

---

Bojan Jerbić

# Osnove računarstva

samo za internu upotrebu kolegija INFORMATIKA

---

Zagreb, 2005.



PROF. DR. SC. BOJAN JERBIĆ

# Osnove računarstva

SAMO ZA INTERNU UPOTREBU (BEZ RECENZIJE I LEKTURE)

---

© BOJAN JERBIĆ

# PREDGOVOR

Računarstvo je iznimno znanstveno područje koje snažno utječe na sve ljudske djelatnosti i cjelokupni društveni razvoj. Upliv suvremenih informatičkih tehnologija osjećamo jednako u profesionalnom kao i u privatnom životu. Računala su danas prisutna u kućanskim aparatima, u proizvodnim strojevima, osobnim vozilima, mobitelima, naprosto svugdje. Poznavanje i razumijevanje računalne tehnologije temeljno je pitanje uključivanja u suvremeno informatičko doba. Stoga odgovarajuće informatičko obrazovanje treba stjecati od najranije dobi i postupno ga nadograđivati tijekom cijelog procesa školovanja, ali i kasnije kroz profesionalno djelovanje. Ovaj je priručnik nastao u nastojanju da se doprinese obrazovanju u području računalne primjene, prvenstveno studenata Fakulteta strojarstva i brodogradnje, ali i studenata drugih visokoškolskih institucija. Priručnik je rezultat iskustava u znanstvenom i nastavnom radu autora u području primjene računarskih znanosti na Fakultetu strojarstva i brodogradnje. Konceptija priručnika oslanja se na spoznaju da je razumijevanje računalne tehnologije osnova za samostalnu i kreativnu primjenu računala, a ne tehnika korištenja pojedinih korisničkih programa. Stoga je priručnik organiziran u šest cjelina koje obrađuju računarstvo od svojih početaka i temeljnih principa, postupno objašnjavajući znanja o građi računala, operativnim sustavima i računalnim mrežama.

U prvome poglavlju kratko je opisana povijest računarstva s namjerom da se istaknu temeljne značajke informatičke tehnologije i pravci razvoja. U potrazi za polaznim oblicima računarstva moguće je jasnije shvatiti suštinu ove tehnologije, upotpuniti ukupnu sliku i obogatiti osobnu tehničku kulturu.

Drugo poglavlje obrađuje osnovne principe računalne tehnologije, ističući model algoritamskog stroja i binarnu logiku kao temeljne aspekte računarstva.

Način implementacije opisanih principa dan je u trećem poglavlju koje otkriva tehničke značajke računala. Opisana je struktura računala i princip rada pojedinih komponenata, od memorije, preko procesora do vanjskih uređaja.

Uloga i zadaće operativnog sustava prikazani su u četvrtom poglavlju. Posebno je obrađen Windows NT operativni sustav, odnosno njegova organizacija i glavne funkcije.

U petom poglavlju otkrivaju se tajne računalnih mreža, uključujući fizičke značajke, kao i protokole na različitim komunikacijskim razinama.

U zadnjem šestom poglavlju dan je osvrt na neke razvojne značajke računarstva, uključujući mikroprocesorsku tehnologiju, paralelizaciju, simulaciju, računalne mreže, normizaciju i umjetnu inteligenciju.

# SADRŽAJ

<b>PREDGOVOR</b>	<b>II</b>
------------------	-----------

---

<b>SADRŽAJ</b>	<b>III</b>
----------------	------------

---

<b>1 UVOD</b>	<b>1</b>
---------------	----------

---

1.1 POVIJEST RAČUNARSTVA	3
1.2 SUVREMENE INFORMACIJSKE TEHNOLOGIJE	12

<b>2 OSNOVNI PRINCIPI RAČUNALNE TEHNOLOGIJE</b>	<b>15</b>
---	-----------

---

2.1 TURINGOV STROJ	17
2.2 BINARNI BROJEVNI SUSTAV	21
2.2.1 PRIKAZ NEGATIVNIH CJELOBROJNIH VRIJEDNOSTI	22
2.2.1.1 Ekscesivna notacija	22
2.2.1.2 Komplementarna notacija	23
2.2.2 PRIKAZ RACIONALNIH BROJEVA	23
2.3 PRIKAZ SIMBOLIČKIH PODATAKA	26
2.4 ARITMETIČKO/LOGIČKE OPERACIJE	30
2.4.1 BINARNA ARITMETIKA	30
2.4.2 BINARNE LOGIČKE OPERACIJE	32

<b>3 GRAĐA RAČUNALA</b>	<b>41</b>
-------------------------	-----------

---

3.1 SABIRNICA	43
3.2 MEMORIJA	46
3.2.1 RADNA MEMORIJA	46
3.2.2 STALNA MEMORIJA	48
3.2.3 VANJSKA MEMORIJA	49
3.2.3.1 Memorijski diskovi	49
3.2.3.2 Traka	55
3.3 PROCESOR	57
3.3.1 SPREMNIK	58

3.3.2	SPREMNIK NAREDABA	59
3.3.3	ARITMETIČKO/LOGIČKA JEDINKA	59
3.3.4	UPRAVLJAČKA JEDINKA	59
3.3.5	PROCESORSKE ARHITEKTURE	59
3.3.5.1	CISC i RISC arhitektura	63
3.3.6	KAKO RADI PROCESOR	65
3.3.6.1	Strojni jezik	65
3.3.6.2	Primjer	67
<b>3.4</b>	<b>ULAZNO/IZLAZNI UREĐAJI</b>	<b>73</b>
3.4.1	TIPKOVNICA	73
3.4.2	Miš	73
3.4.3	ZASLON	74
3.4.3.1	Katodna cijev	74
3.4.3.2	Tekući kristal	75
3.4.4	PISAČI	75

## **4 OPERATIVNI SUSTAV** **79**

---

<b>4.1</b>	<b>ŠTO JE OPERATIVNI SUSTAV</b>	<b>81</b>
<b>4.2</b>	<b>STRUKTURA I ZADAĆE OPERATIVNOG SUSTAVA</b>	<b>83</b>
<b>4.3</b>	<b>MS WINOWS OPERATIVNI SUSTAVI</b>	<b>87</b>
4.3.1	KONCEPT I ORGANIZACIJA NT-A	88
4.3.1.1	FAT	88
4.3.1.2	NTFS	90
4.3.2	NT JEZGRA	92
4.3.3	HAL	92
4.3.4	IZVRŠNE KOMPONENTE	93
4.3.4.1	Upravljanje objektima	93
4.3.4.2	Nadziranje sigurnosnog pristupa	93
4.3.4.3	Upravljanje procesima	94
4.3.4.4	Upravljanje virtualnom memorijom	94
4.3.4.5	Lokalni proceduralni pozivi	94
4.3.4.6	Upravljanje ulazno/izlaznim procesima	94
4.3.5	PODSUSTAV OKRUŽENJA	95
4.3.6	POKRETANJE NT-A	96
4.3.7	RADNI STOL GRAFIČKOG SUČELJA	98
4.3.7.1	Ikone/simboli	98
4.3.7.2	Izbornik "Start"	103
4.3.7.3	Linija zadaća	105
4.3.8	RAD S IMENICIMA I ZAPISNICIMA	105

## **5 RAČUNALNE MREŽE** **113**

---

<b>5.1</b>	<b>OSI REFERENTNI MODEL</b>	<b>117</b>
<b>5.2</b>	<b>TOPOLOGIJA RAČUNALNIH MREŽA</b>	<b>121</b>
5.2.1	ETHERNET	124
5.2.1.1	Komunikacijski protokol	124
5.2.1.2	Fizičke značajke	125
5.2.2	TOKEN RING	128
<b>5.3</b>	<b>TCP/IP PROTOKOL</b>	<b>131</b>
5.3.1	DoD MREŽNI MODEL	131

5.3.2	TCP/IP FORMAT PODATAKA	132
5.3.3	PORODICA TCP/IP PROTOKOLA	133
5.3.4	INTERNET ADRESE	135
5.3.5	INFORMACIJSKE MREŽNE USLUGE	137
<b>5.4</b>	<b>HTTP</b>	<b>144</b>
<b>5.5</b>	<b>NETBEUI/NETBIOS</b>	<b>148</b>
<b>5.6</b>	<b>IPX/SPX PROTOKOL</b>	<b>149</b>
5.6.1	IPX	149
5.6.2	SPX	149
<b>5.7</b>	<b>UNIX U MREŽI</b>	<b>151</b>
5.7.1	IME RAČUNALA	151
5.7.2	PROVJERA	152
5.7.3	PRIJAVA NA UDALJENO RAČUNALO	153
5.7.4	PRIJENOS ZAPISNIKA	156
5.7.5	ELEKTRONIČKA POŠTA	161
5.7.6	OSTALE MREŽNE USLUGE	167
5.7.7	MREŽNI ZAPISNIČKI SUSTAV - NFS	169
<b>5.8</b>	<b>WINDOWS NT U MREŽI</b>	<b>173</b>
5.8.1	PRISTUP RAČUNALIMA U MREŽI	173
5.8.2	"MAP NETWORK DRIVE"	176
5.8.3	PRISTUP MREŽNIM PISAČIMA	178
5.8.4	PODJELA OBJEKATA	179
<b>6</b>	<b>RAZVOJNE TEŽNJE</b>	<b>181</b>
<b>6.1</b>	<b>MIKROPROCESORI BUDUĆNOSTI</b>	<b>183</b>
<b>6.2</b>	<b>PARALELIZACIJA I SUPER-RAČUNALA</b>	<b>185</b>
<b>6.3</b>	<b>SIMULACIJA I VIZUALNO RAČUNARSTVO</b>	<b>187</b>
<b>6.4</b>	<b>RAČUNALNE MREŽE</b>	<b>191</b>
<b>6.5</b>	<b>NORMIZACIJA</b>	<b>193</b>
<b>6.6</b>	<b>UMJETNA INTELIGENCIJA</b>	<b>196</b>
<b>7</b>	<b>PRIMJENA</b>	<b>199</b>
<b>7.1</b>	<b>MS WORD</b>	<b>200</b>
7.1.1	PRIJE POČETKA	200
7.1.1.1	Pokretanje MS Worda	201
7.1.1.2	Spremanje dokumenta prvi puta:	201
7.1.1.3	Spremanje dokumenta s drugim imenom:	202
7.1.1.4	Otvaranje postojećeg dokumenta	203
7.1.2	VRSTE POGLEDA I NAČIN UPOTREBE	204
7.1.3	MIJENJANJE I PRILAGOĐAVANJE DOKUMENATA	205
7.1.4	OSNOVNO O PROMJENAMA TEKSTA	205
7.1.4.1	Korištenje kratica pri unosu teksta (AutoCorrect i Autotext naredbe)	207
7.1.4.2	Pronalaženje i zamjena određene riječi u tekstu	208
7.1.4.3	Ispravljenje gramatičkih pogrešaka	208
7.1.5	UREĐIVANJE IZGLEDA TEKSTA	209
7.1.6	UREĐIVANJE TEKSTA POMOĆU STILOVA	209
7.1.6.1	Pregledavanje i korištenje stilova	210
7.1.6.2	Primjena stila na odlomak teksta	210
7.1.6.3	Promjena postojećeg i kreiranje novog stila	211

7.1.6.4	Kreiranje naslova s više razina	212
7.1.7	KREIRANJE TABLICE SADRŽAJA	213
7.1.8	UNOŠENJE SLIKA	214
7.1.9	KREIRANJE ZAGLAVLJA I PODNOŽJA DOKUMENTA	214
7.1.9.1	Uređivanje različitog zaglavlja prve stranice dokumenta	215
7.1.9.2	Kreiranje zaglavlja i podnožja za parne i neparne stranice dokumenta	216
7.1.10	KREIRANJE UNAKRSNIH SPONA MEĐU ELEMENTIMA DOKUMENTA	216
7.1.11	DEFINIRANJE PRIJELOMA STRANICA	217
7.1.12	FORMATIRANJE STRANICA ZA ISPIS DOKUMENTA	217

---

**8 LITERATURA** **219**



---

# 1 UVOD

---

**R**ačunarstvo je znanstvena disciplina koja se isprepleće kroz brojne ljudske djelatnosti, ali, unatoč nazivu, većina korisnika stječe dojam da najmanje služi računanju.

Računarstvo se bavi gradnjom računala, računalnim programiranjem i obradom informacija. Računalo je u osnovi sekvencijalni (slijedni) uređaj koji je sposoban interpretirati simbolički opisane algoritme u obliku programa i u skladu s njima obrađivati informacije.



U računalu se programi i informacije prikazuju brojevima, štoviše nulama i jedinicama. Složene zadatke obavljaju se kombiniranjem osnovnih aritmetičko-logičkih operacija. Svaku informaciju, bilo diskretnu ili kontinuiranu, slikovnu ili zvučnu, moguće je predstaviti brojkama (numerički, digitalno), a to znači da ju računalo može obraditi. S obzirom da informacije mogu biti podaci o našim bankovnim računima, zdravstvenom osiguranju, parametrima diferencijalne jednačbe gibanja, stanju senzora pritiska ulja u motoru automobila, transportu papira u pisaču, položaju oštrice alata numerički upravljana stroja, i mnogo čemu drugom čijem bi nabravanju teško sagledali kraj, jasno je da računalo nameće svoju primjenu u svakodnevnom životu i svemu onome što nas okružuje. Računalna je primjena često prikrivena i u potpunosti automatizirana u različitim uređajima, a u bliskoj budućnosti možemo očekivati sve veću prisutnost računala, jednako u vrlo složenim, kao i u posve banalnim primjenama.

Računalo je ipak izrazito složen uređaj čije prave mogućnosti dolaze do izražaja u rješavanju posebno zahtjevnih zadataka, od složenih matematičkih izračuna, numeričke i grafičke simulacije procesa, geometrijskog oblikovanja, obrade velikih baza podataka, upravljanja robotskim sustavima do grafičkog oblikovanja teksta. Tada se nužno susrećemo s računalom u interakciji koja zahtjeva određeno znanje, ne samo o problematici kojom se bavimo, već i znanje o računalnom programu kojim se služimo, operacijskom sustavu koji nam koristi da upravljamo njime, ali i znanje o računalu samom, njegovoj tehnologiji i principima, koje će nam pomoći da ga kompetentno koristimo u svoj njegovoj tehničkoj raskoši. Ova knjiga ima za cilj doprinijeti razumijevanju računarstva kao znanosti, ali i kao praktične vještine. Računarstvo nije samo programiranje i procesiranje, ali nije niti besciljno pritiskanje miša po izbornicima u potrazi za slučajnim rješenjima. Knjiga nastoji objediniti različita znanja potrebna svakom obrazovanom korisniku da s razumijevanjem upotrijebi i unaprijedi mogućnosti koje računalna tehnologija nudi. Namijenjena je ponajprije studentima tehničkih fakulteta kao podloga za primjenu računala u struci i općim zadacima.



## 1.1 Povijest računarstva

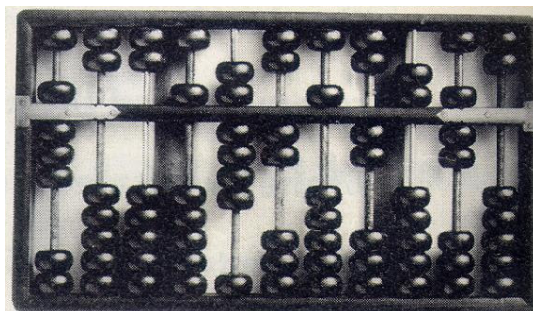
Postoje različite interpretacije o povijesti ljudskog stvaralaštva koje bi se moglo povezati s razvojem računarstva.

U potrazi za korijenima stvaranja informatičkog društva valja poći od glavnih značajki računala. One se mogu sažeti na sposobnosti pohrane podataka, izvođenja algoritamskih zadaća (računanja) i komunikacije.



Te se značajke, barem djelomično, daju prepoznati u različitim drevnim artefaktima koji su služili za pohranu podataka. Najstariji poznati pisani dokument, koji potječe još od starih Sumerana, datiran između 4000. i 1200. godine s.e., načinjen je na glinenoj pločici, a sadrži podatke o trgovačkim transakcijama. Pohrana podataka može se smatrati prvim korakom u razvoju informatičkog društva.

Računaljke s drvenim kuglicama ("Abacus", Sl. 1-1), kakve su koristili Babilonci prije 5000 godina, te kasnije Kinezi, zatim Grci i Rimljani, omogućavale su pohranu podataka kodirajući ih položajem i bojom kuglica, dok su se njihovim propisanim premještanjem obavljale računske operacije (procesiranje). S obzirom da je algoritam računanja izvodio čovjek, drvenim računaljkama se može priznati tek sposobnost srede pohrane podataka.

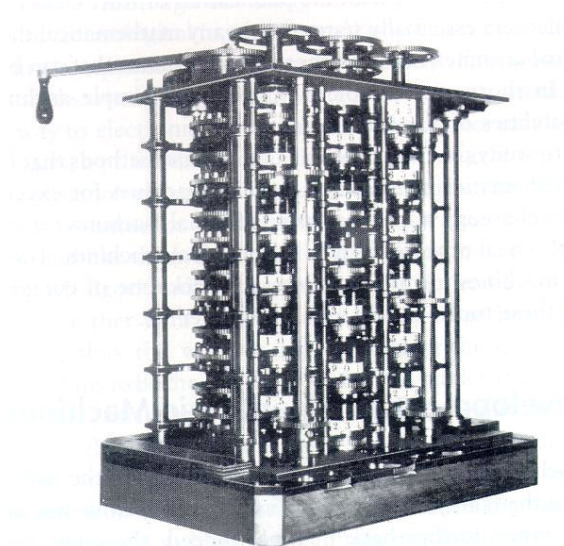


Sl. 1-1 Abacus

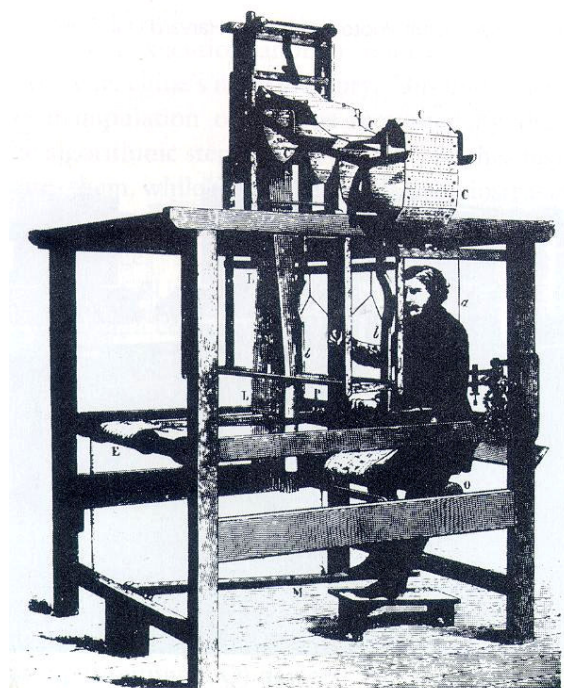
Prvi uređaji sa sposobnošću samostalnog izvođenja algoritama računanja pojavljuju se u 17. stoljeću u obliku automata temeljenih na sustavu zupčanika (William Shickard, "računarski sat", 1623, Blaise Pascal, "Pascalene", Francuska 1623-1662, Samuel Morland, Engleska, 1666, Gottfried Wilhelm Leibniz, "Stupnjevano računalo", Njemačka, 1646-1716). Algoritam računanja je u takvim automatima bio pohranjen u konfiguraciji sustava zupčanika i njihovih odnosa. Taj se princip zadržao sve do nedavno kao osnova za konstrukciju stolnih mehaničkih računalnih strojeva.

U 19. stoljeću Charles Babbage (Engleska, 1792-1871, Sl. 1-2) koristi ideju Josepha Jacquarda (Francuska, 1801, Sl. 1-3), koji je konstruirao tkalački stroj čijim se procesom tkanja, odnosno uzorkom tkanja, upravlja pomoću rupica izbušenih na papirnatim karticama (Jacquardov uzorak). Babbage je isti princip iskoristio pri izradi računalnog stroja koji je mogao izvoditi računalne operacije na temelju podataka zadanih u obliku bušenih kartica ("Diferencijalni stroj"). One se koriste za unos podataka u računalo sve do novijeg doba, odnosno sve do šezdesetih godina 20. stoljeća.

U isto vrijeme započinje i razvoj komunikacijskih sustava, koji će postaviti temelje kasnijem razvoju računalnih mreža. Zato valja istaknuti 1844. godinu kada je Samuel Morse predstavio u SAD-u prvi telegrafski uređaj za prijenos poruka električkim impulsima prenošenih žicom.



Sl. 1-2 Prototip diferencijalnog stroja Charlesa Babbagea



Sl. 1-3 Jacquardov tkalački stroj

Godine 1854. George Boole objavljuje rad pod nazivom "An Investigation of the Laws of Thought", opisujući sustav simboličkog i logičkog zaključivanja, koji će kasnije postati jednom od temeljnih teorija u razvoju računarstva.

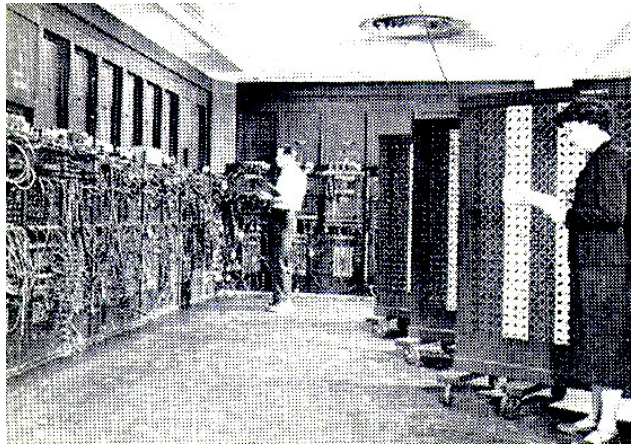
Herman Hollerith razvio je 1889. godine prvi električni stroj za računanje ("Electric Tabulating System").

Godine 1904. John A. Fleming je predstavio električnu vakuumsku diodnu cijev, koja donosi prekretnicu u razvoju elektroničkih uređaja i radio komunikacije.

Ubrzo, 1908. godine, engleski znanstvenik Campbell Swinton prikazuje metodu elektronskog skaniranja, utemeljujući razvoj katodne cijevi, odnosno televizije koja će biti predstavljena prvi put 1927. godine.

U Njemačkoj Konrad Zuse 1936. godine otkriva da se programi opisani binarnim kombinacijama mogu pohranjivati elektromehaničkim principom ("Combination memory").

Godine 1937. Alan Turing objavljuje rad "On Computable Numbers", prikazujući koncept Turingovoga stroja, koji je postao model za izgradnju modernih računala. Od tada događaji vezani uz razvoj modernog računarstva nezaustavljivo napreduju. U nekoliko sljedećih godina, sve do 1945., rađaju se digitalna računala jedan za drugim: prvo elektronsko računalo temeljeno na binarnoj aritmetici, John Vincent Atanasoff, 1939.; "Z1", "Z2", i "Z3", elektromehanička binarna računala, Konrad Zuse, 1938-1941.; "Colossus", prvo elektronsko računalo bazirano na elektronskim vakuumskim cijevima, Alan Turing, Tommy Flowers, M.H.A. Newman, Engleska, 1943.; "The Harvard Mark I", Howard Aiken, 1944.



Sl. 1-4 ENIAC

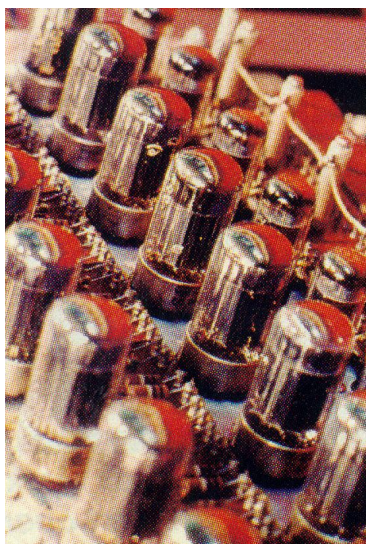
Pojavom računala ENIAC ("Electronic Numerical Integrator and Calculator", Sl. 1-4) 1945. godine nastupa prekretnica u razvoju računarstva. Razvili su ga John Mauchly i J. Presper Eckert na University of Pennsylvania. ENIAC se smatra posebno važnim jer utemeljuje koncept računala koji se zadržao sve do današnjih dana. Stoga se ENIAC smatra prvim računalom, jer se, za razliku od prethodnih razvojnih pokušaja, može izravno uspoređivati s današnjim računalima. Njegove dimenzije bile su zastrašujuće, mase oko 30 tona, a računarske sposobnosti daleko ispod današnjeg džepnog računala.

U sljedećih pedeset godina zbiva se ubrzani razvoj računarskih znanosti, ostavljajući traga u svim segmentima znanosti i društva, uvodeći nas u informatičko doba. U nastavku će biti spomenuta samo neka dostignuća koja su u proteklih pedeset godina presudno utjecala na razvoj računarstva i koja bi mogla pomoći razumijevanju sadašnjeg stanja tehnike i razvojnih trendova:

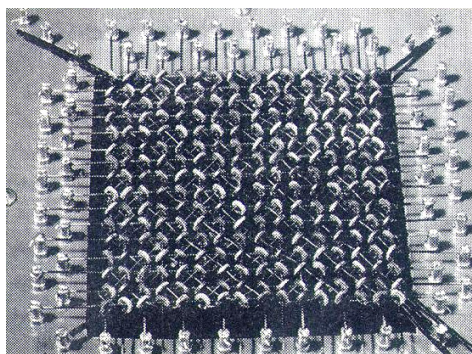
❑ ENIAC ("Electronic Numerical Integrator and Calculator")	John Mauchly i J. Presper Eckert na University of Pennsylvania	1945.
❑ magnetni disk		1947-48.
❑ tranzistor	John Bardeen, Walter Brattain, William Shockley, Bell Lab	1947.
❑ Turingov test inteligencije računala	Alan Turing	1950.
❑ memorija načinjena od željeznih jezgri (Sl. 1-6)		1951.
❑ A-0, prvi kompajler (prevodilac)	Grace Murray Hopper	1951-1952.
❑ UNIVAC 1103A, prvo komercijalno računalo sa memorijom od željeznih jezgri		1954.
❑ Fortran (Formula Translator), prvi viši programski jezik	John Backus, IBM	1957.
❑ osnovan prvi odjela umjetne inteligencije	John McCarthy, MIT	1957.
❑ poluvodički integrirani krug	Jack Kilby, Texas Instruments	1957.
❑ modem	Bell Lab	1958.
❑ COBOL (Common Business Oriented Language), programski jezik namjenjen poslovnim zadaćama		1959.
❑ Lisp, jezik namjenjen razvoju umjetne inteligencije	John McCarthy	1959.
❑ prekidačka tehnologija pakiranja informacija u komunikacijskim sustavima	Paul Baran, Rand Corp.	1960.
❑ Perceptron, umjetna neuronska mreža	Frank Rosenblatt, Cornell University	1960.
❑ robot (Sl. 1-7)	Georg S. Devol, Unimation	1961.
❑ osnovan prvi odjel računarskih znanosti	Stanford i Purdue University	1962.
❑ prva video igra	Steve Russell (student), MIT	1962.
❑ Eliza, "umjetni psihijatar", prvi program koji je oponašao inteligentno ponašanje	Joseph Weizenbaum, MIT	1963.
❑ računalna grafika	Ivan Sutherland	1963.
❑ neizravna logika, "Fuzzy logic"	Lotfi Zadeh, University of California	1963.
❑ BASIC (Beginner's All-Purpose Symbolic Instruction Code)	John Kemeny, Thomas Kurtz, Dartmouth	1964.
❑ CDC 6600 (9 megaflopsa), prvo superračunalo	Seymour Cray, Data Corp.	1964.
❑ CAD (Computer Aided	IBM	1964.

Drafting), program za računalnu podršku konstruiranju		
❑ miš	Doug Engelbart	1964.
❑ džepno računalo sa četiri matematičke operacije	Jack Kilby, Jerry Merryman, Texas Instruments	1967.
❑ osnivanje kompanije Intel za razvoj i proizvodnju procesora	Robert Noyce, Andy Grove, Gordon Moore	1968.
❑ ARPA, decentralizirani komunikacijski koncept računalne mreže	Rand Corp.	1968.
❑ Unix, prvi otvoreni operacijski sustav	Dennis Ritchie, Kenneth Thomson, Bell Lab	1969.-70.
❑ disketa		1970.
❑ prvi mikroprocesor Intel 4004 (4-bitni, 2300 tranzistora, 45 instrukcija, Sl. 1-9)	Ted Hoff, S. Mazor, F. Fagin, Intel	1971.
❑ E-mail		1971.
❑ Pascal, opći strukturirani programski jezik	Niklaus Wirth	1971.
❑ C, univerzalni niži programski jezik	Dennis Ritchie, Bell Lab	1972.
❑ Prolog, jezik namijenjen logičkom programiranju	Alain Colmerauer, University of Marseille	1972.
❑ TCP/IP	Vinton Cerf, Stanford University	1973.
❑ prvo osobno računalo PC-Alto	Xerox	1973.
❑ Ethernet, tehnologija umrežavanja računala	Robert Metcalfe	1973.
❑ laserski printer	IBM	1975.
❑ Cray-1, prvo superračunalo vektorske arhitekture		1976.
❑ Apple I, prvo komercijalno osobno računalo	Steve Jobs, Steve Wozniak	1976.
❑ osnovan Microsoft	Bill Gates	1977.
❑ The Osborn 1, prvo prenosivo računalo		1980.
❑ RISC koncept (Reduced Instruction Set Computer)	David A. Patterson, John Hennessy, University of California	1980.
❑ IBM PC		1981.
❑ AutoCAD, PC CAD program	Autodesk	1982.
❑ početak komercijalizacije e-maila		1982.
❑ Internet		1983.
❑ C++	Bjarne Stroustrup, At&T Bell Lab	1983.
❑ X Window System	MIT	1984.
❑ CD-ROM	Sony i Philips	1984.
❑ MS Windows 1.0	Microsoft	1985.
❑ RISC mikroprocesor, 32-bit Motorola 88000, 17 miliona instrukcija u sekundi	Motorola	1988.
❑ WWW, mrežni protokol	Tim Berners-Lee,	1989.

temeljen na HTML-u	CERN (European Council for Nuclear Research)	
<ul style="list-style-type: none"> <li>❑ 64-bitni RISC Aplha mikroprocesor (0.75 <math>\mu\text{m}</math>, CMOS tehnologija, 1,68 mil. tranzistora, 13.9x16.8 mm, 150 MHz, 300 MIPS-a, 150 MFLOPS-a)</li> <li>❑ Netscape</li> <li>❑ Java</li> <li>❑ MS Windows 95</li> <li>❑ MS Windows 2000</li> <li>❑ Pentium IV, prvi PC 64-bitni mikroprocesor</li> <li>❑ ...</li> </ul>	DEC	1992.
	Netscape	1994.
	SUN Microsystems	1995.
	Microsoft	1995.
	Microsoft	2000.
	Intel	2001.

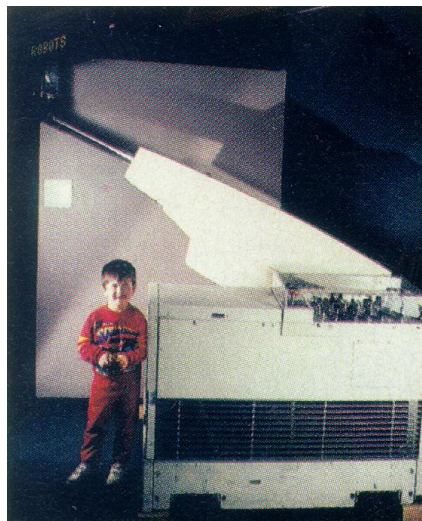


Sl. 1-5 Na početku razvoja računalne tehnologije elektronske vakuumske cijevi bile su osnova implementacije prekidačke logike (prvu elektronsku vakuumsku cijev patentirao je John A. Fleming 1904. godine).

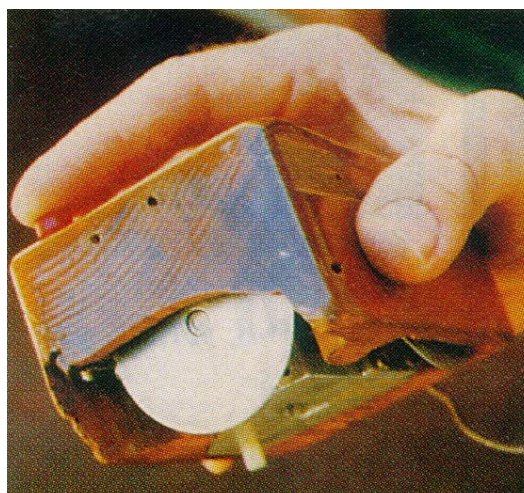


Sl. 1-6 Memorija načinjena od željeznih jezgri, 1951.

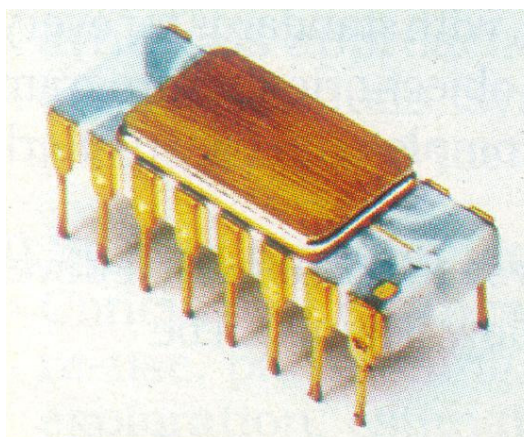




Sl. 1-7 Prvi robot upravljani računalom patentirao je George C. Devol 1961.



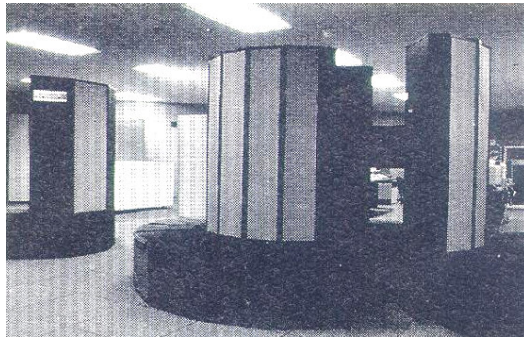
Sl. 1-8 Prvi miš, Doug Engelbart, 1964.



Sl. 1-9 Intel 4004, prvi mikroprocesor, 1971.



Sl. 1-10 Apple I, 1976.



Sl. 1-11 CRAY X-MP, početak superračunalnog doba, 1982. godina

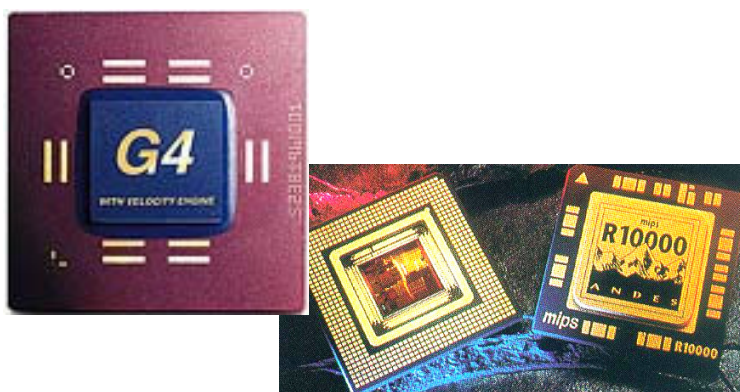


## 1.2 Suvremene informacijske tehnologije

Suvremenu informacijsku tehnologiju karakterizira mikroprocesorska tehnologija visokog stupnja integracije, računalne mreže te globalizacija i konvergencija informacijskih tehnologija.



To znači da računala postaju sve manja i sve većih kapaciteta i brzine obrade podataka. Procesori integiraju više desetaka milijuna tranzistora i sposobni su obrađivati više stotina milijuna operacija u sekundi (Sl. 1-12). Osobna računala posjeduju velike memorijske kapacitete, dovoljne za izvođenje najsloženijih programskih zadataka, od obrade multimedijalnih digitalnih informacija do zahtjevnih inženjerskih proračuna i simulacija (Sl. 1-13).



Sl. 1-12 Power Mac G4 procesor i superskalarni mikroprocesor MIPS R10000, 17x18 mm, 6.8 miliona tranzistora, 800 MIPS-a



Sl. 1-13 Osobna računala na pragu 21. stoljeća

Prijenosna računala (Sl. 1-14), bez obzira na svoju veličinu, u mnogome ne zaostaju za mogućnostima stabilnih računala. Pored toga, integriraju prijenosne komunikacijske usluge i cjelovitu podršku uredskom poslovanju (Sl. 1-15).



Sl. 1-14 Prijenosno računalo



Sl. 1-15 Džepna računala s pristupom na Internet i mobilne komunikacijske usluge

Širenje digitalne tehnologije u komunikacijskim (GSM, ISDN, Internet) i multimedijalnim domenama (TV, film, radio) dovodi do prožimanja, donedavno različitih, tehnologija. Informacija postaje globalni i univerzalni pojam, numerička predstava proizvoljnog sadržaja, koji se može prenositi različitim medijem, npr. žicom,

svjetlovodom ili mikrovalovima. O interpretaciji digitalne informacije brinu procesor i odgovarajući program (u TV prijemu, telefonu ili osobnom računaru), koji su zaduženi za prepoznavanje formata zapisa i njegovo prevođenje (npr. u zvučni signal, video informaciju ili korisnički program). Takav tehnološki trend stavlja poseban naglasak na razvoj sredstava za razmjenu podataka - komunikacijsku i mrežnu tehnologiju.

Zahtjevi u pogledu globalnih mrežnih kapaciteta višestruko se povećavaju svake godine. Stoga brzina primjene modernih informacijskih koncepcija poglavito ovisi o brzini razvoja komunikacijskih tehnologija.

---

## 2 OSNOVNI PRINCIPI RAČUNALNE TEHNOLOGIJE

---

**R**azumijevanje računarske tehnologije nije presudno za prosječnog korisnika, ali za one koji u njoj žele kompetentno sudjelovati, a to znači koristiti računarsku tehnologiju za rješavanje složenih zadataka te prilagođivati i unapređivati je u skladu sa svojim potrebama, kao i sudjelovati u njezinom razvoju, to je znanje iznimno važno. Ono otvara nove prostore i potiče nove ideje koje ruše granice maštanja i pretvaraju budućnost u stvarnost.

Princip rada računala temelji se na Turingovom teorijskom modelu algoritmičkog stroja. Model predviđa stroj koji je sposoban slijedno interpretirati simbolički opisane zadatke - algoritme. U realizaciji Turingovog stroja prvo je trebalo riješiti problem kako efikasno prikazati i interpretirati simbole. Taj problem je riješen primjenom jednostavne prekidačke elektroničke tehnologije, koja u osnovi omogućava razlikovanje tek dva stanja, dva simbola, uključeno - 1 ili isključeno - 0. Stoga je računalo binarni (lat. bini- po dva, dvojni) stroj u kojemu se ostala stanja i simboli moraju kodirati slogovima nula i jedinica. U nastavku će biti pomnije opisani Turingov stroj i binarni brojevni sustav kao osnovni čimbenici tehnologije računala - slijednog, algoritmičkog i binarnog stroja.

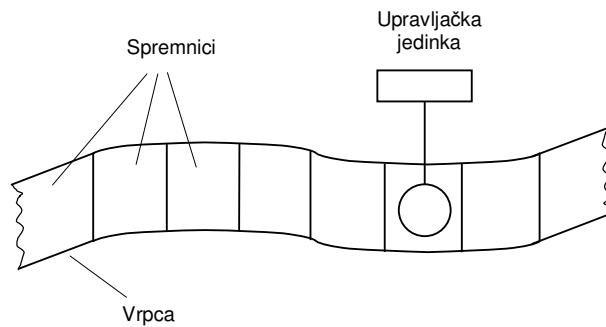




## 2.1 Turingov stroj

Suvremena se računala temelje na konceptu algoritmičkog stroja koji je razvio Alan Turing još 1937. godine, istražujući teoriju algoritmičkih postupaka. Nazvan po autoru, "Turingov stroj" predstavlja teorijski model računala, jer u vrijeme nastajanja nisu postojale tehnološke mogućnosti za realizaciju takvog stroja.

Turingov stroj se sastoji od upravljačke jedinice koja može čitati i pisati simboličke podatke na vrpca (Sl. 2-1). Vrpca beskonačne dužine podijeljena je na spremnike (stanice, eng. cells). Svaki spremnik može sadržavati bilo koji konačan skup simbola. Takav se skup naziva "alfabet".

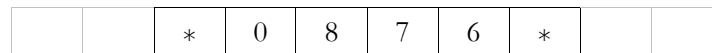


Sl. 2-1 Turingov stroj

Rad stroja se sastoji od niza koraka koje izvodi upravljačka jedinica. Tijekom rada, stroj mijenja svoje stanje u skladu sa sadržajem spremnika. Početak rada započinje sa stanjem "START" i traje do stanja "STOP" (eng. "HALT"). U svakom koraku upravljačka jedinica interpretira sadržaj tekućeg spremnika i potom se kreće lijevo ili desno mijenjajući stanje.

Princip Turingovog stroja najbolje se može ilustrirati na konkretnom primjeru. Pretpostavimo da nam je zadaća realizirati algoritam zaokruživanja cjelobrojnih vrijednosti na vrijednosti djeljive s deset. U tom slučaju znamenke zadanog broja bit će pohranjene u odgovarajućim spremnicima zamišljene vrpce kao što je prikazano na slici

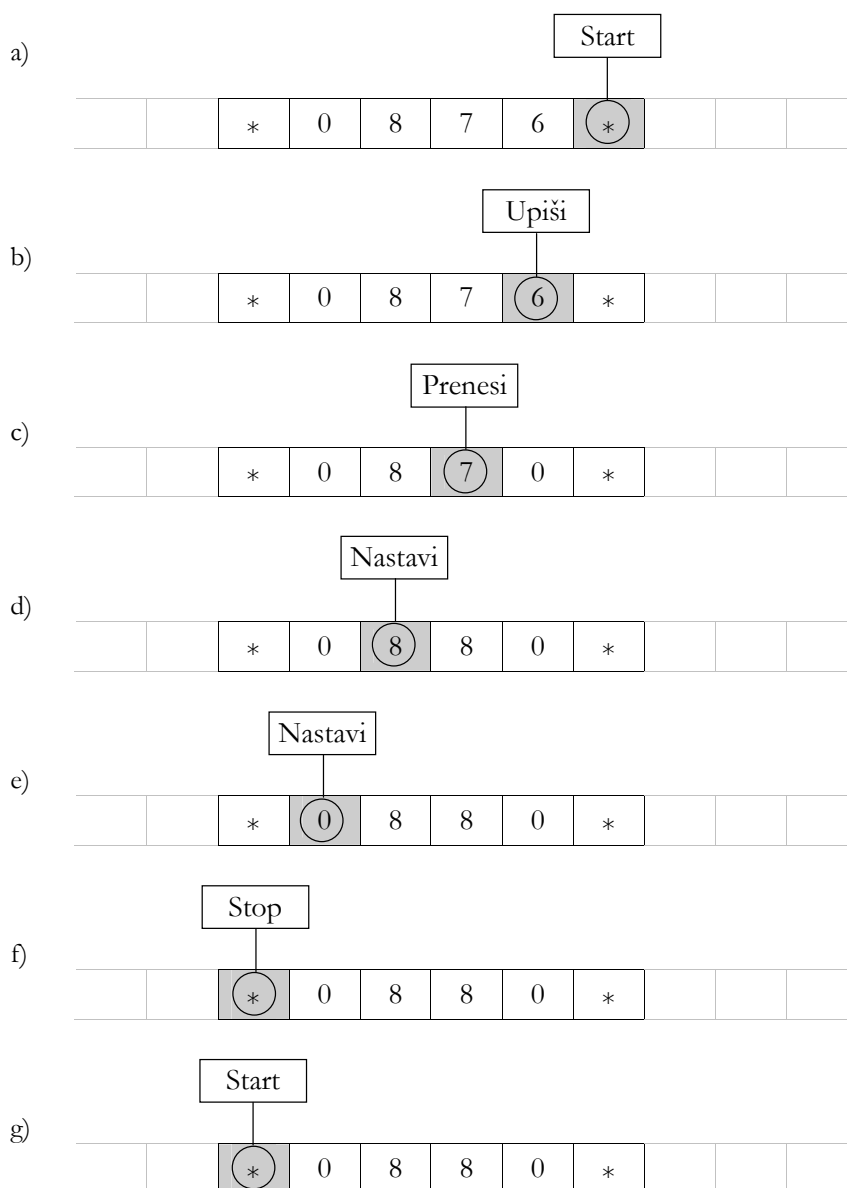
Sl. 2-2.



Sl. 2-2 Primjer pohrane broja u Turingovu memoriju

Znamenke koje definiraju pohranjenu vrijednost omeđene su spremnicima koje sadrže simbol "\*". Pretpostavimo da početna pozicija upravljačke jedinice u stanju "START" polazi od krajnjeg desnog spremnika označenog s "\*". U tom stanju i položaju nužno

se nameće naredba kretanja "LIJEVO". Stoga upravljačka jedinica u sljedećem koraku zauzima poziciju susjednog lijevog spremnika, poprimajući stanje "UPIŠI". Ukoliko je pohranjena znamenka veća ili jednaka pet, stanje upravljačke jedinice prelazi u "PRENESI", a ukoliko je manja od pet stanje prelazi u "NASTAVI". Slijedi upisivanje znamenke 0 u tekući spremnik i potom pomak za jedno mjesto u lijevo. Ako je na novoj poziciji aktivno stanje "PRENESI", a pohranjeni broj je manji od 9, slijedi upisivanje znamenke za 1 veće od tekuće. Kretanje u lijevo nastavlja se s novim stanjem "NASTAVI" (do sljedećeg pohranjenog broja). Ako je pohranjena znamenka 9, uvećavanje za 1 daje 10, što znači da je potrebno upisati 0 i nastaviti ulijevo s stanjem "PRENESI". Pomicanje ulijevo se nastavlja sve do spremnika u kojemu je pohranjen simbol "\*". Tada stanje stroja prelazi u "STOP" i slijedi isti postupak zaokruživanja sljedećeg broja (slika ). Opisani algoritam može se sažeti u Tab. 2-1.



Sl. 2-3 Postupak zaokruživanja Turingovim strojem

Tab. 2-1 Stanja i operacije algoritma zaokruživanja

Tekuće stanje	Sadržaj spremnika	Upis	Kretanje	Novo stanje
START	*	*	LIJEVO	UPIŠI
UPIŠI	$> 5$	0	LIJEVO	PRENESI
UPIŠI	$\leq 5$	0	LIJEVO	NASTAVI
PRENESI	$n \in [0, 8]$	$n + 1$	LIJEVO	NASTAVI
PRENESI	$n = 9$	0	LIJEVO	PRENESI
NASTAVI	$n \in [0, 9]$	$n$	LIJEVO	NASTAVI
NASTAVI	*	*	STANI	STOP
STOP	*	*	STANI	START

Ako sada implementaciju ovoga algoritma zamislimo na računalu, lako ćemo uočiti povezanost s Turingovim strojem. Ulogu upravljačke jedinice preuzeo je središnji procesor računala (eng. "CPU - Central Processor Unit"), a vrpca sa spremnicima postala je radna memorija (eng. "RAM - Random Access Memory").

Dakle, računalo kao Turingov stroj u osnovi se sastoji od procesora i memorije te izvodi programe koji opisuju zadane algoritme koristeći potpuno isti princip. Međutim, za razliku od Turingovog stroja, koji je teoretski predviđao rad s proizvoljnim simbolima, računalo je temeljeno na prekidačkoj tehnologiji koja omogućava pohranu tek dvaju podataka 1 i 0, odnosno koristi se prepoznavanjem stanja "uključeno" i "isključeno". Prisjetimo se, već od pojave bušenih kartica na početku 19. stoljeća, zatim pojavom Booleove logike, 1854. godine, te radovima Konrada Zusea, 1936. godine, postavljeni su principi pohrane podataka temeljeni na prekidačkoj tehnologiji. Isprva je ta tehnologija bila realizirana mehaničkim sklopovima, potom elektromehaničkim, i najzad poluvodičkim integriranim električkim sklopovima. Takav pristup predviđa interpretaciju simbola i signala (brojeva, slova, zvukova, slike i dr.) binarnim kombinacijama jedinica i ničtica.



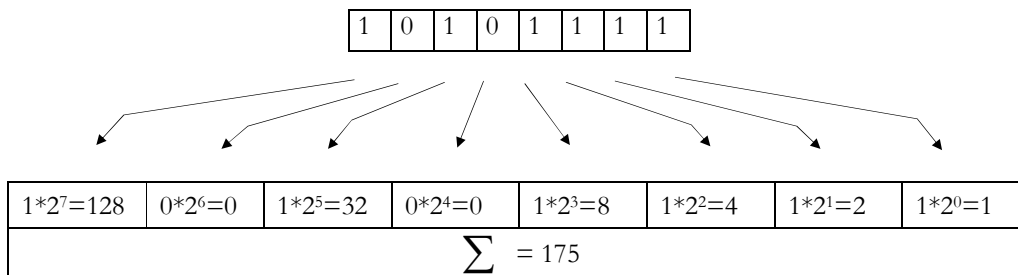
## 2.2 Binarni brojevni sustav

Binarni sustav je zasnovan na potencijama broja dva, koristeći samo dva simbola (znaka) 0 i 1.

Binarni broj je jedinična elementarna informacija nazvana "bit".

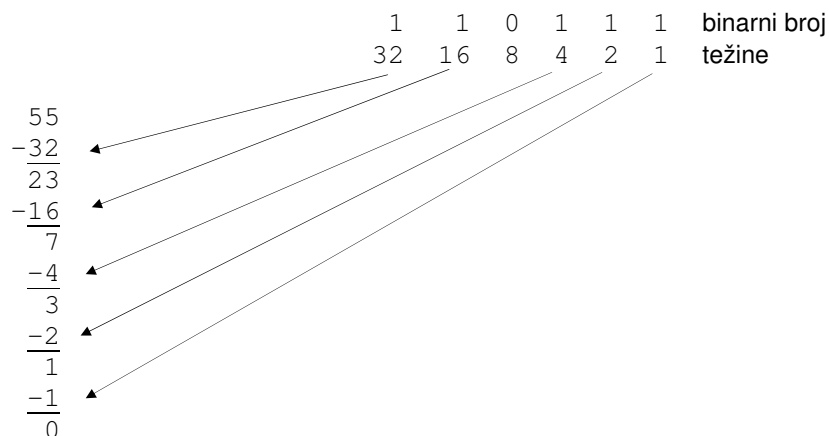


Binarni brojevni sustav pridružuje svakoj binarnoj znamenki odgovarajuću "težinu" zavisno o njezinom položaju u kombinaciji. Težine se povećavaju od lijeva na desno po pravilu  $2^n$  (u decimalnom sustavu to je  $10^n$ ), gdje je  $n = 0, 1, 2, \dots, N-1$ ,  $N$  je broj znamenki kombinacije. Množenjem težine i pripadajuće binarne vrijednosti u kombinaciji i zbrojem dobivenih umnožaka dobiva se odgovarajuća decimalna vrijednost. Na primjer, binarna kombinacija 10101111 odgovara decimalnom broju 175. Sl. 2-4 ilustrira postupak dešifriranja.



Sl. 2-4 Pretvorba binarnog broja u dekadski

Obrnuto, pretvaranje decimalno broja u binarne izvodi se oduzimanjem najveće potencije broja 2 koja je manja ili jednaka zadanom decimalnom broju. Potom se postupak ponavlja za ostatak sve dok ne ostane 0. Npr. za decimalni broj 55 postupak pretvorbe prikazan je na Sl. 2-5.



Sl. 2-5 Pretvorba dekadskog broja u binarni

Drugi način ide obrnutim redom, a svodi se na postupno dijeljenje dekadskog broja bazom 2. Ostatak dijeljenja predstavlja odgovarajuću binarnu znamenku u slogu. Za broj 55 postupak je sljedeći:

$$\begin{array}{r}
 55 : 2 = 27 + \\
 27 : 2 = 13 + \\
 13 : 2 = 6 + \\
 6 : 2 = 3 + \\
 3 : 2 = 1 + \\
 1 : 2 = 0 +
 \end{array}
 \begin{array}{|c|}
 \hline
 1 \\
 1 \\
 1 \\
 0 \\
 1 \\
 1 \\
 \hline
 \end{array}
 \longrightarrow 111011$$

## 2.2.1 Prikaz negativnih cjelobrojnih vrijednosti

### 2.2.1.1 Ekscesivna notacija

Dosada se govorilo isključivo o predstavi pozitivnih cjelobrojnih vrijednosti. Jedan od čestih načina prikaza negativnih cjelobrojnih vrijednosti je "ekscesivna notacija" (eng. excess notation). Kod ekscesivne notacije važno je prvo definirati dužinu binarnog sloga. Kombinacije se više ne dekodiraju na temelju funkcije težina  $2^n$ , bar ne u cijelosti. Prvi bit s lijeva služi kao oznaka predznaka; 1 označava pozitivan, a 0 negativan predznak. Ostalim bitovima kodiraju se brojevi, počevši od kombinacije 000...0 do 111...1, u rasponu od najmanjeg negativnog do najvećeg pozitivnog broja koji se može prikazati raspoloživim brojem bitova u zadanoj dužini sloga, slijedeći pravilo  $(n - 8)$  dekadski ili  $-(1000 - n)$  binarno. U Tab. 2-2 prikazana je ekscesivna notacija četverobitnog sloga.

Tab. 2-2 Ekscesivna notacija

Binarna kombinacija	Odgovarajuća decimalna vrijednost
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

### 2.2.1.2 Komplementarna notacija

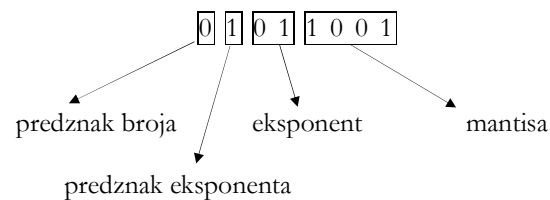
U novije doba sve je češća upotreba tzv. komplementarne notacije. Pritom se prvi bit također koristi kao oznaka predznaka, dok se ostalim bitovima izražavaju brojevi po uobičajenom principu s bazom  $2^n$ , s jedinom razlikom da za negativne vrijedi komplementarno pravilo plus 1, tj. 0 se interpretira kao 1 i obrnuto, pribrajajući jedinicu (Tab. 2-3).

Tab. 2-3 Komplementarna notacija

Binarna kombinacija	Odgovarajuća decimalna vrijednost
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

### 2.2.2 Prikaz racionalnih brojeva

Racionalni brojevi prikazuju se u binarnom sustavu pomoću tzv. notacije pomičnog zareza ili pomične točke. Shema ove notacije prikazana je na Sl.6.



Sl. 2-6 Prikaz racionalnih brojeva

Prvi bit označava predznak, zatim sljedeća skupina bitova definira polje eksponenta, a preostali bitovi prikazuju mantisu. Jedan od načina interpretacije opisan je u nastavku.

Kod dekodiranja izdvaja se prvo mantisa i postavlja se zarez (ili točka) na početak s lijeve strane. Na primjer, 8-bitna binarna kombinacija 11100111 ima mantisu:

,0111

Sljedeća tri bita 110 prikazuju eksponent u ekscesivnoj notaciji, što znači da je eksponent pozitivan (1 na prvome mjestu) i iznosi 2 ( $1*2^1+0*2^0$ ). Stoga se zarez mantise pomiče dva mjesta udesno:

01,11

Binarna kombinacija ispred zareza definira cijeli broj 1 ( $0*2^1+1*2^0$ ), dok bitovi iza zareza prikazuju decimalne dijelove po pravilu  $2^{-n}$ :

$$\frac{1}{1*2^1} + \frac{1}{1*2^2} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$$

Dakle, decimalna interpretacija binarne kombinacije 11100111 je  $1\frac{3}{4}$ .





## 2.3 Prikaz simboličkih podataka

Interakcija s računalom uključuje jednako grafička kao i deskriptivna sredstva. Grafička sredstva podrazumijevaju korištenje slikovnih simbola kojima je pridruženo određeno značenje. Izborom nekog od grafičkih simbola pokreće se odgovarajući program računala. Deskriptivna sredstva podrazumijevaju pismenu komunikaciju s računalom. To znači korištenje simboličkih podataka (grafema, pravopisnih i posebnih znakova) za izražavanje naredbi i prikaz podataka.

S obzirom da računalo može interpretirati jedino binarne kombinacije, potrebno je simboličkim podacima pridružiti odgovarajuće binarne kodove.



Danas je opće prihvaćena ASCII (eng. "American Standard Code for Information Interchange") kodirna norma. Koristi 7-bitni slog za kodiranje velikih i malih slova, brojaka te pravopisnih, matematičkih i posebnih znakova. Prošireni ASCII sustav ima 8-bitni slog i obuhvaća 256 različitih simbola. U Tab. 2-4 prikazan je djelomični ASCII sustav označavanja.

Tab. 2-4 ASCII tablica

Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin
^@ =	0	0x00	00	%00000000	M-^@ =	128	0x80	0200	%10000000
^A =	1	0x01	01	%00000001	M-^A =	129	0x81	0201	%10000001
^B =	2	0x02	02	%00000010	M-^B =	130	0x82	0202	%10000010
^C =	3	0x03	03	%00000011	M-^C =	131	0x83	0203	%10000011
^D =	4	0x04	04	%00000100	M-^D =	132	0x84	0204	%10000100
^E =	5	0x05	05	%00000101	M-^E =	133	0x85	0205	%10000101
^F =	6	0x06	06	%00000110	M-^F =	134	0x86	0206	%10000110
^G =	7	0x07	07	%00000111	M-^G =	135	0x87	0207	%10000111
^H =	8	0x08	010	%00001000	M-^H =	136	0x88	0210	%10001000
^I =	9	0x09	011	%00001001	M-^I =	137	0x89	0211	%10001001
^J =	10	0x0A	012	%00001010	M-^J =	138	0x8A	0212	%10001010
^K =	11	0x0B	013	%00001011	M-^K =	139	0x8B	0213	%10001011
^L =	12	0x0C	014	%00001100	M-^L =	140	0x8C	0214	%10001100
^M =	13	0x0D	015	%00001101	M-^M =	141	0x8D	0215	%10001101
^N =	14	0x0E	016	%00001110	M-^N =	142	0x8E	0216	%10001110
^O =	15	0x0F	017	%00001111	M-^O =	143	0x8F	0217	%10001111
^P =	16	0x10	020	%00010000	M-^P =	144	0x90	0220	%10010000
^Q =	17	0x11	021	%00010001	M-^Q =	145	0x91	0221	%10010001
^R =	18	0x12	022	%00010010	M-^R =	146	0x92	0222	%10010010
^S =	19	0x13	023	%00010011	M-^S =	147	0x93	0223	%10010011
^T =	20	0x14	024	%00010100	M-^T =	148	0x94	0224	%10010100
^U =	21	0x15	025	%00010101	M-^U =	149	0x95	0225	%10010101
^V =	22	0x16	026	%00010110	M-^V =	150	0x96	0226	%10010110
^W =	23	0x17	027	%00010111	M-^W =	151	0x97	0227	%10010111
^X =	24	0x18	030	%00011000	M-^X =	152	0x98	0230	%10011000
^Y =	25	0x19	031	%00011001	M-^Y =	153	0x99	0231	%10011001
^Z =	26	0x1A	032	%00011010	M-^Z =	154	0x9A	0232	%10011010
^[ =	27	0x1B	033	%00011011	M-^[ =	155	0x9B	0233	%10011011
^\ =	28	0x1C	034	%00011100	M-^\ =	156	0x9C	0234	%10011100
^] =	29	0x1D	035	%00011101	M-^] =	157	0x9D	0235	%10011101
^^ =	30	0x1E	036	%00011110	M-^^ =	158	0x9E	0236	%10011110
^_ =	31	0x1F	037	%00011111	M-^_ =	159	0x9F	0237	%10011111
! =	32	0x20	040	%00100000	=	160	0xA0	0240	%10100000
" =	33	0x21	041	%00100001	=	161	0xA1	0241	%10100001
# =	34	0x22	042	%00100010	=	162	0xA2	0242	%10100010
\$ =	35	0x23	043	%00100011	=	163	0xA3	0243	%10100011
=	36	0x24	044	%00100100	=	164	0xA4	0244	%10100100

2 OSNOVNI PRINCIPI RAČUNALNE TEHNOLOGIJE

%	=	37	-	0x25	-	045	-	%00100101
&	=	38	-	0x26	-	046	-	%00100110
'	=	39	-	0x27	-	047	-	%00100111
(	=	40	-	0x28	-	050	-	%00101000
)	=	41	-	0x29	-	051	-	%00101001
*	=	42	-	0x2A	-	052	-	%00101010
+	=	43	-	0x2B	-	053	-	%00101011
,	=	44	-	0x2C	-	054	-	%00101100
-	=	45	-	0x2D	-	055	-	%00101101
.	=	46	-	0x2E	-	056	-	%00101110
/	=	47	-	0x2F	-	057	-	%00101111
0	=	48	-	0x30	-	060	-	%00110000
1	=	49	-	0x31	-	061	-	%00110001
2	=	50	-	0x32	-	062	-	%00110010
3	=	51	-	0x33	-	063	-	%00110011
4	=	52	-	0x34	-	064	-	%00110100
5	=	53	-	0x35	-	065	-	%00110101
6	=	54	-	0x36	-	066	-	%00110110
7	=	55	-	0x37	-	067	-	%00110111
8	=	56	-	0x38	-	070	-	%00111000
9	=	57	-	0x39	-	071	-	%00111001
:	=	58	-	0x3A	-	072	-	%00111010
;	=	59	-	0x3B	-	073	-	%00111011
<	=	60	-	0x3C	-	074	-	%00111100
=	=	61	-	0x3D	-	075	-	%00111101
>	=	62	-	0x3E	-	076	-	%00111110
?	=	63	-	0x3F	-	077	-	%00111111
@	=	64	-	0x40	-	0100	-	%01000000
A	=	65	-	0x41	-	0101	-	%01000001
B	=	66	-	0x42	-	0102	-	%01000010
C	=	67	-	0x43	-	0103	-	%01000011
D	=	68	-	0x44	-	0104	-	%01000100
E	=	69	-	0x45	-	0105	-	%01000101
F	=	70	-	0x46	-	0106	-	%01000110
G	=	71	-	0x47	-	0107	-	%01000111
H	=	72	-	0x48	-	0110	-	%01001000
I	=	73	-	0x49	-	0111	-	%01001001
J	=	74	-	0x4A	-	0112	-	%01001010
K	=	75	-	0x4B	-	0113	-	%01001011
L	=	76	-	0x4C	-	0114	-	%01001100
M	=	77	-	0x4D	-	0115	-	%01001101
N	=	78	-	0x4E	-	0116	-	%01001110
O	=	79	-	0x4F	-	0117	-	%01001111
P	=	80	-	0x50	-	0120	-	%01010000
Q	=	81	-	0x51	-	0121	-	%01010001
R	=	82	-	0x52	-	0122	-	%01010010
S	=	83	-	0x53	-	0123	-	%01010011
T	=	84	-	0x54	-	0124	-	%01010100
U	=	85	-	0x55	-	0125	-	%01010101
V	=	86	-	0x56	-	0126	-	%01010110
W	=	87	-	0x57	-	0127	-	%01010111
X	=	88	-	0x58	-	0130	-	%01011000
Y	=	89	-	0x59	-	0131	-	%01011001
Z	=	90	-	0x5A	-	0132	-	%01011010
[	=	91	-	0x5B	-	0133	-	%01011011
\	=	92	-	0x5C	-	0134	-	%01011100
]	=	93	-	0x5D	-	0135	-	%01011101
^	=	94	-	0x5E	-	0136	-	%01011110
_	=	95	-	0x5F	-	0137	-	%01011111
~	=	96	-	0x60	-	0140	-	%01100000
a	=	97	-	0x61	-	0141	-	%01100001
b	=	98	-	0x62	-	0142	-	%01100010
c	=	99	-	0x63	-	0143	-	%01100011
d	=	100	-	0x64	-	0144	-	%01100100
e	=	101	-	0x65	-	0145	-	%01100101
f	=	102	-	0x66	-	0146	-	%01100110
g	=	103	-	0x67	-	0147	-	%01100111
h	=	104	-	0x68	-	0150	-	%01101000
i	=	105	-	0x69	-	0151	-	%01101001
j	=	106	-	0x6A	-	0152	-	%01101010
k	=	107	-	0x6B	-	0153	-	%01101011
l	=	108	-	0x6C	-	0154	-	%01101100
m	=	109	-	0x6D	-	0155	-	%01101101
¥	=	165	-	0xA5	-	0245	-	%10100101
ı	=	166	-	0xA6	-	0246	-	%10100110
Ş	=	167	-	0xA7	-	0247	-	%10100111
ˆ	=	168	-	0xA8	-	0250	-	%10101000
(ç)	=	169	-	0xA9	-	0251	-	%10101001
ª	=	170	-	0xAA	-	0252	-	%10101010
«	=	171	-	0xAB	-	0253	-	%10101011
¬	=	172	-	0xAC	-	0254	-	%10101100
ˆ	=	173	-	0xAD	-	0255	-	%10101101
(r)	=	174	-	0xAE	-	0256	-	%10101110
ˆ	=	175	-	0xAF	-	0257	-	%10101111
º	=	176	-	0xB0	-	0260	-	%10110000
±	=	177	-	0xB1	-	0261	-	%10110001
²	=	178	-	0xB2	-	0262	-	%10110010
³	=	179	-	0xB3	-	0263	-	%10110011
´	=	180	-	0xB4	-	0264	-	%10110100
µ	=	181	-	0xB5	-	0265	-	%10110101
¶	=	182	-	0xB6	-	0266	-	%10110110
·	=	183	-	0xB7	-	0267	-	%10110111
,	=	184	-	0xB8	-	0270	-	%10111000
ˆ	=	185	-	0xB9	-	0271	-	%10111001
º	=	186	-	0xBA	-	0272	-	%10111010
»	=	187	-	0xBB	-	0273	-	%10111011
¼	=	188	-	0xBC	-	0274	-	%10111100
½	=	189	-	0xBD	-	0275	-	%10111101
¾	=	190	-	0xBE	-	0276	-	%10111110
¿	=	191	-	0xBF	-	0277	-	%10111111
À	=	192	-	0xC0	-	0300	-	%11000000
Á	=	193	-	0xC1	-	0301	-	%11000001
Â	=	194	-	0xC2	-	0302	-	%11000010
Ã	=	195	-	0xC3	-	0303	-	%11000011
Ä	=	196	-	0xC4	-	0304	-	%11000100
Å	=	197	-	0xC5	-	0305	-	%11000101
Æ	=	198	-	0xC6	-	0306	-	%11000110
Ç	=	199	-	0xC7	-	0307	-	%11000111
È	=	200	-	0xC8	-	0310	-	%11001000
É	=	201	-	0xC9	-	0311	-	%11001001
Ê	=	202	-	0xCA	-	0312	-	%11001010
Ë	=	203	-	0xCB	-	0313	-	%11001011
Ì	=	204	-	0xCC	-	0314	-	%11001100
Í	=	205	-	0xCD	-	0315	-	%11001101
Î	=	206	-	0xCE	-	0316	-	%11001110
Ï	=	207	-	0xCF	-	0317	-	%11001111
Ð	=	208	-	0xD0	-	0320	-	%11010000
Ñ	=	209	-	0xD1	-	0321	-	%11010001
Ò	=	210	-	0xD2	-	0322	-	%11010010
Ó	=	211	-	0xD3	-	0323	-	%11010011
Ô	=	212	-	0xD4	-	0324	-	%11010100
Õ	=	213	-	0xD5	-	0325	-	%11010101
Ö	=	214	-	0xD6	-	0326	-	%11010110
×	=	215	-	0xD7	-	0327	-	%11010111
Ø	=	216	-	0xD8	-	0330	-	%11011000
Ù	=	217	-	0xD9	-	0331	-	%11011001
Ú	=	218	-	0xDA	-	0332	-	%11011010
Û	=	219	-	0xDB	-	0333	-	%11011011
Ü	=	220	-	0xDC	-	0334	-	%11011100
Ý	=	221	-	0xDD	-	0335	-	%11011101
Þ	=	222	-	0xDE	-	0336	-	%11011110
ß	=	223	-	0xDF	-	0337	-	%11011111
à	=	224	-	0xE0	-	0340	-	%11100000
á	=	225	-	0xE1	-	0341	-	%11100001
â	=	226	-	0xE2	-	0342	-	%11100010
ã	=	227	-	0xE3	-	0343	-	%11100011
ä	=	228	-	0xE4	-	0344	-	%11100100
å	=	229	-	0xE5	-	0345	-	%11100101
æ	=	230	-	0xE6	-	0346	-	%11100110
ç	=	231	-	0xE7	-	0347	-	%11100111
è	=	232	-	0xE8	-	0350	-	%11101000
é	=	233	-	0xE9	-	0351	-	%11101001
ê	=	234	-	0xEA	-	0352	-	%11101010
ë	=	235	-	0xEB	-	0353	-	%11101011
ì	=	236	-	0xEC	-	0354	-	%11101100
í	=	237	-	0xED	-	0355	-	%11101101

2 OSNOVNI PRINCIPI RAČUNALNE TEHNOLOGIJE

n = 110 - 0x6E - 0156 - %01101110	î = 238 - 0xEE - 0356 - %11101110
o = 111 - 0x6F - 0157 - %01101111	ï = 239 - 0xEF - 0357 - %11101111
p = 112 - 0x70 - 0160 - %01110000	ð = 240 - 0xF0 - 0360 - %11110000
q = 113 - 0x71 - 0161 - %01110001	ñ = 241 - 0xF1 - 0361 - %11110001
r = 114 - 0x72 - 0162 - %01110010	ò = 242 - 0xF2 - 0362 - %11110010
s = 115 - 0x73 - 0163 - %01110011	ó = 243 - 0xF3 - 0363 - %11110011
t = 116 - 0x74 - 0164 - %01110100	ô = 244 - 0xF4 - 0364 - %11110100
u = 117 - 0x75 - 0165 - %01110101	õ = 245 - 0xF5 - 0365 - %11110101
v = 118 - 0x76 - 0166 - %01110110	ö = 246 - 0xF6 - 0366 - %11110110
w = 119 - 0x77 - 0167 - %01110111	÷ = 247 - 0xF7 - 0367 - %11110111
x = 120 - 0x78 - 0170 - %01111000	ø = 248 - 0xF8 - 0370 - %11111000
y = 121 - 0x79 - 0171 - %01111001	ù = 249 - 0xF9 - 0371 - %11111001
z = 122 - 0x7A - 0172 - %01111010	ú = 250 - 0xFA - 0372 - %11111010
{ = 123 - 0x7B - 0173 - %01111011	û = 251 - 0xFB - 0373 - %11111011
= 124 - 0x7C - 0174 - %01111100	ü = 252 - 0xFC - 0374 - %11111100
} = 125 - 0x7D - 0175 - %01111101	ý = 253 - 0xFD - 0375 - %11111101
~ = 126 - 0x7E - 0176 - %01111110	þ = 254 - 0xFE - 0376 - %11111110
^? = 127 - 0x7F - 0177 - %01111111	ÿ = 255 - 0xFF - 0377 - %11111111



## 2.4 Aritmetičko/logičke operacije

### 2.4.1 Binarna aritmetika

Osim sustava binarnog označavanja brojeva potrebno je definirati binarnu aritmetiku. Aritmetika (grč. áarithmos, broj) je znanost o računanju i brojevima. Definira pravila matematičkih operacija. Ta se pravila razlikuju za različite brojevne sustave. U Tab. 2-5 definirana su pravila za četiri osnovne binarne operacije: zbrajanje, oduzimanje, množenje i dijeljenje.

Tab. 2-5 Binarna aritmetika za zbrajanje, oduzimanje, množenje i dijeljenje

Binarno zbrajanje			
Pribrojnici		Zbroj	Prijenos
0 + 0	=	0	0
0 + 1	=	1	0
1 + 0	=	1	0
1 + 1	=	0	1

Binarno oduzimanje			
Umanjenik - umanjitelj		Razlika	Posudba
0 - 0	=	0	0
0 - 1	=	1	1
1 - 0	=	1	0
1 - 1	=	0	0

Binarno množenje			
Množenik x množitelj		Umnožak	
0 x 0	=	0	
0 x 1	=	0	
1 x 0	=	0	
1 x 1	=	1	

Binarno dijeljenje		
Djeljenik / djelitelj		Količnik
0 / 0	=	0
0 / 1	=	0
1 / 0	=	?
1 / 1	=	1

Pravila binarne aritmetike očito su jednostavnija od aritmetike u decimalnom sustavu. Upravo zato su binarni brojevi prikladni za elektroničku implementaciju. U nastavku su dani primjeri računskih operacija s binarnim brojevima.

### Zbrajanje

Binarno zbrajanje obavlja se isto kao i decimalno zbrajanje, osim što se prijenos na sljedeće značajnije mjesto ne obavlja nakon postignutog zbroja 10, već 2 (1+1), npr.:

011000	prijenos
101101	1. pribrojnik (45)
+ 1100	2. pribrojnik (12)
<b>111001</b>	zbroj (57)

### Oduzimanje

Oduzimanje se može izvoditi kao kod decimalnih brojeva ili zbrajanjem dvojnog komplementarnog umanjitelja.

011110	posudba
101001	umanjenik (41)
- 11011	umanjitelj (27)
<b>01110</b>	razlika (14)

Dvojni komplementarni umanjitelj je broj koji se dobije tako da se jediničnom komplementu pribroji 1. Jedinični komplement je broj suprotnih bitova. Dvojni komplement pribrojen originalnom broju daje zbroj jednak nuli i jedinicu preteka, npr.:

010 110 110 110	originalni broj
101 001 001 001	jedinični komplement
+ 1	
101 001 001 010	dvojni komplement
010 110 110 110	originalni broj
+ 101 001 001 010	dvojni komplement
1 000 000 000 000	

Zbroj umanjnika i dvojnog komplementarnog umanjitelja daje razliku s pretekom 1. Za razliku (41 - 27) postupak je sljedeći:

00001	prijenos
101001	umanjnik (41)
- 11011	umanjitelj (27)
00100	jedinični komplement
+ 1	
+ 00101	dvojni komplement
101110	razlika (14)

Rezultat oduzimanja, dobiven pribrajanjem dvojnoga komplementa broja 27, je binarni broj s pretekom jedinice koja se zanemaruje.

### ***Množenje***

Kod binarnog množenja djelomičan umnožak pomiče se za jedno mjesto ulijevo pri svakom uzimanju idućeg množitelja, npr.:

101101	1. množitelj (45)
x 101	2. množitelj (5)
101101	
000000	
101101	
11100001	umnožak (225)

### ***Dijeljenje***

Primjenom pravila binarnog oduzimanja i množenja može se obaviti binarno dijeljenje na isti način kao dijeljenje decimalnih brojeva, npr.:

1010	djeljenik (10)
/ 10	djelitelj (2)
10/10=1	ostatak 0
1/10=0	ostatak 1
10/10=1	ostatak 0
101	količnik (5)

## **2.4.2 Binarne logičke operacije**

Većinu algoritama nemoguće je realizirati samo kroz matematičke operacije. Dapače, veliki broj primjena i općenito programskih cjelina oslanja se na nenumeričke operacije, npr. uspoređivanje simboličkih podataka (naziva, adresa) ili uvjeta za odlučivanje. Nenumeričke operacije nazivaju se logičke ili Booleove operacije. Naziv potiče od autora, Geoga Boola koji je 1854. godine razvio binarnu logiku ili, kako se često zove, Booleovu logiku. Osobiti doprinos ove matematičke teorije očituje se u mogućnosti izražavanja vrlo složenih algoritamskih konstrukcija kombiniranjem jednostavnih binarnih pravila. To znači mogućnost da se složeni algoritmi rješavaju jednostavnom prekidačkom tehnikom. Zato je Booleova logika postala temeljni koncept računalne tehnologije.



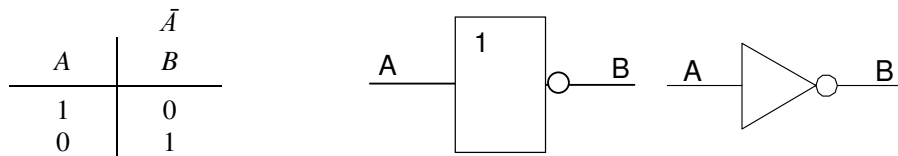
Osnovni logički operatori su: NE (NOT), I (AND, \*, ^), ILI (OR, +, ∨), Ex-ILI (EX-OR, ⊕, ekskluzivno ili), NI (NAND) i NILI (NOR). Logički operatori zovu se još logički članovi ili logička vrata (eng. gate). Temeljne varijable Booleovih operatora su binarne veličine 1 ili 0. U logičkom smislu bit 1 predstavlja stanje istinitosti (eng. true), a bit 0 stanje neistinitosti (eng. false). U tehničkom smislu 1 predstavlja postojanje signala (signalnog napona), a 0 nepostojanje signala.

### NE (NOT)

Logički operator NE (NOT) predstavlja jednostavnu negaciju varijable:

$$NE A = B = \bar{A}$$

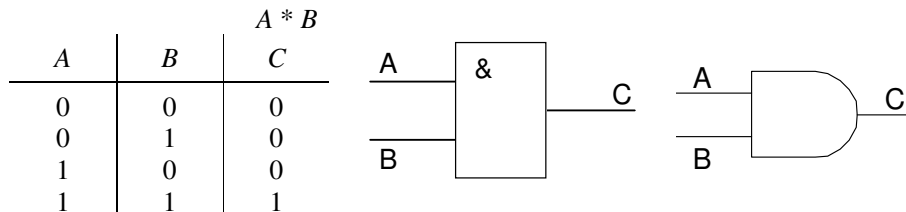
Ako je  $A = 1$ , odnosno "istina", onda je  $\bar{A} = 0$ , i obrnuto. Sl. 2-7 prikazuje tablicu istine, koja sadrži vrijednosti operatora, i njegov simbol koji se koristi u logičkim shemama.



Sl. 2-7 Tablica istine i simboli NE (NOT) operatora

### I (AND)

Operator I (AND) iskazuje istinitost (1) ako su obje ulazne varijable također istinite (Sl. 2-8).



Sl. 2-8 Tablica istine i simboli I (AND) operatora

U programskoj logici I operator se koristi u kondicionalnoj strukturi:

```
if (A and B) then C    //ako su A i B istiniti tada je i C
istinit
```

Može poslužiti za "maskiranje" binarnih slogova, npr.:

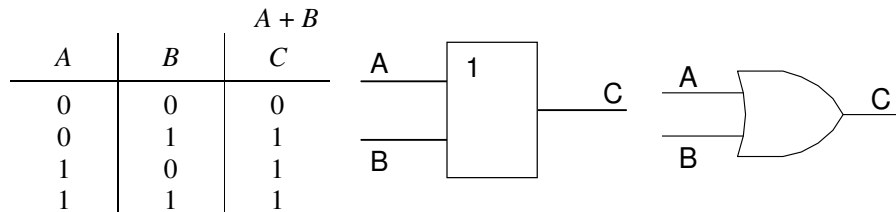
```
00001111
I(AND) 10101010
00001010
```

Prvi binarni slog 00001111 zove se maska. Primjenom I operatora i maske s nekim binarnim brojem dobit će se njegova replika, ali maskirana na mjestima gdje maska

ima nule. Maskiranje se koristi kada je potrebno reducirati ili lokalizirati dužinu binarnog sloga.

### ILI (OR)

Operator ILI je ispunjen uvijek kada je barem jedan ulazni operand jednak 1.



Sl. 2-9 Tablica istine i simboli ILI (OR) operatora

U programiranju ga upotrebljavamo na sljedeći način:

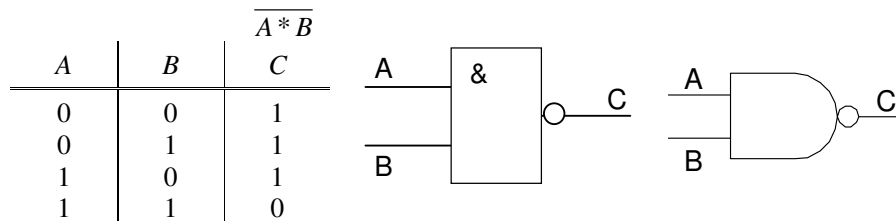
```
if (A or B) then C      //ako su A ili B istiniti tada je i C
istinit
```

Primjenom ILI operanda brojevi se maskiraju znamenkama 1:

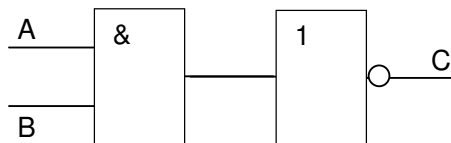
```
11110000
ILI(OR) 10101010
11111010
```

### NI (NAND)

Operator NI predstavlja negaciju I operatora (Sl. 2-10). Može se izraziti kao kombinacija I i NE operatora te se općenito ne smatra osnovnim operatorom.



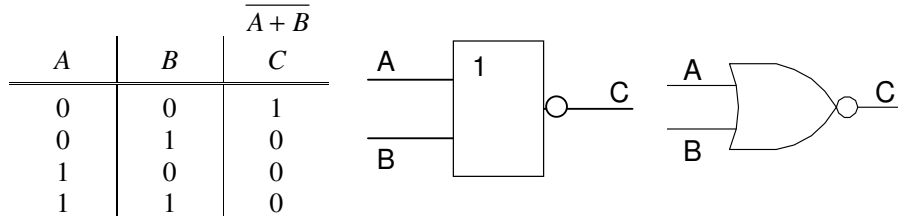
Sl. 2-10 Tablica istine i simbol NI (NAND) operatora



Sl. 2-11 Operator NI kao kombinacija I i NE operatora

**NILI (NOR)**

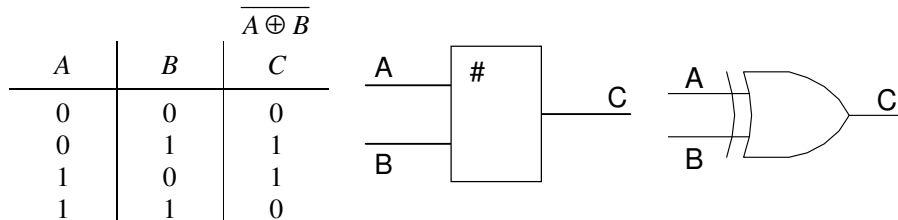
Operator NILI predstavlja negaciju ILI operatora (Sl. 2-12).



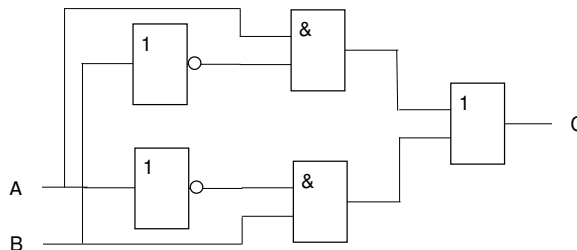
Sl. 2-12 Tablica istine i simbol NILI (NOR) operatora

**EX-ILI (EX-OR)**

Osnovno pravilo Ex-ILI operatora slijedi logiku običnog ILI, rezultat je istinit ako je istinit barem jedan operand, ali s tom razlikom da ne vrijedi slučaj kada su oba operanda istinita (Sl. 2-13). Zato se zove "ekskluzivno ili", dakle samo "ili-ili" (ili A ili B), bez "i" kombinacije (i A i B). Kao što je prikazano na Sl. 2-14, može se izraziti kombinacijom elementarnih operatora I, ILI i NE.



Sl. 2-13 Tablica istine i simbol Ex-ILI (EX-OR) operatora



Sl. 2-14 Logička shema operatora Ex-ILI pomoću elementarnih Booleovih operatora

Tipičnu primjenu operatora Ex-ILI nalazimo u komplementiranju binarnih brojeva. Ako binarni slog dovedemo u odnos sa slogom jedinica putem Ex-ILI operatora, rezultat će biti njegov komplement, npr.:

$$\begin{array}{r} 11111111 \\ \text{Ex-ILI(EX-OR)} \ 10101010 \\ \hline 01010101 \end{array}$$

### PRIMJENA LOGIČKIH OPERACIJA

U nastavku će se pokazati kako se osnovne logičke operacije koriste u realizaciji računarskih operacija na primjeru zbrajanja, memorije i brojanja.

#### Zbrajanje

Promotrimo primjer zbrajanja dvaju binarnih brojeva, ali počevši od najjednostavnijeg slučaja, od zbrajanja pojedinačnih bitova:

$$\begin{array}{r} A \quad 0 \quad \quad 0 \quad \quad 1 \quad \quad 1 \\ B \quad +0 \quad \quad +1 \quad \quad +0 \quad \quad +1 \\ \hline C \quad 0 \quad \quad 1 \quad \quad 1 \quad \quad (1)0 \end{array}$$

Ako zanemarimo prijenos u slučaju zbrajanja (1+1), možemo uočiti da se logika zbrajanja poklapa s Ex-ILI operatorom. Da bismo osigurali prijenos, moramo predvidjeti dva bita na izlazu logičkog sklopa za zbrajanje:

$$\begin{array}{r} A \quad 0 \quad \quad 0 \quad \quad 1 \quad \quad 1 \\ B \quad +0 \quad \quad +1 \quad \quad +0 \quad \quad +1 \\ \hline C \quad 00 \quad \quad 01 \quad \quad 01 \quad \quad 10 \end{array}$$

Odgovarajuća tablica istine prikazana je u Tab. 2-6.

Tab. 2-6 Tablica istine za zbrajanje binarnih brojeva dužine jednog bita s prijenosom

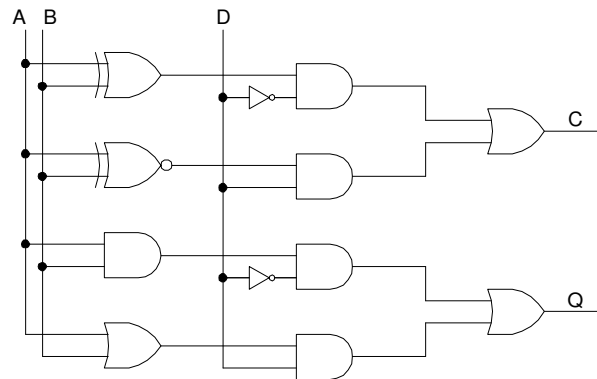
A	B	A + B	
		C	Q
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Iz Tab. 2-6 vidimo da rezultat C odgovara Ex-ILI logici, a P slijedi logiku I operatora. Prijenos ima očito utjecaj na rezultat samo kod zbrajanja višebitnih slogova. Tada moramo uključiti u razmatranje "donos". Dopunjena tablica istine, koja uključuje i donos prikazana je u Tab. 2-7. Takav logički sklop naziva se "puni sumator".

Tab. 2-7 Tablica istine za zbrajanje s prijenosom i donosom.

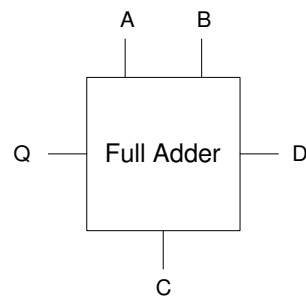
$D$	$A$	$B$	$A + B$	
			$C$	$Q$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Ima više načina logičke realizacije punog sumatora. Na Sl. 2-15 prikazana je logička shema jedne interpretacije sumatora. Temelji se na činjenici da prva četiri retka u tablici istine slijede logiku Ex-ILI operatora, s obzirom na  $A$ ,  $B$  i  $C$  ulaze/izlaze, dok preostala četiri retka odgovaraju logici NEx-ILI operatora (negacija Ex-ILI).

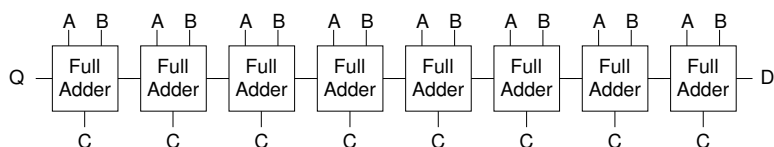


Sl. 2-15 Logička shema sumatora

Sumator predstavlja jednu od osnovnih računarskih operacija te se radi jednostavnosti prikazuje simbolom kao na Sl. 2-16. Ovakav jednobitni puni sumator može se kombinirati u nizove za zbrajanje višebitnih binarnih slogova. Ukoliko se želi realizirati 8-bitni sumator dovoljno je povezati u seriju 8 jednobitnih sumatora (Sl. 2-17).



Sl. 2-16 Sumator

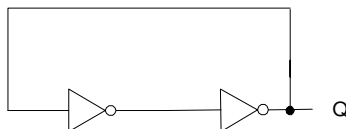


Sl. 2-17 8-bitni sumator

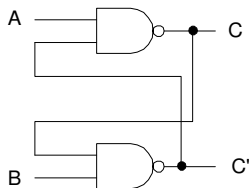
Prikazani sumator se naziva "ripple-carry adder" (eng. ripple - žuboriti, čavrljati, valjati). Naziv dolazi od principa prenošenja prijenosa od sumatora do sumatora u valovima. Prednost mu je logička jednostavnost, a mana sporost. Naime, u elektroničkom krugu svaki prijenos traje radnu periodu koja se mjeri nanosekundama. U slučaju zbrajanja slogova od 32 ili 64 bita zbrajanje može potrajati 100-200 ns. Kod brzih računala to je preveliki utrošak vremena za tako jednostavnu operaciju. Zato se u praksi koriste sumatori daleko složenije logike, koji su mnogo učinkovitiji.

### "FLIP-FLOP" MEMORIJA

Računalna se memorija temelji na konceptu povratne veze (eng. feedback). Povratna veza podrazumijeva princip po kojemu izlazni signal ujedno predstavlja jedan od ulaznih signala (Sl. 2-18).



Sl. 2-18 Logički sklop s povratnom vezom



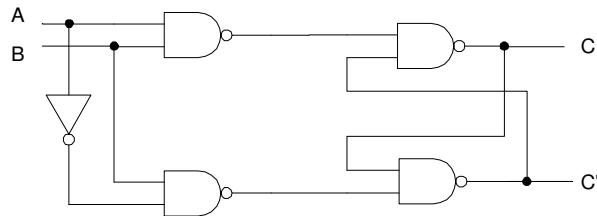
Sl. 2-19 Jednostavan memorijski sklop

Jednostavan memorijski logički sklop s povratnom vezom prikazan je na Sl. 2-19. Sklop je sačinjen od dva NI operatora. Svaki operator ima po jedan otvoreni ulaz i jedan izlaz. Pomoću povratne veze izlazi međusobno povezuju NI operatore. Ako pogledamo tablicu istine (Tab. 2-8) možemo opaziti da izlaz C uvijek odgovara vrijednosti na ulazu B, dok izlaz C' predstavlja inverziju od C. Za  $A=B=1$  vrijednosti na izlazima odražavat će zadnje stanje, odnosno pamtit će se zadnje vrijednosti na izlazima za koje je vrijedilo  $A \neq B$ .

Tab. 2-8 Tablica istine za logički sklop sa Sl.

$A$	$B$	$C$	$C'$
0	1	1	0
1	0	0	1
1	1	pamti	
0	0	nedefinirano	

Problem ovog jednostavnog sklopa za pamćenje predstavlja slučaj kada se na oba ulaza  $A$  i  $B$  pojavi vrijednost 0. Tada izlazi na sklopu sa Sl. 2-19 poprimaju vrijednost 1, što predstavlja nedefinirani slučaj. Da bi se to izbjeglo, memorijski sklop mora imati strukturu kao što je prikazana na Sl. 2-20.



Sl. 2-20 Flip-flop memorija

Takav se memorijski sklop naziva flip-flop memorija. Podatak koji se želi pohraniti zadaje se kao ulazni signal  $A$ , dok ulaz  $B$  služi za uključivanje i isključivanje memorije. Ako je  $B$  jednak 1, izlaz  $C$  će odražavati podatak zadan na ulazu  $A$ . Ako je  $B$  jednak 0,  $C$  će pamtitu zadnju vrijednost sa ulaza  $A$ .

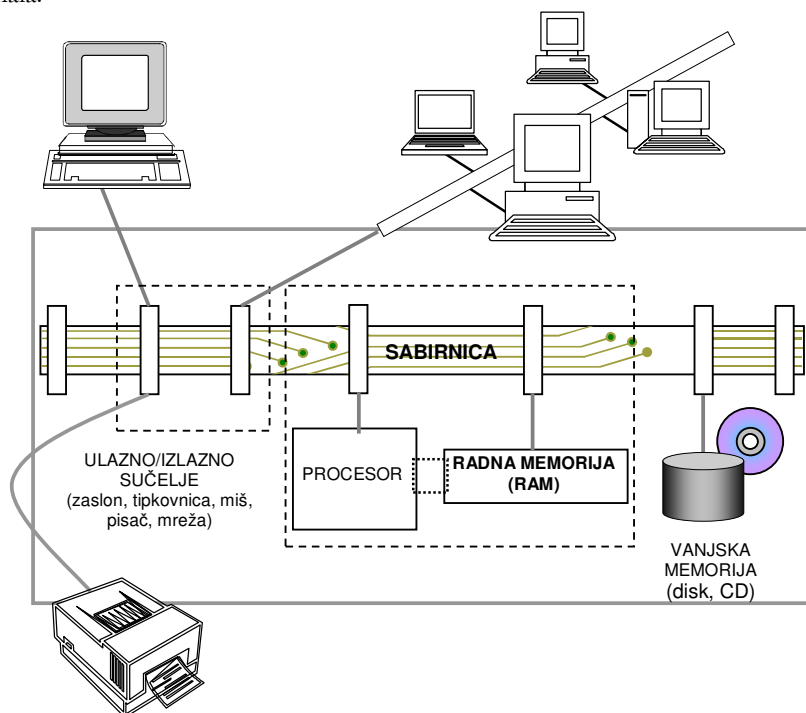
Daljnijim kombinacijama osnovnih logičkih sklopova, memorija, zbrajala i slično, moguće je implementirati vrlo složene matematičko-logičke operacije. S obzirom da se logički sklopovi temelje na binarnoj logici: uključeno-isključeno, njihova tehnička realizacija svodi se na prekidačke elemente koji se danas izvode pomoću poluvodičke tehnologije.





## 3 GRAĐA RAČUNALA

Izučavanjem temeljnih principa računalne tehnologije u prethodnom poglavlju, pokazano je da su osnovni čimbenici arhitekture računala memorija i središnji procesor. Dakako, oni tek sa ostalim pratećim komponentama tvore računalo kao funkcionalnu cjelinu. Građa modernog računala izrazito je složena i nije nam namjera upuštati se u detaljno izučavanje kompleksnog računalnog sklopovlja (hardvera, eng. hardware). Pokušat ćemo dokučiti tajne onih najvažnijih komponenata. Osim memorije i procesora, potrebno je još izdvojiti ulazno/izlazne uređaje koji nam omogućavaju unos podataka te praćenje i čitanje rezultata obrade istih. Putem posebnih ulazno izlaznih sučelja ostvaruje se i komunikacija s drugim računalima, odnosno umrežavanje računala. O radu pojedinih komponenata brinu upravljačke jedinice, koje su preko sabirnice međusobno povezane. Sabirnica stoga predstavlja kičmu svakog računala. Na Sl. 3-1 shematski je prikazana osnovna građa suvremenog računala.



Sl. 3-1 Shema građe računala

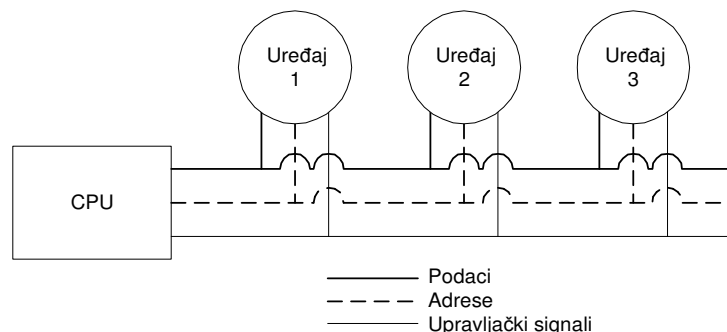


### 3.1 Sabirnica

U računalima se komunikacija između procesora, memorije i upravljačkih jedinica ostalih komponenata ostvaruje putem sabirnice (eng. "Bus"), povezujući ih u funkcionalnu cjelinu.

Sabirnica je snop provodnika koji ostvaruju prijenos binarnih podataka. Svaki je provodnik nosilac jednog bita. Sabirnica prenosi istovremeno cijeli slog (kombinaciju binarnih brojeva), čiji sadržaj može biti podatak o vrijednosti neke varijable, adresa, instrukcija i slično.

Sastoji se od tri dijela: jedna skupina provodnika služi za prijenos podataka, druga adresa i treća upravljačkih signala (Sl. 3-2). Osnovna dužina sloga iznosi osam bitova, a naziva se bajt ili oktava (eng. "byte"). Dva bajta nazivaju se riječ. Suvremena računala omogućavaju istovremenu razmjenu slogova dužine 32, 64, 128 i više bitova.



Sl. 3-2 Struktura sabirnice

Sabirnica ujedno predstavlja standardno komunikacijsko i radno sučelje za različite komponente. Na temelju tog koncepta danas je moguće fleksibilno konfiguriranje i dogradnja računala ovisno o potrebama. Dovoljno je na sabirnicu priključiti potrebne komponente, npr. grafički ubrzivač, video upravljačku jedinku i slično, ili ih jednostavno zamijeniti modernijim i bržim. Na taj način isto računalo može poslužiti različitim namjenama. Pritom se mogu koristiti komponente različitih proizvođača čija su sučelja usklađena posebnim normama (npr. S100, PCI, VESA).

Različite komponente računala zahtijevaju različitu brzinu prijenosa i obim podataka. Svaka sabirnica osigurava određeni kapacitet protoka podataka na temelju svoje širine (broja bitova koje može provoditi istovremeno) i frekvencije radne periode (vremena koje protekne između dva različita signala koji prenose binarne podatke). Posebno je kritičan kapacitet razmjene podataka između središnjeg procesora i memorije, dok npr. neke druge komponente (npr. disk) zbog svojih tehničkih ograničenja ne mogu obraditi tako veliki obim podataka u jedinici vremena. Stoga, u arhitekturi suvremenih računala često možemo uočiti rješenja kojima se zaobilaze standardne sabirnice i na kritičnim komunikacijskim vezama osiguravaju posebno veliki kapaciteti. U računalima koja danas postaju sve više intuitivna, temeljena na procesiranju grafičkih, video i zvučnih informacija, posebno je značajna sposobnost računala da u kratkom vremenu razmjeni izvanredno veliku količinu podataka između pojedinih komponenata (od više

stotina milijuna do nekoliko milijardi bajtova u sekundi, 1 milijuna bajtova = 1 megabajt, MB/s, 1 milijarda bajtova = 1 gigabajt, GB).



## 3.2 Memorija

Memorija je dio računala koji služi za pohranu podataka. U suvremenim računalima razlikujemo više vrsta i namjena memorijskih sustava: glavna ili radna memorija i stalna memorija.



### 3.2.1 Radna memorija

Glavna ili radna memorija namijenjena je pohrani računalnih programa i/ili podataka koji se trenutno izvode i/ili obrađuju (von Neumannov<sup>1</sup> koncept - ista memorija za program i podatke). Program podrazumijeva algoritam sačinjen od niza binarno kodiranih instrukcija koje razumije središnji procesor. Pohrana podataka je organizirana u slogovima čija dužina je obično osam bita (8 bitova = 1 oktava ili 1 byte). Ukoliko je za pohranu jednog podatka potrebno više od 8 bitova, slog se proširuje na više oktava. Dužina sloga za pohranu jednog podatka nije neograničena, a ovisi o procesoru, programskom jeziku i operacijskom sustavu. Simbolički se podaci pohranjuju kao 7-bitni ili 8-bitni slogovi (ASCII). Numerički podaci najčešće koriste više oktava, npr. 32 bita (dvije riječi<sup>2</sup>), što omogućava pohranu cjelobrojnih vrijednosti od -2,147,483,647 do +2,147,483,647 (to je  $2^{31}$  prethodeći s bitom predznaka), ili racionalnih vrijednosti od  $-3,4 \cdot 10^{38}$  do  $+3,4 \cdot 10^{38}$  (bit predznaka, bit predznaka eksponenta, 6 bitova za eksponent i 24 za mantisu). U slučajevima kada je potrebno pohraniti izrazito velike brojeve koristi se alokacija dvostruke točnosti (eng. "Double-Precision") od 64 bita, što omogućava pohranu vrijednosti u rasponu od  $-10 \cdot 10^{308}$  do  $+10 \cdot 10^{308}$  (9 bitova za eksponent).

Memorija je zamišljena, po uzoru na Turingov stroj, kao vrpca podijeljena na osambitne slogove (spremnike). Svakom spremniku pripada jedinstvena adresa, od 0 do više desetaka ili stotina milijuna, ovisno o veličini memorije računala. Danas računala zahtijevaju kapacitete od desetak do više stotina megabajta, a superračunala i više gigabajta. Pored kapaciteta, važna značajka memorije je brzina koja se izražava kroz vrijeme pristupa i ciklus memoriranja.

**Vrijeme pristupa** je vremenski interval koji protekne od trenutka izdavanja zahtjeva za čitanje do trenutka kada se traženi podatak pojavi na izlazu iz memorije.

**Ciklus memoriranja** je definiran najmanjim vremenskim intervalom između čitanja i ponovnog upisa.

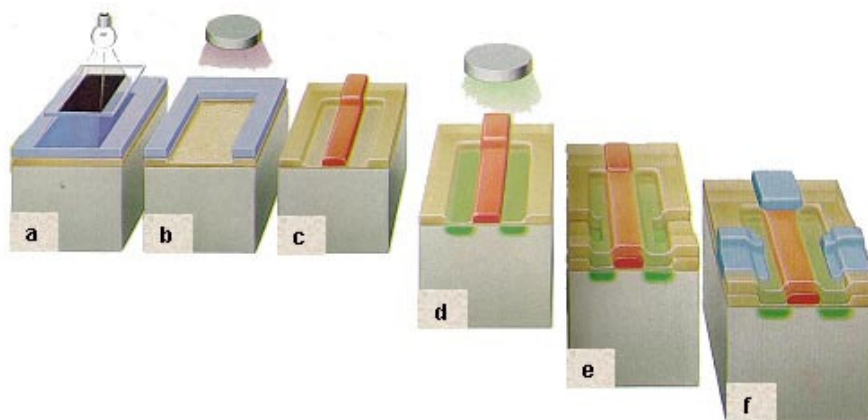


Radom memorije upravlja poseban sklop (upravljačka jedinka memorije, kontroler memorije), koji se brine da se svaki zaprimljeni podatak upiše na određenu adresu, kao i da se svaki traženi podatak pročita s odgovarajuće adrese. S obzirom da se podacima u takvoj memoriji može putem adrese pristupiti izravno i da ih se može obrađivati u bilo kojem redosljedu, radna se memorija još naziva "memorija sa slučajnim pristupom" ili RAM (eng. "Random Access Memory").

<sup>1</sup> Smatra se da je taj koncept prvi razvio J. P. Eckert i da je nepravredno pripisan von Neumannu.

<sup>2</sup> 16 bitova = 1 riječ ili 1 "word"

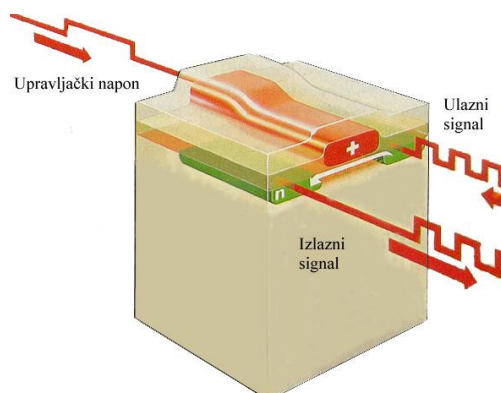
Memorija je načinjena od poluvodičkih integriranih krugova - pločica (eng. "chip" - u slobodnom prijevodu znači listić, npr. prženog krumpira, ili iverak). Poluvodički integrirani krug može sadržavati više milijuna memorijskih spremnika. Izrađuje se od izreska poluvodičkog (silicijskog) kristala, obično promjera 50 mm i debljine manje od 0.2 mm, na kojemu se dobije više tisuća memorijskih "chipova". Naziv poluvodič dolazi od električkih svojstava silicija i sličnih materijala čiji je specifični električki otpor ( $\rho = 3 \cdot 10^5 \Omega/\text{cm}$ , pri temperaturi od  $27^\circ\text{C}$ ) između specifičnog otpora izolatora (npr. staklo  $\rho = 2 \cdot 10^{13} \Omega/\text{cm}$ ) i vodiča (npr. bakra  $\rho = 1.7 \cdot 10^{-6} \Omega/\text{cm}$ ).



Sl. 3-3 Postupak izradbe poluvodičkih komponenata

Postupak izradbe prikazan je na Sl. 3-3. Izrezak se prvo izolira tankim oksidnim slojem i potom presvlači tankim slojem mekane foto-osjetljive plastike (Sl. 3-3.a). Površina plastike se prekriva neprovidnim uzorkom. Ultraljubičastim osvjetljavanjem otvrdnjava plastični sloj na providnim zonama i taj se postupak naziva *fotorezistentni postupak* (Sl. 3-3.a). Pomoću kiselina i otapala uklanja se plastični sloj i veći dio oksidnog sloja na neosvijetljenim mjestima (Sl. 3-3.b). Potom se ostatak plastičnog filma skida i nanosi se novi sloj silicija i oksida (Sl. 3-3.c). Na isti se način formiraju nove silicijske zone (crvena). Silicij se potom onečišćuje s različitim kemijskim dodacima (Sl. 3-3.d), stvarajući negativne i pozitivne električki provodne zone na neoksidiranim mjestima (označene crvenom i zelenom bojom). Daljnjim se postupkom mogu formirati višeslojne zone (Sl. 3-3.e), kao npr. pri izradbi mikroprocesora. Naparavanjem aluminijskih<sup>3</sup> provodnika stvaraju se električni kontakti (Sl. 3-3.f).

<sup>3</sup> U novije doba koriste se bakreni provodnici koji imaju bolja električka svojstva, ali je proizvodnja tehnološki složenija.



Sl. 3-4 Princip rada poluvodičke komponente

Kada pozitivni upravljački signal proteče polisilicijskim vratima (crvena zona), privlače se negativni elektroni ulaznog signala, tvoreći most između dviju odvojenih negativnih silicijskih zona. Na taj se način kod djelovanja upravljačkog signala premošćuje, odnosno uključuje, tok radnih impulsa. Kada se prekine upravljački signal počinje djelovati električki otpor koji prekida tok radnih impulsa, ostvarujući osnovni prekidački princip binarne logike (Sl. 3-4).

### 3.2.2 Stalna memorija

Stalna memorija (eng. "ROM - Read Only Memory") je posebna vrsta memorije koja se u pravilu može samo čitati i služi za trajnu pohranu podataka. Upis podataka se obavlja već u postupku izradbe ili pomoću posebnih uređaja. Služi za pohranu specijalnih podataka, obično permanentnih ili rijetko promjenjivih, npr. podaci o ugrađenim komponentama, lokaciji operacijskog sustava, kodnim tablicama i slično. Takvi se podaci moraju zaštititi od brisanja i interferencije s ostalim korisničkim operativnim podacima, i moraju ostati sačuvani i nakon gašenja računala. Izrađuju se u poluvodičkoj tehnologiji, a mogu se svrstati u tri kategorije: memorije programirane maskom (ROM), električki programirane memorije (PROM) i reprogramabilne memorije (EPROM).

Standardni ROM-ovi programiraju se posebnim maskama u proizvodnome postupku. U ovisnosti od primijenjene maske, koja određuje kombinaciju prespajanja vodiča bitova, bilo u diodnoj ili tranzistorskoj tehnici. Npr. u tranzistorskoj tehnici jetkanja uklanja se oksidni sloj na mjestima upravljačke elektrode gdje treba ostvariti vrijednost 1.

Električki programabilna memorija (eng. "PROM - Programmable ROM") izrađuje se "prazna" bez pohranjenih podataka. Svaki spremnik bitova sadrži prenosnik (diodu s rastalnim osiguračem ili elektrokemijski osjetljivu upravljačku elektrodu) koji se aktivira dovođenjem strujnog impulsa u skladu s zahtjevanom binarnom kombinacijom. Ovako pohranjeni podaci ne mogu se više mijenjati te ostaju stalno zapisani u PROM-u.



Programabilni ROM omogućava višekratni upis i brisanje podataka. Stoga se mogu višekratno koristiti za različite namjene. Imaju vrlo dugo vrijeme brisanja, a obavlja se neelektrički ozračivanjem ultraljubičastim ili rendgenskim zrakama. Programiranje se izvodi električki dovođenjem visokog napona (oko 80 V) između kolektora i emitera koji trebaju ostvariti vrijednost 1. Taj napon uzrokuje trajan naboj u polukristaličnoj silicijskoj upravljačkoj elektrodi, ostvarujući trajan strujni krug.

### 3.2.3 Vanjska memorija

Podaci koji se pohranjuju u radnu memoriju raspoloživi su samo tijekom svojeg operativnog ciklusa, odnosno dok novi operativni podaci ne zauzmu njihove lokacije. Kada se računalo isključi svi podaci pohranjeni u RAM nestaju. Za permanentnu pohranu podataka koriste se različiti oblici vanjske memorije: diskovi, diskete, CD-ovi (eng. "Compact Disk") i trake. Oni nam omogućavaju da potrebne podatke trajno pohranimo te po potrebi učitamo u radnu memoriju.

#### 3.2.3.1 Memorijski diskovi

Memorijski diskovi dijele se na tvrde diskove (eng. "hard disk") i diskete (eng. "floppy diskette"). Tvrde se diskove upotrebljavaju kao stabilne računalne komponente velikih kapaciteta, dok se diskete koriste kao prijenosni mediji.

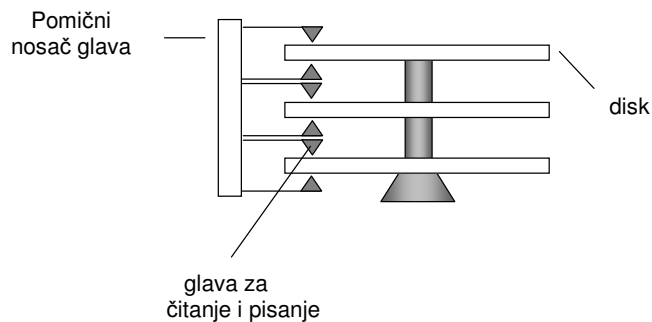
#### TVRDI DISK

Tvrđi disk (Sl. 3-5) je nezaobilazna komponenta većine računala. Može pohraniti veliku količinu podataka. Osim toga, osigurava veliku brzinu pristupa podacima te mogućnost razmjene (čitanja i pisanja) vrlo velike količine podataka u jedinici vremena. Okreće se brzinom od oko 3600 do 7200 okretaja u minuti<sup>4</sup>, s brzinom pristupa od 10 - 30 ms. Pritom se ostvaruje protok podataka od 5 do 80 MB/s.



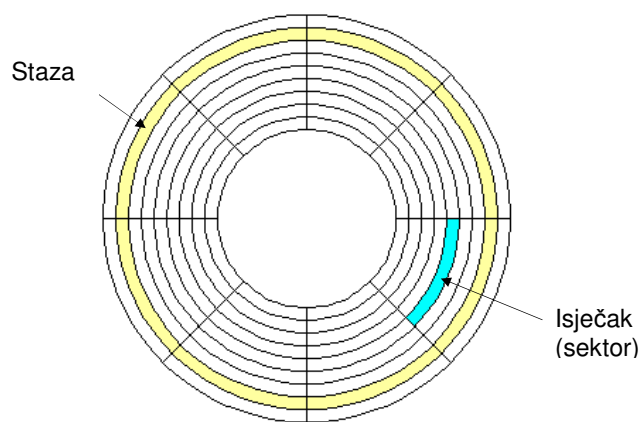
Sl. 3-5 Tvrđi disk

<sup>4</sup> Pri brzini od 7200 okretaja u minuti, obodna brzina na krajnjim stazama diska, ovisno o njegovom promjeru, može biti više desetaka km/h.



Sl. 3-6 Shema tvrdog diska

Disk je u osnovi sačinjen od više diskova, odnosno okruglih ploča (Sl. 3-6), najčešće promjera  $5\frac{1}{4}$ " ili  $3\frac{1}{2}$ ". Ploče su sa svake strane presvučene magnetnim slojem željeznog oksida, slično kao na magnetnoj (npr. audio) kaseti. Taj se magnetni sloj na istom principu polarizira pomoću glave za čitanje i pisanje, ali pohranjujući tek dvije vrste signala, od kojih je jedan nosilac vrijednosti jedan, a drugi ničice. Na svakoj strani ploče nalazi se po jedna glava za čitanje i pisanje. Glave se kreću radijalno u odnosu na rotirajuće diskove, svaka na udaljenosti od svega nekoliko desetaka  $\mu\text{m}$  od površine diska ne dodirujući ga. Zato ih zovu "plivajuće glave". Format zapisa organiziran je u koncentričnim krugovima - stazama. Staze su podijeljene na isječke (eng. sectors) i blokove. Isječci (sektori) su najčešće dimenzionirani tako da sadrže 256 ili 512 bajtova (bytes, osmina). Formatiranje blokova ovisi o interpretaciji zapisničkog sustava, a predstavlja kontinuirani zapis niza bitova, obično duljine 512 ili 1024 bajtova.



Sl. 3-7 Format zapisa na disku

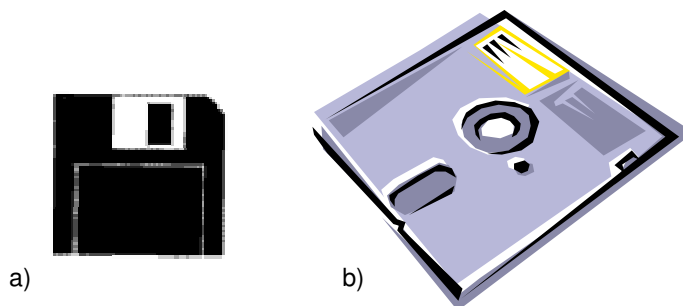
Pristup pojedinim blokovima ostvaruje se putem odgovarajućih adresa koje su pohranjene u alokacijskim tablicama, kreiranim na disku u skladu s formatom

zapisničkog sustava. To ujedno znači da je moguće izravno pristupiti traženim podacima, bez obzira gdje su fizički pohranjeni na disku.

Tehničke značajke tvrdih diskova kontinuirano se unapređuju, prvenstveno u pogledu kapaciteta i brzine. Kapaciteti suvremenih diskova najčešće se kreću oko više gigabajta. U posebnim primjenama, gdje se obrađuju velike baze podataka, ukupni diskovni kapaciteti mjere se u terabajtima.

#### DISKETA

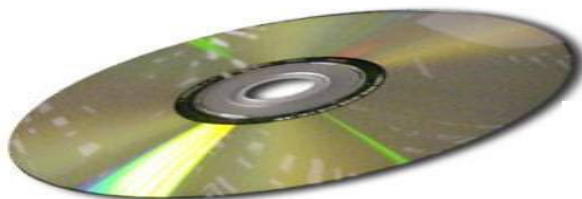
Disketa je, za razliku od tvrdog diska, predviđena kao prijenosni medij. Načinjena je od tanke plastične folije kružnog oblika, presvučene tankim magnetnim slojem i zaštićene papirnatom košuljicom (5¼") ili plastičnom kutijicom (3½",). Zato se još naziva mekani disk (eng. "floppy disk"). S obzirom na tehničke značajke, kapacitet disketa iznosi najviše 2 MB, a s brzinom od 300 okretaja u minuti ostvaruje brzinu pristupa do 300 ms. Te su značajke znatno skromnije u usporedbi s tvrdim diskom te se disketa rijetko koristi kao operativni medij. Prvenstveno služi kao osobni prenosivi medij za pohranu ili distribuciju podataka.



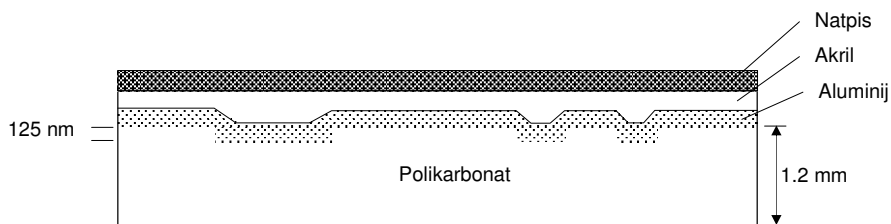
Sl. 3-8 Diskete, (a) 3.5", (b) 5.25"

#### CD / DVD

U nastojanju da se objedine standardi za nosače različitih vrsta informacija razvijen je tzv. **kompaktni disk ili CD** (eng. "Compact Disk", Sl. 3-9). CD tehnologija temelji se na svjetlosnom principu. Jednako se primjenjuje za pohranu audio, video i računalnih podataka. Disk je izrađen od polikarbonske plastike, promjera 120 mm i debljine 1.2 mm. S jedne je strane prekriven reflektirajućim materijalom, aluminijem zaštićenim akrilnim filmom (Sl. 3-10). Podaci se zapisuju kreiranjem spiralnih nizova (Sl. 3-11) reflektirajućih polja, a očitavaju se putem laserskog snopa.

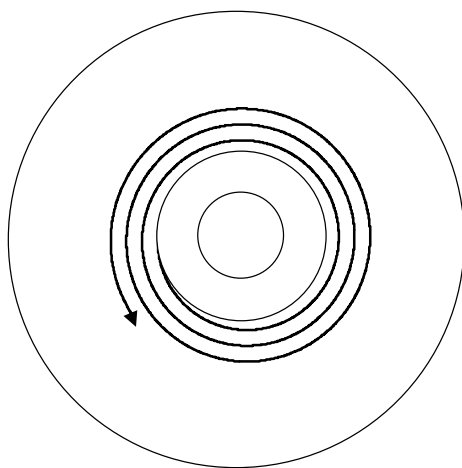


Sl. 3-9 Kompaktni disk - CD



Sl. 3-10 Presjek CD-a

Na polikarbonatu se oblikuju udubine (ili obrnuto uzvišenja, ako se promatraju sa strane laserskog čitača), najmanje  $0.97 \mu\text{m}$  dugačka i  $125 \text{ nm}$  visoka<sup>5</sup>, koje rezultiraju različitim stupnjem refleksije kada se prevuku aluminijskim filmom. Optoelektrički senzor očitava promjene u refleksiji koje se potom interpretiraju kao bitovi, odnosno binarni slogovi. Refleksija se interpretira kao znak 1, a izostanak refleksije znači 0. Spiralni zapis započinje iz unutrašnjeg polumjera diska, šireći se prema van, tj. prema vanjskom polumjeru diska. Stoga disk može po potrebi biti manjeg promjera od 12 cm. Spiralna je staza široka svega  $0.5 \mu\text{m}$ , a razmak između staza iznosi  $1.6 \mu\text{m}$ <sup>6</sup>. CD uređaj okreće disk standardnom brzinom od 500 – 200 okretaja u minuti. Brzina se smanjuje proporcionalno udaljenosti laserske glave od središta vrtnje (prema vanjskom obodu diska), osiguravajući kontinuiranu relativnu obodnu brzinu staze u odnosu na laserski čitač. To ujedno znači ujednačeno uzorkovanje podataka u jedinici vremena. Standardna jednostruka (1x) brzina CD-a ostvaruje podatkovni kapacitet od 150 KB/s.



Sl. 3-11 Spiralni trag zapisa na CD-u

<sup>5</sup> Na strani laserske glave koja očitava reflektirajuća polja na CD-u vidljiva su uzvišenja (eng. "bumps"), no ukoliko se gleda s druge strane diska onda vidimo udubine ili jamice (eng. "pits").

<sup>6</sup> Ukupna dužina spiralne staze na CD-u iznosi skoro 8 km.

Najčešće je u upotrebi CD s tvornički snimljenim podacima, koji se ne daju mijenjati, te se stoga takav medij naziva CD-ROM, odnosno CD sa kojega se može samo čitati. Na njega se može standardno pohraniti 600 MB podataka.

Raširena je također upotreba CD uređaja i medija koji omogućavaju pisanje (CD-R<sup>7</sup>) ili pisanje i brisanje (CD-RW<sup>8</sup>) CD-a. CD-R tehnologija koristi poseban (oksidni) premaz staza koji mijenja boju na mjestu osvjetljavanja (progrijavanja) laserskom zrakom za pisanje (jačeg intenziteta od zrake za čitanje). Na mjestima promijenjene boje mijenja se svojstvo refleksije (smanjuje se odraz). Upotreba CD-RW tehnologije jednostavna je kao i upotreba diskete, ali omogućava daleko veće brzine i kapacitet pohrane podataka. Poseban sloj materijala (kombinacija srebra, antimonija, telurijuma i indijuma) može mijenjati svojstva promjenom snage progrijavanja laserskim snopom. Na praznom disku taj je sloj u kristalnom stanju potpuno proziran te propušta svjetlosne zrake do reflektirajućeg aluminijskog sloja. Snažan laserski snop za snimanje zagrijava ga do temperature tališta (oko 600°C). Na protaljenim mjestima materijal postaje zamućen (amorfan), sprečavajući refleksiju laserskog snopa (pohranjujući 0 na tim mjestima). Umjerenom snagom (zagrijavanjem do 200°C) postiže se kristalizacija zamućenih područja koja ponovno propuštaju svjetlo. Na taj se način disk briše, tj. priprema za ponovno upisivanje.

**DVD<sup>9</sup>** (eng. "Digital Versatile Disk" - digitalni raznovrsni disk) standard objedinjuje DVD-audio i DVD-video specifikacije za masovnu distribuciju filmova i muzike, zatim DVD-ROM za distribuciju programa i multimedijalnih sadržaja, DVD-RAM za opću namjenu čitanja i pisanja podataka, DVD-R za jednokratnu pohranu podataka i DVD-RW za višekratnu pohranu podataka. Sve specifikacije predviđaju medij istih dimenzija kao i CD format. Razlike u odnosu na CD nevidljive su za korisnika, a poglavito se odnose na novu optičku tehnologiju, obradu signala, tehnologiju izradbe diskova i novi zapisnički sustav UDF (eng. "Universal Disk Format"). Kao rezultat dobiven je veći kapacitet za pohranu podataka, 4.7 GB na svakoj strani diska (9.4 GB ukupno) i veća brzina pristupa podacima. DVD pohranjuje podatke isto kao i CD, u obliku jamica udubljenih na površini diska, samo što je u slučaju DVD-a gustoća zapisa oko 4.5 puta veća. Razmak između tragova staze iznosi 740 nm, što omogućava pohranu zapisa dugog skoro 10 m na samo jednom sloju.

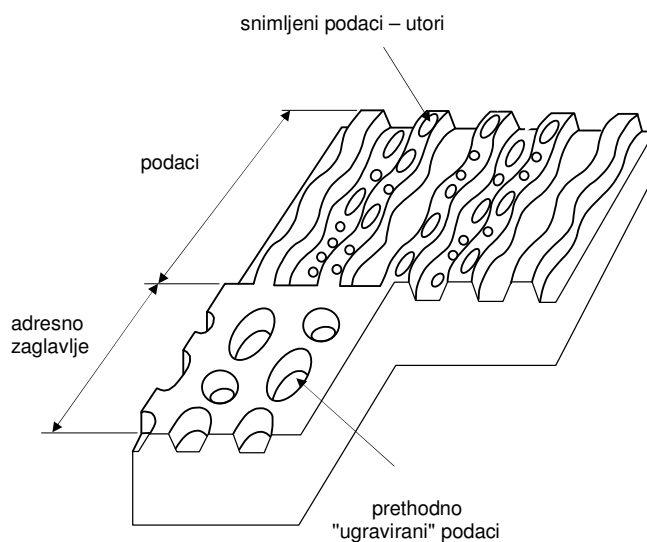
Podaci se logički adresiraju u isječcima (sektorima) veličine 2KB, kojima se pristupa putem ECC blokova (eng. "ECC - Error-Correcting Code") od 32 KB (Sl. 3-12). Podaci su na disku položeni u spiralnoj formi kao i na CD-u. Svaki isječak prethodi prethodno gravirano zaglavlje pomoću kojega se vrši identifikacija isječka. Dok uređaj čita unutar jednog područja, disk se okreće stalnom brzinom koja odgovara polumjeru staze na kojoj se područje nalazi (10.5 – 25.5 okretaja u sekundi). Kada DVD uređaj prelazi na čitanje u drugom području, mijenja se kutna brzina u skladu s novim polumjerom, održavajući protok podataka kontinuiranim (3.49 – 3.84 m/s). Podatkovni kapacitet DVD-a pri jednostrukoj brzini iznosi 1250 KB/s.

---

<sup>7</sup> CD-R(ecord) – oznaka za CD po kojem se jednokratno mogu zapisivati (eng. "Record") podaci pomoću CD-R ili CD-RW uređaja.

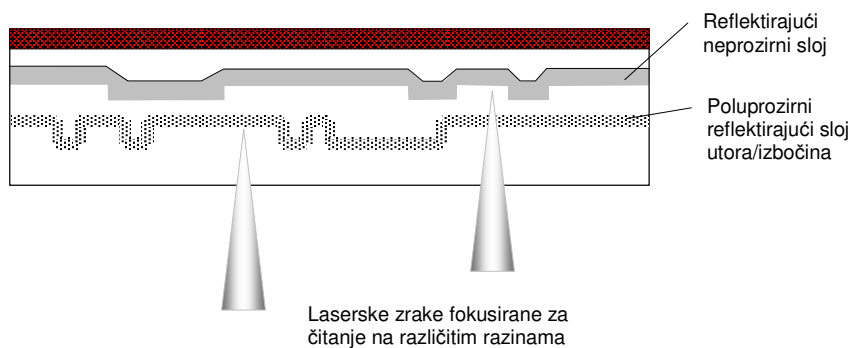
<sup>8</sup> CD-R(e)W(rite) – oznaka za CD po kojemu se može višekratno pisati i brisati (eng. "Rewrite") pomoću CD-R ili CD-RW uređaja.

<sup>9</sup> DVD format je objavljen 1995. godine, nakon što je usklađeno dugogodišnje nadmetanje tehnoloških divova Sonya, Philipsa, Toshiba i Matsushite oko sljedeće generacije CD formata. Sony i Philips su prvi puta objavili novi CD format - MultiMedia Compact Disk 1994. godine, a Toshiba i Matsushita svoj Super Density format 1995. godine.



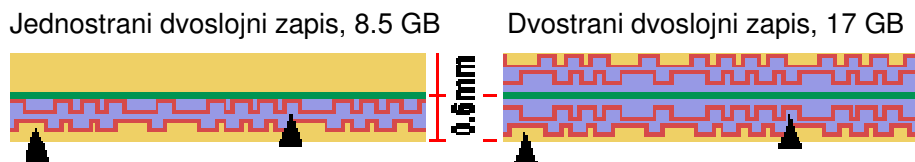
Sl. 3-12 Isječak DVD zapisa

Zahvaljujući IBM-ovoj "višeslojnoj disk tehnologiji", DVD omogućava čitanje podataka na dvije razine, bez potrebe preokretanja diska ili korištenja dvije čitače glave. Gornji sloj reflektirajućeg materijala je poluproziran i omogućava očitavanje zapisa sa drugog sloja, odnosno druge strane diska. Da ne bi došlo do interferencije zapisa na različitim slojevima, laserski snop se po potrebi fokusira na odgovarajući sloj, čitajući istovremeno podatke samo s jedne razine.



Sl. 3-13 Princip jednostranog čitanja višeslojnog DVD zapisa

Dvoslojna pohrana podataka primijenjena na obje strane diska udvostručava kapacitet DVD-a na 17 GB (Sl. 3-14).



Sl. 3-14 Dvoslojni zapis omogućava pohranu do 17 GB podataka

Uređaji za snimanje DVD-a (DVD-RAM, DVD-RW, DVD-R) rade na sličnom principu kao i oni za snimanje CD-a. Pritom se koriste laseri snage od 8 do 10 mW i valne dužine 635 ili 650 nm. U procesu snimanja računalo mora osigurati protok podataka od najmanje 11.08 Mbit/s. DVD-RAM medij predviđa 100000 presnimavanja.

### 3.2.3.2 Traka

Magnetna traka, uz bušene kartice, najstariji je medij za permanentnu pohranu velike količine računalnih podataka. Načinjena je od tanke plastične vrpce, prekrivene magnetnim slojem. Podaci se pomoću magnetne glave zapisuju sekvencijalno. To znači da im se ne može pristupiti izravno kao u slučaju diskova, jer ne postoji mogućnost da se utvrdi lokacija pojedinih podataka, osim slijednim pretraživanjem trake. Stoga je primjena trake spora u odnosu na druge medije. Ipak, i danas je u upotrebi jer je traka jedini prenosivi medij kapaciteta od više GB. Služi za pohranu velikih baza podataka i kao podrška za periodičnu pohranu diskovnih podataka radi sigurnosti u slučaju kvara koji bi mogao rezultirati trajnim gubitkom podataka.

Postoji više vrsta tračnih uređaja, no danas je najčešće u upotrebi DAT - Data Archive Tape standard.







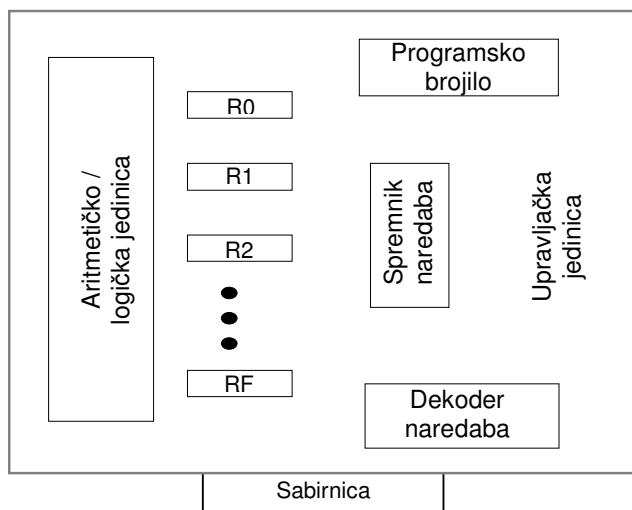
### 3.3 Procesor

Procesor (središnja jedinka, eng. "CPU - Central Processor Unit") je elektronički sklop koji ima sposobnost obrade podataka u skladu s programskim naredbama.



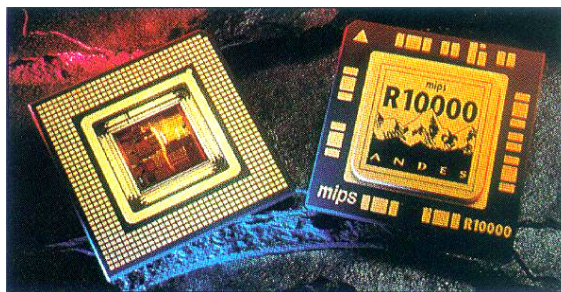
Sposoban je izvoditi osnovne aritmetičko/logičke operacije relativno jednostavnim transformacijama binarnih podataka. Svaka operacija se izvodi u jednom ili više ciklusa. Broj ciklusa koje procesor može izvršiti u jedinici vremena izravno određuje njegovu efikasnost. Naziva se frekvencija procesora ili vrijeme radne periode, koja se u današnjim tehnološkim uvjetima kreće od 200 do 1500 Mhz, što omogućava izvršavanje više stotina milijuna naredbi u jednoj sekundi (eng. "MIPS - Millions Instructions Per Second").

Suvremena računalna tehnologija temelji se na posebnoj vrsti procesora koji se nazivaju mikroprocesori. Mikroprocesor objedinjuje različite aritmetičko/logičke i upravljačke funkcije, koje su nekada izvodili odvojeni procesori. Sačinjen je od sabirnice, spremnika, upravljačke jedinice i aritmetičko/logičke (radne) jedinice. Na prikazana je tipična struktura jednog mikroprocesora.



Sl. 3-15 Struktura mikroprocesora

Izrađuje se u poluvodičkoj tehnologiji visokog stupnja integracije. Prosječni mikroprocesor sadrži više milijuna tranzistora integriranih na površini manjoj od kvadratnog centimetra. Veći dio gotovog procesora (skoro 90%), Sl. 3-16, čine spojevi, keramička zaštita i hladilo.



Sl. 3-16 Mikroprocesor "mips R10000"

Na sabirnicu su spojeni aluminijski provodnici preko kojih se procesor povezuje s radnom memorijom i ostalim komponentama, primajući ulazne (instrukcije, adrese i ostale podatke) i predajući obrađene informacije.

### 3.3.1 Spremnik

Spremnik služi za pohranu operanda koje aritmetičko/logička jedinka treba obraditi. Tu se zapisuju podaci preneseni iz glavne memorije, kao i rezultat njihove obrade. Dakle, u radnu se memoriju pohranjuju podaci sa diska koji će biti potrebni u procesu obrade, dok se u procesorski spremnik pohranjuju podaci iz radne memorije koji su trenutno u obradi. Rezultat, koji procesor zapiše u spremnik, pohranjuje se u radnu memoriju, da se oslobodi kapacitet za novi ciklus obrade, a potom i na disk, ako je rezultat potrebno trajno spremiti. O pohrani podataka u spremniku i aktiviranju aritmetičko/logičke jedinice brine upravljačka jedinka.

Prijenos podataka između spremnika i radne memorije zahtijeva izvjesno vrijeme (određeni broj ciklusa, taktova) koje značajno utječe na konačno vrijeme ukupnog ciklusa obrade. Moderni se procesori stoga dopunjuju posebnom vrstom spremnika (eng. "Cache Memory"), koji mogu prihvatiti veći obim podataka (32 - 256 KB), osiguravajući izravnu razmjenu podataka s procesorom i njegovim spremnicima.

Od suvremenih se računala ponajprije očekuje podrška obradi grafičkih i vizualnih podataka (grafičko korisničko sučelje, multimedija, prividna stvarnost), koji se u računalu interpretiraju jedva zamjetnim vizualnim elementima - točkama. Svaka točka, kao vizualni element određena je s najmanje tri podatka (x i y koordinate i podatak o boji). Za opis vizualnih podataka u rezoluciji 640x480 potrebno je definirati 307,200 točaka, a u rezoluciji 1280x1024 čak 1,310,720 točaka. To znači da za animaciju grafičkih elemenata i interakciju s geometrijskim objektima u realnom vremenu procesor mora obraditi više stotina milijuna podataka u sekundi. Stoga se često "Cache" memorija dopunjuje sa sekundarnom "Cache" memorijom, obima 512 KB - 2MB, poglavito na grafičkim radnim stanicama, omogućavajući da se jedan ili više grafičkih zapisa izravno pripremi za obradu. Veza između procesora i memorije jedan je od najosjetljivijih čimbenika u podizanju procesorske učinkovitosti. Zato moderne računalne arhitekture teže ostvarivanju svojevrsne integracije u pogledu procesorske i memorijske komunikacije.

### 3.3.2 Spremnik naredaba

U spremnik naredaba pohranjuju se programske naredbe koje nalažu procesoru što treba činiti s podacima iz spremnika. O slijedu tih naredaba skrbi programsko brojilo. Dekoder naredaba određuje zadaću koju procesor mora izvršiti. Postoje tri osnovne skupine naredaba za: prijenos podataka, aritmetičko/logičke operacije i upravljanje tijekom programa.

### 3.3.3 Aritmetičko/logička jedinka

Aritmetičko/logička jedinka izvršava osnovne matematičke operacije (pomoću kojih se nadalje izvode složene funkcije) i logičke operacije I, ILI (eng. "AND", "OR"). Te se operacije, u skladu s binarnom aritmetikom, uglavnom svode na jednostavne promjene bitova unutar pojedinih spremnika ili pak njihovo pomicanje ulijevo ili udesno (naredbe SHIFT ili ROTATE).

### 3.3.4 Upravljačka jedinka

Upravljačka jedinka koordinira radom procesora. Izvršava naredbe za prijenos podataka i upravljanje tijekom izvođenja programa.

U skupinu naredaba za prijenos podataka spadaju naredbe kao što su LOAD i STORE. Prva služi za učitavanje podataka iz radne memorije u procesorski spremnik, a druga obrnuto, za pohranu podataka iz spremnika u radnu memoriju. U ovu skupinu spadaju i naredbe za komunikaciju s ostalim komponentama računala. Ponekad se kategoriziraju kao zasebna skupina ulazno/izlaznih naredaba.

Skupina upravljačkih naredaba se brine za tijek izvođenja programa. One omogućavaju preskakanje programskih naredaba u nizu i preusmjeravanje izvršavanja na različitim mjestima u programu. Tu spadaju naredbe kao što su JUMP ili BRANCH. Pritom skok na neku od naredaba može biti bezuvjetan ili pak uvjetan, a to znači da se o tijeku izvođenja odlučuje na temelju zadanih uvjeta (npr. ako je A veće od 0 skok na naredbu br. 5, ako ne nastavi slijedom izvođenja).

### 3.3.5 Procesorske arhitekture

Kada je 1951. godine proizveden Univac kao prvo komercijalno računalo, bio je izgrađen uglavnom od vakuumskih elektronskih cijevi, koje su služile kao prekidački elementi. Postupno su zamjenjivane tranzistorima, a već 1959. godine u Fairchildu i Texas Instrumentsu proizvedeni su prvi poluvodički integrirani krugovi. Njihova integracija započinje od RTL komponenata (eng. "Resistor-Tranzistor Logic" - otpornik-tranzistor logika), a nastavlja se uvođenjem TTL komponenata (eng. "Transistor-Tranzistor Logic" - tranzistor-tranzistor logika), zatim slijede komponente TTL MSI (eng. "Medium-Scale Integration"), srednjeg stupnja integracije, LSI (eng. "Large-Scale Integration"), visokog stupnja integracije, i konačno VLSI (eng. "Very Large-Scale Integration"), vrlo visokog stupnja integracije. Integrirani krugovi izrađuju se u tehnologijama MOS (eng. "Metal-Oxyd Semiconductor" - metal-oksidni poluvodič), PMOS (s P kanalom), NMOS (s N kanalom) i danas uglavnom u CMOS (nadopunjujućoj) tehnologiji.

Od 1961 godine do 1971. godine broj tranzistora u "chipu" udvostručavao se iz godine u godinu. Intel je 1971. godine predstavio prvi komercijalni mikroprocesor

Intel 4004, koji je integrirao 2300 tranzistora i 45 naredbi, a već je godinu kasnije 8-bitni Intel 8008, s radnim taktom 0.5 MHz, sadržavao 3300 tranzistora.

### PRVA GENERACIJA

Prva generacija procesora izvodila je operacije slijedno. To znači prvo učitavanje prve naredbe iz spremnika, potom se ona dekodira i najzad izvodi. Kada je naredba izvršena, upravljačka jedinica pomiče kazalo na sljedeću naredbu, slijedi njeno učitavanje, dekodiranje i tako dalje sve do završetka programa (Sl. 3-17).



U – učitavanje naredbe, D – dekodiranje, I – izvođenje

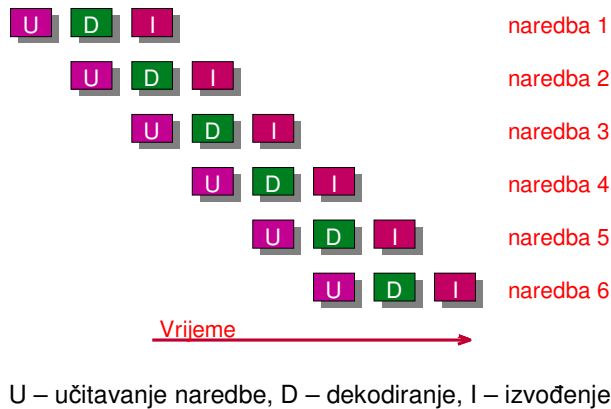
Sl. 3-17 Princip izvođenja naredbi prve generacije procesora

### DRUGA GENERACIJA

S razvojem 16-bitnih i ubrzo 32-bitnih procesora, krajem sedamdesetih i početkom osamdesetih godina, javlja se paralelizacija pojedinih procesorskih zadaća, uvođenjem paralelne protočnosti (eng. "Pipelining"). S obzirom da električki impulsi putuju kroz provodnike brzinom svjetlosti, najveća brzina razmjene podataka između procesora i memorije traje nekoliko nanosekundi ( $1 \text{ ns} = 1 \cdot 10^{-9} \text{ s}$ ), te sljedećih nekoliko nanosekundi treba za dekodiranje, izvođenje naredbe i pohranu rezultata. Jedan od načina ubrzavanja rada procesora je očito minijaturizacija, kojom se smanjuju udaljenosti između modula procesora i komponenata računala. Drugi je način povećanje protočnosti procesora ("pipelining"). Protočnost označava sposobnost procesora da obradi određeni broj naredbi u jedinici vremena, neovisno o njegovoj brzini izvođenja pojedine naredbe (Sl. 3-18). Naziv dolazi od engleske riječi "pipeline", što znači cjevovod, a označava princip nizanja naredbi (kao kroz cijev), pri čemu se na svakoj naredbi iz cjevovoda obavlja neka od operacija (učitavanja, dekodiranja ili izvođenja), zavisno o fazi obrade u kojoj se nalazi.

Najbolji je primjer druge generacije procesor Motorola MC68000. Na njemu je integrirano 68000 tranzistora i sa internom 32-bitnom arhitekturom ostvarivao je na 8 MHz 2 MIPS-a (eng. "Milion Instructions Per Second" - milijuna instrukcija u sekundi). Taj je procesor omogućavao istovremeno izvođenje, dekodiranje i učitavanje naredaba u slijedu (slika).

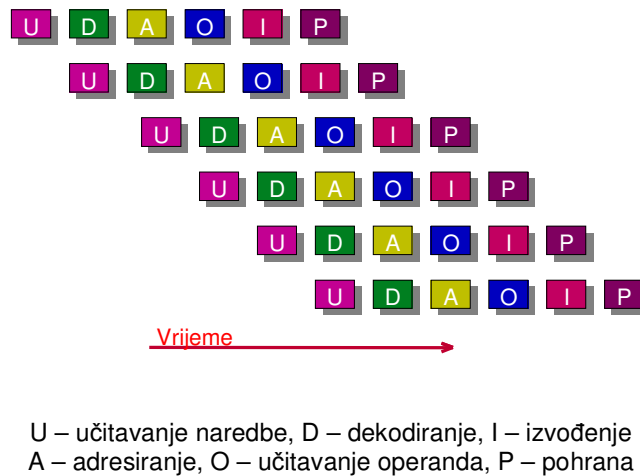
Učinkovitost druge generacije procesora pokrenula je ubrzani razvoj malih računala, ali i razvoj radnih stanica i osobnih računala (eng. "PC - Personal Computer").



Sl. 3-18 Princip rada druge generacije procesora

### TREĆA GENERACIJA

Treća generacija procesora uvodi primarni spremnik (eng. "Cache Memory") kao posebnu novinu. Primarni spremnik je integralni dio procesora, a omogućava pohranu većeg obima izravno dostupnih podataka. Na taj je način riješen glavni problem prethodne generacije procesora - zastoji u radu prilikom učitavanja operanda iz radne memorije, uzrokovani zato što se pristup operandima i naredbama izvodi intermitentno kroz istu sabirnicu. Pritom se proširuje dubina paralelizacije (protočnosti) i procesor istovremeno izvodi još i adresiranje, čitanje i pisanje operanda (Sl. 3-19). U treću generaciju procesora spada npr. Motorola 68020 (250 000 tranzistora), kao i prvi RISC procesori (eng. "Reduced Instruction Set Computer") koji su dali glavni zamah široj primjeni radnih stanica.



Sl. 3-19 Treća generacija procesora

**ČETVRTA GENERACIJA**

Daljnji razvoj procesora poglavito karakterizira RISC arhitektura, visoki stupanj integracije (više od milijun tranzistora) i mogućnost izvođenja više od jedne naredbe u istom ciklusu (skalarna arhitektura, ). Na taj se način osigurava na izlazu završetak barem jedne naredbe u svakoj radnoj periodu, unatoč neizbježnoj međuovisnosti naredbi i njihovih operanda koji uzrokuju povremena čekanja (Sl. 3-20).



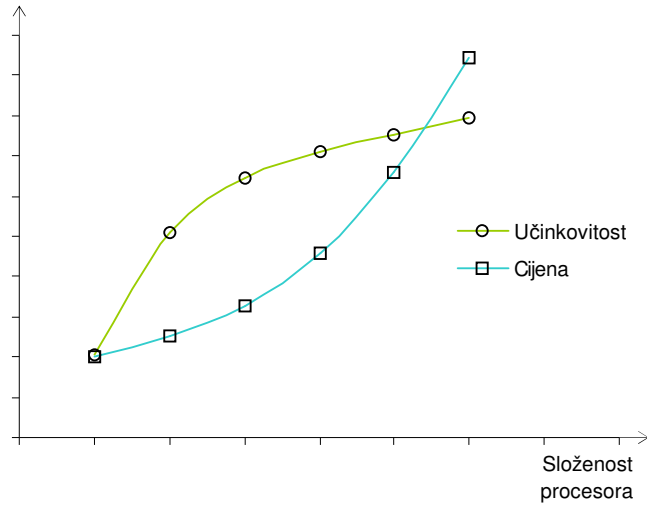
U – učitavanje naredbe, D – dekodiranje, I – izvođenje  
A – adresiranje, O – učitavanje operanda, P – pohrana

Sl. 3-20 Četvrta generacija

**PETA GENERACIJA**

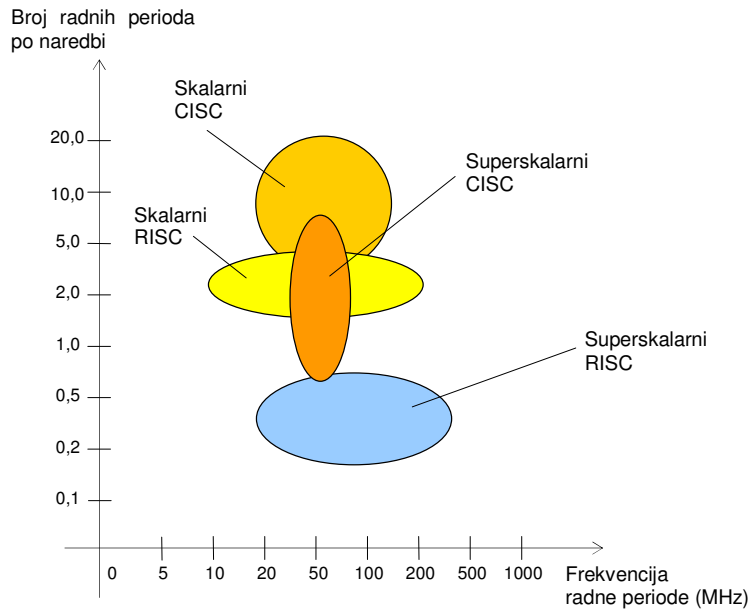
Današnji razvoj mikroprocesora pripada novoj petoj generaciji koju odlikuje primjena superskalarne i superprotočne arhitekture (Sl. 3-21). Posjeduju veći broj paralelnih aritmetičko/logičkih jedinki i velike primarne i sekundarne "cache" spremnike za naredbe i instrukcije. Rade na frekvenciji od više stotina do više tisuća MHz. U stanju su istovremeno obaviti više naredaba u jednoj radnoj periodu. U tome mu pomažu odgovarajući prevodioci programa (kompajleri) i operacijski sustavi, koji programe priređuju za paralelno izvođenje, osiguravajući izvanredne brzine obrade podataka.





Sl. 3-22 Teoretski odnos učinkovitosti i cijene procesora u odnosu na njegovu složenost

Naime, složenost integriranih funkcija iziskuje izvršavanje, čak i jednostavnih, naredbi u više radnih ciklusa, produžujući vrijeme obrade. S druge strane, istraživanja su pokazala da se tijekom rada računala češće koriste elementarne naredbe. Iz toga se rodio novi pogled na razvoj procesora nazvan RISC - eng. "Reduced Instruction Set Computer", odnosno procesor s smanjenim brojem naredbi. Iako istraživanja na ovom području započinju još 1975. godine u IBM-u, pojam RISC procesor uvode D. A. Patterson i D. R. Ditzel 1980. godine. Takvi procesori izvršavaju tek stotinjak jednostavnijih naredbi, ali svaku obavljaju u jednoj radnoj periodu (ciklusu ili taktu). Budući da su takve naredbe najučestalije, učinkovitost RISC procesora dolazi do izražaja i kod nižih radnih frekvencija. Ako se još uzme u obzir da je izradba RISC arhitekture neusporedivo jednostavnija i jeftinija, onda je jasno zašto je većina modernih procesora temeljena upravo na RISC konceptu (Sl. 3-23).



Sl. 3-23 Značajke CISC i RISC procesora



### 3.3.6 Kako radi procesor

Računalo izvodi program po zadavanju naredbe za njegovo izvođenje. Potom slijedi njegovo učitavanje u radnu memoriju (RAM), ako već nije u memoriji kao npr. dio operacijskog sustava ili nekog drugog rezidentnog programa. Pomoću upravljačke jedinice procesor učitava naredbe redom, ili prema zadanom slijedu (npr. uslijed naredbe JUMP). Naredbe se učitavaju u spremnik naredaba. Upravljačka jedinica obnavlja brojač naredaba, koji sadrži adresu sljedeće naredbe za izvođenje.

Procesor obavlja tri osnovne operacije: **učitavanje, dekodiranje i izvođenje naredbe**. Učitavanje se vrši pomoću upravljačke jedinice koja predaje zahtjev upravljačkoj jedinici radne memorije da pročita naredbu na zadanoj adresi. Kada upravljačka jedinica procesora dobije traženu naredbu, smješta ju u spremnik naredaba i postavlja brojač na adresu sljedeće naredbe. Sa naredbom u spremniku procesor započinje njeno prevođenje (dekodiranje), analizirajući samu naredbu i njene operande. Nakon dekodiranja, upravljačka jedinica inicira izvođenje naredbe, aktivirajući odgovarajuće integrirano sklopovlje da obave predviđenu zadaću (učitavanje podataka, računanje i drugo). Nakon izvođenja ponavlja se opisani ciklus, upravljačka jedinica poziva sljedeću naredbu i tako dalje. Dakako, kod modernih superskalarnih i superprotočnih procesora ove se operacije izvršavaju dijelom paralelno na više naredaba.

Prije nego što se rad procesora pokaže na primjeru jednostavnog programa, potrebno je detaljnije razmotriti strukturu naredbe, odnosno strojnog jezika.

#### 3.3.6.1 Strojni jezik

Procesor izvršava naredbe na temelju prepoznavanja njihovih uređenih binarnih kombinacija. Sustav kodiranja binarnih kombinacija naziva se strojni jezik. Kodirana naredba obično se sastoji iz dva polja. Prvo polje sadrži oznaku naredbe, a drugo adrese operanda. Radi jednostavnijeg i sažetijeg interpretacije strojni se jezik prikazuje u heksadecimalnom obliku. Heksadecimalni sustav se sastoji od 16 znamenaka, od 0 do F (Tab. 3-1).

Tab. 3-1 Heksadecimalni sustav s dekadskom i binarnom interpretacijom

Dekadski broj	Binarni broj	Heksadecimalni broj
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Tako se tek jednom znamenkom zamjenjuju četiri bita binarnoga sustava. Dakle, ukoliko je struktura naredbe temeljena na 16-bitnom zapisu, prva četiri bita otpadaju na oznaku naredbe (znači najviše 16 naredaba), dok preostalih dvanaest bitova mogu definirati adrese operanda ili neke druge podatke ovisno o naredbi.

Iako je prikaz instrukcije sažetiji u heksadecimalnom obliku, za lakše razumijevanje svakoj se instrukciji pridružuje simbolička oznaka. Skup simboličkih oznaka instrukcija naziva se "assembly jezik" ili jednostavno "*assembler*"<sup>10</sup>. U Tab. 3-2 prikazan je osnovni skup asemblerskih naredbi s odgovarajućim heksadecimalnim i binarnim instrukcijskim kodovima.

Tab. 3-2 Osnovni skup asemblerskih naredbi

Naredba	Opis	Heksadec.	Operandi	Binarno
<b>LOAD</b>	Učitaj binarni slog sa memorijske (RAM) adrese XY u spremnik R	1	RXY	0001
<b>LOAD (CON)</b>	Učitaj binarni slog XY u spremnik R	2	RXY	0010
<b>STORE</b>	Pohrani binarni slog iz spremnika R u memoriju na adresi XY	3	RXY	0011
<b>MOVE</b>	Premjesti binarni slog iz spremnika R u spremnik S	4	ORS	0100
<b>ADD</b>	Zbroji binarne kombinacije pohranjene u spremnicima S i T i rezultat pohrani u spremnik R	5	RST	0101
<b>SUB</b>	Oduzmi binarne kombinacije pohranjene u spremnicima S i T (S-T) i rezultat pohrani u spremnik R	6	RST	0110
<b>MUL</b>	Pomnoži binarne kombinacije pohranjene u spremnicima S i T i rezultat pohrani u spremnik R	7	RST	0111
<b>DIV</b>	Podjeli binarne kombinacije pohranjene u spremnicima S i T i rezultat pohrani u spremnik R	8	RST	1000
<b>COM</b>	Uspoređi binarne kombinacije pohranjene u spremnicima S i T i rezultat pohrani u spremnik R	9	RST	1001
<b>JEQ</b>	Skoči na programsku adresu XY ako su binarne kombinacije iz COM	A	RXY	1010

<sup>10</sup> Eng. assembly, u ovom kontekstu znači konstrukcija ili kompilacija, a odnosi se na jezik za opisivanje algoritama pomoću procesorskih naredaba.

	spremnika S i T jednake			
<b>JUMP</b>	Bezuvjetno skoči na programsku adresu XY	B	0XY	1011
<b>STOP (HALT)</b>	Prekid izvođenja programa	C	000	1100
<b>JG</b>	Skoči na programsku adresu XY ako je binarna kombinacija iz COM spremnika S veća od T	D	RXY	1101
<b>JL</b>	Skoči na programsku adresu XY ako je binarna kombinacija iz COM spremnika S manja od T	E	RXY	1110
<b>JNEQ</b>	Skoči na programsku adresu XY ako su binarne kombinacije iz COM spremnika S i T različite	F	RXY	1111

Uzmimo kao primjer heksadecimalni zapis naredbe 106E. Njena interpretacija je sljedeća: heksadecimalno 1 odgovara binarnoj kombinaciji 0001, ili npr. naredbi LOAD (učitaj); sljedeća četiri bita definiraju adresu spremnika u koji treba učitati podatak (R0), čija memorijska adresa je pohranjena u preostalim osam bitova (6E heksadecimalno ili 01101110 binarno). Ukoliko pretpostavimo naredbu 535A, tada prva znamenka 5 označava prema uobičajenom kodnom sustavu za strojne jezike, naredbu ADD - zbroji cjelobrojne vrijednosti<sup>11</sup> pohranjene u spremnicima R5 i RA, a rezultat pohraniti u spremnik R3.

### 3.3.6.2 Primjer

Pretpostavimo da želimo programirati algoritam zbrajanja dva broja. Algoritam očito mora sadržavati naredbe za učitanje pribrojnika iz memorije u procesorske spremnike, naredbu za zbrajanje te pohranu rezultata. Odgovarajući algoritam zbrajanja, pohranjen u memoriju na adresama od A2 do AB, prikazan je heksadecimalnim kodovima na Sl. 3-24.

Radna memorija	
Adresa	Sadržaj spremnika
A2	10
A3	5A
A4	11
A5	5C
A6	52
A7	01
A8	32
A9	C2
AA	C0
AB	00
AC	

Sl. 3-24 Primjer programa u heksadecimalnom kodu

<sup>11</sup> Postoje dvije naredbe zbrajanja (obično kodirane 5 i 6), prva za zbrajanje cjelobrojnih, a druga za zbrajanje decimalnih vrijednosti.

S obzirom da je oblik naredbe temeljen na 16-bitnom zapisu, a svaka memorijska adresa označava 8-bitni spremnik u radnoj memoriji, naredbe su pohranjene tako da zauzimaju po dvije memorijske lokacije. Zbog toga će procesor za svaku naredbu učitati po dvije memorijske lokacije, a procesorski brojač će uvećavati tekuću adresu za 2.

Na Sl. 3-25 prikazan je cijeli tijek izvođenja programa zbrajanja. Izvođenje programa započinje učitavanjem prve naredbe 105A (sa lokacija A2 i A3) u spremnik naredaba. Dekoder naredaba će ju interpretirati na sljedeći način:

- 1 - LOAD (učitaj)
- 0 - R0 (u procesorski spremnik R0)
- 5A - memorijska adresa 5A (sadržaj memorijskog spremnika 5A)

Brojač naredaba postavlja memorijsku adresu sljedeće naredbe (A4). Slijedi učitavanje naredbe 115C i njeno dekodiranje. Oznaka naredbe je opet jedan što znači LOAD (učitaj), ali sada u procesorski spremnik R1, sadržaj lokacije 5C.

Sljedeća naredba 5201, koja počinje na adresi A6, definira zbrajanje. Prvi broj 5 označava naredbu zbrajanja (ADD). Sljedeći broj (2) definira adresu procesorskog spremnika u koji treba pohraniti rezultat zbrajanja. Operandi 0 i 1 definiraju adrese spremnika (R0, R1) u kojima su pohranjeni prethodno učitani pribrojnici.

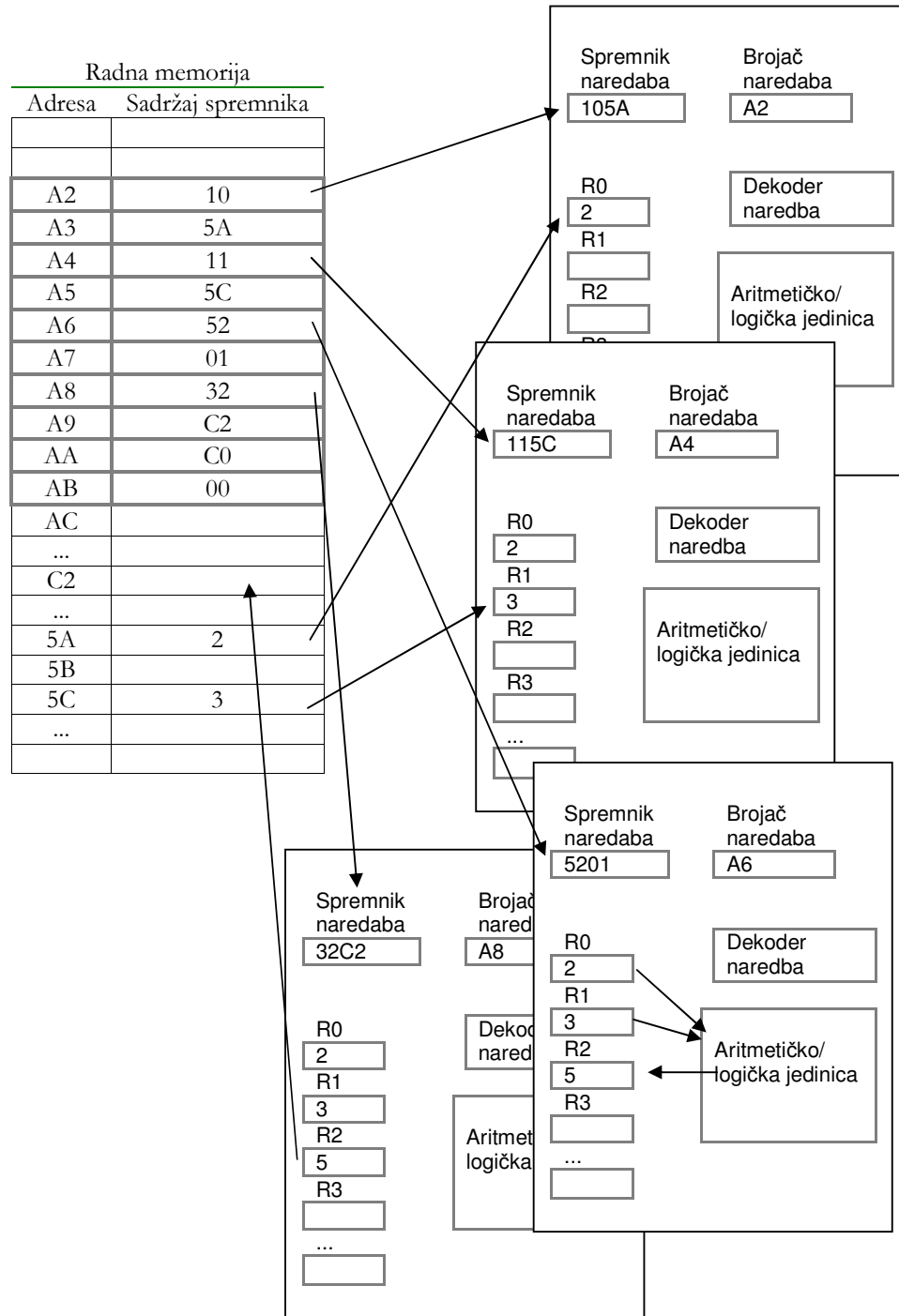
Nakon što je obavljeno zbrajanje rezultat treba pohraniti u radnu memoriju. Naredba 32C2, čiji zapis počinje na adresi A8, definira pohranu (3 - STORE) sadržaja procesorskog spremnika R2 na memorijsku adresu C2.

Zadnja naredba C000 znači završetak programa (C - HALT).

Razmotrimo sada nešto složeniji program, npr. za izračunavanja faktoriijela od broja 5,  $5! = 1*2*3*4*5 = 120$ . U C programskom jeziku program bi mogao izgledati ovako:

```
a = 1;
f = 1;
while ( a <= 5 )
{
    f = f * a;
    a = a + 1;
}
```

Na kraju programa varijabla *f* će sadržavati rezultat 5!. Ovako simbolički opisan algoritam u C jeziku potrebno je prevesti u assembler. To se u praksi izvodi pomoću posebnog programa prevodioca koji se zove "kompajler" (eng. "compiler").



Sl. 3-25 Primjer izvođenja programa za zbrajanje dvaju brojeva

Ukoliko kompajler slijedi assembler iz Tab. 3-2 dobit ćemo naredni programski kod:

Adresa naredbe u memoriji	Assemblerski kod	Objašnjenje
0	LOAD(CON) 0 1	// a = 1
1	STORE 0 80	// na RAM adresi heksad. 80 ili dek. 128
2	LOAD(CON) 0 1	// f = 1
3	STORE 0 81	// na RAM adresi heksad. 81 ili dek. 129
4	LOAD(CON) 0 5	// if a > 5 jump to end
5	LOAD 1 80	
6	JG 1 10	
7	LOAD 0 81	// f = f * a
8	LOAD 1 80	
9	MUL 2 0 1	
A	STORE 2 81	
B	LOAD 0 80	// a = a + 1
C	LOAD(CON) 1 1	
D	ADD 2 0 1	
E	STORE 2 80	
F	JUMP 0 4	// nazad na naredbu 4
10	STOP (HALT) 0 0	
	0	

Isti program, ali sada u heksadecimalnom i binarnom kodu ima sljedeći oblik:

Adresa u memoriji	Heksad. kod	Binarni kod	Objašnjenje
0	2 0 1	0010 0000 0000	// a = 1
1	3 0 80	0011 0000 1000	// na RAM adresi heksad. 80 ili dek. 128
2	2 0 1	0010 0000 0000	// f = 1
3	3 0 81	0011 0000 1000	// na RAM adresi heksad. 81 ili dek. 129
4	2 0 5	0010 0000 0000	// if a > 5 jump to end
5	1 1 80	0001 0001 1000	
6	D 1 10	1101 0001 0001	
7	1 0 81	0001 0000 1000	// f = f * a
8	1 1 80	0001 0001 1000	
9	7 2 0	0111 0010 0000	
A	3 2 81	0011 0010 1000	
B	1 0 80	0001 0000 1000	// a = a + 1
C	2 1 1	0010 0001 0000	

```

D      5 2 0 0001 0101 0010 0000
      1 0001
E      3 2 80 0011 0010 1000
      0000
F      B 0 4 1011 0000 0000 // nazad na naredbu 4
      0100
10     C 0 0 1100 0000 0000 // stop
      0 0000
```

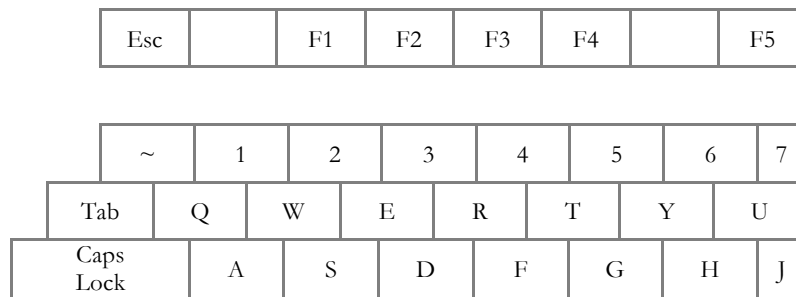




## 3.4 Ulazno/izlazni uređaji

### 3.4.1 Tipkovnica

Tipkovnica je uređaj za unos alfanumeričkih znakova. Putem tipkovnice se zadaju naredbe, tekstualni i numerički podaci. Svaka tipka ima svoje 7-bitno tumačenje propisano ASCII normom (poglavlje ). Pritiskom na tipku ostvaruje se elektromehanički kontakt koji generira u računalu odgovarajući binarni signal. Raspored tipki propisan je ISO normom koja se popularno zove QWERTY<sup>12</sup>, a odgovara rasporedu prvih šest slovnih znakova pridruženih gornjem lijevom nizu tipaka (Sl. 3-26).



Sl. 3-26 Dio QWERTY tipkovnice

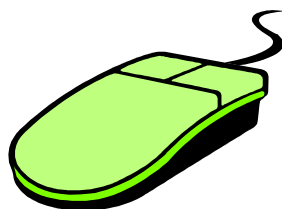
Široka primjena grafičkog korisničkog sučelja značajno je smanjila upotrebu tipkovnice, a interakciju s računalom usmjerila na korištenje miša. Osim toga, sve šira primjena sustava za interpretaciju govora dodatno smanjuje potrebu unošenja podataka pomoću tipkovnice.

### 3.4.2 Miš

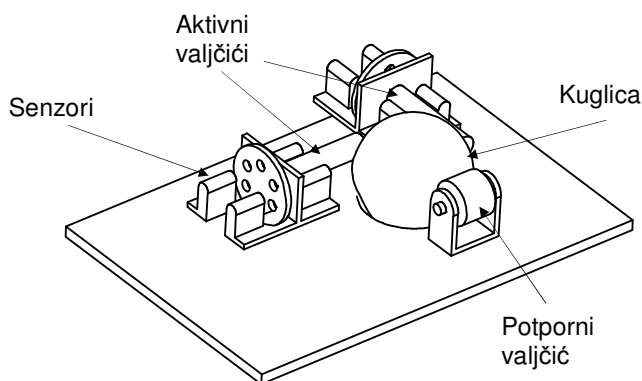
Miš je ulazna jedinka čije se pomicanje po ravnoj podlozi odražava na zaslonu kao pomicanje grafičkog pokazivača (Sl. 3-27). Na podnožju miša nalazi se kuglica koja se prilikom pomicanja miša okreće, prenoseći gibanje na okretne elemente. Pomoću ugrađenih potencijometara ili optičkih osjetila očitavaju se relativna kretanja i pretvaraju u numeričke vrijednosti koje računalu koristi za izračun smjera i dužine pomaka grafičkog pokazivača na zaslonu (Sl. 3-1). Na gornjem dijelu miša nalaze se dvije ili tri tipke koje se koriste za aktiviranje različitih funkcija, ovisno o interpretaciji položaja grafičkog pokazivača na zaslonu, odnosno grafičkom korisničkom sučelju. Lijeva tipka miša obično se koristi za aktiviranje glavnih sučelju pridruženih funkcija, a desna tipka služi za npr. otvaranje pomoćnih izbornika.



<sup>12</sup> Raspored tipaka QWERTY patentirala je 1878. godine tvrtka "Remington" na svojim pisačim strojevima. QWERTY postaje službenom ISO normom tek 1991. godine.



Sl. 3-27 Miš



Sl. 3-28 Unutrašnjost miša

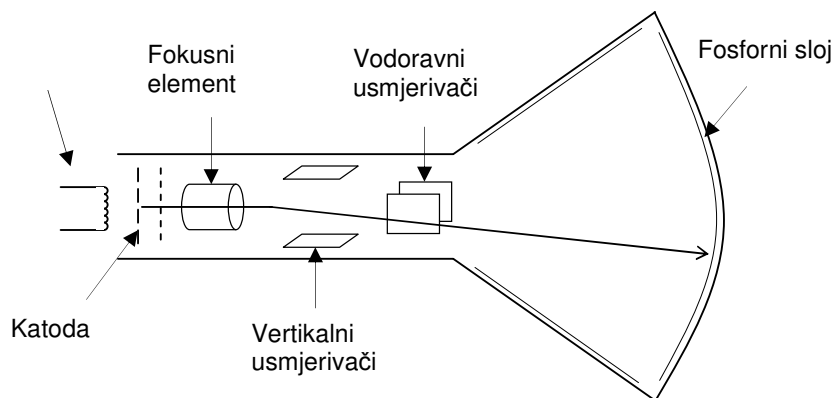
### 3.4.3 Zaslون

Zaslون računala je izlazni vizualni uređaj na kojem se prati rad računalskih programa. S obzirom na vizualnu prirodu naših mentalnih funkcija i suvremenim dosezima na području vizualnog računarstva, uređaji za vizualizaciju postaju sve važniji u interakciji čovjek - računalo. Najrašireniji su zasloni koji koriste tehnologiju katodne cijevi, sličnu kao i u TV prijemnicima i tekućeg kristala.



#### 3.4.3.1 Katodna cijev

Na slici prikazana je shema katodne cijevi. U osnovi se sastoji od "elektronskog topa" i vakuumske cijevi s zaslonom (). Elektronski top emitira struju elektrona iz katode smještene na početku vakuumske cijevi. Elektroni ubrzavaju pod djelovanjem razlike napona između katode i zaslona na kraju vakuumske cijevi. Primjenom elektrostatskog ili magnetskog polja elektroni se fokusiraju i skreću na način kao što se to čini sa svjetlosnim zrakama u optičkim sustavim pomoću leća. Zaslون na kraju vakuumske cijevi presvučen je s fluorescentnim fosforim materijalom. Emitirani elektroni podražavaju fosforim materijala tako da oslobađaju energiju u obliku vidljive elektromagnetske radijacije. To se svojstvo naziva fluorescencija.



Sl. 3-29 Katodna cijev

Elektronska zraka osvjetljava zaslon obilazeći ga liniju po liniju, od gornjeg lijevog kuta desno i dolje. Fosforni materijal na zaslonu ima svojstvo da zadržava uzbudu i nakon djelovanja struje elektrona u trajanju od 10 do 70 ms (mikrosekundi). Vrijeme trajanja uzbuđenja fosfornog materijala ujedno je minimalno vrijeme između dva prolaza elektrona, a da pritom slika na zaslonu ostane postojana. S obzirom da tromost ljudskog oka dozvoljava uočavanje svjetlosnih promjena dužih od  $1/40$  s, nužno je osigurati kraće vrijeme osvježavanja zaslona da ne dođe do treperenja slike. Na suvremenim katodnim cijevima vrijeme osvježavanja se kreće od  $1/50$  do  $1/90$  s.

Zaslon je podijeljen na mrežu točaka (eng. "pixels"), koje se u računalu interpretiraju kao rasterski elementi slike. Svakom elementu slike pripada odgovarajuća x,y koordinata na zaslonu i podatak o boji. Ti se podaci pohranjuju u posebnom spremniku grafičkoga sučelja (eng. "image memory" ili "frame buffer"). S obzirom na veličinu grafičkog spremnika najčešće susrećemo 8-bitna i 24-bitna grafička sučelja. Taj podatak govori kolika je dužina binarnog sloga predviđena za pohranu informacije o boji točke. U slučaju 8-bitne grafike moguće je pohraniti 256 nijansi boja, dok s 24-bitnim slogom na raspolaganju nam stoji 16,7 miliona boja.

### 3.4.3.2 Tekući kristal

Tekući kristal je tvar koja ima svojstva tekućina i kristala. Osim što egzistira u tekućem stanju, njegove molekule su djelomično usmjerene. Promjenom električnog polja ili temperature mijenja se molekularna usmjerenost tekućeg kristala. To se očituje u promjeni boje i providnosti. Koristeći ovo svojstvo, tekući kristal se može upotrijebiti slično kao fluorescentni zaslon za prikaz digitalnih informacija. Polarizacijom pojedinih segmenata (točaka) na zaslonu usmjeravaju se molekule tekućeg kristala, formirajući neprovidna polja, odnosno vizualni učinak koji odgovara očekivanoj slici.

## 3.4.4 Pisači

Pisači su uređaji koji služe za ispis tekstualnog i/ili grafičkog sadržaja na papir, plastičnu foliju ili tkaninu. Postoji cijeli spektar različitih tehnika koje se koriste za ispis računarskih podataka (najčešće na papir), a u skladu s primijenjenom tehnikom pisači su dobili naziv:



- ❑ linijski
- ❑ matrični
- ❑ termički
- ❑ sublimacijski
- ❑ tintni "InkJet"
- ❑ laserski.

**Linijski pisači** koriste elektromehaničku tehniku otiskivanja znakova na papir, slično kao na klasičnim pisačim strojevima. Znakovi se nalaze na vrpci ili posebnoj pisačkoj glavi. Okretanjem vrpce ili pisače glave pozicionira se određeni znak prema papiru. Udarcom u vrpcu natopljenu tintom znak ostaje otisnut na papiru. Takvi su se pisači odlikovali velikom brzinom ispisa, ali nisu prikladni za ispis grafičkog sadržaja. Danas su rijetko u upotrebi.

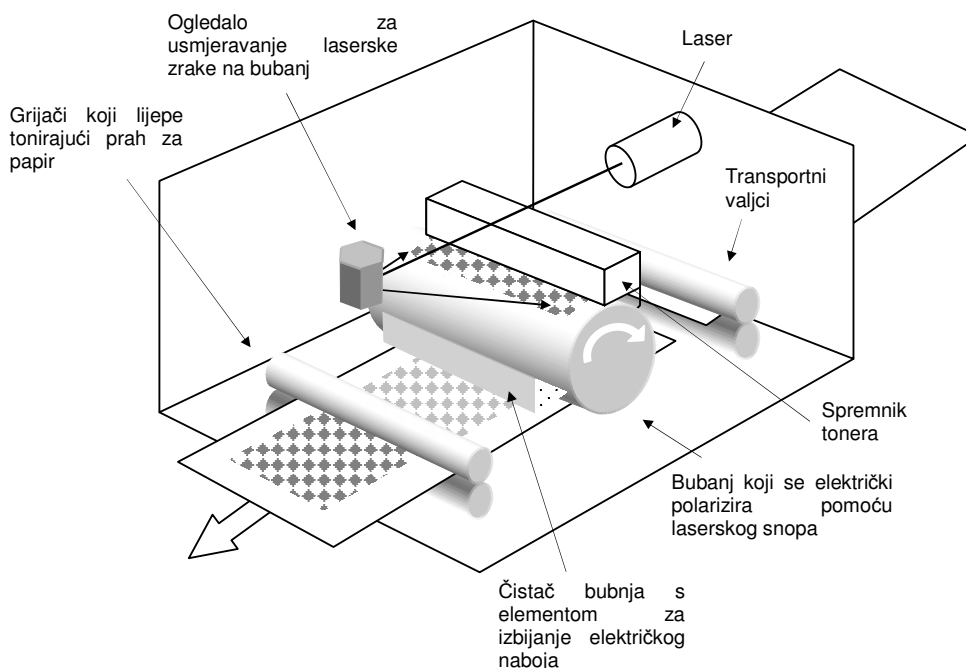
Na sličnom principu rade i **matrični pisači**. Pritom je pisača glava zamjenjena s matričnom glavom koja sadrži jedan ili više uspravnih nizova od obično devet ili 24 iglica. Koordiniranim udarima iglice, preko tintne vrpce, oblikuju na papiru rasterski (matrični) uzorak u skladu s željenim izgledom znaka. S obzirom da iglice mogu ispisivati različite rasterske oblike, matrični su printeri pogodni i za ispis grafičkog sadržaja. Osnovna je zamjerka ovoj tehnologiji ispisa sporost i slaba kvaliteta.

**Termički pisači** nadopunjuju matričnu tehnologiju zamjenom tintne vrpce za termički osjetljiv papir. Iglice se na matričnoj glavi zagrijavaju i u doticaju s termički osjetljivim papirom ostavljaju tamni otisak. Termički pisači su iznimno tihi te se poglavito koriste u knjižnicama i sličnim prostorima gdje bi buka mogla ometati rad. Nedostatak ove tehnologije očituje se uglavnom u potrebi primjene specijalnog papira i nepostojanosti otiska, točnije pod djelovanjem ultravioletnog zračenja otisak s vremenom izbljedi ili cijeli papir potamni.

**"Ink-jet" pisači** koriste tekuću tintu koja se istiskuje na papir zagrijavanjem. Spremnici putuju na klizaču kao pisače glave. Na spremnicima tinte integrirani su termičke sapnice. Tinta kapa iz spremnika u sapnicu koja se električki zagrijava. Zagrijana kapljica ekspandira i višak izliječe iz sapnice na papir. Spremnici sadrže u pravilu četiri osnovne boje: žutu, cyan, magentu i crnu. Kombiniranjem kapljica različite boje nijansira se otisak željene boje. Tintni pisači daju vrlo kvalitetne ispise u boji, a mogu se koristiti i za ispis fotografija.

**Laserski pisači** danas su najrasprostranjeniji u primjeni zbog svoje kvalitete i relativno niske cijene. Princip rada naslijeđen je od fotokopirnih strojeva. Središnji dio laserskog pisača je valjak presvučen tankim slojem poluvodičkog materijala, najčešće selenom. Takva poluvodička presvlaka omogućava elektrostatsku polarizaciju površine valjka pomoću svjetlosnog snopa. Kako je selen zapravo izolator, inducirani statički elektricitet zadržava se samo na površini valjka. Toner, suha boja u prahu, pohranjena je u posebnom spremniku koji je integriran sa selenskim valjkom. S obzirom da je valjak u osnovnom stanju istog polariteta kao i toner, odbijaju se. Površina valjka polarizira se suprotnim nabojem pomoću laserskog snopa koji se odbija od rotirajućeg zrcala (Sl. 3-30). Laserska zraka se pali i gasi sinkronizirano s zrcalom dok ono putuje uzduž valjka, bombardirajući površinu valjka na mjestima gdje treba biti otisak. Valjak se okreće prolazeći kroz toner koji se prima za površinu valjka na mjestima suprotne polarizacije. U slučaju pisača u boji, ovaj se postupak ponavlja četiri puta, za svaku boju (žutu, magentu, cyan i crnu) posebno. Papir koji se uvlači iz spremnika polarizira se suprotno od tonera na bubnju pomoću tanke žice (korone). Stoga kada dođe u doticaj s površinom bubnja toner prelazi na papir. Slijedi izbijanje zaostalog naboja na

površini bubnja, odnosno priprema za ponovnu polarizaciju. Papir s tonerom potom prolazi kroz grijane valjke koji rastapaju boju i sublimiraju u papir.



Sl. 3-30 Shema laserskog pisača

Upravljački sklop laserskog pisača opremljen je procesorom, RAM-om i ROM-om, i predstavlja računalo specijalne namjene. Pisač pohranjuje sadržaj koji treba ispisati u RAM. Pomoću programa (pohranjenog u ROM-u) za prevođenje sadržaja u rasterski format, podaci se pretvaraju u oblik pogodan za upravljanje laserom. Drugim riječima, sadržaj se pretvara u točke, a svaka točka znači jedno paljenje lasera. Ovakvo računalno sklopovlje pisača naziva se RIP - "Raster Image Processor" (procesor rasterske slike). Rasterski format se pohranjuje u RAM koji mora biti dimenzioniran za prihvatanje rasterske slike sukladno primijenjenoj razlučivosti. Npr., za razlučivost 600x600 točaka po inchu i stranicu A4 formata potrebno je  $600 \times 600 \times 8.3 \times 11.7 = 34\,959\,600$  bitova, odnosno 4.1675 MB za ispis u jednoj boji. Znači, za ispis slike u boji potrebno je osigurati najmanje 16 MB RAM-a u pisaču. Postoji mogućnost korištenja računalnog procesora za pretvorbu podataka u rasterski format. Tada se može koristiti računalna memorija za pohranu obrađene slike, a pisaču se prenose linija po linija već pripremljenih rasterskih podataka.



---

## 4 OPERATIVNI SUSTAV

---

**K**ada uposlamo računalo da obavi neku zadaću ne opterećujemo se previše pitanjem da li tu zadaću obavlja računalo, odnosno procesor i ostale komponente, ili program koji se na njemu izvodi. To pitanje nalikuje na ono da li je prije nastalo kokoš ili jaje, jer program i računalo tvore simbiozu u kojoj je računalo beskorisno bez programa, kao i sam program, ako se ne može nigdje izvesti. Dakle, računalo ne može raditi bez uređenog niza električnih impulsa, koji protječu poluvodičkim sklopovljem procesora u skladu s programskim naredbama, odnosno binarnim kombinacijama koje i nisu ništa drugo nego električki impulsi (1 - uključeno, 0 - isključeno). Da bi računalo moglo izvršavati zadane naredbe i programe, ono mora znati interpretirati naše zahtjeve i u skladu s time upravljati radom svojih komponenata. Tu sposobnost računalu osigurava poseban program (u pravilu više njih), koji se zove operativni sustav i koji se aktivira odmah po uključivanju računala. Operativni (radni) sustav osigurava interakciju čovjek-računalo, pretvarajući simboličke podatke, kojima opisujemo naše zahtjeve i probleme, u procesorske naredbe, binarni strojni jezik, odnosno električne impulse i obrnuto – binarne rezultate u simbole koji se na razumljiv način predstavljaju na zaslonu računala.

U početku su se operativni sustavi razvijali po uzoru na konkretan model računala i njegovu arhitekturu. Takav pristup izrazito je komplicirao primjenu računala, jer su se korisnici morali dugotrajno i skupo školovati uvijek iznova za svaki novi tehnološki pomak ili jednostavno zbog odabira računala drugog proizvođača. S vremenom, opće postavke i pristupi u teoriji operativnih sustava približavali su se jedinstvenom konceptu. S razvojem operativnog sustava UNIX postavljeni su temelji razvoju takozvanih otvorenih sustava. Otvorenost pritom znači da se operativni sustav može bez ograničenja implementirati na različitim računalima i da podržava opće prihvaćene norme koje omogućavaju komunikaciju s drugim računalima i operativnim sustavima. S obzirom da je operativni sustav sučelje između korisnika i računala, otvoreni sustavi su učinili primjenu računala univerzalnom, stvarajući iluziju da računalo i računalna mreža tvore uvijek isto prepoznatljivo računalno ustrojstvo, bez obzira na proizvođača i konfiguraciju opreme. Stoga je UNIX postao široko prihvaćen operativni sustav na svim razinama, otvarajući vrata računalnoj komunikaciji i razvoju računalnih mreža.

U domeni osobnih računala razvoj se zbiva na ponešto drugačijim osnovama, diktiran od strane masovnog tržišta i glavnih proizvođača procesora i programske podrške za PC-e - Intela i Microsofta. Svojom jednostavnošću, mnogostrukom funkcionalnošću i cijenom, osobna su računala postupno preuzimala primjenu u svim djelatnostima gdje se javlja potreba za obradom podataka bilo koje vrste. U skladu s time PC operativni sustav morao je biti jednostavan i prilagođen općim zadaćama. Potrebe povezivanja osobnih računala, i to ne samo međusobno već i s drugim računalima, učinile su otvorenost jednako važnom značajkom PC operativnih sustava. Normizacija na

području tehnologije osobnih računala, usmjeravana je ponajviše utjecajem najvećeg proizvođača PC programske podrške - firme Microsoft, što je rezultiralo dominacijom Microsoft Windows NT/95 operativnih sustava. Danas je stoga računalna primjena poglavito svedena na dva operativna sustava: UNIX i Microsoft Windows NT/95. UNIX se primjenjuje na većim i složenijim računalima; višekorisničkim i/ili višeprosorskim radnim stanicama ili poslužiteljima; a Windowsi NT/95 najrašireniji su operativni sustav osobnih računala. U nastavku će biti obrađene osnovne funkcije i ustrojstvo operativnih sustava, a potom je pokazano kako su te funkcije implementirane u UNIX-u i NT-u.



## 4.1 Što je operativni sustav

Iz perspektive krajnjega korisnika, za operativni bi se sustav moglo reći da je skup naredaba putem kojih se izdaju nalozi računalu. Međutim, osim o komunikaciji s korisnikom, računalo mora brinuti i o radu svojih sklopovskih komponenata, memoriji, diskovima, ulazno/izlaznim jedinicama i slično, realizirajući svoj funkcionalni smisao. Operativni se sustav brine o pohrani podataka, oblikujući razumljivu zapisničku strukturu neovisno o fizičkom smještaju podataka na disku, sinkronizira rad procesora i ulazno/izlaznih jedinica kada na računalu istovremeno radi više korisnika i čakako više programa, interpretira ulazne podatke jednako kao i izlazne rezultate, prikazujući ih na razumljiv i slikovit način na zaslonu računala, upravlja radnom memorijom učitavajući aktivirane programe s diska i pohranjujući pasivne radi optimiranja radnog prostora, te obavlja mnogobrojne druge zadaće, često nevidljive za prosječnog korisnika, ali nužne za rad računala. Operativni sustav stvara iluziju kod korisnika da računalo komunicira, razumije i izvršava zadano na temelju problemski orijentiranih pojmova, iako ono funkcionira jedino na razini bitova, odnosno nula i jedinica. Zato je definicija operativnog sustava nužno složeniija od prvotno iskazane te bi se mogla sažeti na sljedeći način:

operativni je sustav skup programa koji brinu o radu računalnih komponenata, komunikaciji s korisnikom i izvođenju korisničkih programa.



Operativni sustav osigurava najviši stupanj automatizacije računalnih funkcija. U isto vrijeme nudi obilje najrazličitijih programskih mogućnosti, čineći računalo gotovo univerzalnim strojem, ali i prilično složenim sredstvom. Stoga primjenu računala ne možemo ograničiti na jednodimenzionalni pristup koji se svodi na nizanje pojedinačnih naredbi. Poznavanje logike i strukture operativnog sustava pomoći će nam da dokučimo nepregledne mogućnosti računala te kreiramo i rješavamo složene zadaće.



## 4.2 Struktura i zadaće operativnog sustava

Operativni se sustav aktivira prilikom uključivanja računala pomoću programa za pokretanje (eng. "Boot program" ili "Bootstrap"<sup>13</sup>), pohranjenog u permanentnoj programibilnoj memoriji (EPROM-u). U EPROM-u pokretački program ostaje sačuvan i kad je računalo isključeno. Kod uključivanja, programsko brojilo procesora inicijalno postavlja adresu programa za pokretanje i započinje s njegovim izvođenjem. Program za pokretanje sadrži naredbe za učitavanje operativnog sustava s memorijskog diska u radnu memoriju i aktiviranje.

Operativni sustav u osnovi čine:

- ❑ procesor naredaba (monitor)
- ❑ uslužni programi
- ❑ upravljački programi:
  - program za raspoređivanje procesa (eng. scheduler)
  - program za alokaciju komponenata
  - program za rad s zapisnicima
  - programi za upravljanje računalnim komponentama
  - program za upravljanje memorijom
  - izvršni program

**Procesor naredaba** je poseban dio operativnog sustava putem kojeg računalo komunicira s korisnikom. To je program koji nadzire ulazne jedinice (tipkovnicu, miša, ...) i interpretira primljene signale, odnosno naredbe. Kada primi razumljivu naredbu, koja u pravilu znači pokretanje nekog uslužnog ili korisničkog programa, proslijeđuje nalog za njegov izvođenje programu za raspoređivanje procesa.

**Uslužni programi** jesu programi operativnog sustava koji korisniku omogućavaju rad s zapisnicima, računarskim procesima, mrežom i okolnim uređajima. Tu spadaju programi za kreiranje, kopiranje, brisanje, čitanje itd.

**Program za raspoređivanje procesa** pokreće procese izvođenja programa, proslijeđujući ih odgovarajućim programima za upravljanje zapisnicima, memorijom i ostalim komponentama.

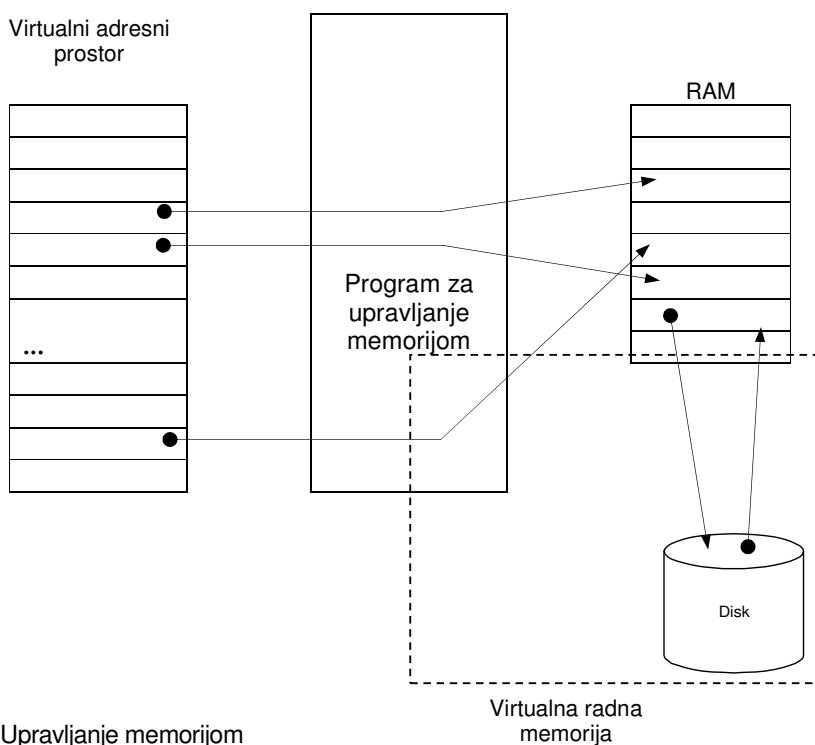
Pritom, **program za alokaciju komponenata** brine o njihovoj raspoloživosti, osiguravajući svakom aktivnom procesu da bez ometanja koristi dodijeljeno raspoloživo vrijeme (npr. za vrijeme dok disk zapisuje podatke za jedan aktivni proces, ne može se zaposliti isti uređaj da čita ili zapisuje podatke nekog drugog aktivnog procesa).

**Program za rad s zapisnicima** vodi brigu o pohrani podataka na disk, njihovom kopiranju, brisanju, čitanju i slično. *Zapisnik* (datoteka, eng. file) je logički objekt operativnog sustava u koji se pohranjuju podaci (informacije). Podaci mogu sačinjavati program u programskom jeziku ili binarnom kodu, bazu podataka, tekst, sliku, zvuk i drugo. Podaci se fizički pohranjuju na disk u organizaciji koja odgovara tehnologiji uređaja. Fizička organizacija podataka slijedi kriterije iskorištenja diskovnog prostora.

<sup>13</sup> Eng. "boot" znači čizma, ali i udarac, zapaljenje i slično. U računalnoj terminologiji opisuje početak (inicijaciju) rada računala.

Podaci koji logički pripadaju jednom zapisniku mogu biti fizički zapisani na više odvojenih diskovnih područja, jer upravljački program diska nastoji popunjavati eventualne praznine nastale povremenim brisanjem podataka. Takav oblik zapisa podataka nepregledan je i nerazumljiv. Operativni sustav stoga brine o interpretaciji podataka u obliku *zapisničkog sustava* (eng. file system) koji dozvoljava logičnu organizaciju podataka.

**Program za upravljanje memorijom** preuzima podatke s diska i pohranjuje na raspoložive lokacije u radnoj memoriji. Ukoliko je za neki program potrebno dodijeliti više memorijskih lokacija nego što je raspoloživo, upravljački program memorije otvara na disku tzv. "virtualnu" memoriju proširujući tako kapacitet RAM-a. Ovisno o konfiguraciji računala i operativnog sustava, ukupni adresni prostor može se protezati na više GB. Dijeli se na manje memorijske cjeline - stranice, npr. veličine 4 KB. Program za upravljanje memorijom preslikava virtualne adrese u RAM adrese (Sl. 4-1).

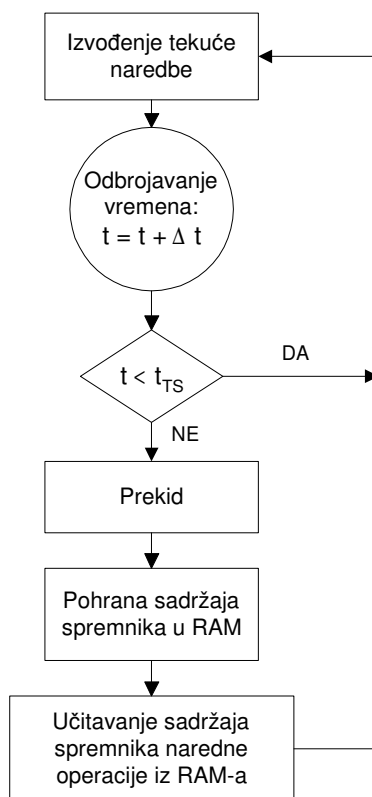


Sl. 4-1 Upravljanje memorijom

Ukoliko u RAM-u nema više raspoloživih memorijskih stranica, poseban algoritam ocjenjuje koje su stranice RAM-a trenutno pasivne te ih premješta (eng. "Swap") na disk. Tada se rad računala ponešto usporava, jer je potrebno podatke učitavati u procesorske spremnike putem niza dodatnih operacija pisanja i čitanja na relaciji disk - RAM - procesor. Ipak, na taj se način osigurava rad s vrlo obimnim programima, unatoč ograničenjima u pogledu memorijskih kapaciteta.

Programske naredbe i podatke pohranjene u radnoj memoriji preuzima na obradu procesor u skladu s **izvršnim programom** operativnog sustava. Taj se dio operativnog sustava brine za raspodjelu procesorskog vremena (eng. "Time Sharing"). S obzirom da se na računalu u pravilu obrađuje istovremeno više procesa, od jednog ili više korisnika, potrebno je svakom omogućiti podjednak pristup procesorskim

kapacitetima, odnosno mogućnost da se prividno paralelno obrađuju. Stoga izvršni program u kratkim vremenskim intervalima prekida rad procesora (eng. interrupt), izmjenjujući izvršavanje naredaba čas jednog čas drugog procesa. Kada procesor primi od izvršnog programa operativnog sustava signal za prekid procesa kojeg trenutno izvršava, obaviti će radnu periodu do kraja, znači do učitavanja sljedeće naredbe. Potom će izvršni program pohraniti sadržaj svih spremnika u radnu memoriju i nastaviti izvođenje narednog procesa na mjestu prekida, učitavajući iz radne memorije zadnje stanje njegovih spremnika prije nego što je prekinut. Započinje novo odbrojavanje vremena te ponovno prekid i tako dalje. Dio izvršnog programa koji prekida rad procesora, pohranjuje i učitava procese na mjestu prekida zove se *prekidačka procedura* (eng. interrupt routine, Sl. 4-2).



Sl. 4-2 Postupak dijeljenja procesorskog vremena ("Time Sharing")

Vrijeme prekida se obično definira u rasponu od 10 do 100 ms. Ukoliko se na računalu "istovremeno" obavlja velik broj procesa, čekanje u redu za obradu na procesoru može prouzročiti značajno usporavanje izvršenja programa. To posebno dolazi do izražaja u interaktivnom radu, manifestirajući se kao usporeni odziv računala. Taj je problem immanentan velikim poslužiocima. Međutim, suvremena se računala rijetko sučeljavaju s brojnošću istovremenih procesa kao ozbiljnim problemom. Zbog snižavanja cijena računalne opreme i stalnog unapređivanja procesorske moći, primjena se računala ponajviše temelji na distribuiranim procesorskim kapacitetima, odnosno osobnim računalima i radnim stanicama, poslužujući u pravilu samo po jednoga korisnika.



---

### 4.3 MS Winows operativni sustavi

Osobna računala (PC) najrašireniji su računarski koncept u općoj primjeni. Kao što naziv govori, osobna su računala namijenjena osobnom korištenju, a to je bar u početku značilo proizvodnju jednostavnih računala skromnijih mogućnosti, podržanih programima za razvoj i korištenje jednostavnijih uslužnih programa, npr. za obradu teksta, manjih baza podataka, jednostavnije proračune, crtanje i drugo.

Prvo računalo s značajkama današnjih PC-a možemo prepoznati u računalu PC-Alto, nastalom u laboratorijima Xeroxa 1973. godine. Tek nakon što uprava Xeroxa raspušta razvojni projekt PC-a, nerazumijući dalekosežnost vizije svojih znanstvenika, slijedi razvoj već legendarnog Applea. Načinili su ga 1976. godine bivši Xeroxovi stručnjaci Steve Jobs i Steve Wozniak, pokrenuvši nezaustavljivi kotač razvoja "PC računarstva". Ubrzo IBM uviđa ozbiljnost novog računalnog koncepta i formira svoju razvojnu skupinu koja 1981. godine objavljuje prvi IBM PC, čvrsto zacrtavši pravac daljnjeg razvoja PC računarstva.

S obzirom da su se veliki i skupi poslužiooci koristili za složene i komercijalne zadaće, a pristup je bio ograničen raspoloživim brojem terminala, mogućnost posjedovanja malog osobnog računala, koje je moglo u velikoj mjeri zamijeniti i čak dopuniti velika, pokazala se iznimno važnom. Stalno unapređivanje i snižavanje cijena osobnih računala otvaralo je nova područja primjene, prvenstveno zamjenjujući postojeće poslužioce i preuzimajući na sebe potpuno nove poslove (npr. pisačkog stroja, crtače daske, adresara, ...). Stvara se nesagledivo računalno tržište u stalnoj ekspanziji. Ono postaje izvorište ogromnih zarada, koje pak daju poticaj sve bržem razvoju računarstva i posredno svih ljudskih djelatnosti gdje se računala mogu primijeniti. Računarstvo prestaje biti privilegija bogatih instituta i kompanija te postaje imperativ sadašnjosti.

S obzirom da se osobna računala poglavito usmjeravaju prema korisnicima koji ne moraju biti visoko obrazovani na području računarstva, jednostavnost primjene jedan je od najvažnijih aspekata masovne primjene PC-a. Stoga se razvoju operativnog sustava i njegovog sučelja posvećivala posebna pažnja, daleko veća nego u slučaju radnih stanica ili velikih poslužitelja namijenjenih stručnim korisnicima. U tome je smislu posebnu ulogu odigrala tvrtka Microsoft, prvo s razvojem operativnog sustava DOS (eng. "Disk Operating System"), zatim razvojem prvog grafičkog korisničkog sučelja MS Windows 1-3.11, 1985. godine, i potom operativnog sustava Windows NT /2000 i 95/98<sup>14</sup>. Danas Windows operativni sustavi sa svojim intuitivnim sučeljem i brojnim uslužnim programima predstavljaju glavne alate masovne računalne primjene. U nastavku će biti objašnjena koncepcija NT operativnog sustava koji predstavlja osnovu ostalih MS Windows inačica koje su ga slijedile.

---

<sup>14</sup> Windowsi 95 i 98 predstavljaju osobnu (kućnu) verziju operativnog sustava namijenjenu jednostavnijim zadaćama, dok su Windowsi NT i 2000 razvijeni za profesionalnu primjenu.

### 4.3.1 Koncept i organizacija NT-a

Windows NT operativni sustav je prije svega višezadaćni i višeprocorski sustav. Razlikujemo dvije izvedbe NT-a: "Server" i "Workstation".

**NT Server** je namijenjen posluživanju lokalnih mreža (eng. "NOS - Network Operating System") na principima korisnik/poslužilac (eng. client/server) pristupa. Omogućava pristup većeg broja korisnika kroz mrežu, kao i višeprocorsko simetrično posluživanje do 32 procesora.

**NT Workstation** je poglavito prilagođen radu s korisničkim programima, s mogućnošću korištenja mrežnih i zapisničkih usluga NT Servera.

Osnovne komponente NT-a su:

- ❑ NT kernel (jezgra),
- ❑ "HAL" (eng. "Hardware Abstraction Layer", u opisnom prijevodu - izdvojena sklopovska razina),
- ❑ Izvršne komponente
- ❑ Operativni podsustav (Win32.exe).

NT jezgra čini osnovni dio operativnog sustava koji putem posebne sklopovske razine (HAL) koordinira radom programa i računalnih komponenata.

Rad s korisnikom osigurava operativni podsustav. On upravlja uslužnim programima i korisničkim sučeljem.

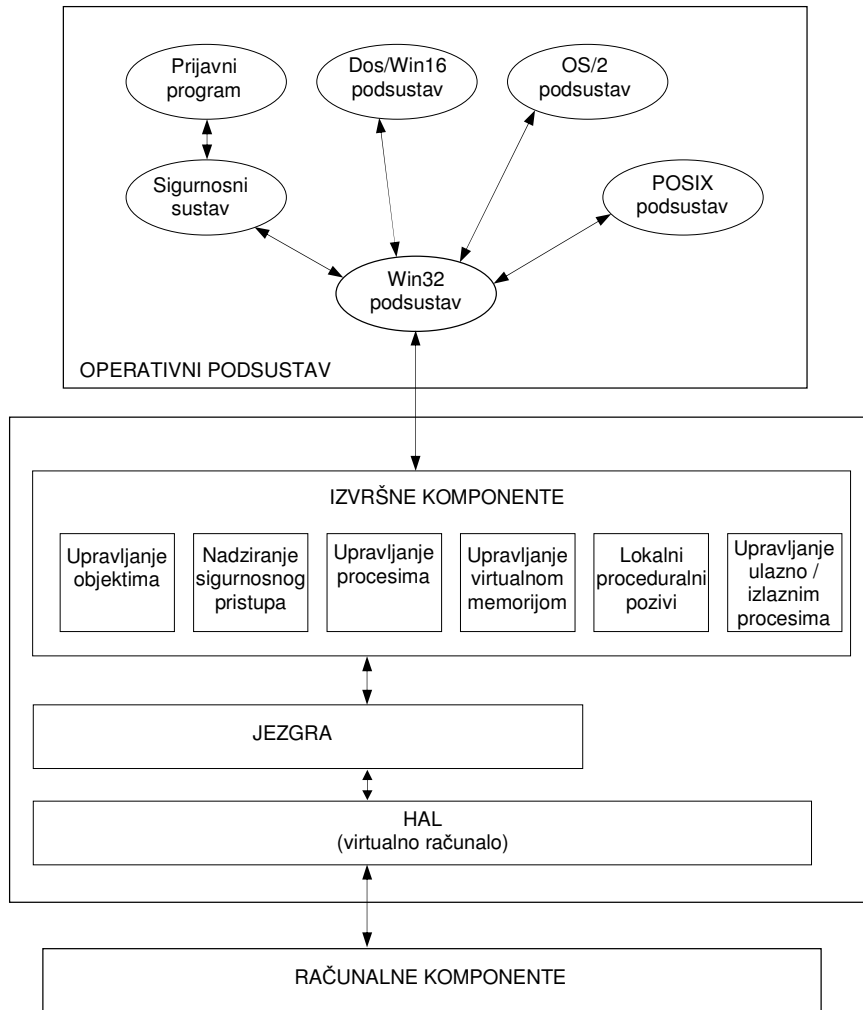
Arhitektura NT operativnog sustava prikazana je na Sl. 4-3.

NT podržava dva tipa zapisničkih sustava: FAT i NTFS. FAT (eng. "File Allocation Table") je zapisnički sustav kompatibilan DOS-u i ranijim Windows operativnim sustavima. Dozvoljava imenovanje zapisnika s najviše osam znakova i prikladan je za manje baze podataka (do 200 MB). NT zapisnički sustav (eng. "NTFS - NT File System") podržava NT sigurnosni sustav, a to znači restriktivni pristup podacima kakav nalazimo kod višekorisničkih sustava (UNIX). Omogućava rad s bazama podataka većim od milijardu GB.

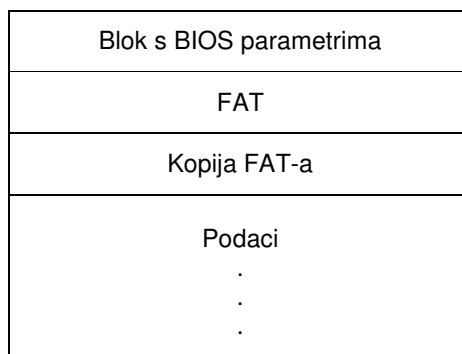
#### 4.3.1.1 FAT

FAT je zapisnički sustav koji koristi alokacijsku tablicu za pohranu i pronalaženje zapisnika. Tablica je sačinjena od popisa zapisnika. Svakom zapisniku pridružen je popis blokova ili njihovih skupina (eng. "clusters") koje zauzima na disku računala. Osnovna veličina bloka je 512 bajtova. FAT može registrirati najviše 65535 blokova u slogu od 16 bita. Stoga, uz veličinu bloka od 512 bajtova, FAT može alocirati tek 33 MB diskovnog prostora. Stoga se veći disk mora logički podijeliti na particije koje ne prelaze 33 MB ili pak primijeniti veće blokove. Windows NT automatski prilagođava veličinu bloka prilikom formatiranja diska, razmjerno njegovoj veličini, odnosno veličini logičke particije. Na Sl. 4-4 prikazana je organizacija diskovnog prostora s FAT zapisničkim sustavom.



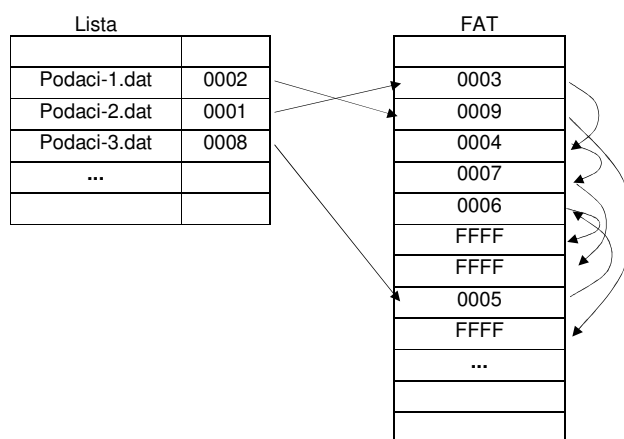


Sl. 4-3 Struktura NT operativnog sustava



Sl. 4-4 Organizacija diska s FAT zapisničkim sustavom

Osim imena zapisnika, u tablici se pohranjuju podaci o njihovoj veličini, atributima, datumu zadnje promjene i početnoj alokacijskoj jedinici (eng. "starting allocation unit"). Početna alokacijska jedinica je glavni pokazivač zapisnika u tablici. S obzirom da često nije moguće pohraniti zapisnik u kontinuiranom nizu blokova, zbog nejednolike popunjenosti diska, operativni sustav mora kreirati više pokazivača koji označavaju skupine zauzetih blokova, a nazivaju se lanci. Na Sl. 4-5 prikazan je primjer alokacijske tablice, npr. zapisnik Podaci-1.dat ima kao početnu alokacijsku jedinicu naveden blok 0002. Za blok 0002 definiran je pokazivač na blok 0009, a za njega pokazivač na heksadecimalno FFFF, što označava kraj alokacijskog lanca. Dakle, zapisnik Podaci-1.dat pohranjen je na diskovnim blokovima 0002 i 0009. Na isti način definiran je alokacijski lanac za zapisnik Podaci-2.dat 0001-0003-0004-0007, i zapisnik Podaci-3.dat 0008-0005-0006.



Sl. 4-5 Primjer alokacijske tablice

Iako je FAT početno zamišljen da prihvaća nazive zapisnika do 8 znakova, plus jedno mjesto za točku i tri za ekstenziju, Windows NT i 95 operativni sustavi omogućavaju korištenje naziva do 155 znakova.

Glavna prednost FAT-a je njegova jednostavnost i kompatibilnost s DOS i Windows 1-3.xx operativnim sustavima.

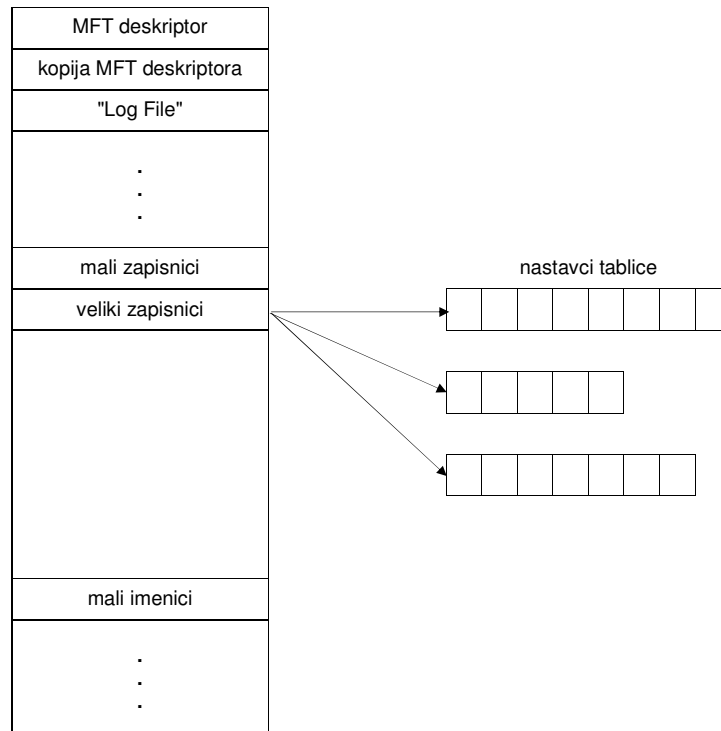
#### 4.3.1.2 NTFS

U NTFS-u podaci o zapisniku pohranjuju se sa njegovim sadržajem i zajedno predstavljaju zapisnički objekt. Osim uobičajenih podataka (veličini, datumu promjene, ...), NTFS predviđa attribute koji definiraju dozvole pristupa svakom zapisniku. To znači da je moguće kreirati sigurnosni višekorisnički sustav na zapisničkoj razini. NTFS omogućava pohranu zapisničkih naziva do dužine od 255 znakova.

Smještaj zapisnika na disku pohranjen je u tablici koja se zove MFT - eng. "Master File Table". Deskriptor tablice i njezina kopija kreiraju se u okviru "boot" područja na disku. MFT se odlikuje redundantnošću podataka koji omogućavaju automatsko

obnavljanje oštećenih zapisa. Osim toga, kopija "boot" područja s MFT-om pohranjuje se na sredini logičke diskovne jedinice. Tako se osigurava pouzdanost rada i u slučaju oštećivanja "boot" područja. Kao što je vidljivo sa slike, iza replike deskriptora zapisničke tablice slijedi "Log File" - zapis koji služi za popravljanje oštećenih zapisnika. U nastavku MFT opisuje zapisnike i imenike. Ako je zapisnik ili imenik dovoljno malen (manje od 1500 bajta), cijeli njegov sadržaj bit će pohranjen u tablicu. Na taj način pristup malim zapisnicima postaje izvanredno brz jer se u jednom čitanju prikupe svi podaci traženog zapisnika. U FAT zapisničkom sustavu alokacijski lanci se identificiraju kroz zasebne operacije čitanja. U NTFS-u svi podaci koje sadrži zapisnik ili imenik (podaci, ime, sigurnosni deskriptor i drugo) nazivaju se atributi. Kada su svi atributi jednog zapisnika pohranjeni u tablici nazivaju se rezidentni atributi. U slučaju velikih zapisnika atributi se pohranjuju u nastavcima (eng. "extentima") tablice i nazivaju se nerezidentni atributi (Sl. 4-6).

NTFS koristi posebne skrivene zapisnike u kojima čuva podatke o zapisničkom sustavu i glavnu tablicu (\$Mft, \$MftMirr, \$LogFile, \$Volume, \$BadClus, \$AttrDef, itd.).



Sl. 4-6 Glavna zapisnička tablica NTFS-a

### 4.3.2 NT jezgra

Jezgra NT operativnog sustava upravlja računarskim zadaćama.



Unutar određenog procesa naredbe se nižu u kratke logičke nizove (eng. "Threads"). Svaki proces može biti sačinjen od više nizova. Svakom nizu odgovara određeni prioritet. Veći prioritet znači prednost kod izvođenja. Ako računalo ima više procesora NT jezgra se brine o distribuciji nizova naredaba, opterećujući procesore podjednako i osiguravajući najveću moguću paralelizaciju zadaća (eng. "SMP - Symetric Multy Processing", simetrično višeprocesorsko procesiranje).

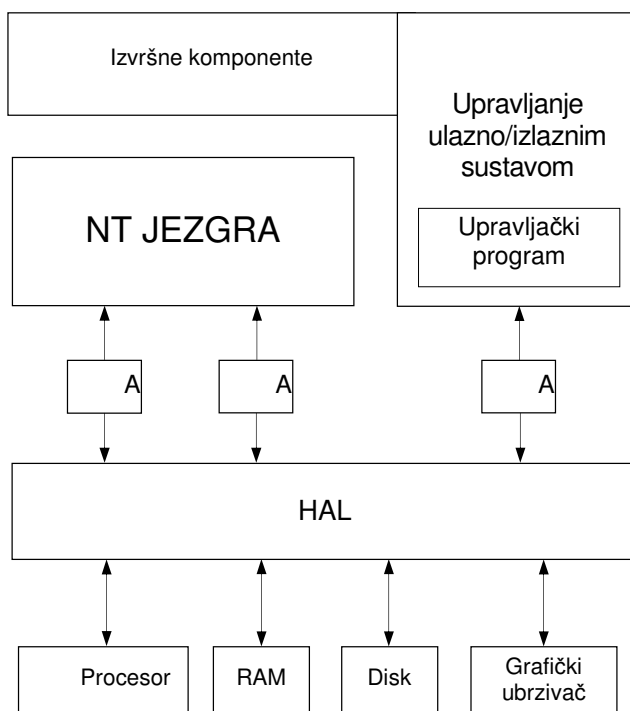
NT jezgra upravlja dvjema klasama objekata: dispečerskim i upravljačkim objektima. Dispečerski objekti služe za sinkronizaciju i odašiljanje nizova naredbi. Upravljački objekti upravljaju funkcijama NT jezgre.

### 4.3.3 HAL

HAL je virtualno računalo koje predstavlja sučelje između NT jezgre i sklopovlja.



Na taj način NT jezgra putem standardnih programskih poziva (eng. "API - Application Programming Interface", korisničko programsko sučelje), neovisno o konfiguraciji računala i vrsti komponenata, izdaje zadaće HAL-u, koji pak putem upravljačkih programa (DLL) upravlja radom računala (Sl. 4-7).



Sl. 4-7 Princip virtualnog računala

API - korisničko programsko sučelje predstavlja skup programskih poziva, odnosno standardnih funkcija, pomoću kojih se Windows programi sporazumijevaju. Upravljački programi napisani su da upravljaju određenom vrstom i tipom računalnih komponenata. Zadaće dobivaju od HAL-a putem API poziva i prevode ih u specifične upravljačke naredbe. Stoga operativni sustav ne mora znati ništa o posebnostima pojedinih komponenata. Prilikom promjene neke od računalnih komponenata dovoljno je instalirati odgovarajući upravljački program, odnosno pridružiti ga ostalim upravljačkim programima operativnog sustava. U posebnim Windows spremnicima (registrima) pohranjuje se, prilikom instalacije ili na temelju automatske detekcije, konfiguracija računala. Kod pokretanja operativnog sustava HAL učitava aktualne podatke iz spremnika i aktivira odgovarajuće upravljačke programe.

#### 4.3.4 Izvršne komponente

NT izvršni sustav upravlja procesima i memorijom.



Osigurava izvođenje svakog procesa unutar odgovarajućeg memorijskog prostora, bez međusobnog ometanja. Uključuje izvršne komponente za sljedeće zadaće:

- upravljanje objektima
- nadziranje sigurnosnog pristupa
- upravljanje procesima
- upravljanje virtualnom memorijom
- lokalni proceduralni pozivi
- upravljanje ulazno/izlaznim procesima.

##### 4.3.4.1 Upravljanje objektima

NT operativni sustav interpretira računarske pojmove (imenik, zapisnik, proces, niz naredaba, zadaća, događaj i drugo) kao objekte. Objekt opisuje njihovo stanje, parametre i pridružene metode (procedure). Metode definiraju sljedeće procese (programe, naredbe):

- kreiranje objekata
- zaštita objekata
- nadzor korištenja.

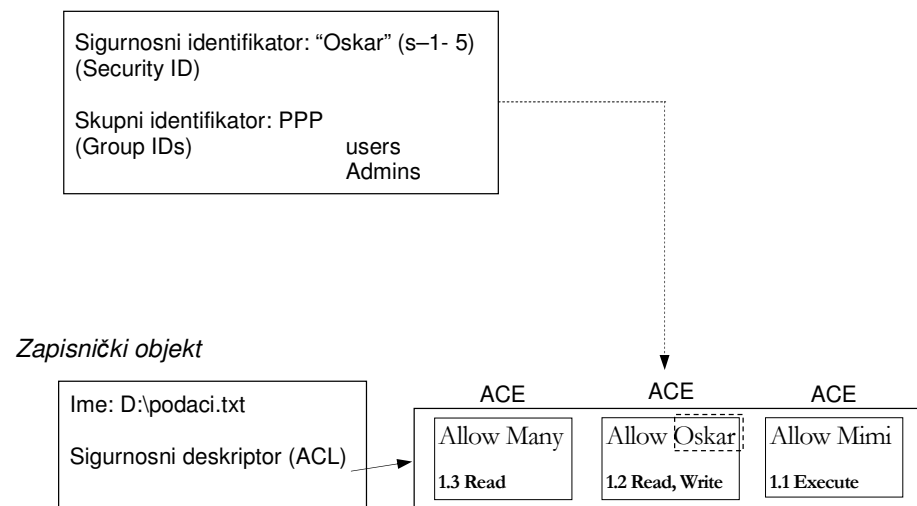
Npr. zapisnički objekt obuhvaća podatke o imenu zapisnika, njegovom statusu (raspoloživost, veličina, vrijeme nastanka itd.) i metode koje podrazumijevaju operacije kao što su kreiranje zapisnika, otvaranje, brisanje i slično.

##### 4.3.4.2 Nadziranje sigurnosnog pristupa

Pristup objektima definiran je NT sigurnosnim sustavom (eng. "SRM - Security Reference Monitor"). Sigurnosni sustav ograničava pristup računalu samo autoriziranim korisnicima, kojima je administrator definirao korisničko ime, pripadnu skupinu, korisnička prava i lozinku (eng. password). Svaki objekt posjeduje sigurnosni deskriptor ACL (eng. "Access Control List", pristupna kontrolna lista). Pomoću pristupnih kontrolnih ulaza ACE (eng. "Access Control Entry") specificira se sigurnosni identifikator SID za korisnika ili skupinu, i njima pripadajuće dozvole ili

zabrane. Kada se korisnik prijavi na računalo pridružuje mu se sigurnosni žeton SAT (eng. "Security Access Token"). Prilikom pristupa objektu (npr. otvaranje zapisnika) NT sigurnosni sustav uspoređuje identifikatore definirane u SAT-u i ACE-u te zaključuje da li je zahtijevani pristup dozvoljen (Sl. 4-8).

#### Sigurnosni žeton korisnika (SAT)



Sl. 4-8 NT sigurnosni koncept

#### 4.3.4.3 Upravljanje procesima

Osnovna zadaća komponente za upravljanje procesima je kreiranje i brisanje objekata koji su nosioci procesa. Objekti procesa definiraju odgovarajući virtualni adresni prostor, potrebne računalne komponente i nizove naredbi (eng. "threads").

#### 4.3.4.4 Upravljanje virtualnom memorijom

Komponenta za upravljanje virtualnom memorijom služi za organizaciju korištenja memorije i omogućava operativnom sustavu alociranje većeg radnog memorijskog prostora nego što je stvarno raspoloživ u RAM-u. Virtualni adresni prostor seže do 4GB. Prvih 2 GB je rezervirano za korisničke programe, a preostalih 2 GB za operativni sustav. Memorijske stranice su dimenzionirane po 4KB.

#### 4.3.4.5 Lokalni proceduralni pozivi

Podsustav okruženja poslužuje korisničke programe, komunicirajući s njima putem lokalnih proceduralnih poziva (eng. "LPC - Local Procedure Calls"). Poseban dio izvršnog podsustava brine se o interpretaciji poziva i njihovo prenošenje do razine NT jezgre.

#### 4.3.4.6 Upravljanje ulazno/izlaznim procesima

Ulazno/izlaznim procesima upravlja se putem upravljačkih programa podijeljenih na više funkcionalnih slojeva:

- ❑ Programi za upravljanje proširenim procesorskim spremnicima (Cache Manager)
- ❑ Zapisnički sustav (File System)

- ❑ Mrežni upravljački programi (Network Drivers)
- ❑ Upravljački programi računalnih uređaja (Device Drivers)

Zadaća je ove izvršne komponente koordinacija razmjene ulazno/izlaznih funkcija s odgovarajućim upravljačkim programima računalnih uređaja.

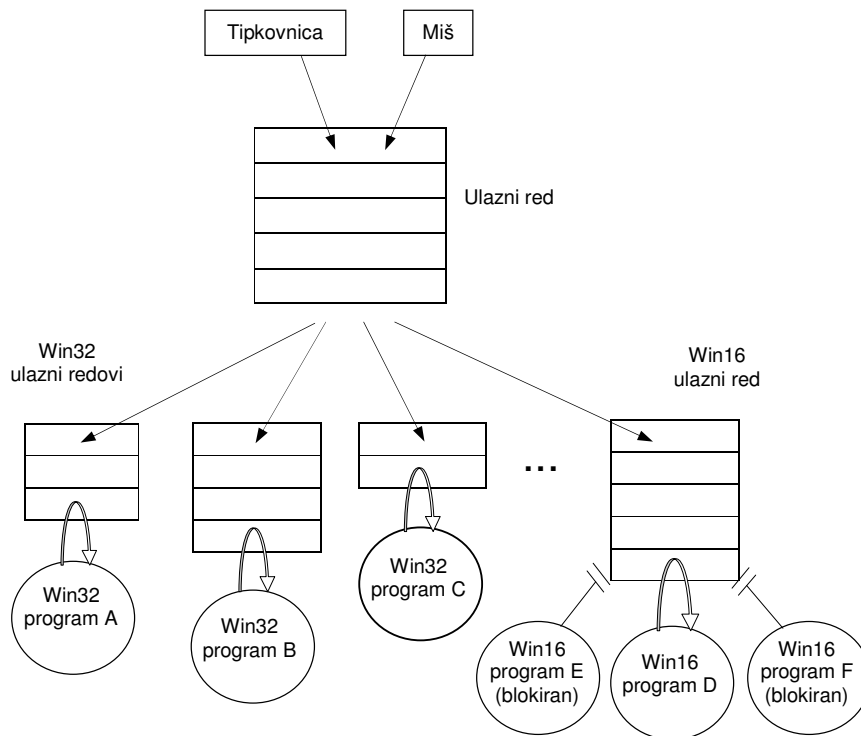
#### 4.3.5 Podsustav okruženja

U NT operativnom sustavu komunikacija s korisnikom odvija se preko 32-bitnog podsustava okruženja, odnosno Win32 podsustava.

Win32 preuzima ulazne informacije od miša i tipkovnice i brine se za prikaz izlaznih i interaktivnih informacija na zaslonu računala.



Koristi nesinkronizirani model upravljanja ulaznim porukama. Sve poruke koje se preuzimaju od ulaznih uređaja, pohranjuju se u "sirovi" ulazni niz. Potom se prenose u nizove odgovarajućih primjena. Na taj način se razdvaja utjecaj ulaznih podataka pri izvođenju različitih programa (Sl. 4-9). Npr., sinkronizirani Win16 model koristi jedan niz za sve primjene, što uzrokuje blokadu svih programa ukoliko dođe do blokade samo jednog od programa.



Sl. 4-9 Podsustav okruženja

Windows NT operativni sustav također podržava rad s programima razvijenim za prethodne verzije operativnih sustava: MS-DOS, Windows 3.x/Win16, ali i druge operativne sustave: OS/2 i POSIX kompatibilne. Podsustavi MS-DOS, Windows 3.x/Win16, OS/2 i POSIX implementirani su kao klijenti Win32 podsustava, primjenjujući Win32 korisničko programsko sučelje (API).

MS-DOS programi se izvode u okviru virtualnog DOS stroja (eng. "VDM - Virtual DOS Machine"). VDM je Win32 program koji oponaša virtualno x86<sup>15</sup> računalo s MS-DOS operativnim sustavom, odnosno interpretira skup naredaba x86 porodice Intelovih procesora.

Windows 3.x/Win16 programi su 16-bitnog karaktera, a izvode se kroz MS-DOS primjenjujući Windows 3.x grafičko korisničko sučelje. Win32 koristi poseban Win16 VDM podsustav za oponašanje Windows 3.x jezgre, programskog (API) i grafičkog (GDI) sučelja te MS-DOS-a.

#### 4.3.6 Pokretanje NT-a

Kada uključimo PC računalo prvo će se obaviti automatska dijagnostika komponenta i ako je sve ispravno aktivirat će se BIOS (eng. "Basic Input/Output Services"). BIOS je skup programa pohranjenih u PC ROM, a služe za pokretanje operativnog sustava: inicijaliziraju procesorske spremnike i memoriju te provjeravaju disk(ove) sa kojega će započeti proces učitavanja operativnog sustava (Sl. 4-10).

Pripremna faza		Programi i zapisnici
1.1	Provjera osnovnih računalnih komponenta	
1.2	Učitavanje i izvođenje MBR-a	BIOS
1.3	Učitavanje "boot" sektora	
1.4	Učitavanje NTLDR-a	
Faza učitavanja		
2.1	Prebacivanje procesora na 32-bitni radni protokol	NTLDR
2.2	Aktiviranje zapisničkog sustava	NTLDR
2.3	Izbor operativnog sustava	NTLDR, BOOT.INI
2.4	Učitavanje operativnog sustava	NTLDR, NTDETECT.COM
2.5	Identifikacija računalne konfiguracije	NTDETECT.COM
3.1	Učitavanje NT jezgre	NTOSKRNL.EXE
3.2	Inicijalizacija jezgre	NTOSKRNL.EXE
3.3	Učitavanje uslužnih programa	SMSS.EXE, AUTOCHK.EXE
3.4	Pokretanje Win32 podsustava	WINLOGON.EXE, LSASS.EXE, SCREG.EXE

Sl. 4-10 Proces pokretanja NT operativnog sustava

<sup>15</sup> Osobna računala temeljena na Intel porodici procesora 286, 386 i 486 označavaju se zajednički kao x86 računala.



Slijedi učitavanje i izvođenje "Master Boot Record" (glavnog zapisa za učitavanje programa). MBR je zapisan na tzv. "boot" područje (sektor, odjeljak) tvrdog diska i ima zadaću "naučiti" računalo kako da učita ostale programe. MBR identificira ostatak programa za učitavanje operativnog sustava. Potom se sa "boot" područja pokreće **NTLDR** program za učitavanje operativnog sustava.

Nakon ove pripremne faze započinje proces učitavanja. Program **NTLDR** će se prvo pobrinuti da mikroprocesor prebaci sa 16-bitnog na 32-bitni radni protokol. Zatim će učitati pojednostavljeni upravljački program za rad s zapisničkim sustavom (FAT ili NTFS) da bi se mogao snalaziti na disku. Sada će moći pročitati zapisnik **BOOT.INI** u kojem se, uz ostalo, nudi mogućnost zadavanja verzije operativnog sustava. Na zaslonu će se pojaviti npr.:

```

Windows NT Workstation Version 4.0
Windows NT Workstation Version 4.0 (VGA)
Windows NT Workstation Version 3.51
Windows NT Workstation Version 3.51 (VGA)

```

Nakon izbora verzije, pomicanjem bijele linije pomoću strelica na tipkovnici, **NTLDR** pokreće program **NTDETECT.COM** koji provjerava konfiguraciju računala (memorija, video kartice, miš itd.). Dobiveni podaci se pohranjuju u poseban zapisnik - "Windows registry" i prosljeđuju **NTLDR**-u. Učitavanje operativnog sustava obuhvaća četiri faze:

1. učitavanje jezgre (NT kernela),
2. inicijalizacija NT jezgre,
3. učitavanje uslužnih programa i
4. pokretanje "Windows 32" podsustava.

Učitavanje jezgre započinje kada **NTLDR** aktivira program **NTOSKRNL.EXE**, predajući mu podatke o računalo koje je dobio od programa **NTDETECT.COM**. U toj fazi učitava se "Hardware Abstraction Layer" (HAL) i Windows registri radi identifikacije upravljačkih i uslužnih programa koje će se kasnije aktivirati.

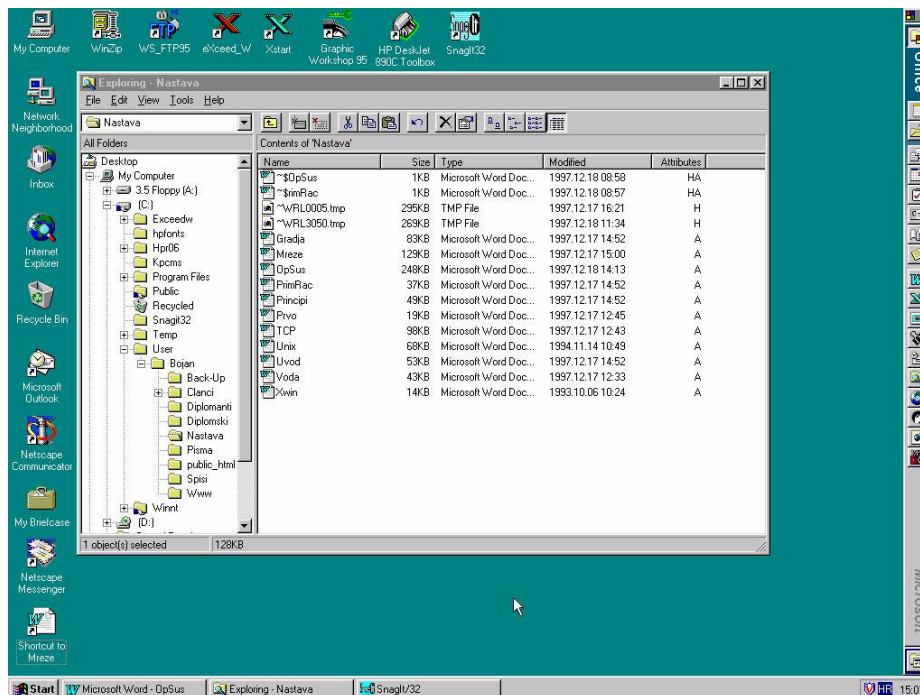
Inicijalizacija NT jezgre znači ujedno i inicijalizaciju upravljačkih programa nakon njihovog učitavanja.

Učitavanje uslužnih programa izvodi program **SMSS.EXE**, "Session Manager". Pritom se konfigurira virtualna memorija, provjerava konzistencija diskova (**AUTOCHK.EXE**) i pokreće učitavanje Win32 podsustava.

Win32 pokreće **WINLOGON.EXE**, koji aktivira program **LSASS.EXE**, lokalni sigurnosni sustav autorizacije. On će na zaslonu prikazati prijavni prozor. Pritiskom na tipke Ctrl+Alt+Del pojavljuje se prijamni obrazac. U rubriku "**Username**" upisuje se korisničko ime koje prethodno administrator računala ili domene mora autorizirati. Rubrika "**Domain**" ili "**From**" pojavljuje se samo u slučaju da je računalo konfigurirano kao korisnik lokalne mreže u domeni NT poslužioca. Ukoliko računalo sudjeluje u lokalnoj mreži kao član radne skupine (MS Windows Workgroups) prethodna će rubrika izostati. U narednu rubriku "**Password**" potrebno je upisati lozinku (1 do 14 znakova) za provjeru autoriziranog pristupa računalu. Ukoliko upisani podaci ne odgovaraju onima iz baze podataka o korisnicima (eng. "User accounts data base"), pojavit će se poruka o odbijanju pristupa: "**The device is not accessible**". Za autorizirani pristup operativni sustav će otvoriti na zaslonu korisničko sučelje.

### 4.3.7 Radni stol grafičkog sučelja

Po prijavi na ekranu se pojavljuje korisničko sučelje temeljeno na konceptu "radnoga stola" (eng. Desktop, Sl. 4-11). Kao i kod drugih sličnih grafičkih sučelja, na ekranu su raspoređeni *simboli* (ikone), a na donjem dijelu zaslona nalazi se izbornik *Start* i "linija (traka) zadataća" (eng. Taskbar).



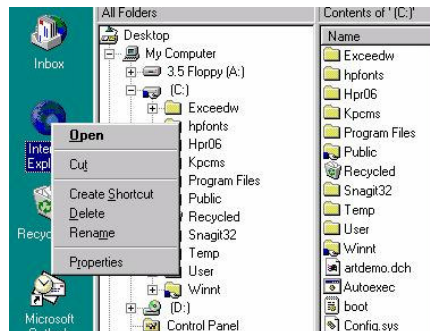
Sl. 4-11 Radni stol NT korisničkog sučelja

#### 4.3.7.1 Ikone/simboli

Ikone radnoga stola mogu prikazivati:

- programe,
- dokumente (zapisnike),
- prečice (eng. shortcut) i
- imenike (eng. folders).

Ikone uslužnih programa služe za aktiviranje programa pohranjenih na lokalnom računalu ili mrežnom poslužiocu. Postavljanjem pokazivača miša na simbol i dvostrukim aktiviranjem lijevog gumba miša pokreće se pridruženi program. Aktiviranje se može izvesti i pritiskom desne tipke miša, kada se pojavljuje izbornik kao na Sl. 4-12. Izborom opcije *Open* slijedi aktiviranje pridruženog programa. Isto je moguće označavanjem simbola i pritiskom tipke *Enter* na tipkovnici.

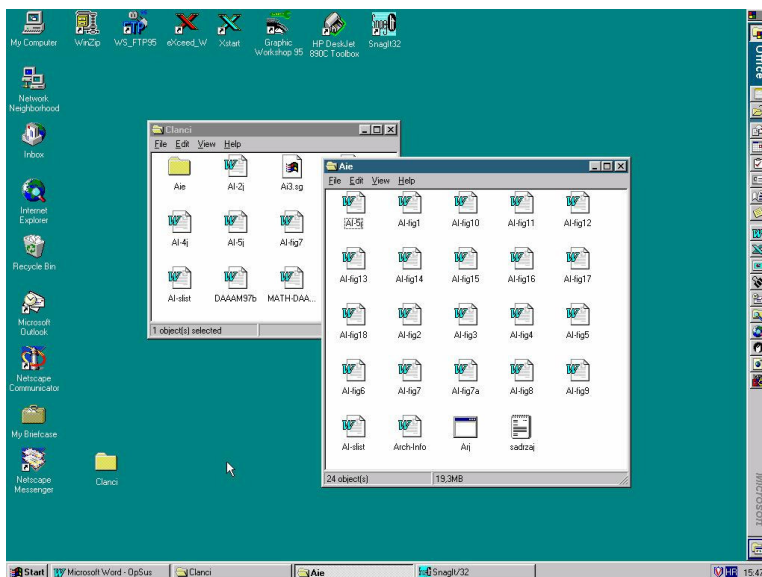


Sl. 4-12 Izbornik simbola na radnome stolu

Ikone dokumenata prikazuju zapisnike kojima je pridružen uslužni program za obradu pohranjenih podataka (npr. program za obradu teksta, slike, crteža, zvuka i drugo). Aktiviranjem simbola pokreće se odgovarajući uslužni program s učitanim podacima (tekstom, slikom, zvukom, ...).

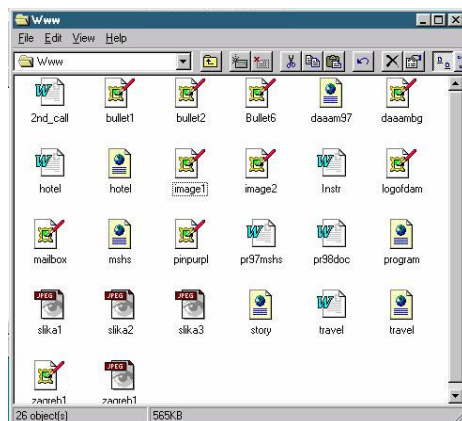
Prečice su pokazivači na zapisnike bilo koje vrste. Njihovim aktiviranjem operativni sustav pokreće odgovarajući proces. Ako prečica pokazuje na zapisnik u kojem je pohranjena slika, pokreće se odgovarajući uslužni program za obradu slika, ako pokazuje na zapisnik u kojem je pohranjen program, započinje njegovo izvođenje.

Ikone imenika omogućavaju izravan uvid u njihov sadržaj. Po aktiviranju pokreće se uslužni program koji otvara prozore s prikazom sadržaja imenika (Sl. 4-13).

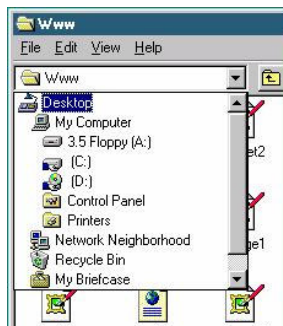


Sl. 4-13 Imenički prozori prikazuju sadržaj imenika

Radni stol je u stvari imenik koji je definiran na najvišoj razini NT zapisničkog sustava. Možemo ga pronaći tako da u imeničkom prozoru otvorimo izbornik *View* i odaberemo *Toolbar*. Prozor će promijenit izgled kao na Sl. 4-14.



Sl. 4-14 Imenički prozor s funkcijskim tipkama



Sl. 4-15 Zapisnički sustav na imeničkom prozoru

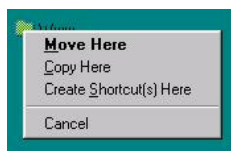
Otvaranjem izbornika s Sl. 4-15 pojavljuje se struktura zapisničkog sustava s imenikom radnoga stola. Označavanjem imenika Desktop otvara se prozor sa sadržajem radnoga stola (Sl. 4-16).



Sl. 4-16 Sadržaj imenika radnoga stola

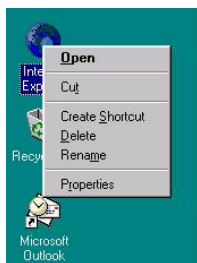
Svaki korisnik može definirati svoj osobni radni stol, njegov sadržaj i izgled. Svaka promjena osobnog radnog stola pohranjuje se u imenik **C:\Winnt\Profiles\ime-korisnika\Desktop**. U radni se stol obično pohranjuju prečice često korištenih NT objekata (programa, zapisnika i imenika). Najjednostavniji način dodavanja prečice nekog objekta na radni stol je sljedeći:

- u imeničkom prozoru označiti objekt (program, zapisnik, ...) desnom tipkom miša i držati dok se miš pomiče do pozadine radnoga stola, odnosno mjesta na kojem se želi postaviti prečica. Po otpuštanju desne tipke miša pojavit će se izbornik sa Sl. 4-17. Aktiviranjem *Create Shortcut(s) Here* na radnome stolu će se pojaviti prečica koja upućuje na odgovarajući objekt.

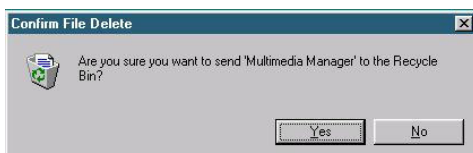


Sl. 4-17 Izbornik

Ukoliko se želi neki od objekata radnoga stola ukloniti ili preimenovati dovoljno ga je označiti s desnom tipkom miša i pojavit će se izbornik sa Sl. 4-18. U slučaju da objekt želimo ukloniti aktivirat ćemo opciju *Delete* i potom potvrditi upit o brisanju (Sl. 4-19). Isti ili sličan prozor s upitom javljat će se uvijek pri izvođenju naredaba brisanja, da se spriječi gubitak podataka zbog omaške u interakciji.



Sl. 4-18 Pomoćni izbornik



Sl. 4-19 Komunikacijski prozor za potvrdu brisanja

No, niti pogreška nakon ovog upita nije konačna, jer obrisani podaci nisu uistinu obrisani već su premješteni u "sanduk za recikliranje", odnosno poseban imenik nazvan *Recycle Bin*. Tek brisanjem sadržaja imenika *Recycle Bin* podaci se trajno uklanjaju. Brisanje sanduka za recikliranje izvodi se na sljedeći način:

- ❑ desnom tipkom miša aktivira se izbornik na simbolu imenika *Recycle Bin*, bilo na radnome stolu ili "Windows Exploreru" (Sl. 4-20),
- ❑ aktiviranjem naredbe *Empty Recycle Bin* prazni se sadržaj istoga.

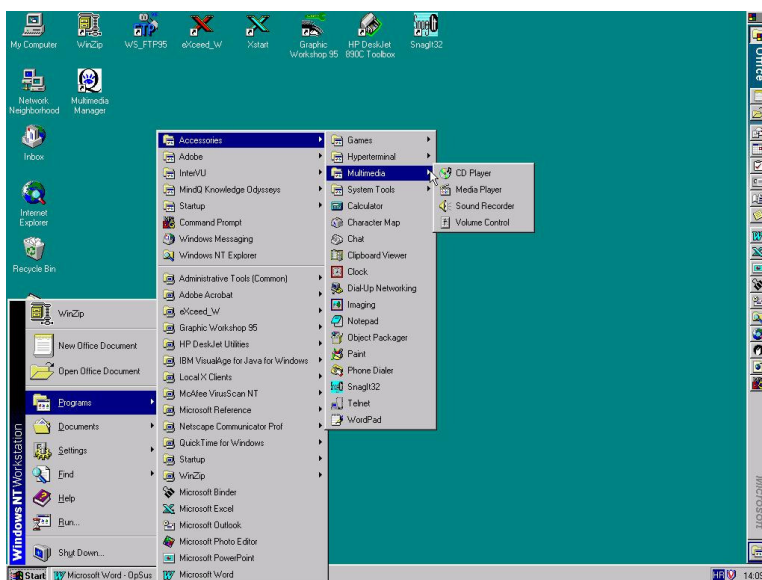


Sl. 4-20 Izbornik imenika "Recycle Bina"

Promjena imena objekta na radnome stolu izvodi se jednostavno aktiviranjem opcije *Rename* sa izbornika (Sl. 4-18). Tekst ispod simbola postat će aktivan i dovoljno je samo upisati novo ime.

#### 4.3.7.2 Izbornik "Start"

Izbornik *Start* vidljiv u lijevom donjem kutu zaslona (Sl. 4-21). Aktivira se lijevom tipkom miša. U osnovnom sadržaju uključuje opcije *Programs*, *Documents*, *Settings*, *Find*, *Help*, *Run* i *Shut Down*.



Sl. 4-21 Izbornik "Start"

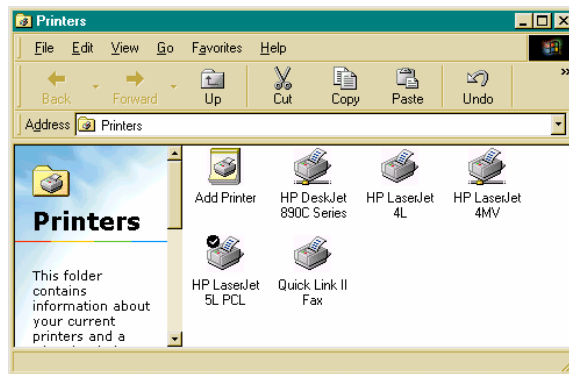
Kao što se može vidjeti na Sl. 4-21, opcija *Programs* (programi) nudi izbornike različitih instaliranih programa. Po aktiviranju tipke *Start* dovoljno je pomicati kazalo miša kroz izbornike, bez dodatnog aktiviranja lijeve tipke miša. Tek pošto je označen traženi uslužni program, pritiskom na lijevu tipku miša pokreće se izvođenje programa.

U izborniku *Documents* operativni sustav pohranjuje nazive 15 objekata (tekstova, crteža, tabličnih izračuna i slično) koji su zadnji korišteni. Izborom jednog od dokumenata pokreće se pripadajući program za njegovu obradu.

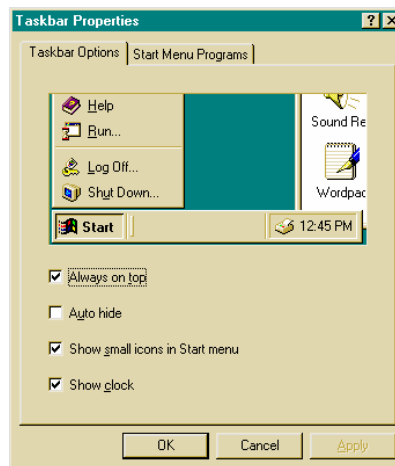
Pod izbornikom *Settings* nalaze se opcije za otvaranje sustavničkog imenika *Control Panel*, posebnog imenika za pisace *Printers* i prilagodbu linije/trake zadaća *Taskbar*. *Control Panel* otvara prozor sa programima za podešavanje parametara operativnog sustava i grafičkog sučelja, uslužnih i upravljačkih programa, instalaciju korisničkih programa i drugo (Sl. 4-22). Opcija *Printers* (pisači) služi za definiranje vrste i konfiguracije pisaca (Sl. 4-23). Za prilagođivanje izgleda linije zadaća služi opcija *Taskbar* (Sl. 4-24).



Sl. 4-22 "Control Panel" (s izgledom prozora MS Windows 98 i 2000)



Sl. 4-23 "Printers"



Sl. 4-24 "Task Bar"



Sljedeći na izborniku *Start* nalazimo izbornik *Find* (nađi) koji omogućava pretraživanje računala ili računalne mreže. Pomaže nam u potrazi za zapisnicima, imenicima, računalima ili korisnicima, koje ne znamo izravno locirati u složenoj zapisničkoj ili mrežnoj strukturi.

*Help* (pomoć) služi za izravan pristup priručnicima za pomoć u radu s NT operativnim sustavom.

Kroz opciju *Run* NT nudi mogućnost pokretanja programa bez grafičkog sučelja i interakcije. U odgovarajući obrazac upisuje se naziv programa ili naredbe i potom izbor tipke *OK*. Tipka *Browse* (razgledati) omogućava lociranje programa u zapisničkom sustavu.

Izborom *Shut Down* odjavljuje se rad na računalu, gasi ili ponovno pokreće operativni sustav. NT je mrežni operativni sustav koji stalno nadzire aktivnosti korisnika i mreže. Svaka se promjena stanja pohranjuje u radne zapisnike ili "Windows registre". Stoga, isključivanju računala mora prethoditi gašenje (eng. shut down) operativnog sustava. Pritom NT rezimira dinamičke podatke (o grafičkom sučelju, aktivnim programima i slično) i pohranjuje ih da bi se kod ponovnog pokretanja ili prijave moglo rekonstruirati posljednje stanje radnoga okruženja.

#### 4.3.7.3 Linija zadaća

Linija zadaća (eng. Taskbar) posebnost je grafičkoga sučelja NT operativnoga sustava. Glavna mu je zadaća prikazivanje aktivnih programa pomoću tipki s odgovarajućim nazivima. Jednostavnim aktiviranjem tipki usmjerava se pristup aktivnim programima. Dakako, s obzirom da je NT višezadaćni operativni sustav, fokusiranje određenog programa ne prekida rad ostalih već usmjerava interakciju. Kada na zaslonu računala imamo više prozora nosilaca različitih programa, aktiviranjem (fokusiranjem) pojedinih prozora ukazuje se operativnom sustavu s kojim programom želimo ostvariti interakciju. To se može jednostavno učiniti pozicioniranjem kazala miša na određeni prozor i pritiskom na lijevu tipku miša. No, isto se može pritiskom miša na odgovarajuću tipku linije zadaća. To je poglavito korisno kada imamo više prozora na zaslonu i većim se dijelom prekrivaju.

Linija zadaća na desnoj strani može prikazivati vrijeme i, pomoću posebnih vizualnih elemenata, aktivnost određenih programa ili parametara, npr. rad printera, vrstu slovnog skupine (HR - hrvatska slova, EN - engleska slova itd.) i drugo.

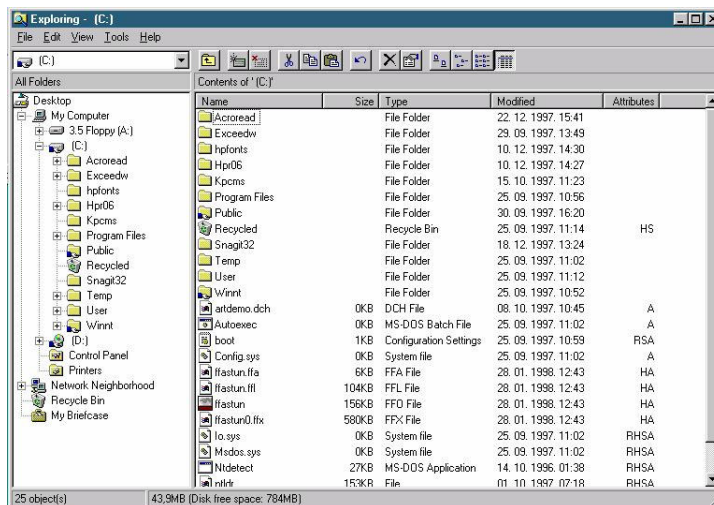
#### 4.3.8 Rad s imenicima i zapisnicima

Zapisnički sustav je ustrojstvo koje operativni sustav koristi za sređenu pohranu podataka. Windows NT operativni sustav primjenjuje hijerarhijsku imeničku strukturu koja omogućava višerazinsku pohranu podataka. Svaki imenik<sup>16</sup> može sadržavati popis zapisnika, ali i drugih imenika, koji pak mogu na isti način sadržavati popis drugih zapisnika i imenika itd. Osim imenika i zapisnika, opet slično kao i u UNIX-u, zapisnički sustav uključuje računalne uređaje (npr. disk, mrežnu karticu, pisac, i slično). Osim izravnog pristupa pojedinim imenicima pomoću imeničkih

---

<sup>16</sup> U Windows NT operativnom sustavu imenici se nazivaju mape (eng. "folders").

prozora, opisanih u prethodnom poglavlju, za rad s zapisničkim sustavom koristi se "Windows Explorer" (Sl. 4-25).

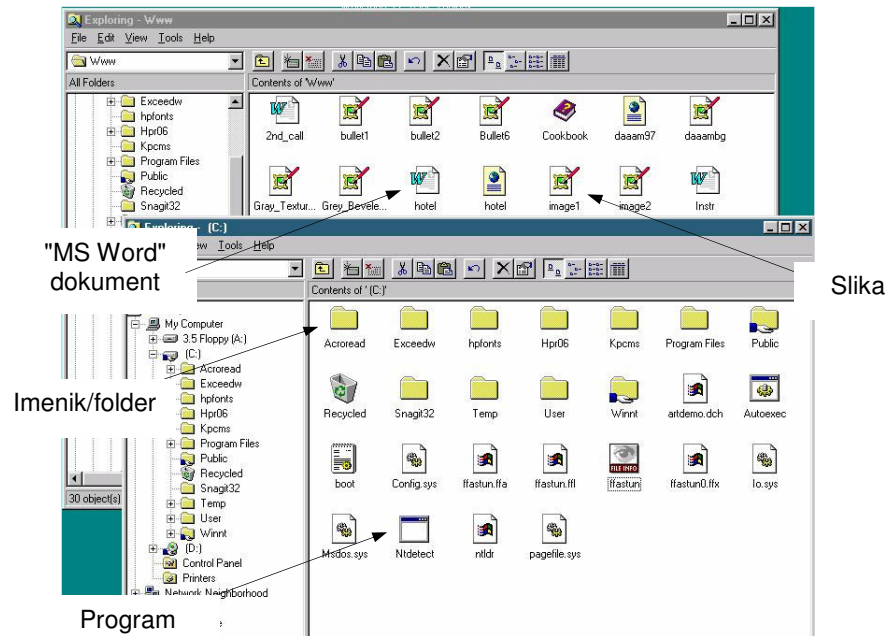


Sl. 4-25 "Windows Explorer"

"Windows Explorer" se pokreće ili preko izbornika "Start/Programs/Windows NT Explorer" na liniji zadataka ili pritiskom desne tipke miša na simbolu *My Computer* (moje računalo) odabirom opcije *Explore* na izborniku.

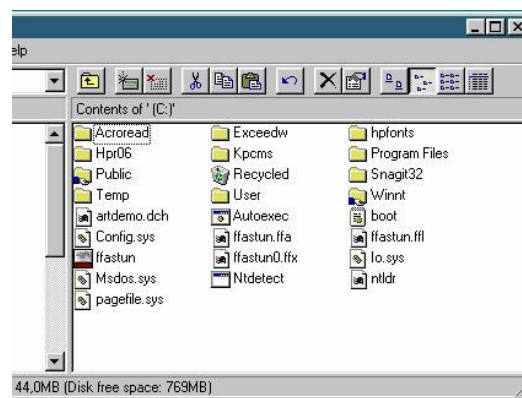
Prozor "Windows Explorera", za razliku od imeničkog prozora, podijeljen je na dvije liste ("podprozora"). Lijeva lista prikazuje strukturu zapisničkog sustava, a za označeni imenik prikazan je njegov sadržaj na desnoj listi prozora. Struktura zapisničkog sustava prikazana je hijerarhijski, potpisivanjem imenika iste razine. Pritom oznaka "+" označava da unutar imenika postoji još najmanje jedan imenik, odnosno jedna imenička razina. Isprekidanom linijom povezani su imenici iste razine. Simboli/ikone upućuju na vrstu i sadržaj objekta. Na Sl. 4-26 opisani su neki od simbola koje najčešće susrećemo.

Iznad prozora nalazi se traka s alatima i izbornicima. Na dnu prozora je statusna linija koja prikazuje dodatne informacije o označenim objektima. Statusna linija je podijeljena na dva područja. Ako se označi na lijevoj listi neki od imenika, na statusnoj će liniji biti naveden broj zapisnika koji sadrži, kapacitet koji zauzimaju i u zagradi koliko je još diskovnog prostora raspoloživo. Pritom se raspoloživi prostor odnosi na logičku jedinicu diska na kojemu je pohranjen označeni imenik (računalo mreže imati više diskova i svaki može biti organiziran kao jedna ili više logičkih diskovnih jedinica - particija).

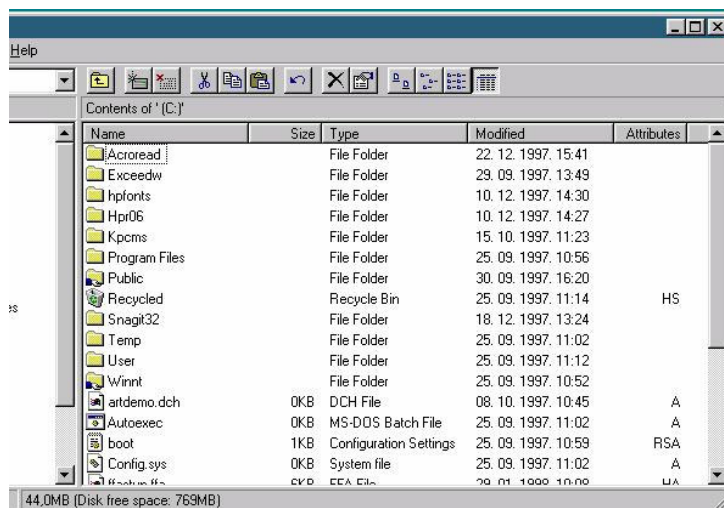


Sl. 4-26 Ikone simboliziraju vrstu pridruženog objekta

Sadržaj imenika moguće je prikazati na dva osnovna načina: velikim ili malim simbolima. Sl. 4-26 prikazuje "Windows Explorer" s velikim simbolima. Prikaz malim simbolima nudi dvije mogućnosti: mali simboli u nizovima (Sl. 4-27) ili u jednom nizu s detaljnim podacima o objektima (Sl. 4-28). Odabir prikaza postavlja se na izborniku *View* (*Large Icons* - veliki simboli, *Small Icons* - mali simboli u nizovima poredanim abecednim redom po redcima, *List* - mali simboli u nizovima poredanim po abecedi u stupce, *Details* - mali simboli s detaljnim podacima). Isto možemo postići aktiviranjem odgovarajućih tipki na traci alata "Windows Explorera" (Sl. 4-29).



Sl. 4-27 Prikaz objekata malim simbolima



Sl. 4-28 Prikaz objekata s malim simbolima i dodatnim podacima



Sl. 4-29 Gumbi za promjenu načina prikaza sadržaja u "Windows Exploreru"

Rad s objektima zapisničkog sustava izvodi se na tri načina:

- dvostrukim aktiviranjem ("klikom") lijeve tipke miša na simbol objekta
- označavanjem simbola objekta lijevom tipkom miša i povlačenjem (eng. "drag and drop")
- označavanjem simbola desnom tipkom miša i izborom funkcije s izbornika.

Dvostrukim aktiviranjem lijeve tipke miša na simbolu objekta pokreće se odgovarajući pridruženi program, jednako kao i na radnome stolu opisanom u prethodnom poglavlju. Ako je aktivirani simbol označava imenik, ispisuje se njegov sadržaj u desnoj listi prozora "Windows Explorera", ako se radi o tekstualnom sadržaju pokrenut će se program za obradu teksta i tako dalje.

Označavanjem i pomicanjem simbola možemo izvoditi operacije kopiranja, premještanja objekata unutar zapisničkog sustava, kao i radnoga stola. Možemo kreirati prečice ili pridružiti objekte određenim programima, npr. za obradu teksta, ispis na pisaču i drugo. Premještanje (eng. move) objekta izvodi se jednostavnim

pritisakom lijeve tipke miša i istovremeno tipke *Shift* na simbolu i pomicanjem (bez otpuštanja tipke) s desne liste "Windows Explorera" na željeni simbol imenika s lijeve liste. Ukoliko želimo istu operaciju izvesti na više objekata potrebno je prethodno označiti objekte lijevom tipkom miša paralelno s pritiskom tipke *Shift* ili *Ctrl*. Tipka *Shift* će nam omogućiti da označimo objekte u nizu, a *Ctrl* pojedinačno. Kopiranje objekata obaviti ćemo na isti način samo što će pomicanje simbola u imenik mora izvesti s istovremenim pritiskom na tipku *Ctrl*. Brisanje objekta iz zapisničkog sustava postiže se pomicanjem simbola objekta do simbola *Recycle Bin* na lijevoj listi "Windows Explorera" ili na pozadini radnoga stola. Iste operacije mogu se primjeniti između različitih prozora "Windows Explorera" ili imeničkih prozora. Postavljanjem kazala miša na ime simbola i sporijim<sup>17</sup>, dvostrukim pritiskom lijeve tipke miša aktivirat će se tekst koji potom možemo mijenjati.

Aktiviranjem desne tipke miša na simbolu pojavljuje se izbornik koji nudi iste mogućnosti:

- Open* (otvaranje odnosno pokretanje pridruženog programa),
- Print* (ispis pohranjenih podataka na pislač),
- Quick View* (brzo pregledavanje pohranjenih podataka),
- Send To* (premješta podatke na disketu ili poseban imenik *My Briefcase*<sup>18</sup>),
- Cut* (brisanje u *Clipboard*<sup>19</sup> spremnik),
- Copy* (kopiranje u *Clipboard* spremnik),
- Paste* (prijenos podataka iz *Clipboarda*),
- Create Shortcut* (kreiranje prečice),
- Delete* (brisanje u *Recycle Bin*),
- Rename* (promjena naziva),
- Properties* (podaci o objektu - tip, lokacija u zapisničkom sustavu, MS-DOS naziv, veličina u bajtima, datumi kreiranja, modifikacije i pristupa, atributi pristupa).

S obzirom da operativni sustav prepoznaje vrstu objekta, sadržaj izbornika neće biti jednak za sve simbole, već prilagođen vrsti pohranjenih podataka. Na primjer, ako je objekt imenik na izborniku će se pojaviti opcija *Explore*, a u slučaju tekstualnih podataka nudi se mogućnost ispisa na pislač (*Print*).

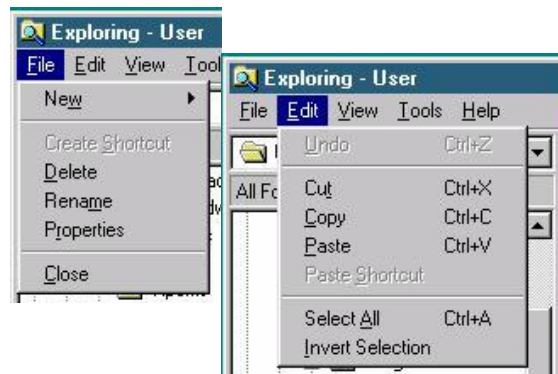
Funkcije za rad s zapisničkim sustavom nalaze se također na izbornicima *File* i *Edit*, smještenim na glavnoj liniji izbornika "Windows Explorera" (Sl. 4-30).

---

<sup>17</sup>S dužim vremenskim intervalom između dva pritiska tipke.

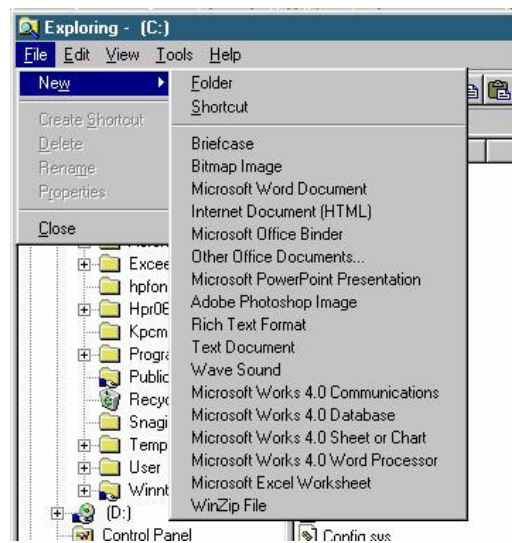
<sup>18</sup>"My Briefcase" (torba) poseban je imenik prvenstveno namjenjen za prijenos zapisnik između prijenosnih računala, osiguravajući uvijek jednu, odnosno zadnju, kopiju zapisnika.

<sup>19</sup>"Clipboard" je opći naziv za međuspremnik kojega NT operativni sustav koristi za privremenu pohranu podataka prilikom kopiranja i premještanja objekata pomoću "Copy"/"Paste" ili "Cut"/"Paste" kombinacije naredaba.



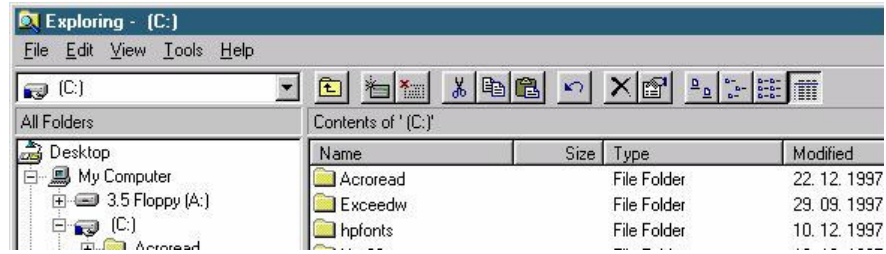
Sl. 4-30 "File" i "Edit" izbornici

Kreiranje imenika i zapisnika omogućeno je preko izbornika "File/New". Izbornik *New* otvara podizbornik (Sl. 4-31) koji nudi mogućnost kreiranja imenika (*Folder*), prečica (*Shortcut*) i zapisnika različitih formata (*Microsoft Word Document*, *Wave Sound*, *Bitmap Image*, ...).



Sl. 4-31 "File/New" izbornik

Osnovne funkcije nalaze se na raspolaganju i u obliku tipki na traci alata prozora "Windows Explorera" (Sl. 4-32).



Sl. 4-32 Linija alata "Windows Explorera"

Pored "Windows Explorera" operativni sustav Windows NT integrira brojne korisničke alate, kao što su programi za kreiranje datoteka ("Notepad", "WordPad"), za kreiranje i obradu slika ("Paint", "Imaging"), računanje ("Calculator"), korištenje multimedijalnih podataka ("CD Player", "Sound Recorder"), priključivanje i rad u računalnoj mreži ("Dial-Up", "Telnet", "Internet Explorer") i drugo.





---

## 5 RAČUNALNE MREŽE

---

**I**nformacija kao objekt računarstva podrazumijeva obradu informacija, koja jednako uključuje transformaciju i prijenos podataka. Dakle, *informacija podrazumijeva komunikaciju*, jer ona predstavlja glavni smisao postojanja informacija.

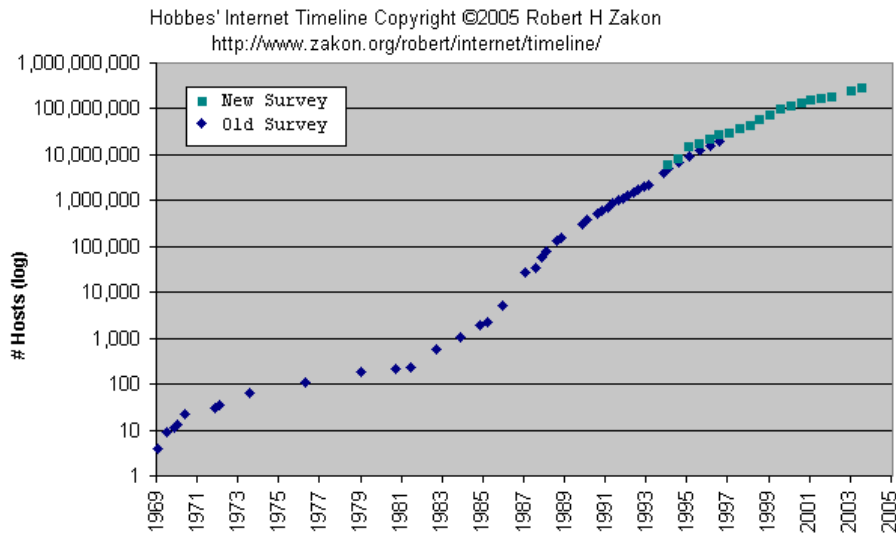
Računala su isprva predstavljala informatičke otoke, unutar kojih su izdvojeno kolale informacije i završavale na papirima, magnetskim trakama ili terminalskim zaslonima. Prijenos podataka s jednog računala na drugo svodio se na ponovni unos ili učitavanje s magnetske trake. To je bilo sporo i mukotrpno, a nesavršenost i neusklađenost uređaja i medija za pohranu često su stvarala nepremostive poteškoće.

Intenzivnija istraživanja i razvoj tehnologija povezivanja računala i odgovarajućih uređaja u računalne mreže započeli su šezdesetih godina u SAD, u Rand Corporation, na inicijativu Ministarstva obrane (US Defense Department), kao projekt ARPA<sup>20</sup> (eng. "Advanced Research Projekts Agency"). Pod okriljem tog projekta ili na njegov poticaj radala su se revolucionarna tehnološka rješenja koja su otvorila vrata budućoj informatičkoj integraciji. Da bi se ostvarila asinkrona i paralelna komunikacija više računala, od presudnog je značaja bio razvoj tehnologije prijenosa dijeljenjem podataka na manje cjeline - pakete, tzv. "packet-switching" tehnologija, zadržavši se aktualnom sve do danas. Prva računalna mreža Arpanet uspostavljena je 1969. godine, a prvi "čvor" instaliran je u Los Angelosu na University of California. Arpanet mreža je stalno povećavala broj korisnika kao i mogućnosti primjene mreže, uvodeći elektroničku poštu, podršku umrežavanju zapisničkih i drugo. Predstavljala je glavni katalizator razvoja drugih mreža i konačno današnjeg Interneta. TCP/IP (eng. "Transmission Control Protocol/Internet protocol"), protokol na kojem se bazira današnji Internet, mreža svih mreža, razvijen je u okviru Arpaneta u prvoj polovici osamdesetih godina.

Paralelno s Arpanet mrežom, US National Science Foundation pokreće 1986. godine projekt razvoja mreže NSFnet, koja je imala za cilj povezivanje nacionalnih superračunalnih centara. Ubrzo je postala glavnom mrežnom infrastrukturom povezujući lokalne mreže (eng. internetwork, međumreža) nazvanom Internet. Najzad, 1990. godine Arpanet se gasi u zamjenu za novu globalnu mrežu Internet, koja najzad ujedinjuje sve ostale mreže. Od 1990. godine do danas broj umreženih računala na Internetu eksponencijalno se povećava, od 500.000 1990. godine do više stotina milijuna računala u 2000. godini (Sl. 5-1).

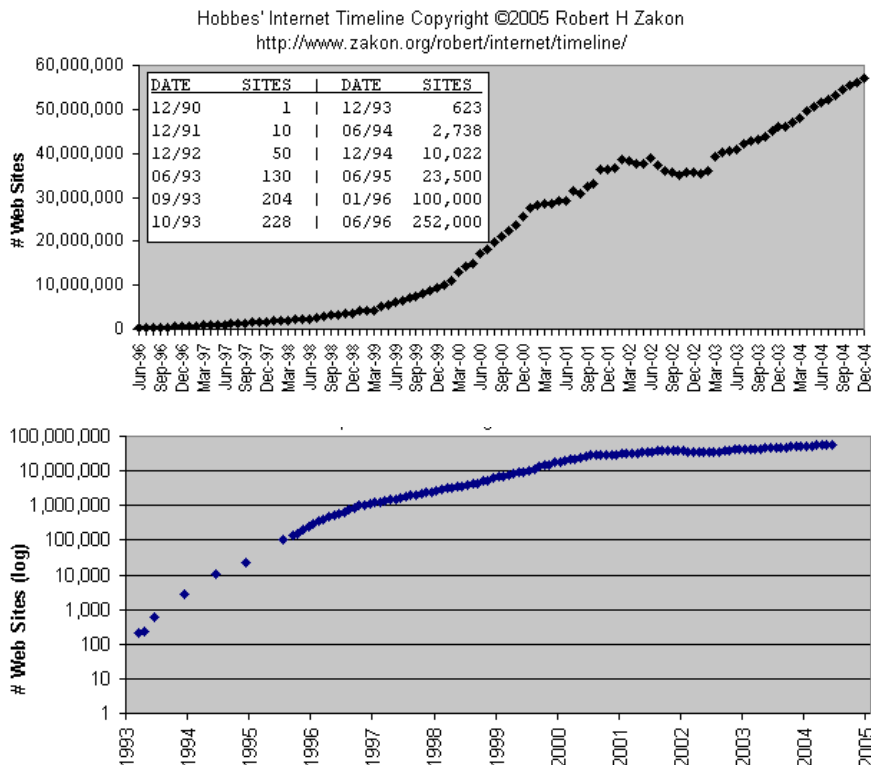
---

<sup>20</sup> Isti se projekt naziva i DARPA kao kratica za eng. "Defense department's Advanced Research Projekts Agency"



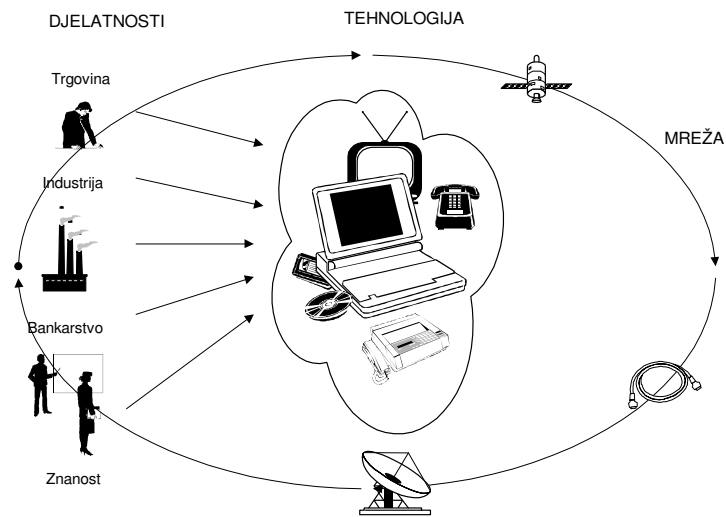
Sl. 5-1 Trend razvoja Interneta kroz povećanje broja korisnika

Arpanet i Internet su utemeljili sustave za prijenos podataka na lokalnoj i globalnoj razini. Potonji razvoj "WWW - World Wide Weba", odnosno HTTP protokola (eng. Hyper Text Transfer Protocol), i kasnije JAVA koncepta, omogućio je razmjenu i obradu multimedijalnih informacija (teksta, slike, tona, programa). Bogatstvo podataka i usluga koji se putem WWW poslužilaca i preglednika mogu razmjenjivati Internetom potaklo je neslućeno omasovljenje primjene računalnih mreža i općenito globalizaciju informacijskih resursa (Sl. 5-2).



Sl. 5-2 Trend razvoja Interneta kroz povećanje broja web sjedišta

Unatoč stalnom unapređenju mrežnih tehnologija i infrastrukture (svjetlovodni provodnici, brzi usmjerivači, satelitska komunikacija), nove primjene uvijek iznova dostižu raspoložive kapacitete. Suvremene računalne mreže uspostavile su komunikaciju kao središnji aspekt računarstva, snažno utječući na opći tehnološki razvoj i društvene promjene. Visoke tehnologije u inženjerstvu, medicini i drugim znanstvenim područjima izgradile su nove pristupe usko povezane s novim komunikacijskim potencijalima: istovremeno inženjerstvo, udaljene kirurške operacije, itd. Danas inženjeri udaljeni kontinentima mogu istovremeno kreirati složeni proizvod na računalu, usklađujući njegove komponente i tehnološke parametre putem zajedničkog/umreženog projektnog prostora i zvučno/vizualnom komunikacijom, kao da rade jedan pored drugoga za istim računalom. Snažni potencijali prijenosa multimedijalnih podataka omogućavaju istovremenu obradu podataka na udaljenim računalima, globalnu informacijsku integraciju, privid prostorne integracije i simulaciju udaljene ili nepostojeće stvarnosti. To se odražava na stvaranje informacijskog konvergencijskog trenda. Konvergencija znači integraciju medija i tehnologija: računarstvo, televizija, radio, novine, telekomunikacije, zabavna elektronika; kao i integraciju djelatnosti: izdavaštvo, trgovina, bankarstvo zabavna industrija i drugo (Sl. 5-3), pretvarajući računalno u univerzalni informacijski stroj.



Sl. 5-3 Konvergencija informacijskih tehnologija



## 5.1 OSI referentni model

Računalna mreža je skup računala povezanih komunikacijskim sklopovljem i odgovarajućom programskom podrškom.



**Komunikacijsko sklopovlje** uključuje mrežna sučelja na računalima, usmjerivače i provodnike.

**Mrežna programska podrška** podrazumijeva operacijski sustav i mrežne uslužne programe koji omogućavaju prijenos i interpretaciju podataka putem komunikacijskog sklopovlja.

**Čvorovi** u mreži mogu biti računala ili druge mreže. S obzirom da čvorovi općenito mogu poticati od različitih proizvođača, da mogu koristiti različite procesore, operacijske sustave i uslužne programe, potrebno je definirati sustave zajedničkih pravila - protokole putem kojih različita računala uspostavljaju komunikaciju "govoreći isti jezik".

**Protokol** je skup pravila za komunikaciju, dakle konvencija prema kojoj se komunikacija obavlja.

Sustav protokola mora definirati jednako fizičke i logičke aspekte računalne komunikacije. Danas je prema Međunarodnoj organizaciji za standarde (eng. "ISO - International Organization of Standards") usvojen OSI (eng. "Open System Interconnection" - otvoreni sustav povezivanja) referentni model za razvoj standardnih mrežnih protokola. OSI uspostavlja i klasificira zadaće potrebne za ostvarivanje računalne komunikacije. Podijeljene su na sedam razina (Sl. 5-4). Svaka razina predstavlja zasebnu zadaću posebne namjene i može se samostalno obaviti. Zadaće na susjednim razinama, koje se međusobno nadovezuju, korespondiraju kao apstraktne funkcije.

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Sl. 5-4 OSI referentni model

Svaka razina obavlja određeni dio posla i osigurava usluge za višu razinu. Tako će npr. fizička razina ("Physical") brinuti za prijenos signala fizičkom vezom. Na toj razini se uspostavlja električka veza bez interpretacije sadržaja signala. "Data Link" razina ima zadaću prepoznati u signalima pakete podataka. Sljedeća "Network" razina preuzima pakete od "Data Link" razine itd. Za potpunu komunikaciju sva računala u mreži moraju implementirati sve razine komunikacija. Važno je uočiti da na udaljenim računalima komuniciraju odgovarajuće razine (npr. "Session" razina na jednom računalu komunicira s odgovarajućom "Session" razinom na drugom računalu).

OSI predstavlja samo model procesa komuniciranja, i ujedno nudi predložak za formiranje konkretnih mrežnih protokola. OSI model ne definira konkretan komunikacijski protokol već definira razine i funkcije na pojedinim razinama koje bi protokoli trebali zadovoljavati.

**Fizička razina** ("Physical") definira mehanička, električka, funkcionalna i proceduralna svojstva medija za prijenos. Drugim riječima to znači definirati dimenzije priključaka i raspored pinova (mehanička svojstva); dozvoljeni naponi (električka svojstva); značenje pojedinih signala (funkcionalna svojstva); dozvoljeni redoslijed signala (proceduralna svojstva). Osnovna funkcija ove razine je prijenos niza bitova.

**Razina podatkovne veze** ("Data Link") mora osigurati konzistenciju prijena podataka. Na toj se razini otkrivaju greške i otklanjaju komunikacijske smetnje. Identifikacijom paketa (okvira) na "Data Link" razini, čija osnovna jedinica može biti oktet ili bit, provjerava se stanje kontrolnih polja kod prijema. Okviri se dijele na adresno polje, kontrolno polje, podatkovno polje i kontrolno polje grešaka. *Adresno polje* definira primaoca paketa. *Kontrolno polje* sadrži podatak o vrsti paketa. U *podatkovnom polju* nalaze se podaci koji su predmet komunikacije. Na temelju podataka iz *kontrolnog polja grešaka* "Data Link" razina ispituje valjanost paketa i ako je došlo do greške u prijenosu može se zatražiti ponovni prijenos paketa.

**Mrežna razina** ("Network") osigurava usmjeravanje (eng. "routing") između udaljenih računala kroz mrežu. Ako dva računala pripadaju različitim lokalnim mrežama između njih moraju postojati mrežni uređaji koji ih povezuju. Isti uređaji mogu povezivati i mnoštvo drugih mreža, usmjeravajući pakete na odredišta. S obzirom da paketi mogu stizati iz mreža različitih topologija (npr. Token Ring i Ethernet), koji imaju različite oblike adresiranja, na mrežnoj se razini mora osigurati jedinstveni mehanizam adresiranja, npr. protokoli IPX<sup>21</sup> (eng. "Internet Packet Exchange") ili IP<sup>22</sup> (eng. "Internet Protocol").

**Prijenosna razina** ("Transport") mora osigurati prijenos podataka između komunikacijskih programa na udaljenim računalima, za razliku od prethodnih razina koje definiraju protokole za uspostavljanje komunikacije između računala, mrežnih sučelja i/ili računalnih mreža. Budući da više programa ili komunikacijskih procesa može ostvarivati vezu putem iste mrežne adrese, na ovoj se razini definira prijenosna adresa koja nadopunjuje mrežnu adresu s jedinstvenim identifikacijskim brojem programa (eng. "socket" ili "port number"). Primjer protokola prijenosne razine su TCP (eng. "Transport Control Protocol") i SPX (eng. "Sequenced Protocol of Exchange"). Ujedno, na ovoj se razini osigurava pouzdanost u prijenosu paketa. Paketi moraju na odredište doći bez greške, točno u nizu u kojem su poslani i ne smije doći

<sup>21</sup> Razvijen u Novellu.

<sup>22</sup> Razvijen u Ministarstvu obrane SAD (DoD - Department of Defense).

do ponavljanja paketa. Stoga paketi mogu sadržavati slijedni broj na temelju kojega se može ustanoviti potpunost primljenih podataka.

**Razina sastanka/susreta** ("Session") proširuje funkcije prijenosne razine, upravljajući procesom razmjene podataka između udaljenih procesa. Komunikacija (sastanak) između dva udaljena procesa se može odvijati kao dvosmjerni ili jednosmjerni dijalog, a komuniciranje može biti istovremeno ili naizmjenično; poruke se mogu prenositi kao normalne ili kao hitne itd.. Ova razina je također zadužena za postavljanje točaka provjere ili točaka sinkronizacije. Npr., ukoliko tijekom prijenosa podataka između dva računala dolazi do učestalog prekida veze, svako novo uspostavljanje veze moglo bi značiti ponovni pokušaj prijenosa. Točke provjere mogu poslužiti za utvrđivanje posljednjeg stanja u prijenosu s mogućnošću da se nastavi od zadnjeg zaprimljenog paketa, sinkronizirajući tako mrežne usluge u intermitentnom radu. Protokol pod zajedničkim nazivom TCP/IP implementira također zadaće razine sastanaka. U Windows 95 ili NT operacijskom sustavu nalazimo "NetBIOS" protokol (eng. "Network Basic Input/Output System") kao tipičan primjer protokola razine sastanaka.

**Razina prezentacije** ("Presentation") upravlja prikazom podataka. Brine se o poslovima kao što su sažimanje i raspakiranje podataka, pretvorbe grafičkih prikaza i općenito pretvorbe podataka različitih oblika koji se prenose. Ova razina mora osigurati ispravni prijem podataka neovisno o međusobnim razlikama udaljenih računala. SNMP (eng. "Simple Network Message Protocol") je tipičan protokol za dekodiranje paketa na razini prezentacije.

**Razina primjene** ("Application") je najviša razina u OSI modelu i predstavlja mrežno sučelje prema korisniku. Prethodne razine tek su u funkciji primjene. Korisnik ne mora biti svjestan procesa koji se odvijaju na tim prethodnim razinama. Na razini primjene definiraju se usluge i protokoli po kojima komuniciraju mrežni programi kao što je npr. elektronička pošta (sendmail), prijenos zapisnika (ftp), prijava na udaljeno računalo (telnet), distribucija zapisničkog sustava (nfs), udaljeno izvođenje programa (rsh) i mrežne usluge kao što su npr. "World Wide Web" (http), "Gopher" i slično.

Ukratko rečeno svaka razina OSI modela definira *usluge* koje se trebaju davati na toj razini, i *nacrtni protokoli* po kojem će se odvijati komunikacija s istom razinom na udaljenom računalu. Funkcije pojedinih razina u OSI modelu su jasno definirane, ali u realizaciji stvarnih mrežnih arhitektura (protokola) dolazi do odstupanja, preklapanja ili dupliciranja pojedinih funkcija.





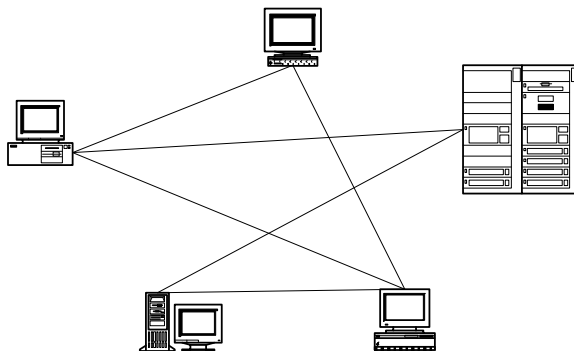
## 5.2 Topologija računalnih mreža

Topologija mreže je geometrijsko uređenje čvorova i veza. Ovisi o veličini, namjeni i tehnologiji mreže, kao i udaljenostima povezivanja.



Tehnologija mreže podrazumijeva komunikacijsku tehnologiju i protokol. Računalne se mreže u osnovi dijele na globalne mreže (eng. "Wide Area Network - WAN") i lokalne mreže (eng. "Local Area Networks - LAN").

Globalne mreže povezuju računala i lokalne mreže na većim udaljenostima (preko 20 km). Pritom se koristi globalna komunikacijska infrastruktura (telefonski provodnici, sateliti ili namjenski provodnici). Topologija globalne mreže u pravilu je nepravilna, osiguravajući izravno ili posredno međusobno povezivanje (Sl. 5-5).



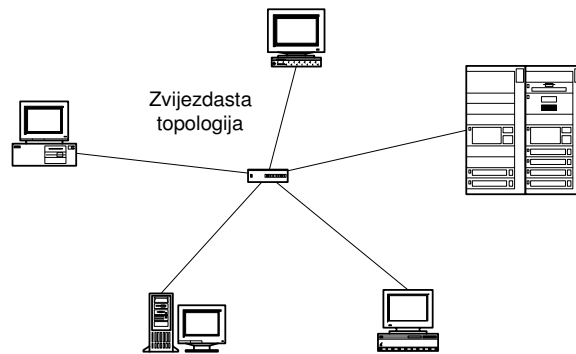
Sl. 5-5 Nepravilna topologija

Čvorovi u globalnoj mreži su posebna mrežna računala, usmjerivači ili modemi, koji na različitim hijerarhijskim razinama povezuju računala, globalne podmreže i lokalne mreže. U OSI referentnom modelu definiraju se na trećoj ("Network") razini. Brzine prijenosa podataka ovise primijenjenim komunikacijskim tehnologijama i kapacitetima komunikacijskih sučelja: od 32 KB/s (modem) do više stotina MB/s (ATM), a kapaciteti središnjih komunikacijskih čvorova veći su od 1 GB/s. S obzirom da se ukupni kapaciteti globalnih mreža dijele na veliki broj potencijalnih korisnika, brzine prijenosa podataka između pojedinih korisnika uveliko variraju ovisno o opterećenju.

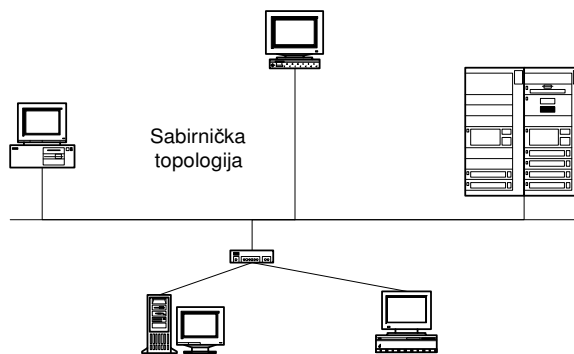
Lokalne mreže povezuju računala na pojedinim prostorno bliskim lokacijama (manje od 20 km). Temelje se na vlastitim provodnicima i u pravilu osiguravaju veće brzine prijenosa podataka (od 10 MB/s do 660 MB/s). Primjenjuju tri osnovne *logičke topologije*:

- ❑ zvjezdastu,
- ❑ sabirničku i/ili
- ❑ prstenastu.

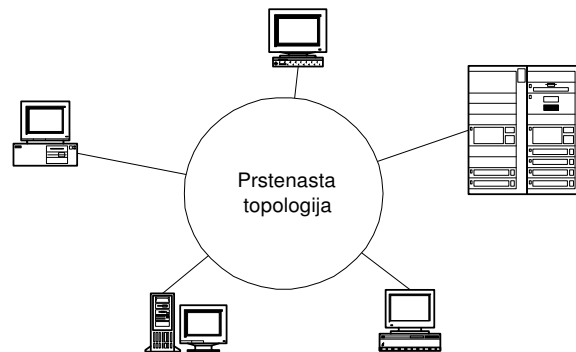
U OSI referentnom modelu pripadaju prvoj ("Data Link") i drugoj ("Physical") razini normizacije.



Sl. 5-6 Zvezdasta topologija



Sl. 5-7 Sabirnička topologija



Sl. 5-8 Prstenasta topologija

U **zvjezdastoj topologiji** (Sl. 5-6) komunikacija između računala odvija se kroz središnji preklonni mrežni uređaj. Njegova je zadaća odvajanje i preusmjeravanje komunikacijskih signala između odgovarajućih računala. Pritom se može odvojiti promet između računala ili podmreža, odnosno promet između bilo koja dva računala ili komunikacija unutar podmreže odvija se neovisno o komunikaciji ostalih učesnika u mreži. To znači da dva računala mogu razmjenjivati podatke kroz jedinstveni komunikacijski kanal čiji je kapacitet neometan od ostalih računala u lokalnoj mreži. Podjelom mreže na manje cjeline postiže se strukturalna fleksibilnost i bolja iskoristivost komunikacijskih resursa jer se promet unutar jedne podmreže ne dijeli s ostalim dijelovima lokalne mreže. Na taj način se osiguravaju postojeće brzine prijenosa podataka pa se stoga ova topologija danas najčešće koristi. Nedostatak zvjezdaste topologije je osjetljivost na kvar središnjeg preklonnog uređaja. S obzirom da sva računala iz lokalne mreže ostvaruju komunikaciju preko njega, kvar će prouzročiti otkazivanje cijele mreže. Nadalje, sve je provodnike potrebno dovoditi do jednog središnjeg mjesta, što u slučaju kada su računala prostorno raspršena može poskupiti troškove instalacija. Zvjezdastu topologiju koriste tehnologije Ethernet, ATM i FDDI (eng. "Fiber Distributed Data Interface"). Posebnim standardima za svaku je tehnologiju propisan odgovarajući komunikacijski protokol i način fizičkog povezivanja (provodnici i sučelja).

**Sabirničku (paralelnu) topologiju** (Sl. 5-7) karakterizira središnji provodnik na koji se putem spojnih sklopova (eng. "transceivera") povezuju računala u nizu. Takva lokalna mreža je jeftina, no, budući da sva računala dijele kapacitet središnjeg provodnika putem kojeg zajednički emitiraju informacije, nije prikladna za mreže s velikim brojem aktivnih računala. Da računalo otkrije koji je od komunikacijskih signala što prolaze središnjim provodnikom adresiran na njega, mrežno sučelje mora stalno preispitivati ukupan promet lokalne mreže. Ako veliki broj računala komunicira istovremeno može doći do zagušenja mreže, odnosno značajnog usporavanja prijenosa podataka. Ethernet tehnologija tipičan je primjer sabirničke topologije, bilo da je temeljena na standardnom "debelom" provodniku (10BASE5<sup>23</sup>) ili pak na "tankom" koaksijalnom provodniku (10BASE2). S obzirom na cijenu i jednostavnost umrežavanja Ethernet je danas najraširenija mrežna topologija. Osigurava kapacitet prijenosa podataka do ukupno 10 Mb/s, a novi "Fast Ethernet" i do 100 Mb/s.

Značajka **prstenaste topologije** (Sl. 5-8) je povezivanje računala u zatvorenu petlju. Signali se šalju u jednom smjeru, od jednog do drugog računala u zatvorenom nizu. "Token Ring" tehnologija se temelji na prstenastoj topologiji. Računala mogu biti povezana koaksijalnim, svjetlovodnim ili TP provodnicima. Kao što ime ove tehnologije govori, njena glavna značajka je "token" - logički nosač podataka (24 bita), koji kruži u mreži konstantnom brzinom. Na taj način se osigurava nepromjenjiva brzina odziva, odnosno kapacitet prijenosa podataka (1 Mb/s, 4 Mb/s ili 16 Mb/s). Često se fizički izvodi zvjezdasto, no iako sva računala ostvaruju komunikaciju preko središnjeg uređaja, logika rada je ista.

Opisane topologije i odgovarajuće mrežne tehnologije mogu se pomoću različitih pretvornika kombinirati te primjenjivati sukladno potrebama. Na primjer, snažan kapacitet ATM tehnologije primijenit ćemo na većim udaljenostima, povezujući više

---

<sup>23</sup> Prema IEEE 802-3 normi koji definira Ethernet tehnologiju (eng. "IEEE - Institute of Electrical and Electronic Engineers", međunarodna inženjerska udruga)

manjih lokalnih mreža temeljenih na jeftinom Ethernetu. Ethernet i ATM preklopnice odvajaju se lokalni promet na mreži i osiguravaju nesmetano i učinkovito korištenje pojedinih mrežnih instalacija.

### 5.2.1 Ethernet

Ethernet protokol definira fizičke karakteristike medija za prijenos podataka (koaksijalni kabel, svjetlovod ili upletene parice). Ethernetom su također definirana električka svojstva signala, oblik podatkovnih paketa koji putuju Ethernet medijem i komunikacijski protokol.



Format Ethernet paketa je prikazan u Tab. 5-1 (brojevi ispod pojedinih polja označavaju dužinu polja u oktetima):

Tab. 5-1 Format Ethernet paketa

Preambula	Adresa odredišta	Adresa polazišta	Tip	Podaci	Kontrolni niz
Preamble	Destination address	Source address	Type	Data	Check seq.
8	6	6	2	46-1500	4

Osim Ethernet protokola na tržištu je vrlo raširena njegova modifikacija koja se naziva IEEE 802.3. Većina mrežne opreme koja se radi sa Ethernet protokolom podržava obje varijante. Format IEEE 802.3 paketa prikazan je u Tab. 5-2:

Tab. 5-2 IEEE 802.3 Ethernet format

Preambula	Odjeljivač	Adresa odredišta	Adresa polazišta	Dužina	Podaci	Kontrolni niz	
Preamble	Start frame delimiter	Destination address	Source address	Length	Data	Pad	Check seq.
7	1	2 ili 6	6	2	46-1500		4

#### 5.2.1.1 Komunikacijski protokol

Ethernet je temeljen na sabirničkoj tehnologiji, a to znači da sva računala na lokalnoj razini dijele isti provodnik. Zato se samo jedan paket (okvir podataka) može emitirati u jednom vremenskom intervalu. Stoga Ethernet koristi mehanizam CSMA/CD koji sprečava računala da međusobno interferiraju. Naziv CSMA/CD dolazi od eng. "*Carrier Sense Multiple Access with Collision Detection*". Taj se mehanizam pojednostavljeno naziva "slušaj prije nego što govoriš".

"Carrier Sensing" označava mehanizam slušanja. Kada računalo želi emitirati pakete kroz Ethernet mrežu, prvo mora ispitati da li je mreža zauzeta slanjem podataka od nekog drugog računala. Ako jest, Ethernet uređaj računala će nakratko pričekati te ponovno ispitati stanje zauzetosti mreže. Ako nije registriran nikakav promet na mreži, započinje emitiranje podataka.

"Multiple Access" znači da više umreženih računala može koristiti isti prijenosni medij.

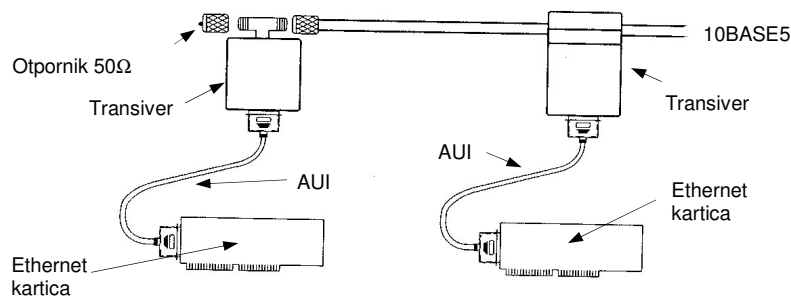
"Collision Detection" definira ponašanje računala ako se dogodi da dva ili više računala započnu emitiranje podataka u isto vrijeme. Tijekom emitiranja podataka Ethernet uređaj nastavlja osluškivati mrežu. Ako se u međuvremenu pojavi paket nekog drugog računala nastaje kolizija i računalo prekida emitiranje. Slijedi odašiljanje signala o zakrčenosti, koji upozorava ostala računala u mreži da podaci koji nailaze kolidiraju. Potom sva računala u lokalnoj mreži zaustavljaju emitiranje za slučajni iznos vremena čekanja. Na taj način vjerojatnost da će opet dva računala započeti istovremeno emitiranje značajno se smanjuje. Kolizije se događaju zbog kašnjenja signala koji putuje kroz provodnik, od trenutka u kojem jedno računalo osluškuje, a drugo započinje emitirati. U uvjetima normalnog prometa kroz mrežu, koji odgovara kapacitetu Etherneta (10 ili 100 Mb/s), kolizije se događaju tek povremeno ne usporavajući znatno propusnost mreže. Međutim, kod jako opterećene mreže, sa velikim brojem povezanih računala, učestalost kolizija može biti toliko velika da se praktički blokira protočnost mreže. Zato je Ethernet protokol posebno prikladan za manje lokalne mreže. Smatra se da je prikladno opterećenje Ethernet mreže u rasponu od 30% do 60% punog kapaciteta. Ukoliko je prosječno opterećenje veće, potrebno je mrežu podijeliti na manje podmreže povezane posebnim mrežnim uređajima mostovima (eng. bridge) ili usmjerivačima (eng. router), koji odvajaju promet lokalnih mreža.

S obzirom da je Ethernet zasnovan na slučajnom pristupu, kapacitet prijenosa podataka nije u pravilu kontinuiran. Kada se radi o prijenosu vremenski nepovezanih podataka onda ta značajka ne dolazi do izražaja. Ali, ako se radi npr. o video ili audio informacijama, svako kašnjenje, nastalo zbog zauzetosti mreže ili kolizija, odrazit će se izravno na kvaliteti reprodukcije istih.

### 5.2.1.2 Fizičke značajke

#### DEBELI ETHERNET

Ethernet mreža se originalno temelji na koaksijalnom provodniku promjera  $\frac{1}{2}$ ". Naziva se debeli Ethernet i može doseći duljinu do 500 m. Podržava kapacitet od 10 Mb/s. Službena mu je oznaka 10BASE5 (10 Mb/s, osnovnog - **BASE** raspona operativnosti i 500 m duljine). Računala se povezuju na debeli Ethernet putem AUI (eng. "Attachment Unit Interface") priključnog provodnika i transivera. Transiver ostvaruje spoj s glavnim provodnikom najčešće pomoću penetrirajućih spojeva koji probijaju izolaciju, ili pak rezanjem glavnog provodnika, kao što je prikazano na Sl. 5-9.



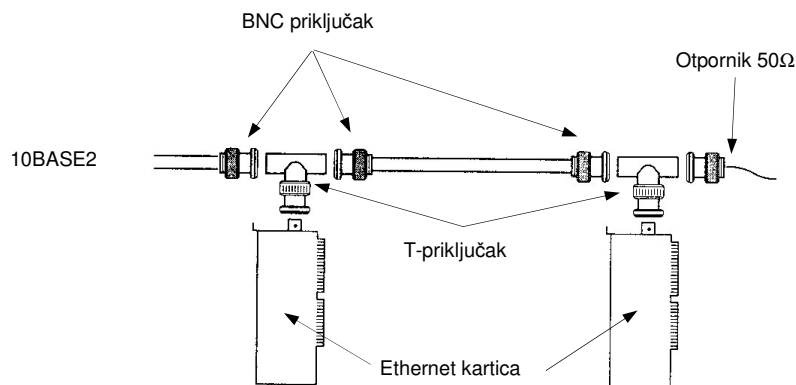
Sl. 5-9 Povezivanje na 10BASE5 Ethernet

Transiveri se mogu povezivati na glavni provodnik na udaljenosti od najmanje 2,5 metara. AUI provodnik može biti dugačak do 50 metara. Krajevi glavnoga provodnika moraju biti uzemljeni i terminirani s otpornicima od  $50 \Omega$ .

Svaki segment, dužine do 500 metara, može primiti najviše 100 priključaka. Ukoliko je potrebno osigurati povezivanje računala na udaljenosti većoj od 500 metara, moguće je pomoću ponavljača (eng. repeater) povezati više Ethernet segmenata.

### TANKI ETHERNET

Tanki Ethernet ima oznaku 10BASE2, što ukazuje, po prethodnoj analogiji, da najveća dužina segmenta (2) ne smije prelaziti 200 metara, odnosno točnije 185 metara. Naziv "tanki Ethernet" dolazi od osnovnog koaksijalnog provodnika koji je tanji od 10BASE5 provodnika. Značajna prednost tankog ožičenja je cijena i jednostavnost povezivanja. Računala se putem tankog Etherneta povezuju u nizu pomoću BNC ("BayoNet Connector") priključka, kao što je prikazano na Sl. 5-10. Dakle, ovdje nije potrebno osigurati osnovni provodnik, već se postojeće veze po potrebi kaskadno produžuju od računala do računala. Važno je samo na krajevima niza osigurati terminatore od  $50 \Omega$ . Segmenti duljine do 185 metara mogu se, jednako kao i debeli Ethernet, povezivati preko ponavljača.

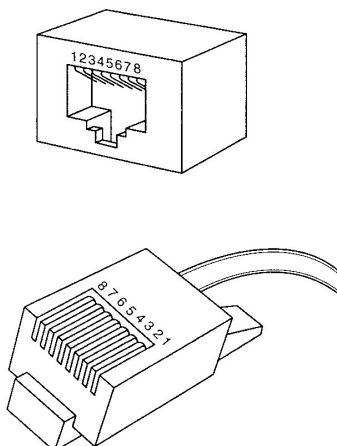


Sl. 5-10 10BASE2 Ethernet

### OŽIČENJE PLETENIM PARICAMA

Provodnici sa pletenim paricama (eng. "TP - Twisted Pair"<sup>24</sup>) izrađeni su, kao što naziv sugerira, od parova međusobno opletenih žica da se smanji utjecaj električne interferencije. Skraćeno se zovu UTP provodnici (eng. "UTP - Unshielded Twisted Pair"), jer ne posjeduju posebnu zaštitu od magnetne interferencije (Sl. 5-11). Postoje i zaštićena verzija pletenih parica STP (eng. "STP - Shielded Twisted Pair"), koje se poglavito koriste u uvjetima gdje se očekuju intenzivnije električne i magnetske interferencije koje bi mogle stvoriti neželjene smetnje u prijenosu signala.

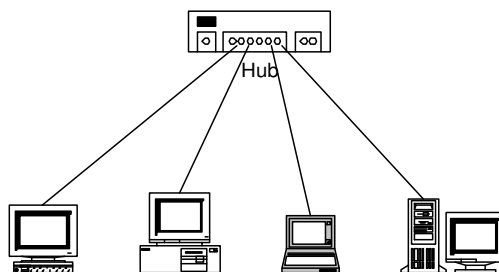
<sup>24</sup> Obično se naziva UTP provodnik (eng. "UTP - Unshilded Twisted Pair"), jer ne posjeduje posebnu zaštitu od magnetne interferencije.



Sl. 5-11 UTP priključci

TP provodnik se sastoji od dvije pletene žice (parice). Jedan par služi za emitiranje, a drugi za primanje signala.

Ožičenje pletenim paricama koristi zvjezdastu topologiju povezivanja. Žice se razvode iz središnjeg distribucijskog uređaja koji se zove "Hub" (Sl. 5-12).



Sl. 5-12 Zvjezdasta topologija primijenjena na sabirničku tehnologiju

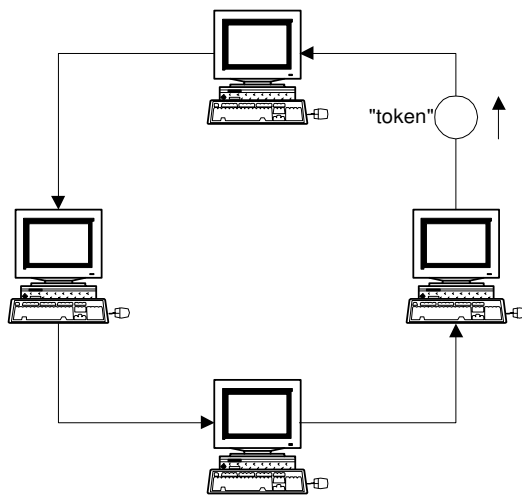
Pletene parice imaju oznaku 10BASE-T i 100BASE-T. Oznaka "10" ukazuje na brzinu prijenosa od 10 Mb/s, a "100" označava vrstu provodnika s mogućnošću prijenosa do 100 Mb/s. "T" na kraju označava da se radi o "Twisted pair" provodniku. Na "Hub" se povezuju TP provodnici do 100 metara dužine, u slučaju 10BASE-T provodnika, ili do 250 metara dužine, ako se koristi 100BASE-T provodnik.

Povezivanje pletenim paricama postupno zamjenjuje prethodno opisane načine povezivanja i primjenjuje se kod većine suvremenih mrežnih standarda. Koristi se jednako za povezivanje u Ethernet, Token Ring, ATM i drugim mrežama. Pritom treba napomenuti da primjena TP zvjezdaste topologije kod Ethernet mreže ne znači promjenu njene logičke sabirničke topologije. Zvjezdasta topologija ovdje definira fizičku topologiju povezivanja. Kod Ethernet mreže povezane TP provodnicima, središnji "Ethernet Hub" preuzima ulogu sabirnice, odnosno glavnog provodnika, a

TP provodnici postaju priključni provodnici, kao npr. AUI provodnici kod debelog Etherneta.

### 5.2.2 Token Ring

Token Ring je, kao i Ethernet, tehnologija namijenjena lokalnim mrežama. Razvijena je u IBM-u. Odlikuje se postojanim kapacitetom prijenosa podataka, kao glavnom funkcionalnom značajkom.



Sl. 5-13 "Token Ring" koristi "token" za usklađivanje komunikacije

Token ring primjenjuje prstenastu logičku topologiju. Odašiljač svakog računala spojen je s primačem susjednog računala, tvoreći zatvoreni krug. Na taj način računala prenose podatke od jednog do drugog u krug. Pritom, poseban podatkovni paket, koji se zove "token" (simbol, oznaka, žeton), kruži prstenom od računala do računala. Mrežni uređaj računala može odašiljati podatke samo kada je u posjedu tokena (Sl. 5-13). Nakon što su podaci poslani, računalo otpušta token te će sljedeće računalo moći poslati podatke. Ako u trenutku zaprimanja tokena računalo nema podataka za slanje, prosljedit će ga do narednog računala. Na taj način svako računalo ima jednaku mogućnost emitiranja podataka, za razliku od Etherneta koji ne osigurava mogućnost slanja i ona ovisi o slučajnom stanju zauzetosti mrežne sabirnice. Token Ring osigurava konstantni kapacitet protoka podataka od 16 Mb/s ili 4 Mb/s ovisno o primijenjenoj vrsti provodnika.

Iako je naizgled Token Ring tehnologija jednostavna, skriva mnoge složenosti u svom protokolu i upravljačkim algoritmima. Stoga je znatno skuplja od Etherneta (do šest puta), što je glavni razlog nešto rjeđe primjene, unatoč spomenutim kapacitivnim prednostima. Primjenjuje se poglavito u uvjetima gdje je važno osigurati stalnost kapaciteta protoka podataka.



**O Ž I Č E N J E**

Fizički se Token Ring mreža uobičajeno realizira u zvjezdastoj topologiji, koristeći prednosti strukturnog povezivanja<sup>25</sup>. Kao središnji priključni uređaj ("hub") koristi se MSAU - eng. "MultiStation Access Unit". U okviru njega a konfigurirana je prstenasta logička topologija kakvu koristi Token Ring tehnologija. Računala se do uređaja MSAU spajaju najčešće IBM provodnicima ili standardnim UTP provodnicima.

---

<sup>25</sup> Strukturno povezivanje odlikuje zvjezdasta fizička topologija koja dopušta jednostavno povezivanje različitih podmreža (struktura) i njihovih protokola, kao i jednostavno proširivanje i unapređivanje mrežnih resursa.



### 5.3 TCP/IP protokol

Protokol TCP/IP (eng. "Transmission Control Protocol/ Internet Protocol") nastao je u okviru projekta DARPA, koji je prethodio razvoju Interneta. Internet je ime mreže svjetskih mreža (ARPANET, MILNET, NFSNet, ...) i objedinjuje milijune lokalnih računalnih mreža. Pod nazivom "internet" iz kratice TCP/IP skriva se princip ili koncept spajanja lokalnih mreža. TCP/IP dio je višerazinskog hijerarhijskog modela umrežavanja, koji je prethodio OSI referentnom modelu, a naziva se DoD, prema povijesnim korijenima koji ga vežu uz "Department of Defense" (Ministarstvo obrane SAD-a).

TCP/IP podrazumijeva porodicu komunikacijskih protokola. U TCP/IP porodicu protokola se ubrajaju IP protokol, TCP protokol, UDP protokol, ICMP protokol kao i mnogi drugi protokoli. Pod tim imenom obuhvaćene su i mrežne usluge ali i uslužni programi koji ostvaruju mrežne usluge, kao što su npr. udaljena prijava za rad ili prijenos podataka.



#### 5.3.1 DoD mrežni model

Kao i OSI referentni model, DoD koristi hijerarhijski komunikacijski koncept podijeljen na razine. Za razliku od OSI modela, DoD ima 4 razine. Na Sl. 5-14 prikazan je DoD model i njegov odnos s OSI referentnim modelom.

OSI	DoD
Application	Application
Presentation	
Session	
Transport	Transport (TCP)
Network	Internet (IP)
Data Link	Network interface
Physical	

Sl. 5-14 Odnos OSI i DoD modela

**Razina mrežnog sučelja** ("Network interface") rješava probleme pristupanja fizičkom mediju koji se koristi za prijenos. Najčešće je to programski veznik koji komunicira s Ethernet karticom ("device driver") ili mrežnim poslužiocem. Odgovara prvoj i drugoj OSI razini ("Physical" i "Data Link").

**Internet razina** ("Internet") je odgovorna za povezivanje logičkih adresa s fizičkim mrežnim sučeljem, odnosno fizičkim adresama. Implementirana je u obliku internet protokola (IP) koji radi s IP paketima (datagramima) i brine za njihovo usmjeravanje. IP koristi ARP (eng. "Address Resolution Protocol") i RARP (eng. "Reverse Address

Resolution Protocol") protokole za adresiranje računala. Osnovna zadaća ove razine je ostvarivanje komunikacije između dva udaljena računala.

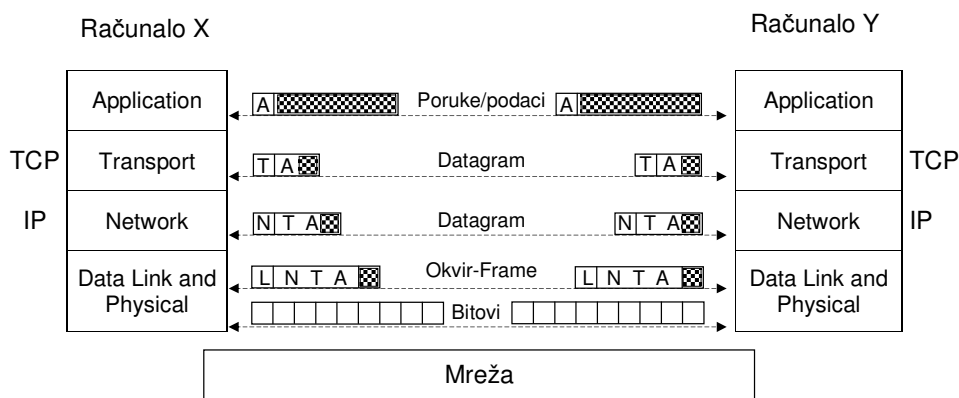
**Prijenosna razina** ("Transport") odgovara četvrtoj OSI razini. Ostvaruje komunikaciju između mrežnih programa udaljenih računala. Programi na udaljenim računalima se adresiraju kao "port number". Podaci se prenose na principu paketne komunikacije, odnosno dijeleći se na datagrame koji se na određitu ponovno sastavljaju. TCP (eng. "Transmission Control Protocol") protokoli su odgovorni za uspostavu komunikacije na ovoj razini, osiguravajući istovremenu dvosmjernu komunikaciju više različitih programa između više udaljenih računala. TCP se ujedno brine o konzistenciji primljenih podataka. U slučaju da se neki od datagrama izgubi zatražit će ponovni prijenos paketa. U slučaju primitka dupliranog datagrama, suvišni će datagram biti izbačen.

**Razina primjene** ("Application") obuhvaća uslužne programe koji koriste TCP/IP protokole za prijenos podataka (ftp – eng. "File Transfer Protocol"), elektronsku poštu (SMTP – eng. "Simple Mail Transfer Protocol") te prijavu (telnet) i rad na udaljenim računalima. Razina primjene je korisničko sučelje prema TCP/IP porodici mrežnih protokola.

Uspoređujući OSI referentni model i TCP/IP važno je uočiti da OSI predstavlja teoretski model dok TCP/IP predstavlja skup konkretnih komunikacijskih protokola koji čine komunikacijsku osnovu za mnoge računalne mreže.

### 5.3.2 TCP/IP format podataka

TCP/IP komunikacija između udaljenih računala odvija se **postupkom enkapsulacije**. Enkapsulacija podrazumijeva postupak koji podatkovne pakete više razine (npr. paketi iz "Application" razine) umeću u pakete niže razine (paketi u "Transport" razini). Svaka razina dodaje svoje zaglavlje na podatke koji se prenose (Sl. 5-15).



Sl. 5-15 Princip enkapsulacije

Kao što je ilustrirano na Sl. 5-15, podaci koji se prenose sa *razine primjene*, npr. program elektroničke pošte (e-mail), dobivaju zaglavlje odgovarajućeg programa.

Zaglavlje će sadržavati između ostalog i adresu primaoca (korisnika i računala). Podaci se potom prosljeđuje na nižu *prijenosnu razinu*, gdje ih "*Transmission Control Protocol*" dijeli u pakete (datagrame). Svaki paket nasljeđuje zaglavlje prethodne razine, ali dobiva i zaglavlje prijenosne razine. Paketi se zatim šalju *internet razini* koja oblikuje svoje pakete sa svojim zaglavljima. Pritom novo zaglavlje preuzima neke podatke od prethodnih razina (npr. adresu primaoca). Na kraju paketi stižu na najnižu razinu *mrežnog sučelja*. Ako je npr. primijenjena Ethernet mrežna tehnologija, paketi će dobiti nova zaglavlja i postati Ethernet paketi. Za Ethernet protokol nevažan je sadržaj paketa. Važno je da njegov oblik (zaglavlje, redosljed i dužina podatkovnih polja) razumljiv za Ethernet sučelje udaljenog računala. Ono će pristigle pakete ponovno raspakirati, predajući sadržaj višim razinama. Kada se podaci skupe na najvišoj razini udaljenog računala, predaju se odgovarajućem uslužnom programu koji ih potom obradi.

### 5.3.3 Porodica TCP/IP protokola

Porodica TCP/IP protokola se može podijeliti u skupinu nižih protokola (kao što su npr. IP, TCP i UDP protokoli) i skupinu viših protokola kao što su Telnet ili FTP protokoli. Viši protokoli se koriste za obavljanje posebnih zadataka kao npr. prijenos datoteka između udaljenih računala dok se niži protokoli koriste bez obzira koju se vrstu posla obavlja. Pregled važnijih TCP/IP protokola je prikazan na sljedećoj Sl. 5-16:

TFTP	Time service	Name resolution	Telnet	FTP	SMTP	LPR
ICMP, UDP			TCP			
IP						

Sl. 5-16 Hijerarhija porodice TCP/IP protokola

**Internet protokol (IP)** je bazični protokol iz TCP/IP porodice i pripada Internet razini TCP/IP modela. Definiran je sa RFC 791. Prema uslugama koje obavlja IP protokol pripada mrežnoj razini iz OSI modela (usporedba OSI modela i TCP/IP protokola je dana u sljedećem odjeljku). IP omogućuje neslužjednu komunikaciju odnosno prijenos datagrama između računala. Udaljena računala eventualno mogu pripadati potpuno različitim podmrežama koje su na neki način povezane u zajedničku mrežu. IP nema mehanizma za kontrolu toka ili grešaka. Glavni posao IP razine je da datagrame koje dobije od TCP razine pošalje na određenu adresu što kod složenih mreža može biti kompliciran posao jer treba izabrati najbolji prospojni put. Kao i svi ostali protokoli IP protokol definira format paketa (datagrama) kojim se prenose podaci. Na početku datagrama je zaglavlje. IP zaglavlje sadrži adresu određenišnog i izvorišnog računala kao i neka druga polja: verzija, tip usluge, protokol, ...

Ovdje je interesantno primjetiti da se poljem "protokol" mogu definirati različiti viši protokoli koji koriste usluge IP protokola. U praksi su to najčešće TCP i UDP ali to mogu biti i neki protokoli koji ne pripadaju TCP/IP porodici.

**User Datagram Protocol (UDP)** pripada prijenosnoj razini TCP/IP modela. Definiran je sa RFC 768. Ovaj protokol predstavlja proširenje IP protokola, a osnovna zadaća

mu je postavljanje neslijedne (nepovezane, eng. connectionless) komunikacije između udaljenih računala. S obzirom da prijenos nije slijedan u nekim slučajevima može uzrokovati duplicirajuću komunikaciju. Najčešće se upotrebljava za slanje manjih poruka minimalnim mehanizmom. Iako se iz ovakvog "službenog" opisa koji se najčešće nalazi u literaturi može zaključiti da je to neki loš i nepouzdan protokol u praksi se na ovom protokolu baziraju mnogi važni viši protokoli. Kao primjer jednog takvog važnog višeg protokola može se navesti NFS (eng. "Network File System") koji u praksi radi sasvim pouzdano na milijunima instaliranih računala.

**Internet Control Control Message Protocol (ICMP)** je jednostavan protokol za slanje kontrolnih poruka i upita kao npr. "da li je računalo dostupno" i sl. Propada prijenosnoj razini kao i UDP i TCP protokoli. ICMP protokol se npr. koristi u važnoj naredbi "ping" kojom se može ustanoviti da li je moguće ostvariti vezu sa udaljenim računalom i kolika je brzina komuniciranja.

**Transmission Control Protocol (TCP)** pripada prijenosnoj razini TCP/IP modela. Definiran je sa RFC 793. On, za razliku od UDP protokola, omogućuje slijednu komunikaciju. To je "end-to-end" protokol koji povezuje dva procesa na udaljenim računalima, a osigurava sigurnu, neduplicirajuću isporuku okteta udaljenom korisniku. TCP predstavlja bazu za mnoge druge više protokole (telnet, ftp) koji zahtijevaju pouzdan prijenos većeg broja okteta. Kao i IP i UDP protokoli TCP definira format svojih paketa. TCP zaglavlje ima više polja u odnosu na IP zaglavlje:

- source port
- destination port
- sequence number
- acknowledgment number
- data offset
- type
- window
- checksum
- urgent pointer
- data

Najvažniji podaci u TCP zaglavlju su pristupni brojevi (eng. port numbers) i slijedni broj (eng. sequence number). Pristupni brojevi definiraju "izvorišni i odredišni ulaz" unutar računala kamo (i odakle) se šalje poruka. Ovi brojevi su važni jer se istovremeno može slati i primiti više poruka na/sa istog računala. Ti brojevi na jedinstveni način definiraju ulaz/izlaz za poruke koje šalje isti korisnik. Slijedni broj definira broj poslanih okteta (ne broj datagrama). Tako ako su datagrami veličine 500 okteta prvi datagram ima slijedni broj 0, a drugi ima broj 500 itd. Treba primijetiti da se u TCP zaglavlju nigdje ne pojavljuje adresa odredišnog računala već samo "port number" unutar računala. Adresa odredišnog računala se nalazi u IP zaglavlju. Dužnost je TCP razine da napravi datagram i da ga zajedno s adresom odredišnog računala proslijedi IP razini.

TCP i UDP su na istoj razini ali definiraju različite mogućnosti. Osnovne razlike između TCP protokola i UDP protokola se sažeto mogu prikazati u nekoliko točaka:

- TCP dijeli (i ponovno sakuplja) veće poruke na IP datagrame.
- UDP radi s porukama koje stanu u jedan IP datagram (npr. kod traženja adrese udaljenog računala od poslužioca koji ima bazu imena i njihovih adresa - DNS).
- TCP vodi računa o poslanim datagramima i može napraviti retransmisiju.

- ❑ UDP ne vodi računa o poslanim datagramima i ne može napraviti retransmisiju (ako ne dobijemo odgovor u određenom vremenu može se ponovo poslati zahtjev).
- ❑ UDP je jednostavniji, UDP zaglavlje zauzima manje mjesta.

**File Transfer Protocol (FTP, RFC 959)** omogućuje prijenos datoteka između udaljenih računala. Pripada razini primjene u TCP/IP modelu. Osim prijenosa datoteka FTP predviđa punu provjeru identiteta i osnovne operacije s datotekama. Osim FTP protokola postoji i **ftp** uslužni program.

**Telnet** protokol (**Telnet, RFC 854**) je definiran na razini primjene u TCP/IP modelu. Omogućuje prijavu rada na udaljeno računalo. Osim telnet protokola postoji i **telnet** uslužni program.

### 5.3.4 Internet adrese

U TCP/IP mreži svako računalo mora imati jedinstvenu adresu. To znači da u Internet mreži ne može postojati niti jedan čvor u svijetu s istim identifikacijskim brojem (adresom). U odvojenim lokalnim TCP/IP mrežama, koje nisu spojene na globalnu Internet mrežu, adrese moraju bit jedinstvene tek na lokalnoj razini. Međutim, računala iz takve mreže ne mogu komunicirati izvan svoje domene jer preklapanje adresa može dovesti do kolizije i problema u radu mrežnih poslužilaca i usmjerivača. O dodjeli adresa stoga brine središnja međunarodna neprofitna institucija ICANN (The Internet Corporation for Assigned Names and Numbers), osiguravajući jedinstvenu politiku u dodjeli brojeva, naziva domena i parametara Internet protokola<sup>26</sup>.

Internet (IP, Internet Protocol) adresa definirana je kao 32-bitni broj sastavljen od četiri okteta odvojena točkom. Adresa se interpretira dekadski kao četiri broja odvojena točkom, npr. 192.84.94.50. S obzirom da je za svaki od četiri dekadski broja rezerviran jedan oktet, u internet adresi mogu se pojaviti dekadski brojevi u rasponu od 0 do 255, jer je 255 najveći cijeli broj koji se može binarno izraziti u slogu od osam bitova. Konceptija Interneta podrazumijeva povezivanje nezavisnih podmreža, pa je struktura adrese podijeljena na dva logička dijela. Prvi dio sadrži adresu mreže, kojom je jednoznačno definirana podmreža, dok drugi dio adrese označava adresu pripadajućeg računala. S obzirom da podmreže mogu okupljati više ili manje računala i/ili svojih podmreža, adresa može biti različito oblikovana. Postoje tri oblika Internet adrese: A, B, C, D i E klase. Pritom su klase A, B, i C predviđene za adresiranje, klasa D za emitiranje općih poruka, a klasa E je rezervirana za buduću namjenu.

Adrese klase A imaju samo jedan oktet rezerviran za adresu mreže dok se preostala tri okteta koriste za adresiranje računala:

A klasa: net.host.host.host      od 1.x.x.x do 126.x.x.x

U klasi B za adresu mreže su rezervirana dva okteta dok je u C klasama za adresu mreže određeno tri okteta:

B klasa:                    net.net.host.host                    od 128.1.x.x do 191.254.x.x

<sup>26</sup> Nekada je o dodjeli internet adresa brinula američka vladina institucija IANA (International Assigned Numbers Authority, [www.iana.org](http://www.iana.org)), koja se sada nalazi u sklopu ICANN-a.

C klasa: net.net.net.host od 192.1.1.x do 223.254.254.x

Tako će npr. u klasi A biti moguće adresirati  $256 \times 256 \times 256 - 2^{27}$  računala unutar jedne podmreže. U klasi B preostaje za adresiranje  $256 \times 256 - 2^3$  računala unutar iste podmreže, a unutar C klase svega 254 (Tab. 5-3). Prema utvrđenom pravilu početni okteti u adresi određuju kojoj klasi će određena adresa pripadati. Adrese čiji je početni oktet između 1 i 126 decimalno pripadaju klasi A, od 128 do 191 klasi B, a od 192 do 223 klasi C (Tab. 5-3). Stoga će klasa A moći adresirati svega 126 podmreža, klasa B 16 383, a C 2 097 151 različitih podmreža.

Tab. 5-3 Značajke A, B i C klasa Internet adresa

Internet klasa adrese	Raspon prvog okteta	Najveći broj podmreža	Najveći broj računala
A	1-126	126	16 777 214
B	128-191	16 383	65 534
C	192-223	2 097 151	254

Globalne mreže, kao što je ARPAnet, pripadaju A klasi jer one sadrže veliki broj čvorova. B klasa se dodjeljuje manjim zemljama i većim organizacijama, s relativno velikim brojem podmreža i pripadajućih čvorova. Manje organizacije ili mrežne cjeline dobivaju C klasu.

Brojevi 0 i 255 imaju posebno značenje u adresi. 0 je rezervirana kao neodređena oznaka. Npr. 0.0.0.62 označava adresu računala (62) bez pripadnosti mreži (0.0.0). Broj 255 je rezerviran za "broadcast" - emitiranje općih poruka. "Broadcast" je poruka koja se šalje svim računalima u mreži. Npr. ako se žele doznati imena raspoloživih računala u lokalnoj mreži čija je adresa 161.53.117, onda će biti emitirana poruka s adresom 161.53.117.255, gdje je 255 upotrebjeno kao "broadcast", zamjenjujući konkretne adrese računala lokalne mreže. Internet adresa ne smije započinjati s jednim od sljedećih brojeva:

0	nepoznata adresa
127	lokalna petlja
255	"broadcast"
> 233	rezervirano za buduće potrebe

S obzirom da je teško pamtiti dvanaest brojeva Internet adrese, TCP/IP programi predviđaju korištenje imeničkih usluga koje omogućavaju korisniku adresiranje računala pomoću pridruženih simboličkih imena. Stoga svako računalo u mreži posjeduje, osim Internet adrese, i odgovarajuće simboličko ime. UNIX operacijski sustav pohranjuje internet adrese i njihova imena u zapisnik `/etc/hosts` (Sl. 5-17), a Windows NT u zapisnik `\Winnt\System32\drivers\etc\hosts` (NT ujedno koristi i zapisnike LMHOSTS i WINS). Slika prikazuje primjer jednog zapisnika `/etc/hosts`. Adresa 127.0.0.1 označava lokalnu petlju računala (za lokalno adresiranje). Poruke poslone na tu adresu nikada ne stižu do mrežnog provodnika, već služe za adresiranje lokalnog računala. U istom je zapisniku pohranjena Internet adresa

<sup>27</sup> Broj mogućih adresa umanjuje se za dva jer su adrese računala izražene binarno samo nulama (npr. net.0.0.0) ili samo jedinicama (npr. dekadski net.255.255.255) rezervirane. Prva označava lokalno računalo ("loop back", 127.0.0.0), a potonja služi za općenito adresiranje računala u mreži ("broadcast").



lokalnog računala (loghost) "161.53.117.160 pro10  
 pro10.sjever.fsb.hr loghost". Ostale adrese i pridružena imena  
 definiraju ostala računala u lokalnoj mreži.

```
#
# Internet host table
#
127.0.0.1    localhost
161.53.117.160  pro10    pro10.sjever.fsb.hr    loghost
161.53.117.151  pro1     pro1.sjever.fsb.hr
161.53.117.154  pro4     pro4.sjever.fsb.hr
```

Sl. 5-17 Primjer "hosts" zapisnika

Kada se pokrene neki od TCP/IP uslužnih programa (npr. **ftp**), potrebno je navesti ime računala s kojim se želi komunicirati (npr. pro1). Za zadano će ime računala TCP/IP program potražiti pripadajuću adresu u zapisniku **/etc/hosts**. Ukoliko ne nađe ime zadanog računala, program će se obratiti nekom od nadređenih računala poslužilaca koji pružaju imeničke usluge, odnosno sadrže bazu podataka o adresama računala.

### 5.3.5 Informacijske mrežne usluge

TCP/IP protokol predviđa korisnik-poslužilac model umrežavanja i povezivanja lokalnih mreža. To znači da u lokalnoj mreži mora postojati barem jedno računalo poslužilac koje preuzima na sebe zadaće mrežnih usluga, prvenstveno usmjeravanja i adresiranja, osiguravajući komunikaciju s vanjskom mrežnom infrastrukturom. Najčešće primijenjene mrežne usluge su:

- DNS (eng. "Domain Name Service") ili Imenička usluga domene,
- NIS+ (eng. "Network Information Service") ili Mrežna informacijska usluga,
- BIND/Hesoid.

Osnovna svrha ovih usluga je pružanje informacija o zajedničkim objektima (imena, adrese, korisnici, pisači, ...) na mreži. Računalo poslužilac, ono koje osigurava imeničke usluge, posjeduje bazu podataka o imenima računalnih objekata i na zahtjev daje odgovore o adresama, alternativnim imenima, zapisničkom sustavu itd.

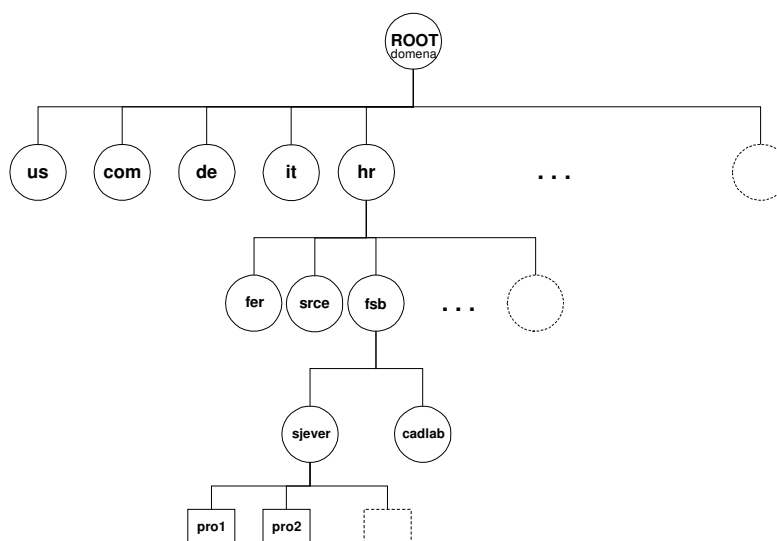
Od navedenih usluga posebno treba istaknuti DNS kao glavnu podršku za interpretaciju (rješavanje) imena u Internetu. TCP/IP protokol je izravni korisnik DNS-a, no njegovim se uslugama mogu koristiti i drugi mrežni protokoli. Na Internet su priključeni milijuni računala i ako ne bi postojale imeničke usluge svako bi računalo moralo imati cjelovit popis ostalih računala na mreži. Takove baze podataka bi zauzimale veliki prostor na disku računala, a njihovo bi pretraživanje predugo trajalo. No najvažnije je da bi ažuriranje podataka o mrežama širom svijeta, koje se svakodnevno mijenjaju, bilo jednostavno neizvedivo. Informacijske mrežne usluge olakšavaju upravljanje i ažuriranje bazama podataka i omogućavaju hijerarhijsko povezivanje distribuiranih baza podataka.

DNS je mrežni protokol definiran na razini primjene u porodici TCP/IP protokola. Osnovna zadaća DNS-a je rješavanje (interpretacija) simboličkih imena udaljenih

računala. DNS je zasnovan na mehanizmu pitanje/odgovor (eng. "query/response") i korisnik-poslužilac modelu. Neka od računala na mreži imaju ulogu DNS poslužioca dok su ostala računala DNS korisnici. DNS poslužilac prima pitanje od DNS korisnika, rješava njihove zahtjeve i šalje im odgovore. Svaki poslužilac ima vlastitu bazu podataka o računalima i njihovim adresama. Ako poslužilac ne može odgovoriti na pitanje pretraživanjem vlastite baze onda može poslati pitanje drugim poslužiocima. DNS koristi UDP kao transportni protokol za komunikaciju s udaljenim računalima (poslužiocima). UDP protokol je prikladan zbog svoje jednostavnosti i zato jer ne uzrokuje veliko opterećenje mreže.

Najraširenija implementacija DNS-a je BIND (eng. "Berkeley Internet Name Domain"), imenički poslužilački program za UNIX operacijski sustav. Windows NT koristi WINS program za interpretaciju imena NetBIOS mreže, ali i DNS korisničku i poslužilačku podršku.

Za razumijevanje DNS usluge bitno je razumijevanje koncepta domene i njihove hijerarhije. Svaka domena pripada određenoj razini u hijerarhijskoj organizaciji domena. Na vrhu hijerarhije je "root" domena ispod koje se nalaze sve ostale domene (Sl. 5-18). "Root" domena se označava točkom (.) na prvome mjestu imena. S obziro da sve domene imaju istu zajedničku polaznu domenu, "root" točka se ispušta iz imena. Na prvoj razini ispod "root" domene nalaze se "top level". One definiraju globalne mreže pojedinih država ili velikih zajednica, npr. **hr** (Hrvatska), **us** (SAD), **it** (Italija), **com** (komercijalna domena), **edu** (domena znanstvenih i obrazovnih ustanova) itd<sup>28</sup>. U okviru "top level" domena nalaze se manje domene (npr. **fsb**), a one se pak mogu dijeliti na još finije mrežne cjeline (npr. **sjever**) kojima su pridružena pripadajuća računala (npr. **pro1**).



Sl. 5-18 Primjer hijerarhije domena

<sup>28</sup> Dvoslovni nazivi većine domena usklađeni su sa ISO-3166 normom.

Tako se dobiva hijerarhijsko ustrojstvo koje se simbolički opisuje uređenim nizom imena odvojenih točkom. Npr. domena **sjever** pripada domeni **fsb**, a ona je pak dio domene **hr** pa će stoga njezino puno ime glasniti:

**sjever.fsb.hr**

Ispred imena domene piše se ime pripadajućeg računala, npr.:

**pro1.sjever.fsb.hr**

Ime na prvome mjestu označava računalo **pro1** koje pripada domeni **sjever.fsb.hr**. Ime ne smije imati više od 255 znakova. Ovakva organizacija domena podsjeća na hijerarhijsku zapisničku strukturu operacijskih sustava UNIX ili Windows NT. Umjesto imena zapisnika ovdje se navode imena domena i računala. Za razliku od hijerarhije zapisnika, imena domena se pišu obrnutim redoslijedom. Najprije se navodi ime odredišnog objekta (računala) pa onda imena domena redom prema vrhu hijerarhije. Kod zapisničkog se sustava počinje od "root" imenika prema odredišnom objektu. Ovakav princip omogućava da se u mreži nalaze računala s istim imenom. Jedino se ne smije dozvoliti da ta računala budu u istoj domeni.

Svaki DNS korisnik mora poznavati DNS poslužioaca svoje domene. Na UNIX operacijskom sustavu, adrese DNS poslužilaca pohranjuju se u zapisniku **/etc/resolv.conf** (Sl. 5-19).

```
;resolv.conf file for DNS client
;
domain          xxx.hr

nameserver 161.53.180.130
nameserver 161.53.160.131
```

Sl. 5-19 Primjer "resolv.conf" zapisnika

U gornjem primjeru za **resolv.conf** zapisnik započinje s tri komentarska retka, a zatim je definirano ime domene "**xxx.hr**" za koju su definirana dva DNS poslužioaca, odnosno njihove IP adrese. Uobičajeno je da se definira više poslužilaca. Ako prvi poslužilac, iz bilo kojih razloga, nije dostupan pokušat će se konzultirati drugi na listi i tako dalje.

*DNS poslužioaci* čuvaju podatke (imena, adrese, kratki opis) u udaljenim računalima i odgovaraju na upite DNS korisnika. Podaci se čuvaju u više zapisnika. U zapisniku **/etc/named.boot** pohranjeni su podaci koji definiraju strukturu DNS baze. Na Sl. 5-20 prikazan je jedan primjer **/etc/named.boot** zapisnika.

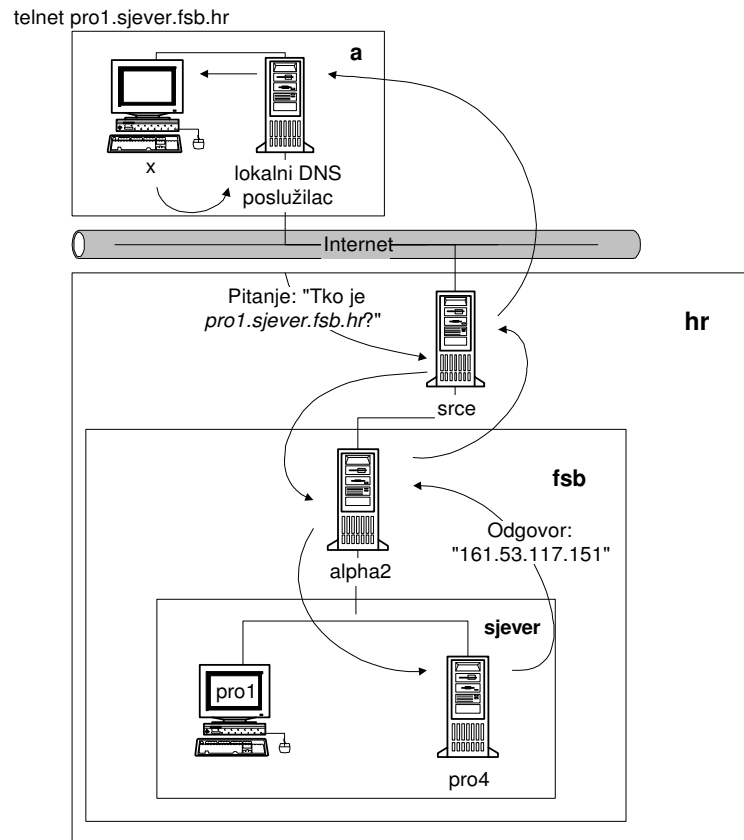
```

; named.boot
;
; boot file for authoritative master name server for Berkeley.EDU
; Note that there should be one primary entry for each SOA record.
;
;
; type      domain                source host/file                backup
file
;
domain     a.b.com
primary    a.b.com                /etc/named/named.hosts
primary    117.53.161.IN-ADDR.ARPA /etc/named/named.rev
primary    0.0.127.IN-ADDR.ARPA   /etc/named/named.local
secondary  b.com                  161.53.116.12
/etc/named/fsb_named.hosts
secondary  116.53.161.IN-ADDR.ARPA 161.53.116.12
/etc/named/fsb_116_named.rev
cache     .                        /etc/named/root.cache

```

Sl. 5-20 Primjer "named.boot" zapisnika

Nakon komentarskih redaka, označenih s ";", slijedi ime domene poslužioca. Potom su definirani zapisnici primarne (lokalne) domene: **/etc/named/named.hosts** (baza podataka s imenima i IP adresama, **/etc/named/named.rev** (popis adresa u obrnutoj notaciji za interpretaciju imena kada je poznata adresa), **/etc/named/named.local** (adresa lokalnog računala), potom slijede adrese sekundarnih poslužilaca nadređenih i podređenih domena. Posljednja je definicija zapisnika **/etc/named/root.cache** u kojem su pohranjene adrese poslužilaca kojima se treba obratiti ako zahtijevano ime ne postoji u lokalnoj DNS bazi.



Sl. 5-21 DNS decentralizirani model rješavanja IP adresa

Princip rada DNS-a objasniti će se na primjeru sa Sl. 5-21. Prikazana je domena **sjever** kao poddomena domene **fsb** (Fakultet strojarstva i brodogradnje), koja je pak sadržana u domeni **hr** (Hrvatska).

Kada neki Internet korisnik zatraži komunikaciju s udaljenim računalom, npr. putem naredbe **telnet**, zadajući njegovo ime (npr. **pro1.sjever.fsb.hr**), a to ime nije definirano u lokalnom zapisniku (**/etc/hosts**), aktivira se program rješavač (eng. resolver). Njegova je zadaća pronalaženje odgovarajuće IP adrese računala. Rješavač će se u skladu s DNS protokolom obratiti DNS poslužiocu iz **/etc/resolv.conf** zapisnika. Na DNS poslužiocu je stalno aktivan uslužni program **"in.named"** ili **"named"** koji osluškuje pitanja korisnika.

Ukoliko lokalni DNS poslužilac ne posjeduje definiciju za zadano ime, proslijedit će pitanje poslužiocima iz **root.cache** zapisnika, obično DNS poslužiocima domene više razine. Poslužiocima najviše razine (npr. **srce** za domenu **hr**) posjeduju adrese poslužilaca svojih podređenih domena ili sekundarnih poslužilaca. Ukoliko poslužilac prepozna domenu iz zadanog imena potražiti će ime u svojoj bazi ili će proslijediti pitanje prema odgovarajućem nižem DNS poslužiocu (računalo **alpha2** domene **fsb**). Budući da računalo **pro1** pripada domeni **sjever** pitanje će biti proslijeđeno njenom DNS poslužiocu **pro4**. U svojoj će DNS bazi računalo **pro4** pronaći

odgovarajuću adresu računala **pro1** i proslijediti ga računalu koje je tražilo uslugu. Tek nakon primljene adrese, TCP/IP program (**telnet**) na računalu koje traži vezu s udaljenim računalom (**pro1.sjever.fsb.hr**) odašilje adresirani poziv, kojeg usmjerivači na sličan način upućuju do traženog računala.



---

## 5.4 HTTP

WWW, odnosno "World Wide Web", je najrašireniji oblik razmjene informacija putem Interneta. Iza naziva WWW krije se način oblikovanja hipertekstualnih informacija pomoću posebnog jezika HTML (eng. "HyperText Markup Language"), kao i protokol HTTP (eng. "HyperText Transfer Protocol") za razmjenu hipertekstualnih i multimedijalnih podataka putem računalne mreže.

Kovanicu *hipertekst* uveo je 1965. godine Ted Nelson opisujući tekst koji nije ograničen slijednom linearnom strukturom podataka. Hipertekst je višerazinski model strukturiranja podataka koji omogućava proširivanje pojedinih pojmova iz teksta jednostavnim ulančavanjem zapisa čija detaljnost u pravilu raste s razinama ulančavanja. Ted Nelson je uveo i pojam hipermedija (eng. "Hypermedia"), označavajući sadržaj iz teksta koji ne mora biti tekst, već bilo koji multimedijalni zapis (slika, zvuk, video, geometrijski model, ...).

Hipertekstualni koncept se razvija kroz različite projekte tijekom mnogih godina, no svoju najznačajniju implementaciju doživljava kroz razvoj WWW-a 1989. godine. Te je godine Tim Berners-Lee, znanstvenik glasovitog europskog instituta za nuklearna istraživanja CERN ("Conseil Europeen pour la Recherche Nucleaire") iz Ženeve u Švicarskoj, predložio hipertekstualni sustav "HyperText and CERN" za razmjenu podataka između članova istraživačkih skupina. Glavne značajke sustava obuhvaćale su sljedeće:

- korisničko sučelje i pristup podacima moraju biti jednoznačno definirani neovisno o računalnoj platformi,
- mogućnost pristupa multimedijalnim podacima,
- razmjena podataka putem računalne mreže,
- primjena koncepta korisnik/poslužilac.

Uskoro, 1990. godine, prototip WWW-a razvijen je na NeXT računalu. Paralelni razvoj Interneta i mrežne infrastrukture poticali su ubrzani razvoj i primjenu Web koncepta. Kada se 1993. godine pojavljuje prvi Web korisnik (preglednik) s grafičkim sučeljem, WWW i Internet progresivno otvaraju svjetski informacijski prostor. Danas se sa sigurnošću može ustvrditi da eksplozivno širenje Interneta velikim dijelom treba zahvaliti upravo WWW-u, koji je na jednostavan, multimedijalan i intuitivan način omogućio ponudu i korištenje najrazličitijih informacija i usluga. O koordiniranom normiranom razvoju WWW-a danas brine "World Wide Web Organization", udruga nastala na poticaj MIT-a ("Massachusetts Institute of Technology") i CERN-a 1994. godine.

HTTP protokol definira način prijenosa hipertekstualnih podataka kroz računalnu mrežu, poglavito koristeći TCP/IP protokol. Zato se HTTP smatra članom TCP/IP protokola. Međutim, HTTP je, teorijski gledano, protokol neovisan o transportnom protokolu. To znači da može koristiti i druge protokole, npr. UDP i X.25. U odnosu na OSI ili DoD model, HTTP pripada razini primjene. S obzirom da koristi koncept korisnik/poslužilac, implementiran je, s jedne strane kao Web poslužilac (npr. "Apache HTTP Server"), a s druge kao Web preglednik (npr. "Netscape", "Mosaic", "Microsoft Internet Explorer").



Web poslužilac distribuira podatke pohranjene u obliku HTML zapisnika na zahtjev korisnika.

Web preglednik (korisnik) šalje zahtjeve poslužiocu i brine se o intepretaciji pristiglih HTML i multimedijalnih podataka na lokalnom računalu korisnika. Preglednik dakle "prevodi" HTML i druge standardne zapise u oblik razumljiv specifičnom računalu. Ovakvim pristupom problem entropije i redundancije tehničke i tehnološke neusklađenosti računarskog razvoja jednostavno nestaje, rušeći glavne komunikacijske prepreke. Osim HTTP-a Web preglednik implementira i druge TCP/IP protokole i usluge:

- FTP, protokol za prijenos tekstualnih i binarnih podataka,
- TELNET, protokol za prijavu na udaljeno računalo,
- NNTP ("Network News Transfer Protocol"), protokol za asinkronu razmjenu podataka putem "Usenet news",
- Gopher, protokol za razmjenu podataka putem izbornika.

Pristup određenoj Web bazi podataka definira se pomoću URL (eng. "Uniform Resource Locator", jedinstveni lokator izvora) adrese. URL se sastoji od identifikatora protokola i adrese. Adresa uključuje Internet ime ili adresu Web poslužioca i lokaciju zapisnika u kojem su pohranjeni podaci. Opći oblik URL adrese je:

**protocol://ime-web-poslužioca/lokacija/zapisnik,**

npr.:

**http://www.sjever.fsb.hr/pub/dokumenti/zapisnik.html,**

ili:

**ftp://edgar.stern.nyu.edu/pub/papers/edgar.ps.**

Prvi primjer definira pristup Web poslužiocu putem HTTP-a (**http://**), ime poslužioca je **www.sjever.fsb.hr**, a lokacija zapisnika u zapisničkom sustavu poslužioca je **/pub/dokumenti/zapisnik.html**. Ekstenzija **".html"** ukazuje da zapisnik sadrži podatke u HTML obliku.

Drugi primjer URL-a definira primjenu FTP protokola (**ftp://**), adresirajući FTP poslužioca **edgar.stern.nyu.edu** i zapisnik **/pub/papers/edgar.ps**.

S obzirom da WWW hipertekstualni koncept omogućava ulančavanje podataka koji mogu biti pohranjeni na različitim Web poslužiocima, HTTP protokol transparentno integrira WWW informatički prostor. Na taj način cijeli Internet i svi u njemu aktivni Web poslužiocu tvore jedinstveni hipertekstualni i hipermedijalni dokument.

Komunikacija između Web korisnika i poslužica odvija se pomoću zaglavlja standardne e-mail specifikacije, definirajući način komunikacije i vrstu podataka. Kada Web poslužilac pošalje podatke korisniku, njima će prethoditi zaglavlje u MIME (eng. "Multipurpose Internet Mail Extension") formatu u kojem se definira vrsta podataka što slijede. Na temelju definicija iz zaglavlja preglednik će pokrenuti odgovarajuće uslužne programe za interpretaciju podataka, npr. "image viewer", "movie player" i slično.

Važna značajka HTTP protokola je intermitentni komunikacijski mehanizam. To znači da nakon uspostave komunikacije između poslužioca i korisnika veza se prekida nakon jednog transakcijskog ciklusa. Ukoliko se HTML dokument sastoji od više cjelina, bit će potrebno ponoviti uspostavu komunikacije da se obavi sljedeći ciklus prijenosa podataka itd. Takav princip usporava donekle HTTP komunikaciju, ali doprinosi dijelom uštedi mrežnih resursa.



## 5.5 NetBEUI/NetBIOS

NetBEUI označava protokol koji se uglavnom koristi u manjim lokalnim mrežama osobnih računala. Nastao je na osnovi LAN mrežne podrške NetBIOS koju su razvili IBM i Sytek 1984. godine kao jednostavan komunikacijski program koji dozvoljava "peer-to-peer" ili korisnik/poslužilac komunikaciju u lokalnoj mreži. Usklađivanje s višerazinskim OSI referentnim modelom, dovelo je do podjele na NetBEUI razinu (eng. "Network BIOS Extended User Interface"), koja definira komunikacijski protokol, i NetBIOS razinu (eng. "Network Basic Input/Output System"), ograničenu na funkcije sučelja mrežnih programa. Odnos NetBEUI/NetBIOS modela i OSI referentnog modela prikazan je na Sl. 5-22.

OSI	
Application	NetBIOS/SMB
Presentation	
Session	
Transport	NetBEUI
Network	
Data Link	
Physical	

Sl. 5-22 Odnos OSI referentnog modela i ustrojstva NetBEUI/NetBIOS protokola

NetBIOS predviđa opće sučelje između mrežnih programa, ali ne pruža specifične usluge, kao što su usluge za korištenje mrežnih pisaača, rad sa zapisnicima u mreži itd. Za takve se usluge brine SMB (eng. "Server Message Bloks"). NetBEUI/NetBIOS osigurava tri osnovne mrežne usluge:

- ❑ **Imeničke usluge** (eng. "Name service") omogućavaju identifikaciju računala. Svako računalo koristi logičko ime za identifikaciju u mreži. Ukoliko želi s nekim komunicirati, računalo emitira poziv. Ostala računala osluškuju svoje ime i kada prime poziv uspostavlja se komunikacija. Takav princip neusmjeravanog emitiranja (eng. broadcasting) uzrokuje intenzivan promet u lokalnoj mreži, što se može odraziti kao usporavanje komunikacijskih odziva.
- ❑ **Usluge sastanaka** (eng. "Session service") osiguravaju razmjenu poruka između računala po uspostavi komunikacije (sastanka). Pritom se vodi računa o redoslijedu poruka i potvrdi njihovog primitka.
- ❑ **Usluge datagrama** (eng. "Datagram service") služe za javno emitiranje poruka neovisno o uspostavljenoj komunikaciji. To su poruke koje nisu upućene konkretnom računalu, već objavljuju neku opću informaciju. Stoga ova usluga ne uključuje provjeru primitka objavljene poruke.

MS Windows operacijski sustavi (3.xx, 95, 98, NT, 2000) koriste NetBEUI/NetBIOS protokol za rad u lokalnoj mreži "Windows for Workgroups" ili "LAN Manager". Glavni nedostatak ovog protokola je nemogućnost usmjeravanja, odnosno komunikacija s drugim podmrežama.

---

## 5.6 IPX/SPX protokol

IPX/SPX protokol potječe od NetWare mrežnog modela razvijenog u tvrtki Novell. Ima vrlo široku primjenu u mrežama osobnih računala. IPX/SPX protokol se bazira na Xerox-ovom XNS protokolu; IPX - "Internet Packet Exchange" koristi XNS IDP (eng. "Internet Datagram Protocol"), a SPX - "Sequenced Packet Exchange" koristi XNS SPP (eng. "Sequenced Packet Protocol"). IPX odgovara mrežnoj (trećoj) razini OSI referentnog modela, a SPX ispunjava zadaće razine transporta, četvrte OSI razine. Za razliku od NetBEUI/NetBIOS protokola, IPX/SPX omogućava usmjeravanje poruka, odnosno komunikaciju između različitih podmreža.

### 5.6.1 IPX

IPX je mrežni protokol koji osigurava usluge prijenosa podatkovnih paketa (datagrama). Protokol ne zahtijeva prethodnu uspostavu veze između računala i zato se naziva "connectionless" protokol. Stoga ne postoji mogućnost provjere primitka odaslanih paketa. IPX je zato prikladan za emitiranje javnih poruka (eng. broadcasting).

IPX datagram sadrži uz podatke i mrežnu adresu. Mrežna adresa obuhvaća 32-bitni broj mreže i 48-bitni broj računala. Slijedi "socket number", odnosno 16-bitni broj koji jednoznačno identificira mrežni program kojem su podaci upućeni.

### 5.6.2 SPX

SPX protokol brine o redoslijedu i toku podataka, osiguravajući da svaki paket stigne na zadanu adresu. Stoga SPX traži prethodnu uspostavu komunikacije između računala pomoću posebne kontrolne poruke. SPX očekuje potvrdu za svaki odaslani paket. Ukoliko unutar određenog vremenskog intervala ne stigne potvrda, proizlazi zaključak da se paket izgubio te se ponovno šalje. Na temelju rednog broja upisanog u zaglavlje paketa osigurava se ispravna sinteza podataka na odredištu.



## 5.7 UNIX u mreži

Operativni sustav UNIX integrira mrežne usluge zajedno s ostalim sistemskim rješenjima, čineći mrežu računala jedinstvenom računalnom okolinom. Mrežne usluge se mogu grupirati u dvije skupine: osnovnih, odnosno najčešće korištenih usluga i dodatnih mrežnih usluga. U osnovne mrežne usluge svrstavaju se programi koji se najčešće koriste:

- ❑ udaljena prijava za rad
- ❑ prijenos zapisnika
- ❑ elektronička pošta

U dodatne mrežne usluge ubrajaju se programi koji :

- ❑ mrežni sustav zapisnika (NFS)
- ❑ korištenje udaljenih pisaača
- ❑ udaljeno izvođenje naredbi
- ❑ imeničke mrežne usluge (DNS, NIS)

### 5.7.1 Ime računala

Računalna komunikacija u mreži podrazumijeva međusobno poznavanje računala i korisnika. Prvo što je potrebno znati o udaljenim računalima su njihova imena. Iza svakog imena računala krije se odgovarajuća Internet (IP) adresa koja se pohranjuje u sistemski zapisnik **/etc/hosts** lokalnog računala ili u se nalazi u nekoj od baza podataka DNS ili NIS poslužilaca mrežnih usluga koje daju informacije o imenima i adresama udaljenih računala. Osnovni princip DNS usluga objašnjen je poglavlju 5.3.5.

Naredba **hostname**, na BSD UNIX-u, ispisuje ime lokalnog računala, npr.:

```
$ hostname
aramis
$
```

Internet adresu lokalnog računala možemo pronaći u zapisniku **/etc/hosts** ili pomoću naredbe **ifconfig**:

```
pro10% ifconfig hme0
hme0: flags=863<UP, BROADCAST, NOTRAILERS, RUNNING, MULTICAST>
mtu 1500
inet 161.53.117.160 netmask ffffffff broadcast 161.53.117.255
pro10%
```

Iz ispisa je vidljivo da je adresa lokalnog računala **161.53.117.160**. U produžetku su ispisani **netmask** varijabla i **broadcast** adresa koji služe za ograničavanje prometa u podmreži određene klase (vidi poglavlje 5.3.4).

Argument naredbe **iconfig** je naziv Ethernet sučelja koje ovisi o konfiguraciji i proizvođaču konkretnom računalu. Ako ne znamo ime sučelja pomoći će nam naredba **"netstat -i"**:

```

pro10% netstat -i
Name Mtu Net/Dest Address IpKts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 452109 0 452109 0 0 0
hme0 1500 pro10 pro10 12040362 0 7242413 0 1146581 0

```

Naredba **netstat** daje prikaz za dva sučelja: "**hme0**" i "**lo0**" čija su imena navedena u prvom stupcu. Za svako sučelje je prikazana i sakupljena statistika. "**Hme0**" je ime za standardno sučelje, a "**lo0**" je ime lokalnog sučelja koje se koristi za "loopback" testiranje.

## 5.7.2 Provjera

Iako je računalna komunikacija vrlo pouzdana, u isto je vrijeme vrlo složena i može ovisiti o velikom broju posrednih mrežnih uređaja, što može biti uzrokom određenih problema. Ukoliko je neki od ključnih mrežnih čvorova ili ciljno računalo izvan funkcije, a ne postoje alternativni uređaji, komunikacija neće biti moguća. Pomoću naredbe **ping** moguće je ispitati mogućnosti komuniciranja. Pri provjeri valja početi od vlastitog računala, a zatim ispitati komunikaciju s udaljenim računalom. Za vlastito računalo potrebno je koristiti naredbu "**ping localhost**". Argument **localhost** standardno je pridjeljena adresa 127.0.0.1 koja adresira lokalno mrežno sučelje - *loopback*.

```

pro10% ping -s localhost
PING localhost: 56 data bytes
64 bytes from localhost (127.0.0.1): icmp_seq=0.
time=0. ms
64 bytes from localhost (127.0.0.1): icmp_seq=1.
time=0. ms
64 bytes from localhost (127.0.0.1): icmp_seq=2.
time=0. ms
64 bytes from localhost (127.0.0.1): icmp_seq=3.
time=0. ms
64 bytes from localhost (127.0.0.1): icmp_seq=4.
time=0. ms
^C
----localhost PING Statistics----
5 packets transmitted, 5 packets received, 0% packet
loss
round-trip (ms)  min/avg/max = 0/0/0
pro10%

```

Nakon uspostave veze **ping** odašilje pakete (datagrame) veličine 64 okteta i povratno ih prima koristeći jednostavni ICMP protokol. Nakon prekida (**Ctrl-c**) usporedbom poslanih i primljenih paketa dobiva se statistika koja pokazuje uspješnost komunikacije. Izravno o tome govori podatak **packet loss** (izgubljeno paketa). Taj bi podatak za lokalno računalo morao biti 0%, jer se na lokalnoj razini ne očekuju smetnje koje utječu na gubitak slijeda podataka. Za testiranje komunikacije s udaljenim računalom naredba **ping** se koristi na isti način s razlikom da se kao argument zadaje ime udaljenog računala (njegova Internet adresa mora biti upisana u lokalnu **/etc/hosts** datoteku ili pohranjena u imeničkoj bazi nekog od NIS ili DNS poslužioca):

```

pro10% ping -s www.fesb.hr
PING marjan.fesb.hr: 56 data bytes
64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=0.
time=12. ms

```



```

64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=1.
time=7. ms
64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=2.
time=7. ms
64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=3.
time=7. ms
64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=4.
time=6. ms
64 bytes from marjan.fesb.hr (161.53.166.3): icmp_seq=5.
time=7. ms
^C
----marjan.fesb.hr PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 6/7/12
pro10%

```

Podaci **min/avg/max = 6/7/12** prikazuju statistiku o brzini komuniciranja. Prvi broj predstavlja minimalno vrijeme putovanja paketa između računala, drugi broj prikazuje prosječno vrijeme putovanja dok treći broj prikazuje najduže vrijeme putovanja poruke. Sva vremena su dana u milisekundama.

Ukoliko veza s udaljenim računalom nije uspješno uspostavljena dobit ćemo podatak **100% packet loss**:

```

pro10% ping -s pro5.sjever.fsb.hr
PING pro5.sjever.fsb.hr: 56 data bytes
^C
----pro5.sjever.fsb.hr PING Statistics----
4 packets transmitted, 0 packets received, 100% packet
loss
pro10%

```

Bez opcije **-s** naredba **ping** će javiti mogućnost komunikacije bez statistike, koristeći formulaciju **"alive"** (komunikacija moguća, udaljeno računalo je "živo") ili **"not alive"** (udaljeno računalo nije dostupno):

```

pro10% ping bjesomar.srce.hr
bjesomar.srce.hr is alive
pro10%

```

### 5.7.3 Prijava na udaljeno računalo

Prijava na višekorisničko računalo (lokalno ili udaljeno) moguća je jedino za prethodno autorizirane korisnike. To znači da je potrebno posjedovati "korisnički prostor" (autorizirani imenik) na računalu i poznavati prijavno ime i zaporku (lozinku, eng. password). Za prijavu rada na udaljeno računalo mogu se koristiti dvije naredbe koje u osnovi obavljaju istu funkciju: **telnet** i **rlogin**.

Naredba **telnet** predstavlja implementaciju službene mrežne usluge koja je definirana s RFC 854 i RFC 855. **Telnet** je standardni uslužni program u svim implementacijama TCP/IP-a, neovisno o operativnom sustavu.

Naredba **rlogin** pojavljuje se samo na UNIX računalima i ne pripada standardnim TCP/IP mrežnim uslugama.

**TELNET**

Zadatak naredbe **telnet** je uspostava TCP/IP komunikacije s udaljenim računalom. Po uspostavi veze, sa udaljenog računala javlja se "telnet daemon" (rezidentni **telnet** program koji se brine o udaljenoj prijavi). Udaljeni **telnet** proces započinje s postupkom prijave. Ukoliko je autorizacija potvrđena otvara se na udaljenom računalu UNIX Shell (UNIX korisnička ljuška) i u nastavku korisnik može koristiti udaljeno računalo na isti način kao i lokalno, izdajući naredbe u terminalskom prozoru koje će obrađivati procesor udaljenog računala. Npr.:

```
pro10% telnet alpha.fsb.hr
Trying 161.53.116.11...
Connected to alpha.fsb.hr.
Escape character is '^]'.

Digital UNIX (alpha.fsb.hr) (ttypl)

login: oskar
Password:
Last successful login for bojan: Thu Dec  9 13:43:36
1999 from bojan@pro10.sjever.fsb.hr

Digital UNIX V4.0F (Rev. 1229); Mon Jul 17 15:32:04
MET DST 2000

alpha.fsb.hr>
```

Poruka o "**Escape**" znaku '^]' znači mogućnost prijelaza u interaktivni rad s **telnet** naredbom. U interaktivnom radu **telnet** podržava nekoliko naredbi čiju namjenu možemo upoznati pomoću naredbe **help**:

```
pro10% telnet pro1
Trying 161.53.117.151...
Connected to pro1.sjever.fsb.hr.
Escape character is '^]'.

SunOS 5.7

login: artur
Password:
Last login: Fri Apr  7 14:59:32 from pro10
Sun Microsystems Inc. SunOS 5.7 Generic October 1998
Fri Aug  4 12:43:42 MET DST 2000
pro1%
telnet> help
Commands may be abbreviated.  Commands are:

close      close current connection
logout     forcibly logout remote user and close the connection
display    display operating parameters
mode       try to enter line or character mode
open       connect to a site
quit       exit telnet
send       transmit special characters ('send ?' for more)
```

```

set      set operating parameters ('set ?' for more)
unset    unset operating parameters ('unset ?' for more)
status   print status information
toggle   toggle operating parameters ('toggle ?' for more)
slc      change state of special charaters ('slc ?' for more)
z        suspend telnet
!        invoke a subshell
environ  change environment variables ('environ ?' for more)
?        print help information
<return> leave command mode
telnet>

```

pro1%

Najvažnija je naredba **z** koja omogućava privremeni prelazak na Shell lokalnog računala. Po završetku lokalnog rada vraćamo se na udaljeno računalo naredbom **exit**. Npr., ukoliko smo već prijavljeni na udaljenom računalu, opisani postupak bi izgledao kao na sljedećem primjeru:

```

$ hostname      (naredba na udaljenom računalu)
pro1            (ime udaljenog računala je "pro1")
$ ^]           (prijelaz u interaktivni rad s telnet naredbom)
telnet> z
pro10> hostname (naredba na udaljenom računalu)
pro10          (ime lokalnog računala "pro10")
pro10> exit    (završetak privremenog rada na lokalnom računalu)
$ hostname
pro1
$ exit        (kraj rada s telnet-om, povratak na lokalno računalo)
Connection closed
pro10>

```

Kod primjene telnet-a mogu se pojaviti problemi s tumačenjem terminalskih normi, odnosno interpretacija posebnih slogova koje terminali koriste za grafičke efekte (poglavito kod primjene **vi** editora ili naredbe **more**). Telnet izvorno podržava VT100 normu. Stoga je potrebno na udaljenom računalu, čija UNIX ljuska koristi drugačiji terminal, postaviti parametre terminala na sljedeći način:

```
$ setenv TERM vt100 (za C-Shell)
```

ili

```
$ TERM=vt100; export TERM (za Bourne ili Korn Shell)
```

Za završetak rada i prekid veze s udaljenim računalom potrebno je izvršiti odjavu rada pomoću naredbe **logout** ili aktiviranjem kombinacije tipki **Ctrl-d**.

## R LOGIN

Naredba **rlogin** obavlja istu funkciju kao i **telnet**. Razlika se prvenstveno očituje u tome da **rlogin**, jednako kao i ostale UNIXove "remote" naredbe: **rsh** i **rsh**, dodatno koristi UNIXov protokol za udaljeni pristup. To se očituje u mogućnosti dodatne autorizacije vlastitog pristupa, ili pak pristupa nekog drugog korisnika, udaljenom računalu. Dodatna autorizacija podrazumijeva izravnu prijavu na udaljeno

računalo bez zaporke. Za takav način korištenja potrebno je u osobnom imeniku (**\$home**) na udaljenom računalu kreirati zapisnik **.rhosts** (od eng. remote hosts, udaljeni domaćini/računala). U zapisnik treba upisati naziv računala za kojeg se želi izvršiti autorizacija pristupa. Uz računalo treba navesti korisničko ime, ako je različito na tom računalu:

```
#Primjer za .rhosts
#Ime računala      Korisnik
alpha1.fsb.hr      oskar
pro7                artur
```

Mogu se navesti i imena drugih korisnika kojima se želi dati mogućnost pristupa na svoj korisnički prostor<sup>29</sup>.

Primjer primjene naredbe **rlogin** za dodatno autorizirani pristup:

```
pro10% rlogin pro1
Last login: Fri Aug  4 12:43:42 from pro10
Sun Microsystems Inc. SunOS 5.7 Generic October 1998
Fri Aug  4 12:48:43 MET DST 2000
pro1%
```

U slučaju da ne postoje odgovarajuća autorizacija u zapisniku **.rhosts** ili **/etc/hosts.equiv** udaljenog računala:

```
$ rlogin pro4
Password: _____
Login incorrect
Login:
```

Isti oblik dodatne autorizacije može se izvršiti na sistemskoj razini u zapisniku **/etc/hosts.equiv**. Međutim, pristup i mogućnost kreiranja podataka u taj zapisnik ima samo ovlašteni korisnik "super-user" ("root"), odnosno inženjer sustava.

Po prijavi naredba **rlogin** automatski izvodi sve potrebne naredbe iz zapisnika **.login**, **.profile** i **.cshrc**, što naredba **telnet** ne radi. Na taj način postavljaju se svi predviđeni parametri operativnog sustava kao kod lokalne prijave.

#### 5.7.4 Prijenos zapisnika

Udaljena prijava omogućava interaktivnu komunikaciju s udaljenim računalom, zadavanje naredbi i praćenje rezultata. Ukoliko želimo razmjenjivati zapisnike između računala potrebno je koristiti uslužne programe **ftp** ili **rnp**. Oba programa koriste TCP/IP protokol. Naredba **ftp** pripada porodici TCP/IP mrežnih usluga dok je **rnp** standardna UNIX naredba. Razlika se ponajviše očituje u načinu korištenja. **Ftp** se primjenjuje slično kao i **telnet**, odnosno korisnik se prijavljuje na udaljeno računalo i potom izvodi odgovarajuće naredbe za razmjenu zapisnika. **Rcp** više naliči običnoj naredbi za kopiranje zapisnika.

---

<sup>29</sup> Takav oblik autorizacije drugih korisnika nije uobičajen zbog sigurnosnih aspekata.

**FTP**

FTP, kao protokol i mrežna usluga, definiran je s RFC 959. Naredba prvo traži prijavu na udaljeno računalo kako bi se provjerila autorizacija korisnika. To znači da korisnik mora imati odgovarajuće korisničko ime i zaporku za pristup udaljenom računalu, slično kao i kod naredbe **telnet**. Primjenjuje se sa ili bez argumenta. Argument označava ime udaljenog računala, npr.:

```

pro10% ftp pro1
Connected to pro1.sjever.fsb.hr.
220 pro1 FTP server (SunOS 5.7) ready.
Name (pro1:oskar):
331 Password required for oskar.
Password:
230 User oskar logged in.
ftp>

```

Nakon prijave javlja se "ftp daemon" sa udaljenog računala ispisujući na zaslonu prompt **ftp>**. Popis i objašnjenje **ftp** naredbi možemo dobiti pomoću naredbe **help**.

```
ftp> help
```

Commands may be abbreviated. Commands are:

!	cr	macdef	open	send
\$	delete	mdelete	prompt	status
account	debug	mdir	proxy	struct
append	dir	mget	sendport	pwd
ascii	disconnect	sunique	put	tenex
bell	form	mkdir	quit	trace
binary	get	mls	quote	type
bye	glob	mode	recv	user
case	hash	mput	remotehelp	rename
cd	help	nmap	reset	?
cdup	lcd	verbose	rmdir	
close	ls	ntrans	runique	

```
ftp>
```

Naredbe sliče UNIX naredbama, a posebnost je da neke od naredaba imaju dvostruku sintaksu. Tako se npr., naredba za ispis sadržaja imenika **ls -l** može izvesti i pomoću naredbe **dir** (koja se podudara sa sintaksom VMS i MS-DOS operativnih sustava):

```

ftp> ls -al
200 PORT command successful.
150 ASCII data connection for /bin/ls (161.53.117.160,42016) (0
bytes).
total 572
drwxr-x--x  32 bojan  staff      2048 Jul 28 11:22 .
drwxr-xr-x   3 bojan  staff      512 Oct 17 1997 ..
-rw-----   1 bojan  staff      589 Jul 28 11:22 .Xauthority
-rw-r--r--   1 bojan  staff     1572 Oct  8 1999 .cshrc
-rw-r--r--   1 bojan  staff     3602 Sep 13 1999
.desksetdefaults
drwxr-xr-x  15 bojan  staff      512 Jul 28 12:30 .dt
-rwxr-xr-x   1 bojan  staff     5400 Nov 27 1997 .dtprofile
drwxr-xr-x   2 bojan  staff      512 Sep 23 1999 .hotjava

```

```

-rw-r--r-- 1 bojan staff      84 May 12 1997 .login
-rw-r--r-- 1 bojan staff     776 Jan 10 1997 .mailrc
drwx----- 6 bojan staff     512 Jul 28 12:30 .netscape
-rw----- 1 bojan staff    127964 Sep 18 1998 .newsrc
-rw-r--r-- 1 bojan staff     174 Mar 19 1996 .profile
-rw-r--r-- 1 bojan staff     291 Nov 14 1997 .rhosts
drwxr-xr-x 2 bojan staff     512 Jan 19 1999 .wastebasket
drwxr-xr-x 2 bojan staff     512 Jan  8 1998 Mail
-rwxr-xr-x 1 bojan staff     501 Jun 23 1997 SunVideo
drwxr-xr-x 2 bojan staff    1536 Apr  4 12:03 WWW
drwxr-xr-x 15 bojan staff   3584 Apr  4 12:14 cad
-rw-r--r-- 1 bojan staff   3657 Aug 28 1998 cde-
ideas.html
-rwxr-xr-x 1 bojan staff      61 Oct 22 1996 cp-lic1
-rw----- 1 bojan staff   3602 Oct  8 1997 mbox
drwx----- 2 bojan staff   1024 Jul 28 11:24 nsmail
drwxr-xr-x 2 bojan staff     512 Apr  4 1997 papers
drwxr-xr-x 5 bojan staff   1024 Nov 23 1998 prg
drwxr-xr-x 4 bojan staff   2048 Jun  6 11:15 public_html
drwxr-xr-x 3 bojan staff     512 Mar  4 13:43 slike
drwxr-xr-x 2 bojan staff   1024 Nov 27 1997 sys
drwxr-xr-x 3 bojan staff     512 Feb 17 09:00 tmp
226 ASCII Transfer complete.
remote: -al
4722 bytes received in 0.32 seconds (14.28 Kbytes/s)
ftp>

```

Prije nego se odlučimo za prijenos zapisnika moramo odrediti vrstu podataka koji će se kopirati. U pravilu **ftp** prenosi znakovne ASCII zapisnike. Ako se radi o binarnim podacima (programima, komprimiranim zapisnicima, slikama, video ili audio zapisnicima) potrebno je prethodno zadati naredbu **binary**. Promjena na ASCII prijenos podataka dobiva se pomoću naredbe **ascii**.

Nakon što smo ispravno odredili vrstu podataka koju prenosimo možemo pristupiti kopiranju zapisnika. Za prijenos zapisnika sa udaljenog računala koristi se naredba **get**, a za prijenos zapisnika na udaljeno računalo naredba **put**. Naredba **get** će kopirati zapisnik u tekući imenik lokalnog računala, odnosno imenik iz kojeg je pozvana naredba **ftp**. Prilikom prijenosa zapisnika sa lokalnog na udaljeno računalo, naredba **put** kopirat će podatke u tekući imenik udaljenog računala:

```

ftp> binary
200 Type set to I.
ftp> get sklop1.Z
200 PORT command successful.
150 Opening BINARY mode data connection for sklop1.Z
(161.53.117.160,42021) (372
54 bytes).
226 Transfer complete.
local: sklop1.Z remote: sklop1.Z
37254 bytes received in 0.06 seconds (601.95 Kbytes/s)
ftp> quit
221 Goodbye.
pro10%

```

Promjena trenutnog imenika moguće je ostvariti pomoću naredbe **cd**. Ukoliko se želi promijeniti imenik na lokalnom računalu potrebno je koristiti naredbu **!cd**:

```

ftp> cd slike      (promjena imenika na udaljenom računalu)
ftp> !cd download (promjena imenika na lokalnom računalu)

```

```
ftp> !ls -l
...
ftp>
```

(sadržaj imenika na lokalnom računalu)

Znak **!** može se primijeniti i kod ostalih **ftp** naredbi označavajući izvedbu na lokalnom računalu.

U slučaju da se želi kopirati više zapisnika primjenom meta-znakova mogu se koristiti naredbe **mget** i **mput**, koje prihvaćaju argumente kao što je npr. **\*.dat** i slično. Međutim, ne postoji formulacija u izvornoj verziji programa **ftp** koja omogućava kopiranje podimenika. Prilikom kopiranja niza zapisnika zadanih meta-znakovima **ftp** će zahtijevati potvrdu za svaki sljedeći zapisnik.. Ako ne želimo potvrdu za svaki zapisnik, možemo zadati **ftp** naredbu **prompt** ili u pozivu naredbe **ftp** dodati opciju **-i** (**ftp -i**).

Ako se namjerava izvršiti prijenos veće količine podataka uputno je koristiti neinteraktivni pristup u kombinaciji s UNIXovom naredbom **at**. U tom je slučaju prvo potrebno kreirati komandni program koji će pokrenuti **ftp**, npr.:

```
#
# Shell program za neinteraktivni prijenos podataka
#
ftp -in << XXXX prol.sjever.fsb.hr
user oskar abcdxy # neinteraktivna prijava
cd /pub          # pridruživanje imenika u kojem je zapisnik
get podaci.dat   # kopiranje zapisnika
XXXX
```

U komandnom programu **ftp** naredba se poziva sa opcijama **-in**. Nova opcija **-n** omogućuje da se ime udaljenog korisnika posebno definira. Niz znakova "**<< XXXX**" označava da će se za pokrenutu **ftp** naredbu ulaz uzimati iz redaka koji slijede i da će kraj ulaza označiti niz znakova "**XXXX**". Kreirani program potrebno je potom aktivirati pomoću UNIX naredbe **at** u zadano vrijeme:

```
pro10% echo "/bin/csh /usr4/user/pokreni_ftp.sh" | at 10:00p
```

Ovako zadana naredba **at** automatski će pokrenut komandni program **pokreni\_ftp.sh** u 22 sata.

FTP kao protokol predviđa i usluge za korisnike koji nisu autorizirani na udaljenom računalu. Takav oblik usluge traži instalaciju tzv. "*anonymous*" *ftp poslužitelja* na udaljenom računalu. Za njega se kreira jedan ograničeni zapisnički sustav unutar kojeg se mogu kretati i njime se služiti korisnici koji se "ftp daemonu" predstavljaju kao **anonymous** (bezimen, anonimni). Za lozinku treba unijeti svoju e-mail adresu:

```
pro10% ftp prol.sjever.fsb.hr
Connected to prol.sjever.fsb.hr.
220 prol FTP server (SunOS 5.7) ready.
Name (prol.sjever.fsb.hr:oskar): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> dir
200 PORT command successful.
150 ASCII data connection for /bin/ls (161.53.117.160,42070)
(0 bytes).
```

```

total 18
dr-xr-xr-x  9 root    1000      512 Jun  9 12:43 .
dr-xr-xr-x  9 root    1000      512 Jun  9 12:43 ..
drwxr-xr-x  3 root    other     512 Jun 14 10:52
Kolokvij
dr-xr-xr-x  2 root    other     512 Mar 18 1997 bin
dr-xr-xr-x  2 root    other     512 Mar 18 1997 dev
dr-xr-xr-x  2 root    other     512 Mar 18 1997 etc
drwxr-xr-x  3 root    other     512 Jun  9 12:44 nastava
dr-xr-xr-x  4 root    other     512 Nov 14 1997 pub
dr-xr-xr-x  4 root    other     512 Mar 18 1997 usr
226 ASCII Transfer complete.
547 bytes received in 0.065 seconds (8.22 Kbytes/s)
ftp>

```

Takav anonimni pristup udaljenim računalima uobičajen je za tzv. "shareware" ili "public domain" poslužitelje, koji predstavljaju javne baze podataka nudeći određene sadržaje na slobodno korištenje i kopiranje.

FTP uslugu moguće je susresti implementiranu u različitim programima s grafičkim korisničkim sučeljem ili WWW preglednicima, koji njenu primjenu čine još jednostavnijom.

### R C P

Upotreba naredbe **rcp** vrlo je jednostavna i njezina sintaksa odgovara uobičajenoj naredbi za kopiranje zapisnika. Prvi argument definira zapisnik koji se želi kopirati (prenijeti), a drugi argument definira ime kopije. Razlika je u tome da argument može dodatno sadržavati oznaku računala na kojem se zapisnik nalazi, npr.:

```
pro10% rcp pro1:/usr/user/slika.jpg download/nova-slika.jpg
```

U gornjem primjeru prvi argument definira ime udaljenog računala **pro1**, sa kojeg se želi kopirati zapisnik **/usr/user/slika.jpg**. Drugi argument definira ime zapisnika u koji se vrši kopiranje na lokalnom računalu.

Naredba će biti uspješno obavljena teka ako korisnik ima pristup udaljenom računalu, uključujući dodatnu autorizaciju za korištenje UNIX "remote" usluga. To znači da na svojem korisničkom imeniku udaljenog računala mora posjedovati zapisnik **.rhosts** u kojem je navedeno ime računala s kojeg se izvodi kopiranje (slično kao i za primjenu naredbe **rlogin** bez zaporke). U protivnom kopiranje neće uspjeti, a operativni sustav će objaviti grešku "**permission denied**".

Važna značajka naredbe **rcp** je mogućnost prenošenja imenika, i svih njegovih podimenika. U tom slučaju treba koristiti opciju **-r**. Za prijenos imenika s lokalnog računala na udaljeno računalo naredba ima sljedeći oblik:

```
pro10% rcp -r /usr/user/artur/video pro1:/usr/user/artur/tmp
```

U ovom primjeru oznaku udaljenog računala sadrži drugi argument, jer se vrši kopiranje s lokalnog na udaljeno računalo. Dakle, ako ispred zapisnika navedemo neko ime koje završava s dvotočkom podrazumijevat će se da je to ime udaljenog računala.



### 5.7.5 Elektronička pošta

Razmjena poruka pomoću računala, odnosno elektronička pošta (**e-mail**), najstariji je oblik široke primjene računalnih mreža. Danas predstavlja, zajedno s telekomunikacijskim uslugama (WAP, SMS), nezamjenjivi način globalne komunikacije u svijetu. Usluge e-maila raspoložive su u obliku različitih korisničkih programa, najčešće s grafičkim interaktivnim sučeljem koje dozvoljava razmjenu i interpretaciju multimedijalnih sadržaja - glasovnih i vizualnih podataka. Svi takvi programi u osnovi koriste isti protokol – SMTP (eng. "Simple Mail Transfer Protocol"), protokol za razmjenu poruka iz porodice TCP/IP protokola, i istu uslugu – **sendmail**, program za slanje i primanje poruka pomoću SMTP-a. Glavne funkcije usluge elektroničke pošte objedinjene su u naredbi **mail**, standardnoj naredbi UNIX operativnog sustava.

#### MAIL

Slanje i primanje pisama na računalu može se odvijati u lokalnoj mreži ili na globalnoj (Internet) razini. S obzirom da je glavni mrežni protokol Interneta TCP/IP, to znači da će i korištenje elektroničke pošte slijediti pravila poznata kod drugih IP mrežnih uslužnih programa. To se ponajprije odnosi na adresiranje pisama.

Za **slanje** pisma potrebno je jednostavno zadati naredbu **mail** (ili Mail, Berkeley inačicu<sup>30</sup>) s argumentom koji adresira primaoca, npr.:

```
pro10% mail marko@sjever.fsb.hr
```

Adresa **marko@sjever.fsb.hr** sadrži dva dijela odvojena znakom @ (at, eng. monkey - majmun). Prvi dio adrese (**marko** u primjeru) definira ime primaoca, najčešće to je korisničko ime, ali može biti i "*mail alias*", odnosno ime posebno kreirano za potrebe elektroničke pošte. Drugi dio adrese (**sjever.fsb.hr** u primjeru) definira domenu primatelja uskladu s Internet formatom. Drugi oblici adresiranja, immanentni npr. UUCP ili X.400 protokolima, vrlo se rijetko koriste.

Nakon što je izdana naredba **mail** s ispravnom adresom primaoca slijedi upis naslova poruke (**Subject: ...**) i njenog sadržaja:

```
pro10% mail marko@sjever.fsb.hr
Subject: Sretna Nova Godina
Sve najbolje u Novoj Godini zeli ti
tvoj brat Vedran.
^D                                     (Ctrl-d ili . za kraj)
pro10%
```

Pismo se prethodno može pohraniti u zapisnik pomoću nekog od raspoloživih programa za kreiranje teksta, npr. **vi** editora. Tada sadržaj poruke nije potrebno upisivati interaktivno po aktiviranju naredbe **mail**, već je dovoljno formulirati naredbu pomoću redirekcije < **ime zapisnika**:

```
pro10% mail marko@sjever.fsb.hr < pismo.txt
```

Čitanje pristiglih pisama obavlja se pomoću naredbe **mail** bez argumenta. Ako ne postoji niti jedno novo ili nepročitano pismo pojavit će se poruka "**No mail for** (*ime*)

---

<sup>30</sup> Mail, Berkeley e-mail inačica, ponešto se razlikuje od mail programa, poglavito u pogledu sintakse naredbi.

*korisnika*)". Kada stigne novo pismo operativni sustav će upozoriti korisnika nakon prijave ili u tijeku rada porukom "**You have mail**" (Imate poštu). Izvođenjem naredbe **mail** pojavit će se sadržaj zadnje poruke:

```
pro10% mail
From bojan@pro1.sjever.fsb.hr Fri Aug 25 10:58:39 2000
From: bojan@pro1.sjever.fsb.hr
Date: Fri, 25 Aug 2000 11:01:54 +0200 (MET DST)
Message-Id: <200008250901.LAA02309@pro10.sjever.fsb.hr>
Content-Length: 46
```

```
Proba
Slijedi tekst poruke.
Pozdrav,
```

```
Bojan.
```

```
?
```

Pomoću naredbe **h** (eng. head – zaglavlje) **mail** će ispisati zaglavlja (naslove) svih pristiglih poruka, npr.:

```
? h
2 letters found in /var/mail/bojan, 0 scheduled for
deletion, 0 newly arrived
> 2 1178 bojan@pro1.sjever.fsMon Aug 28
11:59:57 2000 Mon Aug 28 11:59:
57 2000
1 1166 bojan@pro1.sjever.fsMon Aug 28
11:59:51 2000 Mon Aug 28 11:59:
51 2000
?
```

Primjenom **Mail** naredbe slijedi ispis zaglavlja (naslova) svih pristiglih poruka:

```
pro10% Mail
mailx version 5.0 Tue Oct 6 00:48:25 PDT 1998 Type ?
for help.
"/var/mail/bojan": 2 messages 2 new
>N 1 bojan@pro1.sjever. Mon Aug 28 11:59 26/1166
proba1
N 2 bojan@pro1.sjever. Mon Aug 28 11:59 25/1178
proba2
?
```

U ispisu se također navodi ime sistemskog poštanskog sandučića u koji se sprema pristigla pošta. To je najčešće zapisnik pohranjen u imeniku **/var/mail** istog imena kao i korisnik.

Znak **?** (ili **&**) označava prompt **mail** programa. Tu možemo unijeti neku od **mail** naredbi. Popis i kratko objašnjenje naredbi dobiva se pomoću naredbe **?**:

```
? ?
? print this help message
# display message number #
- print previous
+ next (no delete)
! cmd execute cmd
```

```

<CR>          next (no delete)
a             position at and read newly arrived mail
d [#]        delete message # (default current message)
dp           delete current message and print the next
dq           delete current message and exit
h a          display all headers
h d          display headers of letters scheduled for deletion
h [#]        display headers around # (default current message)
m user       mail (and delete) current message to user
n            next (no delete)
p            print (override any warnings of binary content)
P            override def.'brief' mode and disp. ALL header lines
q, ^D       quit
r [args]     reply to (and delete) current letter via mail [args]
s [files]    save (and delete) current message (default mbox)
u [#]        undelete message # (default current message)
w [files]    save (and delete) current message without header
x            exit without changing mail
y [files]    save (and delete) current message (default mbox)
?

```

U slučaju **Mail** inačice koristi se naredba **help** ili **?**:

```

? help
      mailx commands

alias,group user ... declare alias for user names
alternates user      declare alternate names for your
                     login
cd,chdir [directory] chdir to directory or home if none
                     given
!command            shell escape
copy [msglist] file save messages to file without
                     marking as saved
delete [msglist]    delete messages
discard,ignore header discard header field when printing
                     message
dp,dt [msglist]     delete messages and type next
                     message
echo string         print the string
edit [msglist]      edit messages
folder,file filename change mailboxes to filename
folders            list files in directory of current
                     folder
followup [msglist]  reply to authors of messages and
                     save copy
Followup [message] reply to all recipients of message
                     and save copy
from [msglist]      give header lines of messages
header [message]    print page of active message headers
help,?              print this help message
hold,preserve [msglist] hold messages in mailbox
inc                 incorporate new messages into
                     current session
list                list all commands (no explanations)
mail user           mail to specific user
Mail                mail to specific user, saving copy
mbox [msglist]     messages will go to mbox when
                     quitting
next [message]      goto and type next message
pipe,| [msglist] shell-cmd pipe the messages to the shell comm.
print,type [msglist] print messages
Print,Type [msglist] print messages with all headers
quit                quit, preserving unread messages
reply,respond [msglist] reply to authors (only) of the

```

Reply, Respond [message]	messages reply to the author and recipients of the msg
save [msglist] file	save (appending) messages to file
Save [msglist]	save messages to file named after author
set variable[=value]	set variable to value
size [msglist]	print size of messages
source file	read commands from file
top [msglist]	print top 5 lines of messages
touch [msglist]	force the messages to be saved when quitting
undelete [msglist]	restore deleted messages
undiscard, unignore header	add header field back to list printed
unread, new [msglist]	mark messages unread
version	print version
visual [msglist]	edit list with \$VISUAL editor
write [msglist] file	write messages without headers
xit, exit	quit, preserving all messages
z [+/-]	display next [last] page of 10 headers

[msglist] is optional and specifies messages by number, author, subject or type. The default is the current message.  
?

Za čitanje pisma potrebno je navesti redni broj pisma, a ako se žele pisma čitati redom dovoljno je pritiskom na tipku Enter prelaziti njihovim sadržajem:

```
pro10% mail
From bojan@pro1.sjever.fsb.hr Fri Aug 25 10:58:39 2000
From: bojan@pro1.sjever.fsb.hr
Date: Fri, 25 Aug 2000 11:01:54 +0200 (MET DST)
Message-Id: <200008250901.LAA02309@pro10.sjever.fsb.hr>
Content-Length: 46
```

```
Proba
Slijedi tekst poruke.
Pozdrav,
```

Bojan.

?

ili

```
? 1
From bojan@pro1.sjever.fsb.hr Fri Aug 25 10:58:39 2000
From: bojan@pro1.sjever.fsb.hr
Date: Fri, 25 Aug 2000 11:01:54 +0200 (MET DST)
Message-Id: <200008250901.LAA02309@pro10.sjever.fsb.hr>
Content-Length: 46
```

```
Proba
Slijedi tekst poruke.
Pozdrav,
```

Bojan.

? quit

Saved 1 message in /ul/tom/mbox

```
Held 0 messages in /usr/spool/mail/tom.
pro10%
```

Ako želimo obrisati pročitano pismo upotrijebit ćemo naredbu **d** (ili **delete** u **Mail** naredbi). Istom naredbom možemo obrisati više pisama ako se uz nju navedu odgovarajući redni brojevi pisama.

Na pristiglo pismo možemo izravno odgovoriti pomoću naredbe **r** (ili **reply**). Naredba **r** adresirat će odgovor na autora pisma, a naredba **Reply** (s velikim početnim slovom u **Mail** naredbi) adresirat će odgovor ujedno na sve primatelje koji su u poruci navedeni.

Pomoću naredbe **s** (ili **save**) moguće je pohraniti pismo u poseban zapisnik. Izlaz iz **Mail** naredbe izvodi se na dva načina, pomoću naredbe **quit** ili **exit**. Naredbom **quit** izlazi se iz programa **mail** premještanjem pristigle pošte iz sistemskog zapisnika **/var/spool/mail/ime\_korisnika** u osobni poštanski sandučić (zapisnik) **mbox** koji se nalazi u korisničkom imeniku. Ako želimo pismo ostaviti u sistemskom sandučiću moramo koristiti naredbu **exit**.

#### PRINCIP SUSTAVA ELEKTRONIČKE POŠTE

E-mail sustav se poglavito oslanja na temeljni mrežni i prijenosni protokol TCP/IP, na kojem je zasnovana mreža svih mreža Internet. Na taj je način izravno osigurana mogućnost globalne primjene ove vrste komunikacije. Međutim, kao i u slučaju ostalih mrežnih uslužnih programa, unutrašnja struktura sustava za poštu je relativno složena. Korisnik uočava tek korisničko sučelje programa za slanje i primanje pošte (npr. **mail**, ili MS Outlook). Osim interakcije s korisnikom "mail programi" imaju zadaću sporazumijevati se s programom koji razumije poštanski protokol SMTP. Takvi se programi obično zovu "Mail Transport Agent" (MTA). U UNIX operativnom sustavu to je program **sendmail**. Takav program koristi TCP/IP protokol za uspostavu komunikacije s udaljenim računalom na kojemu adresirani korisnik ima poštanski sandučić, odnosno koristi ga kao poštanskog poslužioaca. Komunikacija se u biti uspostavlja između udaljenih MTA programa koristeći zajednički "SMTP jezik".

U praksi se koristi i POP - "Post Office Protocol". POP je protokol koji povezuje računala sa središnjim mail poslužiocem, ograničavajući pristup samo autoriziranim korisnicima. Na taj način se omogućava udaljeno korištenje mail poslužioaca, koji će pružiti usluge slanja i čitanja pošte za korisnike koje operativni sustav prepozna kao ovlaštene korisnike. Zadnje verzije ovog protokola imaju oznaku POP2 i POP3.

Norme koje definiraju osnove sustava elektroničke pošte, SMTP protokol i RFC 821, 822 norme, nastali su paralelno s razvojem suvremenih računalnih mreža (tijekom osamdesetih godina 20. stoljeća). Međutim, tada nije postojala potreba za prijenosom audio ili vizualnih podataka. Nedostatak podrške za prijenos grafičkih podataka i prijenos pisama čiji je sadržaj sastavljen od znakova međunarodnih kodnih stranica, tražio je proširenje baznih normi u pogledu multimedijalne komunikacije i globalizacije. Tako je nastala relativno nova norma poznata pod imenom MIME.

#### MIME FORMAT

MIME (eng. "Multipurpose Internet Mail Extensions") predstavlja proširenje postojećih normi kojima je definiran format Internet pisma. Format pisma uključuje dvije glavne komponente: zaglavlje i sadržaj. U zaglavlju je definiran pošiljatelj, primatelj, naslov pisma (subject), vrijeme i ostali opći podaci o pismu. Sadržaj pisma

obuhvaća podatke koje je korisnik kreirao kao poruku ili neki drugi oblik informacije. MIME norma uvodi sljedeća nova zaglavlja:

- Mime-Version
- Content-Type
- Content-Transfer-Encoding

Primjer jednog MIME pisma prikazan je u nastavku:

```

Received:
    from alpha1.fsb.hr (alpha1.fsb.hr [161.53.116.11]) by pro1.sjever.fsb.hr
    (8.9.1b+Sun/8.9.1) with ESMTMP id KAA28509 for <bojan@sjever.fsb.hr>; Tue, 18
    Jul 2000 10:53:03 +0200 (MET DST)
with
    Received:
    from alpha2.fsb.hr (alpha2 [161.53.116.12]) by alpha1.fsb.hr (8.9.3/8.9.3)
    ESMTMP id KAA01738 for <bojan@alpha1.fsb.hr>; Tue, 18 Jul 2000 10:55:42 +0200
    (MET DST)
(8.9.3/8.9.3) with
    Received:
    from as104.tel.hr (as104.tel.hr [205.219.255.36]) by alpha2.fsb.hr
    ESMTMP id LAA13136 for <bojan.jerbic@fsb.hr>; Tue, 18 Jul 2000 11:11:45 +0200
    (MET DST)
with
    Received:
    from darko (ad8-m204.tel.hr [195.29.231.204]) by as104.tel.hr (0.0.0/0.0.0)
    ESMTMP id KAA73664 for <bojan.jerbic@fsb.hr>; Tue, 18 Jul 2000 10:57:31 +0200
    (MEST)
Message-ID: <200007180857.KAA73664@as104.tel.hr>
Reply-To: ipz-spelprojekt@zg.tel.hr
-----
From: "Darko" <ipz-spelprojekt@zg.tel.hr>
To: <bojan.jerbic@fsb.hr>
Subject: Obavijest
Date: Tue, 18 Jul 2000 10:54:17 +0200
X-MSMail-Priority: Normal
X-Priority: 3
X-Mailer: Microsoft Internet Mail 4.70.1162
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="====_NextPart_000_01BFF0A6.84A97A40"
Content-Transfer-Encoding: 7bit
X-Mozilla-Status: 8001
X-Mozilla-Status2: 00000000
X-UIDL: abdc7a3380297e620a828af94c0d7ddb

Sastanak će se održati u subotu.

Pozdrav,

Darko.

Content-Type: application/octet-stream; name="IPZ-spelprojekt.doc"
Content-Transfer-Encoding: base64
Content-Description: IPZ-spelprojekt.doc (Microsoft Word Document)
Content-Disposition: attachment; filename="IPZ-spelprojekt.doc"

```

/sadržaj kodiranog dokumenta/

Prva tri retka nakon crte predstavljaju zaglavlje prema RFC 821, 822 specifikaciji. Slijede različiti podaci (datum, tip korisničkog mail programa itd.). Osmi redak definira MIME verziju kao 1.0.

Slijedi MIME zaglavlje **Content-Type** koje definira vrstu podataka u pismi (**multipart/mixed**). To znači da se pismo sastoji od više dijelova (eng. multipart) i da su ti dijelovi pomiješani (eng. mixed) te se moraju interpretirati redom kako su navedeni. Prvi dio poruke je pisani tekst, a drugi dio poruke je kodirani (binarni, heksadecimalni) zapis.

MIME zaglavlje **Content-Transfer-Encoding** definira tip kodiranja (**7bit**, **Base64**) i proizvoljan niz znakova koji označavaju granicu između dijelova pisma. Odvojeni moraju biti podaci koji sadrže čitljivi ASCII tekst, koji se može prenositi klasičnim sustavima, od kodiranih podataka koji mogu sadržavati binarne audio, video ili grafičke informacije.

MIME norma vrlo detaljno opisuje sva moguća **Content-Type** zaglavlja, kao i načine kodiranja. MIME ostavlja mogućnost da korisnici sami definiraju nove tipove sadržaja i nove načine kodiranja. **Content-Type** zaglavlje se najčešće sastoji od dvije komponente odvojene znakom "/". Prva komponenta definira glavni tip, a druga definira podtip sadržaja. Slijedi opis nekih najčešće korištenih definicija **Content-Type** zaglavlja:

<b>Content-Type: text/plain</b>	- čisti neformatirani tekst
<b>Content-Type: text/richtext</b>	- jednostavan tekst s naredbama za formatiranje; omogućuje najosnovnije naredbe za izbor fonta ili prikaza.
<b>Content-Type: multipart/mixed</b>	- pismo od više dijelova koji se moraju prikazivati slijedno jedan iza drugoga
<b>Content-Type: multipart/alternative</b>	- ista poruka u nekoliko alternativnih formata
<b>Content-Type: multipart/parallel</b>	- više poruka koje moraju biti prikazane istovremeno; npr. slike popraćene glasom
<b>Content-Type: image/gif</b>	- grafički prikaz u GIF formatu
<b>Content-Type: application/PostScript</b>	- PostScript tekst

**Content-Type** zaglavlje može također sadržavati definiciju korištenog znakovlja (fontova s posebnim izborom slova), npr.:

**Content-Type: text/plain; charset=ISO-8859-2**

### 5.7.6 Ostale mrežne usluge

Ostale mrežne usluge predstavljaju proširenje standardnih naredbi operativnog sustava, s razlikom mogućnosti izvođenja istih operacija na udaljenim računalima:

<b>rup</b>	- ispis aktivnih računala u mreži primjenom RPC protokola
<b>rusers</b>	- popis aktivnih korisnika u lokalnoj mreži primjenom RPC-a
<b>ruptime</b>	- ispis aktivnih računala u mreži (kao <b>rup</b> )
<b>rwho</b>	- popis aktivnih korisnika u lokalnoj u mreži (kao <b>rusers</b> )
<b>rwall</b>	- šalje poruku aktivnim korisnicima na mreži
<b>rdate</b>	- sinkronizira sistemsko vrijeme s udaljenim računalom
<b>rdump</b>	- kopiranje (back-up) na udaljenu jedinicu vrpce
<b>rrestore</b>	- čitanje kopije s udaljene jedinice vrpce
<b>rdist</b>	- upravlja kopijama zapisnika na više računala
<b>whois</b>	- popis registriranih Internet korisnika
<b>rsh</b>	- udaljeno izvođenje naredbi

Navedeni mrežni programi pripadaju standardnoj TCP/IP porodici usluga i uglavnom su podržani u UNIX operativnom sustavu. Ovisi o radu odgovarajućih sistemskih programa ("daemoni"). Npr., naredbe **ruptime** i **rwho** očekuju komunikaciju s **in.rwhod** daemonom na udaljenim računalima i ako od njega ne dobiju odgovor neće biti u stanju izvršiti uslugu. Naredbe **rup** i **rusers** koriste RPC

protokol ("Remote Procedure Calls") koji je standardno podržan u TCP/IP protokolu:

```
pro10% rup
pro10          up 4 days,  2:41, load average: 0.30, 0.05, 0.01
pro3.sjever.fsb up 23:55,          load average: 0.02, 0.02, 0.02
pro5.sjever.fsb up 6 days, 20:46, load average: 0.14, 0.15, 0.10
studentski-zbor up 6 days, 20:47, load average: 0.02, 0.01, 0.01
pro6.sjever.fsb up 4 days,  2:24, load average: 0.00, 0.00, 0.01
```

Naredba **rup** odašilje upite (eng. broadcast) svim računalima u lokalnoj mreži i ispisuje podatke o njihovoj aktivnosti po redu kojim su pristizali odgovori. U prvom stupcu navedeno je ime računala (npr. **pro3.sjever.fsb**, prvih 15 znakova), zatim slijedi podatak o vremenu u kojem računalo neprekidno radi (**up 6 days, 20:46**) i prosječno opterećenje procesora u zadnjoj minuti, posljednjih 5 i 15 minuta (**load average: 0.14, 0.15, 0.10**).

```
pro10% rusers
Sending broadcast for rusersd protocol version 3...
pro10          bojan
pro3.sjever.fsb. gojani
Sending broadcast for rusersd protocol version 2...
pro10%
```

Ovdje su uglavnom nabrojani programi koji su nastali na BSD verziji UNIX-a, ali su se proširili na sve verzije UNIX-a. Osim programa **whois** ovo su programi koji ne pripadaju standardnoj TCP/IP porodici usluga i mogu se sresti jedino na UNIX računalima.

U korištenju nekih od prikazanih usluga treba voditi računa o opterećenju računala i mreže. Tako je npr. poznato da proces-poslužilac za **rwho** uslugu (**rwhod** daemon) zahtjeva puno memorije i stvara veliki promet po mreži pa se ne preporučuje ako nije neophodno.

#### UDALJENO IZVOĐENJE NAREDBI

Naredba **rsh** (od eng. remote shell) omogućuje izvođenje naredbi na udaljenom računalu:

```
pro10% rsh pro2 ls /usr
```

U gornjem primjeru naredba **rsh** ispisuje sadržaj imenika **/usr** sa udaljenog računala **pro2**. Naredba **ls /usr** bit će proslijeđena računalu **pro2** i tamo aktivirana. Za naredbu **rsh** vrijede ista pravila autorizacije pristupa kao i u slučaju naredbi **rlogin** i **rnp**, a to znači da u zapisniku udaljenog računala **\$home/.rlogin** ili **/etc/hosts.equiv** mora biti navedeno ime računala sa kojega se pristupa i ime odgovarajućeg korisnika.

Kombiniranjem naredbe **rsh** i naredbi za preusmjeravanje (**>**, **>>**, **<**, **<<**) moguće je ostvariti prenošenje podataka sa udaljenog računala, npr.:

```
pro10% rsh pro2 cat podaci.1 > podaci.2
```

Gornji će primjer pokrenuti naredbu **cat** na udaljenom računalu **pro2**, a sadržaj udaljenog zapisnika **podaci.1** pohraniti u lokalni zapisnik **podaci.2**. Drugim



riječima, izvršeno je prenošenje podataka s udaljenog računala **pro2** na lokalno računalo **pro10**. Međutim, ako se znak preusmjerenja zada u navodnicima:

```
pro10% rsh pro2 cat podaci.1 '>' podaci.2
```

onda će preusmjerenje biti interpretirano na udaljenom računalu. Dakle, sadržaj udaljenog zapisnika **podaci.1** (na računalu **pro2**) bit će kopiran u zapisnik **podaci.2** na istom udaljenom računalu **pro2**.

Udaljeno izvođenje naredbi posebno je korisni ukoliko želimo uposliti više računala istovremeno, distribuirajući zadaće (programme) na više udaljenih procesora, čime se rasterećuju resursi lokalnog računala. S obzirom da je Unix operacijski sustav u pravilu podržan X Window grafičkim korisničkim sučeljem, koje pruža posebnu grafičku mrežnu funkcionalnost, moguće je grafičke prikaze udaljenih računala pratiti na lokalnom računalu, kao da se izvode lokalno.

X Window sustav zasnovan je na modelu korisnik-poslužilac. Prema X Window konceptu program-korisnik obavlja obradu grafičkih (i ostalih) podataka, a rezultate (opisane u skladu s X Window protokolom) interpretira i prikazuje program-poslužilac (pretvarajući ih u lokalni jezik grafičkog sučelja). Oba procesa, obrada i interpretacija grafičkih podataka, mogu se obavljati na istom računalu, ali se mogu i razdvojiti na dva ili više računala. Ako se korisnik i poslužilac nalaze na različitim računalima moraju se koristiti mrežnim uslugama za razmjenu informacija od jednog do drugog programa/računala. Program-korisnik mora pri tome znati s kojim računalom, odnosno X poslužiocem, razmjenjuje podatke. Program-korisnik, u pravilu, adresira X poslužioca putem Shell varijable **DISPLAY** Unix operacijskog sustava. Varijabla **DISPLAY** standardno definira lokalnog X poslužioca na sljedeći način:

```
DISPLAY = 0.0
```

Oznaka **0.0** definira logičku adresu lokalnog grafičkog sučelja (**0**) i zaslona (**0**): (pretpostavlja se da jedno grafičko sučelje može imati više zaslona). Stoga, prije pokretanja X programa na udaljenom računalu treba izdati naredbu (prethodno se prijavivši s naredbom **telnet** ili **rlogin**):

```
pro2% setenv DISPLAY pro10.sjever.fsb.hr:0.0
```

Ime računala **pro10.sjever.fsb.hr**: ispred oznake sučelja definira računalo kojemu će X program/korisnik slati grafičke rezultate u X protokolu i od kojega će primiti, u istome formatu, ulazne podatke. Prije pokretanja udaljenog X programa mora se još na lokalnom računalu izdati sljedeća naredba:

```
pro10% xhost + pro2.sjever.fsb.hr
```

Njome se daje dozvola lokalnom X poslužiocu za pružanje usluga X programima koji se izvode na udaljenom računalu **pro2.sjever.fsb.hr**.

### 5.7.7 Mrežni zapisnički sustav - NFS

Mrežni zapisnički sustav - Network File System (NFS) je uslužni protokol koji povezuje Unix zapisničke sustave udaljenih računala. Oslanja se na TCP/IP protokol, točnije na UDP protokol (eng. "User Datagram Protocol"), koji pripada prijenosnoj razini TCP/IP modela. Ovaj protokol ima osnovnu zadaću postavljanje neslijedne

(nepovezane, eng. connectionless) komunikacije između udaljenih računala. Jednostavan je i zato dovoljno učinkovit za ostvarivanje brze razmjene podataka između udaljenih zapisničkih sustava.

Pri povezivanju zapisničkih sustava razlikujemo računalo koje dijeli (daje na raspolaganje) svoj zapisnički sustav (u pravilu samo određeni dio) i računalo koje pristupa udaljenom zapisničkom sustavu. Operacijski sustav računala koje daje na korištenje dio svojeg zapisničkog sustava mora raspolagati podacima o ponuđenim imenicima i računalima koja ih mogu posuditi (priključiti svojem zapisničkom sustavu). Ti se podaci definiraju u odgovarajućem sistemskom zapisniku, npr. **/etc/dfs/dfstab** ili **/etc/exports** (ovisno o korištenoj Unix inačici), u sljedećem obliku:

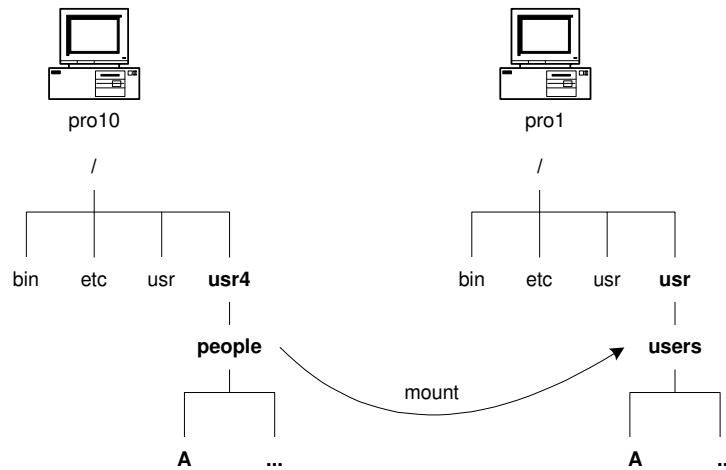
```
# Primjer za /etc/dfs/dfstab
#
share -F nfs -o ro /usr3/staff
# Dozvola pristupa imeniku /usr3/staff za sva računala (samo čitanje, -o ro,
"read only")
share -F nfs -o root=pro1:pro2 /usr4/people
# Dozvola pristupa imeniku /usr4/people za sva računala
# (dozvola pristupa administratoru samo s računala pro1 i pro2, -o
root=pro1:pro2)

# Primjer za /etc/exports
#
/usr/people
# Dozvola pristupa svim računalima
/usr2/fem -access=pro10
# Dozvola pristupa samo za računalo pro10
```

Povezivanje zapisničkih sustava ostvaruje se zadavanjem naredbe **mount** na računalu koje pristupa udaljenom zapisničkom sustavu, npr.:

```
pro1% mount -F nfs pro10:/usr4/people /usr/users
```

Prikazana naredba će pridružiti imenik **/usr4/people** (i sve njegove zapisnike i podimenike) sa računala **pro10** i uvrstiti u lokalni zapisnički sustav kao imenik **/usr/users**. Nakon toga, podaci posuđeni s računala **pro10** prividno postaju sastavni dio lokalnog računala **pro1** i pristupa im se na isti način kao i lokalnim podacima (Sl. 5-23).



Sl. 5-23 NFS – mrežni zapisnički sustav

Obrnuta naredba od **mount** je naredba **umount** kojom se prekida veza između udaljenih zapisničkih sustava.

Mrežni zapisnički sustav čini računalnu mrežu potpuno transparentnom, objedinjujući računala i njihove podatke u jedinstveni mrežni zapisnički sustav. Na taj je način moguće na jednome računalu pohraniti baze podataka ili instalacije velikih korisničkih programa te ih primjenom NFS protokola učiniti dostupnim svim računalima u mreži. To znači značajnu uštedu u diskovnome prostoru, jednostavnije korištenje, ažuriranje i održavanja računarskih podataka i programa.



## 5.8 Windows NT u mreži

NT operacijski sustav pruža usluge za korištenje različitih komunikacijskih protokola: NetBIOS/NetBEUI, TCP/IP, IPX/SPX i druge. Imanentan mu je NetBIOS/NetBEUI protokol koji podržavaju svi mrežni MS Windows operacijski sustavi. Putem njega se ostvaruje glavna podrška komunikaciji osobnih računala u lokalnoj mreži ("Network Neighbourhoods"). NT nudi mrežnu podršku za podjelu i pristup zapisničkim objektima i pisačima.

Mreža može biti organizirana na modelu radnih skupina ("Workgroups") ili domena.

**Radna skupina** je skup računala logički povezanih skupnim imenom. Ime radne skupine služi za identifikaciju računala i njihovih resursa u strukturiranoj organizaciji mreže. Pritom sva računala radne skupine sudjeluju kao ravnopravni učesnici mreže. Zato se takav model naziva "Peer-to-Peer"<sup>31</sup> mreža. Svako računalo u skupini može istovremeno imati ulogu poslužioca i korisnika.

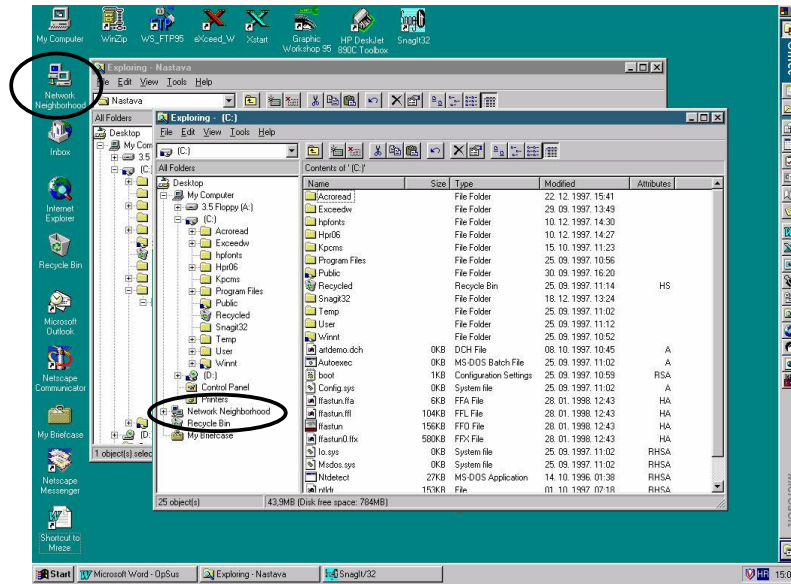
**Domena** podrazumijeva skupinu računala okupljenu oko NT poslužioca koji pruža potrebne mrežne usluge, kao što je autorizacija korisnika i kontrola pristupa umreženim objektima. Za razliku od radnih skupina, pristup određenoj domeni ovisi o dozvolama i uslugama kojima upravlja NT poslužilac. Takav korisnik/poslužilac model omogućava jedinstveno upravljanje i administraciju mrežnim resursima. Osigurava restriktivno strukturiranje mreže, odnosno pristup poddomenama definira se kroz odgovarajući sigurnosni sustav.

Računala u mreži dozvoljavaju pristup samo resursima koji su definirani kao "shared" (podijeljeni) objekti. Podijeljeni imenik ("folder") nekog računala pristupačan je zajedno sa svim svojim podimenicima. Pristup nadređenom imeniku nije moguć. Ako operacijski sustav koristi zapisnički sustav NTFS, tada je pristup ostvariv u skladu s definiranim dozvolama, odnosno pojedinih objekata.

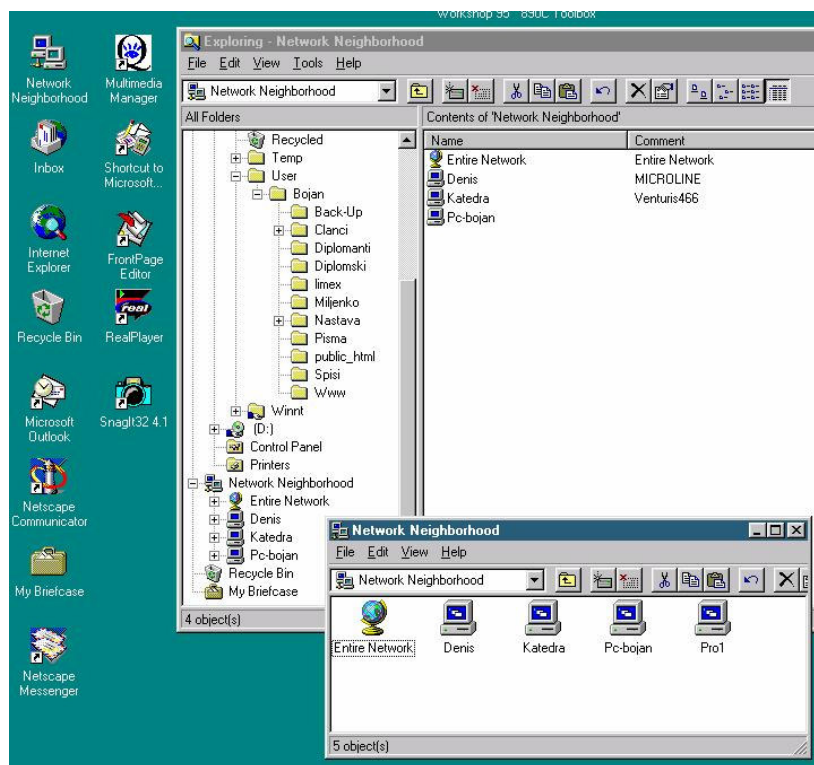
### 5.8.1 Pristup računalima u mreži

Pristup drugim računalima u lokalnoj mreži ostvaruje se aktiviranjem "Network Neighborhood" (mrežno susjedstvo) sučelja sa radnog stola ili kroz "Windows Explorer" (Sl. 5-24). Po aktiviranju "Network Neighborhood" sučelja računalo će emitirati brojna pitanja kroz mrežu u cilju utvrđivanja raspoloživih računala u lokalnoj mreži. Nakon prikupljenih odziva pojavit će se popis računala pripadajuće radne skupine ili domene (Sl. 5-25). Ostale radne skupine ili domene lokalne mreže dostupne su kroz objekt "Entire Network" (cijela mreža)/Microsoft Network (Sl. 5-26). Odabirom radne skupine i domene pojavljuju se raspoloživa računala (Sl. 5-27).

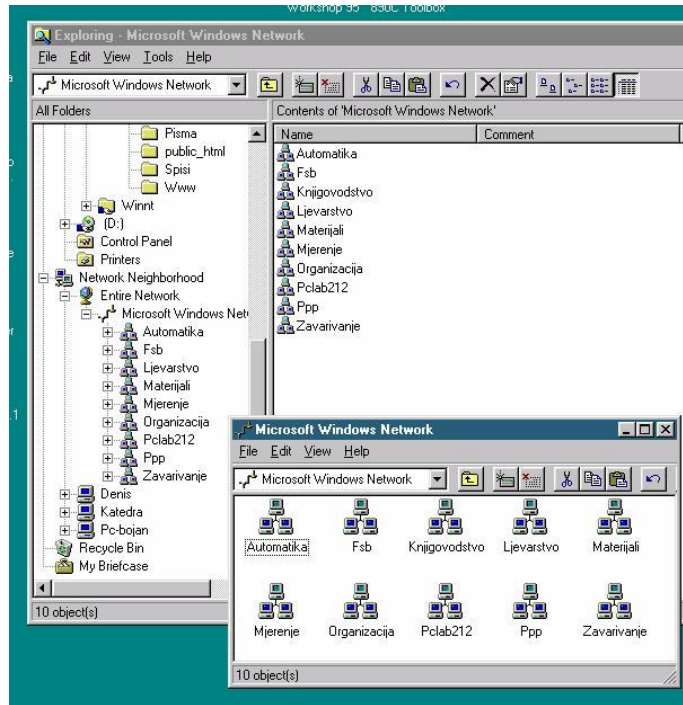
<sup>31</sup> Eng. "peer" znači ravan, jednak, jednakopravan.



Sl. 5-24 "Network Neighborhood"

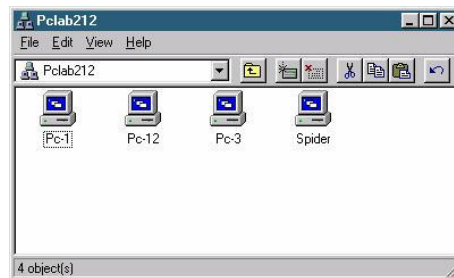


Sl. 5-25 Na prvj se razini ispisuje popis računala lokalne radne skupine

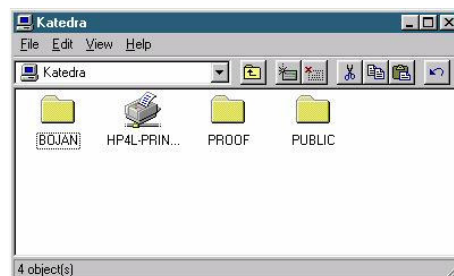


Sl. 5-26 "Entire Network" sadrži popis svih radnih skupina raspoloživih u lokalnoj mreži

Označavanjem željenog računala otvara se pristup njegovim podijeljenim objektima: imenicima, zapisnicima ili pisacima (Sl. 5-28).



Sl. 5-27 Označavanjem radne skupine dobiva se popis pripadajućih računala

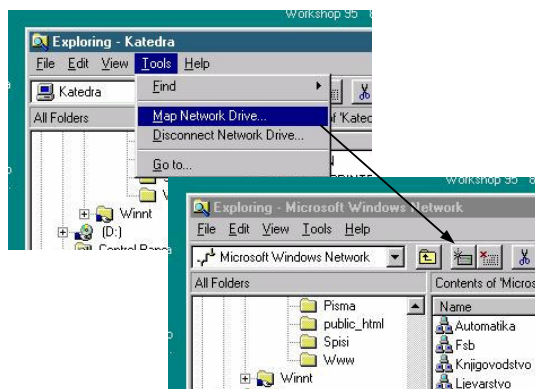


Sl. 5-28 Za označeno računalo pojavit će se u prozoru ponuđeni resursi

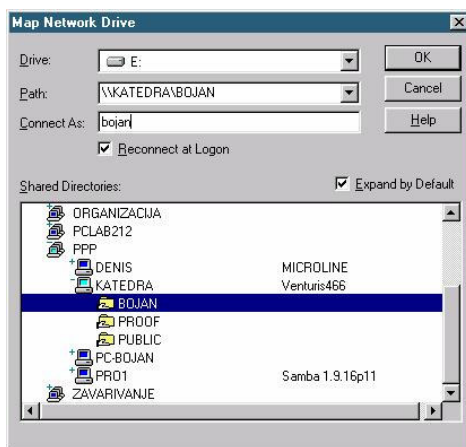
Pritom odabrano računalo, ili poslužilac domene, provjerava autorizaciju pristupa. To znači da korisnik mora imati svoje korisničko ime definirano na računalu kojem želi pristupiti ili na poslužiocu domene. U protivnom će pristup biti uskraćen. Korištenje raspoloživih mrežnih resursa temelji se na istim interaktivnim funkcijama operacijskog sustava kao i u slučaju lokalnih objekata.

### 5.8.2 "Map Network Drive"

Pristup računalima u mreži i njegovim objektima moguć je također primjenom naredbe "Map Network Drive". Naredba "Map Network Drive" omogućava pridruživanje mrežnog računala ili njegovih objekata u obliku logičke diskovne (zapisničke) jedinice koja se označava npr. "D:", "E:" itd. Tada pridružena oznaka postaje "root" imenik umreženog objekta. Naredba "Map Network Drive" je raspoloživa na izbornicima "File" ili "Tools" te na liniji alata "Windows Explorera". Ako se naredba aktivira sa izbornika "Tools" ili linije alata (Sl. 5-29), pojavit će se prozor prikazan na Sl. 5-30.



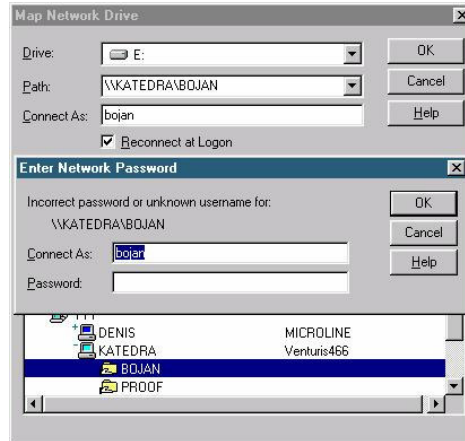
Sl. 5-29 Naredba "Map Network Drive" nalazi se u izborniku "Tools" i na liniji alata



Sl. 5-30 Prozor naredbe "Map Network Drive"

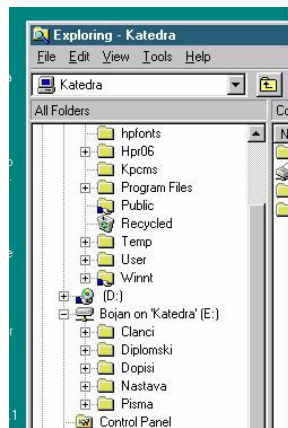


Pod rubrikom "Drive" definira se oznaka logičke jedinice, odnosno "root" imenika. U sljedećoj rubrici "Path" upisuje se, ili odabire iz donjeg prozora s popisom raspoloživih resursa, željeno računalo ili objekt. U rubriku "Connect As" treba upisati korisničko ime za koje postoji autorizacija na računalu kojem se pristupa. Označavanjem opcije "Reconnect at Logon" zahtijeva se automatska uspostava komunikacije nakon svakog ponovnog uključivanja računala. Nakon potvrde ("OK") računalo će zatražiti autorizaciju s lozinkom ("passwordom") otvorivši novi prozor kao što je prikazano na Sl. 5-31.



Sl. 5-31 Za prijavu na udaljeno računalo potrebno je upisom lozinke ("passworda") dokazati autorizaciju pristupa

Po uspostavi komunikacije na "Windows Exploreru" će se pojaviti odgovarajuća oznaka logičke jedinice (Sl. 5-32).



Sl. 5-32 Po uspostavi komunikacije s udaljenim računalom raspoloživi resursi bit će pridruženi zadanoj zapisničkoj logičkoj jedinici (npr. "E:")

Ako se u "Windows Exploreru" označi neki mrežni zapisnik ili imenik na izborniku "File" pojavit će se naredba "Map Network Drive". Svrha naredbe je ista kao i u

prethodnom slučaju, stom razlikom da će se po aktiviranju naredbe "File"/"Map Network Drive" pojaviti prozor kao na Sl. 5-33. S obzirom da je već označen objekt koji se želi umrežiti na prozoru su raspoložive jedino rubrike "Drive" i "Connect As".

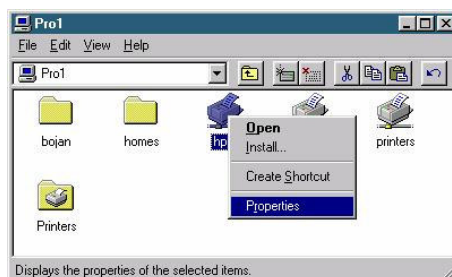


Sl. 5-33 Prozor naredbe "File/Map Network Drive"

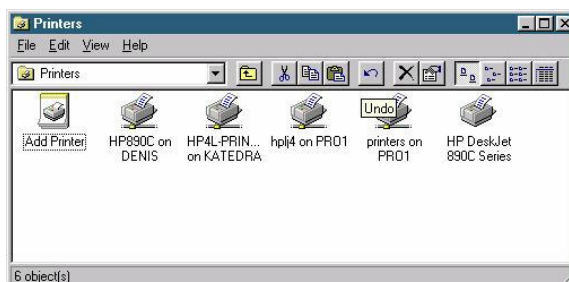
Isključivanje veze uspostavljene s "Map Network Drive" izvodi se naredbom "Disconnect Network Drive".

### 5.8.3 Pristup mrežnim pisačima

Pristup mrežnim pisačima provodi se na sličan način kao i zapisničkim objektima. Da bi se mrežni pisač mogao ispravno koristiti potrebno je instalirati odgovarajući upravljački program. Ako se mrežni pisač označi s desnom tipkom miša pojavit će se izbornik s naredbom "Install" (Sl. 5-34). Aktiviranjem naredbe "Install" pokrenut će se postupak za učitavanje upravljačkog programa, sa udaljenog računala ili lokalne distribucije operacijskog sustava (npr. NT CD-a), i potom će odabrani pisač bit pridružen u lokalni popis pisača (Sl. 5-35). Daljnja primjena pisača neće se razlikovati od upotrebe onog lokalno spojenog na računalo.



Sl. 5-34 Pomoćni izbornik na ikoni pisača

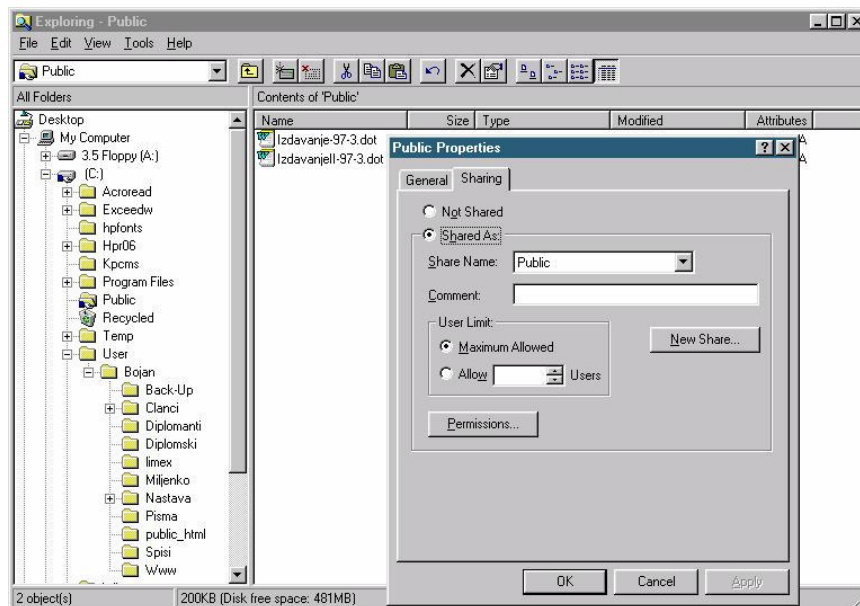


Sl. 5-35 Prozor "Printers", ikone sa provodnikom ispod simbola pisača označavaju mrežne pisače

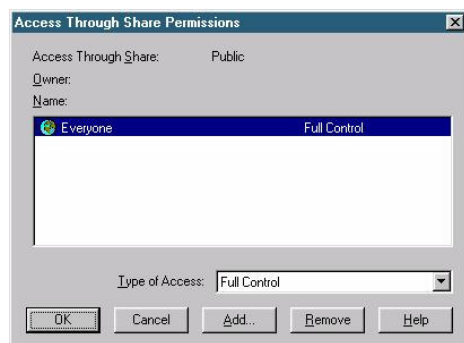
#### 5.8.4 Podjela objekata

Dozvole za pristup udaljenih računala (korisnika) pojedinim objektima (imenicima, zapisnicima i pisačima) zadaju se putem opcije "Share". Za označeni objekt u "Windows Exploreru" potrebno je aktivirati naredbu "File/Properties". Prozor "Properties" (Sl. 5-36) sadrži mogućnost podjele ("Sharing") objekta u lokalnoj mreži. Izborom opcije "Shared As" objekt postaje uvjetno raspoloživ ostalim računalima. U rubriku "Share Name" upisuje se ime objekta sa kojim će se on predstaviti u mreži. U rubriku "Comment" može se pobliže opisati sadržaj objekta.

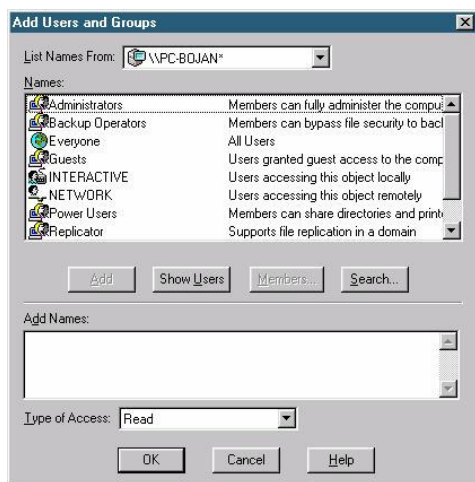
Dozvole pristupa definiraju se aktiviranjem gumba "Permissions". U prozoru koji će se potom otvoriti (Sl. 5-37) naveden je popis korisnika i/ili skupine korisnika koji mogu pristupati objektu. Za označenog pristupnika u rubrici "Type of Access" pojavit će se lista dozvola. Ukoliko želimo promijeniti dozvole koristit ćemo naredbu "Remove" za brisanje pristupnika sa liste ili naredbu "Add" za dodavanje novih. Po aktiviranju gumba "Add" pojavit će se novi prozor "Add Users and Groups" (Sl. 5-38). U njemu se mogu odabrati novi korisnici ili skupine korisnika i njima pripadajuća dozvola pristupa.



Sl. 5-36 Prozor "Properties"



Sl. 5-37 Popis korisnika i/ili skupine korisnika za koje su definirane dozvole pristupa



Sl. 5-38 Prozor za odabir korisnika i/ili skupine korisnika i njima odgovarajuće dozvole pristupa

---

## 6 RAZVOJNE TEŽNJE

---

**R**azvoj računarstva podjednako ovisi o razvoju računalnog sklopovlja, kao i o razvoju novih programskih (numeričkih) postupaka. Iako se mnogi razvojni pravci oslanjaju na postojeće tehnološke temelje (Turingov stroj, binarnu logiku i aritmetiku, Von Neumanovo računalo), može se u budućnosti očekivati pojava zasada nepoznatih znanstvenih otkrića, zasnovanih na potpuno novim principima i tehnologijama, koji nisu npr. binarni ili slijedni. Stoga je teško predvidjeti dalju informatičku budućnost, no bez obzira na značajke budućih tehnologija, informacijska globalizacija i sveobuhvatna podrška intelektualnim djelatnostima bit će temeljne odrednice razvoja.

Današnje stanje razvoja računarstva karakteriziraju sljedeće tehnologije, pristupi i znanstveni trendovi:

- ❑ mikroprocesorska tehnologija visokog stupnja integracije
- ❑ paralelno procesiranje i super-računala
- ❑ računalne mreže, "World Wide Web", Java
- ❑ vizualno računarstvo, grafičko korisničko sučelje, simulacija, prividna stvarnost
- ❑ normizacija
- ❑ umjetna inteligencija

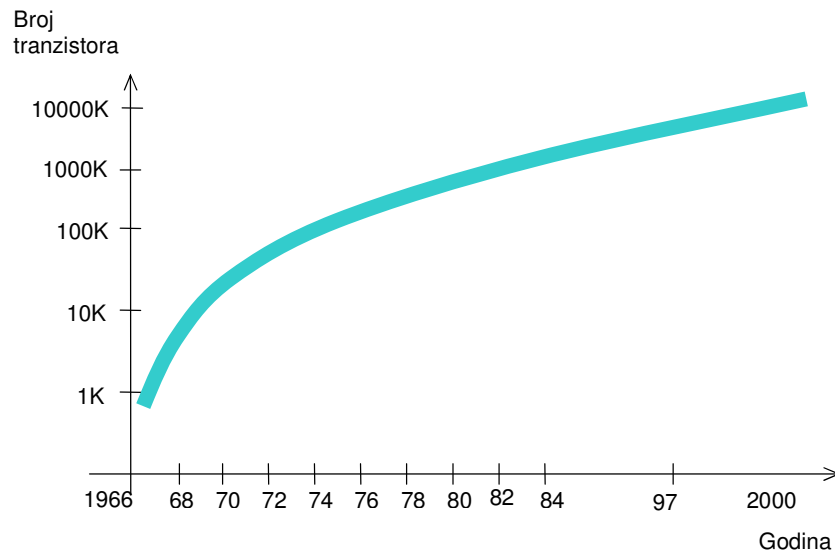


## 6.1 Mikroprocesori budućnosti

Suvremeni dosezi u procesorskoj tehnologiji omogućavaju obradu više stotina milijuna matematičkih operacija u sekundi, integrirajući više desetaka milijuna tranzistora na prostoru od svega dva do tri kvadratna centimetra (Sl. 6-1). Temelje se na superskalarnoj RISC<sup>32</sup> arhitekturi koja omogućava izvođenje više operacija istovremeno. Očekuje se daljnje povećanje stupnja integracije tranzistora, ali i ubrzavanje procesora povećavanjem frekvencije radne periode preko 1 GHz-a (Sl. 6-2).



Sl. 6-1 Intel Pentium 4, 1.4 GHz



Sl. 6-2 Trend povećanja broja tranzistora

<sup>32</sup> Eng. "RISC - Reduced Instruction Set Computer", računalo sa smanjenim brojem naredaba.

Snaga mikroprocesora, koja se mjeri brzinom obrade podataka, udvostručava se otprilike svakih 18 mjeseci. Slična je situacija i sa stupnjem integracije; broj tranzistora na poluvodičkim komponentama (procesoru i memoriji) udvostručava se svake druge godine (Tab. 6-1).

Tab. 6-1 Pregled razvoja procesora u pogledu brzine i integracije

Mikroprocesor	Početa/sadašnja cijena	Početa/sadašnja brzina	Broj tranzistora	Frekvenc.	Širina ALU sabirnice	Tehnologija
	[USA \$]	[MIPS]		[Hz]	[bit]	[ $\mu\text{m}$ ]
8086 (1978)	360	0.33/0.75	29000	2	8	3
80286 (1982)	360/8	1.2/2.66	134000	6	16	1.5
80386 (1985)	299/91	5.0/11.4	275000	16	32	1.5
i486 (1989)	950/317	20.0/54.0	$1.6 \cdot 10^6$	25	32	1
Pentium (1993)	900	1120	$3.1 \cdot 10^6$	60	32/bus 64	0.8
Pentium II (1997)	780	400	$7.5 \cdot 10^6$	233	32/64	0.35
MIPS R10000 (1997)			$6.7 \cdot 10^6$	250	64	0.35
Alpha 21264 (1998)			$15.5 \cdot 10^6$	500	64	0.35
Pentium III (1999)		1000	$9.5 \cdot 10^6$	450	32/64	0.25

Prema SIA ("Semiconductor Industry Association"), do 2012. godine može se očekivati integracija od 180 miliona tranzistora po  $\text{cm}^2$ , primjenjujući minijaturizaciju procesorskih značajki do  $0.035 \mu\text{m}$  napajanja 0.5 volti.



Ista će tehnologija omogućiti stvaranje vrlo velikih memorijskih kapaciteta, do 64 GB po "chipu". Složenost računala sve više raste, a istim trendom rastu i troškovi razvoja. Većina računalnih primjena u svijetu otpada na osobna računala, čiji ubrzani razvoj omogućava preuzimanje sve raznovrsnijih zadataka u svakodnevnom životu i radu. Veliko tržište osobnih računala osigurava velike zarade, pa tako i veliku mogućnost ulaganja u razvoj, ali i kraće vrijeme amortizacije razvoja. Na taj način cijene osobnih računala postaju sve niže, a računalne mogućnosti sve veće. S druge strane proizvođači radnih stanica i višeprocorskih poslužioaca ne mogu računati s jednakim povećanjem tržišta, dapače značajne dijelove tržišta otimaju im proizvođači osobnih računala. Stoga cijene snažnih višeprocorskih računalnih sustava, namijenjenih zahtjevnim inženjerskim i znanstvenim zadacima, ostaju i dalje vrlo visoke. Iako se čini da su se polarizirala dva računalna svijeta: osobna računala (temeljena na Intel procesorima i Microsoft programskoj podršci) i UNIX radne stanice i poslužioaci, razvoj računarstva u bliskoj budućnosti neće ovisiti o prevladavanju ili gašenju nekog od spomenutih računalnih pristupa. Taj je privid stvoren kao rezultat marketinške borbe za prevlast. Budućnost valja tražiti u razvoju potpuno novih računalnih koncepata, kao što je mrežno računarstvo, paralelizacija i konačno razvoj potpuno novih tehnologija (npr. optički procesori, bioprocorsori ili kvantni procesori).



---

## 6.2 Paralelizacija i super-računala

Super-računala i paralelna obrada podataka postaju informatička svakodnevnica. Čak i osobna računala opremaju se s dva i više procesora. Suvremeni operacijski sustavi za simetričnu višeprosorsku obradu informacija omogućavaju gotovo linearno ubrzavanje računala s povećanjem broja procesora, obavljajući više desetaka milijardi računskih operacija u sekundi. Pritom brzina pojedinih procesora ne dolazi toliko do izražaja, snaga se temelji na učinkovitosti operacijskog sustava i programa za paralelizaciju i distribuciju procesa.



Sl. 6-3 Porodica skalabilnih višeprosorskih računala



---

### 6.3 Simulacija i vizualno računarstvo

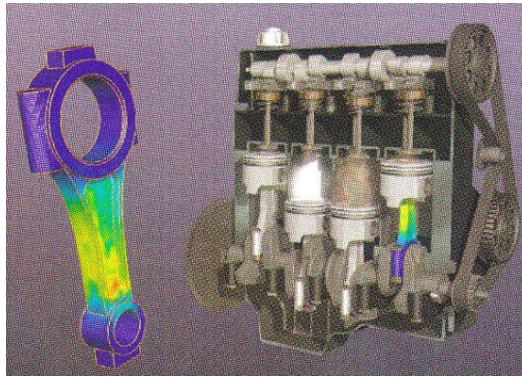
Stroga određenost i jednoznačnost svake računalne naredbe oduvijek se sukobljavala s nesavršenom i vizualnom ljudskom prirodom. To je uvijek bio izvor brojnih pogrešaka u interakciji s računalom. Stoga je razvoj računalne grafike, grafičkog korisničkog sučelja i "vizualnog računarstva" bio prirodan put ka ljudskom vizualnom asocijativnom načinu razmišljanja. Vizualno računarstvo podrazumijeva obradu slika, odnosno grafičkih informacija. Slike se na računalu interpretiraju nizom točaka. Svaka točka na zaslonu računala pohranjena je kao broj. Ovisno o veličini, rezoluciji i broju boja, jedna slika može sadržavati milijun točaka. Ako pritom uzmemo u obzir da je sadržaj slike najčešće proizvod zahtjevnih numeričkih proračuna, onda je jasno da obrada grafičkih informacija zahtijeva izrazito brza računala koja mogu u realnom vremenu obraditi veliku količinu podataka. Tek su mikroprocesori u '80-tim godinama omogućili razvoj vizualnog računarstva.

Danas, umjesto da se mučimo sa strogom određenošću računalnih naredbi, upisujući putem tipkovnice slovo po slovo pojedinih naredbi, suvremeni operacijski sustav opremljen je grafičkim sučeljem koje putem intuitivnih simbola u interakciji pomoću računalnog miša omogućava korisniku jednostavnu komunikaciju s računalom i njegovim funkcijama.

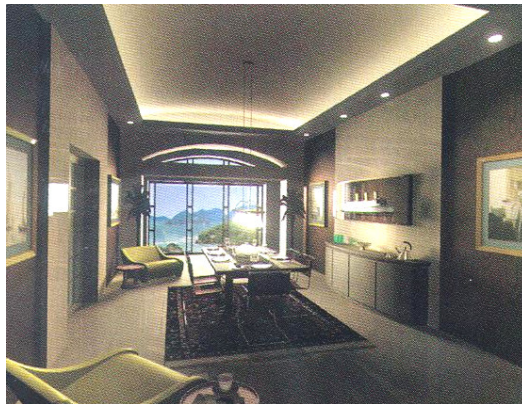
Jednako tako većina uslužnih programa i općenito računarskih pristupa temelji interakciju s korisnikom na vizualnoj interpretaciji i simulaciji realnog svijeta. Nekada su računala pomagala inženjerima u izradi tehničkih crteža putem kojih su stručnjaci razmjenjivali informacije. Danas, inženjeri na računalu oblikuju modele svojih proizvoda simulirajući njegova geometrijska, fizikalna i funkcionalna svojstva (eng. "CAE - Computer Aided Engineering" - računalom podržano inženjerstvo). U mogućnosti su doživjeti i istražiti proizvod u eksploataciji prije njegove materijalizacije (Sl. 6-4, Sl. 6-6). U nastojanju da računalo postane sastavni dio korisničkog radnog okruženja razvijeno je potpuno novo znanstveno područje koje se bavi simulacijom stvarnosti (eng. "VR - Virtual Reality" - prividna stvarnost). Pomoću posebnih uređaja za simulaciju percepcije (digitalne naočale, rukavice, CAVE<sup>33</sup> - VR sobe) korisnik "uranja" u simulirane svjetove s kojima komunicira na sličan način kao i u stvarnosti (Sl. 6-7). Danas se "VR" pristup svakodnevno koristi za verifikaciju preglednosti i upravljivosti u automobilske i avionske industriji, zatim za uvježbavanje pilota, vozača, kirurga i općenito u svim djelatnostima gdje se simulacijom izbjegavaju opasnosti i visoki troškovi u stvarnom svijetu.

---

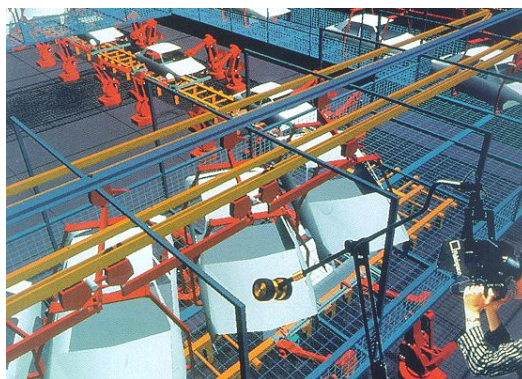
<sup>33</sup> CAVE - eng. "Cave Automatic Virtual Environment".



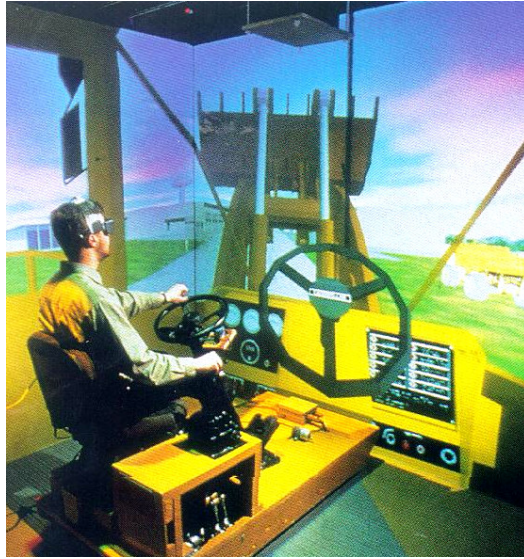
Sl. 6-4 Vizualizacija rezultata analize naprežanja komponenata motora primjenom metode konačnih elemenata i CAE (Computer Aided Engineering) sustava koji omogućavaju oblikovanje i simulaciju umotvorina (Lana Rushing, Engineering Automation, Inc.)



Sl. 6-5 Fotografski vjerno oblikovanje i vizualizacija arhitektonskih objekata (C. Mosher, R. Burdock, Peripheral Vision Animations)



Sl. 6-6 Pomoću sustava prividne stvarnosti moguće je "prošetati" tvornicom već u fazi projektiranja



Sl. 6-7 Računalni sustavi za simulaciju prividne stvarnosti omogućavaju ispitivanje funkcionalnost budućih proizvoda ili uvježbavanje njihovih korisnika (Caterpillar)

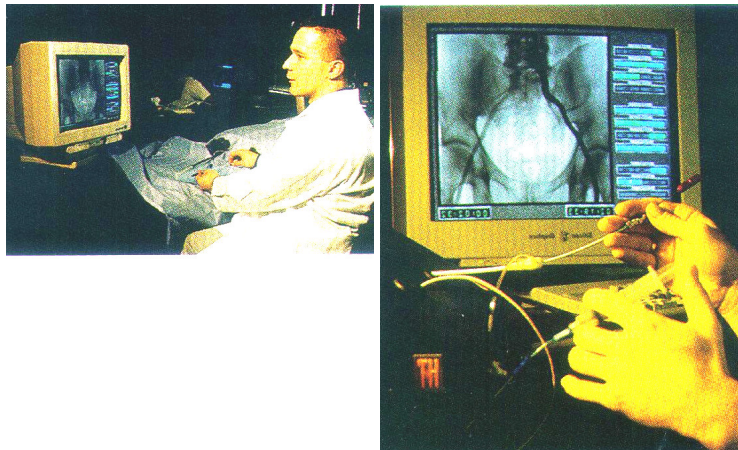


## 6.4 Računalne mreže

Računalne mreže revolucionalizirale su korištenje računala i ponajviše utjecale na razvoj informatičkog društva. Već danas je mnogim korisnicima važnija brzina računalne mreže nego procesorska moć računala. Učinkovita mreža otvara brojne mogućnosti razmjene informacija i općenito komunikacije. Fizička raspoloživost opreme i ljudi postaje potpuno nevažna. Sa priključkom na mrežu svaki korisnik može doseći bilo koje umreženo računalo u svijetu i ako je autoriziran za njegovo korištenje može se poslužiti jednako podacima, programima kao i procesorskim kapacitetima udaljenog računala. Više nije presudno da ljudi sjede jedan pored drugoga da bi surađivali. Putem računalnih mreža danas možemo istovremeno prenositi multimedijalne i računarske informacije na sve strane svijeta. Složenom operacijom u Meksiku može upravljati poznati kirurg iz Švicarske, sjedeći za računalom u svojem domu i prateći video sliku iz operacione dvorane (Sl. 6-8). Na razvoju složenih proizvoda mogu surađivati razvojne skupine širom svijeta, razmjenjujući izravno simulacije svojih ideja.

Multimedijalnom integracijom računalo postaje univerzalni komunikacijski uređaj, postupno zamjenjujući konvencionalne tehnologije, kao što su telefon, fax, televizor, pismo, izravno utječući na promjene u svakodnevnom načinu života. Odlazak u kupovinu ili banku danas je moguće obaviti pomoću računalne mreže jednako učinkovito.

Iako su u početku osobna računala imala cilj jednostavnost višenamjenske primjene, danas su postala vlastita suprotnost - previše su složena i za određene namjene preskupa. Mreže nam omogućavaju primjenu jednostavnijih i jeftinijih računala, o čijim uslužnim programima brinu jaka mrežna računala koja održavaju kvalificirane osobe na jednome mjestu. Tako se izbjegavaju brojni korisnički problemi oko instalacije i usklađivanja uslužnih programa, kao i bespotrebni višestruki utrošak zasebnih kapaciteta (svaki korisnik ima vlastitu instalaciju potrebnih programa).



Sl. 6-8 Primjenom računalnih mreža i simulacijom moguće je obavljati dijagnostiku i izvoditi udaljene operacije

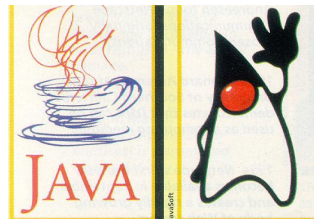




## 6.5 Normizacija

Informacijska tehnologija, kao globalna komunikacijska i integracijska tehnologija, nužno nalaže usklađivanje između različitih komponenata i njihovih proizvođača. To znači normiranje jednako programskih kao i sklopovskih sučelja. Posljedice uvođenja svjetskih normi imaju daleko šire značenje od osiguranja puke razmjene podataka. Temeljni pojam koji višeznačno označava doprinos suvremene informacijske normizacije je *otvorenost* s posljedicom smanjenja redundancije i entropije.

Nekada su računala predstavljala tehničke otoke, primjenjujući specifična programska i sklopovska rješenja. Svako novo računalo bilo je u velikoj mjeri inkompatibilno s postojećom opremom, iako je možda poticalo od istog proizvođača. Programi pisani za jedno računalo morali su se iznova programirati za druga računala. Isti algoritmi varirali bi se na nebrojeno načina, a posljedica su bili tzv. "špageti kodovi" - nepregledni, predugački i neučinkoviti programi. *Redundancija* se ponajviše očitovala u potrebi da se računalni sustavi i njihova programska podrška u cjelini umnožavaju, bez mogućnosti postupne nadgradnje. Nemogućnost nadgradnje postojećih informacijskih sustava ima za posljedicu kratkotrajna i nedovoljno razvijena rješenja, a to znači brzo zastarijevanje znanja i nepovratni razvojni gubitak, odnosno povećani stupanj razvojne *entropije*.



Sl. 6-9 JAVA

Danas, primjenom odgovarajućih normi različiti proizvođači računala i računalnih komponenata osiguravaju međusobno korištenje sklopovlja bez ikakve prilagodbe. Npr. standardna PCI sabirnica povezuje sve računalne komponente, od diskova do grafičkih kartica, na temelju točno propisanih normi, od priključka do električkih signala. Ugrađena je danas u gotovo sva računala. Zahvaljujući tome moguće je u računala potpuno različitih sklopovskih arhitektura i proizvođača ugraditi potpuno iste komponente. Ista grafička kartica ili magnetski disk jednako će raditi na osobnom računalu kao i na skupoj radnoj stanici. Normizacijom na razini objektnih koncepata u domeni operacijskih sustava, uslužnih i korisničkih programa, značajno je smanjenja potreba reprogramiranja korištenjem standardnih programskih funkcija. Suvremeni JAVA koncept ide još jedan korak dalje. Implementira računalni program kao mrežni program koji se može izvoditi na bilo kojem računalu, bez potrebe prethodnog prevođenja u strojni jezik. Interpretera ga poseban program - preglednik, ili jednostavno JAVA operacijski sustav. Danas korisnik više ne mora uvijek iznova učiti koristiti računalo, jer normizacija grafičkih sučelja osigurava jedinstveni pristup operacijskom sustavu i uslužnim programima, neovisno o vrsti i proizvođaču računala. Nakon mnoštva operacijskih sustava (VMS, DOS, UNIX, ...) tek su dva operacijska

sustava podijelila računalni svijet: UNIX, koji se poglavito koristi na radnim stanicama, i Windows 95/NT, neizbježna operacijsko okruženje osobnih računala.

Pokretač koncepcije otvorenih sustava je američka udruga IEEE (eng. "Institute of Electrical and Electronic Engineers" - Institut inženjera elektrike i elektronike). Otvoreni sustavi su oni koji osiguravaju svoju funkcionalnost u skladu s javno prihvaćenim specifikacijama. To ukratko znači osiguravanje sljedećih mogućnosti:

- prenošenje korisničkih programa između različitih računala uz neznatne promjene;
- sposobnost međusobnog razumijevanja informacija bez obzira da li se odvijaju na istom ili različitim računalima;
- jednoobrazan pristup operacijskom sustavu i korisničkim programima putem standardnog korisničkog sučelja.

S obzirom na složenost računalne tehnologije normizacijske djelatnosti su razgranate u okvirima različitih svjetskih institucija, pokrivajući pojedine specifična problemska područja. Često se koristi podjela skraćeno nazvana "MUSIC" po početnim slovima naziva sljedećih područja:

- **Management** - upravljanje računalnim sustavom i mrežom; obuhvaća normizaciju uslužnih programa za administriranje, a na tome rade udruge ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), IEEE (POSIX 1003.7) i OSF (Open System Foundation).
- **User Interface** - korisničko sučelje; normizacijom korisničkog sučelja propisuje se zajednički oblik radnog okruženja za operacijske sustave i uslužne programe, osiguravajući jedinstveni način korištenja računala; X Consortium se brine o razvoju X Windows sučelja UNIX operacijskih sustava, OSF definira izgled grafičkog radnog okružja (OSF/Motif, CDE - Common Desktop Environment).
- **Service Interface for Programs**- programsko sučelje; obuhvaća normizaciju programskih jezika (FORTRAN - ISO 1359, PASCAL - ISO 7185, C - ISO 9899, ...), grafičkih funkcija i knjižnica (GKS, PHIGS, PEX, OGL), kao i sustavnosnih funkcija operacijskog sustava (IEEE - POSIX 1003).
- **Information and Data Formats** - formati informacija i podataka; ISO propisuje format podataka kroz jezike za pristup bazama podataka, kao što je SQL (Structured Query Language - ISO SQL 9075:19898).
- **Comunication Interfaces** - komunikacijska sučelja; komunikacijski protokoli definirani su ISO OSI (Open Systems Interconnection) referentnim modelom, propisujući na sedam razina način spajanja računala u mrežu pa sve do protokola razmjene podataka i načina njihovog prikaza.



## 6.6 Umjetna inteligencija

Umjetna inteligencija je znanstveno područje koje se bavi razvojem računarskih tehnika za rješavanje zadataka povezanih s ljudskom inteligencijom. Inteligencija je sposobnost stjecanja i korištenja znanja u cilju indukcije novog. Očituje se u moći komunikacije, učenja i zaključivanja.



Uz inteligenciju se vezuju pojmovi kao što su kreativnost, intuicija, znanje, razmišljanje ..., dakle sve ono što razumijeva kvalitetu daleko dublju od obične obrade informacija i one koju bi mogli prepoznati u strojevima ili bilo kojim drugim tvorevinama. Budući su pri tome saznanja o mentalnim procesima danas još uvijek skromna, postavljaju se mnoga pitanja u kontekstu razvoja umjetne inteligencije :

- može li računalo misliti ?
- nije li računalo stroj koji radi samo s nulama i jedinicama ?
- ako računalo može misliti, onda je inteligencija nešto mehaničko pa čovjek nije ništa bolji od stroja ?
- neće li stoga strojevi preuzeti čovjeku primat ?
- ... itd.

Uz sva ta i ostala pitanja te skepticizam, umjetna inteligencija pokušava dati odgovor bar na neka, razvijajući teoriju i računalu prilagođene modele procesa mišljenja, učenja i percepcije. U svojem nastojanju umjetna inteligencija teži oponašanju prirode (ljudskog uma), pa se često dovodi u vezu s umjetnošću. Većina se autora slaže da je izazov ove discipline upravo u tome što objedinjuje čari i znanosti i umjetnosti kroz matematiku, heuristiku i programsku tehniku.

Umjetna inteligencija se nalazi tek u začetku svoga razvoja, te se još uvijek ne može očekivati realizacija uistinu inteligentnih strojevi, već ju treba razumjeti kao izvjesni alternativni instrumentarij u razvoju sustava za podršku intelektualnim aktivnostima. Osnovne pravce razvoja čine ekspertni sustavi, neuronske mreže, fuzzy logika i genetski algoritmi. Primjena je raznolika:

- ekspertni sustavi: dijagnostika (u medicini i industriji), prirodni govor, prevođenje, inženjering ...,
- neuronske mreže: vizualni sustavi (umjetni vid u robotici, biomedicini, analizi materijala ...),
- fuzzy logika: matematika, upravljanje procesima ...,
- genetski algoritmi: konstruiranje/oblikovanje ...

Svi ovi pristupi temelje se na tezi da inteligencija počiva na znanju i mehanizmu njegove obrade, ali se razlikuju u koncepciji njihovog oblikovanja.

Fundamentalna pitanja kojima se dakle bavi AI jesu tehnike formalizacije znanja i metodologije zaključivanja i učenja.



Za razliku od konvencionalnih pristupa, ovdje se podrazumijeva da znanje mora biti eksplicitno i simbolički definirano. Korelacije između elemenata toga znanja i njihovi

međusobni utjecaji tvore odgovarajuću strukturu na koju se nadovezuje mehanizam zaključivanja. Mehanizam zaključivanja je problemski orijentirano znanje o korištenju znanja, pa se jednostavno može tretirati kao znanje o znanju. Ono je generičko znanje koje služi stvaranju novoga znanja, odnosno sastavni je dio procesa učenja.

Danas također postoje tendencije kombiniranja različitih AI metoda u rješavanju kompleksnih problema i teško je predvidjeti koji će pristupi u budućnosti prevladati. Svakim se danom otvaraju nova područja istraživanja i novi problemi. U želji da stvara sve savršenije tvorevine, čovjek se suočava sa sve složenijim sustavima i procesima kojima ne može vladati konvencionalnim pristupima. Stoga je jedino izvjesno da razvoj umjetne inteligencije nije tek puki znanstveni trend, već će o njemu ovisiti razvoj mnogih disciplina, poglavito tehničkih. Uostalom, kao što zaključuje Tanimoto u svojoj knjizi "The Elements of Artificial Intelligence": *"Ništa nije intelektualnije od stvaranja drugog intelekta"*.

Doprinos umjetne inteligencije treba promatrati i u svjetlu razotkrivanja specifičnih stručnih znanja i metodologija, inače zatumljenih u umovima pojedinaca, koja se formalizirana mogu efikasnije usvajati, primjenjivati i unapređivati.

Najveći doprinosi u razvoju inteligentnih tvorevina postignuti su u domeni upravljanja strojevima, industrijskim vozilima, računalnoj potpori inženjerskim djelatnostima, prepoznavanju vizualnih i zvučnih podataka, medicinskoj dijagnostici i mrežnom računarstvu. Suvremeni roboti sposobni su prepoznavati promjene u radnom okolišu, učiti na iskustvu i kreativno se prilagođavati novonastalim zahtjevima. Inteligentne tvorevine sposobne samostalno djelovati nazivamo inteligentni agent. Mogu biti jednako strojevi kao i računarski programi. Primjer inteligentnih programskih agenata su mrežni agenti koji prate rad svojega korisnika i putuju internetom u potrazi za informacijama koje bi mogle interesirati njegovoga vlasnika. Na taj način mrežni agent preuzima na sebe odgovornost za selekciju nepreglednog mnoštva podataka koji niču na Internetu.

Inteligentni sustavi odigrat će u budućnosti značajnu ulogu kao podrška ljudskom kreativnom radu. Posebno kao nadgradnja poglavito egzaktnih znanosti i tehnologija, jer naša se inteligencija dobrim dijelom oslanja upravo na nesavršenost ljudske prirode, sposobne stvarati nebrojene asocijacije i raznolika rješenja do kojih je nemoguće doprijeti uvijek ponovljivim algoritmima.



---

# 7 PRIMJENA

---

---

## 7.1 MS Word

Microsoft Word® zasigurno je najrašireniji program za obradu teksta (text editor) na svijetu. Svakodnevno ga koriste milijuni ljudi, od učenika u osnovnim školama, studenata na fakultetima, sveučilišnih profesora, liječnika, pa sve do pojedinaca koji imaju potrebu zapisati kakav tekst i pohraniti ga u računalo ili ispisati na papir. Zato je nužno da svatko od nas poznaje barem osnove ovog programa, kako bi mogao uspješno komunicirati u eri računalne tehnologije. S osnovama, dalje je moguće samostalno istraživati unutar programa i unapređivati svoje vještine u kreiranju dokumenata.

Može se reći da je Word zamijenio klasičnu pisaću mašinu, tako da se više ne može vidjeti novinara ili pisca kako strpljivo piše na mašini redak po redak, ispravlja greške korektorom, umeće novi list papira i slično.

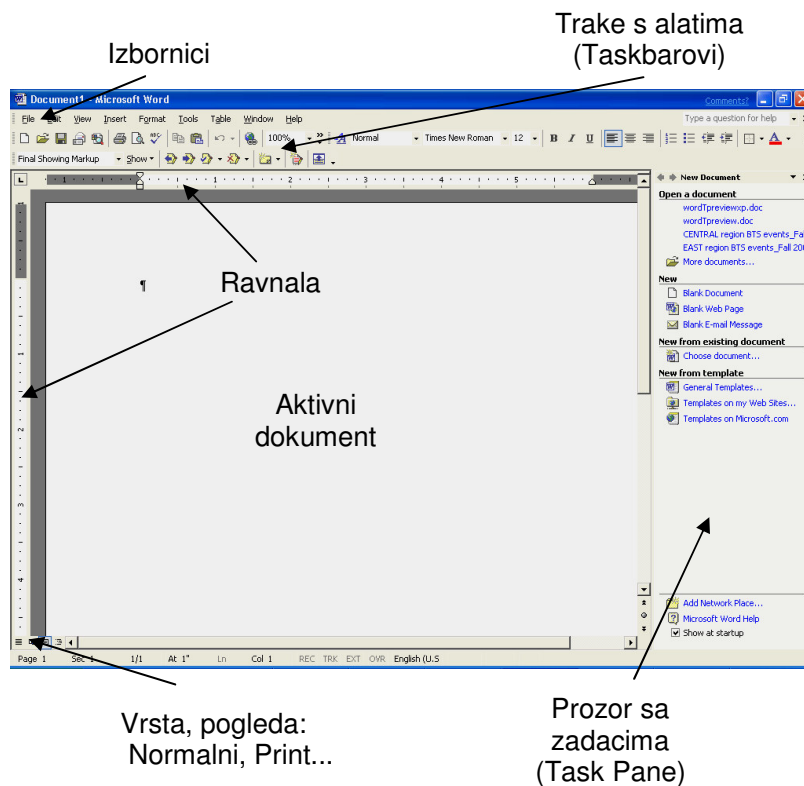
Iako je Word zamijenio pisaću mašinu u užem smislu, njegove prednosti ne svode se samo na to da je primjerice jednostavnije ispraviti neko slovo koje je krivo napisano ili da na kraju svakog ispisanog retka teksta ne treba rukom vratiti valjak kako bi se prešlo u novi redak. Word omogućuje komunikaciju s računalnim komponentama, tvrdim diskom, pisačem, disketnom jedinicom, DVD-om. Omogućuje nadalje jednostavno i brzo pretraživanje teksta po nekom parametru npr. ključnoj riječi, kreiranje uzoraka (templatea), povezivanje s drugim programskim aplikacijama, daje gramatičke savjete i još mnogo toga. U svakom slučaju, ubrzao je i olakšao pisanje teksta u odnosu na pisanje pisaćom mašinom, a i sačuvao mnoga stabla. Nažalost, izgubio se onaj šarmantni zvuk koji je to pisanje na mašini pratio.

Obično je slučaj takav da s mogućnostima koje neka aplikacija nudi, rastu i opterećenja na korisnika, koji treba naučiti raditi sa željenom aplikacijom. Tako je u nekom smislu i Word za uporabu kompliciraniji od pisaće mašine. Ipak se za Word može reći da je jedna od najjednostavnijih aplikacija za učenje i svatko od nas već nakon par minuta provedenih u njegovom okruženju, u stanju se koristiti s osnovnim funkcijama. Za neke naprednije funkcije potrebno uložiti malo truda, proučiti help ili pročitati neku knjigu, ali kasnije se ima zadovoljstvo što se jednostavno i brzo može napisati i urediti neki tekst kako se želimo.

### 7.1.1 Prije početka

Za korištenje Worda ne zahtijevaju se neka posebna računalna predznanja, on je u svom osnovnom dijelu namijenjen najširem krugu korisnika, a možda je i jedan od najčešće korištenih programa u svijetu. Ipak, potrebno je nešto što bi svatko od nas trebao znati prije nego što krene raditi s Wordom. Prije svega treba znati pokretati programe pod Windows operativnim sustavom, kao i spremiti podatke na neko mjesto u računalu i biti ih u stanju opet pozvati. Na slici 1.1 prikazano je sučelje Worda s pripadajućim aktivnim dokumentom i objašnjeni su osnovni elementi tog sučelja.





Sl. 7-1 Osnovno sučelje

Na slici prikazani su osnovni elementi sučelja programa Word. Središnji dio zauzima prostor aktivnog dokumenta, to je "papir", po kojem se piše. Izbornici pomažu da dosegemo određenu naredbu, trake s alatima predstavljaju direktan pristup nekoj naredbi, dok iz prozora sa zadacima možemo odabrati određenu akciju s dokumentom, npr. pohraniti ga, otvoriti novi, zadnji na kojem smo radili i slično.

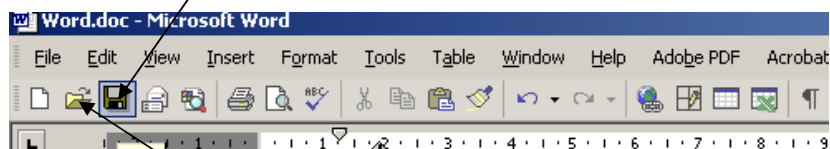
#### 7.1.1.1 Pokretanje MS Worda

- ❑ Na Start izborniku, odabrati Programs, a zatim odabrati Microsoft Word. Otvoriti će se novi dokument u Print pogledu.
- ❑ Kliknuti na Blank Document. Otvara se novi, prazni dokument. (ekvivalentno je ovome da odmah nakon startanja Worda počinjemo raditi s dokumentom koji se po defaultu otvara pri svakom novom pokretanju programa).
- ❑ Kliknuti bilo gdje na prazan dokument i počnite unositi tekst putem tipkovnice.

#### 7.1.1.2 Spremanje dokumenta prvi puta:

- ❑ Na Standard traci s alatima kliknuti na ikonu diskete (Save). (Sl. 7-1)
- ❑ U Save as sučelju definirajte mjesto na koje želite pohraniti dokument (koristite strelicu za odabiranje određene mape). (Sl. 7-2)
- ❑ Definirajte ime dokumenta unosom putem tipkovnice.
- ❑ Odaberite Save klikom miša.

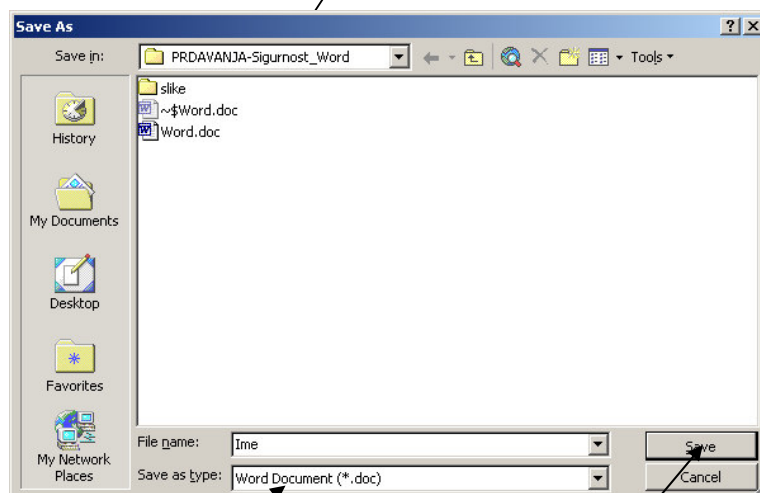
Spremanje dokumenta



Otvaranje postojećeg dokumenta

Sl. 7-2 Standardna traka s alatima

Odabrati lokaciju dokumenta

Upisati ime  
dokumentaPotvrditi  
spremanje

Sl. 7-3 Save As sučelje

### 7.1.1.3 Spremanje dokumenta s drugim imenom:

- ❑ Iz File izbornika odaberite opciju Save as.
- ❑ Odaberite mjesto i ime za datoteku (Sl. 7-3)..
- ❑ Kliknite naredbu Save.

Naredbi za spremanje dokumenta moguće je pristupiti kraćenjem putem tipkovnice, na način da pritisnemo tipku **Ctrl**, držimo ju pritisnutu i pritisnemo tipku **S (Ctrl+S)**. Kraćenja su dosta popularna kod naprednijih korisnika jer ubrzavaju rad. Na isti način moguće je skratiti put i do drugih funkcija npr **Ctrl+O** kratki put do naredbe File/Open i slično.



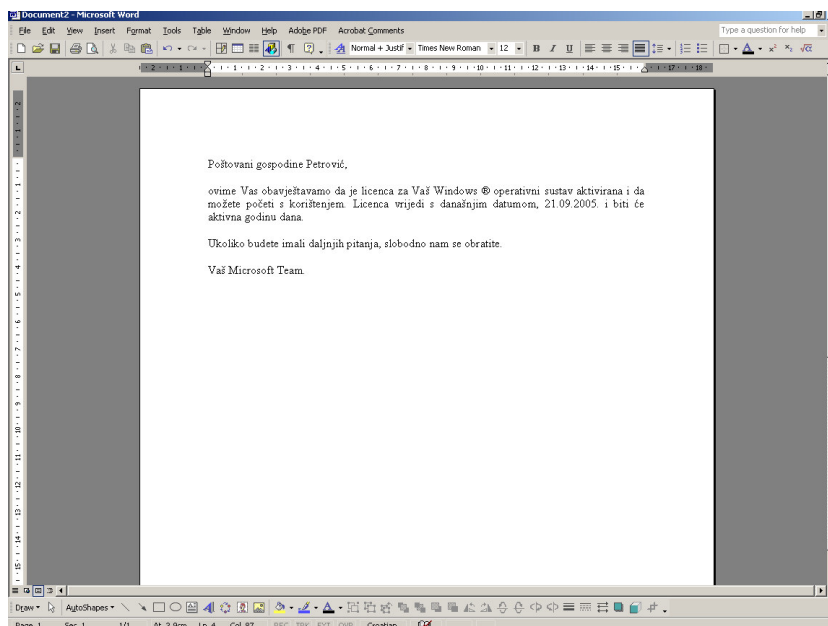
#### 7.1.1.4 Otvaranje postojećeg dokumenta

- Iz Standard trake s alatima odabrati naredbu Open.
- Odabrati dokument koji želite otvoriti.
- Kliknuti mišem na naredbu Open.

**Savjet:** preporučljivo je periodički snimati aktivni dokument iz sigurnosnih razloga. Npr. nestankom struje gubimo sav rad ukoliko prethodno dokument nije snimljen. Moguće je i da Word umjesto samostalno vrši periodičko snimanje u zadanom intervalu. Kako bi se aktivirala ova funkcija, potrebno je u izborniku **Tools/Options** odabrati karticu **Save**, aktivirati polje **Save AutoRecover Info every** i upisati u minutama vrijeme perioda automatskog snimanja, npr 1 min, 5 min. ili slično, na taj način se osiguravamo da u bilo kojem slučaju greške na računalu imamo sačuvanu zadnju snimljenu verziju dokumenta, a koja je stara maksimalno onoliko koliko je dugaćak odabrani vremenski interval snimanja ili nakon zadnjeg ručnog snimanja.

#### Vježba 1.

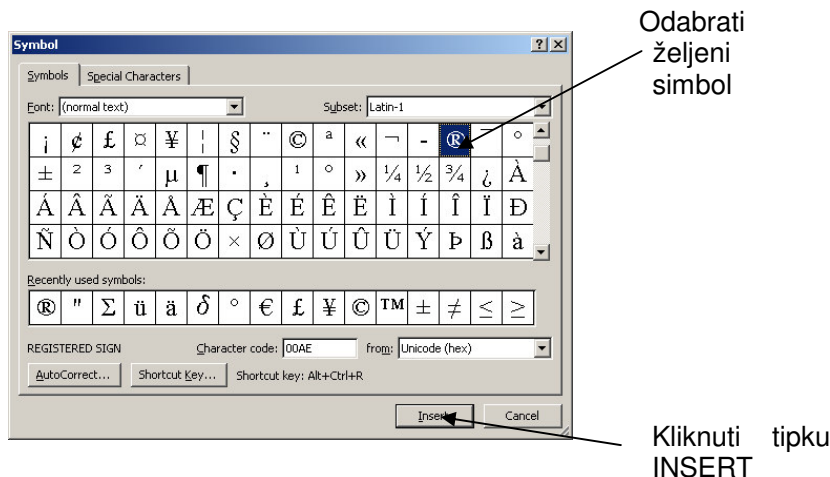
- Napisati tekst kao na Sl. 7-4.
- Pohraniti dokument na Desktop, ime datoteke je ime\_prezime.doc
- Otvoriti dokument, promijeniti datum aktivacije licence i spremi pod drugim imenom.



Sl. 7-4 Uz vježbu 1.

**Rješenje:**

- ❑ U poglavlju 2. objašnjeno je kako kreirati novi dokument.
- ❑ Nakon što je kreiran novi dokument, može se odmah početi unositi tekst. Prazan redak dobiti će se ako se pritisne tipku Enter.
- ❑ Simbol ® unijeti će se tako da se iz izbornika Insert odabere Simbol i potom u tablici pronade odgovarajući element. (Sl. 7-5)
- ❑ Snimanje dokumenta je objašnjeno u poglavlju 7.1.1.2.

**Sl. 7-5 Umetanje simbola**

U prethodnoj vježbi pokazano je kako otvoriti novi Word dokument, unijeti jednostavan tekst i simbole, kako spremiti dokument na željeno mjesto s odabranim imenom te kako otvoriti i napraviti jednostavne promjene na postojećem dokumentu.

U okviru ovog poglavlja još će se pokazati kako mijenjati vrstu pogleda na dokument i kada koristiti koji pogled.

**7.1.2 Vrste pogleda i način upotrebe**

Na Sl. 7-1 prikazano je, u donjem lijevom uglu, gdje se može promijeniti vrstu pogleda u kojoj se radi i to tako da se klikne na ime pogleda kojeg se želi aktivirati.

Vrste pogleda su:

- ❑ Normal View : pojednostavljeni prikaz dokumenta koji omogućuje jednostavno i brzo prilagođavanje ispisanog teksta
- ❑ Web Layout View: Prikazuje dokument kako bi izgledao prikazan u nekom Web pregledniku.
- ❑ Print Layout View: Ovaj pogled prikazuje dokument onakvim kakv će biti nakon ispisivanja na pisaču. Ovo je najčešće korišteni pogled.
- ❑ Outline View: Ovaj pogled prikazuje strukturu dokumenta s obzirom na numeriranje i razine kojima su pojedini stilovi dodijeljeni. Omogućuje i jednostavno mijenjanje tih razina.

### 7.1.3 Mijenjanje i prilagođavanje dokumenata

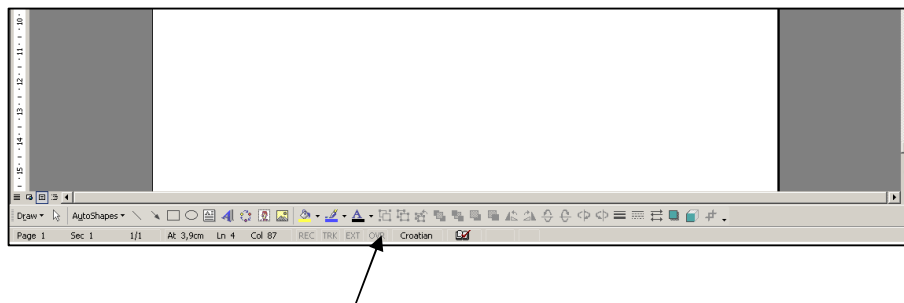
Većina Word dokumenata sadrži tekstualni dio kao smislenu cjelinu, uz eventualno još neke dodatne elemente poput slika, dijagrama ili nekih drugih objekata. Međutim, nije dovoljno samo unositi tekst, ne obazirući se na izgled i strukturu samog dokumenta. Word omogućuje korisniku, pomoću svojih alata za kreiranje stilova, pretrage dokumenta, gramatičkih alata, itd., da dokument personalizira ili stavi u neke određene estetske norme i norme kvalitete. To se pogotovo odnosi na dokumente koji su predviđeni za distribuciju prema većem broju korisnika. Isto tako, nije rijetkost da tvrtke imaju točno određene obrasce prema kojima se kreiraju svi dokumenti na razini te tvrtke.

U ovom poglavlju prikazati će se rad s nekim od alata za promjenu i prilagođavanje dokumenta, kojima se postižu gore navedeni zadaci.

### 7.1.4 Osnovno o promjenama teksta

Često se događa, dok se piše tekst da se želi u neku rečenicu nešto nadopisati. To se može napraviti tako da se klikne mišem na mjesto u rečenici gdje se želi umetnuti tekst i jednostavno počne pisati dodatni tekst. Može se na željeno mjesto u rečenici pozicionirati i pomoću strelica na tastaturi.

Ovdje je potrebno pripaziti na jednu stvar, jer se, ukoliko je uključena opcija **OVR** može dogoditi da se tekst koji se nadopisuje ne umeće u postojeći tekst, već se proširi preko postojećeg teksta i pobriše ga. Ovu opciju isključuje se ili tako da se na tipkovnici pritisne tipku **Insert** ili da se na dnu ekrana dvoklikom miša na **OVR** naredbu, uključi ili isključi naredbu (Sl. 7-6).



Naredba **OVR**

Sl. 7-6 Naredba **OVR**

**Kako obrisati tekst?** Na više načina, ovisno situaciji u kojoj se briše, npr. želi li se brisati samo jedno slovo, riječ, rečenicu i sl. Naredbom **BACKSPACE** na tastaturi briše se slovo po slovo, u smjeru s desna na lijevo u tekstu. (može se držati i tipku cijelo vrijeme pritisnutu). Može se koristiti i naredbu **Delete** s tastature, tada se briše s lijeva na desno.

Cijelu riječ može se selektirati na način da se 2 x mišem klikne na riječ, rečenicu se može selektirati na način da se klikne na jednu riječ u rečenici držeći pritisnutu tipku **Ctrl**, a odlomak teksta na način da se 3 puta klikne na tekst unutar odlomka. (Pod odlomkom teksta se u načelu podrazumijeva dio teksta koji je od ostatka odvojen pritiskom na tipku **Enter**). Nakon što se selektira dio teksta, on će promijeniti boju i pritiskom na tipku **Delete** ili **Backspace**, obrisati će se označeni dio teksta. Cijeli

tekst s pripadajućim dodatnim elementima može se obrisati tako da ga se prethodno selektira kraticom **Ctrl+A**, i zatim pritisne tipku **Delete**.

**Undo i Redo naredbe** su naredbe koje se dosta često koriste. Dok se rade bilo kakve promjene na dokumentu, Word te promjene bilježi. Tako je tipkom **Undo** moguće na jednostavan način vratiti posljednji, ili posljednjih nekoliko učinjenih koraka. Tipke **Undo i Redo** nalaze se na **Standard** traci s alatima. Moguće je i vraćanje direktno do zadnjeg željenog koraka, upotrebom **Undo Liste**.

Nakon što se selektira neki dio teksta, može ga se premjestiti na neku drugu lokaciju unutar dokumenta, kopirati ga i postaviti na novu lokaciju ili ga obrisati, što je već objašnjeno. Opcija kopiranja dosta se često koristi pa je dalje detaljno objašnjena..

**Kopiranje teksta** radi se tako tako da se prvo selektira dio koji se želi kopirati. Zatim se iz izbornika **Edit** odabere naredbu **Copy**. Sada se kopija odabranog elementa (može biti slika, tekst ili nešto drugo) nalazi u radnoj memoriji računala, u tzv. **Clipboardu**. Sljedeći je korak da se pozicionira končanicu na mjesto gdje se želi umetnuti kopirani element (ako je kopiran odlomak teksta, na mjesto gdje je končanica doći će gornja lijeva riječ paragrafa). Sada treba ponovno otići u izbornik **Edit** i odabrati **Paste** naredbu. Kopija elementa pojavljuje se na novoj lokaciji. Prilikom umetanja kopiranog elementa može se odrediti želi li se kopirani element prilagoditi formatu destinacije ili zadržati izvorno formatiranje.

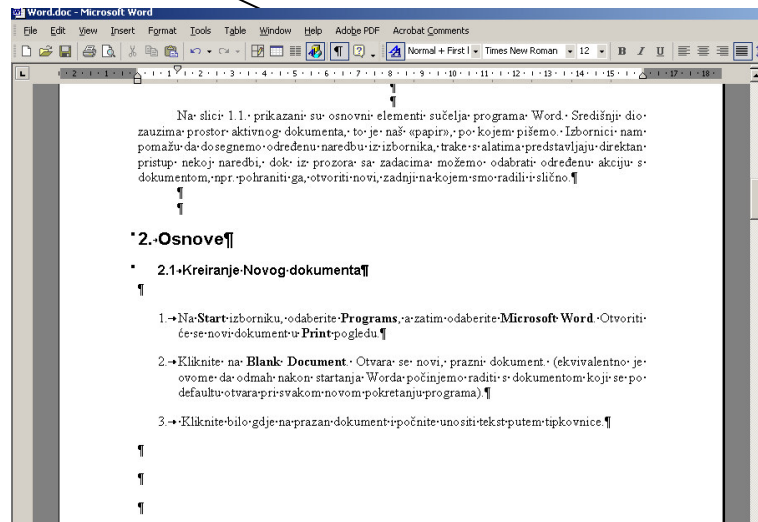
**Premještanje teksta** vrši se na dva osnovna načina. Prvi je **Drag and Drop** metodom. Potrebno je prvo selektirati tekst, zatim kliknuti mišem na selektirani tekst, držati tipku miša pritisnuta i povući tekst na novu lokaciju. Druga je metoda slična postupku kopiranja, samo što se umjesto naredbe **Copy** koristi naredba **Cut**. Ostatak postupka je isti. Treba napomenuti da se ove naredbe mogu kratiti putem tipkovnice, što napredniji korisnici široko koriste. Kraćenja su: **Ctrl+C (Copy)** za kopiranje u **Clipboard**, **Ctrl+X (Cut)** za kopiranje u **Clipboard** i brisanje izvorišnog elementa, a **Ctrl+V (Paste)** je umetanje elementa.

Bitno je napomenuti da ove naredbe kopiranja teksta nisu ograničene samo na jedan Word dokument. Moguće je kopirati iz jednog dokumenta u drugi, pa i kopirati i umetati elemente između raznih Microsoft Office aplikacija, npr. između PowerPointa i Worda.



Jedna korisna naredba pri uređivanju teksta je **Show/hide ¶** naredba (Sl. 7-7). Ako je uključena, dobivaju mnoge informacije o samom tekstu, odlomci su označeni simbolom ¶, razmaci između riječi označeni su točkom. U načelu, dok se piše tekst, stalno se uključuje i isključuje ovu naredbu i ispravljaju greške, npr. dvostruke razmake između riječi itd. Do ove naredbe dolazimo preko trake s alatima **Standard**.

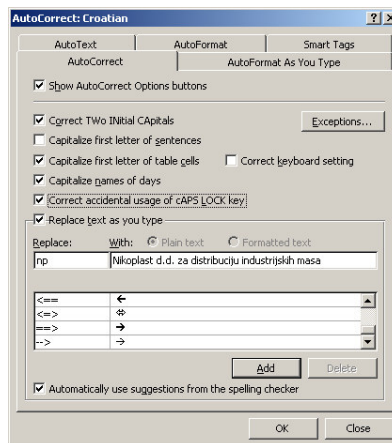
Show/hide naredba



Sl. 7-7 Uređivanje teksta

#### 7.1.4.1 Korištenje kratica pri unosu teksta (AutoCorrect i Autotext naredbe)

Često se pri unosu teksta rade slične pogreške. Na primjer, ne počne se rečenicu velikim slovom, zamijene se dva slova u nekoj riječi, ili se namjerno želi neke fraze koje se često koriste, unijeti putem kratice u tekst. Ovo se može izvesti **AutoCorrect i Autotext** naredbama. Na primjer, neka se tvrtka zove *Nikoplast d.d. za distribuciju industrijskih masa*, a u dokumentima koji se pišu, često treba napisati ime firme. Korisno bi bilo napraviti kraticu, koju kad bi se napisalo napisali, Word automatski pretvori u ime firme. Recimo da se za kraticu želi odabrati **np**. Dakle, kad se napiše **np**, želi se da umjesto tog teksta dobijemo **Nikoplast d.d. za distribuciju industrijskih masa**.



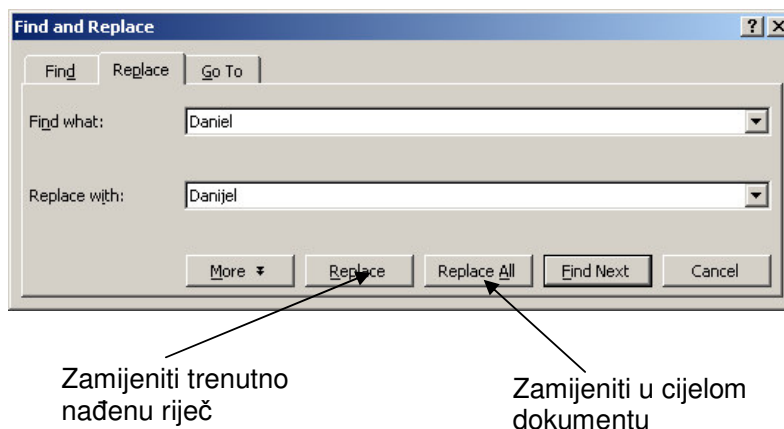
Sl. 7-8 Kartica automatskih ispravaka

Potrebno je napraviti sljedeće: iz izbornika **Tools** odabrati **AutoCorrect Options**. U polju **replace** unosi se *np*, a u polju **with** unosi se *Nikoplast d.d. za distribuciju industrijskih masa*. Time je definirano da nakon što se napiše *np* i pritisne razmaknicu, dobija se na ekranu ispisano puno ime firme. U okviru ove naredbe moguće je definirati još mnoge korisne akcije koje Word samostalno izvodi, Sl. 7-8.

Slična je situacija korištenje **AutoText** naredbe. Do naredbe se dolazi preko izbornika **Insert/AutoText** gdje se može preddefinirati elemente teksta koje često koristimo.

#### 7.1.4.2 Pronalaženje i zamjena određene riječi u tekstu

U situaciji kada je kreiran dokument s velikom količinom teksta i ustanovljeno je da jedna riječ cijelim dokumentom krivo napisana, koristiti će se naredba nađi i zamijeni. Primjerice umjesto imena *Danijel*, u cijelom tekstu piše *Daniel*. Bilo bi prilično mukotrpno riječ po riječ pregledavati tekst, dugo bi trajalo, a vjerojatno bi se na kraju i negdje isпустиo napraviti ispravku. Word na sreću sadrži alat pomoću kojega je jednostavno moguće pronaći neku riječ u tekstu, zamijeniti neku riječ samo na jednom mjestu, ili u cijelom dokumentu odjednom. Naredbe koje za to služe su **Find** i **Replace**, a nalaze se u izborniku **Edit**. Dakle, nakon pozivanja jedne od ovih dviju naredbi, potrebno je samo definirati želi li se pronaći sljedeću zadanu riječ, zamijeniti samo trenutno pronađenu riječ ili zamijeniti riječ u cijelom dokumentu odjednom, Sl. 7-9



Sl. 7-9 Nađi i zamijeni

#### 7.1.4.3 Ispravljenje gramatičkih pogrešaka

Čitanje teksta s mnogim gramatičkim pogreškama ostavlja loš dojam na čitatelja. Gotovo da je neminovno da će se tijekom pisanja pojaviti greške u tekstu, bilo slučajne, tzv. *tipfeler* bilo da će se pojaviti greške uslijed autorovog nepoznavanja gramatike i pravopisa. Word nije svemoguć u odstranjivanju ovakvih pogrešaka, ali ipak neke osnovne greške mogu se ispraviti korištenjem ugrađenih alata. Zato završna provjera teksta treba uvijek biti pažljivo i detaljno iščitavanje teksta, kako bi se uklonile sve zaostale pogreške. Dok se tekst unosi, ukoliko naiđe na neku riječ koju nema u svom riječniku, Word ju podvlači crvenom linijom. Ukoliko naiđe na moguću



gramatičku grešku, podvlači ju zelenom bojom. Ipak, neke verzije Worda nemaju dobro definirane ove funkcije za hrvatski jezik. Nailaskom na riječ koja je podvučena, crvena, u većini slučajeva odmah je vidljivo u čemu je greška i može je se ispraviti manualno, ili pritisnuti desnom tipkom miša na tu riječ i tražiti sugestije od Worda. Naredba kojom se provodi pregledavanje dokumenta nalazi se pod **Tools/Spelling and Grammar**, gdje je ujedno moguće i uređivati postavke vezane uz provjeru i korigiranje teksta.

### 7.1.5 Uređivanje izgleda teksta

Izgled i struktura nekog teksta pomaže čitatelju u prenošenju informacija i lakšem razumijevanju iznesene materije. Svi žele čitati tekst koji je smisleno strukturiran, s istaknutim bitnim detaljima, tekst u kojem se lako snaći i brzo je moguće vratiti se na neke bitne točke. Word sadrži niz alata i metoda pomoću kojih je moguće uređivati izgled teksta.

Prije nego što se krene s izlaganjem, uvesti će se nekoliko osnovnih pojmova koji definiraju tekst. Najprije to je **font**. Pod fontom se smatra skupina znakova kojima je dizajn zajednički. Fontovi mogu biti dakle različiti, a sama slova različite veličine, koja je definirana **veličinom fonta**. Neki najčešće korišteni fontovi su Times New Roman, Arial, Courier.

Sljedećih nekoliko atributa definiraju osnovni stil fonta:

- ❑ **Veličina fonta**, mjeri se u točkama (points), jedna točka predstavlja 1/72 dio inča.
- ❑ **Stil fonta**, svakom fontu može se po želji promijeniti stil. Tu se prvenstveno misli na sljedeće značajke: osnovni, **bold**, *italic*, **bold italic**.
- ❑ Moguće je još mijenjati razne efekte kao podvlačiti ili precrtati tekst, osjenčati tekst, promijeniti boju teksta ili razmak među slovima.

Napomena: uporabom različitih efekata moguće je povećati atraktivnost dokumenta i učiniti ga da izgleda profesionalno. Ipak, nije poželjno pretjerivati jer će se u suprotnom postići suprotni efekt. Pravilo je da u nekom dokumentu ne bi trebalo biti više od tri različita fonta.

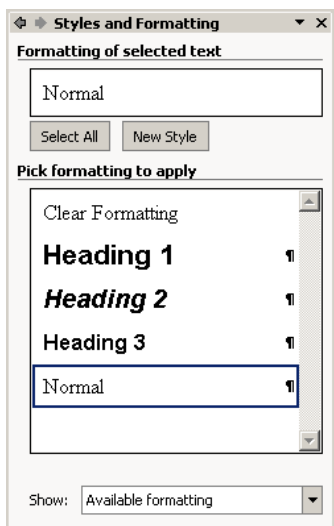
### 7.1.6 Uređivanje teksta pomoću stilova

Uređivanje teksta je možda i najzahtjevnija operacija u smislu utroška vremena pri izradi dokumenta. Uređivanje teksta uključuje više zadataka, poput oblikovanja teksta, dosljednosti u oblikovanju, poštivanje smjernica na razini neke organizacije i slično. Ispravnim korištenjem stilova moguće je u značajnoj mjeri smanjiti utrošak vremena i truda potrebnog za željeno oblikovanje teksta.

Stil sadrži detaljne informacije o izgledu odlomka teksta. Te informacije odnose se na font, margine, prazni prostor ispred i iza odlomka, hijerarhijske razine odlomaka i još neka obilježja. Umjesto da se svaka od ovih informacija unosi posebno za svaki pojedini odlomak, kreirati će se *stilovi* koji će te informacije sadržavati, a odabirom željenog stila, nalaziti ćemo se u potpuno definiranom prostoru teksta. Prednosti ovakvog načina rada ne prestaju ovdje. One posebno dolaze do izražaja pri radu sa višerazinskim naslovima, kreiranju automatski generiranih tabela sadržaja, unakrsnih spona pri numeriranju elemenata i još mnogo toga.

### 7.1.6.1 Pregledavanje i korištenje stilova

Prilikom otvaranja novog Word dokumenta, definirana su četiri osnovna stila. Oni najčešće nisu dovoljni za uređivanje cijelog dokumenta, no na temelju njih definiraju se osnovni stilovi prilagođeni svakom pojedinom dokumentu i kreiraju novi. Odabere li se iz izbornika Format naredba Styles and formatting, s desne strane ekrana pojaviti će se prozor s dostupnim stilovima i naredbama koje pomoću kojih je moguće kreiranje novih i mijenjanje svojstava postojećih stilova.



Sl. 7-10 Osnovni stilovi

Na Sl. 7-10 Prikazani su stilovi koji su sadržani u svakom novootvorenom dokumentu. Heading 1 do 3 služe za definiranje naslova, dok je stil normal predviđen za osnovni tekst. Clear formatting nije zapravo stil već naredba kojom je moguće ukloniti primjenu stila. Pri dnu slike potrebno je obratiti pažnju na Show padajući izbornik. Ukoliko kliknemo na strelicu izbornika dobiti ćemo 5 dodatnih mogućnosti odabira:

- ❑ Available formatting: prikazani su samo osnovni stilovi, bez obzira na to jesu li već korišteni u dokument, svaki dodatno korišteni stil, te svaka promjena koja je napravljena na tekstu neovisno o stilu (npr. masna slova, kosa slova)
- ❑ Formatting in use: prikazani su stilovi koji su korišteni u dokumentu te dodatne izmjene teksta neovisno o stilu.
- ❑ Available styles: prikazuje osnovne stilove i sve dodatno korištene stilove.
- ❑ All styles: prikazuje veću listu stilova koje je moguće koristiti. Primjerice, umjesto tri osnovna stila za naslov, ovdje je prikazano 9 stilova za naslove, od heading 1 do heading 9.
- ❑ Custom: ovdje je moguće detaljno definirati što se želi imati prikazano u listi.

### 7.1.6.2 Primjena stila na odlomak teksta

Potrebno je končanicu pozicionirati unutar odlomka na koji želimo primijeniti stil (ne označavati dio teksta ili cijeli odlomak!), te iz izbornika Styles and Formatting odabrati željeni stil. Ukoliko se stil kojega želimo primijeniti ne vidi u listi, potrebno je pomoću naredbe Show prikazati cijelu listu pa odabrati željeni ili kreirati vlastiti stil. Ako je slučajno selektiran dio odlomka, pa zatim odabran stil, tada će se promjene odnositi samo na selektirani dio, a ne na cijeli odlomak.

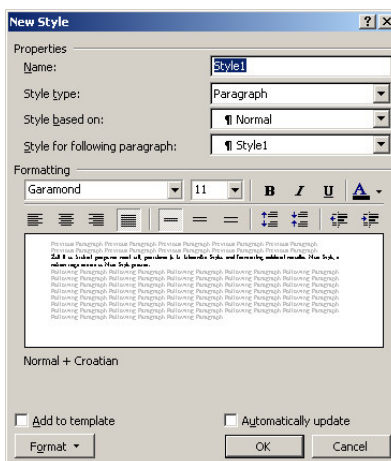
Neke dodatne informacije o stilu mogu se dobiti ako se končanica pozicionira ispred imena stila iz liste i drži nekoliko trenutaka. Informacije se odnose na font, veličinu slova, poravnanje odlomka.

Također, moguće je mijenjati stil nekog odlomka kroz cijeli dokument odjednom. U situaciji kada se npr. želi zamijeniti stil headnig 4 stilom heading 3 kroz cijeli dokument odjednom, treba pozicionirati končanicu iznad stila heading 4, odabrati iz padajućeg izbornika koji se pojavljuje nakon nekoliko trenutaka naredbu *select all x instances* te zatim odabrati stil headnig 3.

### 7.1.6.3 Promjena postojećeg i kreiranje novog stila

Već je prije spomenuto da najčešće osnovni stilovi ne zadovoljavaju potrebe pri izradi nekog dokumenta prilagođenog specifičnim potrebama. Stoga je potrebno ili modificirati postojeće stilove, ili kreirati nove, a najčešće se radi i jedno i drugo.

Želi li se kreirati potpuno novi stil, potrebno je iz izbornika Styles and formatting odabrati naredbu New Style, a nakon toga otvara se New Style dijaloški okvir Sl. 7-11.



Sl. 7-11 New style dijaloški okvir

- ❑ Name: dodjeljuje se ime budućeg stila
- ❑ Style type: odabire se Paragraph, table, character ili list, te u skladu s odabranim objektom na koji će se budućći stil odnositi, aktiviraju se dostupne opcije za uređivanje stila
- ❑ Style based on: moguće je, ukoliko se to želi, odabrati stil na kojemu će se bazirati budućći stil. Ova funkcija može uzrokovati probleme u nekim situacijama. Ukoliko se u kasnijem radu s dokumentom promijeni stil koji je baza nekog drugog stila, može se i taj drugi stil promijeniti, iako se to ne želi. Stoga je u dokumentu potrebno prije početka rada definirati jedan stil, na kojemu se temelje svi ostali, a koji se tijekom daljnjeg rada s dokumentom ne smije mijenjati. Na taj način osiguravamo kontrolu nad dokumentom i stilovima.
- ❑ Style for following paragraph: određuje se što se događa nakon pritiska tipke enter, odnosno izlaza iz aktivnog odlomka. Određuje se kojim stilom nastavljamo daljnji rad.
- ❑ Formatting: ovdje se zapravo oblikuje stil. Ako se kreira Paragraph stil, ovdje je moguće definirati razmake između linije, uvlake, fontove, boju slova.
- ❑ Add to template: aktivirati ovu opciju ukoliko se želi snimiti kreirani stil kako bi se kasnije mogao koristiti s drugim dokumentima.

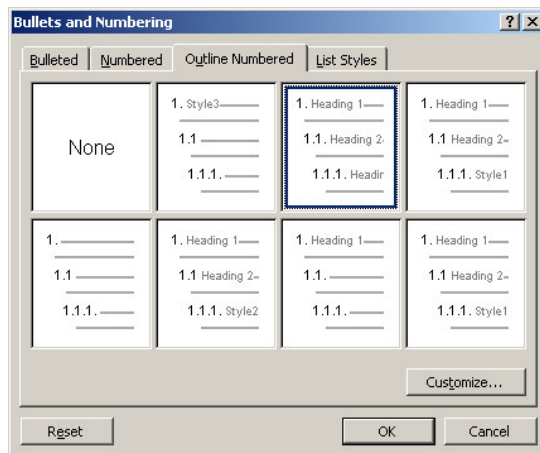
- ❑ Automatically update: ova se opcija najčešće ne aktivira jer onemogućuje korisniku kontrolu nad dokumentom. Radi se o tome da u nekom dokumentu pri bilo kakvoj promjeni u odlomku koji koristi određeni stil, ista promjena će se izvesti na svim ostalim odlomcima koji koriste isti stil.
- ❑ Format aktivacijom ove naredbe moguće je u potpunosti detaljno oblikovati stil.

Ispravan način rada sa stilovima podrazumijeva da se, nakon kreiranja novog dokumenta a prije početka unošenja teksta, kreiraju stilovi za koje je moguće odmah pretpostaviti da će se koristiti. Ukoliko će se u dokumentu koristiti naslovi s više razina, potrebno je definirati stilove za takve naslove, odrediti jedan stil, najčešće je to Normal, koji dolazi sa svakim novim dokumentom, na kojem će se bazirati ostali stilovi. U tom slučaju, stil Normal ne primjenjuje se za formatiranje odlomaka niti se u kasnijim fazama rada mijenja, već služi isključivo kao baza za ostale stilove pa kao takav mora biti stabilan. Prije početka rada nije najčešće moguće predvidjeti potrebu za svim stilovima koj u dokumentu mogu zatrebati, pa je takve stilove moguće dodati i kasnije.

#### 7.1.6.4 Kreiranje naslova s više razina

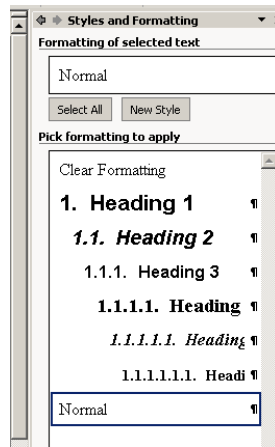
Dokumenti s razgranatom hijerarhijskom strukturom često sadrže ovakve naslove s više razina i numeracijske liste koje ih povezuju, a ažuriraju se automatski u ovisnosti o razini naslova. Ovdje će se prikazati kreiranje višerazinske liste, krenuvši od praznog dokumenta.

Nakon otvaranja novog dokumenta, dostupni stilovi prikazani su Sl. 7-10. Dostupna su tri stila za naslove, s definiranim razinama, međutim nisu numerirani. Potrebno je desnim klikom miša odabrati heading 1 stil, odabrati opciju Modify/ Format/ Numbering. Pojavljuje se dijaloški okvir prikazan na slici, iz kojeg je potrebno odabrati Outline Numbered karticu i jedan od stilova numeriranja, Sl. 7-12.



Sl. 7-12 Kreiranje višerazinskih listi

Nakon što se definira numeracija, u izborniku stilova pojavljuju se novonastali numerirani stilovi za naslove. Ukoliko treba dodati novi stil sljedeće, niže razine, potrebno ga je odabrati pomoću naredbe Show (Sl. 7-10) i aktivirati. Nakon ovog niza operacija lista sa stilovima trebala bi izgledati ovako:



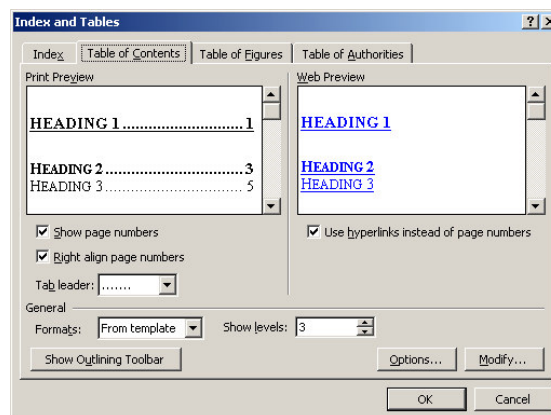
Sl. 7-13 Numerirane višerazinske liste za naslove

Osim što je svaki naslov definiran za zasebnu cjelinu, vidljivo je i da je definiran s različitim fontom, a i ostalim parametrima koje je potrebno prilagoditi željenim vrijednostima, dakle razmak prije i poslije odlomka u odabranom stilu, stil koji slijedi, font, razmak slova.

### 7.1.7 Kreiranje tablice sadržaja

Tablica sadržaja sadrži popis svih naslova sadržanih u dokumentu, uz odgovarajući broj stranice na kojoj se u dokumentu naslov nalazi. Ukoliko je dokument kreiran na ispravan način, korištenjem stilova za naslove, kreiranje tablice sadržaja se znatno pojednostavljuje jer ne treba prolaziti cijelim dokumentom i označavati svaki navod koji se želi izlistati u sadržaju. Za kreiranje tablice sadržaja potrebno je napraviti sljedeće korake:

- ❑ Postaviti končanicu na mjesto gdje se želi pozicionirati tablica sadržaja.
- ❑ Odabrati izborniki Insert/Reference/Index and tables
- ❑ Odabrati format tablice. Kako se odabiru razni formati, moguće je promjene pratiti putem Print preview i Web preview prozora. Ako je pri kreiranju tablice odabrana opcija From template, naredbom Modify moguće je urediti izgled naslova za svaku posebnu razinu.



Sl. 7-14 Table of contense kartica izbornika Index and Tables

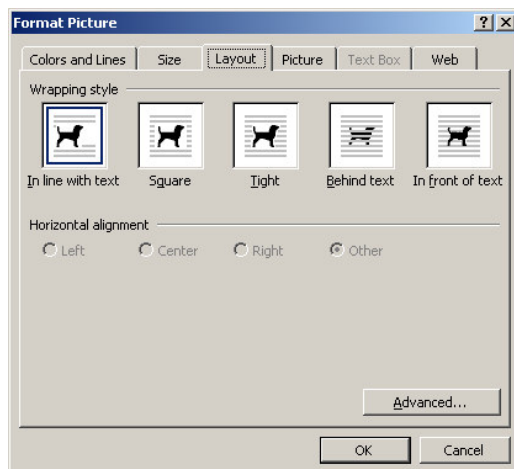
- ❑ Putem kartice sa Sl. 7-14 moguće je još odrediti želi li se ispis stranica, poravnanje liste s desne strane kao i izgled linija koje spajaju broj stranice s odgovarajućim naslovom.

### 7.1.8 Unošenje slika

Ukoliko postoji potreba za unošenjem slike u dokument, potrebno je iz izbornika Insert odabrati naredbu Picture. Nudi se nekoliko mogućnosti izbora:

- ❑ Clip Art: odabire se slika ukoliko postoji potrebu za standardnim elementima, npr. slikom računala, miša, ili nešto slično. Baza ovih slika je integrirana u Word, a može se nadopunjavati vlastitim slikama
- ❑ From file: ukoliko je poznata mapa neke slike koja se želi umetnuti u dokument, odabrati će se ova opcija, a zatim će se izabrati željena slika.
- ❑ From scanner or camera: ukoliko je računalo spojeno sa skenerom ili kamerom i želi se umetnuti slika s jednog od ova dva uređaja, odabrati će se ova opcija.

Nakon unošenja, sliku je putem izbornika Format/Picture dodatno urediti. To se odnosi na razmještaj teksta oko slike, koji može biti sa strane, ispred ili iza slike, oko slike. Također, moguće je precizno odrediti fizičku veličinu slike, i pozicionirati, Sl. 7-15.



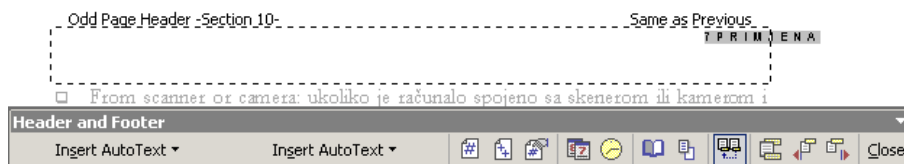
Sl. 7-15 Dijaloški ovir za uređivanje slike

### 7.1.9 Kreiranje zaglavlja i podnožja dokumenta

Zaglavlja i podnožja dokumenta definirani su prostorom pri vrhu odnosno dnu stranice, u koje je moguće umetnuti odgovarajući tekst, sliku ili polje, a pojavljuju se na svakoj stranici dokumenta, ukoliko nije drugačije određeno. To se možem zamisliti kao da pišemo po nekom papiru iz bloka, na kojem u vrhu ili dnu stranice na svakoj stranici dolazi isto polje. Najčešće se podnožje stranice koristi da se u njega umetne polje s brojem stranice, a zaglavlje da se umetne naslov cjeline odnosno poglavlja dokumenta. Moguće je definirati posebno zaglavlje za prvu stranicu dokumenta, te posebno za parnu i neparnu stranicu dokumenta. Kako se broj stranice dokumenta povećava, povećava se i vrijednost polja broja stranice u podnožju, a kako se mijenjaju naslovi i poglavlja, mijenja se u zaglavlju tekst koji označava to poglavlje.

Kako bi se dodalo zaglavlje ili podnožje dokumentu, potrebno je učiniti sljedeće korake:

- ❑ Odabrati iz izbornika View/Header and Footer. Dokument se prikazuje u Print layout pogledu, uz traku s alatima Header and Footer. Područje koje zatvara podnožje odnosno zaglavlje označeno je iscrtanom linijom (ta se linija neće vidjeti prilikom ispisa dokumenta). Tekst dokumenta je izblijeđen, a tekst zaglavlja ili podnožja pojavljuje se u normalnoj boji. Za promjenu između podnožja i zaglavlja, treba kliknuti naredbu Switch Between Header and Footer.



### 7.1.13 Kreiranje zaglavlja i podnožja dokumenta

#### Sl. 7-16 Definiranje podnožja i zaglavlja

- ❑ Unijeti tekst u označeno područje, unijeti AutoText polja, oblikovati tekst.
- ❑ Ako imamo potrebu za unošenjem polja koja nisu dostupna direktno iz header and footer izbornika, potrebno je iz izbornika Insert odabrati naredbu Field. Na taj način moguće je definirati polje koje će npr. sadržavati broj i oznaku poglavlja dokumenta.

Ukoliko se ne žele ista zaglavlja i podnožje za dokument s više odlomaka, moguće je to promijeniti tako da se pozicionira u na mjesto u dokumentu gdje se želi prekinuti kontinuitet, te odabere opcija Same as previous. Tako se gubi veza s prethodnim dijelom dokumenta i moguće je kreirati novo zaglavlje i podnožje.

#### 7.1.9.1 Uređivanje različitog zaglavlja prve stranice dokumenta

Mnogi dokumenti imaju različito zaglavlje i podnožje od ostatka dokumenta, ili uopće nemaju zaglavlje i podnožje na prvoj stranici. Iz tog razloga Word daje mogućnost definiranja različitog zaglavlja i podnožja prve stranice svakog dijela dokumenta. Za kreiranje različitog zaglavlja, treba napraviti sljedeće korake:

- ❑ Odabrati View/Header and footer
- ❑ Odabrati Show previous ili Show Next da bi se pozicionirao dio dokumenta u kojem se želi definirati prva stranica
- ❑ Odabrati naredbu page setup iz izbornika Header and Footer
- ❑ Odabrati Layout karticu
- ❑ U Header and Footer dijelu, odabrati Different First Page
- ❑ Odabrati This Section iz Apply to liste
- ❑ Kliknuti OK.
- ❑ Ukoliko se ne želi nikakav header ili footer na prvoj stranici, potrebno je ostaviti prostor praznim.
- ❑ Odabrati Close na izborniku Header and Footer.

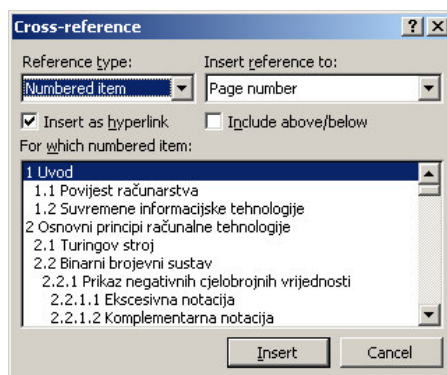
### 7.1.9.2 Kreiranje zaglavlja i podnožja za parne i neparne stranice dokumenta

Ponekad se želi u dokumentu kreirati različita zaglavlja i podnožja za parne i neparne stranice. To je korisno npr. kod dokumenata koji su predviđeni za uvezivanje, pa se želi da uvijek zaglavlje bude pozicionirano na vanjsku marginu stranice. U tom slučaju odabrati će se naredba Different Odd and Even u Page Setup dijaloškom okviru Headers and Footers izbornika.

### 7.1.10 Kreiranje unakrsnih spona među elementima dokumenta

Unakrsna spona kreira se na način da se umetne fiksni tekstualni dio, kojemu se kasnije pridijeli referenca. Unakrsna spona može sadržavati više referenci. Na primjer može se reći: Pogledati poglavlje 10 na str. 225. U ovom primjeru, napisalo bi se Pogledati, zatim bi se umetnula referenca na željeno poglavlje, zatim napisati str. pa umetnuti referencu na stranicu 225. Unakrsne spona mogu se umetati i u podnožja i zaglavlja dokumenta. Za umetanje treba izvršiti sljedeće korake:

- ❑ Postaviti končanicu na točno ono mjesto gdje se želi umetnuti unakrsnu sponu.
- ❑ Odabrati Insert/Reference/CrossReference, vidi Sl. 7-17.



Sl. 7-17 Kreiranje unakrsnih referenci

- ❑ Odabrati kategoriju u Reference Type listi. For which i Insert Reference To liste mijenjaju se u ovisnosti o referentnom tipu reference koji je odabran. Na primjer, ako je referentni tip Heading, For Which lista sadrži sve naslove iz dokumenta, (naravno pod pretpostavkom da su korišteni stilovi za naslove), dok Insert Reference To lista prikazuje različite tipove informacija koje je moguće umetnuti. Npr tekst naslova ili broj stranice.
- ❑ Iz insert reference to liste odabrati informaciju koja se želi umetnuti u unakrsnu sponu. Primjetno je da svaka kategorija sadrži kao mogućnost umetanja broj stranice, s kojim se može referirati na stranicu na kojoj se dotični element nalazi.
- ❑ Odabrati točnu referencu koja se želi iz For Which liste. Npr. ako se kao referenca odabre Heading, For Which Heading lista koristi sve Headinge dokumenta.
- ❑ Odabrati Insert. Cross-Reference dijaloški okvir se ne zatvara kako bi se moglo dodati još referenci.
- ❑ Kad je umetanje završeno, treba zatvoriti ovaj dijaloški prozor.

Sljedeća opcija u Cross-reference dijaloškom okviru je Insert As Hyperlink pomoću koje je moguće uspostaviti vezu između reference i njenog izvršnog elementa u tekstu.

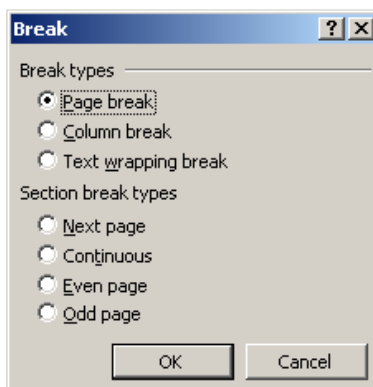


### 7.1.11 Definiranje prijeloma stranica

Kreira li se novi dokument koristeći naredbu New/Blank Document, Word će cijeli dokument tretirati kao jednu cjelinu. Unosom prijeloma između elemenata dokumente, moguće je dijeljenje dokumenta na proizvoljan broj cjelina. Jedan primjer uporabe naredbe za prijelom dokumenta je u slučaju da je na razini dokumenta definiran format papira A4, a unutar dokumenta potrebno je umetnuti stranicu A3 formata. U tom slučaju, ispred stranice A3 formata bio bi definiran prijelom, unesena nova vrijednost za format stranice, dakle A3 format, te nakon te stranice opet unesen prijelom i preuzeta svojstva formata od ostatka dokumenta. Dakle, naredbama za prijelom moguće je izdvojiti određene cjeline dokumenta i mijenjati im svojstva neovisno od preostalog dijela.

Za unos prijeloma potrebno je učiniti sljedeće:

- Pozicionirati končanicu na mjesto gdje se želi prijelom
- Odabrati Insert/Break. Pojavljuje se Break dijaloški okvir Sl. 7-18
- Odabrati jednu od ponuđenih mogućnosti, objašnjenje slijedi niže u tekstu
- Kliknuti mišem na naredbu OK.

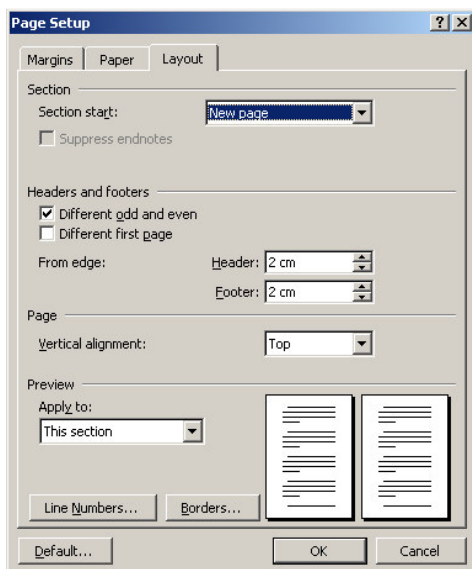


Sl. 7-18 Dijaloški okvir naredbe za prijelom

- Next page: prijelom uz prijelaz na novu stranicu i početkom nove cjeline na početku nove stranice
- Continuous: rijelom bez prijelaza na novu stranicu
- Even Page: prijelom uz prijelaz na parnu stranicu
- Odd page: prijelom uz prijelaz na neparnu stranicu.

### 7.1.12 Formatiranje stranica za ispis dokumenta

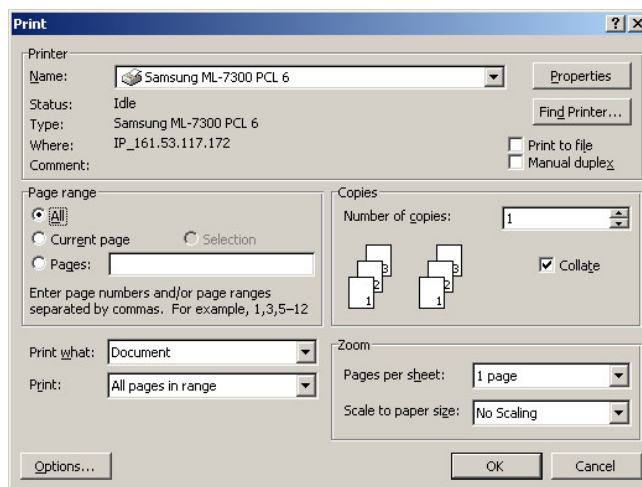
Pod formatiranjem se podrazumijeva uređivanje margina dokumenta, orijentacije i veličine papira, te zaglavlja i podnožja dokumenta. Ovi elementi dokumenta uređuju se svakako prije ispisa na pisaču, a detaljno ih je moguće urediti putem izbornika File/Page Setup, Sl. 7-19.



Sl. 7-19 Page Setup dijaloški okvir

Margine predstavljaju prostor između rubova teksta i kraja papira. To ne mora biti prazan prostor, može sadržavati podnožja ili zaglavlja dokumenta. Veličina margina može se mijenjati ili pomoću ravnala ili pomoću Page Setup naredbe. Formatiranje se može odnositi na jednu stranicu, dio dokumenta ili cijeli dokument. To ovisi o odabiru elemenata prije no što se pozove Page Setup naredba.

Nakon formatiranja dokumenta, želi li se dokument ispisati, poziva se naredba Print iz izbornika File, Sl. 7-20.



Sl. 7-20 Ispis dokumenta

U Print dijaloškom okviru potrebno je definirati pišač za ispis dokumenta, definirati stranice koje se želi ispisati ukoliko se ne ispisuje cijeli dokument, broj kopija, broj stranica dokumenta po jednom listu papira. Nakon definiranja navedenih parametara, pritiskom na naredbu OK počinje slanje dokumenta na pišač i ispis.

---

## 8 LITERATURA

---

1. Brookshear, J. G., Computer Science, The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1993.
2. Karanjit, S. S., Windows NT Server 4, Professional Reference, New Riders Publishing, Indianapolis, 1996.
3. Minasi, M., Anderson, C., Smith, B., Toombs, D., Windows 2000 Server, Sybex, San Francisco, 2000.
4. Graham, I. S., HTML Sourcebook, John Wiley & Sons, Inc., New York, 1995.
5. Winder, R., Roberts, G., Developing Java Software, John Wiley & Sons, Inc., New York, 1998.
6. Razni autori, Computer, Innovative Technology for Computer Professionals, IEEE Computer Society Magazine, IEEE, 1995.-2000.
7. Grupa autora, Using UNIX, QUE Corporation, Carmel, 1990.
8. Lee, K., Principles of CAD/CAM/CAE Systems, Addison Wesley Longman Inc., Reading, 1999.
9. Tanimoto, S. L., The Elements of Artificial Intelligence, Computer Science Press, Rockville, 1987.

