mreži mogla da bude efikasna razvijaju se sistemi komunikacije u kojima se poruke razmenjuju u paketima. Kako računari u ovakvom sistemu mogu fizički da budu smešteni na geografski udaljenim lokacijama za takve mreže se koristi naziv WAN (Wide Area Computers Network).

U prvim fazama razvoja računarske mreže su bile organizovane unutar radnih organizacija tako što su računari povezivani direktnim linijama ili su za vezu korišćene zakupljene telefonske linije. Vremenom rastu potrebe za komunikacijom između različitih organizacija tako da se za komunikaciju koriste javne telefonske mreže, održavane od strane javnih telekomunikacionih sistema kakva je pošta. Posle niza međunarodnih dogovora razvijeni su međunarodni standardi tzv. protokoli po kojima se ost

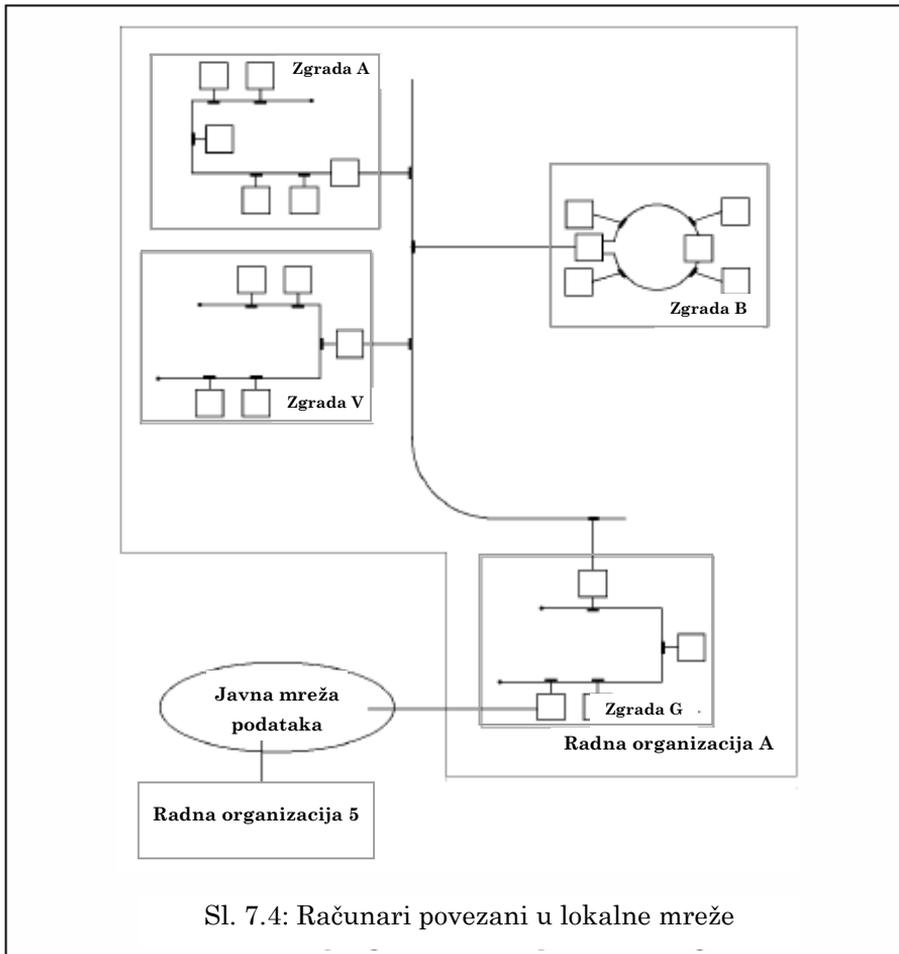


varuje komunikacija u ovakvim sistemima. Na Slici 7.3 šematski je prikazana struktura ovakvih javnih sistema.

Danas se u poslovnim sistemima računari koriste u različite svrhe, tako da skoro na svakom radnom mestu postoji potreba za njihovom primenom. Koriste se za obradu teksta, izradu tehničke dokumentacije, analizu i praćenje proizvodnje, u projektovanju, prezentacijama, za evidenciju svakodnevnog poslovanja i sl. Takav koncept poslovanja i primene računara zahteva neprestanu komunikaciju između korisnika. Zbog toga se danas računari obično povezuju u lokalne računarske mreže u okviru kojih postoji mogućnost brze komunikacije između računara, podržane savremenim operativnim sistemima. Više lokalnih mreža u okviru jedne organizacije se međusobno povezuje u širi sistem koji obično ima izlaz na javnu mrežu, a time i vezu sa drugim sličnim sistemima.

Na Slici 7.4 prikazana je jedna tipična organizacija lokalnih računarskih mreža.

Brzina prenosa u lokalnim računarskim mrežama iznosi do 100 Mb/s. Kao prenosni medijum, u lokalnim mrežama se koriste

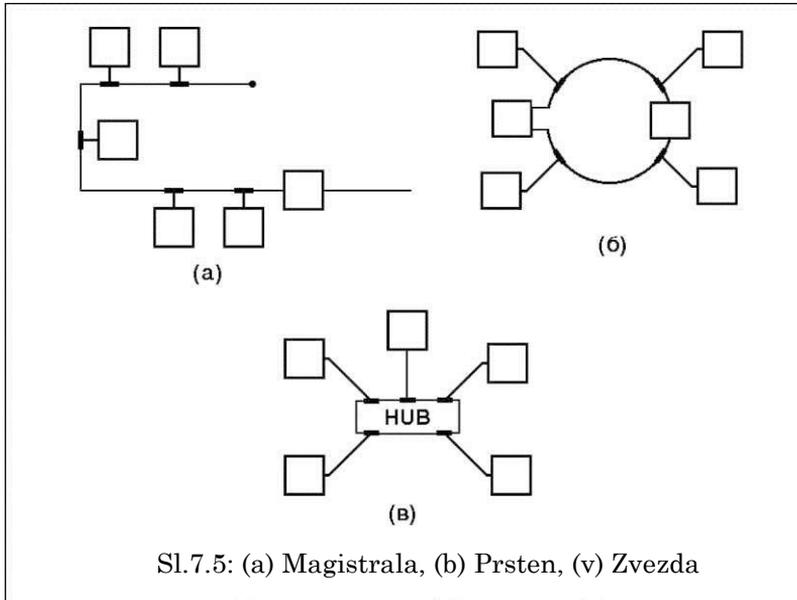


simetrične parice, koaksijalni i optički kablovi. Veze računara u okviru lokalnih računarskih mreža ostvaruju se po različitim topologijama. Karakteristične su topologija zvezde, prstena i magistrale, koje su prikazane na Slici 7.5

Magistrala (BUS ili linijska, topologija) namenjena je korisnicima kod kojih je razdaljina između prvog i poslednjeg računara najviše 180 m. Komunikacija je dvostrana, lako se instalira i malo košta, ali se hardverske greške teško otkrivaju, pa je otežano i održavanje.

Prsten (zatvorena petlja) omogućava veće brzine prenosa, lako se prilagođava na veze sa optičkim kablovima, ali se greške i kvarovi teže otkrivaju, komplikovano je priključenje nove radne stanice i podešavanje parametara konfiguracije.

Zvezda koristi centralni računar-server za povezivanje rad-



nih stanica. Lako se uočavaju i otklanjaju greške na nekoj od radnih stanica, jednostavno se instalira, omogućava povezivanje računara na većim daljinama, ali se kablovi neracionalno raspoređuju i neprikladna je za komunikaciju između krajnjih tačaka na krakovima mreže. Za realizaciju ove topologije potreban je dodatni elemenat, u odnosu na linijsku i prsten topologiju, mrežni koncentrador. Koncentrador obezbeđuje da se podaci, prema odgovarućim protokolima, prenose do drugih računata. Najjednostavniji mrežni koncentratori su tzv. habovi (hubs). Podatke koje računar pošalje stiže do haba, a zatim se ti podaci dalje proslede drugim računarima preko njihovih kablova.

7.1.1 Prenos podataka

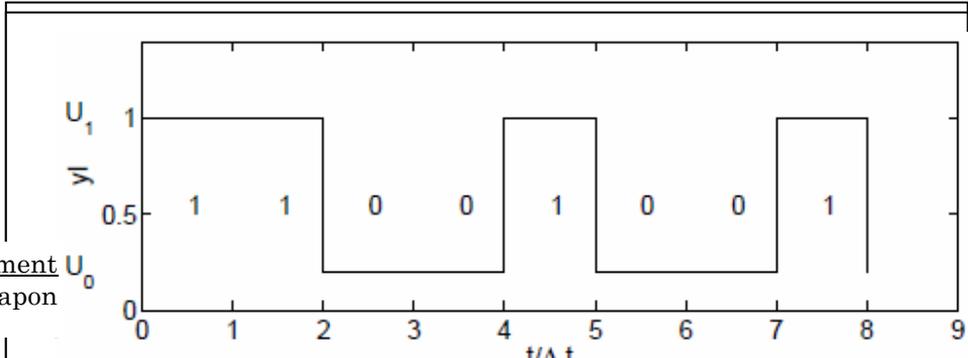
Prenos podataka linijom veze može se obavljati serijski ili paralelno.

Ako se između predajnog i prijemnog mesta prenos podataka obavlja kroz samo jedan kanal¹, onda se elementarni signali-

¹Kanal je deo kapaciteta linije veze kroz koju se ostvaruje veza između predajnika i prijemnika. Dakle, kroz jednu liniju veze može se prenositi više nezavisnih poruka

bitovi (fizički to su signali u obliku impulsa sa dva nivoa) moraju slati sukcesivno, jedan za drugim pa je reč o serijskom prenosu. U tom slučaju se u jednom taktu, tj. jednoj jedinici vremena, može da pošalje samo jedan bit.

Na Slici 7.6 prikazan je vremenski izgled signala u slučaju serijskog prenosa bajta 11001001.



Sl.7.6: Diskretni naponski signal koji predstavlja binarnu reč 11001001. Napon U_1 odgovara binarnoj cifri 1, a napon U_0 binarnoj cifri 0. sa Δt je označen jedan takti interval tj. trajanje jednog takta

U slučaju paralelnog prenosa između predajnog i prijemnog uređaja, tj. u okviru jedne veze, mora postojati n kanala. To omogućava da se istovremeno, dakle u jednom taktu, šalje n bitova.

Unutar računara prenos se, radi postizanja veće brzine, obavlja paralelno. U komunikacijama između računara, prenos se, iz ekonomskih razloga obavlja serijski.

Pošto se unutar računara vrši paralelni prenos podataka, dok se između računara i udaljenih terminala/računara obavlja serijski prenos podataka, to se u izlaznom portu računara mora izvršiti konverzija paralelnog u serijski prenos podataka, a u ulaznom portu računara konverzija serijskog u paralelni prenos.

Brzina kojom računar/predajnik šalje elementarne simbole u prenosni kanal naziva se *protok* ili *brzina prenosa podataka*. Protok se najčešće izražava brojem poslatih bitova u sekundi pa se onda naziva *bitska brzina* i označava sa b/s. Propusna moć, odnosno kapacitet kanala, predstavlja maksimalno mogući protok kroz kanal tj. maksimalni broj elementarnih signala koji mogu da prođu kroz neku tačku u kanalu u jedinici vremena, a da tokom

prenosa ne pretrpe izobličenja koja bi onemogućila da na prijemu budu prepoznati. Kapacitet kanala se, kao i protok, izražava brojem bitova u sekundi.

Jedan računar/predajnik može da počne da šalje podatke tek kada je drugi računar/prijemnik spreman da počne da ih prima. Ako bi se prnosio samo jedan bajt prenos bi se, uz poznavanje brzine prenosa, obavio bez problema. Međutim, u stvarnosti retko se u stvarnosti prenosi samo jedna binarna reč već se sekvencijalno prenosi niz bajtova, odnosno dugački niz bitova. Računar/prijemnik mora da ima informaciju kada se jedan bajt počinje, a kada se završava. U protivnom dolazi do pogrešnog prijema informacija. To znači da računar/prijemnik mora biti u sinhronizaciji sa računarom/predajnikom, jer se samo tako može znati kada se završava jedan i počinje drugi bajt. Na primer, ako treba poslati reč RADIO na izlazu računar/predajnik će generisati sledeći niz binarnih cifara, koji predstavljaju osmorbitni ASCII ekvivalent za svako veliko slovo u reči RADIO:

11010010 01000001 01000100 11001001 11001111

← smer prenosa

Ako bi, na primer, računar/prijemnik započeo prijem samo za jedan bit kasnije, računar/prijemnik bi primio potpuno pogrešnu informaciju.

Da bi se ovaj problem ršio koriste se dva načina prenosa informacija: *asinhroni* i *sinhroni*.

Pri asinhronom prenosu podataka se u grupu binarnih cifara određene dužine, na primer, uz svaki bajt, dodaju dodatni bitovi i to na početak niza jedan bit, tzv. *početni* ili *startni* bit, i na kraj niza jedan ili više bitova, tzv. *krajnji* ili *stop* bit, odnosno, bitovi. Broj stop bitova obično iznosi jedan ili dva. Stop bit ima suprotnu vrednost od start bita. Dakle:

- Početni bit kod asinhronog prenosa daje informaciju računaru/prijemniku da sledi podatak. Obično je početni bit nula.
- Krajnji bit kod asinhronog prenosa daje informaciju računaru/prijemniku da je prenos podatka završen. Obično je krajnji bit jedinica.

Na taj način je lako otkriti početak naredne grupe bitova.

Između prenosa dve uzastopne grupe binarnih cifara obično se pojavljuje vremenski razmak.

Pri asinhronom prenosu raste broj bitova koji treba preneti. Na primer, ako se koristi po jedan start i stop bit po jenom bajtu, sada umesto osam treba preneti deset bitova što znači da se vreme prenosa binarne reči povećalo za 25 %. Zbog toga se asinhroni prenos koristi u slučaju kada se prenos obavlja povremeno, u vremenskim intervalima različitog trajanja, i kada je količina podataka koja se prenosi mala. Tipičan primer je komunikacija koju korisnik uspostavlja sa računarom preko terminala. Poruke koje se prenose linijom, koja povezuje terminal sa računarom, su obično veoma kratke i šalju se u nejednakim vremenskim intervalima. U toku vremena kada nema prenosa, podatak koji odgovara stop bitu biće sve vreme prisutan na ulazu u računar. Sa pojavom start bita (promena napona na ulazu u računar) započinje prenos nove binarne reči. Treba napomenuti da i dok nema prenosa podataka oba uređaja, i terminal i računar, ispituju stanje u kolu da bi mogli da pošalju odnosno da prime novu binarnu reč čim se za to ukaže potreba.

Primer 7.1 *Brzina prenosa kroz neku liniju iznosi 9600 b/s. Ako se radi o asinhronom prenosu, pri čemu se u svaki bajt ubacuje po jedan start i stop bit, za slanje jednog bajta korisnih podataka treba ukupno 10 bitova. Prema tome, od ukupnog broja poslanih bitova 20 % predstavljaju upravljačke bitove, što znači da od 9600 bitova koji se pošalje u jednoj sekundi 80 % predstavlja korisne bitove. Dakle, sa korisnikove tačke gledišta prenos se odvija brzinom od*

$$9600 \cdot 0.8 = 7680 \text{ b/s}$$

Kod sinhronog prenosa bitovi podataka se kontinualno prenose. Brzina prenosa podataka je konstantna, a preko linije veze se obezbeđuje da oba uređaja rade u istom taktu. Na taj način se može identifikovati svaki pojedinačni bit. Da bi moglo da se u grupi primljenih bitova identifikuju, na primer, osmobitne binarne reči, prijemnik se mora podešiti na nizove od osam bitova. Podešavanje se postiže tako što se pri započinjanju prenosa podataka vrši neprekidno prenošenje znaka sa sinhronizaciju. Znak za sinhronizaciju (SYN) u ASCII kodu je 10010110. Predajnik neprekidno šalje ovaj znak. Pošto je prijemnik podešen za prijem

niza 10010110, on može da, na osnovu neprekidno dolazećeg niza sinhronizacionih bitova, podese granicu između dva sinhronizaciona znaka, a time i granicu između pojedinih bajtova. Sada je prijemnik podešen na predajnik i može da prepozna svaki niz od osam bitova kao neku poslatu binarnu reč.

Zbog vremena koje je potrebno da se predajnik i prijemnik dovedu u sinhronizaciju, sinhroni prenos je neefikasan u slučaju kratkih i neredovnih poruka. Međutim, sinhroni prenos ne zahteva unošenje dodatnih bitova u svaku binarnu reč.

Vreme koje je potrebno da neki podatak stigne iz jednog računara u drugi zavisi od

- kapaciteta tj. propusne moći kanala,
- brzine prostiranja signala kroz prenosni medijum i
- dužine linije koja povezuje računare.

Brzina prostiranja signala kroz prenosni medijum zavisi od vrste medijuma. Ako se radi o bežičnom prenosu signala, tj. prenosu elektromagnetnih talasa kroz atmosferu, brzina prostiranja signala je jednaka brzini svetlosti koja iznosi $c = 300\ 000$ km/s. Ako se pak, prenos obavlja kroz fizičke provodnike kao što je koaksijalni kabl ili optički kabl, brzina prostiranja signala je za oko trećinu manja od brzine svetlosti i iznosi oko $v = 200\ 000$ km/s.

Primer 7.2 *Neka je rastojanje između dva računara $L = 2\ 000$ km, a prenos se obavlja brzinom od 1 Mb/s kroz optički kabl. Vreme prostiranja signala između računara t_{prop} , koje se još naziva i vreme kašnjenja, iznosi*

$$t_{prop} = \frac{L}{v} = \frac{2000000}{200000000} = 0.001s$$

Vreme t_{emit} , koje je potrebno da računar pošalje na liniju 1 NJ podataka brzinom od 1 Mb/s iznosi

$$t_{emit} = \frac{1000}{1000000} = 0.001s$$

Ukupno vreme potrebno da se 1 kb podataka prenese između dva računara koja se nalaze na rastojanju 2 000 km iznosi

$$t_{tot} = t_{prop} + t_{emit} = 0.01 + 0.001 = 0.011s$$

7.2 Internet-globalna računarska mreža

Za skup povezanih računarskih mreža koje komuniciraju zajedničkim protokolima koristi se termin "Internet". Najveća računarska mreža na svetu je Internet i sastoji se od mreža koje koriste TCP/IP (Transmission Control Protocol/Internet Protocol) protokole².

U terminologiji koja se koristi na Interneti, za označavanje mrežne stanice (računara ili terminala u mreži) koristi se termin host. Hostovi se povezuju na mrežne čvorove (DCE – Data Communication Equipment) koji čine granice komunikacione mreže. Tipičan primer DCE uređaja je modem.

Internet komunikacioni sistem se sastoji od povezanih paketnih mreža čiji hostovi komuniciraju korišćenjem Internet protokola. Mreže su povezane pomoću računara za prebacivanje paketa koji se, u Internet terminologiji, zovu mrežni prolazi (gateways).

Struktura Interneta se zasniva na sledećim pretpostavkama:

- Internet je mreža sačinjena od mreža (mreža mreža). Polazi se od toga da je svaki host računar povezan u neku mrežu. Dva host računara u istoj mreži komuniciraju međusobno koristeći isti skup protokola koji koriste i za komunikaciju sa hostovima na udaljenim mrežama.
- Sistem mora da toleriše razlike među mrežama. Kada se govori o različitim karakteristikama mreža, tu se pre svega misli na širinu propusnog opsega (bandwidth) i kašnjenje kao i na maksimalnu veličinu paketa.
- Treba postići otpornost na otkaze pojedinih mreža, mrežnih prolaza i host računara. Da bi se poboljšala robusnost komunikacionog sistema, mrežni prolazi prosleđuju svaki datagram nezavisno od ostalih. Na taj način, omogućeno je korišćenje redundantnih puteva, čime se postiže otpornost na otkaze pojedinih mrežnih prolaza i mreža.

Host računar ili jednostavno host, je krajnji potrošač komunikacionih usluga. Host uglavnom izvršava aplikacione pro-

²Protokoli interneta predstavljaju skup pravila koja treba ugraditi u računarsku mrežu i mogu biti sinhroni i asinhroni. Međusobna komunikacija računara u mreži odvija se prema ovim pravilima.

grame u ime jednog ili više korisnika, upotrebljavajući lokalne mrežne i/ili Internet komunikacione servise.

Hostovi se razlikuju po brzini i mogućnostima ali i po funkcijama koje obavljaju. To mogu biti različiti računari, od malih mikroprocesorskih sistema do radnih stanica i super računara. Po funkciji, kreću se od hostova sa jednom namenom do hostova koji podržavaju različite mrežne servise, obično rad na udaljenom računaru, prenos datoteka i elektronsku poštu.

Mrežni prolaz (gateway) je opšti naziv za uređaj koji vrši prevođenje između nekompatibilnih protokola, bez obzira na kom se nivou ti protokoli nalaze. Na taj način, mrežni prolazi obezbeđuju komunikacioni put između različitih mreža.

Mostovi (bridges) su mrežni prolazi koji rade na drugom nivou (nivo veze) OSI modela. Mogu se koristiti, na primer, za povezivanje Ethernet i Token Ring mreža. Efikasni su samo za povezivanje manjeg broja mreža kada je broj mogućih puteva između pošiljaoca i primaoca relativno mali.

Kada se povezuje više mreža, treba odrediti optimalni put za svaki paket u cilju smanjenja nepotrebnog saobraćaja kroz mrežu. U tom slučaju, za povezivanje mreža koriste se ruteri (routers). To su složeniji i skuplji uređaji od mostova i koriste se za povezivanje velikih lokalnih mreža koje mogu biti veoma udaljene međusobno. I mostovi i ruteri pretpostavljaju da se u višim nivoima koriste isti protokoli.

Mrežni prolazi za elektronsku poštu (mail gateway) omogućuju razmenu elektronske pošte između Interneta i drugih značajnih sveskih mreža kao što je Bitnet.

Mreže koje sačinjavaju Internet mogu se svrstati u tri hijerarhijska nivoa: kičmene mreže (backbone networks), mreže srednjeg nivoa (mid-level networks) i mreže najnižeg nivoa (stub networks).

Kičmene mreže su najvišeg hijerarhijskog nivoa. Sve mreže srednjeg i najnižeg nivoa zakačene na istu kičmenu mrežu međusobno su povezane i mogu da razmenjuju podatke.

Mreže srednjeg nivoa su tranzitne mreže (transit networks) koje povezuju mreže najnižeg nivoa na kičmenu mrežu. Tranzitna mreža prenosi saobraćaj između drugih mreža. Tranzitne mreže prenose i saobraćaj između lokalnih hostova.

Mreže najnižeg nivoa prenose pakete samo između lokalnih hostova, tj. ne prenose saobraćaj drugih mreža.

7.2.1 Protokoli Interneta

Komunikacija računara u mreži ostvaruje se po unapred definisanim dogovorima - protokolima. Grupa protokola koja se koristi na Internetu poznata je pod nazivom TCP/IP. Obuhvata preko sto različitih protokola, a njihov broj stalno raste. Protokoli TCP/IP se baziraju na pristupu komunikacijama koji uključuje tri tipa entiteta: procese, računare i mreže. Procesu su fundamentalni entiteti koji komuniciraju. Oni se izvršavaju na računarima koji obično podržavaju veći broj procesa istovremeno. Komunikacija među procesima se odvija preko mreže na koju su računari povezani.

Danas je na Internetu u upotrebi IPv4 protokol koji je nastao pre oko tri decenije. U međuvremenu se mnogo što sta promenilo. Broj mreža je porastao iznad svih predviđanja tako da će vrlo brzo adresni prostor koji je omogućava protokol Ipv4 postati nedovoljan. Krug i profil korisnika se značajno proširio, pa su se shodno tome promenili i zahtevi. Zato je razvijena nova verzija IP protokola, IPV6, koja se mnogo razlikuje od protokola IPv4.

7.2.2 Adresiranje na Internetu

Svaki mrežni priključak (host ili mrežni prolaz) na Internetu jednoznačno je definisan svojom adresom. Koriste se dva načina za predstavljanje adresa numerički i simbolički.

Numerička ili IP adresa je binarni broj dužine četiri bajta (32 bita). Adresa počinje mrežnim brojem (network number) a ostatak se naziva "lokalna adresa".

Uobičajeni zapis IP adrese je u obliku niza od četiri decimalna broja razdvojena tačkama. Svaki od njih predstavlja adekvatni ekvivalent jednog bajta 32-bitne adrese. Na primer, jedna IP adresa glasi 160.99.11.2.

U praksi se mnogo češće koriste simboličke adrese. Zasnivaju se na "sistemu domena" Domain Name System DNS, To je hijerarhijski, distribuirani metod organizacije imena na Inter-

netu. DNS grupiše hostove hijerahijski, po "domenima". U simboličkoj adresi hosta navodi se naziv samog hosta, kao i nazivi (mnemoničke oznake) svih domena višeg nivoa kojima on pripada.

FQDN (Fully Qualified) je sekvenca oznaka domena razdvojenih tačkama gde svaka oznaka počinje i završava alfanumeričkim znakom i može da sadrži znake "-". Na primer, FQDN za host 160.99.11.2 je **orion.eknfak.ni.ac.yu** ili 160.99.71.12 je **efak.ni.ac.yu**. Prva komponenta (europa) je naziv hosta, a ostale komponente se odnose na domen. **efact.ni.ac.yu** je FQDN za domen Elektronskog fakulteta u Nišu u okviru **ni.ac.yu** domena ili **eknfak.ni.ac.yu** je FQDN za domen Ekonomskog fakulteta u Nišu u okviru istog domena **ni.ac.yu**.

7.2.3 Pristup Internetu

Za priključenje na Internet potrebna je određena hardverska i softverska oprema. Postoje tri glavne varijante:

- Pristup nekom Internet hostu preko telefonske veze. Potreban je računar sa modemom, komunikacioni softver i "račun" (korisničko ime i lozinka) na tom hostu.
- Korišćenje telefonske veze uz SLIP (Serial Line Internet Protocol) ili PPP (Point to Point Protocol) protokol. SLIP (PPP) omogućuje slanje IP paketa preko asinhronne serijske linije. Potrebni su: računar sa SLIP ili PPP softverom, aplikacioni softver za telnet, FTP i druge servise, modem i račun na hostu koji razume SLIP. Na taj način računar priključen preko komutirane telefonske mreže postaje host na Internetu kao da je direktno priključen iznajmljenim linijama.
- Direktna priključak (full Internet connection). Omogućuje da svi računari lokalne mreže pristupaju Internetu. Potreban je ruter i veza do rutera neke druge mreže Interneta (recimo rutera firme koja obezbeđuje priključak). Tipično, veza je iznajmljena telefonska linija. Lokalna mreža može biti sastavljena samo od rutera i PC-a. Na računarima je potreban odgovarajući softver: TCP/IP grupa protokola zajedno sa programima za telnet, FTP i ostale servise.

Telefonska mreža je predviđena za prenos govornog, dakle analognog signala. Zato se na izlazu iz računara, a pre ulaska

na telefonsku liniju, moraju digitalni signali da konvertuju u analogne signale. Na ulazu u računar se mora obaviti inverzni postupak: analogni signali koji stižu sa telefonske linije moraju se predhodno transformisati u digitalne signale. Ove konverzije obavlja uređaj koji se naziva *modem* modulator/demodulator).

Uređaji koji generišu ili koriste digitalne signale nazivaju se opštim imenom *DTE uređaji* (Data Terminal Equipment). Na primer, DTE uređaji su računari, štampači, i sl. Dakle, svaki uređaj koji predstavlja izvor ili krajnje odredište digitalnog signala predstavlja DTE uređaj.

U slučaju modemske veze komunikacija između dva računara se odvija u četiri faze. Neka komunikaciju započine računar A. U tom slučaju se kaže da je računar A inicijator veze i da računar B odgovara.

1. Uspostavlja se veza između računara i modema. Računar i modem izmenjuju poruke da li su spremni za rad.
2. Uspostavlja se fizička veza između modema A i modema B. To se ostvaruje tako što računar A šalje modemu A poruku: zahtev za slanje. Nakon toga modem A šalje signal nosioca modemu B. Kada modem B detektuje signal nosioca, on obaveštava računar B da će prenos početi. Pošto je poslao signal nosioca, modem A obaveštava "svoj" računar da može da pošalje poruku.
3. Računar A šalje niz bitova u modem A kao i sinhronizacioni impuls. Modem A konvertuje digitalne podatke u analogni signal i šalje ga po telefonskoj liniji. Prijemni modem, modem B, prima analogni signal, konvertuje ga u digitalni signal i prosleđuje ga, zajedno sa sinhronizacionim imulsom, računaru B.
4. Kada se prenos obavio, oba računara deaktiviraju svoja koja šalju zahtev za slanje, modemi isključuje nosioce signala, detektore signala koji dolaze sa linija.

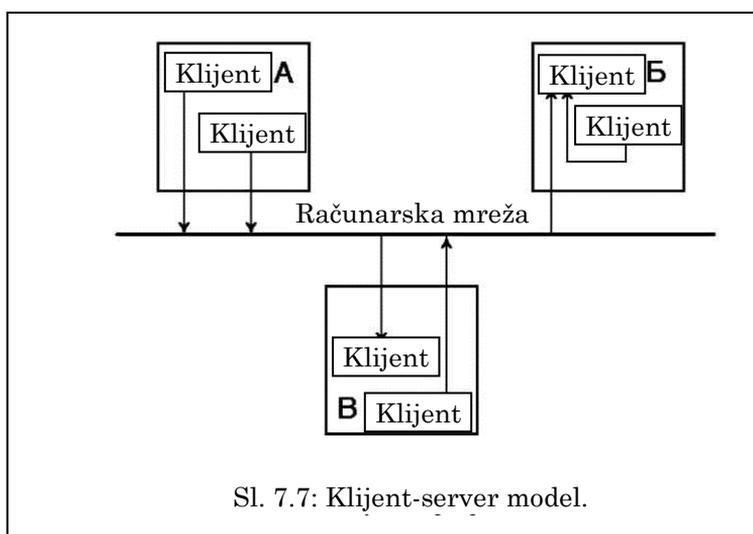
7.2.4 Klijent-server model

Korisnik vidi Internet kao skup mrežnih usluga i servisa. Usluge i servisi Interneta su uvek organizovani po klijent-server

principu. Server je specijalni softver na nekom mrežnom računaru koji opslužuje zahteve korisnika. Onaj koji upućuje zahtev mora da startuje odgovarajući program, zvani kljent, na svom računaru.

Klijent aplikacija koja želi neku uslugu, poručuje to serveru slanjem poruke sa opisom zahtevanog zadatka. Server opslužuje zahtev i odgovara klijentu svojim porukama. Skup važećih poruka definisan je protokolom. Naravno, server može usluživati više klijenata odjednom.

Na Slici 7.7 sa A, B i V su označeni hostovi na kojima se može izvršavati više klijent/server aplikacija.



7.3 Usluge i servisi Interneta

Osnovne usluge u okviru Internet-a su: elektronska pošta, prenos datoteka, interaktivni rad na udaljenom računaru, interaktivne poruke i informacije o korisnicima na udaljenom računaru.

Usluge višeg nivoa nazivaju se mrežni servisi. Obično nastaju tako što se osnovne mrežne usluge obogaćuju dodatnim sadržajima. Na primer, elektronske konferencije se mogu posmatrati kao nadgradnja elektronske pošte. Postoji mnogo servisa i njihov broj stalno raste, ali neki od najvažnijih su elek-

tronske konferencije, interaktivni razgovori više korisnika, gopher, WAIS i WWW.

Elektronska pošta (electronic mail, e-mail) omogućava razmenu tekstualnih poruka (pošte) između korisnika. Svaki korisnik ima svoje "poštansko sanduče" (to je zapravo datoteka) na nekom hostu, a poštansko sanduče ima pridruženu elektronsku adresu (to je tzv. e-mail adresa korisnika). Da bi se nekom poslalo pismo ili pročitala pristigla pošta, koristi se odgovarajući program (mail, elm, pine ili neki drugi). Pošiljalac predaje pismo ili tako što ga direktno otkuca u nekom od pomenutih programa ili tako što navede ime datoteke u kojoj je prethodno pripremljen tekst poruke. Za prenos pisma do poštanskog sandučeta primaoca na udaljenom računaru koristi se SMTP (Simple Mail Transfer Protocol). Primaalac ima mogućnost da pročita sve poruke koje je dobio, da ih briše, štampa, arhivira i da odgovori na one na koje želi.

Zahvaljujući mnoštvu mrežnih prolaza, koji spajaju Internet sa drugim mrežama, elektronska pošta pokriva praktično ceo svet. Glavna prednost u odnosu na običnu poštu je velika brzina slanja i primanja poruka.

Prenos datoteka (File transfer) predstavlja niz pravila i procedura pomoću kojih se ostvaruje prenos datoteka sa jednog računara na drugi. Odgovarajući protokol zove se FTP (File Transfer Protocol). Korišćenjem posebnog programa (najčešće pod nazivom ftp) ostvaruje se veza sa ftp serverom na udaljenom računaru. Ukoliko se ne traži unošenje lozinke radi se o takozvanom anonimnom ftp serveru. Najznačajnije mogućnosti su pretraživanje direktorijuma da bi se pronašla željena datoteka, preuzimanje datoteke sa sistema (download) i slanje datoteke sistemu (upload).

Interaktivni rad na udaljenom računaru (telnet) korisniku daje privid da je fizički vezan na udaljenu mašinu kao terminalski korisnik. Vazira se na TELNET protokolu. Omogućuje zadavanje komandi i izvršavanje programa koji se nalaze na udaljenom računaru.

Interaktivne poruke omogućavaju da dva udaljena korisnika uspostave razgovor (talk). To su kratke poruke, obično dužine jednog reda teksta, koje se, u vrlo kratkom roku pošto su otkucane, prikazuju na ekranu primaoca.

Informacije o korisnicima na udaljenom računaru dobijaju

se pomoću komande **finger**. Mogu se dobiti podaci o konkretnom korisniku ili o svim trenutno prijavljenim korisnicima na tom udaljenom računaru. Među podacima se, pored ostalog, nalaze ime korisnika, datum poslednjeg javljanja, kao i "plan" - neka vrsta rezimea koji je taj korisnik definisao.

Elektronske konferencije (Usnet, Netnews) su sačinjene od poruka (članaka) nastalih razmenom mišljenja korisnika o najrazličitijim temama. Protokoli i softver za distribuciju **Usenet** konferencija su u upotrebi u vie mreža, među kojima su Internet, UUCP, EARN/Bitnet i Fidonet. Usenet je, zapravo, ogromna grupa računara koji razmenjuju članke. Članci su, po temama, uređeni u grupe (newsgroups) tj. konferencije. Postoje hiljade konferencija, od kojih se veliki broj distribuira po celom svetu. Ukoliko korisnik ima pristup nekom od ovih računara, on može da prati konferencije koje su tamo dostupne i da šalje odgovore porukama koje su drugi poslali. U tu svrhu, koristi se program za čitanje konferencija (newsreader), kao što je rn (read news) ili nn (no news) ili tin (threaded netnews reader). Program pristupa lokalnim konferencijama ili koristi NNTP (Network News Transfer Protocol) da bi pristupio konferencijama na nekom drugom računaru u mreži.

Interaktivni razgovori više korisnika poznati su pod nazivom IRC (Internet Relay Chat). Najpre se pristupi IRC serveru, pogleda se ko sve trenutno komunicira i izabere grupa ljudi u čiji se razgovor uključuje. Svakoj diskusiji pridružen je po jedan "kanal". Na različitim kanalima odvijaju se potpuno neza-visne diskusije. Pošto je korisnik odabrao kanal, njegove poruke upućuju se svim ljudima u svetu koji trenutno koriste taj kanal. Mehanizam slanja i prijema poruka baziran je na slanju interaktivnih poruka, što znači da se vreme isporuke meri sekundama.

Gopher je distribuirani sistem za isporuku dokumenata. Omogućava pristup najrazličitijim tipovima informacija kroz jednostavan interfejs na bazi menija. Struktura menija lici na organizaciju direktorijuma sa mnogo poddirektorijuma i fajlova. Stavke menija mogu biti datoteke, podmeniji i veze (linkovi) na druge **gopher** i ne-gopher servere. Iz gophera se može pristupiti različitim mrežnim uslugama i servisima kao što su FTP, telnet i WAIS.

WAIS Wide Area Information Server) omogućuje pretraživanje baza

podataka kroz mrežu pomoću jednostavnog (easy-to-use) korisničkog interfejsa. Vaze podataka su uglavnom kolekcije tekstualnih dokumenata o različitim temama: od poljoprivrede do društvenih nauka, ali mogu sadržati i zvuk, slike i video zapis. Sistemi baza podataka su različiti, ali korisnik ne mora da nauči sve njihove upitne jezike. On formuliše upite na prirodnom jeziku koji sadrže ključne reči po kojima se vrši pretraživanje. WAIS bazama se može pristupiti pomoću WAIS, gopher i WWW klijenata.

World Wide Web (WWW, W3, triple, W, ili samo Veb) je jedan od najnovijih i najpopularnijih servisa. To je distribuirani hiper-medijalni sistem koji omogućava uniformni pristup širokom spektru dokumenata (ASCII, GIF, PostScript...), protokola (FTP, TELNET, NNTP,...) i servisa (gopher, WAIS). Veb servis se zasniva na protokolu nazvanom HyperText Tranfer Protocol (HTTP), dok je os-novni programski jezik za opis WWW dokumenata HTML (Hyper Tekst Markup Language).

7.4 Veb-multimedijalni servis interneta

Veb je razvijen u Evropskom centru za nuklearna istraživanja (CERN) od strane Veb tima na celu sa Tim Berners-Leeom. Nastao je kao odgovor na potrebu za lakšom međusobnom komunikacijom naučnika CERN-a. Članovi ove organizacije raspoređeni su na širokom geografsko području, u više zemalja.

Sistem je osnovan sa ciljem da se na lak način prezentiraju mnogobrojne informacije dostupne na Internetu. Prve verzije su napravljene novembra 1990. na NeXT računarima. Značajan doprinos razvoju i prihvatanju Veba dao je grafički Veb čitač Mosaic za X i MS Windows, razvijen januara 1993. u NCSA instituciji (National Center for Supercomputing Applications). Jedan od programera, članova prvobitnog tima autora Mosaica, Marc Andersen, napustio je NCSA da bi osnovao svoju sopstvenu kompaniju za pružanje Internet usluga - Netscape. Mosaic je bio prvi grafički čitač koji je obezbeđivao pristup Vebu. Prenet je na mnoge druge računarske platforme (Sun, Silicon Graphich, Macintosh, VAX, računare sa raznim verzijama UNIX operativnog sistema), te je stekao veliku popularanost, kao i sam Veb. Naime, Veb je postao javno dostupan korisnicima Interneta januara 1994. godine i do

danas je postao najbrže rastuća usluga u istoriji (količina informacija koje se prenose korišćenjem Veba povećava svake godine više stotina puta).

Veb ima elemente multimedije, jer omogućava prikazivanje tekstova, slika, zvučnih i video zapisa i bilo kog drugog dokumenta za koji postoji program za prikazivanje na korisnikovom računaru. Svi ti podaci se nalaze na mrežnim serverima, a Veb je zadužen za prenos datoteka kroz mrežu. Kada se pristupi nekom Veb serveru prikazuje se njegova naslovna strana (home page). Primer jedne naslovne strane dat je na Slici 7.8



Sl. 7.8: Primer jedne naslovne strane

Sa slike se vidi da su neke reči podvučene, što nam ukazuje da se radi o hipertekst sistemu. Hipertekst (hypertext) je tekst koji sadrži hiperlinkove (hyperlink). To su obično reči ili grupe reči kojima se referenciraju drugi dokumenti. To znači da je dovoljno aktivirati mišem podvučenu reč da bi se automatski otvorio novi dokument. Na primer, ako se uvodi novi pojam u tekst, hiperlink može ukazivati na drugi dokument koji taj pojam detaljnije objašnjava. Naravno, moguć je i skok na drugu poziciju unutar istog dokumenta. Veb nije samo hipertekst već i hipermedijalni sistem. Osim teksta, dokumenti mogu da sadrže slike i da budu povezane hiperlinkovima sa datotekama koje sadrže zvučne ili video zapise. Slike (ili delovi slike) takođe mogu da se koriste kao hiperlinkovi.

Veb je distribuirani hipermedijalni sistem. Distribuiran je zato što se hipermedijski dokumenti mogu nalaziti na bilo kom Veb serveru u mreži. Prateći hiperlinkove možemo se kretati od računara do računara, kroz pravu globalnu hipermedijsku mrežu.

Kako se referencirani dokumenti mogu nalaziti na velikom broju računara u mreži, radi se o pravom moru informacija. Programi koji nam olakšavaju "navigaciju" zovu se Veb čitači (browsers). Jedan od najpoznatijih je Netscape. Za pristup određenim dokumentima sve što korisnik treba da zna je adresa polaznog dokumenta (obično home page nekog Veb servera). Na primer, da bi dobio naslovnu stranu sa prethodne Slike 7.8 korisnik treba da unese adresu: **<http://factae.elfak.ni.ac.yu>**

Nakon što je otvoren polazni dokument, potrebno je samo pratiti hiperlinkove da bi se pristupilo ostalim dokumentima koji mogu biti na bilo kom računaru u mreži.

7.4.1 Pristup Vebu

Kao i svi drugi Internet servisi, Veb koristi klijent-server model da bi obezbedio pristup informacijama. Server je program koji se stalno izvršava (tzv. daemon) na nekom hostu Interneta. Veb klijent je program kojim korisnik "prelistava" Veb dokumente smeštene na tim hostovima. Takvi programi su Veb čitači. Pored Veb čitača potrebno je postojanje SLIP, PPP ili nekog drugog TCP/IP softvera na računaru.

Postoje tri osnovna načina za pristup Vebu.

- Pomoću Veb klijenta (obično grafičkog) na lokalnom računaru. Potrebno je da računar bude direktno priključen na TCP/IP mrežu ili da koristi SLIP/PPP preko telefonske veze.
- Pomoću (tekstualnog) Veb klijenta na udaljenom računaru. Koristi se ako računar nije direktno priključen na mrežu (pristup preko telefonske veze i modema) ili nema instaliran Veb klijent.
- Pristup preko elektronske pošte. Ako korisnik ima pristup samo elektronskoj pošti, može da šalje zahteve u vidu e-mail poruka odgovarajućim serverima, koji mu isporučuju Veb dokumente u elektronsko poštansko sanduče. Zahtevi sadrže komande koje su ekvivalent interaktivnih komandi nekog Veb čitača.

Danas je veoma aktuelan koncept integrisanja Veba sa drugim servisima tako da se iz Veb čitača može pristupiti i drugim objektima Interneta. Veb čitač se npr. može koristiti za slanje elektronske pošte. Elektronska pošta je danas jedan od najpopularnijih Veb servisa. Kada korisnik izabere svog Internet Service Provider-a, potpuše sa nime ugovor o korišćenju Interneta, odmah se definiše Internet adresa. Ova adresa sastoji iz dva dela: adresa korisnika i adresa Internet Service Provider-a.

Prvi deo kreira korisnik i on se najčešće odnosi na ime korisnika, mada može da bude i drugo ime. Drugi deo Internet adrese korisnika određen je elektronskim imenom Internet Service Provider-a. Tako, na primer, sinisaminic predstavlja prvi deo adrese, a **yahoo.com** drugi deo elektronske adres, dok se znak @ koristi za razdvajanje ta dva pojma. Kompletna elektronska adresa autora ovog udžbenika je sinisaminic@yahoo.com. Preko elektronske pošte se može komunicirati sa korisnicima Interneta iz čitavog sveta, i ne postoji mogućnost dupliranja bilo koje elektronske adrese.

Objekat (resurs) dostupan preko Veba može da bude fajl (npr. HTML dokument) neka usluga (ftp, telnet i dr.) ili program (CGI server programi). Ovim objektima se pristupa korišćenjem različitih protokola, bilo da su oni ranije definisani (FTP, TELNET) ili projektovani za potrebe Veba (HTTP).

Jedinstveni metod adresiranja različitih objekata na Internetu poznat je pod nazivom URL (Uniform Resource Locator).

URL je formalizovana informacija, adresa koja služi za lociranje i pristup nekom resursu na Internetu. Predstavlja se rečima određenog formata pri čemu se za opis resursa u okviru različitih servisa koriste različiti formati (šeme).

7.4.2 Veb čitači

Najpoznatiji grafički čitači je svakako Netscape. Može se naći u verzijama za MS Windows, Macintosh, X Windows i druge operativne sisteme. Nešto kasnije pojavljuje se i Microsoftov proizvod Internet Explorer koji je danas standardni čitač Windows okruženja i najveći konkurent Netscape-u.

Netscape prikazuje dokumente inkrementalno, tj. delovi dokumenta se pojavljuju na ekranu onako kako se prenose (ne čeka završetak prenosa celog dokumenta da bi ga prikazao). Podržava mnoga proširenja HTMLa od kojih neka nisu deo predloženog HTML 3.0 standarda. Netscape je komercijalni proizvod ali je dostupna i besplatna verzija koja se može naći na anonimnim Ttr serverima.

TkWWW je čitač za X Windows sisteme koji u sebi sadrži i editor za pisanje HTML dokumenata.

Tekstualni Veb čitači se koriste uglavnom na UNIX i VMS sistemima.

7.5 Kreiraše HTML dokumenta

Dokumenti koje koristi Veb su tekstualne datoteke u kojima se posebnim sekvencama za formatiranje opisuje slog strane (definiše se šta je naslov, šta je zaglavlje, šta su hiperlinkovi i slično).

Osnovni jezik kojim se opisuju Veb dokumenti je HTML (Hypertext Markup Language). Zasniva se na internacionalnom standardu ISO 8879 – Standard Generalized Markup Language (SGML). Svi Veb čitači razumeju taj jezik. Postoje tri načina za kreiranje HTML dokumenata:

- upotrebom bilo kog tekst editora, (zato što su to **ASCII** tekstovi);
- korišćenjem posebnih HTML editora, od kojih su neki WYSIWYG (What You See Is What You Get) ili blizu toga, a drugi jednostavno asistiraju pri pisanju HTML dokumenata umećući tagove koje biramo iz menija;
- konvertovanjem drugih formata u HTML i korišćenje raznih "filtara" koji omogućavaju WYSIWYG editovanje u raznim programima za obradu teksta (wordprocessors).

Na primer, za Microsoft Windows okruženje postoji nekoliko alata koji mogu bit veoma korisni prilikom kreiranja i objavljivanja Veb stranica, pored ostalih to su FrontPage i Flash MX. Dos-tupan je i predložak (engl. template) za konverziju Word dokumenata u HTML

Pri kreiranju HTML dokumenata poželjno je pridržavati se sledećih pravila:

- korisnici ne treba da čekaju mnogo na prenos dokumenta (preporučuje se da to bude do 5 sekundi), što se pre svega odnosi na naslovnu stranu prezentacije (home page); slike na naslovnoj strani treba da su male, po mogućnosti da to budu ikone koje ukazuju na sadržaj narednih dokumenata, a tek kada korisnik izabere ikonu uključuju se velike slike (za eksterne datoteke sa slikama, zvukom i sl. preporučuje se da vreme prenosa ne bude veše od 30 sekundi);
- dokumenti kojima se pristupa sa udaljenih servera nisu brzi kao "lokalni", pa treba voditi računa da vreme čekanja bude u razumnim granicama;
- uz hiperlink se može navesti veličina (broj bajtova) dokumenta na koji on ukazuje (naročito za velike slike, audio i video); tako se korisniku olakšava da odluči da li da izabere dati hiperlink;
- ne treba mešati veliki broj različitih fontova, stilova, boja i sl. na jednom ekranu;
- ne pisati poruke tipa "klikni ovde" jer je to suvišno;
- na dno svake stranice treba staviti neku referencu, na primer elektronsku adresu autora, da se zna kome se treba

obratiti u vezi dokumenta; ovo se može postići ADDRESS tagom;

- stranice ne treba da su predugačke; treba kreirati hijerarhijski skup dokumenata.

7.5.1 HTML - jezik Veb dokumenata

Hyper Text Markup Language (HTML) je u upotrebi od 1990. godine. Kreirao ga je CERN-ov stručnjak Tim Berners-Lee u okviru World Wide Web projekta.

Od 1993. godine mnogi članovi Interneta su doprineli razvoju HTML-a. Veb čitač NCSA Mosaic je odigrao značajnu ulogu u utemeljivanju HTML-a. Mosaic je prokrčio put in-line slikama, slikama osjetljivim na dodir, ugnježenim listama i formularima. Postojanje razlika u podršci ovih proširenja od strane različitih Veb čitača dovelo je do formiranja HTML radne grupe (formiranje radnih grupa za pojedina pitanja je uobičajena praksa na Internetu). Specifikacija HTML 2.0 (HTML Level 2) uspostavlja standard formalizujući defacto situaciju 1994. godine.

HTML 3.0 (Level 3), poznat i kao HTML+, proširuje mogućnosti jezika da bi podržao formulare, tabele i matematičke formule. Pri tom je zadržana jednostavnost i kompatibilnost sa postojećim dokumentima.

HTML je još uvek jezik u razvoju. Naime, standardi još uvek nisu definisani pa trenutno veliki broj Veb čitača ne podržava sve HTML 3.0 tagove. S druge strane, pojedini Veb čitači podržavaju neke tagove koji nisu deo HTML 3.0 specifikacije (npr. FONT i BLINK i Netscape-i).

7.5.2 Struktura HTML/ dokumenta

Proces pisanja HTML datoteke obuhvata unošenje sa tastature HTML oznaka i običog teksta u tekst editor (npr. Notepad), snimanje datoteke na disk pod nazivom ime.htm, i potom učitavanja iste u neki Veb čitač (npr. Internet Explorer) da bi se videlo šta je od posla ispalo. Ukoliko se uoči neka greška, postupak se ponavlja ispočetka.

Za svaki formalni jezik koji računari treba da razumeju

definiše se skup pravila kojima se upravlja njegovim elementima i njihovim rasporedom. Taj skup pravila nazivamo sintaksom jezika. Sintaksa jezika HTML opisuje način na koji Veb čitač prepoznaje i interpretira instrukcije sadržane u njegovim oznakama.

Specijalni znaci kojima se HTML oznake razdvajaju od običnog teksta jesu uglaste zagrade:< (leva zarada) i > (desna zagrada). Ti znaci govore Veb čitaču da obrati posebni pažnju na ono što se nalazi unutar njih. U tekstu oznaka se ne pravi razlika između malih i velikih slova, mada se radi bolje čitljivosti češće koriste velika slova, čime se HTML oznake izdvajaju od ostalog teksta.

Ceo dokument se može posmatrati kao jedan HTML element koji se sastoji od zaglavlja dokumenta (NEAB element) i tela (VOSU element):

```
< HTML >
  < HEAD > elementi zaglavlja dokumenta < /HEAD >
  < BODY > elementi tela < /BODY >
< /HTML >
```

HTMLJ oznake su oivičene uglastim zagradaama i običo se javljaju u parovima tako da < *HEAD* > označava početak a < /*HEAD* > kraj zaglavlja dokumanta. Ampersand (&) označava početak, a simbol tačka-zapeta kraj znakovne celine (α). Numerička celina počinje znakovima &#, a završava se tačkom i zapetom (α -HTML kod za malo slovo alfa). Oznake mogu imati jedan ili više atributa radi specificiranja *URL* adresa ili dodatnog teksta na koji se oznaka odnosi. Na primer, oznaka < *IMG*>, koja se koristi za umetanje grafike, može koristiti sledeće attribute koji pomažu u specifikaciji izvora i načina smeštanja slike na Veb stranu:

- < *5KS* > je izvor slike, odnosno njena *URL* adresa
- < *ALT* > je alternativni tekst koji će biti prikazan ako čitač nema grafičke mogućnosti, ili je prikazivanje grafike isključeno
- < *ALIGN* >= (*TOP* | *MIDDLE* | *BOTTOM*) upravlja smeštajem grafike.
- < *ISMAP* > označava da grafika predstavlja mapu osetljivu na dodir, koja sadrži jednu vezu do drugih lokacija ili više njih, naznačenih sadržinom slike.

U ovako definisan dokument pored slika ubacuju se i drugi elementi kao što su tabele, animacije, formulari, linkovi ili delovi aplikacija napisani u drugim programskim jezicima.

Primer 7.3 Jvdnostavan primer Veb dokumenta.

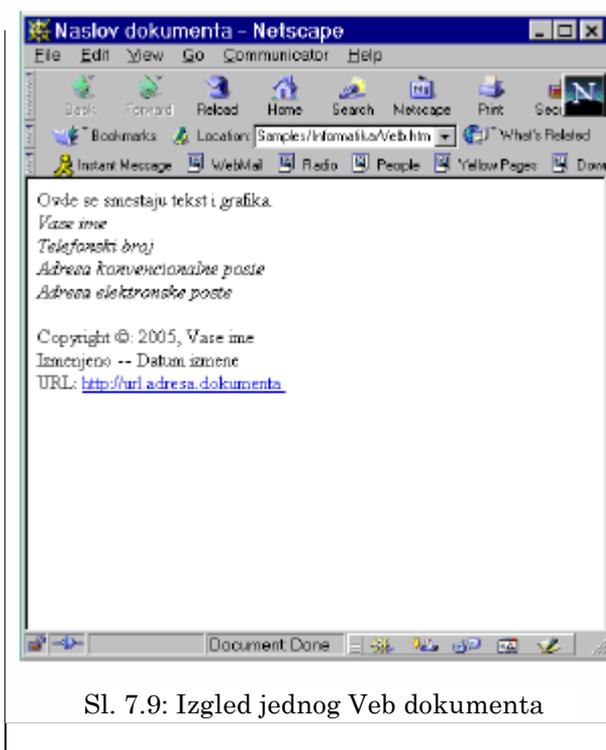
```
< HTML >
< HEAD >< TITLE >Naslov dokumenta</ HEAD >
< BODY >
< BODY BGCOLOR = "# ffffff" >
< p > Ovde se smeštaju tekst i grafika.
< ADDRESS >
< Vaše ime < BR >
< Telefonski broj < BR >
Adresa konvencionalne pošte < BR >
Adresa elektronske pošte < BR >
</ ADDRESS >
< P > Copyright &copy: 2005, Vaše ime < BR >
Izmenjeno – Datum izmene < BR >
URL :< A HREF = http://url.adresa.dokumenta >
< BODY >
</ HTML >
```

Oznaka `<BODYBGCOLOR = "rrggbb">` označava boju pozadine koja se dobija aditivnim mešanjem primarnih boja crvene, zelene i plave. Za svaku boju je rezervisano dva bajta, što za naš primer znači da su količine primarnih boja iste: $R = G = B = ff$. Kao rezultat se dobija bela boja pozadine. Oznaka `< P >` odnosi se na "pasus", deli tekst u razdvojene oblasti. Oznaka `< ADDRESS >` odnosi se na adresu, informacija za obraćanje autoru. Oznaka `< BR >` odnosi se na novi red, prelazak na novi red prilikom ispisivanja teksta na ekranu. Oznaka `< A >` odnosi se na sidro, vidljivi upravljači element koji povezuje jedan resurs Veba sa drugim.

Datoteka se snimi na disk pod nazivom **Primer.htm** u odabrani direktorijum.

Odgovarajući HTML dukument prikazan je na Slici 7.9, a dobijen je učitavanjem datoteke **Primer.htm** u **Internet Explorer** Veb čitač. Treba najpre otvoriti prozor **Internet Explorera** tako što se dva puta klikne levim tasterom miša na ikonu **Internet Explorer-a** koja se nalazi na radnoj površini operativnog sistema Windows,

a zatim u "meni liniji" otvorenog prozora otvori padajući meni **File** u kome se bira opcija **Open** da bi se pronašao direktorijum u kome se nalazi datoteka **Primer.htm**. Desnim klikom miša na opciju **browse...** otvara se novi prozor **Microsoft Internet Explorer** koji omogućava pronalaženje snimljenog fajla **Primer.htm**. Nakon pronalaženja fajla dokument se otvara tako što se klikne na njega i otvori prozor u kome se bira taster **OK**.



Sl. 7.9: Izgled jednog Veb dokumenta

Na raspolaganju stoje naslovi različitih veličina, kojima je cilj da pomažu organizaciju sadržine dokumenta i omoguće lakše praćenje. Veličine i karakteristike ispisa različitih tipova naslova zavise od korišćenog čitača. Oznaka za nivo naslova je `<H*>... </N*>`.

Veb serveri zahtevaju sva četiri znaka u nastavku imena datoteke da bi prepoznali nastavak .htm u vezama unutar Veb dokumenta. Ako se Veb strana nalazi na serveru koji radi pod operativnim sistemom Windows, server će ignorisati četvrto slovo u nastavku (slovo "l"). Ako se sa Windows računara prenose datoteke s nastavkom .htm na neki UNIX server, nastavci se moraju promeniti u .html, ili izvršiti proveru da taj server prepoznaje datoteke s nastavkom .htm kao HTML dokumente.

U *HTML* dokumete danas se integrišu delovi Veb aplikacija napisani drugim jezicima kojima se postižu različiti dodatni efekti. Tako na primer, dokument može da sadrži animirane sekvence koje se jednostavno ubacuju kao sekvence slika u *GIF* formatu, zatim deo koda napisan kao *JAVA* apleti koji omogućavaju brzi prikaz animirane grafike i drugih specijalnih multimedijalnih efekata. Primenom *JAVA-script* jezika Veb dokument postaje aktivan i može se ostvariti interakcija između aplikacije klijenta i servera.

Svi nabrojani mehanizmi, uz još mnogo novih, predstavljaju jedan novi koncept koji nalazi sve širu primenu i nosi zajednički naziv Internet (ili Veb) tehnologije.

7.5.3 HTML, kategorije

HTML sadrži veliki broj oznaka koje se mogu svrstati po kategorijama. Kategorije pomažu da se objasni gde i kada se koriste pojedine HTML oznake. Veliki broj oznaka se koristi u parovima, korišćene su tri tačke (...) između zatvarajuće i otvarajuće oznake.

nake kako bi se ukazalo na to gde se mogu pojaviti tekst i ostali elementi.

Atribut unutar **HTML** oznaka obično se javlja u jednom od sledeća dva oblika:

1. *ATRIBUT* = "vrednost". U ovom slučaju je vrednost sadržaj koji je koji je ovičen znacima navoda, i može pripadati jednoj od sledećih vrsta;

- **URL:URL** adresa.
- **Ime:** Ime koje definiše korisnik, obično nekog ulaznog polja.
- **Broj:** Numerička vrednost koju definiše korisnik.
- **Tekst:** Tekst koji definiše korisnik.
- **Server:** Ime koje zavisi od servera
- **(X\Y\Z):** Jedan element iz skupa fiksnih vrednosti.
- **#rrggbb:** Boja predstavljena u heksadecimalnom obliku.

2. *ATRIBUT*: Samo ime atributa daje informaciju o tome kako će se oznaka ponašati (na primer, prisustvo atributa *ISPAM* unutar oznake *< IMG >* znači da grafika predstavlja mapu ostljivu na dodir). Na primer, u oznaci *< IMG SRC = "sample.gif" >*, *SRC* je atribut koji se mora navesti. Zancima navoda je ovičena adresa datoteke koja sadrži sliku. Oznaka *< IMG >* stoji sama za sebe, ne postoji zatvarajuća oznaka *< /IMG >*, te nije naveden par oznaka, kao što se to obično čini.

ALIGN = TOP\MIDDLE\BOTTOM znači da da atribut *ALIGN* može uzeti samo jednu od tru ponuđene vrednosti: *TOP*, *MIDDLE* ili *BOTTOM*. Ukoliko se atribut odnosi na oznaku ** time se odlučuje da li će slika biti prikazana na vrhu, u sredini ili na dnu ekrana.

U Tabeli 7.1 su obuhvaćene oznake strukture dokumenta koje se koriste da bi se u **HTML** dokumentu primenili traženi elementi strukture. Na raspolaganju su naslovi različitih veličina i stilova, kojima je cilj da pomognu organizaciju sadržine dokumenta i učine je lakšom za praćenje. Važi konvencija po kojoj je naslv nivoa 1 krupniji od i naglašeniji od naslova nivoa 2 itd. Međutim U Tabeli 7.1 su navedeni samo naslovi prvog (najvećeg) nivoa i šestog (najmanjeg) nivoa. Podrazumeva se da postoje i naslovi drugog, trećeg, četvrtog i petog nivoa.

Tabela 7.1: Globalna struktura HML kategorije i odgovarajuće

Oznaka	Ime oznake	Kratko objašnjenje
<code><ADDRESS>... </ADDRESS></code>	Adresa	Informacije za obraćanje autoru
<code><BODY>... </BODY></code>	Telo	Oivičava telo dokumenta
<code><DIV>... </DIV></code>	Odeljak	Označava odeljke u dokumentu i omogućuje primenu stilova na te odeljke.
<code><H1>... </H1></code>	Naslov nivoa 1	Naslov nivoa 1
<code><H2>... </H2></code>	Naslov nivoa 2	Naslov nivoa 2
<code><H6>... </H6></code>	Naslov nivoa 6	Naslov nivoa 6
<code><HEAD>... </HEAD></code>	Заглавље	Oivičena zaglavlja dokumenta
<code><HTML>... </HTML></code>	HTML	Oivičava se ceo HTML dokument
<code>... </code>	Opseg	На оивичени текст примен- Na oivičen tekst primenjuje se stil specificiram oznakom <code><STILE></code>
<code><TITLE>... </TITLE></code>	Naslov	Naslov celokupnog dokumenta
<code><!--...--></code>	Komentar	Umeće autorove komentare koje Veb čitač ignoriše

Napomenimo da se naslov dokumenta ispisuje u naslovnoj liniji prozora čitača. Ako se naslov ne definiše, podrazumevani naslov biće ime HTML datoteke. Samo običan tekst može se nalziti unutar `<TITLE>... </TITLE>`, jer se saržina tog para oznaka ispisuje samo u naslovnoj liniji prozora Veb čitača.

Komentar se koristi radi dokumentovanja upotrebljenih oznaka, ostavljanja napomena za kasniju obradu ili za davanje dodatnih informacija onima koji budu čitali izvorni tekst. Komentari se mogu koristiti za privremeni uklanjanje delova dokumenta tako što se tekst koji treba ukloniti oiviči oznakama komentara.

Oznake vezane za tekst date su u Tabeli 7.2. Ove oznake obezbeđuju logičku strukturu sadržine. Tom strukturom se može, ali ne mora izmeniti način prikazivanja sadržine HTML dokumenta.

Oznaka `<PRE>... </PRE>` primorava čitač da tekst ispiše

Tabela 7.2: Oznake vezane za HTML kategorije

Oznaka	Ime oznake	Kratko objašnjenje
<code><ABBR>... </ABBR></code>	Skraćenica	Označava skraćenicu
<code>
</code>	Novi red	Prelaz u novi red prilikom ispitivanja teksta na ekranu
<code><CITE>... </CITE></code>	Kratak citata	Označeni tekst koji označava citat
<code><CODE>... </CODE></code>	Font za listinge	Koristi se za delove listinga programa
<code><DFN>... </DFN></code>	Definisani izraz	Koristi se za naglašavanje izraza koji će biti definisan u nastavku teksta
<code><P>...</P></code>	Pasus	Deljenje teksta u razdvojene oblasti
<code><PRE>... </PRE></code>	Prethodno formatiran tekst	Čuva izgled originalnog teksta u fontu fiksne širine znakova ширине знакова
<code><SUB>... </SUB></code>	Indeks	Ispitivanje oivičenog teksta sitnijim zakovima nešto niže u odnosu na ostatak teksta
<code><SUP>... </SUP></code>	EkspONENT	Ispitivanje oivičenog teksta sitnijim znakovima i nešto više u odnosu na ostatak teksta
<code><VAR>... </VAR></code>	Promenljiva	Označava promenljivu zamenu za neku drugu vrednost

koristeći njegovo originalno formatiranje, uvlačenje i razmake. To je korisno za prikazivanje već formatiranih tabela i ili programskih listinga. Dužinae redova ne treba da budu veće od 80 znakova, pošto se oivičeni tekst obično ispisuje fontom fiksne širine znakova. Ova oznaka je veoma pogodna za ispisivanje čisto tekstualnih informacija.

Sledeća HTML kategorija prikazana u Tabeli 7.3, se odnosi na spiskove, odnosno na obezbeđivanje metoda za prikazivanje nizova elemenata unutar sadržine dokumenta. Dakle, oznake u ovoj kategoriji se koriste za definisanje raznorodnih spiskova.

tABELA 7.3: Oznake vezane za HML spiskove

Oznaka	Ime oznake	Kratko objašnjenej
< DD >	Definicija	Označava definiciju izraza y u spisku tipa rečnika
< DIR > ...	Spisak tipa	Definiše neobeleženi
< /DIR >	direktori- juma	spisak kratkih elemenata (dužine ispod 20 znakova)
< DL > ...	Spisak tipa	Definiše specijalni format
< /DL >	rečnika	namenjen navođenju izraza i njihovo objašnjenje
< DT >	Izraz	Označa izraz koji se definiše u spisku tipa rečnika
< LI >	Element spiska	Označava element spiska bilo kog tipa
< MENU > ...	Spisak tipa	Definiše spisak elementa
< /MENU >	menija	između kojih se može birati
< OL > ...	Numerisani	Definiše spisak numerisa-
< /OL >	spisak	nih elemenata

U Tabeli 7.3 su navedene kategorije koje obezbeđuju postupke za prikazivanje nizova elemenata unutar sadržine dokumenta. Ako je u pitanju spiska < OL > će biti označen rednim brojem. Način ispisivanja rednog broja može se podesiti atributom *TYPE*. Na sličan način, u slučaju nenumerisanog spiska (< UL >), atributom *TYPE* može se podesiti vrsta znaka kojom će biti označeni elementi spiska. Spiskovima tipa < DIR > i < MENY > ne može se podesiti vrsta znaka kojom će biti označeni elementi spiska. Spiskovima tipa < DIR > i < MEMY > ne može se upravljati na ovaj način, jer se od njih ne zahteva da im elementi budu označeni.

Atrubut *TYPE* = (1|a|A|i|I) definiše arapske brojeve, mala slova, velika slova, male rimske brojeve i velike rimske brojeve, respektivno. Arapski i rimski brojevi počinu od jedinice, dok slova počinju sa **a**.

Primer 7.5 *Nenumerisani ili obeležvni spisak vrlo je pogodan za isticanje informacija organizovanih u nekoliko redova. Sledeći nenumerisani spisak može se prikazati pomoću programa Netscape Navigator*

```

< UL >
< LI>    Ovo je prvi red
< LI >   Ovo je prvi red
< LI >   I ovo
< /UL >

```

Oznake Tabela definišu strukturu i izgled tabela. Tabele se mogu koristiti za uređivanje podataka ili kao pomoć u formiranju strane i organizacije njenog izgleda. U Tabeli 7.4 dat je spisak oznaka koji se koristi za upravljanje tabelama.

Tabela 7.4: Oznake vezane za HML tabelama

Oznaka	Ime oznake	Kratko objašnjenje
<code>< CAPTION >... < / CAPTION ></code>	Natpis tabele	Definiše natpis koji odgovara tabeli, a nalazi se izvan nje
<code>< COL ></code>	Stupci	Definiše osobine stubaca unutar grupe stubaca
<code>< COLGROUP ></code>	Grupe stubaca	Definiše osobine grupe stubaca
<code>< TABLE >... < / TABLE ></code>	Tabela	Formira tabelu
<code>< TBODY... < / TBODY ></code>	Telo tabele	Definiše telo tabele onda kada su definisana i podnožja
<code>< TD >... < / TD ></code>	Ćelije u tabeli	Oivičava podatke i upravljačke elemente koji pripadaju ćeliji unutar tabele
<code>< TFOOT >... < / TFOOT ></code>	Podnožje tabele	Definiše podnožje tabele onda kada su definisani telo i zaglavlje
<code>< TH >... < / TH ></code>	Zaglavlje Stupca	Piivičava podatke i upravljačke elemente koji pripadaju zaglavlju stupca
<code>< THEAD >... < / THEAD ></code>	Zaglavlje tabele	Definiše zaglavlje tabele onda kada su definisani telo i podnožje

Tabela je prazno dok se u njoj ne formiraju redovi ćelija korišćenjem oznaka `< TR >` (table row) kojom se definiše svaki red u tabeli, `< TD >` (table data) kojom se definiše svaka ćelija u tabeli i `< TH >` (table header) koja se koristi za označavanje

pojedinih stubaca i/ili redova u tabeli. Svaku ćeliju treba zatvoriti oznakom `< /TD>` jer to olakšava čitanje. Ako je tabela uključena u ćeliju druge tabelle, treba zatvoriti sve ćelije i redove; u protivnom pojedini Veb čitači neće na ispravan način obraditi umetnute tabelle. Veb čitači obično na različit način obrađuju prave ćelije od ćelija u kojima se nalazi razmak. Ako se u ćeliji tabelle koristi slika, ne treba definisati vrednost atributa `< WIDTH >` i `< HEIGHT >` unutar oznake `< IMG >`, čime se omogućava Veb čitacu da počne iscrtavanje tabelle pre nego što slika bude učitana.

Oznaka `< TABLE >` prihvata attribute `ALIGN`, `BORDER`, `CELLPADDING`, `CELLSPACING` `WIDTH`. Oznaka `< TD >` prihvata attribute `ALIGN`, `COLSPAN`, `HEIGHT`, `NOWRAP`, `ROWSPAN`, `VALING` i `WIDTH`

Primer 7.6 *Sledeći Veb dokument generiše tabelu sa heksadecimalnim vrednostima za belu, crvenu i zelenu boju pozadine. Tabela zauzima 50% širine prozora.*

```
< TABLE BORDER="1" width="50%" >
< CAPTION > Tabela boja < /CAPTION >
< COLGROUP >
< COL ALIGN= RIGHT >
< COL ALIGN= LEFT >
< THEAD > < TR >
< TH width="67%" > Heksadecimalna vrednost < /TN >
< TH width="33%" > Boja < /TN >
< /TR > < / THEAD >
< TBODY > < TR >
< TD width="67%" > FFFFFFFF < /TB >
< TD width="33%" > Belo < /TD >
< /TR >
< TR > < TD width="67%" > FF0000 < /TD >
< TD width="33%" > Crveno < /TD >
< /TR > < /TBODY >
< TF00T > < TR >
< TV width="67%" > 00FF00 < /TD >
< TD width="33%" > Zeleno < /TD >
< /TR > < /TF00T >
< /TABLE >
```

Na Slici 7.11 prikazan je tabela boja.



Сл. 7.11: Tabela boja pozadine

Oznakama, koje se odnose na upravljanje prikazivanjem, menja se način prikazivanja sadržine, uticajem na stilove fontova ili iscrtavanjem horizontalnih linija. U tabeli 7.5 date su oznake koje se odnose na prikazivanje teksta u HTML

Tabela 7.5: Oznake vezane za prikazivanje teksta u HML dokumentima

Oznaka	Ime oznake	Kratko objašnjenje
<code>< B > ... < / B ></code>	polucrno	proivodi polucrno ispisan tekst
<code>< CENTER > ... < / CENTER ></code>	Centriranje	Centrira oivičeni tekst
<code>< HR ></code>	Horizontalna linija	Crta horizontalnu liniju preko cele strane
<code>< I > ... < / I ></code>	Kurziv	Proizvodi tekst ispisan kurzivom
<code>< S > ... < S ></code>	Precrtan tekst	prikazuje oivičen tekst precrtano
<code>< SMALL > ... < / SMALL ></code>	Sitan tekst	Čini tekst sitnijim
<code>< TT > ... < / TT ></code>	Pisaća mašina	Proizvodi tekst ispisan fontom koji odgovara pisaćoj mašini
<code>< U > ... < / U ></code>	Podvučeno	Podvlači oivičen tekst

dokumentu.

Pored navedenih oznaka postoje još oznake koje se odnose na

okvire u oblasti čitača koje služe za prikazivanje i upravljanje njima, zatim oznake obrazaca kojima se upravlja unošenje podataka, kao i oznake veze kao što su vv opisano sidro `< A >... </ A >` koje predstavlja vidljivi upravljački element koji povezuje jedan resurs Veb stranice i veza `< LINK >... </ LINK >` koja utvrđuje odnose između tekućeg i drugih dokumenata.

7.6 FrontPage

Microsoft poseduje dva alata koji mogu biti veoma korisni prilikom kreiranja i objavljivanja Veb stranica. Jedan je FrontPage koji omogućuje korisniku brzo i jednostavno kreiranje Veb dokumenta. Drugi je Web Publishing Wizard. On omogućava korisniku postavljenje Veb stranica na Internet.

FrontPage je Microsoft-ov program za izradu Veb dokumenta. Dokument je pisan u programskom jeziku *HTML*. U FrontPage-u, *HTML* stranice izrađuju se kao što se izrađuju Word dokument u programskom paketu WORD. FrontPage je alat koji ne zahteva programersko znanje. Za izradu Veb dokumenta koriste se FrontPage Explorer i FrontPage Editor.

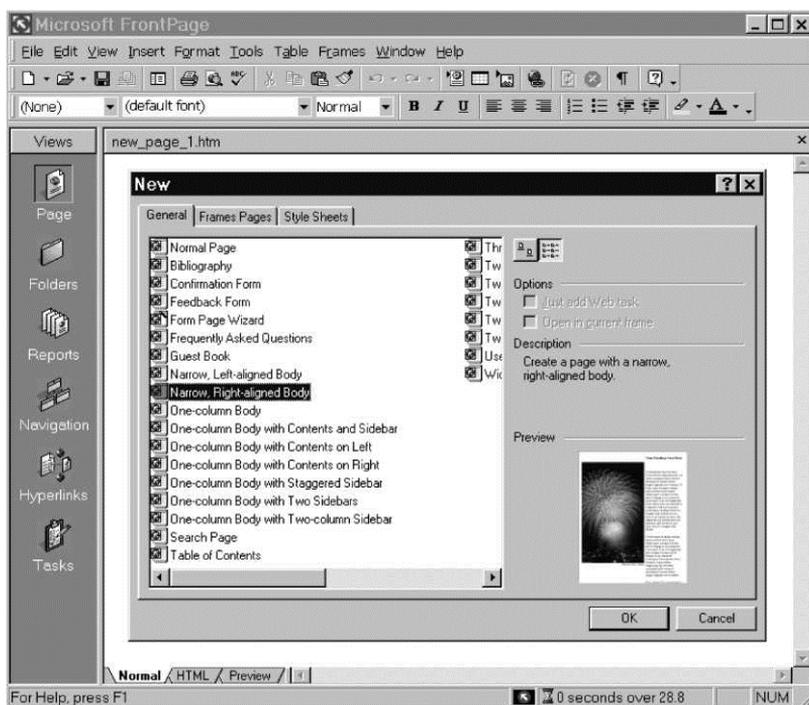
FrontPage Explorer se koristi za kreiranje i pregledanje Veb dokumenta, ali i za objavljivanje na lokalnom računaru, mreži ili na Internetu. FrontPage Explorer sadrži nardebe za administriranje Veb stranica, testiranje hiperlinkova, pregled svih Veb foldera, dodavanje objekata i pokretanje ostalih aplikacija za kreiranje veba.

FrontPage Explorer Prikazuje tekstove, slike, tabele, forme i druge elemente na WWW serveru tačno onako kako su definisani na korisničkim stranicama.

7.6.1 Kreiranje Veb stranice

Korisnik pokreće **FrontPage** izborom **Programs/Microsoft FrontPage** u **Start** meniju, ili klikom na ikonu **FrontPage** koja se nalazi na desktopu računara. Nakon pokretanja programa, korisnik dobija dijalog okvir koji se naziva i FrontPage editor. Delovi prozora koji se pojavljuju su isti kao kod ostalih programa Microsoft-a: naslovna traka, traka sa menijima, dugmad za zatvaranje prozora,

traka sa alatima i statusna traka. Ostali delovi prozora su specifični za FrontPage. Na Slici 7.12 prikazana je radna površina FrontPage-a sa predlošcima za kreiranje Veb stranica.



Sl. 7.12: Radna površina FrontPage-a sa ponuđenim predlošcima za kreiranje Veb stranica

Levu stranu ekrana zauzima traka koja sadrži nekoliko vrsta pogleda na celokupnu Veb stranicu koju korisnik kreira:

1. **Page** - pogled u kojem se kreira i edituje Veb stranica upravo onako kako će izgledati u Vep pretraživaču. Ovaj pogled se otvara pokretanjem programa.
2. **Folders** - pokazuje mape i datoteke koje su sastavni deo Veb stranice.
3. **Reports** - služi za kreiranje izveštaja na Veb stranicama.
4. **Navigation** - pokazuje veze među stranicama.
5. **Hyperlinks** - prikazuje stranice koje imaju nepovezane linkove.

6. **Task** - lista za zadacima osobe koja kreira Veb stranice.

Ukoliko korisnik ne želi koristiti traku za pregled, može je ukloniti sa ekrana klikom na desni taster miša bilo gde unutar otvorenog prostora na traci ili iz glavnog menija izabere naredbu **Hide Views Bar**. Traka se ponovo vraća u okvir dijaloga ako se u meniju aktivira opcija **View bar**.

FrontPage editor nudi tri vrste pregleda stranica koju korisnik kreira:

1. **Normal** je standardni pogled editora, svaki dokument koji korisnik kreira ili menja otvoriće se u ovom pregledu.
2. **HTML** prikazuje izvorni HTML kbd dokumenta koji se kreira. Koristi se za ručno menjanje kbda. Promene koje se učine u ovom kbdu odražavaju se na **Normal** i **Preview** pre-gled stranice.
3. **Preview** prikazuje stranicu tačno onako kako izkleda u pretraživaču; i u njemu se može uređivati stranica.

7.6.2 Editovanje stranice u FrontPage-u

U FrontPage-u korisnik edituje stranicu upisujući tekst, ubacujući datoteke, slike, tabele druge elemente koristeći menije i liniju alata.

Kreiranje hiperlinkova

Kada se obeleži neki deo teksta, slika ili drugi objekat, pritiskom na prečicu < *Ctrl* > + < *K* > ulazi se u dijalog **Umetanje hiperveze**. U paleti grupa **poveži** treba odabrati cilj hiperveze: to može biti neka datoteka ili Web stranica, mesto u dokumentu koje je obeleženo naslovom ili obeleživačem, novi dokument koji će biti kreiran nakon pravljenja veze ili adresa e-pošte; u zavisnosti od izbora, centralni deo dijaloga će biti unosu potrebnih podataka.

Za kreiranje hiperlinkova najpre treba otukucati tekst linka na radnoj površini programa, a zatim koristiti naredbu **Hiperlink** iz padajućeg menija **Insert**. U sledećem koraku se unosi adresa linka u Address polje teksta i klikom na taster OK link će biti

ubačen u dokument što se potvrđuje podvlačenjem teksta koji je otkucan.

Tabele

Najjednostavniji način za kreiranje tabele u FrontPage-u jeste da korisnikne mišem na dugme **Insert Table** u traci sa alatkana. Nakon pojavljivanja matrice, mišem se može birati željeni broj redova i kolona. Otpuštanjem miša omogućva se kreiranje nove tabele.

Drugi način stvaranja tabele je nešto složeniji ali pruža potpunu kontrolu na tabelom. Postupak se satoji iz sledećih koraka:

- Odabrati komandu **Insert Table** u padajućem meniju **Table**, nakon čega se otvara okvir za dijalog u kojem se određuje broj redova i kolona.
- Grupa komandi **Layout** nudi korisniku niz opcija za podešavanje tabele.

Kada se želi dodavanje ili upisivanje teksta u tabelu, treba jednostavno kliknuti na ćeliju u koju treba uneti teks, a zatim se upisuje tekst. FrontPage poseduje impresivni repertoar komandi za doterivanje tabele koje se nalaze u padajućem meniju table.

Meniji

Kreiranje menija tj. tastera korisnicima omogućava jednostavno kretanje kroz Web prezentaciju i vezu sa ostalim stranama u prezentaciji. Izborom **Web Component/Hover Button** iz padajućeg menija: **Insert** moguće je kreirati potreban broj tastera.

U ponuđenom dijalog okviru korisnik može uneti tekst i naziv strane na koju će prelaziti nakon klika na dati taster.

Pored toga ovde se podešava boja, efekat pojavljivanja tastera i slicno. Nakon kreiranja potrebnih tastera, najbolje je da korisnik odmah kreira i taster koji će omogućiti vezu sa elektronskom adresom institucije ili pojedinca. U polju **Link** dijalog okvira **Hover Button** potrebno je da korisnik unese tekst elektronske adrese.

7.7 Mreže za integrisane usluge

Budućnost u komunikacijama predstavljaju mreže za integralno opsluživanje (BISDN – Broadband Integrated Services Data Network) koje treba da omoguće prenos različitih tipova informacija (multimedijalne informacije), odnosno teksta, zvuka (muziku i govor), digitalizovane slike, animacije i televizijske slike. Na ovom planu danas su najznačajnije ATM mreže (Asynchronous Transfer Mode). To su širokopojasne mreže koje treba da omoguće potpunu integraciju svih tipova podataka i uvođenje interaktivnih multimedijalnih komunikacija, (npr. interaktivne televizije ili interaktivnog filma).

7.7.1 Digitalni prenos -18V[^]

ISDN - Integrated Services Digital Network je brz i pouzdan način za komunikaciju kućnog korisnika, odnosno male firme, sa Internet provajderom, dok za Internet provajdere ISDN osnovni način komunikacije preko telefonske veze. Instaliraju se dve vrste ISDN priključka. Primarni koji koriste Internet provajderi i velike firme, kojima preko dve telefonske parice dovodi čak 30 linija koje se mogu koristiti za zvanje i prijem poziva. Za pojedinačne korisnike i manje firme koristi se bazni priključak.

Uvođenje baznog ISDN priključka ne zahteva novu paricu između korisnika i centrale; koliko klasična analogna već postoji, već se ona zamenjuje za ISDN liniju koja koristi istu paricu. Pri proceni tehničkih mogućnosti, Telekom proverava da li je priključak dvojnički ili je priključen preko FM/PCM uređaja. U oba slučaja je potrebno razdvajanje od dvojnika jer je za ISDN potrebna parica bez dvojnika koja povezuje korisnika sa centralom. Takođe je potrebno da centrala bude digitalna.

Telefonska parica se povezuje sa odgovarajućom digitalnom opremom u centrali i uređajem koji se zove mrežni završetak. Modeli mrežnih završetaka se razlikuju po tome da li njima postoje analogni ulazi, za standardne telefonske aparate, i od signalizacije koja se nalazi na kućištu. Postoje mrežni završeci sa USB ulazom, koji obezbeđuje direktnu vezu sa računarom. Za mrežni završetak potrebno je napajanje od 220 V.

Mrežni završetak obično ima dva S-bus priključka na koji mogu da se povežu ISDN uređaji; na isti mrežni završetak može da se poveže do osam uređaja. Računar se povezuje preko terminal adaptera koji može biti eksterni ili interni - interni je najčešć PCI kartica koja se umeće u računar, dok se eksterni povezuje na USB ulaz.

Ukoliko je cilj uvođenja ISDN-a povezivanje lokalne mreže firme na Internet, tada se umesto terminal adaptera koristi ISDN ruter. Kada neka stanica zahteva pristup Internetu, ruter će pozvati Internet provajdera u obezbediti traženu uslugu, a zatim, kada neko vreme nema zahteva, prekida vezu i tako smanjuje troškove telefonskih impulsa i provajdinga.

ISDN telefon nije neophodan; mrežni završetak ima jedan analogni ulaz na koji može da se priključi standardni analogni telefon preko koga se telefonira koristeći ISDN liniju. Ipak, pogodno je nabaviti ISDN telefon jer on omogućava ispis broja koji poziva, ispis cene vođenog razgovora, liste obavljenih razgovora, telefonski imenik sagovornika, preusmeravanje poziva, prikaz datuma i vremena itd. Digitalni telefoni su kvalitetni uređaji sa ekranom i nizom tastera koji mogu da olakšaju telefoniranje.

ISDN se uvodi radi telefoniranja i prenosa podataka. Izborom protokola prenos podataka se obavlja jednim kanalom brzinom 64 kb/s, drugi kanal je slobodan za telefoniranje. Ako treba da se prenese veća količina podataka, tada se za poziv mogu koristiti oba kanla. U tom sličaji ISDN broj je zauzet za pozive.

7.7.2 Spedijalizovani telekomunikadioni servisi

Razvoj komunikacija prati veoma brzi razvoj različitih specijalizovanih telekomunikacionih servisa u okviru kojih se klasične telekomunikacione usluge kombinuju sa korišćenjem računara. Nabrojaćemo kratko neke najznačajnije servise.

Teletekst

Teleteks je sistem koji omogućava prenos podataka (teksta i tablica) integrisano sa televizijskom slikom. Ovi podaci umeću

se u osnovni televizijski signal tako što se koriste intervali u osnovnom signalu predviđeni za vraćanje elektronskog mlaza sa dna slike na vrh. Posebnim adapterom u televizijskom prijemniku postiže se da se ova informacija vidi na ekranu televizijskog prijemnika. Ovakvi sistemi već imaju široku primenu za prenos informacija (elektronske novine), reklama, oglasa i slično.

Videotekst

Videotekst sistemi su sistemi u kojima se klasičan televizijski prijemnik, uz dodatne adaptore pretvara u neku vrstu video terminala koji se može koristiti za interaktivnu komunikaciju u okviru različitih servisa za informisanje (redovi vožnje), kupovinu (karata npr.) ili samo pretraživanje baza podataka. Samtra se da je budućnost ovih sistema u njihovom integrisanju u Internet okruženje.

Audiotekst

Audiotekst sistemi omogućavaju korišćenje javne telefonske mreže za dobijanje unapred pripremljenih zvučnih informacija. Ovi sistemi se danas već široko koriste u okviru službi tačnog vremena, za dobijanje informacija o kursnoj listi, repertoaru bioskopa isl. Širom primenom računara u okviru ovih sistema mogu se znatno proširiti njihove mogućnosti tako da se mogu primenjivati za dobijanje informacija iz baza podataka (npr. stanje na tekućim računima u bankama i si.)

Elektronska razmena podataka (EDI)

EDI je skraćenica za elektronsku razmenu podataka (Electronic Data Interchange), odnosno međunarodni standard za elektronsku razmenu poslovnih podataka između poslovnih partnera kakvi su: proizvođači, izvoznici, trgovina, distributeri, transportni sistemi, banke, osiguravajuće organizacije, vladine agencije, carina i sl.

Najpoznatiji podskup EDI standarda je EDIFACT (Electronic Data Interchange For Administration, Commerce and Transport) - elektronska razmena podataka za administraciju, trgovinu i transport.

Ovaj standard je izrađen pod pokroviteljstvom Ujedinjenih nacija i definiše standardne formate elektronskih poruka kako za nacionalne tako i za međunarodne komunikacije u oblasti administracije trgovine i transporta.

Takođe postoji međunarodni standard za elektronski prenos finansijskih sredstava EFT (Electronic Funds Transfer), koji omogućava plaćanje elektronskim putem i sve ostale novčane transakcije.

Danas se takođe mnogo radi na primeni Internet tehnologija i Interneta u oblasti novčanih transakcija u okviru sistema trgovine preko Interneta. Posebno su značajni problemi zaštite koji postoje u jednom ovako otvorenom sistemu kakav je Internet.

Sistemi za pretraživanje podataka

Savremene informacione tehnologije imaju veoma značajnu primenu u sistemima naučno tehničke dokumentacije i drugim sistemima zasnovanim na velikim bazama podataka kao što su katalogi stanovništva, medicinska dokumentacija, sudski dosijei i sl. Mnogi od ovih sistema su danas organizovani tako da nude javne usluge i pristup preko telekomunikacionih mreža. To su takozvani sistemi za pretraživanje informacija (Information Retrieval Systems).

Najznačajniji iz ove grupe servisa su sistemi za pretraživanje bibliografskih podataka. Organizovani su u okviru najznačajnijih svećkih bibliografskih centara i sadrže ogromne baze bibliografskih jedinica iz svih naučno-tehničkih oblasti. Dokumenti su u ovim bazama registrovani po međunarodnim standardima definisanim u te svrhe, a za pristup bazama koriste se posebni protokoli i specijalizovani upitni jezici za pretraživanje.

Slično bibliografskim sistemima postoje i sistemi sa bazama podataka iz drugih oblasti: statističke baze, baze sa cenama proizvoda, redovima vođnje i sl.

Danas se ovi sistemi integrišu u Internet mrežu i postaju dostupni preko Veb servisa. To je trend koji je neprekidno u porastu i dobija sve nove i nove oblike. Biznis preko Interneta je jedan od novijih koncepata koji se zasniva na mogućnostima pretraživanja velikih baza proizvoda i transakciji novca preko In

terneta.

Na Internetu recimo jako uspešno funkcioniše koncept najveće knjižare na svetu (Amazon.com) preko koje se mogu dobiti informacije o preko 2 miliona naslova knjiga i najveći deo njih i odmah kupiti.

7.8 Mrežne alatke

Veliko broj mrežnih alatki može se koristiti za obavljanje zadataka kao što su dobijanje informacija o drugim sistemima u mreži, pristupanje drugim sistemima i direktno komuniciranje sa drugim korisnicima. Informacije o mreži mogu se dobiti pomoću alatki kao što su ping, finger, traceroute i host. Mrežni klijenti Talk, ICQ, IRS omogućavaju dikomunikaciju sa drugim korisnicima u mreži. Telnet obavlja udaljeno prijavljivanje na nalog koji se nalazi udaljenom sistemu u mreži. Pred toga mreža može da koristi mrežne komande za udaljeni pristup. One su veoma korisne kod malih mreža i omogućavaju direktno pristupanje udaljenim računarima radi kopiranja datoteka ili izvršenja komadi.

7.8.1 Informadije o mreži

Komande **ping**, **finger**, **Traceroute** i **host** mogu se koristiti za prinalaženje informacija o stanju sistema i korisnika mreže. Komanda **ping** sa koristi radi provere da li je udaljeni sistem uključen da li radi. Komanda **finger** se koristi radi pronalaženja informacija o drugim korisnicima u mreži, pri čemu se proverava da li su prijavljeni ili da li su primili poštu. Komanda **host** prikazuje informacije o adresi sistema u mreži, pri čemu daje IP adresu siatema i adresu domena. Komanda **Traceroute** može da se koristi za praćnje prolaska poruke kroz niz računarskih mreža i sistema.

Komanda **ping** detektuje da li je sistem uključen i da li radi. Kao argument uzima ime sistema koji se proverava. Ako je sistem koju se proverava isključen, komanda **ping** proizvodi poruku o istoku vremena, koja ukazuje na to da veza nije mogla da se uspostavi. Sledeći primer proverava da li je **www.yahoo.com** uključen i povezan sa mrežom:

ping www.yahoo.com

Komada **ping** može da se koristi sa IP adresom umesto sa imenom domena. Pomoću IP adrese komanda **ping** može direktno da detektuje udaljeni sistem, pri čemu ne mora da ide preko servera imena domena radi prevođenja imena domena u IP adresu. To može biti korisno u situacijama u kojima je server imena domena lokalne mreže privremeno isključen.

Komanda **finger** može da se koristi za dobijanje informacija o drugim korisnicima u mreži, a komanda **who** za dobijanje informacija koji korisnici rade na sistemu. Komanda **who** prikazuje spisak svih korisnika koji su trenutno prijavljeni sa podacima o tome kada su se prijavili, koliko dugo su prijavljeni i gde su se prijavili. Komanda **who** je namenjena za rad na lokalnom sistemu ili mreži. Komanda **finger** može da radi na velikim mrežama, uključujući Internet, iako većina sistema blokira tu komandu iz bezbedonosnih razloga.

Pomoću komade **host** može se proveriti informacija o mrež-noj adresi koja se odnosi na udaljeni sistem koji je povezan sa našom mrežom. Ta informacija se obično sastoji od IP adrese sistema, adrese imena domena, nadimaka za ime domena i poštanskog servera. Te informacije se dobijaju od servera domenskih imena te mreže. U slučaju Interneta, to obuhvata sve sisteme sa kojima može da se ostvari adresa preko Interneta.

Komanda **host** predstavlja efikasan način određivanja IP adrese udaljene lokacije ili URL-a. Ako je poznata IP adresa lokacije, pomoću komande **host** može se zaznati njeno ime domena

host www.yahoo.com

Internet veze se uspostavljaju preko različitih ruta, pri čemu prolaze kroz čitav niz međusobno povezanih računara koji imaju ulogu mrežnih prolaza. Putanja od jednog do drugog sistema može da ima različite rute, pri čemu neke rute mogu biti duže od drugih. U slučaju spore veze, može se koristiti komanda traceroute radi provere preko koje rute je ostvarena mreža sa računarom u mreži, pri čemu se ima uvid u brzinu i broj uzastopno ostvarenih veza sa mrežnim prolazima u okviru rute. Komanda traceroute kao argument uzima mrežno ime računara ili IP adresu sistema čija ruta se proverava. Postoje opcije za određivanje parametara kao što su tip servera (-t) ili izvorni računar (-t). Komanda

tracert vraća spisak mržnih računara preko kojih ide ruta sa vremenima za tri provere koje se šalju u svakom prolazu. Vremena koja su duža od pet sekundi se predstavljaju zvezdicom, *

Tracert www.yahoo.com

Za obavljanje komande **ping** i praćenje ruta operativni sistem Windows HP sadrži altku *Run...* koja se dobija pritiskom na ikonu Start u levom donjem uglu ekrana.

7.8.2 Mrežni klijenti

Direktna komunikacija sa drugim korisnicima u mreži može se realizovati pomoću alatki Talk, ICQ ili IRC, pod uslovom da je i drugi korisnik u tom trenutku prijavljen na sistem koji je takođe ostvario vezu. Program Talk radi kao telefon, tako što omogućava direktan dvosmerni razgovor sa drugim korisnikom. Alatka Talk projektovana je za korisnike istog sistema ili korisnike koji se nalaze u lokalnoj mreži. ICQ (engl. I Seek You) predstavlja Internet altku koja obaveštava o trenutku uspostavljanja veze sa mrežom drugih korisnika i omogućuje komunikaciju sa njima.

Protokoli ICQ omogućavaju direktnu komunikaciju sa ostalim korisnicima na mreži, i služi i kao alatka za brzo prenošenje poruka. Pomoću ICQ klijenta korisnicima mogu da se šalju poruke, da se ćaska sa njima i da se šalju datoteke. Postoji mogućnost podešavanja spiska korisnika sa kojima se želi kontaktirati. Da bi se koristio ICQ, treba se registrovati na ICQ serveru koji obezbeđuje ICQ broj, koji se naziva UIN (Universal Internet Number)

Pomoću alatke IRC (Internet Relay Chat) možemo se povezati sa udaljenim serverom sa kojim su povezani i ostali korisnici i da razgovaramo sa njima. ICQ radi kao prostor za ćaskanje (engl. chat room), gde se mogu odabrati kanai i razgovarati sa drugim korisnicima koji su već tu. Nejpre treba izbrati ICQ server sa kojim treba da se ostvari veza. Na raspolaganju su različiti serveri zavisno od lokacije, kao i različite teme. Kada se ostvari veza sa serverom, iz odgovarajućeg spiska se može odabrati kanal kome ćemo pristupiti. Interfejs radi slično kao prostor za ćaskanje. Kada se ostvari veza sa serverom može se odabrati nadimak za predstavljanje.

7.9 Pitanja za proveru znanja

1. Objasniti razliku između paralelnog i serijskog prenosa podataka.
2. Objasniti princip asinhronog prenosa. Koje su prednosti, a koji nedostaci ovakvog načina prenosa podataka?
3. Proceniti vreme odlaska i povratka (tzv. round trip time) u slučaju tačka-tačka veze ako se obavlja asinhroni prenos brzinom 1 200 b/s, a dužina paketa iznosi 86 bajtova. Dužina tačka-tačka veze je 2 000 km.³
4. Objasniti princip sinhronog prenosa. Koje su prednosti, a koji nedostaci ovakvog načina prenosa podataka.
5. Kakav prenos podataka se koristi u nutar računara, a kakav između računara. Kako se naziva brzina kojom predajnik šalje podatke u kanal i kako se izražava.
6. Pronađi jednu *WWW* stranicu i jednu sliku *JPEG* formata na temu "Internet i obrazovanje". Stranicu memorisati pod imenom primer.html sliku pod imenom slika.jpg u neki od direktorijuma lokalnog računara.
7. Objasniti kako funkcionišu sledeći delovi *HTML* dokumenta:
 - ``
 - ``
8. Koristeći editor teksta kreirati *HTML* dokumenat kojim se formira tabela.

³U asinhronom prenosu se u svaki informacioni bajt dodaju start i stop bit. Pri brzini preposa od 1 200 b/s biće prepeto ukupno

$$\frac{1200}{8+2} = 120 \text{ b/s} = 15 \text{ b/s}$$

što znači da prepos jednog bajta traje $1/15 \approx 0.067$ sekunde. Slanje 86 bajtova u dva smera traje $t_1 = 86 \times 0.067 \times 2 = 11.524$ s. Vreme preposa (propagacije) iznosi

$$t_2 = 2 \frac{2000}{200000} = 20 \text{ ms}$$

jer je brzina preposa signala kroz fizički vod oko 200 000 km/s. Ukupno vreme je $t = t_1 + t_2 = 11.544$ s.

```

< TABLE BORDER = 1 >
< TR>< TD ROWSPAN = 3 >SLOVA< /TD>      < TD>A</TD>< /TR>
< TR>                                       < TP>B</TD>< /TR>
< TR>                                       < TV>C</TD>< /TR>
< / TABLE >

```

Oznaka `< ROWSPAN >`, primenjena na reč `SLOVA` formira na levoj strani tabele stubac koji se poklapa sa sva tri reda koja sadrže A, B i C. Atribut `< BORDER >` unutar oznake `< TABLE >` nalaže čitaču da iscrta okvire oko tabele i njenih ćelija. Vrednost 1, koja je dodeljena atributu `< BORDER >`, označava debljinu linija od jedne tačke.

9. Kreirati sopstvenu Veb stranicu (Home Page). Za izradu stranice koristiti editor teksta Notepad ili neko od editor za oblikovanje *HTML*. Stranica Microsoft FrontPage. Stranica treba da sadrži:
 - zaglavlje (Head) i telo (Body),
 - naslov (Title) koji sadrži ime i prezime korisnika,
 - naslov na stranici (H2) koji glasi "Ime prezime-moja lična stranica",
 - barem dva poglavlja o korisniku (biografski podaci, interesovanja u vezi sa studijama, hobi i slično),
 - poglavlja treba da imaju barem dve hiperveze,
 - hipervezu za slanje e-mail-a (mailto) na adresu korisnika,
 - barem jednu sliku (fotografiju korisnika ili nešto drugo vezano za sadržinu strane),
 - strukturirati tabelu 3×1 u sredini stranice.
10. Kreiranu WWW stranicu instalirati na lokalni računar. Za pregled i testiranje WWW stranice koristiti Internet Explorer. Ispraviti greške.

Literatura

- [1] T. Aleksić, Računari: *arhitektura i organizacija*. Bograd: Naučna knjiga, 1982.
- [2] W. R. Ashby, *An Introduction to Cybernetics*. London: Champan & Hall, 1956.
- [3] N. Balaban i dr., *Principi Informatike*. Beograd: Savremena administracija, 1996.
- [4] D. Branković i D. Mandić, *Metodika informatičkog obrazovanja – sa osnovama informatike*. Banja Luka: Filozofski fakultet u Banja Luci, 2003.
- [5] S. Brooks and B Byles, "Helping Teacher Use, The Internet Efficiency," 2004. <http://www.internet4classroom.com/excelx-functions.htm>.
- [6] R. Conlon and L. Jonson. „The Land-Grant Training Alliamce – On Line PowerPoint2000 Lesson,“ 2001. <http://www.lgta.org/ppt2000.contents.htm>.
- [7] D. E. Comer, *Povezivanje mreža TCP/IP – Principi, protokoli arhitekture*. Beograd: *Computer Equipmunt and trade*, 2001..
- [8] M. Vujaklija, *Leksikon stranih reči i izraza*. Beograd: Prosveta, 1976.
- [9] B. Grbić, *Sistem Mikrosoft Office 2003 na srpskom*. BEograd: PC-Press, 2004.
- [10] S. Đorđević-Kajan and I. Popović, UNIX - sistemski prilaz. Niš: Elektronski fakultet u Nišu, 1996.
- [11] S. James, HTML 4 za neupu'ene. Beograd: Mikro knjiga, 1998
- [12] E. Kajan, Information Technology encyclopedia and acronymos. Berlin Heidelberg: Springer-Verlag, 2002
- [13] R. Jovanović, Ž. Tošić i M.Stanković, Informatika za ekonomiste. Univeryitet u Nišu, Ekonomski fakultet, 1999.
- [14] G. Lukatela, Statistička teorija telekomunikacija, Beograd: Građevinska knjiga 1981
- [15] N. Milenković, Arhitektura i organizacija računara. Niš Elektronski fakultet, 2004.
- [16] S. Minić, Optimalni prijemnik signala koji se prenose energetskim vodovima. PHD thesys, Tehnički fakultet u Čačku, 2004.

- [17] S. Minić, Osnovi informatike i računarstva, Učiteljski fakultet, Univerzitet u K.Mitrovici, Leposavić, 2005.
- [18] S. Minić, Osnovi informatike i računarstva, Učiteljski fakultet, Univerzitet u K.Mitrovici, Leposavić, 2007.
- [18] S. Minić, Programski jezik FORTRAN, Mašinski fakultet, Univerzitet u Beogradu, Beograd, 2009.
- [19] L. Magid, Mali priručnik za Windows XP, Beograd, Mikro knjiga, 2004.
- [20] B. Nastić, Osnovi televizijske tehnike. Beograd, Naučna knjiga, 1986.
- [21] S. Obradović, osnovi računarske tehnike, Beograd: Viša elektrotehnička škola, 1996.
- [22] R.Pearlman, Interconnections - bridges, routers, switches and inter-working. London: Addison-Wesley, second ed.,2000.
- [23] R. Petersen and I. Haddad, Red Hat enterprise linux Fedora Edition – Kompletni priručnik. Beograd: CET Computer Equipment and Trade, 2004.
- [24] D. Živković i M. Popović, Impulsna i digitalna elektronika. Beograd: Akademska Misao, 2004.
- [25] A. Popović i Z. Budimac, Uvod u operativne sisteme Windows. Novi Sad: Prirodno matematički fakultet - Institut za matematiku, 1995.
- [26] D. Radovanović, 'draGanova THML Školica' 2004. <http://avala.de>
- [27] P. Rančić, Prilozi svetlotehničkim karakterizacijama. Niš: Elektronski fakultet, 1997.
- [28] D. Soleša i Đ. Nadrljanski, Informatičke tehnologije. Sombor: Učiteljski Fakultet u Somboru, 2004.
- [29] V. Stojanović, Diskretne mreže i procesiranje signala. Niš: elektronski fakultet, 2004
- [30] M. Stojčev i B. Petrović, Arhitekture i programiranje mikroracunarskih sistema zasnovanih na familiji procesora 80x86. Niš: Elektronski fakultet u Nišu, 1999
- [31] A. Tanenbaum, Computer network, Engelwood Cliffs, New Jersey: Prentice-Hall, Inc, 1996.
- [32] Ž. Tošić, Osnovi računarske tehnike. Niš: Čuperak plavi, 1994.

- [33] Z. Urošević, Uvod u računarske telekomunikacije i mreže. Čačak: Tehnički fakultet u Čačku, 2004.
- [34] V.C. Hamacher, Z.G. Vranesic, and S.G. Zaky, Computer Architecture and organization. New York: McGraww-Hill Book Company, 1978
- [35] K. Hwang and F.A. Briggs, Computer architecture and parallel processing. New York: McGraww-Hill Book Company, 1984
- [36] K.Hwang, Computer arithmetic: Principles, architecture, and design. New York: John Wiley, 1979
- [37] E. Willet, Office 2003 Biblija, Beograd: Mikro knjiga, 2004.

Indeks pojmov

- Abak, 66
- Aditivno mešanje boja, 139
- Adresa
 - IP, 225
 - numerička, 225
 - simbolička, 225
- Adresa memorijske lokacije, 105
- Adresa URL, 235
- Adresni registar memorije,
 - 88, 91
- Agregacija, 71
- Akumulator, 84, 92
- Akumulatorska arhitektura,
 - 84
- Algoritam oduzimanja, 49
- Algoritam sa sabiranjem, 48
- Analiza
 - leksička, 173
 - semantička, 174
 - Aplikacioni softver, 149
- Arhitektura
 - akumulatorska, 84
 - sa registrima opšte namene,
 - 84
 - stek, 84
 - Arhitektura procesora, 83
- Arhitektura sa registrima opšte namene, 85
- Arhiviranje, 209
 - Aritmetičko-logička jedinica, 32, 75, 76
 - Audiotekst, 256
 - Automat, 65
- Automacka kontrola, 18

- Bajt, 28
- Binarne vrednosti, 107
- Binarni kod, 45
- Binarno-decimalni kod, 45
- Biotika, 70
- Bit, 28

- Bitska brzina, 219
- Brojač, 160
- Brojač instrukcija, 88, 89, 91
- Ćelija, 333
- Celobrojni podaci, 49
- Centralna jedinica, 32
- Centralni procesor, 75
- Čarobnjak, 346
- Čvrst disk, 32
- Cilindar diskete, 117
- Datoteka
- Datoteka FIFO, 182
- Datoteke katalogi, 182
- Dekodiranje, 45
- Digitalni prenos ISDN, 254
- Dijalog
- Direktan pristup memoriji,
 - 105
- Disketa, 116
- Dostavljanje podataka, 21
- Društvo
 - industrijsko, 17
 - informatičko, 17
- Dugme
- Dvoadresna instrukcija, 86
- Dvostruka reč, 49
- Editor, 37
 - WISIWYG, 236
 - Editori, 174
 - Elektronska cev, 18
- Elektronska konferencija
 - WAIS, 229
 - WWW, 229
 - gopher, 229
- Elektronska pošta, 224
- Elektronska razmena podataka, 256
- Elektronski računar
 - EMGAS, 67
 - Element slike, 134
 - Entiteti protokola Interneta, 225
 - Entropija, 22, 26
 - Era automatizacije,
- 18
- Flopi disk, 116

Fon Nojman, 30
Format
 nepokretne tačke, 51
 pokretne tačke, 57
Format instrukcija
 dvoadresni, 81
 dvoadresni, 81
 jednoadresnai, 82
 jednoadresni, 82
 nuladresni, 83
 troadresni, 80
 troadresni, 80
 Format slike
 GIF, 242
Formatizovanje diskete, 117
Formula, 337
Funkcija, 337

Gigabajt, 28
Grafički interfejs, 194

Hafmenov kod, 47
Hardver
 za izlaz, 31
 za memorisanje, 31
 za obradu, 31
 za unos, 31
Hifenacija, 292
Hijerarhija naslova, 267
Hiperlink, 233
Hipertekst, 233
 Horizontalni lenjir, 265
Host računar, 223

Ikona, 198
Indikator, 198
 aktivnog jezika, 264
 čekanja za štampu, 264
provere ispravnosti kucanja, 264
Instrukcija, 94
 ADD, 84
 IN 95
 LD, 84
 LOAD, 95
 MOVE, 95
 OUT, 95
 POP, 95
 USH, 95
 STORE, 95
 ST, 84
Instrukcije
 aritmetičke, 95
 logičke, 95
 obrade, 94
 ostale, 94
 ulazno-izlazne, 94
 upravljačke, 94
 za pomeranje, 95
 za prenos podataka, 94
Instrukcije za
 grananje i preskok, 95
 proceduru, 95
 programske petlje, 95
Integrisana kola
 VLSI, 18
 LSI, 18
Interaktivni razgovori, 230
Internet, 223
Interpretatori, 174
Interpreter, 36
Inverzna poljska notacija, 87
Isključivo ili operacija, 47
Izlazna jedinica, 74
Izmenljivi disk, 32
Izračunavanje, 71
Izraz, 337
Izvršenje instrukcije, 96
Java apleti, 242
Jedinice računara, 72
Jednoadresni procesor, 84
Jednostruki prored, 277
Jezgro operativnog sistema,
 155

Kadrovi, 37
Kanal, 218
Kapacitet kanala, 219
Kapacitet memorije, 106
Katodna cev, 134
Keš memorija, 78
Keš-memorija, 90, 113
Kičmene mreža, 224
Kilobajt, 28 Klasifikacija, 71
Klijent-server model, 227
Klipbord, 211 Klizač, 198

- Kod
 - ASCII, 61
 - ISO 7, 61
 - dva od pet, 46
 - Grejov, 46
 - Hafmenov, 46
 - viška tri, 46
 - Kod operacije, 93
- Kodiranje, 45
- Kodna strana, 62
- Kolekcija
 - Količina informacija
 - jedinična, 28
 - maksimalna, 28
 - srednja, 28
- Kolona, 333
 - Kolor prostor
 - HSL, 139
 - RGB, 138
- Kolor rezolucije, 139
- Komanda Linux sistema
 - less, 208
 - is, 208
 - more, 208
 - tar, 209
 - zip, 210
 - Kompajler, 36
- Kompilatori, 173
- Komplement broja
 - nepotpun, 50
 - potpun, 50
- Komplementarne boje, 147
- Kompresija, 210
 - Komunikacioni softver, 151
- Kontroler diskova, 101
- Kristalni oscilator, 160
- Laser, 33
- Laserski štampač, 146
- Linearna regresija, 35
 - Linker, 37
 - Lista
- Logički tip podataka, 63
- Lokalna mreža, 32
- Louder, 37
- Magacinska memorija, 79
- Magistrala
 - ISA, 101
 - PCI, 101
 - VESA, 101
 - spoljašnja, 88
 - spoljna, 89
 - unutrašnja, 88, 89
- Mantisa, 57
- Mašina, 65
- Mašinske operacije, 93
- Megabajt, 28
- Memorija sa neposrednim pristupom, 110
- Memorija sa proizvoljnim pristupom, 109
- Memorija sa roizvoljnim pristupom, 115
- Memorija sa slučajnim pristupom, 105
- Memorijsaka lokacija, 05
- Memorijski medijum, 22, 107
- Meni, 198
- Mikroinstrukcija, 94
- Mikrooperacije, 94
- Mini računari, 32
- Miš, 31
- Modeli interfejsa, 197
- Modem, 32, 214, 226
- Modifikator adrese, 93
- Monitor, 134
- Most, 224
- Mreža najnižeg nivoa, 224
- Mreža srednjeg nivoa, 224
- Mrežna alatka
 - finger, 258
 - host, 258
 - ping, 258
 - tracerute, 258
- Mrežni klijent
 - ICQ, 258
 - IRC, 258
 - Talk, 258
- Mrežni prolaz, 223, 224
- Mrežni završetak, 254
- Multiprogramiranje, 166
- Nestalne memorije, 106
- Neuroračunari, 68
- Nivo sivog, 139

Nojmanova arhitektura, 67
 Normalni prikaz, 267
 Nula adresni procesor, 85
 Obične datoteke, 182
 Obrada podataka, 21
 Odeljak, 275
 Okruženje
 GNOME, 206
 KDE, 206
 Opcija
 Operativna memorija, 32, 76,
 108, 115
 Operativni sistem, 34, 152
 DOS, 153
 OS/2, 153
 UNIX, 153
 WINDOWS, 153
 Windows 3.0, 186
 Windows 95, 186
 Windows NT, 186
 Windows, 185
 jednokorisnički, 35
 jednoprocesni, 35
 komandni jezik, 156
 planiranje, 155
 upravljanje podacima, 155
 uređajima, 155
 višekorisnički, 35
 višeproceni, 35
 Optički disk, 121
 Organizacija podataka, 17
 Osnovna ploča, 100
 Otvaranje UNIX datoteke, 183
 Oznaka domena FQDN, 226
 Oznaka HTML kategorije
 < A > sidro, 239
 < ADDRESS > adresa, 239
 < ALT > alternativni
 tekst, 238
 < BGCOLOR > boja pozadi-
 ne, 239
 < BODY > telo, 238
 < BR > novi red, 239
 < HEAD > zaglavlje, 238
 < HTML >, 238
 < H* > nivo naslova, 240
 < ISMAP > mapa osetljiva
 na dodir, 238
 < P > pasus, 239
 < SRC > izbor slike, 238
 Paketna obrada, 163
 Pamćenje podataka, 21
 Paralelni prenos podataka,
 219
 Peta generacija sioftvera,
 177
 Petabajt, 28
 Pisanje u Unix datoteku, 183
 Podaci, 37
 Pokazivač steka, 83, 85
 Polarizacija, 57
 Polje za prikaz, 198
 Poluprovodnički laser, 121
 Polureč, 49
 Pouzdanost memorije, 106
 Povezivači, 175
 Prekid, 96
 Prekidačka algebra, 95
 Prenos podataka, 229
 Pretraživanje, 71
 Pretraživanje Unix datoteke, 183
 Prihvatni registar memorije, 91
 Prikazi, 266
 Priključak USB,
 101
 Primarni
 vetlosni izvori,
 136
 Priprema instrukcije, 96
 Prirodni BCD kod, 46
 Procedure, 38
 Proces analize, 173
 sinteze, 174
 Procesor, 75, 79
 Program, 65, 94
 FrontPage, 250
 Minix, 204
 Programiranje, 17
 Programski brojač, 80
 Programski jezik, 38
 BASIC, 174
 HTML, 231

Programski prevodioci, 172
 Propusnost memorije, 106
 Prošireni format pokretne
 tačke, 58
 Protokol
 SLIP, 226
 FTP, 226, 229
 HTTP, 231
 NNTP, 230
 PPP, 226
 SMTP, 229
 TCP/IP, 223
 TELNET, 231
 Usenet, 230
 Protokol Interneta, 225
 Prozor programa, 264
 Prugasti kod, 131

 Računar
 ENIAC, 67
 Računarska mreža
 LAN, 32
 WAN, 215
 Računarska tehnika, 17
 Računarski sistemi
 arhitektura, 17
 organizacija, 17
 Ram memorija, 78
 Reč, 49
 Redudansa, 28
 Registar, 90
 opšte namene, 88, 90
 upravljački, 90
 Registar instrukcija, 88, 92
 registar memorije, 88
 Registar podataka, 161
 registarska memorija, 113
 Registri opšte namene, 92
 Režim dijaloga, 168 Rom
 memorija, 78
 Ruter, 224

 Sabirnica, 89
 Sektor
 sa podacima, 118
 sa podacima o rezervisanom
 prostoru za datoteke, 118
 za kataloge, 118
 za pokretanje računara,
 118
 Sektor diskete, 117
 Serijski prenos podataka, 219
 Server
 WAIS, 230
 Servis
 FTP, 226
 Sistem domena, 225
 Sistem za pretraživanje, 257
 Sistemski softver, 150
 Sjajnost boje, 141
 Skener, 32
 Skrivena jedinica, 57
 Skup instrukcija, 94
 Služba kataloga, 161
 Softver CASE, 177
 Sortiranje, 71,
 Specifična cena, 106 Specijalne
 datoteke, 182
 Spoljna memorija, 108
 Sprežna kola, 90
 Stalne memorije, 106
 Standard
 RDIFAST, 256
 EFT, 257
 Standard ISO, 235
 Start bit, 220
 Statusna traka, 264
 Staze diskete, 117 Stek
 arhitektura, 84
 Stek memorija, 79
 Stek-memorija, 90
 Stop bit, 220
 Supervizor, 158
 Svetlosna olovka, 74

 Taktni signali, 90
 Tastaura, 127
 Taster, 198

- precrtani, 279
- skriveni, 279 T
- ekst procesor, 263
- Telekomunikacije, 17
- Telekomunikaciona mreža
 - ATM, 254
- Telematika, 69
- Teletekst, 255
- Teorija algoritama, 17
- Terabajt, 28
- Težinski kbd
 - BCD, 46
 - Topologija računarske mreže magistrala, 217
 - prsten, 217
 - zvezda, 217
- Transakcioni režim, 168
- Troadresni procesor, 86
- Tvrđi magnetni disk, 118

- Ulazna jedinica, 73
- Ulazno-izlazna ploča, 101
- Umetanje teksta, 269
- Univerzalni režim, 168
- Upravljačka jedinica, 75, 76, 88
- Upravljački panel, 198
- Upravljački signali, 90
- Upravljanje
 - datotekama, 161
 - periferijama, 161

- Upravljanje poslovima, 156
- Uređivač jednačina, 293
- Uslužni modul
 - mcopy, 208
 - Uslužni softver, 36
- Veb alat
 - Flasf Mx, 236
 - FrontPage, 236
- Veb čitač, 233, 235
 - Internet Explorer, 235
 - Mosaic, 231
 - Notscape, 235
 - TkWWW, 235
 - Veb servis, 231
 - Veliki računari, 32
 - Vertikalni lenjir, 265
- Veštačka inteligencija, 17, 70
- Video adapter, 101
- Video zapis, 312
- Videotekst, 256
- Višekorisnički sistem, 203
- Vpristupa memoriji, 106
- Vreme ciklusa memorije, 106
- Vrsta boje, 140

- Zaglavljje
 - Zasićenje boje, 141
 - Zatvaranje Unix datoteke, 183
- Znakovni tip podataka, 61

CIP - Каталогизација у публикацији
Народна и универзитетска библиотека
Републике Српске, Бања Лука

004(075.8)

МИНИЋ, Синиша
Informacione tehnologije / Siniša G. Minić. -
Brčko : Evropski Univerzitet Brčko distrikta, 2012
(Banja Luka : Markos). - 266 str. : ilustr. ; 25
cm

Tiraž 200. - Napomene uz tekst. - Bibliografija:
str. 257-259. - Registar.

ISBN 978-99955-750-6-9

COBISS.BH-ID 3390232