

MR MILOŠ NIKIĆ

Osnovi računarstva

Beograd
2011

Sadržaj

01 Razvoj računara.....	5
<i>Kibernetika, informatika i računarstvo.....</i>	<i>5</i>
<i>Analogno i digitalno.....</i>	<i>5</i>
<i>Kratka istorija razvoja računara.....</i>	<i>6</i>
02 Računar i računarski sistem.....	12
<i>Matična ploča.....</i>	<i>12</i>
<i>Procesor.....</i>	<i>13</i>
<i>Unutrašnja memorija.....</i>	<i>16</i>
03 Periferijski uređaji.....	18
<i>Ulazni uređaji.....</i>	<i>18</i>
<i>Izlazni uređaji.....</i>	<i>21</i>
<i>Ulazno-izlazni uređaji.....</i>	<i>24</i>
04 Spoljna memorija.....	26
<i>Spoljna memorija sa nepromenljivim medijem.....</i>	<i>27</i>
<i>Spoljna memorija sa promenljivim medijima.....</i>	<i>29</i>
05 Brojni sistemi.....	33
<i>Vrednost broja u dekadnom sistemu.....</i>	<i>34</i>
<i>Prevođenje broja iz dekadnog u drugi sistem brojeva.....</i>	<i>36</i>
<i>Veza binarnog, oktalnog i heksadecimalnog sistema.....</i>	<i>37</i>
<i>Prevođenje brojeva pomoću Windows kalkulatora.....</i>	<i>38</i>
06 Predstavljanje podataka na računaru.....	39
<i>Predstavljanje brojeva.....</i>	<i>40</i>
<i>Predstavljanje teksta.....</i>	<i>42</i>
<i>Predstavljanje grafike.....</i>	<i>44</i>
<i>Predstavljanje zvuka.....</i>	<i>45</i>
07 Operativni sistem.....	46
<i>Pojam i struktura operativnog sistema.....</i>	<i>46</i>
<i>Tipovi operativnih sistema.....</i>	<i>51</i>
<i>Moderni operativni sistemi.....</i>	<i>51</i>

08 Rad sa fajlovima.....	54
<i>Fajlovi i direktorijumi.....</i>	<i>54</i>
<i>Arhive i kompresija.....</i>	<i>55</i>
<i>Tipovi fajlova.....</i>	<i>57</i>
09 Programiranje i programski jezici.....	59
<i>Algoritam.....</i>	<i>59</i>
<i>Razvoj softvera.....</i>	<i>60</i>
<i>Programski jezici.....</i>	<i>61</i>
10 Aplikativni softver.....	66
<i>Tehnika.....</i>	<i>66</i>
<i>Poslovanje.....</i>	<i>67</i>
<i>Nauka i obrazovanje.....</i>	<i>69</i>
<i>Računarske igre.....</i>	<i>69</i>
11 Baze podataka i informacijski sistemi.....	72
<i>Baze podataka.....</i>	<i>72</i>
<i>Informacijski sistemi.....</i>	<i>76</i>
12 Grafika, animacija i multimedija.....	78
<i>Rasterska i vektorska grafika.....</i>	<i>78</i>
<i>3D grafika i animacija.....</i>	<i>79</i>
<i>Poslovna grafika.....</i>	<i>81</i>
<i>Multimedija.....</i>	<i>81</i>
13 Komunikacija i Internet.....	84
<i>Internet kao mreža.....</i>	<i>84</i>
<i>Internet kao medij.....</i>	<i>85</i>
<i>Servisi Interneta.....</i>	<i>86</i>
<i>Netiketa.....</i>	<i>90</i>
14 Bezbednost računarskog sistema.....	92
<i>Maliciozni softver.....</i>	<i>92</i>
<i>Hakerski napadi.....</i>	<i>93</i>
<i>Zaštita.....</i>	<i>97</i>
15 Legalnost softvera i piraterija.....	101
<i>Softverske licence.....</i>	<i>101</i>
<i>Piraterija.....</i>	<i>103</i>
Literatura.....	105

Računar (*computer*) je mašina koja, kao što joj i ime sugeriše, suštinski služi za računanje. Međutim, računar nije samo običan kalkulator, već je u neku ruku univerzalna mašina, budući da funkcioniše tako što izvršava neki unapred pripremljen **program**. Tako se računar može „programirati“ da uradi bilo kakav niz matematičkih operacija (npr. da rešava kvadratne jednačine).

Taj program i podaci koji se zadaju, kao i rezultati koji se dobijaju, moraju da se nalaze na nekom dostupnom mestu, odnosno **memoriji računara**. Deo računara koji radi sa memorijom, izvršava program i obavlja sve zadate aritmetičke operacije, naziva se **procesor**.

Kibernetika, informatika i računarstvo

U današnje vreme se navedeni pojmovi često poistovećuju, tako da će biti ukratko objašnjeni.

Kibernetika (reč potiče od grčke reči *κυβερνητης*, što znači navigacija) je nauka koja se bavi upravljanjem, ali ne u smislu u kome se danas proučavaju razni vidovi menadžmenta, već u smislu teorije sistema – na koji način elementi koji čine sistem funkcionišu kao celina. Kao takva predstavlja teorijsku osnovu za razvoj informatike i računarstva.

Informatika (na engleskom govornom području koristi se naziv *Informatics*) je **nauka** koja se bavi proučavanjem informacionih sistema, ili, bolje reći, čuvanjem, prenosom, obradom podataka i njihovom transformacijom u informacije.

Ono što bi trebalo imati u vidu jeste da informatika kao takva nije obavezno povezana sa kompjuterima. Sasvim je moguće kreirati informacioni sistem zasnovan na kartotekama, kurirskoj službi, ručnoj obradi podataka i sl. Ipak, računari i računarske mreže su idealni za kreiranje informacionih sistema, što objašnjava zbog čega se informatika često poistovećuje sa računarstvom.

Računarstvo (na engleskom govornom području koristi se naziv *Computer Science*) predstavlja skup veština i tehnologija, vezanih za konstruisanje, programiranje i upotrebu računara.

Analogno i digitalno

Ova dva termina odnose se na način beleženja podataka, odnosno predstavljanja nekog signala. Termin „analogno“ označava neprekidan signal, odnosno način na koji tumačimo svet oko sebe – zvučni talasi, elektromagnetno zračenje. Digitalni signal predstavlja niz nula i jedinica.

Nekadašnji fotoaparati, koji su koristili filmove i negative, koristili su optiku i hemijske reakcije kako bi se slika zabeležila, predstavljaju primer analogne tehnologije. Današnji digitalni fotoaparati optikom prenose sliku do foto-osetljivog elementa (CCD – *Charge-Coupled Device* senzora), koji suštinski digitalizuje sliku (svetlost pretvara u niz brojeva).

Najjednostavniji način shvatanja razlike između digitalnog i analognog jeste da u digitalnom svetu postoji jasna granica između crnog i belog, dok u analognom uvek postoje sive nijanse.

Analogni računari?!

Ma koliko to u današnje vreme zvučalo neverovatno, postojali su i analogni računari. Odličan primer bi bio računar **MONIAC** (*Monetary National Income Analogue Computer*), kreiran 1949. godine.

U pitanju je bio hidraulični računar koji je služio za predstavljanje tokova novca u britanskoj ekonomiji. U sebi je imao providne rezervoare povezane cevima. Svaki rezervoar predstavljao je jedan aspekt ekonomije (štednja, investicije...) i kako su se neki rezervoari punili, drugi su se praznili.



U vreme kada je MONIAC konstruisan, nisu bili dostupni računari koji su mogli da vrše kompleksne ekonomske simulacije.

Prednost digitalnog signala je što se njegov kvalitet ne gubi prenosom. Kada se video presnimava sa VHS kasete na kasetu, kvalitet snimka postaje sve gori i gori svakim sledećim presnimavanjem. Sa druge strane, svaka kopija digitalnog videa sa jednog DVD-a na drugi uvek je istovetna originalu.

Kratka istorija razvoja računara

Razvoju računara prethodio je dug razvoj matematike, logike, fizike i tehnologije. Nekoliko važnih događaja i velikih ljudi je kroz istoriju utrlo put stvaranju računara kakve danas poznajemo.[1]

2600. p.n.e. U **Kini** je napravljena prva mašina za pomoć u računanju – **abakus**.

300. p.n.e. **Euklid** (*Ευκλείδης*) sažima celokupno znanje matematike i geometrije starih Grka u jednu knjigu „**Elementi**“.

1500. **Leonardo da Vinči** (*Leonardo da Vinci*) osmišljava tzv. „**mašinu za sabiranje**“.

1614. **Džon Nepijer** (*John Napier of Merchiston*) opisuje prirodu logaritama i konstruiše verziju abakusa za množenje, deljenje i korenovanje pod nazivom „**Nepijerove kosti**“.

1621. **Vilijam Otred** (*William Oughtred*) na osnovu Nepijerovih proučavanja konstruiše **Logaritmar** – abakus za računanje logaritama koji se kasnije koristio preko 300 godina.

Logaritmi služe za računanje eksponenta stepenovanja nekog broja, npr.

ako je $2^3 = 8$ onda je $\log_2(8) = 3$

Logaritmi kao takvi su teški za izračunavanje i pojavljivanje mašine koji ih računa predstavlja veliki korak. Logaritme ne treba mešati sa algoritmima.

-
- 1623.** **Vilhelm Šikard** (*Wilhelm Schickard*) pronalazi **Računski sat**, prvi mehanički kalkulator, zasnovan na ideji Nepijerovih kostiju.
-
- 1642.** **Blez Paskal** (*Blaise Pascal*) konstruiše aritmetičku mašinu **Paskalinu**, koja može da sabira i oduzima. U njegovu čast je programski jezik Pascal dobio ime.[2]
-
- 1670.** **Gotfrid Lajbnic** (*Gottfried Wilhelm Leibniz*) poboljšava Paskalinu i dodaje joj mogućnosti množenja, deljenja i korenovanja.
1679. Lajbnic osmišljava aritmetiku sa **binarnim brojevima**. Takođe dokazuje da se svaki broj može predstaviti samo korišćenjem cifara 0 i 1.
-
- 1804.** **Žoze-Mari Žakar** (*Joseph-Maria Jacquard*) **programira razboj za tkanje** koristeći bušene kartice za kreiranje šara na tkanini.
- Ovaj izum se smatra začetkom programiranja. Takođe, izazvao je i **prvi talas straha** od automatizacije. 1811. **Ned Lud** (*Ned Ludd*) je poveo tkače, koji su strahovali da će ostati bez posla, u napad na mašine. Programeri se i dan-danas u većoj ili manjoj meri suočavaju sa ljudima koji se iz straha ili neznanja protive uvođenju informacionih tehnologija.
-
- 1822.** **Čarls Bebidž** (*Charles Babbage*), matematičar i profesor na Kembridžu, konstruiše **Diferencijalnu mašinu** – mehanički kalkulator koji je mogao da sabira i oduzima.
1830. Bebidž projektuje **Analitičku mašinu**, na kojoj radi sledećih četrdeset godina, do kraja života, ali ne uspeva da je završi. To je trebalo da bude prvi mehanički računar koji je mogao da se programira.[2]
-
- 1843.** **Ada Avgusta King** (*Augusta Ada King*), ćerka pesnika Bajrona, kreira **program** koji je trebalo da se izvršava na Analitičkoj mašini. Da je mašina ikada konstruisana, program bi izračunavao Bernulijev niz brojeva.
- Ada Avgusta je ostala upamćena kao prvi programer u istoriji računara. U njenu čast jedan programski jezik nosi ime ADA.
-
- 1854.** **Džorž Bul** (*George Boole*) razvija algebru nad binarnim brojevima i logičke operacije sa logičkim vrednostima – konjunkciju, disjunkciju, negaciju.
- U njegovu čast, ove operacije nose naziv **Bulova algebra**.
-
- 1890.** **Herman Holerit** (*Herman Hollerith*) kreira električni tabelarni sistem za Popisni zavod SAD. Ova mašina je koristila bušene kartice za tabeliranje popisa i posao, koji se ručno radio osam godina, skraćivao na samo godinu dana.[2]
1896. Holerit osniva firmu *Computing Tabulating Recording Machine Company*, koja 1924. prerasta u **International Business Machines – IBM**.
-
- 1903.** **Nikola Tesla** patentira električna logička kola pod imenom „prekidači“.
-
- 1904.** **Džon Ambroz Fleming** (*John Ambrose Fleming*) razvija vakuumske cevi. Iako ih je **Tomas Edison** već ranije otkrio, odbacio ih je kao beskorisne. Vakuumske cevi predstavljaju prekretnicu u razvoju računara – do tada su računari bili mehanički, izračunavanja su se obavljala pomoću zupčanika i prekidača. Vakuumska cev je služila kao prekidač koji se uključuje i isključuje hiljadama puta brže.
-

1926.	Patent za poluprovodnički tranzistor omogućuje da električna struja počne da prenosi podatke kroz računar.
1936.	Konrad Zuse (<i>Konrad Zuse</i>) pravi digitalnu računsku mašinu koja uvodi primenu binarnog sistema i elektronskih cevi i koja se može programirati.[3]
1936.	Alan Tjuring (<i>Alan Turing</i>) objavljuje rad u kome dokazuje da se svaka izračunljiva matematička funkcija (drugim rečima svaki niz operacija) može predstaviti u obliku algoritma za teorijski aparat poznat kao Tjuringova mašina – koncept koji predstavlja osnovu teorije algoritama.[2] Tjuring se takođe bavio kriptografijom i tokom II svetskog rata saradivao sa britanskom obaveštajnom službom na razbijanju nemačkih šifrovanih poruka. 1950. definiše Tjuringov test kojim utvrđuje standard za proveru da li se neka mašina može nazvati „inteligentnom“. Ovaj test je i danas aktuelan u oblasti veštačke inteligencije.
1943.	Britanci prave mašinu Kolos koja služi za razbijanje nemačkih šifara (nacisti su u to vreme imali mašinu za šifrovanje po imenu Enigma).
1944.	Vojska SAD konstruiše računar ENIAC (<i>Electronic Numerical Integrator Analyzer and Computer</i>) za izračunavanje putanja projektila. ENIAC je obavljao 5000 sabiranja u sekundi, težio je 30 tona, bio dugačak 30m i visok 3m. Koristio je vakuumske cevi, trošio veliku količinu struje i veoma se zagrevao.[4] U isto vreme, Harvardski univerzitet i IBM razvijaju računar MARK 1 koji koristi bušene kartice za unos podataka i programa.
1945.	Džon fon Nojman (<i>John von Neumann</i>) definiše elektronski digitalni računar opšte namene koji izvršava unutrašnji program – koncept koji i dan-danas opisuje računar.
1945.	Otkrivena prva računarska „buba“ (<i>eng. bug</i>). U pitanju je bio moljac koji se uhvatio u računar. Danas je uobičajeno da se greške u programima nazivaju „bagovima“.
1948.	Vilijam Šokli (<i>William Bradford Shockley</i>) iz Belovih laboratorija, zajedno sa saradnicima patentira tranzistor . Nekoliko godina kasnije, to će im doneti Nobelovu nagradu. Tranzistor je takođe prekidač, ali na mikroskopskom nivou – daleko manji od vakuumske cevi, pa samim tim troši mnogo manje struje, manje se zagreva, zauzima mnogo manje mesta. Procesori su i danas suštinski zasnovani na ovoj tehnologiji.
1951.	Naučnici koji su kreirali ENIAC, Popisnom zavodu SAD isporučuju novi računar UNIVAC , koji koristi magnetne trake za beleženje podataka.
1954.	Kompanija Texas Instruments počinje proizvodnju silicijumskih tranzistora . Dve godine kasnije, na MIT-u je konstruisan TX-O , prvi računar na bazi tranzistora.
1957.	IBM kreira prvi hard disk kao deo uređaja RAMAC 350. Ovaj uređaj se sastojao od 50 diskova prečnika 24" i težio jednu tonu. Kapacitet mu je iznosio 5 megabajta.
1958.	Džek Kilbi (<i>Jack S. Kilby</i>), iz kompanije <i>Texas Instruments</i> , pravi prvo integrisano kolo .
1959.	Robert Nojs (<i>Robert Noyce</i>), iz kompanije <i>Fairchild Semiconductors</i> , patentira integrisano kolo čije su komponente povezane aluminijumskim linijama na silicijumskoj podlozi. 1961. ova kompanija proizvodi prvo komercijalno integrisano kolo .

1960.	Kompanija <i>Digital Equipment Corporation</i> – DEC predstavlja svoj prvi računar PDP-1 , po ceni od \$120 000.
1963.	Kreiran ASCII kod , kojim je standardizovano predstavljanje teksta na računarima.
1968.	Robert Nojs i Gordon Mur (<i>Gordon E. Moore</i>) napuštaju kompaniju <i>Fairchild Semiconductors</i> i osnivaju kompaniju Intel Corporation u Kaliforniji. Godinu dana kasnije osniva se i konkurentska kompanija Advanced Micro Devices – AMD .
1971.	Inženjeri Intela projektuju prvi mikroprocesor pod nazivom 4004 . Ovo je 4-bitni procesor i radi na 108 KHz. Sastoji se od 2300 tranzistora i može da obavi 60.000 operacija u sekundi. Cena mu iznosi samo 200 dolara.
1971.	IBM predstavlja novi medij za snimanje podataka – disketu (<i>floppy disc</i>) od 8".
1972.	Pojavljuje se prvi kompakt disk .
Jedna od najvažnijih prekretnica u razvoju računara bio je trenutak kada su kompjuteri postali dovoljno mali i jeftini da ih može priuštiti i pojedinac. Tada počinje era personalnih računara.	
Računar sam po sebi ne vredi bez programa koji ga pokreću. Paralelno sa razvojem kompjutera razvijali su se i programski jezici, a sa njima i programi koji su postavili kamen temeljac softveru koji danas koristimo.	
1956.	IBM razvija programski jezik FORTRAN (<i>Formula Translation</i>) kao alternativu programiranju u mašinskom jeziku. Ovaj jezik je pre svega bio namenjen izvršavanju matematičkih proračuna.
1960.	Nastaje i jezik COBOL (<i>Common Business Oriented Language</i>), namenjen poslovnim potrebama. Iste godine nastaje i jezik LISP , kao prvi jezik za programiranje veštačke inteligencije .
1961.	Tri studenta sa MIT-a pišu igru Spacewar , koja se smatra prvom računarskom igrom . Igra je napisana za računar DEC PDP-1.
1964.	Džon Kemeni (<i>John Kemeny</i>) i Tomas Kurc (<i>Thomas Kurtz</i>) razvijaju programski jezik BASIC (<i>Beginners All-Purpose Symbolic Instruction Code</i>).
1967.	Na Kembridžu nastaje programski jezik LOGO , kao jezik namenjen deci.
1969.	Ken Tompson (<i>Ken Thompson</i>), iz Belovih laboratorija, piše prvu verziju operativnog sistema UNIX .
1969.	Bil Gejts (<i>Bill Gates</i>) i Pol Alen (<i>Paul Allen</i>), koji tada sebe nazivaju <i>Lakeside Programming Group</i> , započinju svoj prvi profesionalni posao – pronalaženje grešaka u softveru računara PDP-10.
1972.	Intel proizvodi procesor 8008 – prvi 8-bitni procesor. 1974. Intel kreira procesor 8080.
1973.	Denis Riči (<i>Dennis Ritchie</i>), iz Belovih laboratorija, kreira programski jezik C , prvenstveno namenjen sistemskom programiranju za UNIX. Jezik C postaje verovatno najpopularniji programski jezik, kojim se pišu aplikacije svih namena. Veliki broj jezika danas vuče poreklo iz ovog jezika (C++, C#, Java, PHP...)
1974.	Pojavljuje se računar Altair 8800 sa Intelovim procesorom 8008. 1975. U časopisu Popular Electronics pojavljuje se članak o računaru Altair 8800 – prvom personalnom računaru . Ovo se smatra početkom ere personalnih računara.
1976.	Stiven Džobs (<i>Steven Jobs</i>) i Stiv Vozniak (<i>Steve Wozniak</i>) kreiraju računar Apple , koji je

	imao tastaturu i prikazivao sliku na TV ekranu. Računar prikazuju u Klubu računara kućne izrade (<i>Home Brew Computer Club</i>), koji kasnije postaje poznat kao Silicijumska dolina .
1976.	Bil Gejts i Pol Alen osnivaju Microsoft .
1977.	Apple Computer Company predstavlja računar Apple II .
1977.	Commodore predstavlja svoj prvi model računara PET 2001, sa procesorom 6502 na radnom taktu od 1MHz i sa 4KB RAM-a.
1978.	Intel proizvodi svoj prvi 16-bitni procesor 8086 .
1978.	Pojavljuje se WordStar – program za obradu teksta za sistem CP/M.
1979.	Pojavljuje se prvi program za tabelarne kalkulacije VisiCalc za računar Apple II. Ovaj program je predstavljao malu revoluciju, s obzirom na to da je računare približio poslovnim ljudima i menadžerima koji više nisu morali da znaju programiranje da bi na računaru izvršili proračune.
1981.	IBM predstavlja „ personalni računar “ (PC) zasnovan na Intelovom procesoru 8086. Ono što je dovelo do eksplozije širenja PC-ja je odluka IBM-a da zasnuje računar na otvorenoj arhitekturi i dozvoli konkurentskim firmama da prave PC kompatibilne računare, sve dok ne kopiraju IBM-ov originalni BIOS.
1981.	Microsoft kreira operativni sistem DOS (<i>Disc Operating System</i>).
1981.	Osborne Computer Corporation predstavlja Osborne 1 , portabilni računar koji se nalazio u koferu i napajao se baterijom. Imao je ekran od 5", 64KB memorije i procesor Z80. Bio je težak oko 12kg. Ovaj kompjuter se smatra pretečom današnjih laptop računara . [5]
1982.	Intel kreira procesor 80186 i procesor 80286 . U pitanju su 16-bitni procesori, s tim što je procesor 286 prvi koji je „kompatibilan unazad“, odnosno mogao je da izvršava programe svojih prethodnika. Od tog modela ovo postaje Intelova praksa.
1982.	Sinclair i Commodore izbacuju svoje najuspešnije modele kućnih računara ZX Spectrum (procesor Z80 na 3.5MHz) i C64 (procesor 6510 na 1MHz). Ova dva osmobicna računara služila su za učenje ali i zabavu, zahvaljujući velikom broju igara. Sličnih performansi (oba sa po 64KB ukupne memorije), ušli su u mnoge domove i obeležili čitav jedan period u istoriji računara.
1983.	Voja Antić kreira računar Galaksija . Ovaj računar prvenstveno nije bio zamišljen kao komercijalni kompjuter koji bi se kupovao u prodavnici, već učilo za entuzijaste koji su želeli da sami sklope svoj računar i nauče programiranje. Bio je to projekat iza koga su stajale ideje slobodnog softvera i razmene znanja. Uputstva za sklapanje objavljena su u specijalnom izdanju časopisa „Galaksija“ pod nazivom „Računari u vašoj kući“, koje je kasnije preraslo u samostalni časopis „Računari“. Možda je upravo zbog svojih skromnih mogućnosti u odnosu na tadašnje kućne računare (memorija i grafika) inspirisao tadašnje programere da iz njega izvuku maksimum mogućnosti.
1983.	Microsoft objavljuje prvu verziju Windows-a – grafičkog proširenja za MS-DOS.
1984.	Kompanija Apple predstavlja računar Macintosh , koji koristi miš i grafički korisnički interfejs.
1985.	Intel proizvodi 32-bitni procesor 80386 koji se sastoji iz 275.000 tranzistora. Radio je na

	16MHz i postizao 6 MIPS (miliona operacija u sekundi).
1985.	Pojavljaju se prvi CD-ROM uređaji za PC računare.
1986.	Atari izbacuje računare Atari 520 i Atari 1040 (16-bitni procesor Motorola MC68000 na 8MHz, 512KB / 1MB, 3.5" disk jedinica). 1987. Commodore izbacuje računar Amiga 500 (skoro potpuno istih karakteristika kao Atarijev računar). Zahvaljujući dobroj grafici i odličnom zvuku, Amiga 500 stiće popularnost kakvu je nekada imao C64. Ova dva računara predstavljaju poslednje popularne modele tzv. kućnih računara, koji su bili zastupljeni u velikom broju domova. Sa padom cena, bržim procesorima i povećanjem multimedijjskih sposobnosti, PC računari polako preuzimaju primat ne samo kao poslovne mašine, već i kao računari za zabavu. Iako su obe kompanije nastavile sa konstruisanjem novih modela, nijedan nije postigao uspeh kao ova dva. Devedesetih godina, kompanija Atari je prestala da se bavi proizvodnjom računara, a Commodore je bankrotirao.
1989.	Intel proizvodi 32-bitni procesor 80486DX – prvi model koji je imao ugrađen matematički koprosesor. Prva verzija ovog procesora radila je na 25MHz i postizala 20 MIPS.
1990.	Microsoft objavljuje Windows 3.0 – prvu zapaženu verziju ovog operativnog sistema.
1991.	Linus Torvalds (<i>Linus Torvalds</i>) kreira operativni sistem Linux baziran na UNIX-u.
1992.	Apple predstavlja Newton – jedan od prvih digitalnih podsetnika (PDA – Personal Digital Assistant). Suštinski, ovo je namenski računar koji staje na dlan. Komande su se zadavale posebnom olovkom preko ekrana osetljivog na dodir. Newton je bio posebno popularan zbog svoje mogućnosti da „razume“ ručno napisan tekst.[6]
1993.	Intel proizvodi procesor Pentium koji je radio na taktu od 66MHz i imao preko 3 miliona tranzistora. Procesori postaju kompleksniji sa sve većim brojem tranzistora. Smanjuju se potrošnja električne energije i zagrevanje, a samim tim omogućeno je povećanje radnog takta.
2000.	PaceBlade proizvodi PaceBook – prvi tablet PC računar koji je radio pod modifikovanim Windows operativnim sistemom.[7]
2007.	Asus predstavlja Eee PC 700 tzv. „ netbook “ – prenosni računar malih dimenzija i skromnih mogućnosti koji je prvenstveno bio namenjen korišćenju Interneta. U isto vreme postaje popularan koncept „ rada u oblaku “ (<i>cloud computing</i>), koji softver ne tretira kao proizvod već kao uslugu kojoj se pristupa preko web-a. Iako sam koncept postoji kao ideja još od 50-ih godina, njegova realizacija je morala sačekati odgovarajući tehnološki razvoj.[9]
2010.	Apple predstavlja iPad . Iako su tablet računari već postojali, ovo je bio prvi koji je stekao planetarnu popularnost i otvorio vrata poplavi sličnih uređaja.[8]

Uočava se da, sa napretkom tehnologije, računari postaju sve manji i brži, kapaciteti memorije sve veći, dok cene stalno padaju. Računar koji je nekada zauzimao čitave prostorije i težio nekoliko tona, po mogućnostima nije ni blizu današnjeg „pametnog“ mobilnog telefona koji teži stotinak grama i staje na dlan.

Računar se već dosta dugo definiše kao sistem koji se sastoji iz procesora i memorije povezanih magistralom podataka. Sa druge strane, **računarski sistem** se definiše kao računar povezan sa perifernim uređajima. Ipak, kada se govori o kompjuterima koje koristimo u svakodnevnom životu, obično se za ceo računarski sistem jednostavno koristi termin „računar“.

Matična ploča

Matična ploča je centralni deo računara. Ona povezuje procesor, memoriju i ulazno-izlazne uređaje. Ona sadrži i ROM memoriju, u kojoj se kod PC računara nalazi BIOS (osnovni programi za rad računara), kao i upravljačke čipove (tzv. kontrolere) koji upravljaju radom perifernih uređaja (npr. hard diska, CD-ROM-a...). Matična ploča takođe povezuje računarski sistem sa električnim napajanjem.

Sa stanovišta arhitekture, povezivanje delova mikroracunarskog sistema obavlja se preko tri **magistrale** (put kojim se kreću signali, podaci, naredbe u računaru):

- **magistrala podataka** – služi za prenos podataka kroz računar;
- **adresna magistrala** – magistrala kojom se predaju adrese memorijskih lokacija;
- **upravljačka magistrala** – magistrala kojom se prenose upravljački signali.

Sa razvojem računara, menjali su se i izgled i fizička konstrukcija matične ploče. Tako nove matične ploče imaju drugačiju arhitekturu, drugačije slotove za povezivanje kartica (npr. grafičke karte), spoljne memorije (IDE, SATA), drugačije portove za povezivanje perifernih uređaja (npr. USB, FireWire...).

Konačno, matična ploča je ta od koje zavisi radni takt na kome će raditi procesor i memorija, kao i brzina protoka informacija.

Čipset je skup čipova, odnosno **kontrolera** iz kojih se sastoji matična ploča. Kod današnjih PC računara deli se na dva dela:

- „**severni**“ **deo** matične ploče (*northbridge*) – veoma kompleksan i brz kontroler koji povezuje procesor sa memorijom i magistralom;
- „**južni**“ **deo** matične ploče (*southbridge*) – kontroler koji povezuje magistralu sa perifernim uređajima (disketne jedinice, hard diskovi, CD i DVD ROM-ovi, miš, tastatura, USB...).

Pri kupovini računara obično vodimo računa da matična ploča, procesor i memorija budu dobro usklađeni.

Procesor

Procesor (CPU – *Central Processing Unit*) je centralni deo svakog računara. Da bi računar radio potreban mu je program, a upravo je procesor taj koji izvršava komande zadate u programu. Prvi mikroprocesor proizvela je firma Intel 1971. godine. Bio je to 4-bitni procesor **Intel 4004**. Imao je 2300 tranzistora i radio na radnom taktu od 108 KHz.

Suštinski, procesor se nije promenio već decenijama. Ako se pitate da li bi program sličan Word-u ili 3D kompjuterskoj igri mogao da se napravi za nekadašnje procesore kućnih računara, kakvi su bili *Commodore 64* ili *ZX Spectrum*, odgovor je – DA¹, ali bi kreiranje jedne jedine slike iz neke od modernih igara na takvom procesoru verovatno trajalo mesecima.

Osnovna karakteristika procesora, koja interesuje kupca, jeste njegova **brzina**, odnosno koliko brzo može izvršavati instrukcije zadate u programu. Što je procesor brži, to su kompleksniji programi koje možemo pokrenuti na računaru. Brzina zavisi od nekoliko faktora. Kako su se procesori razvijali, tako su se i ovi faktori menjali. Neki od osnovnih su:

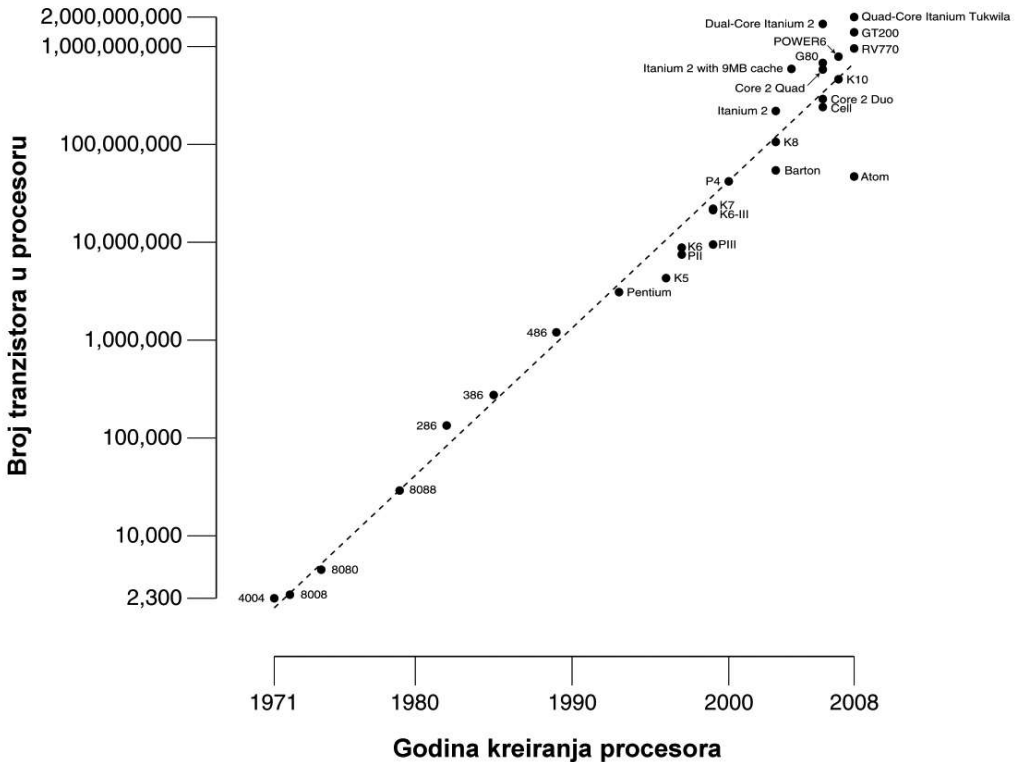
- **arhitektura procesora** – noviji procesor sa boljim konstruktorskim rešenjima i više tranzistora će svakako biti brži (Intel Core 2 je brži od starijeg Pentiuma 4, čak iako rade na istom radnom taktu);
- **radni takt** – takt po kome radi procesor, što je veći takt procesor će brže raditi (Pentium 4 na 2GHz je brži od istog procesora na 1,6GHz);
- **širina procesorske reči** – broj bitova sa kojim procesor može da radi u jednom trenutku – što je veći, procesor će raditi brže (64-bitni procesori su brži od 32-bitnih);
- **prisustvo i veličina keša (cache)** – priručna memorija procesora – velike brzine, veća količina keša znači i veću brzinu procesora.

Konstrukcija i tehnologija izrade je svakako veoma značajan faktor brzine procesora, pogotovu kada se uzme u obzir **Murov zakon**, koji kaže da se procesorska snaga udvostručuje na svake dve godine. Iako je suosnivač Intela, Gordon Mur (*Gordon E. Moore*), ovu tvrdnju izneo kao predviđanje, danas se proizvođači procesora trude da održe korak i da je zaista ostvare.

Tokom godina konstruisanja, razvile su se dve različite „filozofije“ pravca razvoja procesora. Prva ideja je da procesore treba usavršavati tako što im se dodaje sve veći broj naredbi koje mogu da izvrše. Tako nastaju tzv. procesori kompleksnog seta naredbi (**CISC** – *Complex Instruction Set Computer*). Sa druge strane nalaze se procesori redukovanoog seta naredbi (**RISC** – *Reduced Instruction Set Computer*), kod kojih je ideja da procesor treba da poznaje mali broj naredbi, ali koje se izvršavaju veoma brzo. Intel i AMD, koji proizvode x86 procesore kakve koristimo u našim kućnim kompjuterima, slede CISC filozofiju, dok RISC procesore kreiraju Sun Microsystems (SPARC procesor), MIPS (procesor za PlayStation konzolu), IBM (PowerPC procesori). Danas se razlika između ove dve filozofije praktično izgubila, pošto i x86 procesori implementiraju RISC elemente (npr. interno razbijanje instrukcija na mikrooperacije).

Moderni procesori danas u sebi imaju ugrađenu jedinicu za računanje sa decimalnim brojevima (**FPU** – *Floating Point Unit*), mehanizme predviđanja grananja u programu (*branch prediction*), imaju mogućnost paralelnog izvršavanja više instrukcija istovremeno (*instruction pipelining*), imaju dva ili više procesorskih jezgara, rade na višoj frekvenciji u odnosu na ostatak računara itd.[12]

¹ Naravno samo teorijski, pošto takvi računari ne bi imali dovoljno memorije, niti mogućnosti da prikažu grafiku visoke rezolucije sa velikim brojem boja.



Slika 2.1. Broj tranzistora u procesoru kao ilustracija Murovog zakona [54]

Tokom godina se stalno poboljšavala i tehnologija izrade procesora, koja se danas sprovodi tzv. foto-litografijom, metodom koja omogućava smeštanje desetina miliona tranzistora na glavu čiode. Veličina integrisanih kola u procesoru meri se nanometrima (prvi procesor, Intel 4004, bio je proizveden u tehnologiji koja je dozvoljavala veličinu kola od danas ogromnih 10 mikrona (mikron je hiljaditi deo milimetra, a nanometar hiljaditi deo mikrona). Ovako mali elementi omogućuju da procesori zahtevaju manje električnog napajanja i manje zagrevanje, zahvaljujući čemu se ostvaruju viši radni taktovi.[11]

Radni takt predstavlja „ritam“ po kome radi procesor. Intelov 4004 procesor je radio na 108KHz, ZX Spectrum na 3.5MHz, PC sa 386 procesorom je radio na 16MHz, prvi Pentium na 66MHz, dok današnji procesori rade na 3 ili više GHz. Što je radni takt veći, veća je i brzina procesora. **Overklokning (overclocking)** predstavlja povećanje radnog takta na vrednost koja nije inicijalno predviđena za procesor. Loša strana preteranog povećavanja radnog takta je pregrevanje procesora, što uzrokuje smanjenje stabilnosti računara, pojavu grešaka i blokiranja pri radu, kao i skraćenje njegovog radnog veka.

Širina procesorske reči predstavlja broj bitova koje procesor može da prihvati u jednom trenutku sa magistrale podataka (iz memorije). Prema tom broju, razlikujemo 8-bitne, 16-bitne, 32-bitne ili 64-bitne procesore. Veći broj bitova znači da procesor može da radi sa većom količinom podataka. Na primer, ukoliko bi bilo potrebno izvršiti neku operaciju na podatku koji se sastoji od 4 bajta (32 bita), 8-bitni procesor bi mogao da radi samo sa pojedinačnim bajtovima, što bi zahtevalo više operacija i vremena, dok bi 32-bitni procesor to mogao da obavi

znatno efikasnije. Druga bitna veličina vezana za broj bitova procesora jeste **širina adresne magistrale** koja, u stvari, označava količinu memorije kojoj procesor može da pristupi. Širina od 16 bitova znači da procesor može da adresira samo 64KB memorije, dok širina od 32 bita označava mogućnost adresiranja 4GB memorije. Širina adresne magistrale i širina procesorske reči ne moraju biti jednake.

Keš predstavlja „priručnu“ memoriju koja služi da se premosti razlika u brzini između brzog procesora i sporije RAM memorije. Ova memorija je veoma brza, ali i skupa, zbog čega je njen kapacitet mnogo manji od kapaciteta osnovne memorije. Umesto da jedne iste podatke stalno čita i beleži u RAM memoriji, procesor može raditi sa keš memorijom i tako uštedeti vreme. Keš se deli na interni L1 keš i eksterni L2 keš. L1 keš je brži, veoma malog kapaciteta i nalazi se u samom procesoru, dok se L2 keš ugrađuje kao izdvojena komponenta, ima veći kapacitet od internog keša ali i veće vreme pristupa. Veličine keš memorije se povećavaju u novim procesorima, pa je tako za Intel Core 2 familiju procesora kapacitet L1 keša u rangu 64KB (za svako procesorsko jezgro), a L2 keša 6MB.

Velika prevara!

Danas se kao mera brzine procesora najčešće koristi upravo radni takt procesora. Treba da znamo da je radni takt (odnosno frekvencija) samo jedan od faktora koji utiču na brzinu, nikako mera brzine.

Marketinški stručnjaci su svesni da potrošači radni takt zaista poistovećuju sa brzinom procesora, zbog čega se firme koje proizvode procesore utrkuju ne bi li kreirale procesor koji će raditi na višoj frekvenciji (poznato je svojevremeno nadmetanje između Intela i AMD-a oko dostizanja takta od 1GHz).

Ovo je takođe i razlog zbog koga su AMD-ovi procesori, npr. serije Athlon XP, imali relativne oznake, a ne stvarne radne taktove. Athlon XP 2000+ je u stvari radio na 1,67GHz, ali je njegova oznaka implicirala da se po brzini može meriti sa Intelovim Pentiumom 4 na 2GHz.

Mera brzine procesora je **MIPS** (*Million Instructions Per Second*) – broj operacija koje procesor može da izvrši u jednoj sekundi. Ipak, ni MIPS nije idealna mera, budući da su rezultati neuporedivi za različite procesorske arhitekture.

Ne zaboravite da je procesor ipak samo jedan deo računara. Brzina celog sistema u stvari uvek zavisi od najsporije komponente koja predstavlja „usko grlo“ pri radu kompjutera.

Procesor se sastoji iz nekoliko delova, od kojih su osnovni:

- upravljačka jedinica (*CU – Control Unit*),
- aritmetičko-logička jedinica (*ALU – Arithmetic Logic Unit*) i
- registri.

Upravljačka jedinica je deo koji kontroliše rad računara. Ona vrši dekodiranje operacija u programu, upravlja izvršenjem programa, memorijom i ostalim delovima računarskog sistema. Sa razvojem računara ide se ka tome da procesor bude što više rasterećen ovih funkcija, pa tako posebni kontroleri povezani na matičnu ploču preuzimaju deo ovih poslova.

Aritmetičko-logička jedinica jeste deo procesora koji obavlja aritmetičke i logičke operacije nad celim brojevima i pojedinim bitovima. Kao dodatak, procesori danas poseduju i jedinicu za računanje sa brojevima u pokretnom zarezu.

Registri su posebne memorijske lokacije dostupne unutar samog procesora. U njima su smešteni podaci sa kojima procesor upravo radi. Postoje registri opšte namene:

- **registri podataka** – u njih se smeštaju podaci iz memorije sa kojima procesor radi;
- **adresni registri** – u njih se smeštaju memorijske adrese, kako bi procesor mogao da pročita neki podatak iz memorije ili da ga upiše nazad u memoriju.

Osim opštih, postoje i registri specijalne namene od kojih su najvažniji:

- **akumulator** – ovo je registar u koji se smeštaju rezultati svih operacija;
- **registar indikatora** – svaki bit ovog registra predstavlja „zastavicu“ koja se „podiže“ ako se prilikom izvršavanja operacije desi neki događaj, npr. ako je došlo do prekoračenja pri izračunavanju (*overflow flag*) ili ako je rezultat operacije bio nula (*zero flag*) itd.
- **brojač naredbi** – registar koji sadrži adresu sledeće operacije programa.

Osnovni skup naredbi je kod modernih procesora isti kao i kod starijih modela, i možemo videti da procesor zaista može da uradi veoma malo toga:

- aritmetičke operacije sa celim brojevima (povećavanje i smanjivanje za jedan, sabiranje, oduzimanje, množenje, deljenje, upoređivanje);
- logičke operacije (logičko I, ILI, negacija...);
- prebacivanje podataka (iz memorije u registre procesora, vraćanje u memoriju, prebacivanje sa jedne na drugu memorijsku lokaciju, rad sa stekom);
- rad sa bitovima (pomeranje bitova, postavljanje i proveravanje vrednosti bita);
- operacije kontrole toka programa (uslovni i bezuslovni skok na neku novu naredbu);
- operacije za rad sa indikatorima (*flags* – zastavice);
- operacije ulaza i izlaza.

Procesori kakve imamo u računarima kod kuće ili na poslu uglavnom su bazirani na međusobno kompatibilnim Intel ili AMD procesorima, koji danas izvršavaju i mnoge druge instrukcije vezane za rad sa decimalnim brojevima, grafiku, multimediju i sl. (FPU, MMX, SSE setovi naredbi).[10]

Unutrašnja memorija

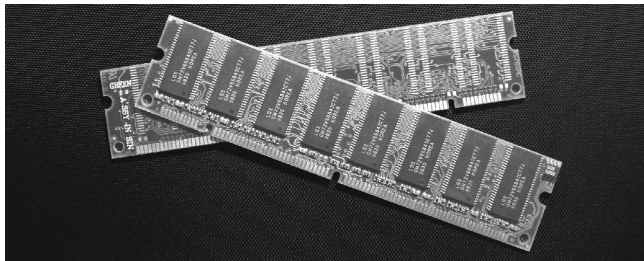
Unutrašnja ili osnovna memorija predstavlja deo računara u kome su pohranjeni podaci i programi sa kojima procesor radi. Prema načinu čuvanja podataka, unutrašnju memoriju možemo podeliti na:

- RAM (*Random Access Memory*)
- ROM (*Read-Only Memory*)
- PROM (*Programmable ROM*)
- EPROM (*Erasable PROM*)
- EEPROM (*Electronically Erasable PROM*)

RAM memorija, odnosno memorija slobodnog ili proizvoljnog pristupa dobila je naziv prema načinu na koji joj procesor pristupa – svakoj memorijskoj lokaciji se pristupa ravnopravno, za isto vreme. Da bi se neki program izvršio, mora se učitati u RAM. Danas je uobičajeno da se u RAM memoriju učitavaju i operativni sistem i aplikativni softver. Tu se uobičajeno nalaze i podaci sa kojima radimo na našem računaru (slike, tekstovi...).

Fizički, RAM memorija se sastoji iz velikog broja memorijskih ćelija od kojih svaka sadrži 1 bit informacije (da li je ćelija pod naponom ili ne). Svaka ćelija se sastoji od jednog kondenzatora i tranzistora. Kondenzator čuva podatak, dok tranzistor služi kao prekidač koji omogućuje da se sadržaj ćelije pročita ili izmeni. Kondenzator može da čuva podatak samo nekoliko milisekundi,

pa zbog toga procesor i kontroler memorije moraju neprekidno da napajaju one ćelije koje sadrže podatke.



Slika 2.2. Memorijski moduli

Posledica toga je da podaci u RAM memoriji ostaju zapisani samo dok postoji električno napajanje. Zbog toga se sve što radimo na računaru gubi pri nestanku struje ili isključivanju ukoliko nije bilo snimljeno. Kada se govori o količini memorije u računaru, misli se na RAM memoriju. Njena najvažnija karakteristika jeste **kapacitet**, a na drugom mestu **brzina**.

Ipak, brzina rada računara će prvenstveno zavisiti od količine RAM memorije. Kapacitet memorije se izražava u **bajtovima** (odnosno kilobajtima, megabajtima, gigabajtima...). Više o jedinicama mere možete pročitati u poglavlju o predstavljanju podataka na računaru. Ukoliko memorije nema dovoljno, operativni sistem mora koristiti tzv. **virtuelnu memoriju**, što znači da se podaci iz RAM-a stalno moraju snimati i učitavati sa medija spoljne memorije (tačnije hard diska), što znatno usporava rad računara. Ukoliko memorije ima dovoljno, postavlja se pitanje njene brzine, koja zavisi od radnog takta matične ploče. Iako je RAM memorija uvek sporija od procesora i keša, ukoliko može da radi na višem radnom taktu, utoliko može da doprinese ukupnoj brzini sistema.

ROM predstavlja memoriju čiji sadržaj je trajan i nepromenljiv. ROM se može samo čitati, ne može menjati ni brisati. ROM sadrži najosnovnije programe i podatke neophodne za funkcionisanje računarskog sistema na najnižem nivou. Drugim rečima, pre nego što se bilo šta učita u memoriju našeg kućnog računara, programi iz ROM-a su aktivni i oni su ti koji u stvari čine mogućim naš početak rada na računaru. Na PC računaru je ovaj osnovni sistemski softver poznat kao **BIOS** (*Basic Input/Output System*).

PROM je tip ROM-a koji je inicijalno prazan, u koji se onda mogu upisati programi i podaci koji se više ne mogu menjati i PROM se nadalje ponaša kao običan ROM.

EPROM je vrsta ROM-a u koji se mogu upisati podaci koji onda ostaju trajno zapisani. Ipak, podaci se mogu i obrisati pomoću specijalnog postupka držanjem u komori sa ultraljubičastom svetlošću.

EEPROM je unapređena vrsta EPROM-a koja se može izbrisati i bez korišćenja ultraljubičaste svetlosti. Postoje različite varijante ovog tipa memorije (npr. *flash* memorija), koje danas imaju primenu kao USB flash memorije, nova generacija hard diskova (SSD) i sl. U stvari, u današnje vreme, čak i u računarima, prava ROM memorija se jako retko koristi. Obično se menja mnogo praktičnijom EEPROM memorijom, zahvaljujući čemu korisnici imaju čak i mogućnost zamene BIOS-a u svom računaru.[10]

U prošlom poglavlju smo definisali računar kao osnovu računarskog sistema. Da bi računar komunicirao sa spoljnim svetom potrebno ga je povezati sa **periferijskim uređajima**. Periferijske uređaje delimo na ulazne, izlazne i ulazno-izlazne.

Svi periferijski uređaji imaju i nešto elektronike u sebi koja je potrebna za njihovo osnovno funkcionisanje. Najčešće je to osnovni program u ROM-u (tzv. *firmware*) i kontroler, međutim neki uređaji, poput laserskih štampača ili grafičkih kartica, imaju čitave procesore i memoriju.

Jedna posebna vrsta memorije koja se vezuje za gotovo sve periferijske uređaje i spoljne memorije je **bafer** (*buffer*). Bafer predstavlja jednu vrstu pomoćne memorije (ne treba je mešati sa procesorskim kešom), koja služi za usklađivanje toka podataka iz i do uređaja. Najbolja ilustracija važnosti bafera je proces snimanja na CD/DVD, kada se podaci smeštaju u bafer uređaja iz koga se ravnomerno upisuju na disk. Bafer je u ovom slučaju potreban zbog toga što računar može biti zauzet i drugim poslovima tokom snimanja, pa podaci koje treba snimiti dolaze do uređaja „stihijski“. Ukoliko se bafer isprazni (*buffer underrun*) pre završetka snimanja, utoliko dolazi do greške.

Ulazni uređaji

Ulazni uređaji predstavljaju uređaje kojima se neki podaci unose u računar. Drugim rečima, oni su način da spoljni svet nešto „kaže“ računaru.

Uređaji za unos alfanumeričkih podataka

TASTATURA je osnovni uređaj za ulaz na računarima. Sastoji se od određenog broja alfanumeričkih tastera (slova i brojke), tastera sa specijalnim znacima (interpunkcija, zagrade, aritmetičke operacije, razmak...), kao i određenog broja tastera posebne namene (funkcijski tasteri, tasteri za kretanje, kontrolni tasteri, brisanje, unos...). Na računarskoj tastaturi obično postoji još jedan izdvojeni deo sa numeričkim tasterima (tzv. „numerička tastatura“). Ukoliko se tastatura izrađuje u verziji koja zauzima manje mesta, ovaj deo se može izbaciti. Tastature laptop računara obično nemaju numeričku tastaturu (iako ju je moguće dokupiti kao zaseban dodatak).

Sa ergonomskog aspekta, prema osećaju koji tasteri daju tokom kucanja, tastature možemo podeliti na „meke“ i „tvrde“, kao i na „plitke“ i „duboke“. Ne postoji objektivni kriterijum po kome se može reći kakve tastature su bolje – ova kategorija zavisi isključivo od preferencija korisnika.

Tastature sa računarom mogu biti povezane kablom ili bežično. Bežična veza zahteva poseban dodatak koji se priključuje na PS/2 ili USB ulaz računara. Iako pruža veću slobodu prilikom rada sa računarom (i do par metara udaljenosti), ovakva tastatura koristi baterije i (zavisno od

kvaliteta) može se desiti da tasteri sporije reaguju. Današnje tastature mogu imati i posebne komande za rad sa multimedijom i Internetom, mogu imati pozadinsko osvetljenje i sl.

Uređaji za unos pozicionih podataka

Uređaji za unos pozicionih podataka doživeli su svoju ekspanziju sa pojavom operativnih sistema sa grafičkim korisničkim interfejsom. Tastatura je ostala primarni uređaj za unos podataka, ali kada je komunikacija korisnika i računara prevazišla nivo zadavanja tekstualnih komandi, pojavila se potreba za novim tipom ulaznih uređaja, koji bi olakšali rad preko ikonica, prozora i menija.

Osnovna namena ovih uređaja jeste pomeranje posebnog pokazivača (najčešće u obliku strelice) po ekranu i aktiviranje komande na koju strelica pokazuje.

MIŠ predstavlja osnovni uređaj za unos pozicionih informacija. Iako se tokom godina menjao, osnovna namena mu je ostala ista. Pomeranjem miša po podlozi ili stolu pomera se i pokazivač (*mouse pointer*) na ekranu. Klikom na neko dugme miša poziva se odgovarajuća akcija operativnog sistema ili aplikacije.

Miševe možemo podeliti na **mehaničke** i **optičke**. Mehanički miševi sa svoje donje strane imaju kuglicu kojom se registruje samo pomeranje miša. Takvi miševi bi se relativno brzo napunili prašinom koja se nakupljala na unutrašnjim točkicama, što je vremenom otežavalo rukovanje i zahtevalo redovno čišćenje. Danas se mehanički miševi sve ređe koriste. Optički miševi umesto kuglice imaju lasersku diodu koja registruje pomeranje miša.

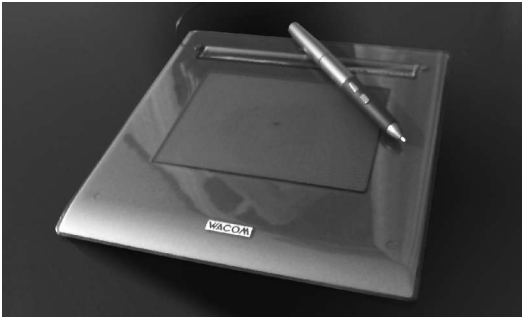
Kvalitetniji miševi imaju precizniju detekciju pokreta (tzv. rezoluciju). Kada su se pojavili, ovi miševi su imali visoku cenu i zahtevali su posebnu podlogu na kojoj bi miš morao da se uvek drži pod pravim uglom. Danas se ovi miševi mogu koristiti bez bilo kakvih ograničenja, s tim što mogu biti osetljivi na određen tip podloge (npr. ne bi funkcionisali na staklenom stolu). Miš može biti **povezan kablom** za računar ili **bežični**. Bežični miševi pružaju veću slobodu u radu, ali zahtevaju baterije.



KUGLICA (*trackball*) je uređaj koji izlazi iz upotrebe. U pitanju je uređaj koji funkcioniše na istom principu kao mehanički miš, s tim što je sam uređaj fiksiran, dok se kuglica nalazi sa gornje strane i pomera se dlanom. Ovi uređaji su služili kao zamena za miševe ukoliko korisnik nije imao dovoljno prostora za pomeranje miša, kao i na laptop računarima.

DODIRNA PODLOGA (*touchpad*) je uređaj koji je danas zamenio kuglicu na laptop računarima. Ovaj uređaj predstavlja površinu osetljivu na dodir. Pomeranjem prsta po površini pomera se i pokazivač na ekranu. U blizini *touchpad-a* nalaze se i tasteri koji imaju ulogu levog i desnog tastera miša. Novije podloge imaju i dodatnu mogućnost da tapkanje prsta po podlozi ima funkciju levog tastera.

GRAFIČKA TABLA predstavlja ulazni uređaj namenjen prvenstveno radu sa grafikom. Ovaj uređaj se sastoji iz dve komponente – posebne radne površine (table) koja registruje poziciju i olovke koja registruje pritisak. Olovka može biti povezana kablom za tablu ili bežična, što je u novije vreme češći slučaj. Sama tabla je uobičajeno povezana kablom sa računarom.



Budući da se ovaj uređaj koristi za rad sa grafikom, poseduje još neke specifične osobine, kao što su mogućnost detekcije jačine pritiska i ugla pod kojim se drži olovka. Programi za grafiku ove parametre tumače kao različite debljine i oblike linija. Novije grafičke table imaju površinu izvedenu kao LCD ekran, na kome se takođe vidi slika. Kvalitetnije i skuplje grafičke table imaju veću površinu, rezoluciju i više nivoa detekcije jačine pritiska olovke.

SVETLOSNO PERO (*light pen*) je još jedan uređaj koji pripada istoriji računara. U pitanju je uređaj u obliku olovke povezan kablom sa računarom koji detektuje svetlost sa monitora i na taj način služi za određivanje pozicije na ekranu.

EKRAN OSETLJIV NA DODIR (*touchscreen*) predstavlja posebnu grupu LCD monitora koji su osetljivi na dodir. Trenutno se koriste najviše na mobilnim uređajima (PDA računari i mobilni telefoni). Obično se za rad sa ovakvim ekranom koristi posebna plastična olovka (rezistivni), kako bi se postigla veća preciznost, mada je uređaj u mogućnosti da radi i na dodir prsta (kapacitivni).

Uređaji za unos grafičkih / video podataka

SKENER je uređaj za digitalizaciju slike. Može biti izveden kao ručni ili stoni. Ručni skeneri se danas manje koriste, a upotrebljavaju se ukoliko je potrebno skenirati nedostupne površine. Nedostatak im je što se skeniranje mora obavljati pažljivo, nepromenljivom brzinom da ne bi došlo do deformacija slike. Stoni skeneri imaju providnu površinu na koju se stavlja slika koju je potrebno digitalizovati. U unutrašnjosti skenera nalazi se izvor svetlosti i optički senzor koji vrši digitalizaciju slike.[1]

Stone skenere razlikujemo po veličini površine za skeniranje (A4 ili A3), brzini skeniranja i optičkoj rezoluciji, koja se izražava brojem tačaka po inču (DPI). Od optičke rezolucije prvenstveno zavisi i kvalitet skenirane slike. Veća vrednost će dati bolju sliku. Ovu rezoluciju treba razlikovati od rezolucije koja se obično navodi u specifikacijama, pošto se navedene visoke vrednosti dobijaju matematičkim algoritmima (interpolacijom) i ne predstavljaju merilo kvaliteta skenera.

WEB KAMERA je uređaj koji se obično koristi u komunikaciji putem Interneta. Obično se postavlja na monitor, dok je noviji laptop računari obično imaju ugrađenu. U pitanju je kamera malih dimenzija, obično znatno slabijih karakteristika u odnosu na digitalne video kamere i fotoaparate.

TV TJUNER se obično ugrađuje u računar u obliku kartice (jeftinija varijanta) ili se povezuje sa računarom putem USB porta. Ovaj uređaj omogućuje praćenje TV programa na ekranu računara. Na njega je moguće povezati koaksijalni (antenski) kabl radi uspostavljanja prijema. Osim antene ili kablovske televizije, na tjuner se može povezati i video rikorder ili DVD plejer. TV program se obično prati uz pomoć specijalizovanog softvera koji se dobija uz tjuner. Osim gledanja, program je moguće i snimiti na računar. Kvalitetnije tzv. **video kartice** koriste se u profesionalne svrhe i pružaju mogućnost kompresije i dekompresije video zapisa u realnom vremenu.

Ulazni uređaji za posebne namene

PALICA ZA IGRU (*joystick*), kao i drugi uređaji (volan, pedale, *joypad*), koriste se za igranje igara. Novije palice za igru su izvedene tako da simuliraju analogne uređaje, odnosno tako da razlikuju nivo pomeranja od strane korisnika. Osim ove mogućnosti, današnji uređaji za igranje imaju i mogućnost „povratne sprege“ (*force feedback*) koja korisnicima treba da pruži još realniji osećaj prilikom igranja.



ČITAČ BAR-KODA je uređaj u neku ruku sličan skeneru, s tim što je namenjen očitavanju bar-koda predstavljenog serijom linija i praznina različitih širina. Bar-kod predstavlja informaciju o proizvodu, koja je prilagođena mašinskom čitanju. Čitači bar-koda mogu biti u obliku olovke koja se prevlači preko koda, u obliku ručnog skenera („pištolja“) ili kao fiksirani uređaj. Ove uređaje obično srećemo na kasama u supermarketima.

Izlazni uređaji

Izlazni uređaji predstavljaju vrstu perifernih uređaja kojima se računar „obraća“ spoljnom svetu. Bez njih ne bismo mogli da vidimo rezultate rada programa.

Grafički podsistem

GRAFIČKA KARTICA predstavlja jedan od esencijalnih delova računarskog sistema. Njena uloga je da omogući prikaz sa računara na monitoru. Osnovne karakteristike grafičke karte su količina memorije i brzina, što je i očekivano budući da ona danas predstavlja ceo računar u malom.

Procesor grafičke kartice omogućuje 3D akceleraciju, što zahteva veliki broj izračunavanja u pokretnom zarezu, kako bi se slika kreirala i kontinuirano prikazivala tokom animacije. Brži procesor će moći da izvrši više matematičkih operacija u sekundi, a samim tim će i 3D scene biti kompleksnije, moći će da se iscrtavaju na većim rezolucijama i uz korišćenje specijalnih efekata. Kada se govori o 3D akceleraciji, bitno je i koje interfejse (*API – Application Programming Interface*) za programiranje 3D grafike podržava kartica. Danas se na računarima uglavnom koriste DirectX i OpenGL i za karticu je značajno da svojim mogućnostima podržava njihove najnovije verzije. Iako se ove mogućnosti grafičkih kartica koriste prvenstveno u igrima, takođe pronalaze upotrebu u projektovanju, animaciji, virtuelnoj realnosti, simulacijama, a odnedavno i operativni sistemi koriste mogućnosti 3D akceleracije radi prikaza korisničkog interfejsa.

Količina memorije takođe igra važnu ulogu. Pre no što su grafičke karte postale 3D akceleratori, količina memorije se kretala od 1MB do 4MB. Kartica sa više memorije mogla je da prikaže sliku u većoj rezoluciji i u većem broju boja. Danas se memorija meri stotinama megabajta, pošto je za kartice neophodno da pamte veliki broj detaljnih tekstura koje se apliciraju na trodimenzionalne modele objekata. Ukoliko kartica nema dovoljno memorije, utoliko će prilikom iscrtavanja podaci morati da se prenose iz sporije RAM memorije, što će se loše odraziti na brzinu.

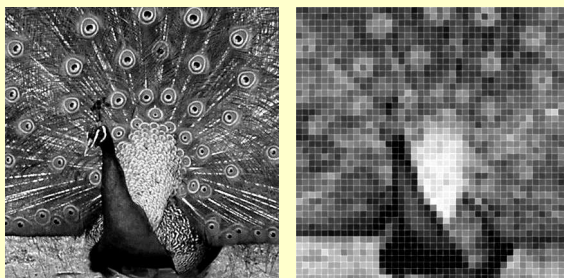
Ukoliko je 3D kartica integrisana na matičnoj ploči, neće imati svoju memoriju već će deliti RAM sa ostalim aplikacijama. Ovakva rešenja su jeftina ali nikada nisu dobra kao adekvatne 3D kartice koje se povezuju sa matičnom pločom preko odgovarajućeg slota.

Još jedna karakteristika koja nas interesuje pri kupovini kartice jeste i kakve su njene mogućnosti izlaza – da li je u pitanju analogni RGB ili digitalni izlaz za monitor, da li kartica podržava dva monitora, kao i da li postoji video izlaz za povezivanje na TV.

Rezolucija

U ovom poglavlju se na nekoliko mesta pominje pojam „rezolucija“, tako da ćemo na ovom mestu razjasniti o čemu se radi.

Rezolucija ima veze sa načinom na koji računari i uopšte digitalni uređaji pamte i prikazuju sliku. Slika se sastoji iz velikog broja tačaka (*pixel* ili *dot*) koje zajedno stvaraju iluziju nekog prikaza. Što su tačke sitnije, slika izgleda prirodnije i oštrije, kao što se može videti na donjem primeru.



Kada se govori o **grafici**, misli se na ekransku rezoluciju koja, najprostije rečeno, predstavlja broj tačaka po horizontali i vertikali ekrana. Tako se i rezolucija izražava preko dva broja, npr. 800x600, 1024x768 ili 1280x1024. Ukoliko tačaka ima više, utoliko će biti sitnije i slika će biti kvalitetnija, ali će zauzimati više memorije i zahtevati više vremena da bi se iscrtala.

Ako se govori o **slici na papiru** (bez obzira da li se štampa ili skenira), rezolucija se izražava kao broj tačaka po inču (DPI – *Dots Per Inch*). Rezolucija za relativno kvalitetnu štampu bila bi 300DPI. Sliku sa računara je moguće odštampati u različitim DPI rezolucijama, gde će ona pri manjem DPI izgledati veće, ali nekvalitetnije.

Sa ekspanzijom **digitalnih fotoaparata** i kamera, pojavila se još jedna popularna mera rezolucije slike, tzv. „megapiksel“, koji u stvari predstavlja broj tačaka iz kojih se slika sastoji izražen u milionima. To bi značilo da se slika veličine 1280x1024 sastoji od 1,310,720 tačkica, odnosno (zaokruženo) od 1,3 megapiksela.

MONITOR predstavlja najvažniji izlazni uređaj računara. Osnovna karakteristika monitora je njegova veličina, koja se izražava, kao dužina dijagonale ekrana, u inčima. Vezano za dimenzije monitora, u današnje vreme, osim klasičnih monitora čiji je ekran proporcija 4:3, postoje i tzv. široki (*widescreen*) monitori.

Prema načinu prikaza slike, monitore danas delimo primarno na monitore sa katodnom cevi (CRT – *Cathode Ray Tube*) i monitore sa tečnim kristalima (LCD – *Liquid Crystal Display*).

CRT monitori se prepoznaju po svom gabaritu i danas polako izlaze iz upotrebe. Slika se formira na osnovu snopa elektrona koji se ispaljuje iz katodne cevi i koji aktivira fosforni sloj na ekranu. Zbog toga, veći monitori i TV uređaji moraju imati dublje kućište sa zadnje strane. Sama tehnologija je ograničavajuća, tako da se suviše veliki ekrani ne mogu ni proizvoditi. Raniji problemi vezani za štetno zračenje ovih monitora ili prepleteni (*interlace*) prikaz slike,

danas su uglavnom rešeni. Od ostalih karakteristika, koje nas interesuju, najvažnije su rezolucije koje monitor može da prikaže (u pikselima) i brzina vertikalnog osvežavanja ekrana (u Hz). Ovo je posebno bitno zbog toga što nedovoljna brzina osvežavanja čini sliku nestabilnom i tako utiče na zamor očiju i pojavu glavobolje kod korisnika, a pri kontinuiranoj upotrebi i do poremećaja vida.

Sa druge strane, **LCD monitori** su mnogo lakši i tanji. Najveći problem ovih monitora svojevremeno bila je brzina odziva, odnosno brzina reagovanja na promene (u mili-sekundama). Ovi monitori su bili prilično spori u osvežavanju slike (veći broj mili-sekundi), što je smetalo pri radu, posebno prilikom stalnih izmena na ekranu. Uvođenjem tehnologija, poput aktivne matrice ekrana (**TFT – Thin-Film Transistor**), ovaj problem je drastično redukovano.



Drugi problem LCD monitora je pojavljivanje tzv. „mrtvih“ piksela. To su tačkice na ekranu koje su postale neaktivne i više se ne mogu upaliti. Različiti proizvođači monitora imaju različite nivoe tolerancije – broja mrtvih piksela koji neće biti obuhvaćeni garancijom (što više – to lošije).

Od ostalih osobina, bitni su još kontrast i širina ugla gledanja. Kontrast predstavlja odnos najtamnije crne i najsvetlije bele boje na monitoru. Što je kontrast veći, slika će biti kvalitetnija. Širina ugla predstavlja ugao pod kojim korisnik može gledati ekran bez deformisanja boje i osvetljenosti slike. Kod starijih monitora korisnik mora gledati ekran pod pravim uglom, dok danas bolji monitori pružaju veću slobodu.

Osim pomenutih, postoje i **plazma monitori**, koji po izgledu veoma podsećaju na LCD monitore. Korišćenje plazma tehnologije svakako ima određenih prednosti, s obzirom na to da takvi ekrani daju svetliju sliku, mogu se gledati iz šireg ugla i mogu se proizvoditi u većim dimenzijama od LCD monitora. Najveći problem kod plazma monitora je tzv. *burn-in* piksela, odnosno pojava tačkica koje vremenom postaju „zamrznute“, tako da mogu prikazivati samo jednu boju. Ovaj problem se manifestuje ako se na istom delu ekrana stalno prikazuje jedna te ista slika (npr. logo TV stanice na plazma televizoru ili prozor programa u kome često radimo na računaru) koja se kasnije stalno vidi na ekranu kao „duh“.

Pre izvesnog vremena najavljena je nova tehnologija tankih CRT monitora (**SED – Surface-conduction Electron-emitter Display**), koja bi trebalo da nadmaši sve pomenute tipove monitora. U pitanju su ekrani kod kojih za svaki piksel postoji poseban snop elektrona. Proizvodnja ovih monitora do sada nije počela.

VIDEO PROJEKTOR (*video beam*) je uređaj koji se može povezati na grafičku karticu. Služi za projekciju sadržaja na platnu. Kod ovih uređaja bitna je jačina svetlosti koju emituju, kontrast koji postižu, kao i dužina života lampe, koja predstavlja njegov najvažniji deo.

Uređaji za izlaz na papiru

ŠTAMPAČ je uređaj za izlaz na papiru. Prema tehnologiji kreiranja otiska na papiru, štampače možemo podeliti na štampače sa lepezom, matrice, laserske, ink-džet i LED štampače.

Štampači sa lepezom podsećaju izgledom na pisaću mašinu. Prilagođeni su štampanju teksta i danas se više ne koriste.

Matrični štampači imaju glavu sa iglicama (prema broju iglica dele se na 9-pinske i 24-pinske) koja preko trake sa bojom ostavlja otisak na papiru. Ovi štampači su spori i bučni. Sa padom cena naprednijih štampača, prestaju da se upotrebljavaju.



Laserski štampači koriste boju u prahu – toner (tehnologija koju koriste i fotokopir mašine). Laser se koristi da naelektriše valjak na koji se nanosi boja, koja se onda prenosi na papir. Ovi štampači su skuplji od ostalih ali su tihi i veoma brzi. Iako se uglavnom koriste u crno-beljoj varijanti, postoje i laserski štampači u boji, koji, umesto jednog, koriste čak 4 tonera sa bojom, na šta treba posebno obratiti pažnju jer jedna od značajnih komponenata cene upravo je i cena tonera koja nije zanemarljiva. **LED štampači** su slični laserskim, s tim što umesto lasera koriste red svetlosnih dioda, zahvaljujući čemu su manji i jeftiniji.

Ink-džet (ink-jet) štampači koriste glavu na kojoj se nalaze veoma precizne dizne kojima se boja prska na papir. Boja je u tečnom stanju i nalazi se u kertridžima. Kapljice boje koje ostaju na papiru su toliko sitne da se mere pikolitrima. Ovi štampači relativno jeftino mogu štampati u koloru, tihi su ali generalno sporiji od laserskih.

Glavne karakteristike štampača su njegova rezolucija (iskazuje se u DPI) i brzina koja se iskazuje kao broj stranica u minuti (*PPM – Pages Per Minute*). Laserski štampači su po ovim osobinama obično bolji od drugih. Ostala pitanja koja interesuju korisnike jesu mogućnost štampe u boji, maksimalan format papira na kome se može štampati, mogućnost automatske dvostrane štampe, buka i mogućnost štampe na drugim medijima (na printabilnim CD-ovima, na paus papiru, itd).

PLOTER je uređaj za vektorsko crtanje. Sastoji se iz pera sa bojom koje se pomera preko papira i na taj način kreira crtež. Ploteri nisu namenjeni štampanju slika, ali zato mogu kreirati tehničke crteže i planove. Termin „ploter“ se danas koristi i za štampače velikih formata.

Ulazno-izlazni uređaji

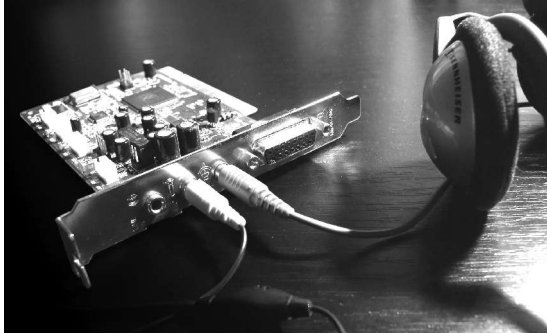
Ulazno-izlazni uređaji predstavljaju kombinaciju ulaznih i izlaznih uređaja. Preko njih računar može i da prima i da šalje podatke.

TERMINAL je najočigledniji primer ulazno-izlaznog uređaja. Sastoji se iz monitora i tastature. Povezuje se na glavni računar, na koji može biti povezano i više terminala. Danas se terminali u bukvalnom smislu reči retko koriste, pošto se zbog niske cene i ceo računar može upotrebiti kao terminal ukoliko je umrežen sa glavnim (*mainframe*) računarom.

Zvučni podsistem

ZVUČNA KARTICA je za računar ulazno-izlazni uređaj budući da se preko nje zvuk može digitalizovati ili reprodukovati. Svaka zvučna kartica ima nekoliko ulaza / izlaza kojima se povezuje na mikrofona, zvučnike, slušalice ili neki drugi muzički uređaj. **Zvučnici i slušalice** u

tom slučaju predstavljaju izlazne uređaje, dok **mikrofon** predstavlja ulazni. Današnje zvučne kartice se mogu povezati na različite sisteme zvučnika – običan stereo, 2.1, 5.1, 7. 1 i sl.



Karakteristike zvučne kartice su odnos zvuka i šuma (*SNR – Signal to Noise Ratio*, izražava se u decibelima), frekvencija digitalizacije odnosno reprodukcije zvuka (u kilohercima kHz) i preciznost zvuka (izražava se u bitovima). Veće vrednosti za sve tri karakteristike označavaju bolji kvalitet.

Zvučne kartice integrisane na matičnoj ploči uglavnom su zadovoljavajućeg kvaliteta za prosečnog korisnika.

Uređaji za povezivanje

MODEM (*MODulator – DEModulator*) je uređaj koji služi za povezivanje dva računara putem telefonske linije. Funkcioniše tako što podatke koje treba poslati pretvara u zvuk koji šalje telefonom, dok zvuk koji prima linijom digitalizuje nazad u podatke. Može biti izveden u internoj ili eksternoj varijanti. Interni modem se ugrađuje kao kartica u računar, dok se eksterni povezuje kablom i obično zahteva sopstveno električno napajanje. Glavni nedostatak modema, osim male brzine, je zauzeće telefonske linije tokom trajanja veze.

Najbitnije karakteristike modema su njegova brzina (izražava se u kbps – kilobiti u sekundi), koja ide do 56kbps, zatim prepoznavanje zauzetog signala, mogućnost slanja i primanja fax-a, mogućnost telefoniranja (*voice*) i posedovanje protokola za kompresiju i proveru podataka (aktuelan je protokol V.92).

KABLOVSKI MODEMI su posebni eksterni uređaji za povezivanje na druge tipove mreža. Njima se računar povezuje na koaksijalni kabl ili *wireless* antenu. Ovo nisu modemi u pravom smislu reči pošto se digitalni signal ne konvertuje u zvuk. U kablovske modeme spadaju i ISDN i ADSL (na slici) modemi. Čak i kada se povezuju na telefonsku liniju, kod ovih modema nema zauzeća telefona dok traje veza. Takvi modemi se povezuju sa računarom preko mrežne karte, USB porta ili bežično.

MREŽNA KARTICA je uređaj koji se ugrađuje u računar i služi za umrežavanje računara. Njome se računar može povezati sa drugim računarom ili nekim od mrežnih uređaja (*hub, switch, router*). Njena najbitnija karakteristika je brzina prenosa (koja se izražava u megabitima po sekundi – Mbps). Današnje kartice uglavnom imaju brzinu od 100Mbps.



Unutrašnja RAM memorija ima par nedostataka – relativno mali kapacitet i gubljenje sadržaja ukoliko prestane napajanje strujom.

RAM memorija obično ima dovoljan kapacitet za podatke sa kojima trenutno radimo. Njen kapacitet se može povećati do određenog nivoa, ali to ni izdaleka ne bi bilo dovoljno (niti isplativo) za sve podatke i programe koji su nam potrebni. Sa druge strane, podaci u RAM-u ostaju zabeleženi samo dok je računar uključen. Ukoliko dođe do prekida napajanja, ako se računar zablokira ili resetuje, podaci će biti izgubljeni.

Spoljna memorija računaru pruža mogućnost snimanja podataka. Ovako snimljeni podaci ostaju zabeleženi i po isključivanju računara, a kapacitet spoljne memorije može biti mnogo puta veći od unutrašnje. Jedini nedostatak spoljne memorije je taj što je ona uvek višestruko sporija u odnosu na unutrašnju.

Odnos računara i spoljne memorije je takav da se ona može istovremeno posmatrati i kao **memorija**, pošto služi za beleženje podataka, i kao **ulazno-izlazni uređaj**, budući da računar šalje podatke prema spoljnoj memoriji prilikom snimanja i prima podatke prilikom čitanja.

Prema načinu pristupa podacima, memoriju uopšte možemo podeliti na:

- **memoriju direktnog pristupa** – vreme pristupa bilo kom podatku je uvek isto – u ovu memoriju spadaju svi tipovi unutrašnje memorije i većina tipova spoljne memorije;
- **memoriju sekvencijalnog pristupa** – da bi se stiglo do nekog podatka, potrebno je proći kroz sve prethodne podatke i javlja se kod nekih oblika spoljne memorije (npr. magnetna traka).

Prema tehnologiji beleženja podataka, spoljnu memoriju možemo podeliti na:

- **papirnu** – ovo je oblik spoljne memorije kod koga se podaci beleže na papirima ili kartonima bušenjem ili zapisom koji se može mašinski pročitati – danas se ne koristi za snimanje veće količine podataka (bušene papirne kartice, bar kodovi);
- **magnetnu** – najtipičniji oblik spoljne memorije, kod koje se zapis obavlja korišćenjem magnetnog sloja (magnetna traka, disketa, hard disk);
- **optičku** – podaci se zapisuju na mediju na kome se vrši fizička ili hemijska promena koja se kasnije očitava optički (CD i DVD);
- **poluprovodničku** – podaci se beleže u EEPROM memoriji (*flash* memorija).

Prema načinu rada sa medijem na kome se vrši snimanje, uređaje spoljne memorije delimo na:

- **spoljnu memoriju sa nepromenljivim medijem** – medij na kome se snima ugrađen je u uređaj spoljne memorije i ne može se menjati (hard disk, *flash* memorija);
- **spoljnu memoriju sa promenljivim medijem** – medij se ubacuje u uređaj kako bi se podaci snimili ili pročitali (disketa, CD/DVD).

Bez obzira na tip uređaja ili način zapisa, svakoj spoljnoj memoriji, koja se koristi u današnje vreme, zajedničko je to da se podaci snimaju u obliku niza bajtova koje nazivamo **fajl** („datoteka“). Radi bolje preglednosti i lakšeg snalaženja, fajlovi su organizovani u logičke strukture preko sistema **foldera** i podfoldera („direktorijumi“). Ova struktura je suštinski struktura u obliku stabla gde se početni folder, koji sadrži sve ostale, naziva osnovni ili korenski (*root*) direktorijum.

Svaki uređaj spoljne memorije se pod *Windows* operativnim sistemom obeležava jednim slovom engleske abecede. Tako su za *floppy* disk jedinice rezervisana slova A i B, za primarni hard disk slovo C, a ostalim uređajima se dodeljuju naredna slobodna slova.

Spoljna memorija sa nepromenljivim medijem

Ova vrsta spoljne memorije se sastoji iz uređaja u koji je ugrađena i sama memorija. Ovi uređaji mogu biti ugrađeni u računar ili se povezivati na njega.

HARD DISK („tvrđi“ ili „čvrsti“ disk) je u današnjim računarima primarni uređaj spoljne memorije. Uobičajeno je ugrađen u kućište računara i u tom slučaju nije namenjen prenosu podataka, već samo njihovom čuvanju. Hard disk je **brži** od većine ostalih spoljnih memorija i ima **veći kapacitet** od bilo kog drugog (pojedinačnog) medija.

Uređaj se sastoji iz jednog ili više aluminijumskih diskova koji se nalaze na istoj osovinu i koji se rotiraju velikom brzinom (5400 rotacija u minuti ili 7200rpm kod novijih diskova za kućne računare). Diskovi su presvučeni feromagnetnim slojem, koji ima svojstvo da kada se namagnetise ostaje namagnetisan, pa se tako podaci mogu upisati i obrisati. Čitanje i upis podataka obavlja se preko magnetne glave koja ne dodiruje površinu već lebdi na vazdušnom jastuku na veoma malom rastojanju od nje. Zahvaljujući

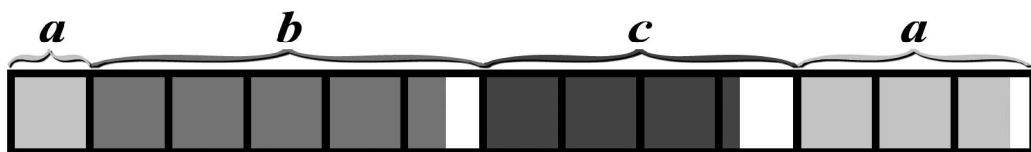


tome, hard disk je mnogo brži i pouzdaniji od drugih magnetnih medija, kao što je npr. disketa kod koje glava dolazi u fizički dodir sa površinom. Tehnologija koja se ubrzano razvija i koja bi trebalo da zameni dosadašnje hard diskove jesu SSD (*Solid-State Drive*) diskovi, koji suštinski predstavljaju EEPROM memoriju velikog kapaciteta i brzine.

Jedna specifičnost hard diska je mogućnost njegove podele na **particije**. Particije su nezavisni delovi diska koji se u operativnom sistemu posmatraju kao zasebni diskovi, bez obzira na to što se fizički radi o istom uređaju. Tako će pod *Windows-om* svaka particija imati dodeljeno zasebno slovo.

Zbog svog kapaciteta i brzine, hard disk se na računarskom sistemu može koristiti kao **virtuelna memorija**. Ukoliko računar tokom rada iskoristi raspoloživu RAM memoriju, operativni sistem može delove RAM-a snimati i učitavati sa diska, kako bi nastavio sa radom. Prostor na disku u kome se vrši ovo snimanje naziva se *swap* fajl. Nedostatak ovakvog „žongliranja“ memorijom je što se rad sa računarom usporava, budući da je disk, bez obzira na svoju brzinu, i dalje mnogo sporiji od unutrašnje memorije.

Način na koji se fajlovi i folderi organizuju na disku naziva se **fajl sistem**. Hard disk je podeljen na veliki broj koncentričnih kružnica („traka“ ili „cilindara“), koje su podeljene na jednak broj **sektora** (obično veličine 512 bajtova), koji su grupisani u **klustere** (*clusters*). Jedan klaster je najmanji mogući prostor na hard disku koji fajl može da zauzme. Kao što se vidi na slici, fajlovi uvek zauzimaju ceo broj klastera. Da bi se disk pripremio za upis fajlova potrebno je izvršiti njegovo **formatiranje** (kreiranje klastera i fajl sistema). Formatiranjem diska se ranije upisani podaci brišu.



Slika 4.1. Šematski prikaz rasporeda fajlova po klasterima

Prilikom snimanja fajla na disku, nije obavezno da klasteri koje taj fajl zauzima fizički budu poređani jedan za drugim. Tokom rada se obavlja veliki broj operacija sa fajlovima i može se desiti da disk postane **fragmentisan** (fajlovi razbacani po klasterima diska), što usporava rad sa diskom. Na priloženoj slici kao primer fragmentisanog fajla dat je fajl „a“. Ovaj problem se rešava defragmentacijom (u nekim sistemima tzv. „optimizacijom“ diska).

Prostor koga nema

Kao što se vidi na slici, prilikom snimanja fajlova, svi klasteri skoro nikada neće biti do kraja popunjeni. To znači da će zauzeće prostora na disku uvek biti veće od proste veličine fajla. Na primer, kada se na disk, čiji su klasteri veličine 32 kilobajta, snimi fajl veličine 1KB, on će zauzeti ceo klaster, što znači da je 31KB ostao neiskorišćen i neupotrebljiv.

Ovaj neiskorišćeni prazan prostor na disku naziva se **slack** i zavisi od veličine klastera, koja zavisi od fajl sistema i veličine diska. Svaki fajl sistem ima ograničen maksimalan broj klastera po disku. Kod starijih Windows fajl sistema FAT16 i FAT32, limit za broj klastera je relativno mali, tako da kod većih diskova i klasteri bivaju veći. Ovo može predstavljati problem ukoliko na disku imamo veliki broj malih fajlova.

Ovaj problem se može rešiti deljenjem diska na dve ili više manjih particija (svaka particija onda ima po maksimalan broj klastera) ili korišćenjem posebnih programa.

Verovatno najbolje rešenje jeste korišćenje novijeg **NTFS** fajl sistema koji može imati do 2⁶⁴ klastera, tako da i kod diskova, čiji kapacitet danas važi za relativno veliki, klaster obuhvata samo jedan sektor – veličine 512 bajtova, što drastično smanjuje *slack*.

Osim internih, danas postoje i **eksterni hard diskovi** koji se povezuju sa računarom, najčešće preko USB porta. Napajaju se iz računara ili zasebno iz električne mreže, ukoliko su u pitanju diskovi većeg kapaciteta. Pre pojave eksternih diskova, jedan od načina za prenos diska (bez otvaranja računara) bio je korišćenje **fioke** (*HDD rack*), odnosno posebnog ležišta koje je omogućavalo da se hard disk ubaci ili izvadi bez otvaranja kućišta računara.

USB FLASH MEMORIJA predstavlja uređaj, obično malih dimenzija, koji se sastoji iz USB priključka, kontrolera i EEPROM memorije.

Karakteristika ove memorije jeste da podaci snimljeni u njoj ostaju sačuvani čak i kada prestane električno napajanje. To se postiže posebnom organizacijom memorije koja se sastoji iz velikog broja ćelija, od kojih se svaka sastoji iz dva tranzistora („kontrolni“ i „plivajući“ tranzistor, razdvojeni tankim slojem metalnog oksida). Svaka ćelija sadrži 1 bit (binarna jedinica ili nula).

Danas, kapaciteti ovih uređaja iznose i do nekoliko desetina gigabajta. Osim kapaciteta, bitna karakteristika je i brzina, koja zavisi od USB interfejsa (sporiji USB 1.1 ili brži USB 2.0 i 3.0) i kvaliteta. Kvalitetniji (i skuplji) uređaji će biti brži. Razlikujemo brzine pri čitanju i pisanju, a performanse uređaja se najbolje proveravaju snimanjem velikog broja malih fajlova.



Ova vrsta memorije postoji i u varijanti sa ugrađenim MP3 plejerom (tačnije, MP3 plejeri se mogu povezati sa računarom putem USB-a i iskoristiti za prenos bilo kakvih podataka). Popularnost MP3 uređaja opada sa sve većim brojem mobilnih telefona koji takođe imaju ovu mogućnost.

Spoljna memorija sa promenljivim medijima

Kod ove vrste spoljne memorije postoji uređaj koji je ugrađen u računar ili se povezuje na njega, a koji ne sadrži sam medij na kome se podaci snimaju, već se mediji ubacuju u uređaj. Ovaj tip spoljne memorije koristi se prvenstveno za:

- prenos i
- čuvanje (*backup*) podataka.

Sigurnost pre svega

Računar, kao i svaki drugi uređaj, može da prestane da radi. Razlozi su mnogobrojni – idu od fizičkih kvarova pa do virusa. Ovo je trenutak da se zapitamo šta nam je zaista važno na računaru. Svaka njegova komponenta može da se zameni. Softver može ponovo da se instalira. Sve to vredi nešto vremena i novca.

Međutim, dokumente koje smo pisali danima, poslovne podatke koje smo prikupljali mesecima ili godinama, nije tako lako nadoknaditi. Iako postoji mogućnost da povratimo izgubljeno (vraćanje podataka koji su obrisani ili vraćanje sa pokvarenog hard diska), čarobna reč koja donosi spas je **bekap** (*backup*), odnosno pravljenje **sigurnosnih kopija** s vremena na vreme – zavisno od toga koliko su nam važni podaci.

DISKETA (*floppy disc*) je po svojim karakteristikama najsličnija hard disku. Unutar zaštitnog omotača nalazi se tanki poliesterski (plastični) disk, premazan fero-magnetnim slojem. Ovom plastičnom disku, disketa i duguje svoje ime – (engl. "floppy" – „savitljivi“) disk. Unutar disk jedinice, ovaj disk se okreće dok magnetna glava čita i upisuje podatke. Organizacija diskete je slična kao kod hard diska (ciklične trake i sektori), s tim što se disketa okreće mnogo sporije i magnetna glava dodiruje njenu površinu.

Diskete su se delile prema veličini na one od 8" (inča), 5.25" i 3.5". Starije diskete su bile upakovane unutar kartonskog omotača i bile su osetljivije na oštećenja u odnosu na kasnije proizvedene 3.5" diskete koje su se pravile u plastičnom omotaču.

Zavisno od kvaliteta magnetne površine, diskete su se delile na jednostrane i dvostrane (*SS – Single Sided* i *DS – Double Sided*), kao i na diskete sa dvostrukom ili visokom gustinom zapisa (*DD – Double Density* i *HD – High Density*). Zavisno od kvaliteta, disketa je mogla da se formatira na veći kapacitet. Poslednjih godina ova podela je postala izlišna, pošto su sve diskete bile pravljene kao DS/HD.

Kapacitet diskete je mnogo manji u odnosu na druge medije. Na jednu 3.5" DS/HD disketu, formatiranu na PC računaru, može da se smesti 1,44MB podataka.



Slika 4.2. Disketa od 8, 5.25 i 3.5 inča

Nesuđeni naslednik diskete bila je tzv. „zipeta“. Ovo je unapređena verzija diskete koja je imala kapacitet od 100MB ili, u novijoj verziji, čak 250MB. Da bi se radilo sa ovim disketama, bio je potreban i poseban uređaj – ZIP drajv, koji je inicijalno proizvodila firma Iomega. ZIP diskete jesu ušle u upotrebu, ali nikada nisu stekle širu popularnost zbog sve jeftinijih CD snimača i diskova.



Slika 4.3. Diskete od 5.25 i 3.5 inča i „zipeta“

Danas diskete, zbog malog kapaciteta, brzine i slabe pouzdanosti medija i uređaja (zbog čega se dešava da podaci ne mogu da se pročitaju čak ni sa novih disketa) izlaze iz upotrebe. Za fizički prenos podataka koriste se CD/DVD diskovi kao i *flash* memorija.

OPTIČKI DISK može biti CD (*Compact Disc*) ili DVD (*Digital Versatile Disc*). Ove dve vrste diskova su naizgled iste, a razlika koja interesuje korisnike je u kapacitetu – na CD se može snimiti do 700MB, a na DVD do 4,7GB.

Veći kapacitet diska se postiže upotrebom preciznijeg lasera, koji emituje svetlost manje talasne dužine. Precizniji laser omogućuje veću gustinu zapisa, a samim tim i više podataka.

Diskovi mogu imati jedan ili dva sloja za zapis podataka, pa tako razlikujemo **SL** (*single layer*) i **DL** (*dual layer*) diskove. DL diskovi imaju dvostruko veći kapacitet ali trenutno mnogo veću cenu i приметно manju brzinu u odnosu na SL diskove kakvi se standardno koriste.

Organizacija snimljenih podataka je malo drugačija u odnosu na hard disk ili disketu. Umesto koncentričnih kružnica, podaci su snimljeni na jednoj dugačkoj **spirali** koja se kreće od centra prema ivici diska. Optički disk takođe ima sektore, ali pošto se oni nalaze na jednoj neprekidnoj spirali, uvek su iste veličine. To znači da CD/DVD čitač mora menjati brzinu okretanja diska (brže kada se čitaju podaci snimljeni blizu centra, a sporije kada se čitaju podaci bliže ivici diska).

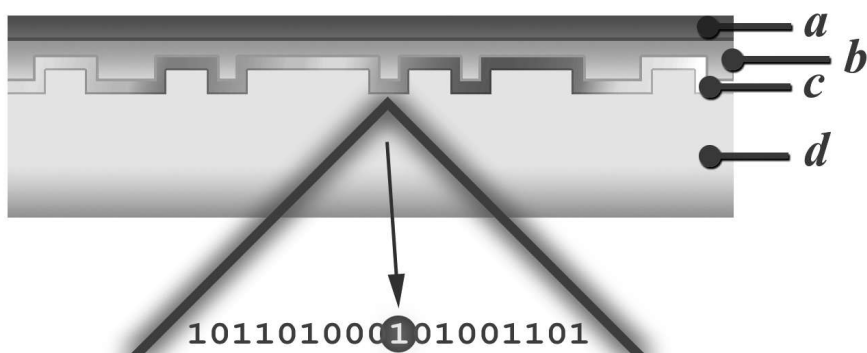
Prema mogućnosti za snimanje i brisanje podataka diskovi se mogu podeliti na:

- **CD/DVD-ROM** diskove na koje su podaci fabrički urezani („odštampani“) i mogu se samo čitati;
- **CD/DVD-R** (*recordable – write once*), „zapisive“ diskove koji su inicijalno prazni i na koje se pomoću snimača podaci mogu snimiti i čak dodavati dok se ne ispunji kapacitet diska, ali se kasnije ne mogu brisati ili menjati;
- **CD/DVD-RW** (*rewritable*), „prepisive“ diskove na koje se podaci mogu snimati, a kasnije i obrisati, pri čemu se disk dovodi u početno stanje, pa se podaci mogu ponovo upisivati.

Za DVD medije koje je moguće snimati (*recordable* i *rewritable*) postoji nekoliko različitih standarda – DVD-RAM, DVD-R/RW, DVD+R/RW. Današnji uređaji uglavnom mogu da rade sa svim ovim formatima, s tim što je potrebno obratiti pažnju koji tip medija najviše odgovara DVD uređaju, odnosno sa kojim može raditi na najvećoj brzini.

Brzina rada sa diskom varira za različite standarde i dodatno zavisi od toga da li je disk jednoslojan ili dvoslojan. Osim toga, brzine se razlikuju za čitanje ili pisanje (pisanje i brisanje su uvek sporiji u odnosu na čitanje).

Nepotrebno je naglasiti da CD uređaji ne mogu čitati DVD-ove, dok svi DVD uređaji mogu raditi i sa CD i DVD medijima.



Slika 4.4. Laser pogada ispučćenje što optički senzor tumači kao binarnu jedinicu.
Značenje oznaka: **a** – natpis, **b** – akrilna masa, **c** – aluminijumski sloj, **d** – providna plastika

Da bi se kreirao CD/DVD-ROM disk, na providnu polikarbonatnu plastiku urezuje se spiralna traka sa udubljenjima (ispupčenjima sa strane lasera), koja se potom sa gornje strane presvlači tankim slojem aluminijuma (zbog refleksije). Iznad aluminijumskog sloja nanosi se sloj akrilne mase, preko koga se nalazi sloj za natpis, odnosno štampu.

Laser se fokusira na traku sa podacima i njegova svetlost se odbija od površine CD-a do optičkog senzora. Zavisno da li je laser „pogodio“ izbočinu ili udubljeni deo na CD-u, senzor će registrovati promenu u odbijenoj svetlosti i na taj način se izbočine i udubljenja tumače kao binarne jedinice i nule.[13]

Ideja kod funkcionisanja zapisivih diskova (*recordable*) je u tome da se na reflektujućem sloju laserom za upisivanje vrši trajna promena (npr. zatamnivanje) površine čime ona više ne reflektuje laserski zrak za čitanje i tumači se kao binarna nula. Kod prepisivih (*rewritable*) diskova reflektujući sloj se kreira od posebne legure koja se laserom može dovesti u kristalizovano ili amorfno stanje. Delovi sloja koji su kristalizovani odbijaju laserski zrak, što predstavlja binarnu jedinicu, dok amorfni delovi ne reflektuju laserski zrak i samim tim predstavljaju binarnu nulu.

Najnoviji format optičkih diskova, koji polako ulazi u sve širu upotrebu, je BD (*Blu-Ray Disc*), čija je najveća primena trenutno u filmskoj industriji, za distribuciju filmova u visokoj rezoluciji, sa mnoštvom dopunskih materijala i različitim zvučnim sinhronizacijama. Na jedan BD može stati oko 25GB podataka, u njegovoj SL (*single layer*) varijanti.

MEMORIJSKA KARTICA predstavlja još jednu EEPROM memoriju, koja se umeće u poseban uređaj, **čitač kartica**, koji je povezan sa računaram.



Kartice mogu biti različitih formata, od kojih su najpoznatiji CF (*CompactFlash*), SD (*Secure-Digital*), MMC (*MultiMediaCard*), xD, *Memory-Stick*.

Najveću upotrebu ove kartice nemaju na računarima, već na drugim uređajima kod kojih postoji potreba za većom količinom (izmenljive) memorije, kao što su digitalni fotoaparati ili mobilni telefoni.

Ovo poglavlje počinjemo jednom zagonetkom:

$$3 + 5 = 10$$

Ako mislite da je ovo nemoguće, nastavite sa čitanjem.

Sisteme brojeva možemo podeliti na:

- pozicione i
- nepozicione.

Nepozicioni sistemi brojeva (aditivni i hibridni) odlikuju se time da ista cifra na različitim mestima u broju ima istu vrednost, a ukupna vrednost broja se dobija izračunavanjem (najčešće sabiranjem ili oduzimanjem) cifara. Primer ovakvog sistema brojeva jesu **rimski brojevi**.

Npr. u rimskom broju „VIII“, cifra „I“ svuda ima istu vrednost (1) iako se nalazi na različitim pozicijama, a ukupna vrednost broja se dobija kao zbir cifara V+I+I+I, odnosno 5+1+1+1, što daje broj 8. Ovakvi sistemi brojeva se danas jako retko koriste.

Kod **pozicionih sistema** vrednost zavisi od cifara i njihovih pozicija u broju.

Npr. u broju 3234 prva i druga „trojka“ imaju sasvim različite vrednosti. Prva cifra „3“ ima vrednost 3000 pošto se nalazi na poziciji za hiljade, dok druga ima vrednost 30, budući da se nalazi na poziciji za desetice.

Sisteme brojeva razlikujemo prema njihovim osnovama. **Osnova brojnog sistema je broj različitih cifara koje se koriste za zapis brojeva**. Neki od brojnih sistema koji su u upotrebi (pogotovo u radu sa računarima) su:

- **dekadni** (koristi se osnova **10**),
- **binarni** (osnova **2**),
- **oktalni** (osnova **8**),
- **heksadecimalni** (naziva se i „heksadekadni“ – osnova **16**).

Npr. u **dekadnom sistemu** brojeva, koji uobičajeno koristimo, svi brojevi mogu da se predstavljaju pomoću cifara **0, 1, 2, 3, 4, 5, 6, 7, 8 i 9**. Pošto cifara ima ukupno deset, kažemo da je osnova dekadnog sistema brojeva – broj **10**.

Računari, zbog svoje konstrukcije, sve podatke i naredbe (u svom pravom obliku, koji je često nevidljiv za čoveka) skladište kao nizove nula i jedinica. Zbog toga je logično i da računar radi sa tzv. **binarnim sistemom** brojeva. Binarni sistem koristi samo dve cifre, a to su 0 i 1 i njegova osnova je broj 2. Jedan tipičan broj u binarnom sistemu izgleda kao **101001,1101** a njegova vrednost u dekadnom sistemu je:

$$(101001,1101)_2 = (41,8125)_{10}$$

Osim binarnog, radi lakšeg zapisa, koriste se još i **oktalni** (osnova 8) i **heksadecimalni** (osnova 16) sistemi brojeva. Ovi brojevi su takođe pogodni za rad sa računarnom, pošto njihove osnove predstavljaju stepen broja 2 (koji je osnova binarnog sistema), pa su tako $8 = 2^3$ i $16 = 2^4$.

Oktalni sistem brojeva koristi cifre 0, 1, 2, 3, 4, 5, 6 i 7, dok heksadecimalni sistem koristi cifre 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E i F, pri čemu cifre od A do F imaju sledeće dekadne vrednosti:

$$\mathbf{A = 10 \quad B = 11 \quad C = 12 \quad D = 13 \quad E = 14 \quad F = 15}$$

Kako bi izgledalo brojanje u ovim sistemima? Pogledajmo primer:

dekadno	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...
binarno	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	1000	1000	...
															0	1		
oktalno	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	...
heksadec.	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	...

Da li možete da pretpostavite koji broj ide posle heksadecimalnog broja $(72FF)_{16}$? Kad brojimo, uvek povećavamo poslednju (najslabiju) cifru. Ako smo „potrošili“ sve cifre, onda se ova cifra postavlja na 0, a sledeća (jača) cifra se povećava za jedan, ako je i ona na maksimalnoj mogućoj cifri, i nju postavljamo na nulu pa povećavamo sledeću itd. Znači prvi sledeći broj bio bi $(7300)_{16}$.

Sada već naslućujemo rešenje zagonetke sa početka ovog teksta. Dati brojevi nisu u dekadnom već u oktalnom sistemu. Kada se vrednosti 3 i 5 saberu dobija se, naravno, vrednost 8, ali u oktalnom zapisu osam se zapisuje kao „10“ (ovo ne bi trebalo da se čita kao „deset“ pošto to i nije). Priznajemo da smo malo „varali“ – kada se koristi neki sistem brojeva različit od dekadnog, potrebno je to jasno naznačiti. Tako je izraz trebalo da bude zapisan kao:

$$(3)_8 + (5)_8 = (10)_8.$$

Vrednost broja u dekadnom sistemu

Vrednost broja u nekom pozicionom brojnom sistemu dobija se kao zbir ukupnih vrednosti cifara od kojih je sačinjen taj broj. Ukupna vrednost cifre dobija se na osnovu vrednosti same cifre i vrednosti koja zavisi od njene pozicije, odnosno „jačine“ (ili „težine“) cifre. Jačina cifre se računa kao odgovarajući stepen osnove brojnog sistema, a taj stepen zavisi od pozicije cifre.

Matematički ta formula izgleda ovako:

$$(X)_N = \sum_{i=-m}^n N^i x_i$$

- N predstavlja osnovu brojnog sistema;
- $(X)_N$ je neki broj zapisan u sistemu brojeva sa osnovom N ;
- x_i je pojedinačna cifra broja X koja se nalazi na i -toj poziciji.

Pozicije cifara obeležavaju se na sledeći način:

$$(n) \dots (3) (2) (1) (0) , (-1) (-2) \dots (-m)$$

Najvažnije je da primetimo kako se poslednja cifra pre decimalnog zareza označava pozicijom 0. Od te pozicije svaka jača cifra (svaka cifra ulevo – desetice, stotine, hiljade) ima za po jednu

jaču poziciju (1, 2, 3...), a svaka cifra posle decimalnog zareza ima po jednu slabiju poziciju (-1, -2, -3...)

Na praktičnom primeru to izgleda ovako:

POZICIJA CIFRE	5	4	3	2	1	0		-1	-2	-3	-4
DEKADNI (osnova 10)	758 122,3413										
cifre	7	5	8	1	2	2	,	3	4	1	3
vrednosti cifara	$7 \cdot 10^5$	$5 \cdot 10^4$	$8 \cdot 10^3$	$1 \cdot 10^2$	$2 \cdot 10^1$	$2 \cdot 10^0$		$3/10^1$	$4/10^2$	$1/10^3$	$3/10^4$
zbir	700 000 + 50 000 + 8 000 + 100 + 20 + 2 + 0,3 + 0,04 + 0,001 + 0,0003										
VREDNOST	758 122,3413										

Kada se za zadati broj **758.122,3413** razdvoje sve cifre, vrednost broja je moguće dobiti sabiranjem ukupnih vrednosti cifara.

Za cifru 7, koja se nalazi na poziciji 5, ukupna vrednost se dobija kao $7 \cdot 10^5$.

odnosno, kada se osnova brojnog sistema **10** stepenuje na poziciju cifre 5 i to se pomnoži samom cifrom 7. Tako se dobija $7 \cdot 100.000$, odnosno **700.000**.

Na ovaj način se računa vrednost svake cifre.

Vrednost cifre na nultoj poziciji (cifra 2) dobija se kao $2 \cdot 10^0 = 2 \cdot 1 = 2$

(pošto ne zaboravljamo da bilo koji broj stepenovan na „nulti“ stepen ima vrednost 1).

Što se tiče cifara posle decimalnog zareza i njihove vrednosti se računaju na isti način. Npr. cifra (4) koja se nalazi na drugom mestu odnosno poziciji (-2):

$$4 \cdot 10^{-2} = 4 \cdot 1/10^2 = 4 \cdot 0,01 = 0,04.$$

Kada se sve ovako dobijene vrednosti saberu, dobija se broj od koga je i započeto računanje, odnosno vrednost **758.122,3413**.

Način koji je prikazan može se koristiti za konverziju brojeva iz bilo kog sistema brojeva u dekadni brojni sistem. Jedino što se menja je osnova koja se stepenuje.

Slede primeri za binarni, oktalni i heksadecimalni sistem (cifre u brojevima su odvojene razmacima radi preglednosti).

BINARNI (osnova 2)	10 1001, 1101										
cifre	1	0	1	0	0	1	,	1	1	0	1
vrednosti cifara	$1 \cdot 2^5$	$0 \cdot 2^4$	$1 \cdot 2^3$	$0 \cdot 2^2$	$0 \cdot 2^1$	$1 \cdot 2^0$		$1/2^1$	$1/2^2$	$0/2^3$	$1/2^4$
zbir	32 + 0 + 8 + 0 + 0 + 1 + 0,5 + 0,25 + 0,0 + 0,0625										
dekadna vrednost	41, 812 5										
OKTALNI (osnova 8)	154 027, 205 4										
cifre	1	5	4	0	2	7	,	2	0	5	4
vrednosti cifara	$1 \cdot 8^5$	$5 \cdot 8^4$	$4 \cdot 8^3$	$0 \cdot 8^2$	$2 \cdot 8^1$	$7 \cdot 8^0$		$2/8^1$	$0/8^2$	$5/8^3$	$4/8^4$
zbir	32768 + 20480 + 2048 + 0 + 16 + 7 + 0,25 + 0,0 + 0,009766 + 0,000977										
dekadna vrednost	55 319, 260 74										
HEKSADECIMALNI (osnova 16)	1D A9 2A, CC 0F										
cifre	1	13	10	9	2	10	,	12	12	0	15
vrednosti cifara	$1 \cdot 16^5$	$13 \cdot 16^4$	$10 \cdot 16^3$	$9 \cdot 16^2$	$2 \cdot 16^1$	$10 \cdot 16^0$		$12/16^1$	$12/16^2$	$0/16^3$	$15/16^4$
zbir	1048576 + 851968 + 40960 + 2304 + 32 + 10 + 0,75 + 0,046875 + 0,0 + 0,000228882										
dekadna vrednost	1 943 850, 797 104										

Prevođenje broja iz dekadnog u drugi sistem brojeva

Racionalni brojevi, odnosno brojevi koji imaju i celi i decimalni deo, prevode se u dva koraka, tako što se zasebno preračunava celi deo, a zasebno decimalni.

Celi deo se prevodi uzastopnim celobrojnim deljenjem sa osnovom brojnog sistema u koji se vrši preračunavanje. Pri svakom celobrojnem deljenju beleže se ostaci koji u stvari predstavljaju cifre novog broja. Razlomljeni deo se konvertuje uzastopnim množenjem sa osnovom brojnog sistema u koji se preračunava i ovde se kao cifre posle decimalnog zareza uzimaju celi delovi dobijenih rezultata.

Konverziju ćemo objasniti na primeru broja $(41.8125)_{10}$, koji ćemo konvertovati u binarni broj.

Prvo se vrši konverzija celog dela, odnosno broja 41. Broj delimo sa osnovom binarnog sistema, odnosno brojem 2. Pošto je deljenje celobrojno, iako je rezultat 20,5, zanemarujemo deo posle decimalnog zareza i beležimo ostatak deljenja ($20 \cdot 2 = 40$, $41 - 40 = 1$, što predstavlja ostatak). Znači:

$$41 / 2 = 20, \text{ uz ostatak } 1$$

Sada ponavljamo deljenje, s tim što kao deljenik uzimamo celobrojni rezultat 20.

$$20 / 2 = 10, \text{ uz ostatak } 0$$

Deljenje se vrši na ovaj način dok se kao rezultat ne dobije 0 (uz ostatak 1). Ostaci koji su dobijeni (pošto je u pitanju deljenje sa 2, ostaci mogu biti samo 0 i 1) predstavljaju cifre celog dela traženog binarnog broja. Ostaci se s leva udesno zapisuju od poslednjeg prema prvom. Tako dobijamo da je

$$(41)_{10} = (101001)_2$$

Decimalni deo se takođe preračunava kroz uzastopne iteracije. Broj 0,8125 se množi osnovom binarnog sistema, odnosno brojem 2. Kao rezultat se dobija 1,625. Od ovog broja se beleži celi deo (broj 1), a decimalni deo se ponovo množi. Znači:

$$0,8125 \cdot 2 = 1,625; \text{ beležimo } 1, \text{ a nastavljamo računanje sa } 0,625,$$

$$0,625 \cdot 2 = 1,25; \text{ opet beležimo } 1, \text{ i računanje nastavljamo sa } 0,25.$$

Množenje se vrši sve dok se kao rezultat ne dobije broj **1,0** (dok ne ostanemo bez decimala). Kod nekih brojeva se računanje može vršiti do u beskonačnost i kod takvih brojeva prekidamo računanje posle zadatog broja decimala. Na ovaj način smo kao rezultat dobili $(0,8125)_{10} = (0,1101)_2$

A kao finalni rezultat pretvaranja broja: $(41,8125)_{10} = (101001,1101)_2$

Ceo proces računanja dat je u tablici.

DEKADNI U BINARNI		CELI DEO			DECIMALNI DEO			
Dekadna vrednost	41.8125	41 / 2	20	1	0.8125 * 2	1.625	1	0.625
OSNOVA	2	20 / 2	10	0	0.625 * 2	1.25	1	0.25
Binarna vrednost 101001,1101		10 / 2	5	0	0.25 * 2	0.5	0	0.5
		5 / 2	2	1	0.5 * 2	1	1	0
		2 / 2	1	0				
		1 / 2	0	1				

Na isti način se vrši i konverzija iz dekadnog u bilo koji sistem brojeva, s tim što se za deljenje i množenje koristi osnova tog drugog sistema (za oktalni 8, za heksadecimalni 16). Slede primeri konverzije brojeva.

Dekadni broj 55319,26074 želimo da pretvorimo u oktalni. Počinjemo sa celim delom broja i delimo ga sa 8 ($55319 / 8 = 6914,875$), onda zadržavamo celi deo rezultata (6914) i tražimo ostatak deljenja ($55319 - 6914 * 8 = 7$). Znači, prva cifra pre decimalnog znaka je 7.

Nastavljamo dalje – ceo proces računanja dat je u tabeli.

DEKADNI U OKTALNI		CELI DEO			DECIMALNI DEO			
Dekadno	55319.26074	55319 / 8	6914	7	0.260742 * 8	2.085938	2	0.085938
OSNOVA	8	6914 / 8	864	2	0.085938 * 8	0.6875	0	0.6875
Oktalna vrednost		864 / 8	108	0	0.6875 * 8	5.5	5	0.5
154027,2054		108 / 8	13	4	0.5 * 8	4	4	0
		13 / 8	1	5				
		1 / 8	0	1				

DEKADNI U HEKSADEC.		CELI DEO			DECIMALNI DEO			
Dekadno	1943850.797	1943850 / 16	121490	10	0.797104 * 16	12.75366	12	0.753662
OSNOVA	16	121490 / 16	7593	2	0.753662 * 16	12.05859	12	0.058594
Heksadecimalno		7593 / 16	474	9	0.058594 * 16	0.9375	0	0.9375
1DA92A,CC0F		474 / 16	29	10	0.9375 * 16	15	15	0
		29 / 16	1	13				
		1 / 16	0	1				

Veza binarnog, oktalnog i heksadecimalnog sistema

Već smo ranije napomenuli da su oktalni i heksadecimalni sistem pogodni za rad sa računarom, budući da su njihove osnove stepeni broja 2 ($2^3=8$ i $2^4=16$), koji je opet osnova binarnog sistema.

Ovo u praksi znači da je prevođenje brojeva iz binarnog u ove sisteme, i obrnuto, veoma jednostavno, skoro i bez računanja. Pogledajmo npr. binarni broj

1101001,1101

Ako grupišemo njegove cifre u grupe od po tri (uvek počinjemo od decimalnog znaka), možemo ga predstaviti na sledeći način (ne zaboravite – **ispred broja**, kao i **posle decimala**, možemo slobodno **ubacivati nule** – one neće menjati vrednost broja):

001 101 001 , 110 100

Svaka od ovih grupa u stvari predstavlja po jednu oktalnu cifru, koju možemo lako izračunati.

000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

Tako smo jednostavno preveli binarni broj u oktalni:

$$(1101001,1101)_2 = (151,64)_8$$

Na potpuno isti način prevodimo binarne brojeve u heksadecimalne. Jedina razlika je što se ovaj put binarne cifre grupišu u grupe od po četiri. Pogledajmo primer za isti binarni broj:

0110 1001 , 1101

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

To znači da je

$$(1101001,1101)_2 = (69,D)_{16}$$

pošto vrednost 13 odgovara heksadecimalnoj cifri D.

Inače, celi binarni brojevi, koji se sastoje iz svega nekoliko cifara, veoma se lako računaju, pošto su i jačine cifara mali brojevi. Evo kako se to radi za poslednji binarni broj iz gornjeg primera (1101):

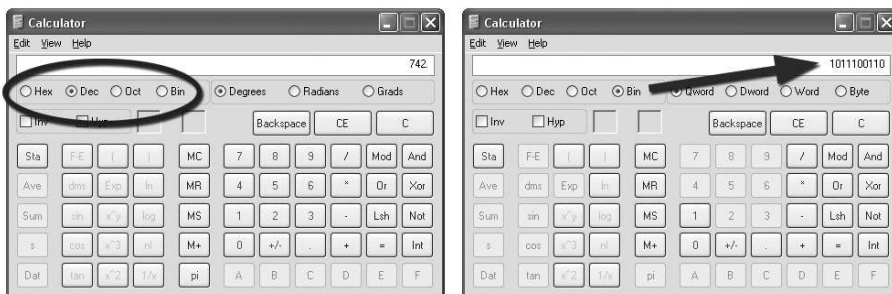
Jačina cifre	8	4	2	1
Cifra	1	1	0	1

U donji deo tablice se ubaci željeni broj, a onda se sabere one jačine kod kojih je binarna cifra jednaka 1. Tako imamo da je $8+4+1 = 13$.

Na potpuno isti način možemo prevoditi brojeve iz oktalnog ili heksadecimalnog sistema nazad u binarni – svaka cifra se pretvara u odgovarajuću grupu binarnih cifara.

Prevođenje brojeva pomoću Windows kalkulatora

Uz operativni sistem **Windows** dolazi i program **Calculator**, koji možemo koristiti kao pomoć pri konverziji brojeva iz jednog sistema u drugi. Sledeći primer ilustruje prevođenje broja iz dekadnog u binarni.



Sa leve strane kalkulatora se nalazi mogućnost izbora brojnog sistema koji želimo da koristimo. Na tom mestu izaberemo sistem iz koga želimo da preračunamo broj. Izabrali smo dekadni sistem i otkucali npr. broj 742.

Posle toga je dovoljno odabrati sistem u koji želimo da prevedemo broj (u ovom slučaju to je binarni). Kalkulator nam odmah ispisuje zadati broj u binarnom obliku, kao 1011100110.

Iako konverzija brojeva, koju nam nudi kalkulator, nije u potpunosti funkcionalna (moguća je samo za cele brojeve) i kao takva nam može biti od koristi.

06

Predstavljanje podataka na računaru

Sušтина ovog poglavlja može se svesti na jednu rečenicu: **svaki podatak na računaru je broj**.

U prošlom poglavlju smo predočili da postoje različiti sistemi brojeva. Osnovni sistem sa kojim rade računari je binarni, a zbog nepraktičnog zapisa binarnih brojeva (ogroman broj jedinica i nula kod većih brojeva), koriste se još i oktalni i heksadecimalni sistem.

Najmanji podatak koji se može zabeležiti na računaru je jedan **bit**. Bit može nositi informaciju o tome da li je nešto tačno ili netačno, da li je odgovor na neko pitanje „da“ ili „ne“, ili, ako je u pitanju broj, jedan bit može beležiti brojeve 0 i 1.

Niz od 8 bitova čini jedan **bajt**. Pošto jedan bit predstavlja jednu binarnu cifru (0 ili 1), znači da bajt može sadržati osmocifren binarni broj. Najmanji takav broj se sastoji od osam nula (što u svim brojnim sistemima ima vrednost 0), dok je najveći mogući osmocifreni binarni broj:

$$(11111111)_2 = (255)_{10} = (\text{FF})_{16}$$

To znači da u jednom bajtu možemo zabeležiti jedan ceo broj od 0 do 255. Takođe vidimo i da se svaki dvocifren heksadecimalni broj može smestiti u jedan bajt. Ovo je očigledno jedan relativno mali podatak. Ilustracije radi, jedan bajt može sadržati jedno slovo. 1024 bajta čini jedan **kilobajt**, 1024 kilobajta jedan **megabajt** i tako dalje. U tablici su predstavljene neke od mernih jedinica i njihove skraćenice.

1 B (1 bajt)	=	8 b (8 bitova)
1 kB (1 kilobajt)	=	1024 B (1024 bajta)
1 MB (1 megabajt)	=	1024 kB (1024 kilobajta)
1 GB (1 gigabajt)	=	1024 MB (1024 megabajta)
1 TB (1 terabajt)	=	1024 GB (1024 gigabajta)

Jedinice koje po redu veličine dolaze iznad terabajta su **petabajt**, **eksabajt**, **zetabajt** i **jotabajt**.

Međutim, za razliku od količine RAM memorije, neke oblasti računarstva nisu toliko vezane za binarni sistem. Na primer, radni takt procesora danas se izražava u gigahercima (GHz), a 1 GHz nema 1024, već 1000 MHz (megaherca). Isto tako, jedinice za brzinu prenosa podataka, koje se označavaju kao broj bitova prenetih u sekundi, **kilobit u sekundi** (kbps, kbit/s ili kb/s) ili **megabit u sekundi**, ne odnose se na 1024 bitova, već na 1000 bitova u sekundi (kilobit), odnosno milion bitova (megabit). Obratite pažnju da se pravi razlika u veličini slova koje označava bit ili bajt, što znači da se razlikuju skraćenice za kilobit (**kb**) i kilobajt (**kB** ili **KB**).

1 B (bajt)	=	8 b (bitova)		
1 kb (kilobit)	=	1000 b (bitova)	=	125 B (bajtova)
1 Mb (megabit)	=	1000 kb (kilobitova)	=	oko 122 kB (kilobajta)

Dvosmislenost postaje još veća ako se uzme u obzir da proizvođači hard diskova i DVD-ova danas kapacitete označavaju u gigabajtima, ali računajući da 1 GB ima 1000 MB, umesto 1024.

Zbog ovoga je pre nekoliko godina organizacija IEC (*International Electrotechnical Commission*) predložila standard po kome bi se redovi veličina koje se koriste po SI sistemu (kilo, mega, giga...) uvek odnosile na dekadne redove veličina (što bi značilo da 1kB ima 1000 bajtova), dok bi se za binarne redove veličina koristile drugačije oznake: **kibi** (*kilobinary*), **mebi** (*megabinary*), **gibi** (*gigabinary*), **tebi** (*terabinary*)...[14] Po tome bi jedan kibibajt imao 1024 bajta. Iako je standard usvojen, još uvek nije ušao u širu upotrebu (za sada samo neke verzije operativnog sistema Linux koriste ovakvu notaciju). Sledeća tabela daje uporedni prikaz ovih veličina:

1 kB (kilobajt) = 1000 B (bajta)	1 KiB (kibibajt) = 1024 B (bajta)
1 MB (megabajt) = 1000 kB (kilobajta)	1 MiB (mebibajt) = 1024 KiB (kibibajta)
1 GB (gigabajt) = 1000 MB (megabajta)	1 GiB (gibibajt) = 1024 MiB (mebibajta)
1 TB (terabajt) = 1000 GB (gigabajta)	1 TiB (tebibajt) = 1024 GiB (gibibajta)

Zbog trenutne slabe prihvaćenosti standarda u praksi, u tekstu će se kilobajt, megabajt ili gigabajt ipak odnositi na binarne redove veličina, odnosno 1 kB će se podrazumevati kao 1024 bajta, 1 MB kao 1024 KB itd.

Zašto baš 1024?

Sada konačno možemo da odgovorimo na (večito) pitanje – ako jedan kilogram ima hiljadu grama, a jedan kilometar hiljadu metara, zašto onda jedan kilobajt ima baš 1024 bajta?

Odgovor proizlazi iz načina kako računar radi i beleži podatke. Budući da se svi podaci beleže u obliku binarnih cifara (nula i jedinica), logično je da računar suštinski koristi binarni brojni sistem. Dok su u dekadnom sistemu okrugli brojevi 10, 100 ili 1000, za računar su to stepeni dvojke – 2, 4, 8, 16... Tako je i broj 1024 u stvari 2^{10} .

Isto tako i količina memorije u našim računarima ne iznosi 250 ili 500 megabajta, već su to uvek vrednosti 256 MB (2^8), 512 MB (2^9), 4 GB (2^2) i sl.

Predstavljanje brojeva

Računar može raditi sa celim i racionalnim brojevima (brojevi koji imaju decimale). Celi brojevi se beleže u određenom broju bajtova. Od toga koliko bajtova se izdvoji za beleženje broja zavisi i opseg – koji je najveći broj koji može biti zabeležen. Kod predstavljanja celih brojeva razlikujemo:

- neoznačene brojeve (samo pozitivni brojevi, počev od nule);
- označene brojeve (i pozitivni i negativni celi brojevi).

Na primer, ukoliko je broj 16-bitni, znači da je za njegovo beleženje potrebno izdvojiti 2 bajta (8 + 8 bitova). Pogledajmo opseg neoznačenih brojeva koji mogu biti zabeleženi u ovoliko memorije:

	najmanja vrednost	najveća moguća vrednost
binarno	00000000 00000000	11111111 11111111
heksadecimalno	0000	FFFF
dekadno	0	65535

Drugim rečima, ako se broj beleži unutar N bitova, opseg će biti od 0 do 2^N-1 . Za 16 bitova tako imamo da je $2^{16} = 65536$, što znači da je opseg od 0 do 65535.

Međutim, ukoliko su brojevi označeni, onda je jedan bit potrebno izdvojiti za znak broja, što znači da se najveći broj koji je moguće zapisati u istom broju bitova prepolovljuje.

Po nekoj logici, negativni brojevi trebalo bi da budu manji od pozitivnih, ali ispostavlja se da ako označen broj 32767, zapisan u 2 bajta, povećamo za 1, dobijamo najniži negativan broj -32768. Ovo naizgled nelogično ponašanje je lako objašnjivo ako primetimo da je kod negativnih brojeva najjači, 16-ti bit, uvek postavljen na 1.

	negativni brojevi	pozitivni brojevi
binarno	10000000 00000000 do 11111111 11111111	00000000 00000000 do 01111111 11111111
heksadec.	8000 do FFFF (-8000 do -0001)	0000 do 7FFF
dekadno	-32768 do -1	0 do 32767

Ipak, negativni brojevi se ne dobijaju tako što se pozitivnom broju najviši bit samo postavi na 1. Da bismo dobili ispravan negativan broj, moramo ga izračunati kao tzv. **potpuni komplement** zadatog pozitivnog broja. Na sledećem primeru izračunavamo broj -5:

Broj 5 u 16-bitnom binarnom zapisu	00000000 00000101
Nepotpuni komplement (invertujemo binarne cifre: 0 postaje 1, a 1 postaje 0)	11111111 1111010
Potpuni komplement broja 5 (nepotpuni sabiramo sa 1) predstavlja broj -5	11111111 1111011

Znači -5 nije kako bismo pretpostavili 10000000 00000101, već **11111111 1111011**.

Osim celih brojeva, računar može raditi i sa **racionalnim (decimalnim) brojevima**. Postoji više načina da se ovi brojevi predstave, ali jedan od najznačajnijih je takozvani prikaz **brojeva u pokretnom zarezu**.

Brojevi tzv. **jednostruke (single) tačnosti** beleže se u 4 bajta, ali na sasvim drugačiji način u odnosu na cele brojeve. Naime, ovi brojevi se pamte beleženjem **mantise i eksponenta**:

Z	7 bitova za eksponent	24 bita za apsolutnu vrednost mantise (u najvišem bitu (Z) beleži se znak)

Drugim rečima, kao $M * 2^E$.

Na ovom mestu nećemo ulaziti u načine kodiranja mantise i eksponenta. Sve što treba da znamo jeste da kod ovih brojeva opseg (od najvećeg do najmanjeg broja) ne igra veliku ulogu. Naime, na ovaj način je moguće predstaviti izuzetno velike (kada je eksponent pozitivan), kao i izuzetno male brojeve (negativan eksponent).

Ono što nas interesuje u vezi brojeva u pokretnom zarezu je njihova **preciznost (tačnost)**. Preciznost broja zavisi od broja bitova koji je određen za beleženje mantise, a u slučaju broja jednostruke tačnosti taj broj je oko 7 dekadnih cifara.

Zamislimo na trenutak broj koji može da zabeleži samo tri precizne cifre u svojoj mantisi i opseg eksponenta od -9 do 9. Pogledajmo primer veoma velikog i veoma malog broja:

$0,123 * 10^9$	123000000,0000...
$0,123 * 10^{-9}$	0,0000000001230000...

Prvi primer pokazuje veoma veliki broj, ali neprecizan – sve značajne cifre se nalaze na početku broja, a ostatak su nule. Isto važi i za drugi primer – izuzetno mali broj, kod koga su sve **značajne cifre grupisane**, dok su sve ostale cifre jednake nuli.

Osim brojeva jednostruke preciznosti, postoje i brojevi dvostruke (*double*) preciznosti (64 bita), pa i četvorostruke (*quadruple*) preciznosti koji se zapisuju u 128 bitova.

Za važna naučna i matematička istraživanja često su potrebni i veći i precizniji brojevi. Čak iako procesor ne podržava ovako velike brojeve, postoje programi i programerske biblioteke koji

uspešno simuliraju izuzetno velike vrednosti brojeva i veoma visoku preciznost. Rad sa ovako velikim brojevima je svakako sporiji, ali što se tiče njihove veličine, ograničeni su samo slobodnom memorijom računara.

Predstavljanje teksta

Kada je tekst u pitanju, kodiranje je relativno jednostavno – svakom simbolu, bez obzira da li je u pitanju slovo, cifra, kontrolni znak, znak interpunkcije ili bilo koji drugi znak koji se može pojaviti u tekstu, dodeljuje se odgovarajući broj.

Pored IBM standarda EBCDIC, najznačajniji standard za kodiranje teksta tokom dugog niza godina bio je ASCII (*American Standard Code for Information Interchange*). Standardni ASCII set sastoji se iz 128 simbola i specijalnih znakova označenih brojevima od 0 do 127. Ovo znači da se svaki znak može predstaviti nizom od 7 bitova.

Tako je u ASCII standardu definisano da se veliko slovo **A** zamenjuje brojem **65**, **razmak** brojem **32**, **tačka** brojem **33**, prelazak u novi red brojem 13 (*carriage return*) ili kombinacijom 13 i 10 (*carriage return i line feed*) itd.

Pošto bi bilo praktično iskoristiti ceo bajt za beleženje simbola (svih 8 bitova), kreiran je i prošireni ASCII standard u kome je dodato još 128 znakova (posebna slova iz francuskog i nemačkog jezika, matematički i grafički simboli) i tako je nastala ASCII tabela sa 256 znakova. Svaki znak je bio zabeležen u tačno jednom bajtu. Na ovaj način bi ceo roman „Hari Poter i relikvije smrti“ (kao čist tekst) zauzeo oko 1,1 MB memorije.

Problem koji se vrlo brzo pojavio bio je nedostatak znakova iz pisama drugih svetskih jezika. Primer za to su svakako ruski i grčki jezik, a na našim prostorima je bio očigledan nedostatak srpskih latiničnih slova i ćirilice.

Prvi pokušaj rešavanja ovog problema bio je kroz kreiranje tzv. YUSCII fontova. Drugim rečima, kreirani su fontovi u kojima su pojedini znakovi osnovnog ASCII standarda zamenjivani nedostajućim slovima (npr. simbol „@“ zamenjen je slovom „Ž“). Problem je bio u tome što su u tekstu ovi znaci i dalje ostajali to što jesu, bio je promenjen samo njihov izgled u različita slova. Bitno je znati da su ovi fontovi i dalje prisutni – postoji veliki broj različitih vrsta. Generalni savet je da se nikako ne koriste prilikom rada sa tekstem, već, ako je neophodno, samo u grafičkom dizajnu, unutar slike.

ASCII	YUSCII	ASCII	YUSCII
@	Ž	`	ž
[Š	{	š
\	Đ		đ
]	Č	}	č
^	Ć	~	ć
Q	Љ	q	љ
W	Њ	w	њ
X	Ћ	x	ћ

Tabela 6.1. Uporedni prikaz YUSCII znakova i ASCII karaktera čije pozicije zauzimaju

Rešenje koje se održavalo godinama i koje je bilo podržano od strane operativnog sistema bile su **kodne strane**. Princip se svodio na to da se znakovi iz proširenog dela ASCII standarda (znači posle 127-og znaka) zamenjuju simbolima nacionalnih pisama.

Za nas su od značaja kodne strane CP852 (latinica) i CP855 (ćirilica) pod MS-DOS operativnim sistemom, odnosno CP1250 (latinica) i CP1251 (ćirilica) pod Windows-om (*CP – code page*).

Šta je to font?

Najprostije rečeno, font predstavlja „rukopis“, odnosno oblik slova na računaru. Danas postoji veliki broj fontova, koje možemo podeliti na više načina, a uobičajeno je na:

- serif fontove** – fontovi u kojima slova imaju „postolja“. Tipičan predstavnik je Times grupa fontova od kojih je najpoznatiji verovatno **Times New Roman**.
- sans-serif fontove** – fontovi sa slovima bez „postolja“. Najpoznatija je „porodica“ Helvetica fontova od kojih najčešće koristimo **Arial** font.

Iako sans-serif fontovi deluju preglednije, za pisanje teksta preporučuje se korišćenje Times fontova, pošto „postolja“ vizuelno kreiraju liniju koja olakšava čitanje.

Druga podela bila bi na:

- fontove varijabilne širine** – fontovi u kojima svako slovo zauzima samo onoliko širinu koliko mu je potrebno. To znači da slova **I** i **W** zauzimaju različito prostora. Većina fontova spada u ovu grupu.
- fontove fiksne širine** – fontovi u kojima svako slovo zauzima jednak prostor, čime se dobija efekat „pisače mašine“. Verovatno najpoznatija grupa ovih fontova su **Courier fontovi**.

Pošto različiti znakovi, unutar različitih kodnih strana, imaju iste kodove (menjaju se u iste brojeve), dolazi do konfuzije u smislu koje znakove treba prikazati. Zbog toga se u nekim programima prilikom podešavanja fonta za prikaz teksta obično mora navesti i koja kodna strana se koristi (*Western, Central European, Russian, Cyrilic...*).

Danas su kodne strane i ASCII standard zastareli i zamenjeni novim UNICODE standardom. Ideja UNICODE-a je da se u okviru jednog standarda objedine svi znakovi nacionalnih pisama, za šta su 256 mogućih znakova ASCII standarda jednostavno suviše tesni. Tako UNICODE za zapis jednog znaka koristi dva bajta (16 bitova, što daje ukupno $2^{16} = 65536$ mogućih znakova) ili u proširenoj verziji i do 4 bajta (preciznije, unutar UNICODE verzije 5.1 standarda, rezervisano je od 00000000 do 0010FFFF, odnosno 1.114.112 pozicija za znakove a upotrebljeno je više od 100.000). Ovoliki broj znakova obuhvata svetske jezike, matematičke simbole, muzičku notaciju, drevna pisma i sl. Ipak, za sada je na računarima najviše u upotrebi 2-bajtna, odnosno osnovna multijezička verzija (*BMP – Basic Multilingual Plane*).[15]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Slika 6.1. Ovo je samo jedan mali deo znakova koji su sadržani u UNICODE standardu. Na slici su prikazane latinica, ćirilica, japanska katakana, runsko pismo i glagoljica.[15]

Pored svih prednosti, standard ima i jednu očiglednu manu – sada je za zapis istog teksta potrebno više memorije nego ranije (2 ili 4 bajta za svaki znak, umesto dosadašnjeg jednog bajta). Osim toga, stariji programi i operativni sistemi mogu imati problema sa UNICODE kodiranim tekstem. Zbog toga su osmišljene različite varijante za „pakovanje“ UNICODE-a, od kojih je verovatno najkorišćeniji standard za kodiranje **UTF-8**. Ovaj standard se zasniva na ideji

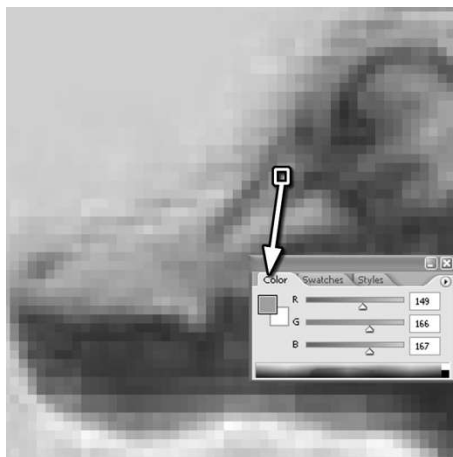
da je najveći deo teksta pisan običnom engleskom latinicom, pa se ti znakovi i dalje beleže u po jednom bajtu. Zato postoje grupe kodova koje definišu kada se koriste 2, 3 ili više bajtova za zapis ostalih znakova.[16]

U sledećoj tabeli možemo videti primer kodiranja teksta „Turistička škola“ prema nekoliko standarda (brojevi su dati u heksadecimalnom obliku).

	T	u	r	i	s	t	i	c	k	a		s	k	o	l	a
ASCII	54	75	72	69	73	74	69	63	6B	61	20	73	6B	6F	6C	61
	T	u	r	i	s	t	i	č	k	a		š	k	o	l	a
CP 1250	54	75	72	69	73	74	69	E8	6B	61	20	9A	6B	6F	6C	61
UNICODE	0054	0075	0072	0069	0073	0074	0069	010D	006B	0061	0020	0161	006B	006F	006C	0061
UTF-8	54	75	72	69	73	74	69	C48D	6B	61	20	C5A1	6B	6F	6C	61
	T	y	p	i	c	t	i	č	k	a		š	k	o	l	a
CP 1251	D2	F3	F0	E8	F1	F2	E8	F7	EA	E0	20	F8	EA	EE	EB	E0
UNICODE	0422	0443	0440	0438	0441	0442	0438	0447	043A	0430	0020	0448	043A	043E	043B	0430
UTF-8	D0A2	D183	D180	D0B8	D181	D182	D0B8	D187	D0BA	D0B0	20	D188	D0BA	D0BE	D0BB	D0B0

Predstavljanje grafike

Kao što znamo, slika koju vidimo na monitoru sastoji se iz velikog broja tačaka (piksela). Što tačkica ima više, one su manje i slika izgleda prirodnije.



Slika 6.2. Na uvećanom detalju ove fotografije vidimo da se ona u stvari sastoji iz velikog broja tačkica, od kojih je jedna izdvojena za primer – njena boja dobijena je kao RGB kombinacija (149, 166, 167).

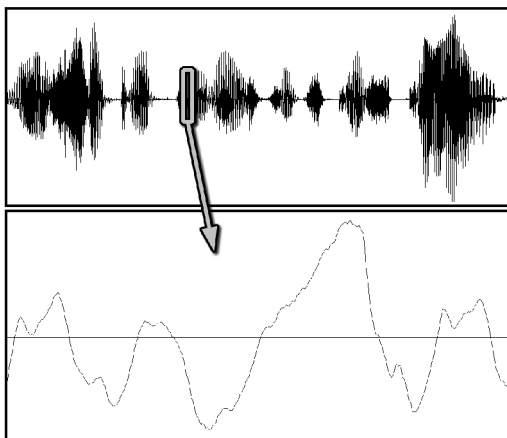
Svaka tačka ima svoju boju. Za prikaz na ekranu koristi se trobojni (**RGB – Red-Green-Blue**) kolorni sistem. To znači da se svaka boja formira kao kombinacija tri osnovne boje – **crvene, zelene i plave**. Ako su sve tri komponente na minimumu, u pitanju je crna boja, sve tri komponente na maksimumu daju belu boju, pojačane crvena i zelena daju žutu boju, itd.

RGB sistem za prikaz boja na ekranu je osmобitni, što podrazumeva da je za svaku od tri komponente odvojeno po 8 bitova, odnosno 1 bajt. Najmanja vrednost koja se može smestiti u jedan bajt je 0, a najveća FF, odnosno 255. Maksimalan broj različitih boja, odnosno kombinacija komponenti koji je moguć, iznosi $256 * 256 * 256$, odnosno otprilike 16,7 miliona boja – takozvani *true color*, što znači da je za prikaz jedne tačkice potrebno izdvojiti 3 bajta. Na taj način se boja svakog piksela kodira u brojeve, pa se tako i slika na računaru jednostavno beleži kao niz bajtova.

Programi za grafiku obično nude izbor nekoliko različitih kolornih sistema (engl. *colorspace*) za kreiranje slike (RGBA, RGB-16, HSV, CMYK...), ali treba da znamo da se, bez obzira na to, slika za prikaz na ekranu uvek pretvara u RGB. Inače, ove druge sisteme boja koriste grafički profesionalci za posebne namene (priprema za štampu, teksture za igre i sl.)

Predstavljanje zvuka

Zvuk predstavlja talas koji ima frekvenciju i amplitudu. Da bismo zvuk mogli da zabeležimo na računaru, potrebno je da ove dve veličine aproksimiramo u digitalni oblik. Za to služe uređaji koji spadaju u grupu analogno-digitalnih konvertera (*ADC – Analog to Digital Converter*). Naravno, za reprodukciju zvuka potreban je digitalno-analogni konverter (*DAC*).



Na slici se vidi grafička reprezentacija zvučnog signala u trajanju od 1,7 sekundi. Na donjoj slici se vidi jedan mali deo (stoti deo sekunde) ovog audio-zapisa.

Da bi računar zabeležio zvuk, potrebno je da mnogo puta u sekundi beleži stanje zvučnog talasa. Što više puta u sekundi se izvrši ovo beleženje i sa što preciznijim merenjem amplitude, to će i zvuk biti kvalitetniji.

Učestalost kojom se vrši beleženje zvuka je **frekvencija** (*sampling rate*), dok je broj nivoa zvučnog talasa koji može da se zabeleži **preciznost** (*sampling precision*). [17]

Zvučni zapis sa slike zabeležen je u frekvenciji od 44100 Hz i sa preciznošću od 16 bitova. To u stvari znači da je ADC uređaj (zvučna karta u računaru) preko 44000 puta u jednoj sekundi zabeležio trenutno stanje zvučnog talasa i to stanje svaki put zapisao kao podatak od 2 bajta, odnosno kao broj u rangi od 0 do 65535. Ovo je kvalitet koji odgovara muzici snimljenoj na CD-u.

Jedna prosečna pesma, od npr. 3 minuta, u ovom istom kvalitetu, u memoriji tako zauzima:

$$44100 \text{ semplova} * 180 \text{ sekundi} * 2 \text{ bajta} * 2 \text{ kanala (stereo)} = 30,3 \text{ MB}$$

Naravno, u praksi, zahvaljujući raznim metodima kompresije, ovaj zvučni zapis može biti i preko 10 puta manji.

Funkcionalan računar se sastoji iz dve komponente:

- hardvera (*hardware*) i
- softvera (*software*).

Do sada smo se upoznali sa fizičkom komponentom računara, odnosno hardverom. Sam po sebi, hardver ne znači mnogo bez programa koji ga pokreću i podataka sa kojima programi rade. Upravo ti programi i podaci čine **softverski sistem** računara.

Softver se deli na tri značajne grupe programa:

- sistemski softver – operativni sistem računara;
- programski softver – programi koji služe za pisanje drugih programa;
- aplikativni softver – programi pomoću kojih koristimo računar u bilo koje druge svrhe.

Pojam i struktura operativnog sistema

Operativni sistem predstavlja paket sistemskih programa koji služe za kontrolu hardvera, kontrolu softvera, kao i za povezivanje hardvera, softvera i korisnika.

Može se reći da operativni sistem predstavlja softver koji je potreban korisniku kako bi upravljao računarom, koristio uređaje spoljne memorije i pokretao ostale programe. Moderni operativni sistemi sastoje se iz nekoliko komponenti:

- jezgro,
- sistemske biblioteke,
- korisnički interfejs,
- mrežni rad i serverski programi,
- pomoćni programi.

Jezgro operativnog sistema

Najniži nivo, sama osnova operativnog sistema, naziva se **jezgro**, odnosno **kernel**. Svi ostali delovi operativnog sistema oslanjaju se na kernel i funkcionišu zahvaljujući njemu. Kernel se sastoji iz nekoliko delova, od kojih su najvažniji podsistemi za upravljanje [19]:

- procesima,
- memorijom,
- ulazom / izlazom,
- datotekama.

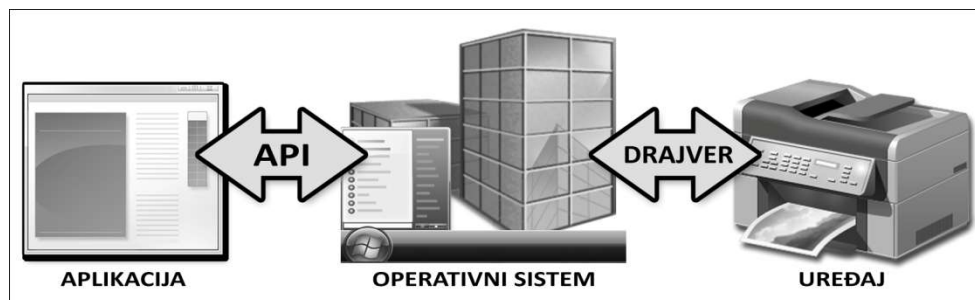
Podsistem za upravljanje procesima služi za kontrolu programa koji se izvršavaju. Današnji operativni sistemi omogućuju rad više programa istovremeno, koristeći tzv. **višeprocesni rad** (*multitasking*). Svaki program se izvršava u zasebnom **procesu** i, kao da to nije dovoljno, svaki program može razgranati svoje izvršavanje u više **нити** (*threads*). Zbog toga je potreban visok nivo kontrole kako bi svi programi imali pristup računarskim resursima i procesorskom vremenu. Da bi se sprečilo da neki program, greškom ili namerno, potpuno preuzme računar, koristi se tzv. **preventivni višeprocesni rad** (*preemptive multitasking*), pomoću koga operativni sistem može i nasilno prekinuti izvršavanje programa i dodeliti resurse drugom programu.[20]

U bliskoj vezi sa kontrolom procesa je i **podсистem za upravljanje memorijom**. Kontrolisanje paralelnog izvršavanja programa je mnogo složenije ako se ima u vidu da jedan program može prebrisati memorijski prostor koji koristi drugi program, što u najboljem slučaju dovodi do blokiranja samih programa, a u najgorem do rušenja celog operativnog sistema. Zbog toga se danas programi izvršavaju unutar zasebnih **virtuelnih mašina**, koje predstavljaju simulaciju „računara u računaru“. Tako je svaki program smešten u svoj „bazen sa peskom“ (*sandbox*) i „misl“ da je jedini koji se u tom trenutku izvršava na računaru. Na ovaj način programi su odvojeni jedan od drugog, a konfliktne zahteve za memorijom i hardverom rešava sam operativni sistem.

Podsistem za upravljanje ulaznim i izlaznim operacijama u stvari predstavlja deo operativnog sistema pomoću koga programi pristupaju perifernim uređajima računara. Ovaj pristup omogućen je preko posebnih upravljačkih programa – **drajvera**.

Kompjuter koji danas nazivamo PC računarom sastavljen je od komponenti velikog broja različitih proizvođača. Zbog toga je ranije svaki program morao da bude pravljen tako da podržava veći broj različitih ulazno-izlaznih uređaja. Tako bi jedna igra pisana za operativni sistem DOS morala da podržava različite grafičke i zvučne kartice. Današnji operativni sistemi stavljaju tačku na takvo programiranje, pošto programima pružaju standardizovani interfejs za upravljanje hardverom.

Drugim rečima, ako bi program u jednom trenutku trebalo da odsvira neki zvuk, ne bi direktno pristupao zvučnoj kartici, već bi se obratio operativnom sistemu koji bi, putem drajvera za tu konkretnu zvučnu karticu, reprodukovao zvuk.



Slika 7.1. Povezivanje softvera i hardvera preko operativnog sistema

Drajvere može isporučivati proizvođač operativnog sistema ili proizvođač uređaja. Obično su drajveri koje kreira proizvođač uređaja noviji i omogućuju korišćenje punih mogućnosti uređaja, ali su takođe i više podložni greškama, koje, budući da se radi o vitalnom delu operativnog sistema, mogu veoma loše da utiču na stabilnost celog sistema.

Ovaj podsistem u sebe uključuje još jednu važnu funkciju, a to je **upravljanje prekidima** (*interrupts*). Kada određeni uređaj (ulazni, odnosno ulazno-izlazni) prosleđuje podatke

računaru, to radi preko zahteva za prekidom (*IRQ – interrupt request*). Tada procesor, odnosno kontroler na matičnoj ploči, „obradi pažnju“ na uređaj kako bi primio podatke. Najjednostavniji primer ovakvog funkcionisanja je komunikacija tastature sa računarom.

Još jedna veoma bitna funkcija operativnog sistema jeste pristup uređajima spoljne memorije, za koji je zadužen **podsystem za upravljanje datotekama** (fajlovima). Ovaj podsystem se oslanja na podsystem za kontrolu ulaza/izlaza, s tim što u sebe uključuje i fajl sistem koji vrši alociranje fajlova na konkretnom uređaju spoljne memorije.

Sistemske biblioteke

Korisnički programi komuniciraju sa operativnim sistemom putem tzv. **sistemskih poziva**. Sistemski poziv je potreban svaki put kada korisnički program želi da izvrši neku operaciju koja je pod kontrolom operativnog sistema. Npr. da bi se pod Windows operativnim sistemom program prikazao korisniku, potrebno je da se otvori prozor. Program ne može sam da otvori prozor na ekranu, već poziva posebnu funkciju operativnog sistema (uz zadate parametre).

Ova komunikacija se odvija preko posebnih **API**-ja. **API** (*Application Programming Interface*) predstavlja biblioteku funkcija, konstanti i definicija tipova preko kojih programi mogu komunicirati sa operativnim sistemom, hardverom i sl. Ove funkcije su obično grupisane prema nekom određenom zadatku kome su posvećene. Svaki ozbiljan programski jezik namenjen pisanju programa za Windows pokriva WinAPI, preko koga se obavljaju sistemski pozivi. Jedan od najpoznatijih bio bi DirectX API, koji služi za programiranje igara i multimedijalnih aplikacija.

```
hwnd = CreateWindow( "GenericAppClass", "Generic Application",
    WS_OVERLAPPEDWINDOW|WS_HSCROLL|WS_VSCROLL,
    0, 0, CW_USEDEFAULT, CW_USEDEFAULT,
    NULL, NULL, hInstance, NULL );

ShowWindow( hwnd, nCmdShow );
```

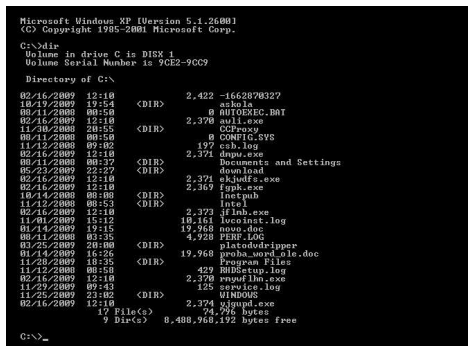
U okviru se vidi primer poziva WinAPI funkcija u jeziku C++ za kreiranje i prikaz prozora na ekranu. Laiku navedeni redovi možda izgledaju komplikovano, ali programeri veoma cene što mogu da se oslone na Windows kako bi u svega par redova programa dobili prozor u kome se izvršava aplikacija.

Korisnički interfejs

Korisnički interfejs predstavlja način na koji program (u ovom slučaju operativni sistem) komunicira sa korisnikom. Ukoliko je korisnički interfejs jednostavan za korisnika, kaže se da je „prijateljski“ (*user-friendly*). Postoje dva osnovna tipa korisničkog interfejsa:

- tekstualni i
- grafički.

Tekstualni korisnički interfejs se danas retko sreće (eventualno na računarima specijalne namene). U ovakvom sistemu korisnik je morao da komunicira sa računarom zadajući mu naredbe preko komandne linije (*command prompt*). Korisnici su morali da znaju makar nekoliko osnovnih naredbi i njihovu sintaksu kako bi mogli da pronalaze i pokrenu programe sa diska.



Operativni sistem MS-DOS, koji je bio jedan od najkorišćenijih na PC računarima, imao je tekstualni korisnički interfejs. Na slici se vidi primer komande DIR koja služi za listanje sadržaja direktorijuma.

Grafički korisnički interfejs (GUI – Graphical User Interface) koristi vizuelne elemente tipa prozora, menija, dugmića, ikona i sl. Osim tastature, kao primarni ulazni uređaj za zadavanje komandi koristi se i miš.



Slika 7.2. Izgled operativnog sistema Mac OS Lion (Apple) [59]

Zahvaljujući elementima korisničkog interfejsa koje nudi operativni sistem, svaka aplikacija može kreirati sopstveni korisnički interfejs bez velikog napora programera. Na ovaj način operativni sistem obezbeđuje komunikaciju korisnika i aplikacije. Sledeći koraci opisuju redosled operacija koje se odvijaju dok radimo u programu za obradu teksta (izazvane pritiskom samo jednog tastera).

- Pritisak tastera na tastaturi kreira signal koji se preko zahteva za prekidom prenosi računaru.
- Podatak o pritisnutom tasteru se putem upravljačkog programa (drajvera) za tastaturu prenosi do operativnog sistema.
- Operativni sistem kreira tzv. **dogadaj** (*event*), koji se preko API-ja prosleđuje aplikacijama koje su aktivne.
- Ovakav događaj obično reguliše aplikacija koja je u prvom planu, koja onda preuzima podatak o pritisnutom tasteru i proglašava događaj rešenim (u suprotnom bi se tekst koji kucamo npr. u *Word-u* pojavljivao i u svim programima koji su aktivni u pozadini).

Potom sledi reakcija aplikacije na pritisnut taster.

- Aplikacija za ispis teksta tumači podatak dobijen sa tastature kao znak koji treba ispisati.
- Aplikacija preko API-ja prosleđuje zahtev operativnom sistemu za iscrtavanje određenog znaka u svom prozoru.
- Operativni sistem prihvata ovaj zahtev i na osnovu ranije zadatih karakteristika prozora (font, veličina slova...) definiše sliku u memoriji (već smo pisali o tome kako računar pamti grafičke podatke), koja treba da se pojavi na određenoj poziciji na ekranu.
- Operativni sistem se zatim obraća grafičkoj karti putem njenog drajvera i zadaje nov grafički prikaz.
- Grafička karta formira nov prikaz ekrana u svojoj memoriji i prosleđuje ga monitoru.

Na sličan način funkcioniše sva komunikacija između korisnika i aplikacije, koju omogućuje korisnički interfejs operativnog sistema.

Mrežni rad i serverski programi

Već duže vreme računari služe i kao uređaji za povezivanje i komunikaciju koja se odvija nekim tipom međusobnog umrežavanja. Mogućnosti za povezivanje su takođe implementirane u operativnom sistemu, počev od drajvera za komunikacione uređaje (modem, mrežnu kartu), preko protokola za razmenu podataka (npr. TCP/IP za povezivanje preko Interneta), pa sve do osnovnih klijentskih aplikacija (telnet, elektronska pošta, pregled web-a).[18]

Ove klijentske aplikacije mogu biti zamenjene drugim, naprednijim programima za korišćenje Interneta (npr. *Mozilla Firefox* umesto *Internet Explorer-a*), ali da bi funkcionisali, neophodna im je osnovna podrška za mrežu koju pruža operativni sistem.

Za razliku od uobičajenih sistema sa kojima radimo svaki dan, postoji posebna klasa operativnih sistema koji se koriste na računarima koji zauzimaju centralno mesto u mreži i služe za opsluživanje svih ostalih računara. To su tzv. **serverski operativni sistemi**, koje razlikujemo od sistema namenjenih običnim korisnicima i koji se često nazivaju **klijentski** (*workstation*). Za razliku od klijentskog operativnog sistema, serverski OS ne mora da ima vrhunsku podršku za grafiku i multimediju, kao ni razvijen korisnički interfejs, pošto nije namenjen stalnom radu korisnika. Idealna situacija za administratora je da se server jednom podesi i ostavi da radi.

Od serverskog sistema se zahteva veća stabilnost u radu, kao i povećana bezbednost. Serverski sistemi u sebe uključuju i veliki broj administrativnih programa koji se koriste za definisanje korisnika i računara u mreži, naprednije mrežne protokole (npr. dodeljivanje IP adresa klijentskim računarima, vezivanje domena sa IP adresama, povezivanje lokalne mreže sa Internetom) i određen broj serverskih programa (web server, FTP server, mail server).

Način funkcionisanja servera svodi se na neprekidno „oslušivanje“ zahteva od strane klijenata u mreži. Kada server dobije zahtev (npr. za nekom web stranicom), aktivira se i prosleđuje klijentu traženi sadržaj.

Pomoćni programi

Osim navedenih programa neophodnih za funkcionisanje računara, operativni sistem poseduje i niz pomoćnih programa koji korisniku služe za održavanje sistema. Ovi programi obično dolaze uz operativni sistem, ali za većinu njih postoje alternative nezavisnih proizvođača koje su često kvalitetnije – jednostavnije za korišćenje i/ili moćnije.

Ovi programi korisniku služe za:

- **rad sa spoljnom memorijom** (defragmentacija ili particionisanje hard diska, kopiranje ili snimanje CD/DVD medija);
- **rad sa fajlovima** (kopiranje, premeštanje, arhiviranje, bekap, kompresija, deljenje ili spajanje fajlova);
- **održavanje sistema** (instalacija i deinstalacija programa, podešavanje opcija sistema, merenje performansi, brisanje nepotrebnih fajlova);
- **bezbednost** (antivirusi, *firewall* programi).

Postoji takođe i jedan skup alata koji spadaju pre u aplikativni nego u sistemski softver, a koji se instaliraju zajedno sa operativnim sistemom. To su programi za rad sa tekstom, grafikom, reprodukciju video klipova i muzike, komunikaciju putem Interneta i njima slični. Ovi programi

obično imaju manju funkcionalnost od komercijalno dostupnih programa i korisniku obično služe kao „prva pomoć“ dok ne instalira specijalizovane aplikacije za te konkretne namene.

Tipovi operativnih sistema

Na osnovu računara koje kontrolišu i zadataka koje obavljaju, operativne sisteme možemo generalno podeliti na 4 grupe.

Operativni sistemi u realnom vremenu (*RTOS – Real-Time Operating Systems*) se obično koriste za kontrolisanje mašina, naučnih uređaja i industrijskih sistema (najčešće ih danas koriste računari u automobilima, TV ili DVD uređajima, roboti, svemirske letelice). Ovi sistemi su usko povezani sa mašinom, namenjeni strogo određenom zadatku i poseduju veoma malo ili nimalo mogućnosti za komunikaciju sa krajnjim korisnikom. Glavna osobina ovih sistema je njihova mogućnost da veoma precizno određuju redosled i trajanje operacija uređaja. Ove operacije ne bi smele da se odvijaju ni previše brzo ni previše sporo.

Jednokorisnički jednoprocensni sistemi (*Single-User Single-Task*) su operativni sistemi kod kojih jedan korisnik može koristiti računar, radeći u jednom programu u jednom trenutku. Primer ovakvog sistema bio bi stari MS-DOS operativni sistem, a u današnje vreme Palm OS, operativni sistem za Palm računare.

Jednokorisnički multitasking sistemi (*Single-User Multi-Tasking*) su danas najkorišćeniji operativni sistemi na kućnim računarima. Njihova karakteristika je da pod njima u jednom trenutku može raditi jedan korisnik, ali sa više programa istovremeno. Najbolji primer ovakvog sistema je Microsoft Windows.

Višekorisnički sistemi (*Multi-User*) su operativni sistemi koji podržavaju rad više korisnika u isto vreme. Ovi sistemi svakom korisniku dodeljuju određeni deo procesorskog vremena i računarskih resursa. Tipičan primer ovakvog sistema je UNIX. Ne treba mešati višekorisničke operativne sisteme sa serverima koji podržavaju više umreženih korisnika. Korisnici koji se svojim računarima povezuju na server (npr. Windows 2003) ne rade zaista na samom serverskom računaru, već server samo odgovara na njihove zahteve šaljući im tražene podatke.

Moderni operativni sistemi

Najveći broj današnjih operativnih sistema može se svrstati u dve velike grupe – Unix i Windows.

Unix predstavlja čitavu porodicu operativnih sistema koji vuku poreklo iz originalnog *Unix* operativnog sistema koji je 1969. godine razvila *Bell* laboratorija. U ove sisteme spadaju *Version 7 Unix* ili *System V Unix* (koji se deklarišu kao tradicionalni *Unix* sistemi), ali i *BSD* i *Linux*. *Unix* sistemi se koriste na korporacijskim serverima, ali i na radnim stanicama u npr. akademskim ustanovama.

Danas je verovatno najpopularnija **Linux** verzija *Unix*-a. Linux je sistem čiji kernel je besplatan (u javnom vlasništvu). Razvio ga je Linus Torvalds 1991. godine dok je studirao na Univerzitetu u Helsinkiju, bazirajući se na tada popularnoj *MINIX* verziji *Unix*-a.

Korisnici danas dolaze do *Linux* sistema kroz takozvane „distribucije“, koje u stvari predstavljaju pakete koji uključuju kernel, drajvere, verzije grafičkog korisničkog interfejsa i određen broj korisničkih programa. Proces instalacije neke od distribucija može varirati od jednostavnih, do onih koje zahtevaju dobro poznavanje *Linux*-a/*Unix*-a i računara. Trenutno postoji preko 200 aktivnih distribucija za PC računare, a neke od najpoznatijih su *Red Hat*,

Debian, Ubuntu, SuSE, Fedora, Slackware, CentOS, Knoppix, Mandriva i sl. Distribucije obično nisu skupe ili su potpuno besplatne i dostupne na Internetu. *Linux* je veoma fleksibilan operativni sistem, pa tako svaki korisnik može na osnovu kernela sastaviti sopstvenu verziju. Čak se i izgled korisničkog interfejsa može razlikovati budući da postoje različita grafička okruženja (najpoznatija su svakako *Gnome* i *KDE*) koja se mogu koristiti.

Poseban kuriozitet su tzv. *live* distribucije koje ne zahtevaju čak ni instalaciju na računaru, već se pokreću sa nekog uređaja spoljne memorije (CD-a, USB *flash* memorije i sl). Ove distribucije su idealne za početnike koji bi želeli da isprobaju *Linux*, bez rizika po svoj računar i postojeći operativni sistem.



Slika 7.3. Izgled operativnog sistema Linux Ubuntu 11.10 (Ubuntu)

Zbog toga što zahtevaju malo veći nivo znanja i zbog toga što za *Linux* i dalje ne postoje verzije velikog broja popularnih *Windows* programa i igara, ovi sistemi čak i potpuno besplatni i dalje nisu dovoljno zastupljeni na kućnim računarima. Sa druge strane, slika je drugačija kada su u pitanju serverski računari na kojima je *Linux* preferirani sistem zbog svoje stabilnosti, bezbednosti i pouzdanosti.

Iza *Linux*-a danas stoji veliki broj programera iz celog sveta koji pojedinačno ili organizovano razvijaju i održavaju korisničke programe ili delove sistema. Većina tog softvera je besplatna.

Windows operativni sistemi predstavljaju proizvod firme *Microsoft*. Prve verzije *Windows*-a su bile samo grafička nadogradnja starog *MS-DOS* operativnog sistema. Tek kasnije verzije *Windows*-a prerastaju u samostalni operativni sistem. Iako komercijalan, *Windows* je danas najrasprostranjeniji operativni sistem, pre svega zahvaljujući jednostavnom korisničkom interfejsu i dobrom marketingu. *Microsoft* već godinama razvija dve glavne linije *Windows*-a: verziju namenjenu radnim stanicama, odnosno računarima u svakodnevnoj upotrebi (*Windows 3.11, 95, 98, ME, XP, Vista, Windows 7, Windows 8*) i verziju namenjenu serverima (*Windows NT, 2000, 2003, 2008...*). I unutar ove dve linije postoje različite verzije *Windows*-a koje se razlikuju po ceni i komponentama koje uključuju (npr. *Home* i *Professional* verzije *Windows*-a *XP*).

Pravac u kome se danas kreću korisnički operativni sistemi jeste poboljšavanje korisničkog interfejsa, koji postaje sve „ljubazniji“ prema običnom korisniku i sakrivanje tehničkih detalja.

Ide se ka tome da se čak i njihova instalacija olakša (iako se danas računari prodaju sa već instaliranim operativnim sistemom), kao i da računar profunkcioniše takoreći „sam od sebe“, bez potrebe za posebnim podešavanjima. Računarski profesionalci sa negodovanjem gledaju na ovakve promene (napredne opcije su obično dobro sakrivene ili nedostupne), ali upravo zahvaljujući ovom trendu, računari danas imaju ogroman broj korisnika.

Osim ova dva najraširenija, različiti računari mogu imati različite operativne sisteme. Među poznatima su **Mac OS X**, koji se koristi na Apple Macintosh računarima, i **Solaris (SunOS)**, koji je na osnovu **Unix**-a razvila firma **Sun** za svoje radne stanice.



Slika 7.4. Izgled operativnog sistema Windows 7 (Microsoft)

Stariji operativni sistemi koji se još uvek upotrebljavaju (a neki od njih se čak i dalje razvijaju) su IBM-ov **OS/2**, **Amiga OS**, HP-ov **OpenVMS**.

Osim **Windows** i raznih operativnih sistema na bazi **Unix**-a, postoje i sistemi koji ne spadaju ni u jednu od ove dve velike grupe. To su sistemi za velike računare, za specijalne namene ili za posebne uređaje kao što su palmtop računari ili mobilni telefoni. Primeri ovih operativnih sistema su **PalmOS**, **Symbian** i **Android**.

Posebno bi trebalo pomenuti **Google Chrome**, operativni sistem koji je još uvek u razvoju i koji je namenjen Internetu i radu sa **web** aplikacijama.

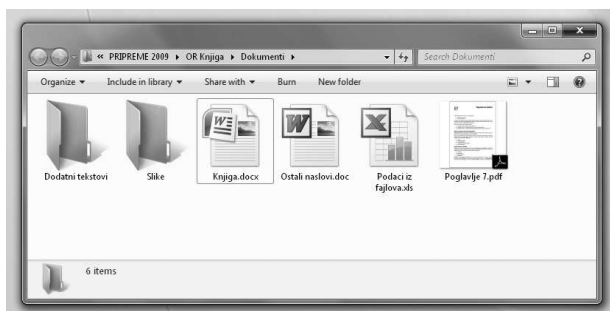
Spoljna memorija predstavlja nezaobilazni deo računarskog sistema. Računari moraju imati mogućnost snimanja i učitavanja sa spoljnog medija. Sa napretkom spoljne memorije, napredovale su i mogućnosti računara vezane za upravljanje snimljenim podacima.

Fajlovi i direktorijumi

Fajl („datoteka“) je niz bajtova snimljen na mediju spoljne memorije. Kao takav, on ima neku veličinu i zauzima mesto na spoljnoj memoriji. **Naziv fajla** se obično sastoji iz **dve komponente** odvojene tačkom. Prvi deo je samo **ime fajla**, a drugi deo je **ekstenzija** ili tip fajla. Pošto i u imenu fajla može učestvovati tačka, za ekstenziju se uzima samo deo posle poslednje tačke u nazivu.

NAZIV FAJLA	
ime fajla	ekstenzija
moj .dokument	. doc

U starom operativnom sistemu MS-DOS, postojalo je ograničenje da ime fajla može imati maksimalno 8 znakova, a ekstenzija maksimalno 3. Iako, sa uvođenjem modernijih fajl sistema, ovo ograničenje više ne važi, ako koristimo neki stariji program pod novim Windows-om, u njemu će se prikazivati skraćeni nazivi fajlova i foldera.



Slika 8.1. Prikaz fajlova i podfoldera u nekom folderu pod operativnim sistemom **Windows 7**. Ikone fajlova su prikazane na osnovu ekstenzije, a u nekim slučajevima i sadržaja fajla.

Ekstenzija ima ulogu da odredi tip fajla. U najvećem broju slučajeva i dalje se sastoji iz 3 znaka, ponekad 4, a vrlo retko više. Na osnovu toga, Windows zna kojim programom da otvori određeni fajl. Nekada je bilo obavezno da se prvo pokrene program, a onda iz njega učita određena datoteka, dok u današnje vreme, korisnik može da klikne na neki fajl, a operativni sistem će pokrenuti odgovarajući program i onda u njega učitati fajl na koji je korisnik kliknuo.

Takođe, na osnovu ekstenzije, Windows ume da prikaže odgovarajuću grafičku reprezentaciju fajla – **ikonu**.

Ako promenimo ekstenziju fajla nećemo zaista promeniti tip fajla, tako se samo stvara konfuzija. Sam fajl se ne menja, a Windows će promeniti ikonu za prikaz i pokušati da otvori fajl u pogrešnom programu.

Neki znakovi imaju posebno značenje, pogotovo kada se zadaju naredbe iz komandne linije (*DOS Prompt*) i ne smeju se koristiti u nazivu fajlova i direktorijuma:

< > : " / \ | ? *

Naziv fajla ne bi trebalo da bude ni jedna od reči koje su rezervisane u DOS-u/Windows-u: CON, PRN, AUX, NUL, COM1, COM2, ..., COM9, LPT1, LPT2, ..., LPT9.

Isto tako, naziv fajla ne bi smeo da se završi razmakom ili tačkom.[24]

Fajlovi se na mediju spoljne memorije organizuju unutar hijerarhijske strukture direktorijuma i poddirektorijuma. **Direktorijum** je logički skup fajlova i poddirektorijuma označen nekim imenom. To znači da svaki direktorijum može sadržati fajlove i druge direktorijume. „Logički“ skup označava da fajlovi na samom disku nisu, niti moraju biti, bilo kako povezani. Oni su u fajl sistemu samo deklarirani kao fajlovi koji pripadaju određenom direktorijumu. Samim tim, direktorijum ne postoji zaista kao objekat na disku, već se samo beleži kao objekat u alokacionoj tabeli (*VFAT – Virtual File Allocation Table*, odnosno *MFT – Master File Table* u novijim verzijama Windows-a).[25]

Kopiranje, premeštanje i brisanje

Zbog svega pomenutog primećujemo interesantnu pojavu – **kopiranje fajlova** iz jednog direktorijuma u drugi **zahteva određeno vreme**, zavisno od veličine fajlova – pošto se kreira nova kopija fajla negde na disku.

Sa druge strane, **premeštanje fajlova** iz jednog direktorijuma u drugi obavlja se **gotovo trenutno** – u tom slučaju, operativni sistem samo „proglašava“ da ti fajlovi više ne pripadaju jednom već drugom direktorijumu. Ovo, naravno, ne važi ako se premeštanje obavlja između dva medija spoljne memorije (sa diska na disk).

To što i **brisanje fajlova** traje mnogo kraće od kopiranja govori nam da se fajlovi **ne brišu zaista sa diska**, već se samo njihovi klasteri proglašavaju slobodnim. Zahvaljujući tome funkcionišu programi koji vraćaju obrisane podatke – ipak, ako se neki drugi fajlovi snime preko tih klastera koje su zauzimali stari fajlovi – tu više nema pomoći.

Inače, različiti sistemi i verzije imaju različite nazive za direktorijume – folderi, fascikle, fioke... Već je pomenuto da se početni direktorijum, koji sadrži sve ostale fajlove i poddirektorijume, naziva osnovni ili korenski (*root*) direktorijum.

Arhive i kompresija

Od kada postoji rad sa fajlovima, postoji i potreba da se fajlovi koji se ne koriste a suviše su vredni da se obrišu ili fajlovi koji se prenose „spakuju“ u arhivu. **Arhiva** je jedan fajl koji u sebi sadrži više fajlova i iz koga ti fajlovi kasnije mogu biti vraćeni.

Inicijalno, arhive su služile samo za smeštanje fajlova (kao npr. TAR fajlovi na Linux-u), međutim, danas svi popularni formati arhiva uključuju i mogućnost kompresije.

Kompresija predstavlja postupak smanjivanja veličine fajla. Postoje dva tipa kompresije [21]:

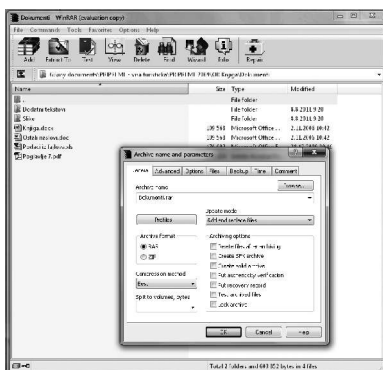
- kompresija bez gubitka podataka (*lossless*) i
- kompresija sa gubitkom podataka (*lossy*).

Kompresija bez gubitka podataka se obično zasniva na nekoj varijanti LZ77 (*Lempel-Ziv*), odnosno LZW (*Lempel-Ziv-Welch*) algoritma. Ovi algoritmi funkcionišu tako što u fajlu pronalaze jednake nizove bajtova (šablone) i izdvajaju ih u tzv. „rečnik“. Na taj način, izbacivanjem dupliciranih šablona smanjuje se veličina fajla.

Kada se vrši **dekompresija** fajl se vraća u originalno stanje. Zbog toga što su vraćeni podaci „do u bajt“ jednaki originalnom fajlu, ovakva vrsta kompresije naziva se "lossless", odnosno kompresija bez gubitaka.

Ovi algoritmi nisu svemogućí. Zavisno od tipa fajla i oblika podataka postiže se veći ili manji stepen kompresije. Na primer, detaljna fotografija će imati mnogo niži nivo kompresije u odnosu na crtež koji ima puno jednojnih površina. Takođe, jednom kompresovani podaci ne mogu se dalje efikasno kompresovati.

Ovakva kompresija se koristi za programe i dokumente. Neki od najpoznatijih formata arhiva koriste ovakav tip kompresije (ZIP, RAR i slični), kao i različiti tipovi dokumenata (Office dokumenti, PDF fajlovi, BMP, GIF i PNG slike).



Slika 8.2. Jedan od najkorišćenijih programa za arhiviranje – WinRAR 4.0 (Alexander Roshal)

Sa druge strane, **kompresija sa gubitkom podataka** koristi se kada podatke nije neophodno vratiti u potpuno isto stanje kao original. Takva kompresija se može upotrebiti na slikama i multimedijском (audio i video) materijalu.

Ova kompresija pre svega zavisi od tipa podataka – slika, zvuk i video se kompresuju na različite načine. Takođe, *lossy* kompresija je, u nekom smislu, prilagođena čoveku. Na primer, kod kompresije zvuka, iz snimka se izbacuju sve promene u jačini i visini zvuka koje ljudsko uho ne primećuje, odnosno koje ne dostižu ovu razliku (*JND – Just Noticeable Difference*).[1]

Podaci se dekompresuju korišćenjem različitih matematičkih metoda i dobijeni rezultat nikada nije jednak originalnim podacima. Međutim, kod ovih podataka to i nije toliko značajno, a stepen kompresije je daleko veći u odnosu na *lossless* kompresiju. Ovi algoritmi se koriste u nekim formatima slika (JPG na primer), zvuku (MP3 muzika) ili za video (MPEG, odnosno čuvena DIVX kompresija).

Da bi se postigla maksimalna kompresija, uvek se vrši i dodatno pakovanje *lossless* metodom – zbog toga je praktično besmisleno npr. „zipovati“ JPG slike ili MP3 muziku.

Tipovi fajlova

Danas postoji na hiljade tipova fajlova, međutim, prosečnom korisniku je značajan samo jedan mali broj. Ovde će biti pomenuti najčešći tipovi sa kojima se susreću korisnici Windows-a.

Izvršni fajlovi

Izvršni fajlovi su u stvari programi koje korisnik može pokrenuti u sistemu. Najvažniji i najrasprostranjeniji tip izvršnog fajla je **EXE**.

COM su takođe programi, koji su predstavljali komande starog MS-DOS-a. U DOS-u su se izvršavale i **BAT** datoteke, koje su, u stvari, tekstualni fajlovi (skripte, u ovom slučaju nazvane *batch* fajlovima) i koje su sadržale jednostavan program sastavljen od DOS naredbi.

JAR je u stvari arhiva koja u sebi sadrži Java program i koja može biti napravljena tako da se izvršava (tako što pokrene Java virtuelnu mašinu koja dalje interpretira program).

DLL (*Dynamic Link Library*) nije zaista nezavisan program već biblioteka funkcija koje mogu koristiti drugi programi ili sam Windows.

Dokumenti

Najprostiji tip dokumenta su, naravno, obični tekstualni fajlovi – **TXT**. Ovo je čist tekst koji ne sadrži nikakvo formatiranje (fontove, stilove ispisa, boju...).

RTF (*Rich Text*) je format koji se danas ređe susreće. Predstavlja tekstualni dokument sa nekim osnovnim formatiranjem. Sa ovim formatom radi program *WordPad* koji dolazi uz Windows.

DOC i **DOCX** su formati *Word* dokumenata, pri čemu se **DOCX** koristi počev od Office-a 2007 i starije verzije *Word*-a ga ne mogu otvarati. **ODT** je format dokumenta *OpenOffice Writer*-a.

XLS, **XLSX** i **ODS** – radne tabele. **XLS** i **XLSX** su fajlovi koje koristi Microsoft *Excel*, a **ODS** *OpenOffice Calc*.

PDF (*Portable Document Format*) – dokument pripremljen za štampu ili čitanje. Ovi dokumenti nisu namenjeni prepravkama. Da bi se kreirali, koristi se aplikacija *Adobe Acrobat* ili neka od alternativa. Neki programi (*OpenOffice*) imaju mogućnost snimanja (*export*) u ovom formatu. Za čitanje PDF dokumenata koristi se besplatna aplikacija *Adobe Acrobat Reader*.

HTML (*HyperText Markup Language*) – stranica na Internet sajtu. Ovo je u stvari tekstualni fajl koji u sebi sadrži posebne komande za formatiranje.

Prezentacije

PPT, **PPTX** i **ODP** – multimedijalne prezentacije kreirane u programima Microsoft *PowerPoint* i *OpenOffice Impress*.

SWF – multimedijalne prezentacije i filmovi koji se mogu videti na Internetu. Kreiraju se u programima *Adobe Director* i *Adobe Flash*.

Grafika

JPG je verovatno najpopularniji format za čuvanje fotografija i ostalih slika. Koristi *lossy* kompresiju, a prilikom snimanja je moguće podesiti nivo kompresije, čime se mogu dobiti veoma dobri rezultati i neprimetan gubitak kvaliteta slike. Ovaj format slike najčešće se koristi i na Internetu.

GIF i PNG – formati koji su takođe zastupljeni na Internetu. GIF je format koji može zabeležiti sliku sa paletom od samo 256 boja, ali je bio popularan zbog mogućnosti da jedna od tih boja bude transparentna (providna). Danas se još koristi zbog svoje druge mogućnosti, a to je da sadrži animaciju. PNG format se sve više koristi na Internetu zbog mogućnosti da beleži 256 nivoa transparentcije, što je daleko bolje u odnosu na GIF.

BMP i TIFF su relativno popularni, mada prosečan korisnik ređe nailazi na njih. BMP je Microsoft-ov format grafike i van Windows operativnog sistema se retko koristi.

PSD – format slika koji koristi Adobe Photoshop. Jedna od prednosti je što slike mogu biti sačuvane sa tzv. *layer*-ima, koji predstavljaju nezavisne delove slike.

AI, CDR, SVG – vektorski formati slika. AI koristi *Adobe Illustrator*, CDR *Corel Draw*, dok je SVG (*Scalable Vector Graphics*) format koji je namenjen prikazu na Internetu. Jedan od programa koji radi sa ovim tipom grafike je besplatni program *Inkscape*. [23]

TTF i OTF bi mogli da se podvedu pod grafički format. U pitanju su definicije fontova.

Zvuk i video

WAV, WMA, AVI i WMV – Microsoft formati. WAV predstavlja nekompresovan zvuk, a AVI je verovatno najkorišćeniji format za video. Za razliku od WAV zvuka, WMA format ima *lossy* kompresiju koju je Microsoft pokušao da nametne kao standard umesto MP3. WMA i WMV imaju mogućnost korišćenja DRM (*Digital Rights Management*) tehnologije kako bi se sprečila piraterija.

MP3 i M3U – MP3 je danas verovatno najkorišćeniji format zvučnog zapisa. Popularan je zbog svoje *lossy* kompresije koja daje zvuk dobrog kvaliteta, uz veličinu fajla koja je desetostruko manja od nekompresovanog audio zapisa. Verovatno najpopularniji program za slušanje MP3 muzike je *Winamp*, a M3U je njegov format zapisa *playliste*.

OGG, OGV i OGA su medijski fajlovi koji koriste jaku Theora i Vorbis *lossy* kompresiju i namenjeni su Internetu. OGG sam po sebi može biti video ili zvuk, ali nekim web čitačima je jednostavnije da se video nazove OGV, a čist audio OGA. **FLV** je takođe tip videa koji je namenjen prikazu unutar Flash *player-a*, zbog čega je takođe popularan na Internetu.

RA, RAM, RM, RMV – ovi formati predstavljaju medijske fajlove za *Real Player*.

3GP, MPG i MKV su video formati. MPG ili MPEG označava video najčešće kompresovan algoritmom MPEG 4. MKV (*Matroska*) postaje sve popularniji kontejnerski format koji, osim videa i različitih audio strimova, može sadržati i titl. 3GP je uobičajen na mobilnim telefonima.

Arhive

ZIP, RAR, 7Z, ARJ – različiti formati arhiva. Trenutno su najpopularnije RAR i ZIP arhive. ARJ se danas slabo koristi – nekada je bio popularan zbog jednostavne mogućnosti deljenja arhive na više fajlova, ali danas RAR pruža tu istu mogućnost uz kvalitetniju kompresiju. 7Z (7-Zip) je arhiva koja pruža sve što i ostale arhive (dobar stepen kompresije, mogućnost zaštite podataka šifrom), s tim što je u pitanju projekat otvorenog koda i samim tim besplatan.

TAR i GZ predstavljaju arhive koje se koriste na Linuxu, ali ponekad i Windows korisnici mogu naići na fajlove spakovane na ovaj način. TAR označava običnu arhivu bez kompresije, dok GZ predstavlja kompresovan fajl. Takav fajl može biti označen i kao TAR.GZ ili TGZ. [22]

Već smo zaključili da su za rad računara potrebni programi, koji nastaju zahvaljujući posebnoj vrsti aplikacija – programskom softveru. U programski softver spadaju sve aplikacije koje služe za pisanje drugih programa. Ponekad je za to dovoljan najjednostavniji tekst editor, a ponekad su u pitanju veoma složeni softverski paketi.

Algoritam

Najjednostavnije rečeno, algoritam predstavlja neki precizno opisan postupak. Taj postupak je obično svrsishodan, što znači da nas vodi do nekog cilja (rešenja problema) do koga možemo doći praćenjem niza koraka. Tako možemo reći da je algoritam **precizno definisan postupak za rešavanje nekog problema u konačnom broju koraka**.

U algoritmu koraci moraju biti jasno odvojeni, a sam algoritam mora biti nedvosmislen i mora se završiti posle nekog određenog broja koraka. Za iste početne vrednosti, algoritam uvek daje iste rezultate i primenljiv je za veći broj različitih početnih vrednosti.

Algoritmi su značajni za programiranje iz prostog razloga što sami programi predstavljaju konkretnu realizaciju algoritama na računaru.

Algoritmi mogu biti opisani na razne načine, od kojih su neki više (pseudokod) ili manje (prirodan jezik) formalizovani. Jedan od najpoznatijih i najpreglednijih načina za predstavljanje algoritama je putem algoritamske šeme (blok-dijagrama).

Postoji nekoliko podela algoritama, a najpoznatija je prema načinu izvršavanja koraka. Po njoj, algoritmi mogu biti:

- **linijski** – algoritmi kod kojih se svaki korak izvršava tačno jedanput;
- **razgranati** – algoritmi kod kojih se neki koraci mogu izvršiti jednom ili nijednom;
- **ciklični** – algoritmi kod kojih se neki koraci mogu izvršiti više puta.

Naravno, svaki malo složeniji algoritam u sebe uključuje elemente sva tri tipa.

Treba da znamo da se kvalitet svakog programa, a samim tim i algoritma, ocenjuje po dva osnova, a to su **brzina** i **korišćenje memorije**. Iz ranijih tekstova o hardveru, sećamo se da brzina i memorija predstavljaju dva ključna elementa u razvoju računara – procesori postaju sve moćniji i brži, a kapaciteti unutrašnje i spoljne memorije se sve više povećavaju.

To znači da, sa jedne strane, proizvođači hardvera pokušavaju sve više da ubrzaju računare i da im ugrade što je moguće više memorije, dok se, sa druge strane, programeri trude da pišu bolje programe koji će raditi brže i zauzimati manje memorije. Na taj način računari mogu da rade sa većom količinom podataka i time budu upotrebljiviji.

Algoritmi protiv Holivuda

Kao dobar primer možemo navesti rad sa video zapisima na računaru. Prosečan film bi zauzeo desetine gigabajta memorije kada se ne bi koristila kompresija (sažimanje).

Jedna od najpoznatijih klasa algoritama za kompresiju video zapisa je **MPEG**, a svima je dobro poznata jedna od njegovih implementacija – **DivX**, koja se danas često koristi za kompresiju i reprodukciju filmova. Zahvaljujući ovom algoritmu, odnosno „kodeku“ (codec), celovečernji film se može smestiti na jedan CD.

Ipak, na nekom starijem računaru, baziranom na procesoru npr. Intel 486 ili Pentium, DivX filmovi ne bi mogli čak ni da se reprodukuju. Srećom, moderni računari su više nego dovoljni za prikaz filmova, a kako kompjuteri postaju sve jači, moguće je i povećati kvalitet video snimka.

Bolji algoritam je onaj koji je efikasniji po pitanju korišćenja memorije i koji u manjem broju izvršenih koraka dolazi do rezultata.

Programi i algoritmi se obično posmatraju kao niz instrukcija koje za zadati ulaz (početne podatke) kreiraju određeni izlaz (rezultat). **Kompleksnost algoritma** se procenjuje na osnovu toga kako se povećava broj koraka koji se moraju izvršiti sa povećanjem ulaza. Što je više koraka potrebno da se dođe do rezultata, algoritam će raditi sporije. Tako postoji nekoliko klasa algoritama, od kojih će ovde biti pomenute samo najznačajnije.[26]

- **O(1)** – bez obzira na ulaz (količinu podataka koji se zadaju na početku), ne menja se broj koraka. Primer predstavlja pretraga preko tzv. heš (*hash*) tabele (pretraga se obavlja veoma brzo na osnovu posebne funkcije koja izračunava poziciju elementa koji se traži).
- **O(n)** – linearna kompleksnost, koliko se poveća ulaz, toliko se poveća i broj koraka. Primer bi bilo sabiranje dva broja sa po N cifara.
- **O(log n)** – dobri algoritmi kod kojih sa povećanjem ulaza dolazi do relativno malog povećanja broja koraka. Primer je binarna pretraga (najbolje je možemo zamisliti kao deljenje intervala prilikom pogađanja brojeva).
- **O(n log n)** – takođe veoma dobri algoritmi gde se broj koraka ne povećava puno. Jedan od primera je „zavadi pa vladaj“ (*divide and conquer*) metod, primenjen u *quicksort* algoritmu za sortiranje podataka.
- **O(n²)** – algoritmi sa dva koncentrična ciklusa u kojima se obično pojavljuje upoređivanje „svaki sa svakim“. Neki od najpoznatijih algoritama za sortiranje su upravo ovog tipa (*bubble sort, shell sort, insertion sort, selection sort...*)
- **O(2ⁿ)** – ovo su obično najgori algoritmi kod kojih se broj koraka drastično povećava sa malim povećanjem ulaznih podataka. Primer su algoritmi tzv. „grube sile“ (*brute force*), koji obično do rešenja dolaze isprobavanjem svih mogućih kombinacija.

Razvoj softvera

Razvoj bilo kakve složenije aplikacije danas nije posao koji se prepušta slučaju. U pitanju je proces koji sve više podleže standardizaciji i kome se i te kako ozbiljno i planski pristupa.

Sam tok razvoja softvera može se podeliti na nekoliko etapa, a jedna podela bila bi na:

- planiranje,

- analizu zahteva,
- dizajn,
- programiranje,
- testiranje i
- korišćenje u realnim uslovima.

Prve tri etape predstavljaju pripremne faze, kojima se određuju plan i metodologija rada i oblikuje koncept aplikacije. Sledi programiranje koje rezultuje radnom verzijom programa. Program se tada testira i ispravljaju mu se greške, da bi potom počelo njegovo korišćenje u probnom periodu, tokom koga se ocenjuje koliko program odgovara početnim zahtevima. Čak i kada program potpuno odgovara zahtevima, angažman programera se nastavlja na održavanju i unapređenju softvera. Suštinski, ove etape predstavljaju 4 faze klasičnog razvoja – definisanje šta program radi, definisanje **kako** program to treba da radi, **programiranje** i **korišćenje**. [29]

Postoji veliki broj metodologija rada koje u manjoj ili većoj meri mogu biti efikasne prilikom razvoja softvera, a dve osnovne su metod vodopada i agilno programiranje.

Metod vodopada predstavlja metodologiju prema kojoj etape u razvoju softvera doslovno prate jedna drugu, počev od analize zahteva, preko dizajna, programiranja, testiranja, pa do instalacije i održavanja. Svaka faza se kompletira i predstavlja osnov za sledeću. Pri tome se svaka etapa realizuje veoma pažljivo kako se ne bi desile greške koje bi bile skupo plaćene u kasnijim fazama. Metod vodopada predstavlja klasičan, disciplinovani način razvoja softvera, koji se u velikoj meri oslanja na detaljnu dokumentaciju u svakoj etapi. [27]

Agilno programiranje predstavlja metodologiju koja nastaje kao odgovor na rigidni metod vodopada. Zagovornici agilnog programiranja smatraju da je nemoguće apsolutno kompletirati ijednu fazu a da ne dođe do grešaka. Kod ove metodologije kreira se veliki broj međuverzija softvera koje prolaze kroz sve navedene etape razvoja, što rezultira malim brojem grešaka i brzim razvojem aplikacije koja odgovara potrebama naručioca. Razvojni tim izbegava uplitanje u hijerarhijske strukture nadređenih i podređenih, a članovi redovno međusobno komuniciraju. U svakom timu učestvuje bar jedan predstavnik klijenta koji je naručio softver. Ova metodologija se zasniva na komunikaciji, a ne na papirologiji i birokratiji. Na ovaj način postiže se visoka fleksibilnost i efikasnost u radu. [28]

U neke od vidova agilnog programiranja spadaju i **ekstremno programiranje** (*XP – eXtreme Programming*), **rapidni razvoj aplikacija** (*RAD – Rapid Application Development*), **minimalističko programiranje** (*LD – Lean software Development*), itd.

Programski jezici

Programski jezik je veštački jezik koji se koristi za kontrolisanje rada računara. Prema nekim autorima, programski jezik je samo onaj jezik kojim se može predstaviti svaki mogući algoritam.

Kao i prirodni jezici, svaki programski jezik ima određena pravila:

- **sintakse** (gramatike) – pravila koja određuju način pisanja programa i
- **semantike** (smisla) – pravila kojima se definiše da ono što je napisano ima neko značenje.

Pogledajmo primer naredbe kojom se proizvod promenljivih X i Y smešta u promenljivu A. Pravila sintakse programskog jezika *Pascal* nam zabranjuju da tu naredbu napišemo npr. kao:

a = *(x, y) .

Prethodni red ima veliki broj grešaka (naredba dodele nije isto što i operator jednakosti, operator množenja se mora naći između dva operanda, posle naredbe se ne stavlja tačka, itd). Sintaksno ispravan način zapisa ove naredbe bio bi:

```
a := x * y;
```

Međutim, šta ako promenljive X i Y nisu numeričke već tekstualne? U tom slučaju, iako smo napisali sintaksno ispravnu naredbu, ona nema smisla, jer ne postoji stvar kao što je proizvod dva teksta.

Tokom godina, razvijeno je nekoliko stotina programskih jezika, namenjenih rešavanju specifičnih grupa problema. Neki od njih su u širokoj upotrebi, dok se neki koriste samo u posebne svrhe.

Programske jezike možemo podeliti na više načina. Verovatno najosnovnija bila bi podela **prema prilagođenosti jezika čoveku odnosno računaru**, pa tako programske jezike delimo na:

- **jezike niskog nivoa**, koji su bliski radu računara i
- **jezike visokog nivoa**, koji su bliski ljudskom načinu razmišljanja.

Programi pisani u jezicima niskog nivoa (mašinski jezik, assembler, jezik C) su veoma brzi, ali teški za pisanje i pronalaženje grešaka. Sa druge strane, u jezicima višeg nivoa, programiranje i podela posla su mnogo lakši, ali su i programi pisani u njima sporiji. Što je jezik na višem nivou to je više prilagođen rešavanju specifičnih zadataka, dok je rešavanje opštih problema teže i proizvodi mnogo sporiji program. Primeri ovakvih jezika su *Flash ActionScript*, *VBA (Visual Basic for Applications)* i slični.

Napredak programskih jezika i orijentacija programera ka višim jezicima i moćnijim radnim okruženjima postaje neminovna, pošto se fokus u poslovnim aplikacijama prebacuje sa „brzina po svaku cenu“ na „upotrebljivost pre svega“. Tako programi postaju sve kompleksniji pošto bivaju prilagođeni sve većem broju korisnika koji ne moraju imati informatičko znanje. Takođe, potrebno je osigurati što brži razvoj kvalitetnih aplikacija, te tako programeri koriste svaku mogućnost da što pre kreiraju aplikaciju koja zadovoljava potrebe naručioca. Srećom, razvoj računara i povećanje njihove brzine i memorije omogućio je da i ovakvi programi zadovoljavajuće funkcionišu.

Danas se veoma retko pristupa programiranju u assembleru (koji je veoma blizak mašinskom jeziku računara). Moderni kompajleri su veoma napredovali u optimizaciji prevedenih programa, a procesori su postali veoma kompleksni. U najvećem broju slučajeva kompajler će napraviti bolji mašinski program u odnosu na čoveka – programera.

Prema načinu izvršavanja, programske jezike možemo podeliti na:

- **jezike koji se prevode (kompajliraju)** – programi pisani u ovim jezicima se prvo prevode na mašinski jezik, a onda se mogu izvršavati;
- **jezike koji se interpretiraju** – programi se odmah izvršavaju tako što se svaka pojedina naredba prevodi i izvršava.

Programi prevedeni na mašinski jezik su veoma brzi, ali su prilagođeni samo jednom tipu procesora, odnosno operativnog sistema. Sa druge strane, interpretirani programi se izvršavaju daleko sporije, ali su mnogo fleksibilniji po pitanju platforme na kojoj se izvršavaju – mogu funkcionisati gde god postoji njihov interpreter.

Većina komercijalnih programa i igara za računare prevodi se na mašinski jezik, pošto se od njih zahteva maksimalna brzina (npr. jezici *C*, *Pascal*, *Delphi*, *C++*). Međutim, i interpretirani jezici imaju svoju upotrebnu vrednost, budući da postoje situacije u kojima brzina nije od presudnog značaja, a zahteva se mogućnost da se neki manji posao automatizuje ili da se ostvari neka interaktivnost. Tipičan primer bili bi skript jezici koji se koriste na Internet prezentacijama (*JavaScript*) i koji se izvršavaju unutar *web* stranice bez obzira na to koji sistem koristi posetilac sajta.

Danas postoji posebna, naprednija klasa interpretiranih jezika, koji omogućuju veću brzinu programa, uz istovremenu fleksibilnost izvršavanja na različitim platformama. U pitanju su programi koji se izvršavaju preko tzv. **virtuelne mašine**. Virtuelna mašina predstavlja simulaciju računara u računaru. Tako se programi prevode, ali ne na mašinski jezik, već u „bajtkod“ (*bytecode*), koji u stvari predstavlja „mašinski“ jezik virtuelnog računara. Ovakav program može se pokrenuti na svakom računaru na kome postoji virtuelna mašina.

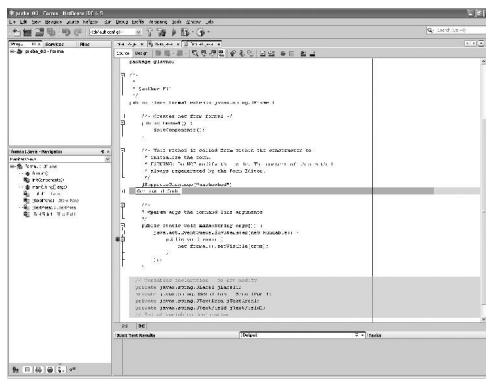
Zavrtnanje česme

Osim mogućnosti izvršavanja na različitim platformama, virtuelna mašina ima i druge prednosti. Jedna od najznačajnijih za programere jeste efikasno rešenje problema tzv. „curenja memorije“ (*memory leak*).

Programi uobičajeno rade sa nekim podacima – za podatke je neophodno rezervirati memoriju pod operativnim sistemom, koji programu obezbeđuje adresu (tzv. „pokazivač“) memorije gde se podaci nalaze. Problem curenja memorije nastaje ukoliko programer zaboravi da oslobodi zauzetu memoriju, a pokazivač usmeri na neki drugi objekat u memoriji. Ovakvi zaboravljeni podaci će se neprekidno gomilati u memoriji računara dok je toliko ne popune da uspire ili onemoguću rad celog sistema.

Virtuelna mašina obezbeđuje takozvani „sakupljač smeća“ (*garbage collector*), koji automatski uklanja ovakve objekte iz memorije.

Ovakvi programi se izvršavaju brže od obično interpretiranih programa, ali ipak i dalje sporije od programa kompajliranih u originalni mašinski jezik. Jedna od optimizacija kojima se ubrzava rad programa u virtuelnoj mašini je „prevođenje u poslednjem času“ (*JIT – Just In Time compilation*), kada se bajtkod neposredno pre izvršavanja prevede u mašinski jezik.[30]



Slika 9.1. NetBeans je jedno od najkorišćenijih okruženja za razvoj Java aplikacija

Programski jezik Java je tipičan primer ove strategije u razvoju programskih jezika. Tako se programi pisani u Javi mogu izvršavati na različitim računarima, ali i na mobilnim telefonima ili

drugim uređajima. Po ugledu na Javu i njenu virtuelnu mašinu, Microsoft je razvio sopstvene .NET (*dot-net*) programske jezike – pandan Javi za .NET platformu je jezik C# ("see sharp").

Većina programskih jezika omogućava rad sa podacima. U trenutku pisanja programa, programer određuje kakvi će biti podaci sa kojima program radi. Tipovi podataka i njihove strukture mogu biti brojni, međutim, osnovnih tipova ima relativno malo, pa tako programi obično operišu sa celim brojevima, racionalnim (decimalnim) brojevima, tekstom i logičkim podacima (tačno – netačno).

Prema načinu na koji rade sa tipovima podataka, programski jezici se mogu podeliti na:

- **netipizirane** i
- **tipizirane**, koji se dele na
 - **slabo tipizirane** i
 - **strogo tipizirane**.

Netipizirani su jezici kod kojih ne postoji nikakva provera tipova podataka. Podaci se posmatraju onako kako programeru odgovara u tom trenutku. Primer ovakvog jezika jeste assembler, kod koga je moguće npr. sabrati znak i broj (uglavnom zato što na toliko niskom nivou svi podaci i jesu brojevi).

Većina modernih programskih jezika su **tipizirani**, te kod njih nije dozvoljeno mešanje različitih tipova podataka u izrazima. Kod strogo tipiziranih jezika uvek se mora vršiti tzv. „konverzija“ tipova. Sa druge strane, ukoliko je jezik slabo tipiziran, utoliko je moguće jedan tip podatka proslediti kao drugi. Iako se čini da je ovakav način lakši za programera, postoji opasnost od sasvim neočekivanih i teško uočljivih grešaka.

Na primer, u jeziku JavaScript je sasvim legalno napisati "**10**" + 7, pri čemu će se kao rezultat dobiti tekst "107", dok će npr. izraz "**10**" * 7, kao rezultat dati broj 70. U prvom slučaju oba podatka su tretirana kao tekst, dok su u drugom oba tretirana kao brojevi. Ako programer ne bi vodio računa o tipovima, onda bi program funkcionisao bez prijavljenih grešaka, ali bi rezultati bili pogrešni.

Paradigma programiranja predstavlja način na koji se vrši programiranje u nekom programskom jeziku, odnosno „filozofiju“ samog jezika. Jezike tako možemo podeliti na dve osnovne grupe:

- **imperativni jezici**, koji nastaju kao realizacija nekog algoritma na računaru i kod kojih se zadaje niz naredbi u obliku iskaza koje računar izvršava (BASIC, Pascal, Fortran, C...);
- **deklarativni jezici**, nisu zaista programski jezici u užem značenju, pošto, za razliku od imperativnih jezika, ne definišu „kako“ nešto uraditi, već „šta“ predstavljaju neki podaci (dobar primer deklarativnog jezika je HTML, jezik koji služi za opis *web* stranica na Internetu).

Ipak, uobičajena podela prema paradigama programiranja bila bi na:

- **strukturno, tj. klasično „imperativno“ programiranje** jeste programiranje u najužem smislu reči – zadavanje komandi računaru, korišćenje struktura kontrole toka (grananje, ciklusi);
- **funkcionalno programiranje** predstavlja malo drugačiji pristup programiranju, kod koga se izbegava korišćenje promenljivih, a ceo program se definiše kao kompleksna funkcija (primer funkcionalnog jezika je Lisp, koji se koristi pri automatizaciji rada u AutoCAD

paketu; rad sa funkcijama u Excel-u, kao i rad sa bazama podataka u SQL-u se takođe mogu podvesti pod funkcionalno programiranje);

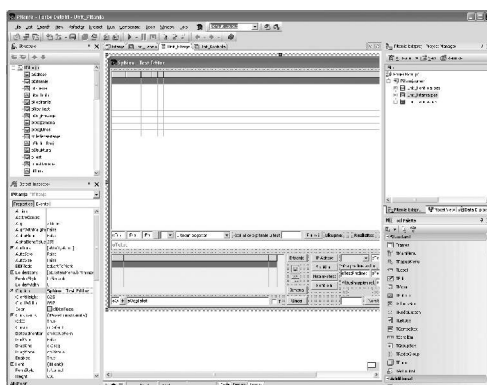
- **logičko programiranje** se koristi u razvoju veštačke inteligencije i najbliže je deklarativnim jezicima, budući da se ovde računaru ne određuje način na koji će rešiti problem, već se sam problem detaljno opisuje (primer ovakvog jezika je Prolog);
- **objektno-orijentisano programiranje** razvijeno je sa idejom modeliranja pojava i situacija iz realnog sveta i danas predstavlja verovatno najpopularniji pristup programiranju (skoro svi najznačajniji moderni programski jezici podržavaju OOP: C++, Java, Delphi...).

Objektno-orijentisano programiranje se najjednostavnije može opisati kao način da se podaci i delovi programa povežu u jednu celinu – **objekat**. Tako objekat ima određena **svojstva** (*properties*) – polja sa podacima i **metode** (*methods*) – procedure i funkcije koje definišu ponašanje objekta.

OOP ima nekoliko tipičnih koncepata, koji ga razlikuju od ostalih paradigmi [31]:

- **kompozicija** – kreiranje klasa objekata koji sadrže podobjekte;
- **nasledivanje** – kreiranje novih klasa objekata na osnovu već postojećih;
- **enkapsulacija** – „privatizacija“ programa u objektu, što omogućuje da se jedan objekat bez problema zameni drugim, koji ima ista svojstva i metode;
- **polimorfizam** – mogućnost da izvedeni objekat bude prosleđen umesto objekta „roditeljske“ klase.

Zahvaljujući navedenim osobinama, OOP omogućuje jednostavnije rešavanje problema, brzo programiranje, laku podelu posla, jednostavno korišćenje tuđih delova programa, kao i manji broj grešaka u programu. Takođe, objektno programiranje se odlično uklapa u programiranje za grafički korisnički interfejs (npr. programiranje Windows aplikacija).



Slika 9.2. Razvojno okruženje Delphi nasleđuje programski jezik Pascal i obogaćuje ga objektno-orijentisanim programiranjem i radom u Windows okruženju

Osim objektno-orijentisanih jezika, danas postoje i jezici zasnovani na objektima. Ovi jezici ne omogućavaju deklarisanje sopstvenih klasa objekata (ili ih dozvoljavaju u nekoj ograničenoj meri), ne poseduju navedene koncepte OOP-a, ali imaju mogućnost korišćenja već gotovih objekata. Primer ovakvih jezika su skript jezici poput JavaScript-a i VBA, koji se u velikoj meri oslanjaju na objektni model (*DOM – Document Object Model*), odnosno objekte koje imaju na raspolaganju preko *web* čitača, odnosno *Microsoft Office* aplikacija.

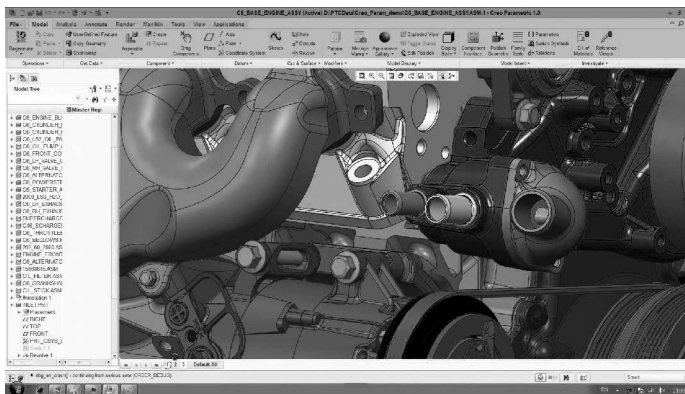
10

Aplikativni softver

Sve do sada napisano bilo je u vezi sa primenom računara za sam računar. Operativni sistemi predstavljaju upravljanje samim računarom, programski jezici – pisanje programa za računar. Aplikativni softver predstavlja sve one programe koji računaru daju mogućnost primene u najrazličitijim oblastima – tehnici, poslovanju, obrazovanju, nauci, komunikaciji ili zabavi, što je u stvari prava upotrebna vrednost računara za većinu korisnika.

Tehnika

Kompjuterski inženjering obuhvata nekoliko oblasti, od kojih je verovatno najznačajnija **kompjutersko projektovanje (CAD – Computer Aided Design)**. CAD se upotrebljava za industrijski dizajn, mašinstvo, arhitekturu, građevinarstvo. Ovi programi omogućavaju kreiranje 2D tehničkih crteža ili celih trodimenzionalnih modela objekata. Ovi modeli kasnije mogu biti osenčeni, tako da se dobije realističan prikaz projektovanog objekta. Osim projektovanja, CAD softver omogućava i automatsko vođenje dokumentacije o projektu. Sličan softver se koristi i za projektovanje elektronskih komponenata. Aplikacije za kompjuterizovanu proizvodnju (**CAM – Computer Aided Manufacture**) služe za kontrolu mašina i obično su u uskoj vezi sa programima za projektovanje, pošto mogu učitavati CAD modele objekata koje treba proizvesti. Neke od najpoznatijih aplikacija su *Creo* (nekadašnji *Pro Engineer*), *Catia* i stariji *AutoCAD*.



Slika 10.1. Aplikacija za računarsko projektovanje *Creo Parametric*, deo paketa *Creo* (PTC)[55]

Priprema za štampu, tzv. „stono izdavaštvo“ (**DTP – desktop publishing**), je još jedna veoma raširena upotreba računara. Pomoću ovih programa, običan tekst se može dovesti u formu za štampu i objavljivanje. Ove aplikacije pružaju daleko veće mogućnosti od programa za obradu teksta na koje smo navikli (npr. smeštanje teksta u zakrivljene okvire, veće mogućnosti preloma

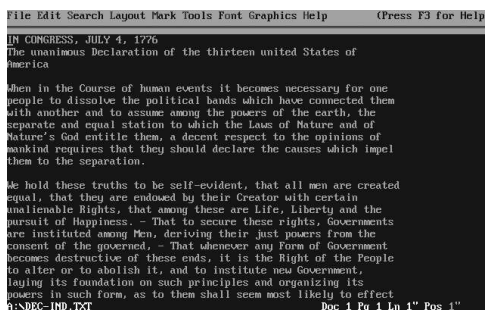
i kontrole nad krajnjim izgledom), ali, sa druge strane, nisu prilagođeni obimnijem unosu i izmenama teksta. U poznatije aplikacije spadaju *Adobe InDesign* i *QuarkExpress*.

Poslovanje

Vrlo brzo se pokazalo da računari ne moraju da se koriste samo u nauci i tehnici, već u svim oblastima života. Sa razvojem računara, postalo je moguće praviti softver koji je sve više okrenut korisniku i koji ne zahteva veliko tehničko obrazovanje. Jedna od najvećih sfera upotrebe računara je poslovna primena.

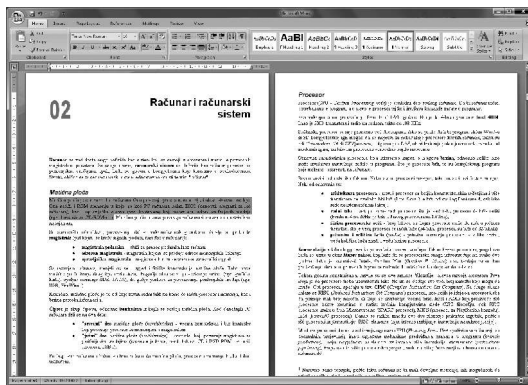
Obrada teksta

Verovatno najpopularnija upotreba računara je u obradi teksta i kreiranju dokumenata. Osnovna ideja koja stoji iza ovih programa jeste pomoć pri uređivanju teksta, posebno u vezi ispravljanja grešaka, prebacivanja i kopiranja istog teksta na više mesta ili pretraživanja i automatskih zamena teksta.



Slika 10.2. WordPerfect 5.1 (Satellite Systems International) DOS verzija nekada popularnog tekst editora iz 1989. godine. Moderna verzija ovog programa postoji i danas u sklopu paketa Corel Office

Sa napretkom softvera osnovne mogućnosti se stalno proširuju. Današnji programi za obradu teksta se svojim mogućnostima formatiranja sve više približavaju programima za stono izdavaštvo. Tekstu je moguće menjati izgled i poziciju na stranici, kao i kreirati čitave stilove kojima se definiše prikaz. Omogućen je rad sa grafikom, tabelama, listama, matematičkim formulama i sl.



Slika 10.3. Primer modernog alata za obradu teksta Word (Microsoft Office 2007)

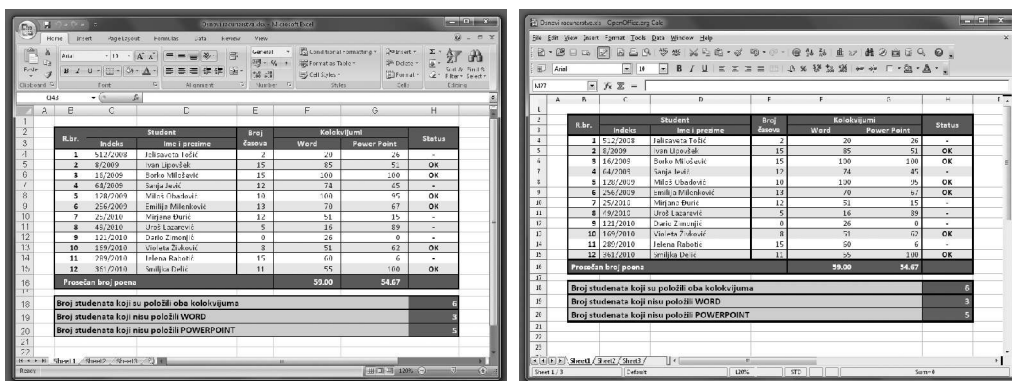
U smislu automatizacije, ovi programi donose mogućnosti automatskog numerisanja strana, kreiranja sadržaja, praćenja fusnota, direktnog slanja putem e-maila, ispravljjanje grešaka u kucanju i gramatici, kao i podršku za zajednički rad na istom dokumentu.

Obrada teksta polako dolazi do granica svoje funkcionalnosti – nema još puno novih mogućnosti koje se mogu ponuditi korisniku. Programi se u novije vreme sve više orijentišu na optimizaciju korisničkog interfejsa, odnosno na što ugodniji i jednostavniji rad.

Tabelarne kalkulacije

Nastanak softvera za tabelarne kalkulacije (*spreadsheet*) obeležio je program **VisiCalc** za računar *Apple II*. Ideja ovakvih programa je da se i poslovnim ljudima, koji se ne razumeju u programiranje i tehniku, pruži mogućnost vršenja proračuna i analize podataka.

Dokument u programima za tabelarne kalkulacije je tabela. U ćelije table se mogu unositi tekstualni ili numerički podaci. Osim toga, u ćelije se mogu ubaciti i formule koje vrše izračunavanje na osnovu podataka.



Slika 10.4. Primer tabelarne kalkulacije u programima Excel (Microsoft Office 2007) i Calc (OpenOffice.org 3)

Na slici se može videti primer jednostavne table. Iako tabela sadrži uglavnom podatke, u nekim ćelijama nalaze se formule kojima se podaci izračunavaju. Na primer, ćelija F16 (u kojoj se računa prosečan broj poena koje su studenti osvojili na testu iz Word-a) sadrži formulu:

$$=AVERAGE(F4:F15)$$

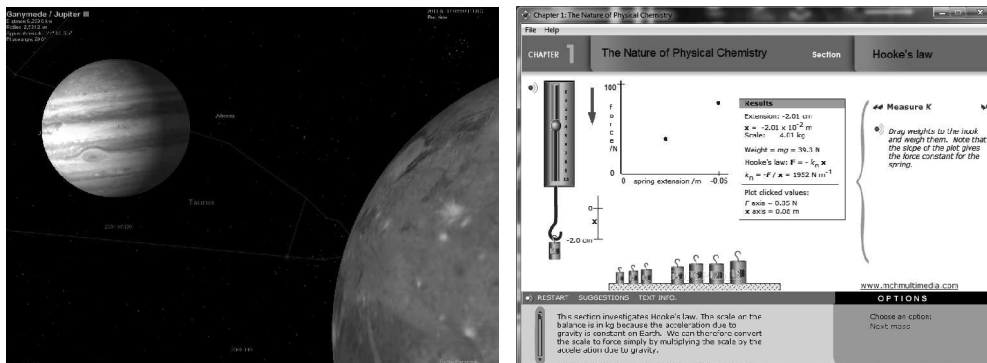
koja služi za izračunavanje aritmetičke sredine brojeva koji se nalaze u ćelijama od F4 do F15. Formule mogu varirati od najjednostavnijih matematičkih izraza poput, "=A5+3", pa do izuzetno kompleksnih. U programima za tabelarne kalkulacije obično postoji velika kolekcija unapred pripremljenih funkcija (poput navedene funkcije AVERAGE) koje olakšavaju izračunavanje.

Jedna od najbitnijih osobina ovih programa je mogućnost „šta-ako“ analize. Budući da se preračunavanja formula obavljaju automatski, moguće je napraviti tabelu i onda menjati početne podatke da bi se videlo kako se sa njima menjaju i rezultati. Ovaj tip softvera ima i velike mogućnosti za kreiranje grafikona na osnovu postojećih tabela.

Nauka i obrazovanje

Izuzetne matematičke mogućnosti i sposobnost izvršavanja programa logično su dovele do toga da se kompjuteri koriste u različitim **naučnim oblastima**. U ovo spadaju različite simulacije fizičkih pojava i hemijskih procesa, rešavanje matematičkih jednačina i crtanje grafikona, rad sa velikim brojevima, traženje uzoraka, statistički softver i sl.

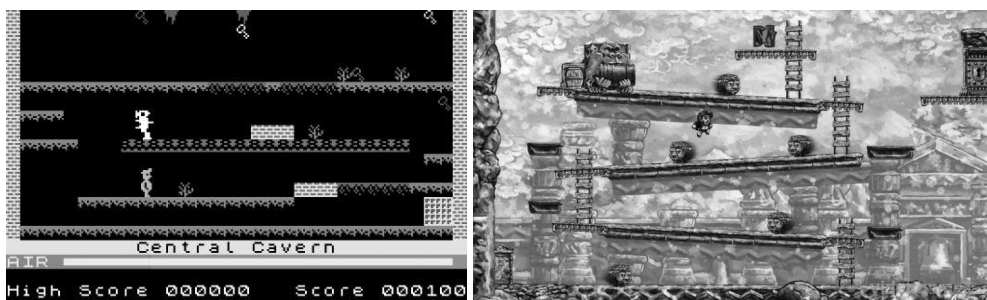
Edukativne aplikacije služe za obrazovanje. Mogu biti prilagođene svim uzrastima, počev od najmlađih (*activity centri*), pa do fakultetskog nivoa. Obrazovni programi mogu biti izvedeni u klasičnoj tutorskoj formi ili u obliku igre (eksplorativne ili kvizovi).



Slika 10.5. Simulacija i edukacija: *Celestia 1.6* i *The Nature of Physical Chemistry* (MHC Multimedia)

Računarske igre

Nemoguće je govoriti o računarima a ne pomenuti jednu od najmoćnijih sila – industriju zabave. Danas u svetu igara za računare, konzole ili mobilne telefone figurira veliki novac. Računarske igre su, kao i sav ostali softver, prešle dugačak put od najjednostavnijih do izuzetno kompleksnih ostvarenja.



Slika 10.6. Igre *Manic Miner*, izdata 1983. (Bug-Byte), i *Braid*, izdata 2008. (Jonathan Blow)

Akcionne igre su verovatno najrašireniji tip igara. Ove igre mogu biti izvedene na različite načine, ali bez obzira na to da li su u pitanju borilačke igre, igre u prvom licu (*FPS – First Person Shooter*), vožnja ili letenje, kod svih igara ovog tipa zahteva se brzina reagovanja i snalaženje u prostoru. Iako često kritikovane, ako se koriste u normalnoj meri, ove igre imaju pozitivan uticaj na razvoj motorike kod dece ili relaksirajuće dejstvo na odrasle.

Avanturističke i logičke igre, sa druge strane, predstavljaju intelektualni izazov – rešavanje logičkih zagonetki, zaključivanje na osnovu datih činjenica i ranijeg znanja, kombinaciju

elemenata, rešavanje slagalica. Igre ovog tipa mogu biti orijentisane isključivo na logičke ili strategijske probleme (razni tipovi slagalica, rešavanje određenog tipa logičkih zadataka, simulacija šaha), a mogu imati i radnju tokom koje igrač nailazi na različite probleme.



Slika 10.7. Jedna od najstarijih računarskih igara **Pong**, kreirana 1972. (Atari Inc), i tipičan primer FPS akcione igre – **Quake 4** (Raven Software) 2005.

Simulacije su tip igara kojima se simulira neko okruženje ili situacija iz realnog života. Tako postoje simulacije letenja, vožnje, sportske, strategijske, upravljanja. Ove igre, osim zabave, mogu ponuditi i određeni nivo obrazovanja, koji zavisi od nivoa realnosti koji nude, pa se tako budući piloti pre sedanja u kokpit aviona obučavaju na posebnim simulatorima letenja (koji doduše nemaju mnogo zabavnih elemenata). Sportske simulacije su po mnogima najciničniji tip kompjuterskih igara, kada se, umesto upražnjavanja sportske aktivnosti, ona izvodi na računaru. Strategijske igre predstavljaju simulaciju vođenja bitaka (ratne kampanje) i obično se dele na strategije u realnom vremenu i potezne. Simulacije upravljanja (menadžerske) su igre u kojima igrač vodi sportski klub, neki posao, grad, državu, kuću, farmu i sl.



Slika 10.8. Avanturističke igre **The Hobbit**, iz 1983. godine (Melbourne House), i **Monkey Island 2 Special Edition**, izdata 2010. godine (LucasArts)

Navedena je samo osnovna podela igara. Budući da je u pitanju milionska industrija, uvek se traži nešto novo. Tako se današnje igre retko mogu svrstati u samo jednu kategoriju. Često logičke igre zahtevaju i brze reflekse, u akcionim igrama se pojavljuju i logički problemi ili makar rešavanje lavirinta. Tako npr. postoji posebna podvrsta akcionih avantura, tzv. *survival horror*, zahteva od igrača da ne ulazi uvek u borbu sa neprijateljima, nego da ih izbegne i preživi.

Još jedan primer bile bi igre „glumljenja uloge“ (*RPG – Role-Playing Game*), nastale iz sličnog tipa društvenih (*FRP – Fantasy Role-Playing*) igara, koje od igrača zahtevaju da kreira

sopstveni lik sa određenim brojem osobina i onda se „uživi u ulogu“ u nekoj priči, zajedno sa ostalim igračima. Internet i mogućnost povezivanja su omogućile igračima sasvim novi nivo igranja RPG-a, gde se kroz različite „potrage“ (*quest*) upoznaju, druže, bore, saraduju i istražuju zamišljeni svet. Mogućnost povezivanja je u tolikoj meri promenila način igranja, da se danas u velikom broju igara kreira mogućnost tzv. on-line igranja protiv „živog“ protivnika.



*Slika 10.9. Multikorisnički RPG (MMORPG – Massively Multiplayer On-line RPG), igra **World of Warcraft** (Blizzard Entertainment)*

Osim za zabavu, postoji i tzv. profesionalno igranje – vrhunski igrači poredama na turnirima bukvalno mogu da zarađuju od igranja. U RPG svetu, igrači koji razvijaju veoma snažne likove ili sakupe određene artefakte mogu ih prodati slabijim igračima.

Danas su igre daleko ozbiljnije, kompleksnije i zabavnije od nekadašnjih „igrice“ i predstavljaju jedan od glavnih faktora napretka računarske industrije. Za uobičajenu poslovnu primenu – pisanje teksta, radne tabele i komunikaciju putem Interneta, dovoljan je i ispodprosečan računar. Sa druge strane, igre spadaju u jedan od najzahtevnijih oblika softvera – uvek tražeći maksimalnu brzinu i memoriju računara. Za igre se koriste najbolji i najbrži algoritmi, angažuju najbolji programeri i dizajneri.

11

Baze podataka i informacijski sistemi

Jedna veoma značajna klasa aplikativnog poslovnog softvera su baze podataka. Kao što i naziv govori, u pitanju je softver namenjen radu sa podacima. Kao takav, predstavlja jedan od tehnoloških osnova za kreiranje informacijskih sistema.

Baze podataka

Baze podataka predstavljaju specijalizovan softver koji primarno služi za **čuvanje velike količine podataka**. Pored toga, osnovne funkcije svake baze podataka su:

- sortiranje podataka;
- pretraga i izdvajanje podataka;
- transformacija podataka (kreiranje novih podataka na osnovu postojećih).

Sortiranje predstavlja mogućnost da se određeni podaci poređaju prema nekom unapred zadatom kriterijumu (npr. ređanje studenata po godini upisa, a unutar svake godine upisa po prezimenu) i to u opadajućem ili rastućem redosledu.

Iako je glavna funkcija baze podataka čuvanje ogromnog broja podataka, praktično nam nikada neće biti potrebni svi podaci odjednom. U stvari, uvek ćemo tražiti od baze da **izdvoji** samo određeni traženi podatak ili grupu podataka (npr. dosije određenog studenta ili spisak studenata koji su prijavili određeni ispit).

Konačno, sami podaci nam jesu značajni, ali vrlo često će biti potrebno da se **na osnovu postojećih podataka kreiraju novi podaci** (npr. prosečna ocena nekog studenta na osnovu njegovih ocena iz baze ili datum rođenja na osnovu matičnog broja). Ove transformacije podataka mogu varirati od vrlo jednostavnih, kao što je povezivanje kraćih tekstualnih podataka u jedan duži tekst, pa do veoma složenih matematičkih izračunavanja.

Modeli baza podataka

Tokom razvoja baza podataka na računarima, osmišljavani su različiti modeli za beleženje i povezivanje podataka.

Hijerarhijski model organizuje podatke u strukturu tipa stabla, gde određeni element može imati jedan ili više podređenih elemenata, ali uvek samo jedan nadređeni element. Osnovna veza koja se uspostavlja među podacima je jedan-prema-više.

Mrežni model je zasnovan na hijerarhijskom modelu, ali ga proširuje mogućnošću da svaki element može imati ne samo više podređenih, već i više nadređenih elemenata. Zahvaljujući ovom modelu moguće je ostvariti veze tipa više-prema-više.

Relacioni model beleži podatke unutar tabela između kojih se mogu uspostavljati veze.

Objektno-relacioni model predstavlja proširenje relacionog modela kojim se u podatke uvode i gotovi objekti – ovo mogu biti različiti dokumenti, audio i video zapisi i sl. Samim tim što su u pitanju dokumenti kojima se bave neke druge aplikacije (npr. radne tabele), proširuju se i mogućnosti same baze podataka.

Objektno-orientisani model nastaje proširivanjem mogućnosti objektno-orientisanih jezika radom sa bazama podataka. Ovo nije obično dodavanje mogućnosti da se bazama postavljaju upiti, već se podaci iz baze uvode kao objekti u same aplikacije.[33]

Osnovni koncepti relacionih baza podataka

Relacione baze podataka su i danas izuzetno zastupljene u informacionim sistemima. Zasnovane su na relacionom modelu, prema kome se podaci smeštaju unutar **relacija (tabela)** koje predstavljaju skupove **slogova (redova)** koji poseduju **atribute (kolone)**. Relacije (tabele) mogu biti **bazične** (koje sadrže same podatke) ili **izvedene** (rezultati kreirani na osnovu bazičnih). Operacije nad relacionom bazom podataka nazivaju se **relaciona algebra** i njen dobar deo potiče iz matematičke teorije skupova. To znači da se nad skupovima podataka mogu kreirati unije i preseki, kao i da se pronalaze podskupovi.[34]

Kod relacionih baza, podaci se beleže u obliku tabela, što nas na prvi pogled može asociirati na programe za tabelarne kalkulacije, međutim, postoje neke suštinske razlike.

Tabele u <i>spreadsheet</i> programima	Tabele u bazama podataka
Iako veoma velik, broj redova i kolona je unapred ograničen	Broj kolona je ograničen, ali broj redova zavisi samo od količine dostupne memorije na disku
Osnovni podatak je ćelija, a podaci su raspoređeni u niz – za svaku ćeliju se tačno zna „adresa“ i bitno je u kojoj ćeliji se nalazi koji podatak	Osnovni podatak je red (slog, zapis), a podaci su raspoređeni u skup – ne postoji „prvi“ ili „poslednji“ podatak – svi su ravnopravni, a kasnije se mogu sortirati prema nekom kriterijumu
Bilo koja ćelija može sadržati bilo koji podatak, bilo kog tipa (tekstualni, numerički i sl.)	Podaci imaju strukturu – svaka kolona sadrži istorodne podatke (npr. kolona za imena, kolona za datume rođenja i sl.)
Ćelije mogu sadržati podatke ali i formule za izračunavanja	U tabelama se beleže isključivo podaci

Tabela 11.1. Usporedni pregled karakteristika tabela u spreadsheet programima i bazama podataka

Osnovni koncepti sa kojima se susrećemo u radu sa bazama podataka su:

- tipovi podataka;
- normalizacija;
- primarni i strani ključevi;
- veze među tabelama i
- indeksi.

Numerički podaci	Tekstualni podaci
1	1
5	1125
6	12
12	32
32	5
579	579
1125	6

Tabela 11.2. Sortiranje numeričkih i tekstualnih podataka

Kako bi rad bio efikasniji, podaci u tabelama se moraju „tipizirati“ prema svom sadržaju. Tako baze podataka prepoznaju različite **tipove podataka** (tekstualni, celobrojni, brojevi sa decimalama, logički, datumski i sl).

Istina je da bi svaki podatak mogao da se predstavi u obliku teksta, međutim, rad sa takvim podacima bi narušio performanse celog sistema – sortiranje, pretraga, izračunavanje rezultata bilo bi veoma usporeno pošto bi podaci morali stalno da se pretvaraju iz teksta u odgovarajuće

tipove. U priloženoj tabeli možemo videti kako se razlikuje sortiranje brojeva, od brojeva predstavljenih u tekstualnom obliku. Brojevi koji su dati u numeričkom obliku sortiraju se po vrednosti, dok se brojevi dati u tekstualnom obliku sortiraju prema ciframa.

U sledećem primeru vidimo tabelu u kojoj su dati podaci o studentima i njihovim položenim ispitima.

INDEKS	GODINA	IME	PREZIME	ISPIT	PROFESOR	OCENA	POENI
5/2010	2010	Milan	Jovanović	INF	55	9	87
49/2010	2010	Saša	Arsenić	INF	55	6	55
156/2009	2009	Ana	Stojanović	PRA	19	7	66
5/2010	2010	Milan	Jovanović	STA	24	10	100
170/2010	2010	Jasna	Gajić	INF	4	6	55
170/2010	2010	Jasna	Gajić	PRA	19	7	65
5/2010	2010	Milan	Jovanović	PRA	19	6	59

Ono što primećujemo jeste da se jedan broj podataka ponavlja (za svaki ispit koji je neki student položio ponavljaju se svi podaci vezani za tog studenta). Mane su višestruke i očigledne – bespotrebno zauzimanje memorije, gubljenje vremena prilikom unosa podataka, veća mogućnost greške. Ovakva situacija je česta prilikom projektovanja baze podataka i rešava se kroz proces koji se naziva **normalizacija**. Normalizacija se na najjednostavniji način može definisati kao proces razdvajanja podataka na više tabela, tako da:

- ne postoje nepotrebna ponavljanja podataka (**prva normalna forma**);
- svi podaci budu direktno zavisni od primarnog ključa (**druga normalna forma**);
- svi podaci budu direktno zavisni od svih delova primarnog ključa, ako se primarni ključ sastoji iz više polja (**treća normalna forma**).[35]

Suštinski, najvažniji rezultat normalizacije je uklanjanje nepotrebnih ponavljanja u podacima. Osim toga, svako polje bi trebalo da bude nezavisno – uz mogućnost da se promeni bez menjanja drugih polja. Takođe, polja ne bi trebalo da sadrže podatke koji se izračunavaju na osnovu već postojećih polja.

Najjednostavniji primer normalizacije u ovom slučaju jeste razdvajanje na dve tabele – u jednoj se nalaze samo podaci o studentima, a u drugoj podaci o položenim ispitima.

INDEKS	GODINA	IME	PREZIME
5/2010	2010	Milan	Jovanović
49/2010	2010	Saša	Arsenić
156/2009	2009	Ana	Stojanović
170/2010	2010	Jasna	Gajić

STUDENT	ISPIT	PROFESOR	OCENA	POENI
5/2010	INF	55	9	87
49/2010	INF	55	6	55
156/2009	PRA	19	7	66
5/2010	STA	24	10	100
170/2010	INF	4	6	55
170/2010	PRA	19	7	65
5/2010	PRA	19	6	59

Već je rečeno da jedan red podataka u tabeli predstavlja jedan element skupa. Kao što je poznato, svaki element skupa je jedinstven (ne mogu postojati dva ista), što znači da je u tabeli zabranjeno da se pojave dva potpuno ista reda podataka. Međutim, neki podaci se zbog svoje prirode moraju ponavljati (npr. sasvim je moguće imati dva studenta sa istim imenom). Zbog toga svaka tabela mora imati bar jednu kolonu (ili kombinaciju kolona) u kojoj će svi podaci garantovano uvek biti različiti – tzv. **primarni ključ**. Primarni ključ predstavlja podatak koji jedinstveno identifikuje svaki red tabele.

Ukoliko među podacima ne postoji podatak koji bi bio prihvatljiv kao primarni ključ (matični brojevi građana, broj indeksa...), moramo kreirati neki nov podatak – veštački primarni ključ (npr. ako uvedemo neku novu šifru kao podatak).

U gornjem primeru, za studente je broj indeksa sasvim prihvatljiv kao primarni ključ. Sa druge strane, u tabeli položenih ispita primećujemo da ne postoji kolona u kojoj će podaci biti uvek različiti. Međutim, ako pažljivije osmotrimo, vidimo da je svaki student mogao samo jednom da položi jedan predmet, pa tako kombinacija kolona student+ispit predstavlja primarni ključ.

Da bi se uspostavila veza između dve tabele potrebno je da postoje kolone u kojima se nalaze odgovarajući podaci. Polje koje služi za povezivanje sa drugom tabelom naziva se **strani ključ**. U primeru, polje student iz tabele o ispitima služi za povezivanje sa tabelom o studentima i samim tim predstavlja strani ključ.

Razdvojene tabele se mogu povezivati kako bi se dobili kompleksni rezultati prilikom upita. Postoje **tri tipa veze** koje se mogu uspostaviti među tabelama.

- **Jedan-prema-jedan.** Jednom podatku iz jedne tabele odgovara tačno jedan podatak iz druge tabele. Ova veza se ostvaruje povezivanjem dva primarna ključa. Npr. možemo imati dve tabele sa podacima o studentima – u jednoj se mogu naći podaci iz dosijea studenta, a u drugoj podaci o njegovom uspehu na prijemnom ispitu. Jednom studentu odgovara tačno jedan podatak iz tabele prijemnog ispita.
- **Jedan-prema-više.** Jednom podatku iz jedne tabele odgovara jedan ili više podataka iz druge tabele. Ova veza nastaje povezivanjem primarnog i stranog ključa. Npr. jedan student može imati više položenih ispita koji se beleže u tabeli ispiti.
- **Više-prema-više.** Ova veza nastaje kada za više podataka iz jedne tabele postoji više odgovarajućih podataka iz druge tabele. Veze više-prema-više u stvari predstavljaju dve uzajamne veze jedan-prema-više. Primer bi bila situacija kada jedan profesor može držati više različitih predmeta u školi, a istovremeno na jednom predmetu može biti više profesora. Ovakve veze se obično ne kreiraju direktno među tabelama, već postoji jedna posrednička tabela koja služi za povezivanje.

Jedan od osnovnih zadataka baze podataka je njihovo pretraživanje. Međutim, što ima više podataka, proces pretraživanja i sortiranja postaje sve sporiji. Ovaj problem se rešava uvođenjem **indeksa**. Indeksi su „interne“ tabele kojima manipuliše sama baza podataka. U njima se nalaze podaci koji su već sortirani, tako da je pretraživanje veoma ubrzano. Sa druge strane, unos, brisanje i promena podataka postaje sporija, pošto za svaku promenu treba osvežavati i sam indeks.

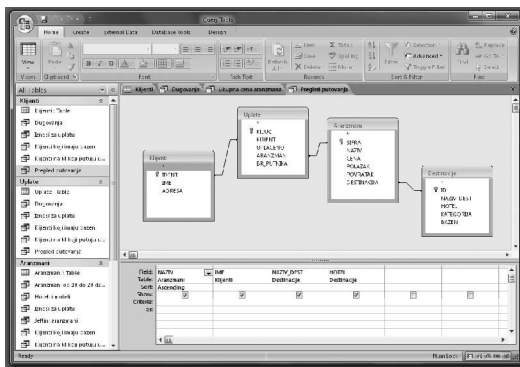
Upravljanje bazom podataka

Određen broj baza podataka pruža mogućnost upravljanja putem **grafičkog korisničkog interfejsa**, što se uglavnom odnosi na manje sisteme (primer bi bio *Microsoft Access*). Ovi sistemi su namenjeni korisnicima koji nemaju iskustva u programiranju baza podataka, a žele da na jednostavan način rade sa njima. Iako mali, ovi sistemi mogu imati čak i poslovnu primenu, a svakako im ide u prilog to što se aplikacije lako i brzo razvijaju.

SQL (*Structured Query Language*) predstavlja poseban jezik za upravljanje bazama podataka. Svi ozbiljniji sistemi, koji omogućavaju pristup podacima kroz mrežu, poseduju mogućnost izvršavanja SQL komandi, tzv. „upita“. Komande SQL-a dele se na dve grupe:

- komande za definiciju tabela (**DDL** – *Data Definition Language*), za kreiranje samih baza, tabela, definisanje strukture (kolona iz kojih se sastoje i tipova podataka) i
- komande za rad sa podacima (**DML** – *Data Manipulation Language*), kojima se vrši pretraživanje, sortiranje, unos, izmene, brisanje.

Veliki broj programskih jezika pruža mogućnost povezivanja sa različitim bazama podataka, prosleđivanja SQL komandi i prihvatanja rezultata. Iako postoje razlike u „dijalektima“, sam SQL je više-manje isti u svim bazama podataka, pa tako, uz veoma male izmene, softver informacionog sistema može pristupiti različitim sistemima baza podataka (neki od najpoznatijih su *Oracle, Microsoft SQL Server, MySQL, Interbase*).



Slika 11.1. Grafički korisnički interfejs za rad sa bazom podataka u programu Access (Microsoft)

SQL jezik je razvijen na osnovu već pomenute funkcionalne paradigme kako bi podržao koncepte relacionog modela. Upit se ne sastoji iz niza koraka koje treba izvršiti već iz jednog jedinog iskaza koji, doduše, može biti i veoma kompleksan i sadržati podiskaze. Jedan od tipičnih DML SQL upita je SELECT iskaz kojim se zahtevaju određeni podaci iz baze:

```
SELECT indeks, ime, prezime, AVERAGE(ocena)
FROM studenti, ispiti
WHERE (indeks=student) AND (godina=2010)
GROUP BY indeks, ime, prezime ORDER BY prezime, ime
```

Navedeni upit je dat na osnovu tabela iz primera – traže se **brojevi indeksa, imena, prezimena i prosečne ocene** studenata generacije 2010, poređane po prezimenima i imenima.

Informacioni sistemi

Informacioni sistem bi mogao da se definiše kao skup elemenata koji funkcionišu kao jedna celina u cilju čuvanja, prenosa i obrade podataka. Nauka koja se bavi proučavanjem informacionih sistema naziva se **Informatika**.

Podatak i informacija

Na ovom mestu bi bilo dobro da razumemo razliku između ova dva pojma.

Podatak predstavlja neku činjenicu. Sam po sebi, broj **9,81** može i ne mora ništa da nam znači. To je neka „suva“ činjenica, ili, jednostavno rečeno – samo običan broj.

Sa druge strane, **informacija je saznanje** koje dobijamo iz podatka, odnosno njegovog značenje. Kada „saznamo“ da je ubrzanje Zemljine gravitacije 9,81 m/s², ili da je prosečna ocena studenata na nekom ispitu 9,81, možemo reći da smo stekli informaciju.

Kako je već pomenuto u prvom poglavlju, informacioni sistem se može implementirati i bez korišćenja računara, međutim, moderne tehnologije (baze podataka, tabelarne kalkulacije, računarske mreže i web) su praktično savršene za kreiranje efikasnih i upotrebljivih informacionih sistema, pa se tako informacioni sistemi generalno uvek vezuju za računare.

Teorija sistema je od posebnog značaja za informacione sisteme – suština je u neprekidnom sagledavanju problema i njegovih delova, kako su oni međusobno povezani i kako sačinjavaju celinu. Prilikom projektovanja sistema uvek se ide ka oblikovanju celine, a ne izolovanih podsistema. Teorija sistema uvodi i pojmove otvorenog, odnosno zatvorenog sistema. Ako je sistem otvoren (poslovni sistemi su gotovo uvek takvi), onda se utvrđuje koliki i kakav je uticaj okruženja. Samim tim, informacioni sistem treba da obezbedi usklađenost sa okruženjem.

Osnovni elementi informacionog sistema su [32]:

- **hardver** (*hardware*) – tehničke komponente sistema koje obuhvataju računare i mrežnu opremu;
- **softver** (*software*) – aplikacije koje se koriste za čuvanje i pristup podacima, njihovu obradu i distribuciju, kao i sami podaci;
- **personal** (*lifeware*) – ljudi koji koriste informacioni sistem, kao i stručnjaci koji učestvuju u njegovom razvoju i održavanju;
- **organizacija** (*orgware*) – metodi i postupci koji se koriste prilikom rada informacionog sistema.

Razvoj informacionih sistema

Istorijski gledano, informacioni sistemi su prošli kroz nekoliko faza razvoja [32]:

Informacioni sistemi za obradu podataka (*DP – Data Processing*). Ovo su najprimitivniji informacioni sistemi. Orijetisani su na obradu poslovnih transakcija. Sami po sebi predstavljali su veliki napredak u odnosu na poslovanje bez upotrebe informacionih tehnologija, budući da su uveli automatizaciju u poslovanje (zamena ručne obrade podataka mašinskom) i samim tim povećali brzinu i tačnost. Međutim, kao takvi, bili su orijetisani isključivo na analizu prošlih događaja. Softver za ove sisteme se kreira u klasičnim programskim jezicima, a podaci moraju biti posebno pripremljeni (grupisani i obrađeni) pre unosa u sistem. Ovi sistemi su zahtevali stalnu prisutnost programera i drugih stručnjaka, pošto se sve dešavalo u računskim centrima.

Upravljački informacioni sistemi (*MIS – Management Information Systems*). Sledeća stepenica u razvoju informacionih sistema nastala je iz potrebe da se obrađene informacije upotrebe za donošenje upravljačkih odluka. Njihov nastanak omogućen je razvojem računarskih mreža i baza podataka. Bave se problemima ekonomičnosti poslovanja, analizom tržišta i cena.

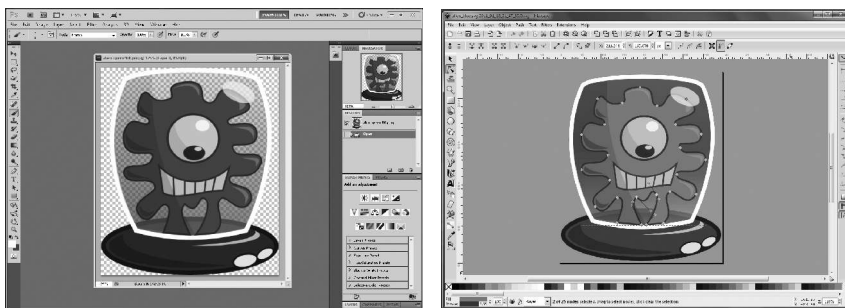
Informacioni sistemi za podršku odlučivanju (*DSS – Decision Support Systems*). Ovi informacioni sistemi se bave ne samo rešavanjem rutinskih, već i specifičnih problema. Namenjeni su top menadžmentu. Da bi ispunili tu ulogu, moraju biti fleksibilni i jednostavni za korišćenje (unapređen korisnički interfejs, grafički prikaz, elektronska pošta). Upotrebljivi su za donošenje taktičkih odluka, a orijetisani su na sadašnje i buduće događaje.

Ekspertni sistemi (*ES – Expert Systems*). Najviši nivo razvoja informacionih sistema su ekspertni sistemi. Ekspertni sistemi predstavljaju jedan od produkata razvoja veštačke inteligencije, koji sublimira znanje stručnjaka za neku oblast. Kao takvi, ovi sistemi više ne koriste običnu bazu podataka, već „bazu znanja“ koja funkcioniše zajedno sa pravilima odlučivanja. Ovi sistemi su sposobni da samostalno uče, pamte dobra rešenja korisnika, a kada dođu do nekog zaključka imaju mogućnost i da ga objasne. Koriste se za donošenje najtežih, dugoročnih strateških odluka.

Jedna od osnovnih upotreba računara je u vizuelizaciji podataka. Bez obzira na to da li je u pitanju posao, umetnost ili zabava, računarska grafika je postala važan deo naše svakodnevice.

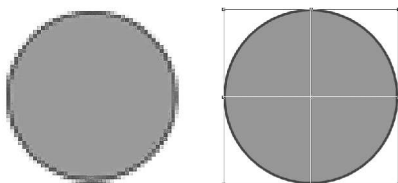
Rasterska i vektorska grafika

Grafiku na računaru, pre svega, možemo podeliti prema načinu na koji se u memoriji pamte podaci o slici. Ranije je bilo reći o predstavljanju grafike na računaru i, iako se sva grafika mora pretvoriti u tačkice u RGB sistemu kako bi se prikazala, sama grafička aplikacija može pamti sliku na dva osnovna načina – kao **rastersku (bitmap) grafiku**, kod koje program pamti boju svakog pojedinog piksela slike, i kao **vektorsku grafiku**, kod koje se pamte geometrijski oblici iz kojih se slika sastoji.



Slika 12.1. Naizgled ista slika u dva različita programa: bitmap verzija u programu **Photoshop** (Adobe) i vektorska verzija u programu **Inkscape 0.48** (www.inkscape.org)

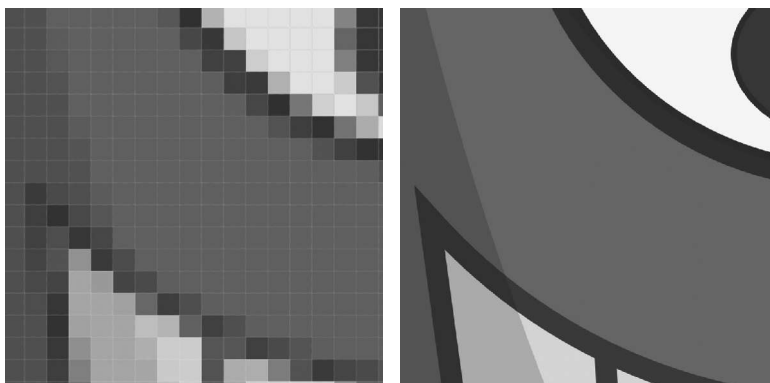
Najjednostavniji primer koji ovo ilustruje je slika kruga. Ukoliko se slika u memoriji računara drži u obliku bitmap grafike, svaka tačkica slike se posebno pamti. Količina memorije koju slika zauzima direktno je zavisna od broja piksela – što je slika veća, biće potrebno više memorije.



Slika 12.2. Levi krug je predstavljen kao bitmap grafika (svaki piksel se pamti u memoriji računara); desni krug je predstavljen kao vektorska grafika (pamte se samo koordinate centra, poluprečnik i boja)

Sa druge strane, kod vektorske grafike, u memoriji računara pamte se samo koordinate centra, poluprečnik i boja kruga. Bez obzira na to koliki je krug, za beleženje ove slike u memoriji biće potrebno svega nekoliko bajtova. Zauzimanje memorije kod vektorske grafike ne zavisi od veličine slike već od broja elemenata koji sačinjavaju sliku.

Dobre strane **rasterske** grafike su u tome što **slika može predstavljati bukvalno bilo šta** – od dijagrama do fotografije. Kod pomenutog kruga, svaka njegova pojedina tačka može biti različite boje. Loše strane se vide kada se slika uveća – gubi se na kvalitetu prikaza, a povećava se zauzeće memorije. Kod **vektorske** grafike je upravo prednost u tome što se **slika može umanjivati ili uvećavati** bez bilo kakvog gubitka kvaliteta i trošenja memorije, što je izuzetno cenjeno kod **pripreme za štampu**, gde postoji potreba i za veoma velikim formatima (plakati, bilbordi). Međutim, kod takve vrste grafike ograničeni smo **samo na crteže** (fotografije se ne mogu predstavljati vektorski). Ukoliko je slika veoma kompleksna, utoliko će **procesor biti više opterećen** pri svakom iscrtavanju slike.



Slika 12.3. Prilikom uveličavanja slike, jasno se vidi gubitak kvaliteta prikaza i pikseli iz kojih se slika sastoji; kod vektorske grafike pri bilo kom nivou uvećanja nema nikakvog gubitka

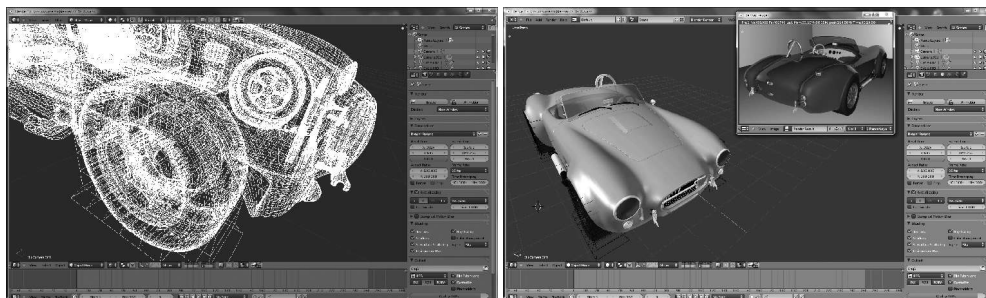
Danas svi programi za crtanje podržavaju mogućnost kombinovanja ova dva tipa grafike. Kao što je u moderne alate za obradu bitmap grafike moguće ubacivati vektorske oblike, tako je moguće ubacivati bitmap objekte i tekstone u programe za obradu vektorske grafike.

3D grafika i animacija

Trodimenzionalna grafika predstavlja korak dalje u odnosu na prethodno opisanu vektorsku grafiku. I u ovom slučaju se elementi slike čuvaju u vidu matematičkih podataka, s tim što su ovaj put u pitanju **koordinate tačaka u prostoru** umesto u ravni. To znači da sada objekti nemaju samo širinu i visinu, već i dubinu. Objektu je zatim moguće definisati **materijal** i dodeliti mu određena svojstva – boju, teksturu, reljef, transparentciju (nivo providnosti), refrakciju (prelamanje svetlosti kroz providne objekte), refleksiju (kako se okolina ogleda u objektu) i sl. Tako definisani objekti se postavljaju na „scenu“ zajedno sa svetlosnim izvorima i kamerama.

Na osnovu zadatih parametara, računar generiše prikaz scene – kako bi sve to izgledalo gledano okom kamere. Ovaj proces senčenja naziva se „praćenje zraka“ (**ray tracing**) i bukvalno prati kretanje svetlosti od izvora do kamere. Zavisno od toga koliko je slika kompleksna i koliki nivo realnosti se zahteva, za prikaz će biti potrebno od nekoliko sekundi do nekoliko sati.

Ovi programi obično imaju i mogućnost animacije. Iluzija pokreta se stvara naizmeničnim i brzim smenjivanjem slika. Kod tradicionalne animacije, potrebno je crtati svaku pojedinu sliku, odnosno frejm (*frame*). Međutim, vektorska animacija se zasniva na podešavanju ključnih slika (*keyframe*), odnosno tzv. „ekstrema“ u pokretu, dok kompjuter izračunava međufaze. Na ovom principu funkcionišu programi za 2D vektorsku animaciju (npr. *Adobe Flash*), kao i programi za 3D animaciju (*3D Studio Max*, *Lightwave*, *Maya*, *Blender*).

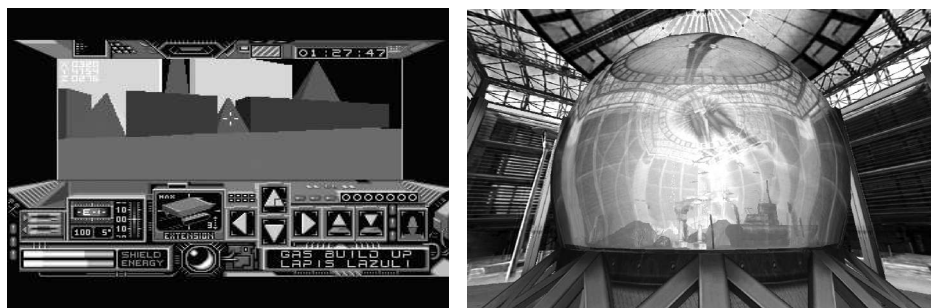


Slika 12.4. Model automobila konstruisan i osenčen u programu **Blender 2.57** (www.blender.org)

Programi za 3D animaciju posebno uključuju i mnogobrojne dodatke za postizanje realističnih specijalnih efekata – vatra, voda, krzno i kosa, dim, lišće, kao i simulaciju fizike – gravitaciju i inerciju. Iako danas postoje problemi koji su i dalje nerešivi za 3D programe (npr. realistična kinematika odeće), nije daleko dan kada 3D animaciju nećemo razlikovati od igranog filma.

3D animacija može biti postignuta i u realnom vremenu (računar generiše sliku veoma brzo – mnogo puta u sekundi, što omogućava animiran prikaz). Animacija u realnom vremenu se koristi kada je nemoguće unapred pripremiti gotovu video sekvencu, budući da dešavanja u animaciji zavise od korisnika (razne simulacije, virtuelna realnost, kompjuterske igre).

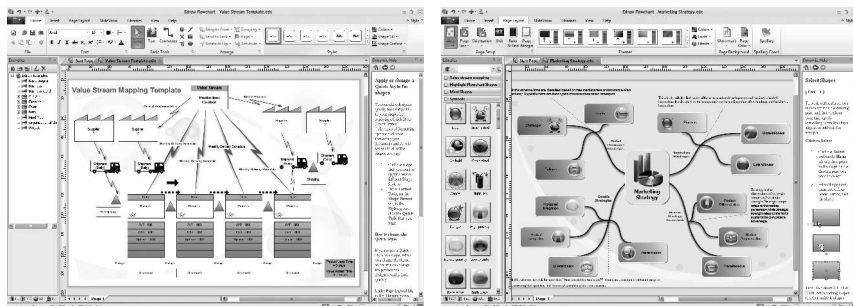
Danas je moguće postići 3D animaciju u realnom vremenu i na kućnim računarima zahvaljujući sve većoj brzini procesora, kao i specijalizovanom hardveru (grafičke kartice za 3D akceleraciju). Ipak, ovo je ostvareno uz gubitak realističnosti prikaza (uprosćeniji modeli, korišćenje nerealne fizike, kao i korišćenje jednostavnijih algoritama za senčenje). Ovi nedostaci se nadoknađuju specijalnim efektima i bogatstvom tekstura, što prikaz možda neće učiniti realističnijim, ali svakako hoće atraktivnijim.



Slika 12.5. Napredak 3D animacije u realnom vremenu: **Driller** (Major Developments), igra iz 1987. godine, jedna od prvih koje su simulirale 3D prostor, i **WebGL Aquarium** demo (Human Engines and Gregg Tavares) – 3D animacija u web čitaču (www.chromeexperiments.com)

Poslovna grafika

Kompjuterska grafika se može koristiti i u poslovne svrhe i predstavlja jedan od važnijih faktora bilo koje poslovne prezentacije ili izveštaja. Već i postojeće *Office* aplikacije imaju odlične mogućnosti za kreiranje različitih grafikona i dijagrama, a ukoliko njihove mogućnosti nisu dovoljne, utoliko postoji i softver specijalizovan za kreiranje ove vrste grafike.



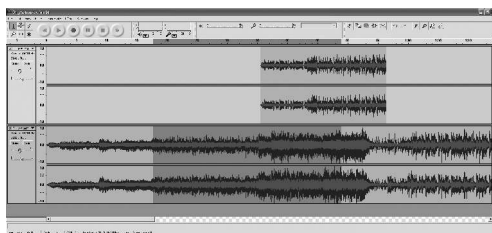
Slika 12.6. Primeri dijagrama u programu Edraw (EdrawSoft)

Suštinski, ovaj tip grafike je vektorski ali sa naglašenim mogućnostima povezivanja elemenata i rada sa tekstom. Slobodno crtanje je donekle zanemareno, a dijagrami se u najvećem broju slučajeva formiraju biranjem, uklapanjem i povezivanjem već postojećih elemenata.

Multimedija

Multimedija označava istovremeno korišćenje teksta, slike, zvuka, animacije i videa radi prenosa informacija. Termin je počeo šire da se upotrebljava kada su PC računari sa adekvatnim hardverskim mogućnostima postali dovoljno jeftini i omasovljeni. Drugim rečima, to se desilo kada su se pojavile jeftine zvučne kartice i kućni PC računari stekli dovoljno memorije i dovoljno dobru grafiku. U to vreme je počela i masovna proizvodnja jeftinih CD čitača za PC računare, što je omogućilo da se (za to vreme) velika količina podataka prenosi putem kompakt diska.

Za razliku od običnog TV-a, PC računari pružaju **interaktivnost** u pregledu multimedijalnih sadržaja – korisnici mogu pretraživati sadržaje ili birati šta žele da pogledaju. Ovo je u najvećoj meri omogućeno upotrebom **hiperteksta** – teksta koji u sebi sadrži hiperlinkove (pojmove koji su posebno obeleženi i koje korisnik može izabrati i tako preći na sledeću temu). Osim teksta, slika takođe može sadržati hiperlinkove, a nove tehnologije to omogućavaju i za video. Ovakav tip multimedije naziva se još i **hipermedija**.



Slika 12.7. Obrada zvuka u programu Audacity 1.2.3 (Dominic Mazzoni)

Sama multimedija povezuje različite elemente, pa je tako za kreiranje multimedijalnog materijala potrebno nekoliko tipova aplikacija.

Obrada zvuka uključuje u sebe programe za snimanje i obradu digitalizovanog zvuka, kao i aplikacije za punu muzičku produkciju. Aplikacije kojima se manipuliše digitalizovanim zvukom (*audio editing*) omogućavaju skraćivanje, povezivanje, primenu efekata (eho, uklanjanje šuma, distorzije, promena jačine, ritma i visine) i sl.

U ovim programima je moguće i digitalizovati zvuk putem mikrofona i snimati ga u različitim formatima. Iako ovi programi imaju mogućnost miksovanja nekoliko različitih semplova, za to se već koristi softver za muzičku produkciju.



Slika 12.8. Aplikacije za muzičku produkciju variraju od početničkih do profesionalnih. Ovde vidimo bogat korisnički interfejs u aplikacijama Music Creator 5 i Studio Instruments (Cakewalk) [56][57]

Aplikacije za muzičku produkciju imaju drugačiji pristup – orijentisane su na kombinaciju i miksovanje velikog broja semplova, kako bi se dobio rezultat. Pozadinska muzika se obično sastoji iz kratkih audio sekvenci (*loop*-ova) koje se lako uklapaju. Ovi programi takođe imaju mogućnost rada sa MIDI muzikom (notni zapis namenjen klavijaturama). Osim prostog uklapanja, programi omogućavaju promenu balansa i jačine semplova u realnom vremenu. Postoje i samostalni delovi programa koji služe za sintetizovanje sopstvenih *loop*-ova, na osnovu digitalizovanih instrumenata. Neki od najpoznatijih muzičkih alata (**DAW – Digital Audio Workstation**) su Acid (Sony), Cubase (Steinberg) i Cakewalk Sonar (Roland).

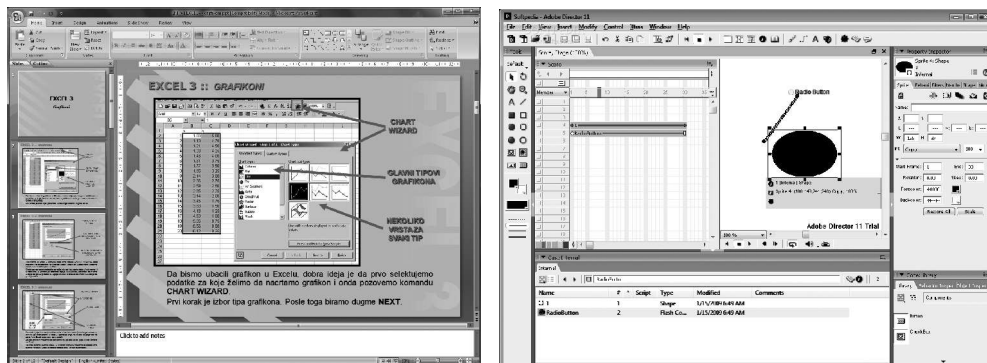


Slika 12.9. Nelinearna video montaža u aplikaciji Premiere 6.0 (Adobe) i alat za multimedijalnu produkciju TrakAxPC 3.01 (HighAndes)

Poput programa za muzičku produkciju, **softver za obradu videa i montažu** uvodi tzv. **nelinearnu montažu**. Za razliku od linearne montaže koja se odnosi na rad sa filmskim i video trakama, nelinearna montaža omogućava kombinovanje više inserata istovremeno, uz dodavanje tranzicija i efekata.

Aplikacije za kreiranje **multimedijalnih prezentacija** se uglavnom baziraju na jednom od dva pristupa. Prvi je kreiranje prezentacije kroz tzv. **slajdove**. Svaki slajd može sadržati tekst, grafiku, video i zvuk. Elementi unutar slajda mogu biti animirani automatski ili reagovati na akcije korisnika. Prelaz sa slajda na slajd takođe može biti animiran. Ovaj pristup podržavaju programi koji su deo različitih *office* paketa (*Microsoft Office PowerPoint* i *OpenOffice Impress*). Ove aplikacije su više namenjene kreiranju poslovnih prezentacija.

Drugi pristup tretira prezentaciju kao **animaciju** sa određenim brojem slika (*frame-ova*) u sekundi i ključnim *frame-ovima* raspoređenim na određenim pozicijama duž vremenskog toka. Ovi ključni *frame-ovi* sadrže početne i finalne pozicije elemenata prezentacije, koje se automatski menjaju tokom faznih *frame-ova*. Programi koji imaju ovakav pristup su *Adobe Flash* i *Director*. Ovakve prezentacije obično pružaju bogatije multimedijalno iskustvo u odnosu na prezentacije zasnovane na slajdovima.



Slika 12.10. Prezentacija zasnovana na slajdovima u programu *PowerPoint 2007* (Microsoft) i prezentacija zasnovana na animaciji u programu *Director 11.5* (Adobe) [58]

Računari se danas veoma mnogo koriste u svrhu komunikacije. Da bi se komunikacija uspostavila, potrebno je da računari budu međusobno povezani. Jednom kada se uspostavi mreža putem koje računari mogu razmenjivati podatke (dakle, bez korišćenja medija spoljne memorije), svaka informacija, koja se može zabeležiti u digitalnom obliku, može biti preneti sa jednog računara na drugi. Tako danas putem Interneta možemo razmenjivati dokumente, sliku, zvuk, video zapise ili aplikacije.

Iako se danas tako misli, Internet nije zaista prvi omogućio komunikaciju korisnika putem računara. Korisnici su komunicirali i mnogo pre nego što je Internet postao dostupan svima, putem elektronskih biltena (**BBS** – *Bulliten Board System*). Ovi sistemi su funkcionisali od kraja 70-ih do sredine 90-ih godina, kada ih je Internet konačno potisnuo. Reč je o sistemima terminalskog pristupa na koje su se korisnici povezivali putem telefonske linije. Za današnje pojmove, takav sistem je bio prilično neugledan – u najvećem broju slučajeva koristio se čisto tekstualni režim i najveći estetski domet bila su slova u boji, što je i logično s obzirom na malu brzinu prvih modema i snagu računara. Ipak, korisnici BBS-ova mogli su da razmenjuju privatne poruke, fajlove, kao i da učestvuju u javnim diskusijama. Sa povećanjem popularnosti BBS-ova, nastao je i **FidoNet** – mreža koja je povezivala BBS-ove, što je omogućilo komunikaciju korisnika sa različitim sistemima. Ova komunikacija je bila spora, pošto su BBS-ovi morali da čekaju period u kome ima malo korisnika (većina BBS-ova su bili besplatni – u vlasništvu entuzijasta, funkcionisali samo tokom noći i imali jednu telefonsku liniju), kako bi se međusobno povezivali.

Interesantna mogućnost terminalskih programa za pristup BBS-ovima bila je povezivanje dva korisnika putem telefonske veze, kada bi mogli da „časkaju“ (*chat*) ili razmenjuju fajlove.

Internet kao mreža

Internet predstavlja globalnu mrežu međusobno povezanih mreža i računara. Osnovni koncept nastao je 1962. godine, kada je pred američku vojnu agenciju **ARPA** (*Advancet Research Project Agency*) postavljen zadatak da osmisli metod kojim bi vojna mreža SAD ostala operativna čak i u slučaju nuklearnog udara. Ideja na kojoj se zasnivalo rešenje problema bila je paketni prenos podataka kroz mrežu. **Paketni prenos** označava deljenje podataka na pakete (datagrame), koji se nezavisno šalju kroz mrežu. U slučaju pada jednog ili više čvorova, paketi će biti usmereni drugačijom rutom i ipak stići na određište. 1968. nastaje preteča Interneta – **ARPAnet**. Mreža se razvijala tokom par decenija kao vojni, a kasnije i akademski projekat, a šira javnost je počela da koristi Internet od 90-ih godina, sa razvojem WWW servisa.

Internet je zasnovan na **klijent-server arhitekturi** mreže. To znači da se na Internetu suštinski mogu pronaći samo dve vrste računara – klijenti i serveri. **Serveri** su oni kompjuteri koji sačinjavaju Internet onako kako ga većina ljudi zamišlja. Uloga servera je da „osluškuje“ zahteve koje dobija preko mreže i da ih izvršava. **Klijenti** su računari običnih korisnika koji se povezuju na Internet i koriste njegove servise. Komunikacija na Internetu funkcioniše kao razmena pitanja (zahteva) i odgovora, koju iniciraju klijenti.

Međutim, osim klasičnog povezivanja „jedan server – više klijenata“, postoji i alternativni način kojim se korisnici Interneta povezuju u svoju internu mrežu. Ovakve mreže slobodnih čvorova (**P2P – peer-to-peer**) funkcionišu tako što su korisnici povezani „svaki sa svakim“ preko svojih IP adresa, što je omogućeno programima koji su specijalizovani za neku namenu kojoj ovakav način povezivanja odgovara. Suštinski, takav program predstavlja istovremeno i klijent i server za druge korisnike. Da bi se neko priključio na P2P mrežu, dovoljno je da pronađe jednog korisnika koji je već povezan. Preko ovakvih sistema funkcioniše npr. sistem za on-line komunikaciju *Skype*, kao i veliki broj programa za razmenu fajlova – npr. *BitTorrent* klijenti.

Da bi komunikacija između različitih mreža, računara, operativnih sistema i aplikacija uopšte bila moguća, bilo je potrebno uvesti **standarde za prenos podataka**, koji tačno definišu način na koji se zahtevi upućuju serveru i u kom obliku mogu biti njegovi odgovori. Ovi standardi nazivaju se **protokolima**. Postoje protokoli **nizih** (npr. način na koji se paketi prenose od računara do računara kako bi stigli na određite – IP, TCP, UDP) ili **viših nivoa** (komunikacija web čitača i servera prilikom posete nekoj prezentaciji – HTTP, razmena elektronske pošte – POP3, SMTP i sl). U sledećem primeru je dat deo komunikacije između klijenta i servera putem **FTP protokola** (*File Transfer Protocol*), koji se koristi za prenos fajlova.

```
USER korisnik          331 Please specify the password.
PASS *****          230 Login successful.
SYST                   215 UNIX Type: L8
CWD /var/www/html     250 Directory successfully changed.
                       Connect ok!
PWD                    257 "/var/www/html"
```

Sa leve strane prikazane su komande koje klijent upućuje serveru, a sa desne odgovori servera.

Internet kao medij

Internet je mreža koja se stalno razvija i povećava. Ne samo da se razvija u tehnološkom pogledu (stalno povećanje brzine protoka podataka nove tehnologije i načini komuniciranja), već se stalno povećava i broj korisnika. Poslednji izveštaji za 2011. godinu procenjuju da Internet koristi nešto preko 2 milijarde ljudi, što je skoro trećina svetske populacije.[36]

Kao takav, Internet potpuno menja način komunikacije među ljudima, kao i način života. **Informacije su dostupne svakome i svuda**. Bez obzira na geografsku udaljenost, korisnici mogu komunicirati ili preuzimati podatke i fajlove sa bilo koje lokacije ili u bilo koje doba.

Internet je omogućio **interaktivnost** kakvu ne pruža nijedan drugi medij. Korisnici mogu pronaći tačno one sadržaje koji ih interesuju. Samim tim, Internet više ne predstavlja simulaciju realnog okruženja, već samostalno, alternativno virtuelno okruženje.

Internet je takođe **dinamičan medij** – stalno rastući, neprestano razvijajući drugačije načine komunikacije. Jednom otkrivena prezentacija na Internetu može biti potpuno drugačija kada je posetimo sledeći put.

Očigledno je da Internet objedinjuje dobre osobine klasičnih medija (štampanih i elektronskih) i dopunjuje ih svojim posebnim karakteristikama. Iako su informacije dostupne u tekstualnom, audio i vizuelnom obliku, glavni deo se i dalje prenosi u obliku teksta, zbog zaostajanja tehnologije za realnim potrebama, tako da se ostali sadržaji uglavnom koriste kao dopuna.

Publika koju pokriva Internet je ceo svet, a prezentacija je neprekidno dostupna. Komunikacija je masovna, brza i jeftina. Mnogobrojni servisi omogućavaju da poruka lako stigne na drugi kraj sveta.

Iz karakteristika Interneta kao medija proizlazi da World Wide Web nosi u sebi i nekoliko **nedostataka**, koji se ogledaju u tome da je veoma teško snaći se u izobilju sajtova nad kojima je nemoguće sprovesti organizaciju i kontrolu kvaliteta.

Servisi Interneta

Servisi Interneta su u stvari usluge koje su korisnicima omogućene upotrebom Interneta i njegovom tehnologijom.

„Svetska mreža“ (**WWW** – *World Wide Web*) ili **Web** predstavlja osnovni servis Interneta, namenjen informisanju. U svom najosnovnijem obliku, tehnički se oslanja na HTTP protokol i HTML (*HyperText Markup Language*) standard. Danas je ovaj servis zastupljen u tolikoj meri da ga prosečan korisnik najčešće poistovećuje sa Internetom. WWW je, u stvari, skup HTML dokumenata koji u sebi mogu sadržati tekst, grafiku, multimediju, pa čak i aplikacije. HTML je standard koji omogućava kreiranje takvih dokumenata i, što je još važnije – njihovo povezivanje putem „linkova“ (hiperlinkovi, hiperveze). Linkovi su posebno označeni delovi dokumenta (reči, pojmovi, slike) koji korisniku omogućuju prelazak na neki drugi dokument. Tekst koji sadrži hiperlinkove naziva se „hipertekst“.

Svakoј stranici, odnosno podatku na Internetu, pristupa se preko njogove jedinstvene **web adrese** (*URI – Universal Resource Identifier*). Ova adresa obično sadrži oznaku protokola koji se koristi za prenos, domen, odnosno sajt na kome se fajl nalazi i, konačno, putanju do samog fajla. Npr:

protokol	domen (sajt)	putanja
http://	www.visokaturisticka.edu.rs	/vazno/ispiti.html

Skup od nekoliko stranica, koje se nalaze u okviru iste adrese, naziva se **web sajt**, odnosno **web prezentacija**. Stranice na prezentaciji su međusobno povezane linkovima, a na isti način mogu se povezivati i različite prezentacije. Obično su stranice prezentacije povezane i sadržajno, kao i izgledom, ali to nije obavezno.

Korisnici pristupaju web-u pomoću posebne aplikacije – **web čitača** (*web browser*). Zbog brzog razvoja web-a i pridruženih standarda, web čitači postaju sve kompleksnije aplikacije – moraju uključiti poštovanje standarda za opis web dokumenta (HTML), standarda za formatiranje izgleda tih dokumenata (CSS), mogućnost izvršavanja bar jednog programskog jezika (JavaScript), mogućnost prikaza nekoliko tipova grafike (gif, jpg, png), video sadržaja (ogv, webm) i zvuka (oga), mogućnost proširenja putem ekstenzija i plug-inova, a sve to uz maksimalnu brzinu, bezbednost i udobnost „surfovanja“ po Internetu. Moderni web čitači omogućavaju izvršavanje čitavih web aplikacija u obliku Internet prezentacije.



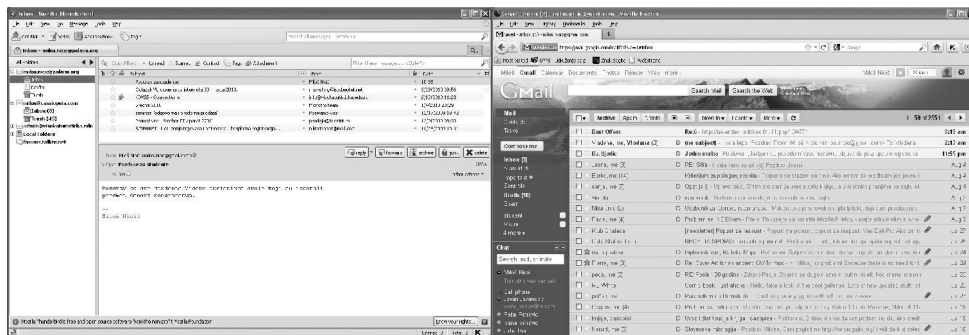
Slika 13.1. Web čitač Firefox 5.0 (Mozilla)

WWW je toliko korišćen servis, da danas svi ostali servisi (osim par izuzetaka) imaju mogućnost korišćenja putem Weba.

On-line kolekcije dokumenata i wiki predstavljaju organizovani način informisanja tipa „sve na jednom mestu“. Tipičan primer kolekcije dokumenata predstavlja on-line enciklopedija ili biblioteka knjiga i časopisa. Ovakvi projekti mogu biti opšti ili posvećeni jednoj temi. Danas su on-line enciklopedije u velikoj meri potisnule ostale vidove enciklopedija.

Wiki sistem predstavlja poseban vid sistema za upravljanje sadržajem sajta (CMS – Content Management System) koji omogućava saradnju na projektu. Tako putem wiki sistema svi koji imaju dozvoljen pristup mogu zajednički doprinositi sadržaju web sajta. Najpoznatiji primer je svakako **Wikipedia** – opšta on-line enciklopedija. Koliko je Wikipedia napredovala govori podatak da je u julu 2011. godine imala izdanja na 268 jezika, a samo na engleskom skoro 3,7 miliona članaka.[37] Cela Wikipedia je u tom trenutku predstavljala oko 30GB podataka.[38] Iako svako može menjati i dodavati tekstove putem wiki-ja, stare verzije se uvek čuvaju tako da je uvek moguće poništiti neku nepoželjnu ili netačnu promenu.

Elektronska pošta (e-mail) predstavlja najosnovniji način komuniciranja putem Interneta. U pitanju je sistem za slanje i primanje tekstualnih poruka i pratećih fajlova. Primalac ne mora biti u tom trenutku prisutan na Internetu da bi primio poruku, pošto se ona najpre skladišti u elektronskom „sandučetu“ na serveru dok ne bude prebačena na računar korisnika.



Slika 13.2. E-mail u aplikaciji Thunderbird 3.1 (Mozilla) i primer web mail sistema Gmail

E-mail je dostupan kao samostalan servis putem specifične aplikacije ili preko web-a, kao web-mail. Ukoliko se koristi samostalna aplikacija, poruke se uobičajeno brišu sa servera kada se

proslede korisniku, dok kod web-mail servisa poruke postoje samo na serveru, a korisnik ih odatle može pročitati pomoću web čitača.

Da bi korisnik poslao/primio e-mail, potrebno je da ima **mail adresu**, koja se razlikuje od web adrese sajta. Ove adrese je lako prepoznati po specifičnom znaku „@“ (majmunsko A), koji se obično izgovara kao „et“ (engl. "at"). Primer mail adrese:

korisničko ime	domen
ime @	visokaturisticka.edu.rs

Pored adrese, korisnik mora imati tajnu šifru i parametre (podešavanja servera i protokola) za pristup ovom servisu. Kada se poruka šalje, postoji nekoliko polja koja se mogu popuniti:

FROM	E-mail adresa pošiljaoca poruke. Ovo polje je obavezno, mada neki programi za slanje e-pošte daju mogućnost unosa lažne adrese, kako bi se omogućile prevare i napadi.
TO	E-mail adresa primaoca poruke. Ovo polje je uvek obavezno.
CC	Ukoliko je potrebno poslati poruku na više adresa, te adrese se mogu navesti u polju CC (<i>Carbon Copy</i>), odvojene zarezom. Već dugo vremena je moguće isto to učiniti i u polju TO. Neki serveri ograničavaju broj adresa na koje je moguće poslati poruku da bi se sprečilo slanje SPAM poruka.
BCC	Polje BCC (<i>Blind Carbon Copy</i>) takođe služi ukoliko želimo da i drugi korisnici dobiju poruku, ali tako da se ne vidi kome je još poruka poslata (TO i CC su uvek vidljiva).
REPLY-TO	Predstavlja mail adresu na koju bi primalac trebalo da odgovori, u slučaju da je ta adresa drugačija od adrese pošiljaoca iz FROM polja.
SUBJECT	Ovo polje predstavlja naziv poruke. Iako nije obavezno, preporučljivo je da se navede.

Svaka e-mail poruka bi trebalo da sadrži tekst i eventualno pridodate fajlove. Veoma bitno je imati u vidu da je većina mail servera podešena tako da ne prihvataju poruke koje prevazilaze određenu veličinu, kao i da „sandučić“ ima određeni kapacitet koji može biti lako popunjen ako se uz poruku šalju i veliki fajlovi.

Korišćenjem **mailing liste** moguće je poslati istu poruku na više adresa. Za ovo se može koristiti običan e-mail klijent ili specijalizovana aplikacija. Jedan od negativnih vidova mailing liste je **SPAM**. U pitanju su neželjene e-mail poruke kojima se reklamiraju različiti proizvodi i usluge.

Diskusione grupe takođe postoje u dva oblika – kao *UseNet news* grupe, koje se prate putem posebne aplikacije (najčešće preko e-mail klijenta), i u web obliku kao **forumi**. One povezuju veći broj korisnika koji javno objavljuju svoje poruke u okviru odgovarajućih tema. Na taj način veći broj korisnika učestvuje u diskusiji.

On-line komunikacija („časkanje“ – **chat**, **video chat** i **VoIP** – *Voice over IP*) predstavlja mogućnost razgovora u realnom vremenu putem Interneta. Za razliku od e-pošte i diskusija, gde se poruka postavi i na odgovor može da čeka od nekoliko minuta do nekoliko dana, kod ovog tipa komunikacije svi korisnici moraju biti u isto vreme povezani na Internet. *Chat* funkcioniše putem tastature i predstavlja najstariji tip on-line komunikacije. Povećanjem brzine Interneta, kao i snage kompjutera koji su mogli da koriste jače algoritme za kompresiju i dekompresiju, postalo je moguće slati audio i video signal i praktično ostvariti mogućnost „telefoniranja“ putem Interneta. Ovi servisi funkcionišu kako za komunikaciju dva korisnika, tako i za istovremenu komunikaciju više njih.

Blogovi i društvene mreže predstavljaju najnoviji trend u Internet komunikaciji. Blog je vrsta on-line dnevnika (*web-log*) gde korisnik može pisati i objavljivati tekstove o ličnim ili drugim temama. Na svaki tekst drugi korisnici mogu ostavljati komentare. Društvene mreže (*Social*

Networks) predstavljaju način povezivanja korisnika i deljenja tekstova, utisaka i sadržaja. Trenutno najraširenije društvene mreže su *MySpace*, *Facebook* i *Twitter*. Blogovi i društvene mreže su jedan od načina na koji firme mogu obavestavati i komunicirati sa svojim klijentima o novostima.

Video i audio striming se logički nadovezuju na prethodna dva tipa servisa. Tehnološki razvoj omogućio je da se na Internetu pojavljuju mesta na kojima je moguće postavljati i gledati/slušati video/audio zapise. Reč „striming“ (*streaming*) označava da nije potrebno da ceo snimak stigne na računar korisnika, već da se materijal gleda onako kako pristize – u realnom vremenu. Pogledani materijal ne ostaje snimljen na korisnikovom računaru. Osnovna upotreba ovih servisa je slušanje radija i gledanje TV prenosa preko Interneta. Drugi tip servisa omogućava korisnicima da postavljaju, gledaju i komentarišu audio/video zapise, čime uvode i elemente društvenih mreža.

Pretraga (*search engines*) i **portali** su sajтови posvećeni pronalazenju sadržaja na Internetu. Sajтови za pretragu funkcionišu po principu unosa ključnih reči i eventualnog podešavanja ostalih opcija. Rezultati koji se dobijaju ređaju se prema relevantnosti. Portal predstavlja uređen web sajt na kome se može pronaći veliki broj kvalitetnih linkova raspoređenih po oblastima i podoblastima. Dobra strana portala je ta što su svi sajтови, koji se mogu naći preko njega, pregledani od strane urednika i samim tim je garantovan neki kvalitet. Sa druge strane, sajтови za pretragu pružaju mnogo veću slobodu prilikom surfovanja Internetom.

Fajl servisi u nekom obliku postoje od kada postoji i Internet. Osnovni servis za prenos fajlova nosi ime kao i već pomenuti protokol – FTP. Kod njega se putem korisničkog imena i šifre pristupa serveru sa koga je moguće učitati, snimiti ili obrisati fajlove. Naprednija i komercijalnija verzija su moderni servisi za razmenu (*share-ovanje*) fajlova, od kojih su najpoznatiji *Rapidshare*, *Filesonic*, *Megaupload*, *Hotfile* itd. Ovi servisi ne omogućavaju pretragu fajlova – pristup je omogućen samo ako dobijemo tačan link do fajla. Linkovi se obično razmenjuju putem chat-a ili na posebnim forumima. Konačno, fajlovi se mogu razmenjivati i putem *BitTorrent*, *eDonkey* i sličnih protokola koji su razvijeni radi prenosa velike količine podataka, tako da se serveri ne opterećuju. Ideja je da se korisnici povezuju međusobno preko P2P mreže – kada se fajl preuzima, ne preuzima se samo sa jednog mesta, već u delovima od velikog broja korisnika koji ga imaju (istovremeno i naš računar služi da prosleđuje delove fajla svim drugim korisnicima).

E-trgovina, on-line aukcije i on-line agenti su servisi Interneta koji su najuže povezani sa elektronskim poslovanjem na Internetu. E-trgovina se odnosi na prodaju dobara putem Interneta na posebnim sajtovima. Sa druge strane, aukcijski sajтови omogućavaju povezivanje korisnika koji nude određenu robu, za koju drugi korisnici mogu davati ponude. Na sličan način funkcionišu i razni on-line agenti. U pitanju je sajt koji povezuje korisnike koji potražuju i korisnike koji pružaju određene usluge. Primer bi bio sajt *freelancer.com* koji povezuje poslodavce i one koji traže honorarni posao.

Web aplikacije i on-line igre su relativno noviji servis Interneta, nastao sa razvojem web-a i njegovih standarda. Povećanjem mogućnosti i brzine interpretacije JavaScript-a, uvođenjem novih elemenata u HTML standard, kao i povećanjem mogućnosti CSS-a, otvorilo se čitavo područje nove interaktivne upotrebe web-a. Sada je moguće kreirati čitave aplikacije koje funkcionišu unutar web čitača. Ovakav on-line servis je praktično predodređen za saradnju više korisnika na istom projektu.



Slika 13.3. Izgled dokumenta kreiranog u web aplikaciji **Google Docs** (Google) i on-line igra **Angry Birds** (Rovio Mobile)

Netiketa

Netiketa (*netiquette* = *network* + *etiquette*) predstavlja pravila lepog ponašanja na Internetu.

Osnovno pravilo lepog ponašanja je – **jasnoća izražavanja**. Na Internetu se ne vidi gestikulacija ili izraz lica, ne čuje se glas – ono što je napisano vrlo lako može biti pogrešno shvaćeno. Na različitim forumima i razgovorima funkcionišu različita pravila – od izuzetnog značaja je da se pre započinjanja komunikacije ta pravila prouče i kasnije poštuju. Takođe, neophodno je **poštovanje sagovornika** i njegovih stavova. Jedno od generalnih pravila je – „Budite veoma strogi u vezi onoga što pišete, a veoma tolerantni po pitanju onoga što drugi pišu“.[39]

Prilikom komunikacije putem e-maila, lepim ponašanjem se smatra ako se odgovor da u roku od 24 sata od prijema poruke. Takođe, od korisnika se očekuje da se ispravno predstavi svojom pravom mail adresom. Ukoliko se poruka šalje na više adresa, utoliko se smatra nekulturnim smeštanje svih adresa u TO ili CC polje – potrebno je poštovati tuđu privatnost i onemogućiti da svaki primalac dobije i adrese svih ostalih. Uvek je potrebno navesti naziv poruke (*subject*) koji na adekvatan način opisuje sadržaj poruke.

Prilikom komunikacije, podrazumeva se da korišćenje **VELIKIH SLOVA** označava vikanje.

Na diskusionim grupama i drugim mestima, na kojima komunicira veći broj korisnika, česta je pojava tzv. „**trolova**“. U pitanju su osobe koje ometaju razgovor neprikladnim, provokativnim i uvredljivim porukama. Na uređenim servisima (forumi, blogovi) protiv trolova se bore urednici (moderatori i administratori) – najpre upozorenjima, a kasnije i privremenim ili stalnim zabranama pristupa (tzv. *ban*). Na neuređenim servisima (*news* grupe) korisnici moraju sami da se izbore, obično koristeći opcije poput ignorisanja poruka određenog korisnika (*ignore*) u svojoj aplikaciji.

Još jedna nepoželjna pojava na servisima javne komunikacije je „**prepucavanje**“ (**flame**). U pitanju su poruke u kojima se korisnici na uvredljiv ili ponižavajući način obraćaju jedan drugome, umesto civilizovanog načina kojim se izražava neslaganje sa nečijim stavovima i uverenjima. Eskalacija ovakvog sukoba među korisnicima naziva se *flame war*. Po izbijanju sukoba, retko kada diskusija može da se vrati u normalan tok, pa urednici obično rešavaju ovakve slučajeve zaključavanjem teme, a neretko i zabranom pristupa.

Usled opšte globalizacije Interneta, dolazi do mešanja ljudi različitih kultura, navika i nivoa obrazovanja. Neke poruke lako mogu biti pogrešno shvaćene, pa se zbog toga na Internetu

ustalilo korišćenje tzv. **emotikona** (*emoticon* = *emotion* + *icon*). Emotikon predstavlja tekstualno-grafički simbol nekog raspoloženja i reprezentuje se kroz različite forme tzv. „smajli“ simbola. Ideja je da se kroz komunikaciju, koja je na početku bila čisto tekstualna, predstavi grafički prikaz simbola nasmejanog lica koji datira još iz šezdesetih godina XX veka. Za klasičnog „smajlija“ koristi se dvotačka, povlaka i zatvorena zagrada (mada se povlaka može i eliminisati).

Kada se ovakav tekst pogleda pod uglom od 90 stepeni, podseća na nasmejano lice. Neki od najkorišćenijih su:

: -) :))) : D	dobro raspoloženje – ako se doda više zagrada označava jače smejanje, široki osmeh – glasno smejanje
; -) ;)	„zez“ – smajli koji namiguje – objašnjava da tekst ne treba ozbiljno shvatiti
: - (: ' (> :- [„tužan“ smajli, smajli sa suzom i „namršteni“ smajli
: - * B -)	„poljubac“ smajli i „cool“ smajli sa tamnim naočarima
%- : - P : - \	„ošamućeni“, „isplažen jezik“ i „oklevajući“ smajli

Prilikom komunikacije korisnici često zaboravljaju još jedno pravilo – smajli odražava nečije raspoloženje, ne čarobni štapić kojim se poništava ono što je napisano – uvrede na nečiji račun se ne mogu pretvoriti u šalu prostim stavljanjem smajlija. Već godinama se u Internet komunikaciji koriste grafički smajliji, odnosno prave sličice, ali tekstualni emotikoni su i dalje rasprostranjeni širom Mreže.

Osim emotikona, u on-line žargonu se koriste i skraćenice koje imaju, nekad više – nekad manje, poznato značenje. Ove skraćenice i kriptične reči proizlaze iz slenga poznatog kao **I337** (*leet* – iskvareni način pisanja reči *elite*).[40] To mogu biti različite vulgarnе skraćenice ili termini koji nastaju menjanjem pojedinih slova ciframa ili simbolima. Ironično, ali hakeri danas jako retko koriste ovakav način pisanja. Neke od skraćenica koje su se održale u široj upotrebi:

LOL ,	"Laughing Out Loud" – glasno se smejem
ROFL	"Rolling On Floor Laughing" – veoma burno smejanje
OMG	"Oh My God" – za ime Boga
BTW	"By The Way" – uzgred
pwn	iskvaren način pisanja reči "own" što označava pobjedu/dominaciju nad nekim
RTFM	"Read The Fine Manual" – komentar na pitanja za koja se odgovor može lako pronaći
IMHO	"In My Humble Opinion" – po mom skromnom mišljenju
n00b	iskvaren način pisanja reči "newbie" – početnik, neznalica (obično u pogrđnom smislu)
MorF	"male or female" – uobičajeno pitanje prilikom započinjanja on-line konverzacije
CU 18er	"see you later" – vidimo se kasnije – jedan od pozdrava na kraju chat-a
10x	"thanks" – hvala

Kada se govori o bezbednosti informacionog sistema, najčešće se misli na odbranu od virusa i hakerskih napada. Istina je da je sigurnost sistema u osnovi težnja da sistem neometano obavlja svoju funkciju.

Gledajući na taj način, prvi nivo zaštite će uvek biti **fizička bezbednost**. Ukoliko sistem radi u lošem okruženju, nisu potrebni hakeri da bi se sistem srušio – dovoljna je neispravna oprema ili nestanak struje. Drugim rečima, informacioni sistem mora raditi na adekvatnom hardveru koji može da „podnese“ sve zahteve sistema i koji je adekvatno zaštićen od spoljnih uticaja. To znači da veliki, opterećeni sistem sa mnogo korisnika mora biti zasnovan na dovoljno snažnim računarima, sa dovoljno brzom vezom kako bi funkcionisao. Isto tako, što je sistem važniji, mora biti bolje zaštićen od fluktuacija napona i nestanka struje, prokišnjavanja i povećane vlažnosti, naglih promena temperature ili pregrevanja.

Osim toga, hakerima je, u određenom broju slučajeva, potreban fizički pristup serverima ili određenim segmentima mreže. Bitno je da ovi kritični elementi sistema budu zaštićeni od provale.[41]

Maliciozni softver

Maliciozni softver (*malware* – *malicious software*) su programi koji rade nešto zlonamerno na računaru. Obično ih karakterišu prikrivenost i razmnožavanje.

Prikrivenost

Osnovna karakteristika zlonamernog softvera je njihova sposobnost da zaraze računar bez znanja korisnika i da ih je kasnije teško pronaći i odstraniti. Osnovne dve strategije su:

- **sakrivanje**, kada se koristi tehnička veština da se program učini „nevidljivim“ i
- **prevara**, kada se program predstavlja kao nešto sasvim drugo.

Jednom instaliran maliciozni softver se može kriti u početnom (**boot**) sektoru hard diska, pri čemu se izvišava svaki put kada se računar uključi, **unutar nekog drugog fajla** (programa ili dokumenta) ili kao **zaseban fajl**, obično imitirajući neki od sistemskih fajlova na mestima teško pristupačnim za običnog korisnika.

Razmnožavanje

Samoreplikacija je karakteristika po kojoj je cela klasa virusa i dobila ime, budući da je u pitanju osobina koja imitira živ organizam.

Virusi (*viruses*) se samoreprodukuju ne samo sa računara na računar, već i na samom računaru, tako što „šire zarazu“ među fajlovima (programi i dokumenti) na hard disku. Što više vremena provedu na nekom računaru, to će više fajlova biti zaraženo. Pokretanje bilo kog od zaraženih fajlova na drugom računaru preneće zarazu i na njega.

Crvi (*worms*) su zasebni programi koji se agresivno šire sa računara na računar. Najčešće se za prenos koristi mreža (lokalna ili Internet), ali i mediji spoljne memorije (diskete, *flash* memorija). Za razliku od virusa, ne šire se unutar samog računara.

Trojanski konji (*trojans*) su programi koji se predstavljaju kao sasvim legitiman i koristan softver i na taj način prevare korisnika da ih pokrene i instalira na računaru. Najčešći izvor trojanaca predstavlja piratski softver preuzet sa Interneta. Korisnici u potrazi za programima, igrama, muzikom i filmovima, preko različitih servisa, dolaze do nelegalnih sadržaja, a veliki deo takvih programa, osim što im je razbijena zaštita, u sebi sadrže i trojance.[41]

Instaleri (*droppers*) predstavljaju prilično opasnu kategoriju zlonamernog softvera. Oni sami ne predstavljaju virus, već ga instaliraju na računar. Sam virus u okviru *dropper*-a može biti kompresovan ili šifrovan na različite načine, kako bi se izbegla detekcija od strane anti-virusa.

Način delovanja malicioznih programa

Postoji mnogo načina na koje se delovanje malicioznog softvera može manifestovati. Ovi programi se, pre svega, mogu podeliti prema trenutku aktiviranja – najčešće su aktivni od samog početka, ali mogu biti i „uspavani“ do određenog vremenskog trenutka (*time bomb*) ili dok korisnik ne izvrši neku određenu akciju na računaru (*logic bomb*).

Sama aktivnost zlonamernog programa može varirati od potpuno benignog (prikaz poruke) do veoma destruktivnog (brisanje podataka, fizički kvar računara). Jedna od najpopularnijih aktivnosti je kreiranje **tajnog ulaza** (*backdoor*), preko koga hakeri imaju pristup zaraženom računaru. Preko ovakvog sistema hakeri mogu preuzimati privatne podatke, fajlove, posmatrati vaš ekran ili pratiti vaše aktivnosti putem špijunskog softvera (*spyware*), beležiti sve što se otkuca na računaru ili iskoristiti zaraženi računar za slanje SPAM poruka i hakerske napade, čime računar ulazi u **BotNet** mrežu, odnosno postaje tzv. „zombi“ (*zombie*) računar – i sve to bez znanja korisnika.

Korisnici koji se na Internet i dalje povezuju putem telefonskog modema na meti su posebne vrste zlonamernog softvera, čija je funkcija da prekinu konekciju sa Internetom, a onda kreiraju novu konekciju koja poziva inostrani telefonski broj (*dialer*). Korisnik koji ne obrati pažnju na proces ponovnog povezivanja na Internet može platiti ogroman telefonski račun.

Virusi se danas obično kreiraju tako da ih nije jednostavno ukloniti iz memorije računara. Mogu se sastojati iz više komponenata koje „štite“ jedna drugu. Dešava se da je u nekim slučajevima nemoguće „ubiti proces“ – izbaciti virus iz RAM memorije, pošto se stalno vraća ili obrisati zaraženi fajl sa diska. Ovi metodi samoodbrane idu dotle da virus može izbaciti čak i neki slabiji antivirusni program. Ponekad je jedini način oporavka računara reinstalacija operativnog sistema, ali „od nule“ – uz obavezno prethodno formatiranje diska.

Hakerski napadi

Hakerski napad predstavlja cilj, odnosno manifestaciju hakerskog delovanja. Motivacija hakerskih aktivnosti je prilično sivo područje, ali može se sa sigurnošću reći da samo manji broj to čini zarad učenja ili ideala, dok većinu pokreće dokazivanje, osećaj moći, lažni osećaj pripadnosti nekoj intelektualnoj eliti ili, prosto, materijalna dobit.

Preuzimanje administratorskih privilegija je najviši cilj hakera – da potpuno ovlada nekim sistemom. U tom slučaju može iščitati sve podatke koji se nalaze u sistemu, menjati ih ili obrisati. Takav računar može koristiti za svoje potrebe – napade na druge računare, širenje virusa, distribuciju nelegalnog sadržaja i slično. Sticanjem visokih privilegija u sistemu, npr. banke, haker može prebaciti novac na svoj račun ili odobriti kredit. Koliko ova aktivnost može biti opasna ilustruje mogućnost upada u sistem bolnice, čime se mogu ugroziti pacijenti (npr. promena istorije bolesti) ili u sistem kontrole saobraćaja, čime se može izazvati automobilska nesreća (npr. promena svetala na semaforu).

Odbijanje usluge (**DoS** – *Denial of Service* i **DDoS** – *Distributed Denial of Service*) predstavlja vrstu napada kojom se server zaguši zahtevima kako obični korisnici ne bi mogli da mu pristupe. Distribuirani DoS napad predstavlja višestruko uvećani DoS, pošto u napadu učestvuje veliki broj računara.[45] Varijanta DoS napada je i bombardovanje e-maila (**mail bombing**), koji predstavlja vrstu napada kojom se određeni korisnik ili mail server zatrpavaju beskorisnim e-mail porukama, koje mogu onesposobiti sistem. Odbrana od ovakvog tipa napada je veoma teška, pošto, u suštini, server dobija sasvim legitimne zahteve, na koje u normalnim okolnostima i treba da odgovori.

Promena izgleda (**defacement**) nekog web sajta (najčešće naslovne strane) obično ima za cilj ostavljanje poruke ili logoa hakerske grupe. Ovakvi napadi deluju spektakularno običnim korisnicima, ali prava šteta je minimalna.

Krađa identiteta je kriminalna aktivnost koja vuče korene još iz vremena daleko pre pojave računara (falsifikovanje potpisa, korišćenje tuđeg pečata), a nastavlja se i danas (korišćenje ukradenih dokumenata, kreditnih kartica, mobilnog telefona). Hakeri su jednostavno preneli ovu aktivnost u digitalni svet, koristeći tehnologiju da prikupe privatne podatke o nekoj osobi, šifre i brojeve kreditnih kartica. Haker tada, lažno se predstavljajući kao ta osoba, koristi njegove servise, komunicira ili plaća robu i usluge.[42]

Hakeri i hakerska etika

Termin „haker“ (*hacker*) potiče od studenata MIT-a iz 60-ih godina XX veka. Označava, blago rečeno, „kompjuterskog entuzijastu“, osobu koja provodi veoma mnogo vremena za računarom, programirajući ili pokušavajući da nauči što više o funkcionisanju sistema. Pronalaženje slabosti, grešaka i zaobilaznje pravila predstavljaju intelektualni izazov. Hakeri se rukovode sopstvenom etikom koja zagovara slobodu informacija i tehnologija. Istinski haker oseća prezir prema državnoj centralizaciji, kontroli i cenzuri, delovanju velikih profitnih korporacija, a pre svega prema neznanju korisnika računara. Suštinski, haking predstavlja „intelektualnu gimnastiku“, razmišljanje van ustaljenih okvira, vid zabave, a ne način zarade ili nanošenje štete. Postoji veliki broj primera u kojima su hakeri na kraju pomogli ili bar pokušali da pomognu administratorima da poboljšaju svoj sistem.

Međutim, termin „haker“ obično ima negativnu konotaciju. Nažalost, istina je da danas ima mnogo više destruktivnih osoba koje tehnologiju koriste kako bi ostvarili profit ili naneli štetu drugima. Postoji inicijativa da se ovakvi zlonamerni hakeri okarakterišu kao „krakeri“. Razlika je u motivaciji – pravim hakerima je cilj savlađivanje izazova – otkriti način da se probije zaštita ili pisanje programa koji probija zaštitu, dok krakeri žele samo da probiju zaštitu, na bilo koji način.

U žargonu se etički hakeri i administratori koji štite sistem nazivaju „belim šeširima“, dok se krakeri nazivaju „crnim šeširima“. Analogija potiče od starih vestern filmova u kojima su „dobri kauboji“ uvek imali bele šešire, a negativci crne.

Danas se slika o hakeru kao usamljenom sajber-jahaču ili članu robinhudovske grupe iz osnova menja, budući da sada i same korporacije zapošljavaju hakere kako bi špijunirali/sabotirali konkurenciju, prikupljali privatne podatke o klijentima, krali e-mail adrese na koje kasnije šalju SPAM poruke. U vreme kada preduzeća sve više koriste e-poslovanje kako bi izborili konkurentsku prednost, informacije o potrošačima, njihovom privatnom životu i navikama postaju veoma cenjen resurs i predmet kupovine i prodaje.

Script Kiddies su verovatno najgora vrsta krakera. U hakerskim krugovima ih smatraju *lamerima* (neznalica). U pitanju su osobe koje nemaju nikakvo znanje o funkcionisanju mreža, operativnih sistema ili programiranju. Koriste gotova rešenja – već napravljene programe i skriptove. Mogu naneti veliku štetu, suštinski i ne razumejući šta su zaista uradili.

Hakerske tehnike

„Rupe“ (**exploits**) su greške u aplikacijama, informacionim ili operativnim sistemima. U pitanju su slabosti ovih programa u slučaju nepredviđenih situacija. Hakeri ih retko objavljuju javno – radije ih razmenjuju međusobno. Kada slabost postane poznata, samo je pitanje vremena kada će biti ispravljena novom „zakrpom“. Samim tim, *exploit*-ima pada vrednost sa proticanjem vremena, a najcenjeniji su tzv. **zero-day exploits**, koji su još uvek nepoznati proizvođaču softvera.

Posebna vrsta exploita su **generičke šifre**. Za određene mrežne uređaje, BIOS računara ili čak operativne sisteme, postoje tajne šifre koje omogućavaju kompletan administratorski pristup. Ove šifre se koriste interno radi servisiranja, ali kada postanu poznate predstavljaju ozbiljnu pretnju sigurnosti sistema.

Osnovni tip napada preko mreže predstavlja skeniranje opštepoznatih slabosti (**vulnerability scanning**), odnosno pokušaj upada isprobavanjem dobro poznatih „rupa“, u nadi da administrator nije adekvatno zaštitio sistem. Ova tehnika uključuje pronalaženje otvorenih portova za komunikaciju i pokušaj da se makar saznaju podaci o operativnom sistemu i serverskom softveru (*banner identification*).

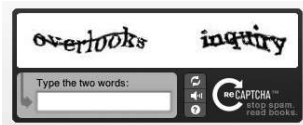
Rootkit predstavlja softver koji omogućava hakeru da koristi administratorske privilegije sistema. Glavni problem je instalacija tog softvera na ciljnom računaru, ali kada se to postigne, *rootkit* otvara *backdoor* koji haker može koristiti. Ovaj softver je tako kreiran da bude u najvećoj meri „nevidljiv“ za administratora – ne manifestuje se u listi pokrenutih procesa, uklanja svoje tragove iz log fajlova, i sl.

Napadi korišćenjem „grube sile“ ili rečnika predstavljaju pokušaj otkrivanja šifre korisnika isprobavanjem svih mogućih kombinacija (**brute force**), odnosno korišćenjem rečnika pojmova (**dictionary**) koji bi mogli biti upotrebljeni za šifru. Zbog ovih tipova napada preporučuju se duže šifre koje ne predstavljaju puke pojmove već kombinacije slova, brojeva i drugih znakova. Sa svakim novim znakom u šifri, broj kombinacija koje moraju biti isprobane u *brute force* napadu se eksponencijalno povećava, a dodavanje cifara i drugih znakova funkcioniše kao odbrana od *dictionary* napada.

Međutim, treba imati u vidu da se brzina računara stalno povećava, a ne treba isključiti ni mogućnost napada od strane čitave mreže računara, koji čak i *brute force* napade čine efikasnim.

Budući da se za ove napade koriste posebni programi koji automatski pokušavaju da pogode šifru, sajtovi uvode dodatne mere zaštite kako bi proverili da li se „sa druge strane“ zaista nalazi ljudsko biće. Jedan od metoda je **CAPTCHA** sistem (*Completely Automated Public Turing test*

to tell Computers and Humans Apart), gde se na ekranu prikazuje slika teksta koja je tako deformisana da ostaje (relativno) čitljiva ljudima, ali ne i računarima.[46]



Slika 14.1. Primer CAPTCHA teksta (www.captcha.net)

Lažno predstavljanje (**spoofing**) podrazumeva tehniku koja se u velikoj meri koristi na Internetu. Svaki paket koji se prenosi putem Interneta nosi informaciju o IP adresi računara kome je upućen i računara sa koga je poslat. Ukoliko se prilikom izlaska paketa na mrežu on promeni, tako da se u njega unese lažna IP adresa polaznog računara, napadnuti računar ne može otkriti odakle napad dolazi. Ova tehnika se koristi najčešće za DOS i DDOS napade, budući da je njihov cilj zagušivanje napadnutog računara – nema potrebe da se dobija bilo kakav odgovor, koji bi ionako bio upućen na lažnu adresu.

„Pecanje“ (**phishing**) predstavlja varijantu lažnog predstavljanja pomoću koje hakeri dolaze do korisničkih imena, šifara ili brojeva kreditnih kartica. Hakeri u ovom slučaju kreiraju čitav web sajt koji izgleda isto kao sajt npr. banke i „službenim“ e-mailom korisnicima šalju zahtev da se zbog nekog razloga prijave putem datog linka, koji u stvari vodi na hakerski sajt. Najvažniji zadatak za hakera je da od korisnika prikrrije da se radi o lažnom web sajtu (koriste slične URL adrese, bezbednosne propuste originalnog sajta ili, ako imaju pristup korisnikovom računaru, menjaju *hosts* datoteku koja vrši preusmeravanje URL adrese na IP adresu hakerskog sajta). Jednostavnija varijacija ovog tipa prevare je slanje zahteva da korisnik pošalje svoju šifru mailom, pošto je jedan od najjednostavnijih *spoofinga* menjanje/prikrivanje e-mail adrese sa koje je poruka poslata.

„Njuškanje“ (**sniffing**) je, suštinski, tehnika koju koriste administratori sistema da otkriju probleme u mreži, a koju hakeri mogu zloupotrebiti. Kao što je već pomenuto, svaki paket koji se prenosi preko mreže sadrži adresu računara sa koga se šalje i adresu računara kome je namenjen. Međutim, paketi na svom putu kroz mrežu prolaze kroz različite uređaje i putuju preko više računara i čvornih tačaka. Uobičajeno ponašanje računara preko koga se odvija saobraćaj jeste da proveriti određenu adresu paketa i, ako paket nije namenjen tom računaru, da ga prosledi dalje.

Putem softvera za analizu saobraćaja mrežni interfejs računara se prebacuje u tzv. „promiskuitetni“ režim, kako bi računar takođe dobijao i svaki paket koji se prosleđuje dalje. Ovi paketi mogu biti pročitani i iz njih se mogu dobiti podaci o brojevima kreditnih kartica, šiframa i slično. Ovakvi napadi predstavljaju najčešći tip napada na bežičnim mrežama.

SQL injekcija (**SQL injection**) je tehnika napada na SQL bazu podataka na Internetu. Svaki iole kompleksniji web sajt ima u pozadini bazu podataka koja može varirati od najjednostavnije, u kojoj se beleže podaci o klijentima, pa do baze koja podržava čitav informacioni sistem preduzeća. SQL injekcija predstavlja dodavanje SQL komandi unutar korisničkog imena ili šifre, pod pretpostavkom da se ti podaci provlače kroz bazu podataka kako bi se dozvolio ili odbio pristup. Ukoliko se ova komanda zada na ispravan način, utoliko može omogućiti hakeru da pročita sva korisnička imena i šifre ili da obriše delove baze podataka ili čak celu bazu.

Manipulacija službenika (**social engineering**) je takođe tehnika prevare. Nema veze sa računarima već sa iskorišćavanjem ljudske lakovernosti. Hakeri koriste lažno predstavljanje, strah od gubitka posla i tehničku neobrazovanost zaposlenih da saznaju dovoljno podataka (IP

adrese računara, interne brojeve telefona, šifre i korisnička imena) kako bi upali u sistem. Da bi prevara bila uspešna, hakeri moraju u najvećoj meri da se upoznaju sa organizacijom, hijerarhijom i poslovanjem firme koju napadaju. Da bi ovo postigli koriste javne publikacije, standarde, časopise, pa čak i preturanje po smeću (*garbage scanning*), što može biti veoma korisno, pogotovu ako firma ne uništava poverljive dokumente koje odbacuje. Zbog velike količine informacija koje hakeri prikupljaju o svojim metama, nikako se ne preporučuju šifre koje su nekako vezane za privatni život korisnika (imena članova porodice i kućnih ljubimaca, nadimci, datumi rođenja, idoli pop kulture i sl).

Ponekad je samo vešto sročeni e-mail dovoljan da korisnik pokrene program u prilogu poruke i instalira virus na svom računaru.

Snimanje tastature (**key logging**) je tehnika krađe šifara i podataka na samom „izvoru“, odnosno beleženjem svega što se otkuca na tastaturi korisnika. *Keylogger* programi su (logično) nevidljivi za korisnika, a funkcionišu tako što presreću signal sa tastature i svaki pritisnuti taster beleže u poseban fajl. Signal se onda propušta dalje kroz sistem, tako da korisnik ni ne primećuje da se nešto dešava. Snimljeni fajl se povremeno šalje putem Interneta. Da bi se ovo postiglo, potrebno je da na računaru korisnika postoji instaliran ovakav program, što znači da računar već mora biti zaražen ili da haker ima fizički pristup računaru.

Podvala (**hoax**) je još jedan vid prevare koji se obavlja putem Interneta. Obično su u pitanju različite igre na sreću i nuđenje lake zarade. Jedan od uobičajenih i dobro poznatih načina je ponuda advokata iz neke zemlje trećeg sveta, koji nas obaveštava o slučaju smrti izvesne bogate osobe bez naslednika. Advokat nam nudi saučesništvo u prevari – nas bi predstavio kao naslednika kako bi on prigrabio najveći iznos, dok bismo mi za tu uslugu dobili određeni manji procenat novca koji bi i takav predstavljao pozamašan iznos. Ako osoba pristane na ovu prevaru, od nje se traži uplata neke manje sume novca kako bi navodni advokat pripremio dokumentaciju. Ova uplata predstavlja svrhu prevare.

Drugi tip prevare nema za cilj zaradu, već širenje e-mail poruka putem Interneta. Može se reći da su ove neželjene (**spam**) poruke same po sebi virusi, pošto opterećuju servere i usporavaju saobraćaj. Tipične poruke su razne vrste lanaca sreće (**chain letter**). Drugi tip predstavljaju „dobronamerne“ poruke koje nas upozoravaju na neki novi virus koji se širi putem e-maila i koji će nam zaraziti računar ukoliko ga otvorimo. Još jedna varijanta je poruka o tome kako *Microsoft* ili neka druga poznata informatička firma prati putanju te poruke i isplaćuje korisnicima određeni iznos novca na osnovu besplatne reklame.

Zaštita

Osnovna fizička zaštita podrazumeva pronalaženje suve i klimatizovane prostorije za smeštanje servera. Poželjno je nabaviti rezervno napajanje (*UPS – Uninterruptable Power Supply*), uređaj ili čak agregat. Odbrana od hakerskih napada u lokalnoj mreži podrazumeva fizičku zaštitu segmenata mreže preko kojih se odvija saobraćaj i čvorova na kojima bi haker mogao da instalira svoj uređaj, odnosno softver.

Šifre za pristup sistemu ne bi smele da budu suviše kratke, da budu vezane za korisnika i trebalo bi da, osim slova, uključuju i cifre i specijalne znakove. Naravno, podrazumeva se da se šifre čuvaju u tajnosti.

Konačno, preporučuje se stalna provera sopstvenog sistema, pregled web sajta u potrazi za nepoznatim fajlovima ili dodatim tekstom na web stranicama. Svaki sistem beleži kritične

aktivnosti u jednom ili više log fajlova, čijim pregledom se može utvrditi šta se događalo na računaru i da li je došlo do pokušaja nelegalnog pristupa.

Rezervne kopije i ažuriranje

Redovno pravljenje rezervnih kopija (*backup*), odnosno čuvanje podataka je u stvari osnova bezbednosti. Svaka hardverska komponenta je zamenljiva, aplikacije je uvek moguće ponovo instalirati, međutim, izgubljeni podaci su nenadoknadivi.

Ako dođe do proboja sigurnosti, brisanja ili menjanja podataka, ili prosto do kvara zbog koga se podaci izgube, posle oporavka samog sistema, podaci mogu jednostavno da se vrate u poslednje sačuvano stanje.

Sačuvani podaci se ne beleže na računarima na kojima se inače nalaze, već na fizički drugom mediju (npr. flash memorija, CD, DVD i sl).

Druga fundamentalno važna stvar za bezbednost i stabilnost sistema jeste stalno ažuriranje (*update*). Instaliranje „zakrpa“ je bitno pošto se njima uklanjaju greške u programima i sigurnosni propusti. Jedna od najvažnijih stavki kod antivirusnog softvera je svakodnevno ažuriranje baze podataka virusa, pošto se za neke sisteme, kao što je npr. Windows, učestalo pojavljuju novi virusi.

Enkripcija i autentifikacija

Ako su i korisnikov računar i server obezbeđeni, hakeri vrše napad na kanal veze između njih. Bilo koja tačka na liniji kojom se vrši komunikacija može biti izložena napadu. Najbolji načini borbe protiv ovog tipa napada su autentifikacija i enkripcija.

Autentifikacija (*authentication*) predstavlja identifikaciju osobe ili računara na mreži. Za tako nešto se koriste sistemi korisničkih imena i šifara, digitalnih potpisa i digitalnih sertifikata.

Digitalne sertifikate izdaju posebne organizacije (*certificate authority – CA*) koje sertifikatom garantuju identitet web sajta ili pojedinca. Digitalni sertifikati i potpisi takođe mogu biti ukradeni ili iskopirani. Budući da se zasnivaju na sistemu javnog i tajnog ključa, korisnik je jedini odgovoran za bezbednost tajnog ključa koji se nalazi na njegovom računaru. Poznati su i slučajevi korišćenja zastarelih ili ukradenih digitalnih sertifikata kako bi se virusi predstavili kao legitiman softver.[43]

Enkripcija (*encryption*) predstavlja šifrovanje podataka tako da se oni ne mogu rastumačiti bez **ključa** (tajne šifre). Kroz istoriju su se razvijali različiti metodi šifrovanja – zamena slova i pojmova, menjanje redosleda slova i drugi. Računari su omogućili automatizaciju procesa šifrovanja, ali i omogućili lakše dešifrovanje. Pošto je svaki podatak na računaru suštinski broj, proces šifrovanja se može posmatrati kao reverzibilna matematička transformacija, što znači da se iz transformisanih podataka mogu povratiti originalni podaci.

Ukoliko se i za šifrovanje i dešifrovanje koristi isti ključ, govorimo o tzv. **simetričnoj enkripciji**. Uspešnost bilo kog algoritma zavisi od veličine ključa koji se koristi – što je veći ključ, to više kombinacija napadač mora da isproba kako bi razbio šifru. Danas se uglavnom koriste algoritmi koji šifruju blokove podataka (*block cyphers*), a neki od najpoznatijih su DES i TripleDES, AES, Blowfish, RC5, RC6 i sl.[44]

Glavni problem sa ovim tipom enkripcije je dostavljanje šifre drugom korisniku. Ne zaboravite – kada se koristi enkripcija uvek se pretpostavlja da je kanal veze kompromitovan.

Rešenje ovog problema nađeno je u **asimetričnoj enkripciji**, kod koje se ne može koristiti isti ključ za šifrovanje i dešifrovanje. Ova enkripcija funkcioniše po principu **dva ključa – javnog i tajnog**. Na primer, ukoliko osoba A želi da pošalje poruku osobi B, poruku šifrjuje javnim ključem osobe B koji je svima dostupan. Međutim, poruku može da dešifruje samo osoba B koristeći svoj tajni ključ.

Jedan od najpoznatijih asimetričnih algoritama je **RSA algoritam**, kod koga se javni i tajni ključ izračunavaju na bazi dva slučajno izabrana velika prosta broja. Iako pružaju veoma visok stepen bezbednosti, problem sa ovim algoritmima je što su računski veoma zahtevni i nisu adekvatni za prenos velike količine podataka. Obično se bezbedna komunikacija na Internetu odvija tako što se putem asimetrične enkripcije pošalje šifra za simetričnu enkripciju preko koje se onda odvija prenos podataka.[43]

Poznati algoritmi za asimetričnu enkripciju su još i DSA (koristi se za digitalne potpise) i algoritmi eliptične krivulje (*elliptic curve*).

Steganografija predstavlja poseban način postizanja bezbedne komunikacije, koji se ne oslanja na šifrovanje već na prikrivanje poruke. Jedan od najpoznatijih načina prosleđivanja poruke je inkorporiranje podataka (npr. teksta) unutar slike, zvuka ili videa. Kao što znamo, svaka tačka slike sastoji se iz tri komponente, odnosno tri bajta. Ako se npr. najniži bit svakog bajta upotrebi za inkorporiranje poruke, svaka tačkica će promeniti boju samo za najfiniju nijansu – ljudskom oku slika se neće vidljivo promeniti, a u sebi će sadržati sakrivenu poruku.[41]

Antivirusni programi

Antivirusni programi u neku ruku preuzimaju kontrolu nad računarom. Ovo je neophodno pošto konstantno vrše proveru snimljenih fajlova, pokrenutih programa i otvorenih dokumenata, sprečavajući aktivaciju virusa. Pošto antivirus zauzima visoko hijerarhijsko mesto među pokrenutim programima – pokreće se ranije i može kontrolisati izvršavanje ostalih programa, nepreporučljivo je pokušavati instalaciju dva različita antivirusna paketa – oni veoma teško koegzistiraju.

Osnovna tehnika zaštite, koja datira još od prvih virusa, jeste **prepoznavanje koda** virusa. Antivirus ispituje (skenira) određeni fajl i upoređuje niz njegovih naredbi sa virusima koje ima u bazi podataka. Još od najranijih dana, tvorci virusa su pokušavali da doskoče ovoj tehnici kreiranjem tzv. mutirajućih virusa koji poseduju određeni gabaritni deo koji ničemu ne služi ali koji virus stalno menja sa svakom svojom kopijom. Antivirusni programi su odgovorili inteligentnijim skeniranjem – pronalaženjem onog dela programa virusa koji predstavlja njegov aktivni deo i koji se ne menja (tzv. „potpis“ – *signature*).

Danas virusi mutiraju mnogo uspešnije nego ranije, a osim toga stalno se kreiraju novi virusi koji su varijacije postojećih. Proizvođači antivirusa najčešće ne mogu da isprate svaku varijaciju, a ne može se ni pobeći od glavnog problema – odbrana od virusa uvek dolazi sa zakašnjenjem – virus mora prvo da se pojavi i napravi štetu, a tek kasnije antivirusni programi počinju da ga prepoznaju.

Zbog toga su se antivirusni paketi okrenuli proaktivnoj zaštiti. Prilikom startovanja nekog programa, antivirus ga prvo „isprobava“, kako bi utvrdio da li program pokušava da uradi nešto nedozvoljeno ili da li ima obrazac ponašanja koji liči na neki već poznati virus i blokira svako sumnjivo ponašanje. Ovaj metod prepoznavanja ponašanja programa naziva se **heuristika**. Da bi program mogao da se isproba a da ne pričinu neku stvarnu štetu, u poslednje vreme se sve više koristi tehnika **virtuelizacije**. Već je ranije pomenuta tzv. „virtuelna mašina“, koja predstavlja simulaciju računara unutar računara. Antivirus za nepoznat program kreira jedno ovakvo

virtuelno okruženje, tako da ako program i uradi nešto maliciozno to u stvari nema nikakvog uticaja na korisnikov računar.



Slika 14.2. Kaspersky Internet Security 2011 (Kaspersky Labs) je istovremeno antivirus i firewall

Zaštitni zid – Firewall

Ova metoda zaštite ne pokušava da se bori protiv samih virusa, već, pod pretpostavkom da je najveći deo napada danas povezan sa Internetom, postavlja blokadu na Internet komunikaciju. *Firewall* može biti izveden kao mrežni uređaj ili kao aplikacija.

Komunikacija putem Interneta se odvija tako što klijent upućuje zahteve ka jednom ili više servera. Zavisno od tipa servisa, komunikacija se obavlja na određenom **portu**. Tako je komunikacija za npr. web surfing odvojena od komunikacije za e-mail ili mrežno igranje. Hakeri mogu pokušati napad preko nekog porta preko koga komunicira neka aplikacija koja je ranjiva na napade.

U ovakvim slučajevima od najvećeg značaja je *firewall* koji predstavlja filter mrežnog saobraćaja i dozvoljava komunikaciju putem Interneta samo na određenim portovima.

Firewall takođe zabranjuje pristup „divljim“ paketima sa Interneta – koji ne dolaze kao odgovor na zahteve korisnikovog računara.

Osim toga, moguće je kreirati pravilo, kojim se automatski odbijaju svi paketi koji dolaze sa nekog određenog mesta (npr. piratski ili porno sajt). Ovakav tip zaštite se posebno koristi u poslovnim mrežama gde firma ne dozvoljava zaposlenima pristup ovakvoj vrsti sadržaja.

Što se tiče zaštite naših podataka, *firewall* se može konfigurisati da dozvoli komunikaciju samo izabranim aplikacijama, dok ostale blokira. Na ovaj način se, čak iako je računar zaražen, ograničava delovanje *backdoor* i *spyware* programa.

Korisnicima je ponekad nejasno da za računarski softver važe sasvim drugačija pravila u odnosu na kompjuterski hardver. Hardver predstavlja predmet – kada ga korisnik kupi, on ga i poseduje. Softver se ne poseduje, on samo može da se koristi.

Kada korisnik uloži novac za neki program, on nije platio disk i sam softver na njemu. Platilo je **pravo korišćenja**.

Softverske licence

Najprostiji model licenciranja je po korisniku i računaru – za dati novac, samo jedan korisnik ima pravo da instalira i koristi program na samo jednom računaru. Ovakva licenca može biti proširena na više korisnika i više računara.

Postoji veliki broj različitih tipova licenci – grupne (za više korisnika/računara), ekskluzivne (softver koji je ustupljen na korišćenje jednom jedinom korisniku), paketne (za ceo paket programa), vremenski ograničene (licence koje se moraju obnoviti posle nekog perioda), itd. [48]

Obično se prilikom instalacije komercijalnog (a ponekad i besplatnog) softvera od korisnika zahteva da prihvati tekst pod nazivom **EULA** (*End-User License Agreement*). To je u stvari ugovor u digitalnoj formi koji korisnik mora da prihvati kako bi mogao da koristi softver. Obično se u ovom tekstu proizvođač oslobađa bilo kakve odgovornosti od štete vezane za upotrebu softvera i definiše način na koji korisnik sme da koristi softver. Većina korisnika ne čita ili ne shvata zaista ovaj dokument, što omogućava legalnost različitim *spyware* i sličnim programima.

Postoji nekoliko interesantnih pojava u svetu softvera koje se od strane korisnika često tumače kao besplatan softver.

Beta verzija predstavlja nedovršenu verziju programa koja služi za pronalaženje grešaka. Ovakav softver se najčešće legalno besplatno instalira na računaru, ali kada istekne period testiranja korišćenje takvog softvera postaje nelegalno. Licence za beta verzije mogu imati definisane posebne uslove – npr. obavezu da se šalju izveštaji o radu programa ili određene posebne korisnike koji smeju koristiti program – ovlašćeni beta-testeri, programeri, novinari i dr.[47]

Demo verzija (*demo* ili *trial*) predstavlja besplatan program koji služi za demonstraciju, odnosno procenu (tzv. *evaluation*) mogućnosti pune verzije. Potreba za ovakvim verzijama

programa nastala je zbog toga što su korisnici zatrpani ponudama različitih proizvođača i često ne znaju da li će program zaista zadovoljiti njihove potrebe.[47]

Ovakvi programi su najčešće ograničeni smanjenom funkcionalnošću (npr. nemogućnost rada sa podacima iznad određene veličine, nemogućnost snimanja i sl). Najbolji primer su demo verzije raznih igara – igra je funkcionalna u potpunosti, ali igraču je dostupan samo prvi deo igre.

Još jedan vid ovakvog softvera (ili tačnije – marketinškog pristupa) je i tzv. **šerver** (*shareware*). U pitanju je softver koji takođe služi za procenu, s tim što u većem broju slučajeva ima punu funkcionalnost, ali može biti ograničen vremenski (npr. 30 dana) ili brojem startovanja.

Bez obzira na naziv (engl. "share" – deliti), mogućnost daljeg kopiranja i distribucije zavisi od slučaja do slučaja, tako da je sasvim moguće da dalja distribucija čak i ovakvog softvera bude zabranjena.[49]

Softver koji dolazi uz opremu (**OEM** – *Original Equipment Manufacturer*) predstavlja verziju softvera koju proizvođači računarske opreme isporučuju uz hardver. Primeri su mnogobrojni – program za snimanje DVD-ova koji se dobija uz DVD snimač, OCR softver koji se dobija uz skener, operativni sistem instaliran na novom računaru...

U najvećem broju slučajeva OEM softver predstavlja stariju ili oslabljenu (tzv. "Home" ili "Standard") verziju softvera. Ovakav softver je zabranjeno deliti ili prodavati bez odgovarajućeg hardvera. Kada su OEM operativni sistemi u pitanju, obično dolaze bez uputstva i originalnog pakovanja, što može stvoriti problem ako je potrebno reinstalirati sistem.[47]

„Napušten“ softver (**abandonware**) je onaj softver koji se više ne prodaje, ne razvija, za koji ne postoji tehnička podrška i čiji je proizvođač nestao (bankrotirana, propala firma). Takav softver i dalje podleže zakonu o zaštiti prava – neko uvek nasleđuje prava, s tim što je mala verovatnoća da će nosilac prava pokretati akciju protiv prekršitelja. Softver je često zastareo i čak i ne postoji veliki broj ljudi koji ga koristi.

Besplatan softver (**freeware**) – ovo su zaista besplatni programi. Ipak, i oni mogu biti izdati pod različitim licencama. Besplatan softver ponekad u svoju licencu uključuje način upotrebe softvera – na primer, ako se program koristi u lične ili edukativne svrhe onda je besplatan, a ako se koristi od strane preduzeća ili u komercijalne svrhe potrebno ga je kupiti. Takođe, mogu postojati ograničenja vezana za distribuciju softvera.

Projekti otvorenog koda (**open source**) predstavljaju lepu ideju o deljenju znanja i zajedničkom postizanju cilja za dobrobit svih. U pitanju su projekti gde je izvorni kod programa javno dostupan i gde je omogućeno programerima iz celog sveta da učestvuju u njegovom razvoju. Najpoznatiji takvi projekti su operativni sistem Linux, nekadašnji OpenOffice (sada LibreOffice), baza podataka MySQL itd.

Ovi programi takođe predstavljaju besplatan softver, bar što se tiče krajnjih korisnika. Oni koji žele da vrše distribuciju takvih programa i programeri koji žele da koriste izvorni kod u svojim projektima moraju pažljivo da se upoznaju sa licencom. Obično se takav softver izdaje pod nekom verzijom **GPL** (*GNU General Public License*) licence, što osigurava da neko neće „prisvojiti“ tuđ rad – distribucija ovih programa je dozvoljena samo ako se distribuira i izvorni kod. Programeri koji koriste ovaj kod moraju da ponude i svoj program pod istim uslovima.

Ukoliko je softver izdat pod **BSD** (*Berkley Software Distribution*) ili **MIT** licencom, utoliko je omogućeno njegovo korišćenje čak i u zatvorenim, komercijalnim programima.

Softver u javnom vlasništvu (**public domain**) – ovo je u stvari jedni pravi „slobodan“ softver, odnosno programi koji ne podležu zaštiti intelektualne svojine.

Piraterija

Prema definiciji **BSA** (*Business Software Alliance*), „Softverska piraterija je neovlašćeno kopiranje ili distribucija softvera zaštićenog autorskim pravom. To može biti kopiranje, preuzimanje, deljenje, prodaja ili instaliranje više kopija programa na lične ili radne kompjutere“.[50]

Piraterija košta softverske kompanije, ali i državu zbog narastanja sive ekonomije – crnog tržišta na kome se razmenjuje piratizovani materijal bez plaćanja poreza.

Prema proceni BSA, u Srbiji je u 2010. godini procenat piratizovanog softvera iznosio čak 74%, a njegova procenjena vrednost iznosila je 95 miliona dolara. Iako je procenat piratskog softvera u opadanju i dalje je jedan od najviših u regionu. Naravno, svaki podatak treba gledati u svetlu ostalih informacija. Na primer, procenat piratskog softvera u Nemačkoj iznosi niskih 27%, ali taj procenat predstavlja softver u vrednosti od 2 milijarde dolara. U svetu se procenjuje da je ukupna vrednost instaliranog piratskog softvera oko 59 milijardi dolara – što u stvari predstavlja štetu koju trpe softverske kompanije. Stopa piraterije je uobičajeno korelisana sa niskim standardom i siromaštvom.[51]

Kanali kojima se najčešće širi piratski softver su:

- P2P (*peer-to-peer*) mreže za razmenu velikih fajlova – otvorena razmena ne samo programa, već i filmova, muzike, TV programa;
- aukcijski sajtovi – prodaja zaštićenog softvera;
- B2B (*business-to-business*) sajtovi – prodaja velike količine „legalnog“ softvera;
- stariji načini – ovi načini su danas zapostavljeni zahvaljujući efikasnijim načinima razmene fajlova (IRC mreže, FTP sajtovi, fizički prenos putem CD/DVD medija).

Jedan od najčešćih oblika piraterije je kupovina jedne licence (jednog „originala“) i njegova instalacija na više računara. Licenca je uvek jasna po tom pitanju – nemoguće je zaobići zakon i kasnije se praviti „nevešt“.

U borbi protiv piraterije prvi na udaru su prodavci računarske opreme i preduzeća. Metodi kojima se koriste nadležni organi su klasična prijava i tzv. „lažni kupac“ (kada se na kupljenom računaru proverava da li postoji instaliran piratski softver). Generalno, pirati su dobro organizovani i tehnički obrazovani. Profit koji se zarađuje na prodaji piratskog materijala često nadmašuje čak i trgovinu drogom ili oružjem.

Malo teži slučaj predstavljaju tipovi piraterije kod kojih ne postoji tok novca koji se može pratiti. U pitanju su svi piratski sadržaji koji se mogu pronaći besplatno putem Interneta. U P2P mrežama počinioci su mnogobrojni – svako ko započne proces preuzimanja (*download*) istovremeno postaje i njegov primalac (*leecher*) i delilac (*seeder*). Ovakav proces se može pratiti stalnom kontrolom sadržaja Internet saobraćaja svih korisnika, što nas uvodi u područje narušavanja privatnosti i slobode komunikacije.

Rizici koje uključuje nabavka piratskog softvera su:

- nedobijanje uputstava i odgovarajuće dokumentacije, tehničke podrške ili unapređenja softvera (*upgrade*);
- dobijanje nekompletnih, izmenjenih ili probnih verzija softvera;

- infekcija računara virusima i drugim malicioznim softverom, čime se kriminalcima omogućava pristup privatnim podacima na računaru i sl.

Zakon o autorskom i srodnim pravima je jasan kada kaže da će biti kažnjeno svako lice koje neovlašćeno objavi, zabeleži, umnoži ili javno saopšti autorsko delo ili ih stavlja u promet, daje u zakup ili poseduje umnožene u komercijalne svrhe. Međutim, kršenje zakona nije samo piraterija. Zabranjeno je i korišćenje piratskih kopija, uklanjanje zaštite, kao i prodaja, reklamiranje i komercijalna upotreba svakog uređaja čija je svrha zaobilazanje tehnoloških mera zaštite.[52]

Kod nas se proverom legalnosti instaliranog softvera bavi **Poreska uprava**. Prva mera koja se sprovodi protiv prekršioca je davanje roka u kome mora biti izvršena legalizacija softvera. Ako prekršilac to ne uradi ili ne podnese dokaze o izvršenoj legalizaciji, protiv njega se podnosi prekršajna prijava ili prijava za privredni prestup. Ovakav način uvođenja u legalne tokove je samo privremena mera – u budućnosti, prekršiocima neće biti pružana mogućnost legalizacije, već će se odmah podnositi prijave.

Osoba koja krši autorska prava može biti krivično gonjena i protiv nje se može pokrenuti krivični ili građanski postupak.

Akcije koje se preduzimaju protiv lica u krivičnom postupku su kazna zatvora, naknada materijalne štete i oduzimanje predmeta.

Akcije koje se preduzimaju u građanskom postupku su [52]:

- utvrđivanje i prestanak povrede prava;
- uništenje ili preinačenje predmeta kojima je izvršena povreda prava, uključujući i sve primerke predmeta zaštite, ambalaže, matrice, negative i slično, kao i uništenje ili preinačenje alata i opreme uz pomoć kojih su proizvedeni ti predmeti;
- naknadu imovinske štete;
- objavljivanje presude o trošku tuženog.

Američko pravosuđe prepoznaje praksu tzv. „**fer upotrebe**“ kada se materijal pod zaštitom autorskog prava koristi u smislu kritike, izveštavanja, parodiranja, proučavanja ili edukacije, ali na način kada se ne ostvaruje zarada. Naš Zakon o autorskom i srodnim pravima takođe predviđa mogućnost da se napravi sigurnosna kopija za sopstvene potrebe, u smislu backup-a. Drugim rečima, korisniku je dozvoljeno slušanje MP3 muzike ako je kupio originalni CD. Ono što nije dozvoljeno je dalja distribucija tih fajlova. Sa druge strane, čak iako postoji originalni CD, njegovo preslušavanje je dozvoljeno za ličnu upotrebu – javno izvođenje (npr. u restoranu ili prodavnici) je zabranjeno. Što se tiče računarskih programa, dozvoljeno je kreiranje samo jedne rezervne kopije. Takođe, dozvoljeno je umnožavanje kratkih odlomaka autorskih dela za nekomercijalne svrhe nastave, ispita ili naučnog istraživanja.

Jedna interesantna reakcija je osnivanje **piratskih organizacija** i političkih partija koje promovišu slobodu deljenja sadržaja i ukidanje ili radikalnu izmenu zakona o autorskim pravima. Predstavnici piratskih pokreta svoja načela zasnivaju na ideji da je Internet potpuno promenio način života i društvo.[53] Ove zanimljive ideje zahtevaju suštinsku promenu svesti u društvu o deljenju, slobodi, pravima i intelektualnoj svojini. Ostaje da se vidi da li će to ikada biti ostvareno ili će ostati zapamćeno kao još jedna utopijska vizija budućnosti.

Literatura

1. White, R. (2008), **How Computers Work**, 9th ed. Indianapolis: QUE Publishing.
2. Bianchi, L.M. (2003), **History of Computing and Information Technology**, <http://www.yorku.ca/lbianchi/sts3700b/lecture00a.html> (15.08.2011)
3. Rahman, F. (2011), **Inventors of the Modern Computer (Konrad Zuse)**, <http://fahmirahman.wordpress.com/2011/04/19/inventors-of-the-modern-computer/> (15.08.2011)
4. Rahman, F. (2011), **The History of the ENIAC Computer**, <http://fahmirahman.wordpress.com/2011/04/19/the-history-of-the-eniac-computer/> (15.08.2011)
5. Old Computers, **Osborne 1**, <http://oldcomputers.net/osborne.html> (16.11.2011)
6. Old Computers, **Apple Newton**, <http://oldcomputers.net/apple-newton.html> (16.11.2011)
7. allBusiness, **PaceBlade Launches Tablet PC**, <http://www.allbusiness.com/electronics/computer-equipment-personal-computers/6004956-1.html> (16.11.2011)
8. MacRumors, **Apple Tablet Media Event Today: "Come See Our Latest Creation"**, <http://www.macrumors.com/2010/01/27/apple-tablet-media-event-today-come-see-our-latest-creation/> (16.11.2011)
9. The Globe and Mail, **Head in the Clouds? Welcome to the Future**, <http://www.theglobeandmail.com/news/technology/article137127.ece> (18.11.2011)
10. Rosch, W.L. (2003), **Hardware Bible**, 6th ed. Indianapolis: QUE Publishing.
11. Intel, **From Sand to Silicon: The Making of a Chip**, <http://newsroom.intel.com/docs/DOC-2476> (15.03.2010)
12. Intel Corporation, **Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture**, http://www.intel.com/Assets/en_US/PDF/manual/253665.pdf (15.03.2010)
13. Harris, T., How Stuff Works, **How CD Burners Work**, <http://computer.howstuffworks.com/cd-burner.htm> (20.03.2010)
14. IEC – International Electrotechnical Commission, **Prefixes for Binary Multiples**, http://www.iec.ch/zone/si/si_bytes.htm (05.06.2010)
15. The Unicode Consortium, **Unicode 5.1.0**, <http://www.unicode.org/versions/Unicode5.1.0/> (05.06.2010)

16. Kuhn, M. (1999), **UTF-8 and Unicode FAQ for Unix/Linux**, <http://www.cl.cam.ac.uk/~mgk25/unicode.html> (05.06.2010)
17. Brain, M., How Stuff Works, **How Analog and Digital Recording Works**, <http://communication.howstuffworks.com/analog-digital1.htm> (10.06.2010)
18. Elmasri, R., Carrick, A. G. and Levine, D. (2010), **Operating Systems: A Spiral Approach**, New York: McGraw-Hill.
19. Stallings, W. (2009), **Operating Systems: Internals and Design Principles**, New Jersey: Prentice Hall.
20. Doeppner, T.W. (2011), **Operating Systems in Depth**, Hoboken: John Wiley & Sons.
21. MaximumCompression, **Lossless vs. lossy compression**, http://www.maximumcompression.com/lossless_vs_lossy.php (10.08.2011)
22. File Extension Library, **Archive and compressed file extension list**, <http://www.file-extensions.org/filetype/extension/name/archive-and-compressed-files> (10.08.2011)
23. WotsIt, **Graphics Files**, <http://www.wotsit.org/list.asp?fc=1> (10.08.2011)
24. Amos, M. (2008), Microsoft MSDN, **Naming Files, Paths, and Namespaces**, [http://msdn.microsoft.com/en-us/library/aa365247\(v=vs.85\).aspx#maxpath](http://msdn.microsoft.com/en-us/library/aa365247(v=vs.85).aspx#maxpath) (10.08.2011)
25. Daily, S. Microsoft TechNet, **Optimizing NTFS**, <http://technet.microsoft.com/en-us/library/cc767961.aspx> (10.08.2011)
26. Urošević, D. (1996), **Algoritmi u programskom jeziku C**, Beograd: Mikro knjiga.
27. Waterfall Model, **All About the Waterfall Model**, <http://www.waterfall-model.com/> (15.06.2011)
28. Coplien, J.O. and Bjornvig, G. (2010), **Lean Architecture for Agile Software Development**, Chichester: John Wiley & Sons.
29. Donaldson, S.E. (2001), **Successful Software Development**, 2nd ed. New Jersey: Prentice Hall.
30. Eckel, B. (2002), **Misliti na Javi**, prev. 2. izd. Beograd: Mikro knjiga.
31. Cantu, M. (2006), **Delphi 2005 bez tajni**, Čačak: Kompjuter biblioteka.
32. Stankić R. i Stankić M. (2008), **Informatika u turizmu**, Beograd: Visoka turistička škola.
33. Unixspace, **Database Models**, <http://unixspace.com/context/databases.html> (20.04.2011)
34. Ambler, S.W. (2011), AgileData, **On Relational Theory: Questioning the Dogma**, <http://www.agiledata.org/essays/relationalTheory.html> (20.04.2011)
35. Ambler, S.W. (2010), AgileData, **Introduction to Data Normalization: A Database "Best" Practice**, <http://www.agiledata.org/essays/dataNormalization.html> (20.04.2011)
36. Internet World Stats, **World Internet Usage and Population Statistics**, <http://www.internetworldstats.com> (07.07.2011)
37. Wikipedia Statistics, **Wikipedias**, <http://stats.wikimedia.org/EN/Sitemap.htm> (07.07.2011)

38. Wikipedia, **Wikipedia: Database Download**,
http://en.wikipedia.org/wiki/Wikipedia_database (07.07.2011)
39. Hambridge, S. (1995), **Netiquette Guidelines**, <http://www.ietf.org/rfc/rfc1855.txt>
(07.07.2011)
40. NetLingo, **Definicija: leetspeak**, <http://www.netlingo.com/word/leetspeak.php> (07.07.2011)
41. Cole, E., Krutz, R. and Conley J. W. (2005), **Network Security Bible**, Indianapolis: Wiley Publishing.
42. Bidwell, T. (2002), **Hack Proofing Your Identity in the Information Age**, Rockland: Syngress Publishing.
43. Mao, W. (2004), **Modern Cryptography: Theory and Practice**, New Jersey: Prentice Hall PTR.
44. Galbreath, N. (2002), **Cryptography for Internet and Database Applications**, Indianapolis: Wiley Publishing.
45. Roebuck, T.A. (2005), Computer Crime Research Center, **Network security: DoS vs DDoS attacks**
<http://www.crime-research.org/articles/network-security-dos-ddos-attacks/> (25.07.2011)
46. The Official Captcha Site, **CAPTCHA: Telling Humans and Computers Apart Automatically**, <http://www.captcha.net> (25.07.2011)
47. Tseng, M., RevGen Group, **25 Software Licensing Models**,
<http://www.revengroup.com/resources/25+Software+Licensing+Models.pdf> (01.09.2011)
48. Ingram, R. (2006), Oil Information Technology Journal, **The Evolution of Software Licensing Models**, <http://www.oilit.com/papers/licenstracker.pdf> (01.09.2011)
49. Association of Software Professionals, **Software Development and Distribution FAQ**,
<http://www.asp-software.org/resources/developer-faq.asp> (01.09.2011)
50. BSA, **Šta je softverska piraterija**, <http://www.bsa.org> (01.09.2011)
51. BSA Global Software Piracy Study, **Eighth Annual BSA Global Software Piracy Study**,
<http://portal.bsa.org/globalpiracy2010/index.html> (01.09.2011)
52. **Zakon o autorskom i srodnim pravima** (2009), Službeni glasnik RS broj 104/2009.
53. Piratpartiet, **Politics and Principles**, <http://www.piratpartiet.se/international/english>
(01.09.2011)
54. Wikipedia, **File:Transistor Count and Moore's Law – 2011.svg**,
http://en.wikipedia.org/wiki/File:Transistor_Count_and_Moore%27s_Law_-_2011.svg
(15.11.2011)
55. PTC, **Creo Parametric**, <http://www.ptc.com/product/pro-engineer> (10.04.2011)
56. Cakewalk, **Music Creator 6**, <http://www.cakewalk.com/Products/MusicCreator/>
(10.04.2011)
57. Cakewalk, **Studio Instruments**,
<http://www.cakewalk.com/Products/StudioInstruments/default.aspx> (10.04.2011)

58. Softpedia, **Adobe Director Screenshots**,
<http://www.softpedia.com/progScreenshots/Adobe-Director-Screenshot-118610.html>
(10.04.2011)
59. Apple, **OS X Lion – OS X. It's what makes a Mac a Mac**,
<http://www.apple.com/macosx/what-is/> (20.11.2011)