INFORMATIKA

Beograd, 2021.

SADRŽAJ

Predgovor VIII

1. HARDVER RAČUNARA 1
1.1. Uvod 1
1.2. Osnovne funkcije računara 1
1.3. Binarni brojni sistem 3
1.4. Hardverske komponente CPU računarskog sistema 7
1.4.1. Kućište računara 8
1.4.2. Matična ploča 9
1.4.3. Centralna procesorska jedinica 10
1.5. Računarske memorije 11
1.5.1. Registri 12
1.5.2. Keš memorije 13
1.5.3. Primarna memorija 14
1.5.4. Sekundarne memorije 14
1.5.5. Ostale memorije računara 15
1.6. Ostale hardverske komponente CPU računara 15
1.7. Ključni termini 17

2. PERIFERIJSKI UREĐAJI 19
2.1. Uvod 19
2.2. Ulazni uređaji 19
2.3. Digitalizacija i skeneri 23
2.4. Izlazni uređaji računarskog sistema 26
2.5. Audio ulaz/izlaz računara 30
2.6. Ulazno/izlazni uređaji za skladištenje podataka 31
2.7. Ergonomija i zdravlje 32
2.8. Zaključak 33
2.9. Ključni termini 34

3. GRAFIKA I MULTIMEDIJA 36
3.1. Uvod 36
3.2. Osnovni pojmovi i istorijski pregled 36
3.3. Proporcija i principi kompozicije 38

3.3.1. Kompozicija slike 39
3.4. Kompresija 41
3.5. Računarska grafika 44
3.5.1. Rasterska grafika 46
3.5.2. Vektorska grafika 47

4. INTERAKCIJA ČOVEKA I KOMPJUTERA 50
4.1. Uvod 50
4.2. Upotrebljivost 51
4.3. Principi i mere upotrebljivosti 52
4.3.1. Stanje sistema na vidnom mestu 52
4.3.2. Paralela između sistema i sveta oko nas 53
4.3.3. Korisnička kontrola i sloboda 54
4.3.4. Doslednost i standardi 55
4.3.5. Prevencija grešaka 55
4.3.6. Prepoznavanje je bolje od pamćenja 56
4.3.7. Fleksibilnost i efikasnost korišćenja 57
4.3.8. Estetika i minimalistički dizajn 57
4.3.9.Prepoznati grešku, postaviti dijagnozu i oporavak od greške 58
4.3.10. Pomoć i dokumentacija 59
4.4. Primeri lošeg dizajna 59
4.5. Proces planiranja novog interfejsa 61
4.5.1. Radno okruženje 62
4.5.2. Korisnici 62
4.5.3. Prototip 63
4.5.4. Testiranje 64
4.6. Ključni termini 65

5. OBRADA SLIKE I KOMPJUTERSKA VIZIJA 69
5.1. Uvod 69
5.2. Primena kompjuterske vizije 70
5.2.1. Pronalaženje lica 71
5.2.2. Industrijske primene 72
5.2.3. Biometrija 73
5.2.4. Autonomna vožnja 75
5.3. Boje, histogrami i operatori 76
5.4. HDR slike 80
5.5. Ključni termini 82

6. VISUAL BASIC **87**
6.1. Uvod 87
6.2. Visual Basic kod 89
6.3. IF klauzula 90
6.4. Zbir cifara celih brojeva 92
6.5. For, While petlje 94
6.6. Čitanje i pisanje u Excel ćelije 97
6.7. Sortiranje 99
6.8. Najveći zajednički delilac i najmanji zajednički faktor 103
6.9. Ključni termini 105

7. BAZE PODATAKA **107**
7.1. Uvod 107
7.2. Osnovni koncepti i definicije 108
7.2.1. Podatak 108
7.2.2. Informacija 108
7.2.3. Metapodaci - podaci o podacima (metadata) 110
7.2.4. Sistem za upravljanje bazama podataka 111
7.3. Klasičan sistem zasnovan na datotekama 112
7.3.1. Nedostaci sistema zasnovanog na datotekama 113
7.4. Pristup zasnovan na bazama podataka 114
7.5. Primene baze podataka 114
7.6. Modelovanje 115
7.6.1. Razvoj konceptualnih modela 116
7.6.2. Entiteti 117
7.6.3. Veze između entiteta 117
7.7. Relacioni model BP 118
7.7.1. Model objekti veze 121
7.8. SQL – jezik relacione baze podataka 122
7.9. Oporavak baze podataka 122
7.10. Ključni termini 123

8. RAČUNARSKE MREŽE **125**
8.1. Uvod 125
8.2. Komunikacioni sistem 126
8.3. Vrste prenosa podataka 127

8.3.1. Prenos podataka sa komutacijom veza (*Circuit Switched*) 127
8.3.2. Prenos podataka sa komutacijom paketa (*Packett Switched*) 128
8.4. Vrste računarskih mreža prema prostoru koji obuhvataju 129
8.4.1. Lokalna računarska mreža (*Local Area Network, LAN*) 129
8.4.2. Regionalna računarska mreža (*Wide Area Network, WAN*) 131
8.5. Pristup umrežavanju 131
8.5.1. Mreže ravnopravnih računara 131
8.5.2. Serverske mreže 132
8.6. Standardne topologije 133
8.7. Pasivna mrežna oprema 134
8.7.1. Koaksijalni kabl 135
8.7.2. Kabl sa upredenim paricama (*Twisted Pair*) 136
8.7.3. Optički kablovi 136
8.8. Aktivna mrežna oprema 137
8.8.1. Ripiter (*Repeater*) 138
8.8.2. Hab (*Hub*) 138
8.8.3. Mrežni most (*Bridge*) 138
8.8.4. Komutator – svič (*Switch*) 139
8.8.5. Ruter - usmerivač (*Router*) 139
8.8.6. Mrežni prolaz (Gateway) 140
8.8.7. Bezbednosna barijera (Firewall) 141
8.8.8. Mrežna kartica 142
8.9. Eternet (*Ethernet*) 142
8.9.1. Internet model (*TCP/IP*) 142

9. KRIPTOVALUTE 145
9.1. Uvod 145
9.2. Razvoj elektronskog poslovanja i elektronske trgovine 145
9.3. Poslovni modeli e-trgovine 149
9.4. Kriptovalute kao moderini način e-trgovine 152
9.4.1. Bitcoin 152
9.4.2. Prednosti i mane bitcoin-a 156
9.5. Ključni termini 159

10. BEZBEDNOST I RIZICI **160**
10.1. Uvod 160
10.2. Računarski Kriminal 161
10.2.1. Krađa preko računara 162
10.2.2. Softverska piraterija i zakon o intelektualnoj svojini 163
10.2.3. Intelektualna svojina i pravo 164
10.3. Zlonamerni softver 165
10.3.1. Virusi 167
10.3.2. Trojanski konji 168
10.3.3. Crvi 168
10.3.4. Antivirusna zaštita 169
10.4. Hakovanje i elektronski upadi 170
10.5. Bezbednost – smanjenje rizika 171
10.5.1. Ograničenja fizičkog pristupa 172
10.5.2. Lozinke 172
10.5.3. Mrežne barijere 173
10.5.4. Kriptografske tehnike – šifrovanje/dešifrovanje 174
10.5.5. Bekapi I druge bezbednosne mere 175
10.6. Bezbednost, privatnost, sloboda i etika 176
10.7. Zaključak 177
10.8. Ključni termini 178

1. HARDVER RAČUNARA

Posle čitanja i razumevanja ovog poglavlja studenti će biti sposobni da objasne principe na kojima računari skladište i manipulišu informacijama, da opišu osnovnu strukturu i organizaciju računara i diskutuju bazične funkcije i interakcije glavnih internih hardverskih komponenti računara, uključujući različite tipove memorija i uređaja za skladištenje.

1.1. UVOD

Za razumevanje rada računara, običan korisnika ne mora znati sve tehničke detalje. Dovoljno je znati odgovore na pitanja: "kako računar radi ono što radi", i to objasniti sa osnovnim, ne-tehničkim rečnikom. Korisnici zainteresovani za dublje tehničke detalje mogu lako naći dodatne informacije u priloženoj literaturi i na web stranicama.

Hardver računarskog sistema obuhvata dve glavne grupe fizičkih komponenti: samog računara–CPU (centralne procesorske jedinice) koje su opisane u ovom poglavlju i periferijskih uređaja, detaljnije opisanih u 2. poglavlju ovog udžbenika.

Da bi korisnik razumeo osnovne funkcije hardvera računara, potrebno je znati šta računar stvarno radi i kako upisuje i očitava podatke sa čvrstog diska.

1.2. OSNOVNE FUNKCIJE RAČUNARA

U suštini računar izvršava samo četiri osnovne funkcije (Slika 1.1): prima ulazne informacije iz fizičkog okruženja, obrađuje informacije izvršavajući aritmetičko-logičke operacije nad ulaznim informacijama, proizvodi izlazne informacije za okruženje - korisnike, druge računare ili mašine i skladišti informacije u memorijske uređaje.

Slika 1.1. Ilustracija osnovnih funkcija računarskog sistema

Sa aspekta funkcionalnosti računara i generičke definicije sistema, računar je – *sistem*. Svaki *računarski sistem* sadrži hardverske komponente – fizičke delove računara, koji

su specijalizovani za izvršavanje svake od navedenih funkcija. Računarski hardver je platforma za procesiranje informacija i na osnovu tog kriterijuma deli se na:

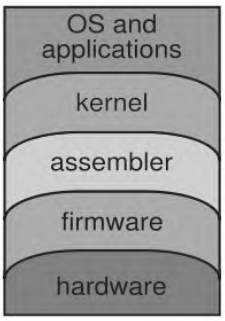
Ulazne uređaje: tastature (*keyboards*), uređaje za pokazivanje (miš, *touchpad, touchpen*), skenere, čitače bar kodova, mikrofone, koji primaju ulazne informacije iz okruženja i predstavljaju interfejs između korisnika i CPU računara;

Procesor (CPU): „mozak“ računara koji procesira informacije izvršavajući aritmetičke proračune i logičke operacije i donosi osnovne odluke na bazi poređenja vrednosti informacija, a obuhvata mikroprocesor, matičnu ploču, primarnu memoriju (RAM), jedinicu za napajanje i dodatne namenske (ekspanzione) kartice;

Izlazne uređaje: monitore koji u okruženje šalju vizuelne izlazne informacije, štampače (printere) – štampane informacije i zvučnike -tonske informacije;

Memorije i uređaje za skladištenje informacija: memorija računara često se naziva *primarna memorija*, *glavna memorija* ili samo *memorija*, najčešće RAM (*Random Access Memory*) tipa, a koriste se za skladištenje programa i podataka kojima CPU neposredno treba da pristupi; uređaji za skladištenje uključuju čvrsti disk – HD (*Hard Drive*) koji se često naziva *interna sekundarna* ili *on-line* memorija, zatim CD, DVD, USB i drugi *spoljni sekundarni uređaji za skladištenje*-ZIP diskovi, magnetne trake i fleš memorije, koji su istovremeno ulazni i izlazni uređaji, u zavisnosti da li se informacije učitavaju ili se očitavaju.

Kombinacija ova četiri tipa komponenti čini hardverski deo računarskog sistema. Naravno, računarski sistem je nekompletan bez *programa* ili *softvera,* koji daje instrukcije hardveru šta treba da radi. Konceptualni dizajn i fundamentalna operativna struktura računarskog sistema u kompjuterskom inženjeringu naziva se *arhitektura računarskog sistema*, koja sadrži funkcionalni opis zahteva, posebno za brzinu i interne veze i implementaciju dizajna za različite delove računara. Ovo se najviše odnosi na interno funkcionisanje CPU i brzinu pristupa adresama memorije računara. Tipična vizija arhitekture računara prikazuje se serijom apstrakcionih slojeva: *hardver, firmware, assembler, kernel, operativni sistem i aplikacije*, (Slika 1.2).



Slika 1.2. Tipična vizija arhitekture računarskog sistema

CPU i primarna memorija su glavne hardverske komponente koje učestvuju u svim računarskim operacijama. Periferijske uređaje računarskog sistema čine ulazni i izlazni uređaji i sekundarni uređaji za skladištenje informacija. Pošto je svaka komponenta računarskog hardvera dizajnirana za transport, ili transformaciju informacija, potrebno je razumeti kako računar zapisuje informacije.

1.3. BINARNI BROJNI SISTEM

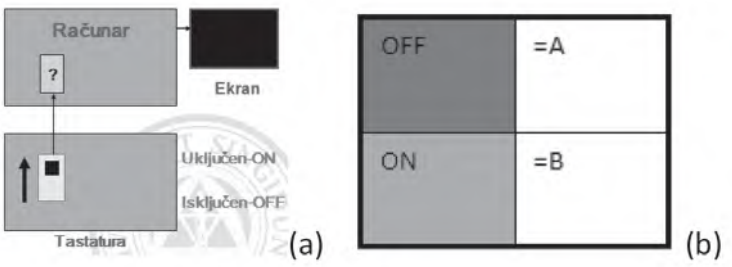
Pojam *informacija* teško se jednoznačno definiše pošto ima više značenja, od kojih su šire prihvaćene dve:

1. *Informacija je svaka komunikacija skupa podataka koji imaju vrednost za primaoca, informiše ga i povećava njegovo znanje*. Ova definicija pravi razliku od podataka saopštenih putem TV, novina, računara i drugih izvora koji primaocu ne donose nova znanja. Dakle, svaki korisnik individualno primenjuje lična merila za definisanje informacije iz obimnog skupa nepovezanih podataka.
2. *Informacija je svaka komunikacija podataka bez obzira ima li ili nema vrednost za primaoca.* Prema ovoj generičkoj definiciji komunikacione teorije informacija može biti u bilo kojoj formi: reč, broj, slika, zvuk, simbol – sve što se može preneti komunikacionim kanalima.

Kako računar generalno transformiše sirove ulazne podatke koji nemaju vrednost za korisnika u tekućem obliku, u izlazne informacije koje imaju vrednost za korisnika, čini se da više odgovara prva definicija informacija. Ovakav pristup, takođe, ističe ulogu računara u procesiranju podataka poslovnih procesa. Međutim, kako izlazna informacija može biti ulaz u drugi računar, teško je primeniti personalna merila o vrednosti tog izlaza, pa je objektivnije i opštije primeniti drugu definiciju informacija. U tom smislu, u ovom udžbeniku obe definicije su primenjene ravnopravno, a iz konteksta se vidi na koju se definiciju konkretno odnosi.

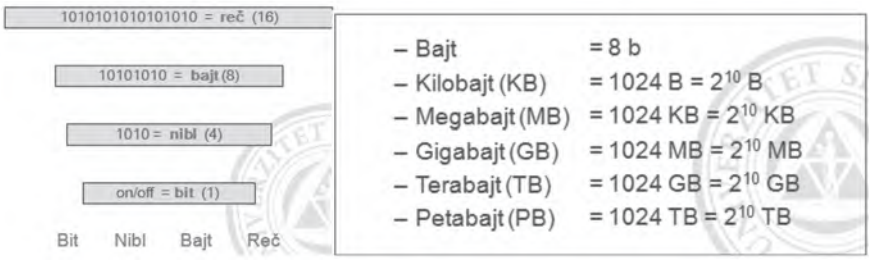
U računarskom okruženju informacije se zapisuju, prenose i skladište u *digitalnom* odnosno *binarnom* obliku; sačinjene su od diskretnih jedinica koje se mogu računati i deliti na manje jedinice. Računar ne razume reči, brojeve, slike, muzičke note ili slova i ne može procesirati informacije bez deljenja u manje jedinice. Tačnije, računar može prepoznati samo informacije razložene do nivoa *bita*. Bit (***Bi**nary Digi**t***) je najmanji deo digitalne informacije i može imati vrednost samo *dva* stanja – **1** i **0**, za razliku od digitalnog signala koji može imati bilo koju vrednost. Priroda i fizičko okruženje su puni binarnih stanja, na primer: ima-nema, uključen-isključen, crno-belo, da-ne, visok-nizak itd. Računar se u krajnjoj aproksimaciji može predstaviti matricom finih prekidača. Ako se zamisli da je memorija računara za upisivanje informacija skup mikroskopskih prekidača (*on/off*) realizovanih, na primer, magnetskim česticama sa polovima N-S, lako je razumeti da računar procesira informacije bit po bit. Svaki prekidač skladišti sićušni deo informacija. Na primer, jednostavni dizajn 1-bitnog računara može se predstaviti

jednim strujnim prekidačem koji signalizira CPU jedinicu po jedno binarno stanje u svakom trenutku. To stanje CPU prikazuje specifičnim simbolom na ekranu CPU (Slika 1.3a). Zavisno od položaja prekidača, na ekranu će biti prikazano, recimo **A** ako je prekidač isključen, a **B** ako je uključen (Slika 1.3b).



Slika 1.3. Jednostavan dizajn jednobitnog računara

U računaru se bitovi (b) grupišu u veće binarne jedinice: nibl – 4 bita=4b, 1 bajt =1B=8 bita ili dva nibla i binarna reč – 16 bita ili 2 bajta (Slika 1.4). Skup od 8 bita ili 1B može predstaviti 2^8=256 brojeva između 0 i 255. Ako su svi prekidači na **Off** vrednost je 0, a ako su svi – **On** vrednost je 255. Veći brojevi od 255 predstavljaju se sa više bajta. Reč od 2 bajta (2B) može predstaviti 2^{16}=65.536 (0 do 65.535) vrednosti. Brojni sistem koji predstavlja sve brojeve sa kombinacijom dve cifre naziva se *Binarni brojni sistem.* Baš kao i dekadni (decimalni) brojni sistem na osnovi 10 kojeg svakodnevno koristimo, binarni sistem ima jasna, konzistentna pravila za sve aritmetičke operacije. Računar memoriše u primarnu memoriju (RAM) u jednom trenutku celu binarnu reč. Po dužini binarne reči koje se mogu odjednom memorisati u jednu liniju RAM memorije, računar se naziva: 8-bitni, 16-bitni, 32-bitni ili 64-bitni.

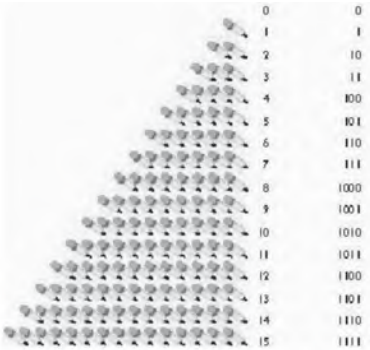


Slika 1.4. Binarni brojni sistem (levo), Odnosi osnovnih binarnih jedinica (desno)

Međutim, pokušaj da se nauči kako računar radi na osnovu ispitivanja operacija računara na nivou bita, isto je što i utvrđivanje izgleda čoveka na osnovu ispitivanja individualnih ćelija. Na sreću, ljudi mogu koristiti računare bez razmišljanja o binarnoj

prezentaciji. Međutim, većina korisnika računara treba da ima osnovno razumevanje međusobnih odnosa binarnih jedinica za kvantizaciju podataka, (Slika 1.4 desno). Skraćenice K, MB, GB, TB opisuju kapacitet memorije i uređaja za skladištenje, a koriste se i za označavanje veličine fajla (datoteke) u fajl sistemu računara. Fajl računara je organizovani skup informacija, uskladištenih u čitljivom obliku za računar. Na primer, tekst od 20-ak stranica uskladišten je u fajl veličine od oko 75KB prostora na fizičkom disku. Da bi se izbegla konfuzija brzina rada računara – transfera podataka, označava se u **Mbps** (megabita/sekundi), a ne MB.

Sa binarnim brojnim sistemom mogu se zapisivati brojevi, tako što se kodiraju svi brojevi kombinovanjem samo dve binarne cifre - 1 i 0. Decimalni brojevi se mogu konvertovati u binarne i obrnuto. Savremeni računari sadrže program koji automatski konvertuje decimalni brojni sistem u binarni i obrnuto, što praktično znači da je procesiranje binarnih brojeva potpuno sakriveno od korisnika. Primer decimalne i binarne reprezentacije brojeva dat je na slici 1.5a.



a)

A	1000001	N	1001110
B	1000010	O	1001111
C	1000011	P	1010000
D	1000100	Q	1010001
E	1000101	R	1010010
F	1000110	S	1010011
G	1000111	T	1010100
H	1001000	U	1010101
I	1001001	V	1010110
J	1001010	W	1010111
K	1001011	X	1010111
L	1001100	Y	1011001
M	1001101	Z	1011010

b)

Slika 1.5. Decimalna i binarna reprezentacija brojeva (a), ASCII kôdovi (b)

Međutim, savremeni računari rade jednako sa tekstom i brojevima. Da bi se reči, rečenice i paragrafi učinili razumljivim za binarna kola računara, programeri su morali razviti kodove koji predstavljaju svako slovo, broj i specijalni karakter kao jedinstven niz bita. *Kodiranjem* binarnih informacija mogu se predstaviti slova, interpunkcijski i drugi karakteri, ili uopšte bilo kakvi signali ili informacije koje se konvertuju u binarnu formu. Najčešće korišćeni kôd u računarstvu je ***ASCII (****American Standard Code for Information Interchange),* koji ima 256 jedinstvenih znakova i predstavlja svaki karakter sa jedinstvenim 8-bitnim kodom. U ASCII kodu mogu se predstaviti 256 jedinstvenih karaktera: 26 slova engleskog alfabeta (velikih i malih), 10 cifara (0-9), (Slika 1.5b) i različitih specijalnih karaktera. Kako ASCII kod nije mogao predstaviti slova drugih svetskih jezika (npr., Grčki, Hebrejski, Arapski i dr.), razvijen je noviji - ***Unicode*** kodni sistemi koji podržava 65.000 jedinstvenih karaktera, što je više nego dovoljno za sve glavne svetske jezike. U tabeli 1.1 prikazane su uporedne vrednosti kodiranja karaktera osnovnih kodnih sistema koji se primenjuju u računarskoj tehnici: *decimalni, heksadecimalni, oktalni, binarni i ASCII*.

Primer: Binarni broj - 01010011 je decimalni - 83, ASCII – S, a heksadecimalni - h53 ili 0x53 ili $53_{1.}$

Grupe bita mogu, takođe, predstaviti boje i zvukove sa kojima savremeni računari rade, kao i bilo koju vrstu informacije koju računari mogu procesirati (temperaturu, pritisak i slika), kvantitativnim, senzorskim merenjem u okruženju. Pored načina na koji računari predstavljaju podatke – informacije iz okruženja, važno je razumeti kako računar skladišti programe. Računar, takođe, skladišti programe kao kolekciju bita. Na primer, kôd 01101010 može značiti instrukciju za računar da *doda dva broja*.

DECIMAL	HEX	OCTAL	BINARY	ASCII	MEANING
57	39	071	0011 1001	9	Nine
58	3A	072	0011 1010	:	Colon
59	3B	073	0011 1011	;	Semi-colon
60	3C	074	0011 1100	<	Less than
61	3D	075	0011 1101	=	Equals
62	3E	076	0011 1110	>	Greater than
63	3F	077	0011 1111	?	Question mark
64	40	100	0100 0000	@	Commercial at
65	41	101	0100 0001	A	Uppercase A
66	42	102	0100 0010	B	Uppercase B
67	43	103	0100 0011	C	Uppercase C
68	44	104	0100 0100	D	Uppercase D
69	45	105	0100 0101	E	Uppercase E
70	46	106	0100 0110	F	Uppercase F
71	47	107	0100 0111	G	Uppercase G
72	48	110	0100 1000	H	Uppercase H
73	49	111	0100 1001	I	Uppercase I
74	4A	112	0100 1010	J	Uppercase J
75	4B	113	0100 1011	K	Uppercase K
76	4C	114	0100 1100	L	Uppercase L
77	4D	115	0100 1101	M	Uppercase M
78	4E	116	0100 1110	N	Uppercase N
79	4F	117	0100 1111	O	Uppercase O
80	50	120	0101 0000	P	Uppercase P
81	51	121	0101 0001	Q	Uppercase Q
82	52	122	0101 0010	R	Uppercase R
83	53	123	0101 0011	S	Uppercase S
84	54	124	0101 0100	T	Uppercase T
85	55	125	0101 0101	U	Uppercase U

Tabela 1.1. Tabela konverzija kôdnih sistema

Računar koristi *Binarni brojni sistem* za upisivanje programa, podataka instrukcija na fizički (čvrsti) disk. Sačuvani (memorisani) programi i podaci se smeštaju kao skupovi bita na fizičkom disku, a programske instrukcije se predstavljaju u binarnom zapisu kao odgovarajući kodovi instrukcija.

Korišćenjem dva binarna simbola (0 i 1) mogu se predstaviti svi ulazni brojevi, slova i drugi karakteri u računarski sistem i nad njima se mogu vršiti proizvoljne aritmetičke operacije. Bilo koji binarni broj se takođe, konvertuje u dekadni, sumiranjem proizvoda binarnih cifara i njihovih *težinskih faktora*, **2^n, n=0, 1, 2, ...m. Primer**: Decimalni broj 19 se predstavlja u binarnom brojnom sistemu sa 00010011 (Slika 1.6).

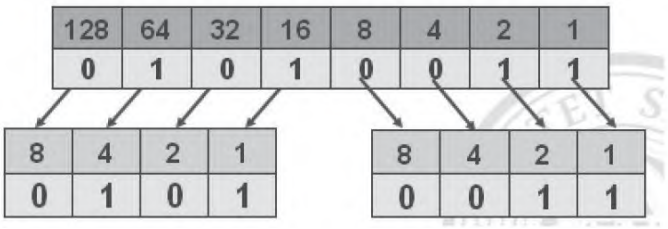
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	0	1	0	0	1	0	1	=> 1*32+1*4+1*1 = 37

Slika 1.6. Konverzija decimalnog u binarni broj

Osnovni nedostatak binarnog zapisa brojeva je predugačak zapis i preveliko trošenje memorije računara. Zato se koriste drugi, kraći brojni sistemi. U računarskim sistemima najčešće se koristi *heksadecimalni (hex)* brojni sistem koji ima 16 cifara: **0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.**

Primeri: 0100 0001 b = 41 h, a 1010 0111 b = A7 h.

Konverziju binarnog u *hex* zapis vrši računar. Za određivanje *hex* vrednosti treba podeliti bajt u 2 nibla i ponderisati ponovo svaki nibl sa desna na levo (Slika 1.7).

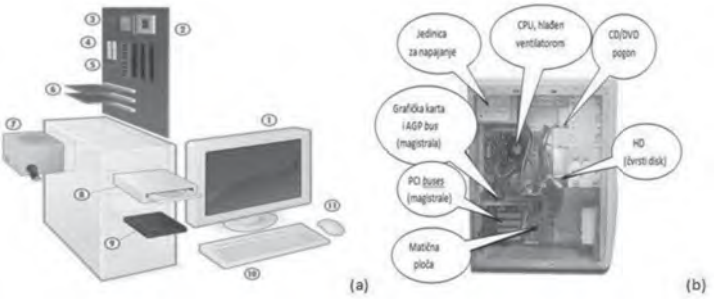


Slika 1.7. Konverzija binarnog u hex zapis

1.4. HARDVERSKE KOMPONENTE CPU RAČUNARSKOG SISTEMA

Hardver računarskog sistema prikazan je na slici 1.8 (a): (1) monitor, (2) matična ploča, (3) BIOS (ROM), (4) procesor, (5) primarna memorija (RAM), (6) grafička i zvučna kartica, (7) jedinica za napajanje, (8) pogonska jedinica za CD/DVD, (9) pogonska jedinica za

disketu, (10) tastatura, (11) miš i glavne unutrašnje komponente računara (Slika 2.10b). Hardver se mnogo ređe menja nego softver. Mogućnosti računara u najvećoj meri zavise od kvaliteta performansi hardvera.



Slika 1.8. Hardver računarskog sistema

Periferijske jedinice tastatura i miš su, takođe, povezani sa glavnom pločom kablom preko ulazno/izlaznog (I/O) panela na zadnjoj strani računara. Monitor je povezan preko I/O panela kroz port na glavnoj ploči ili na grafičkoj kartici. Nekoliko funkcija, implementiranih sa čipsetom (chipsets) može biti integrisano na glavnoj ploči, tipično to su USB i mrežni konektor, ali mogu biti i grafička i zvučna kartica. Grafička i zvučna kartica mogu imati elektromagnetni oklop za zaštitu analognih elemenata od elektromagnetne radijacije u računaru. Za skladištenje velike količine podataka mogu se koristiti dodatni HD instaliran u kućište računara ili eksterni HD u posebnom kućištu.

Hardverski kapaciteti personalnog računara mogu se proširiti dodavanjem ekspanzionih kartica, povezanih na glavnu ploču preko ekspanzione magistrale (bus). Neke standardne periferijske magistrale, koje se često koriste za dodavanje ekspanzionih kartica u PC računarima od 2005 godine su PCI i AGP (PCI velike brzine namenjene za grafičke adaptere) i PCI Express. Većina PC računara od 2005 imaju više fizičkih PCI ekspanzionih slotova. Mnogo računara takođe uključuje AGP magistralu i ekspanzione slotove, ali samo nekoliko sadrži obe magistrale.

Savremeni personalni desktop računari (PC) tipično sadrže sledeće interne hardverske komponente u kućištu računara sa jedinicom za napajanje (često se naziva CPU): matičnu ploču (motherboard), CPU-centralnu procesorsku jedinicu (procesor), primarnu memoriju (RAM), video (grafičku) karticu, čvrsti disk (hard disc), optički disk (DVD-ROM), tastaturu i miš i monitor.

1.4.1. Kućište računara

Kućište računara je struktura koja sadrži glavne komponente računara, obični izrađena od čelika, aluminijuma ili plastike, mada se mogu koristiti i drugi materijali, kao što su drvo, pleksiglas i dr. Kućišta mogu imati različite veličine i faktore forme, koji su uglavnom

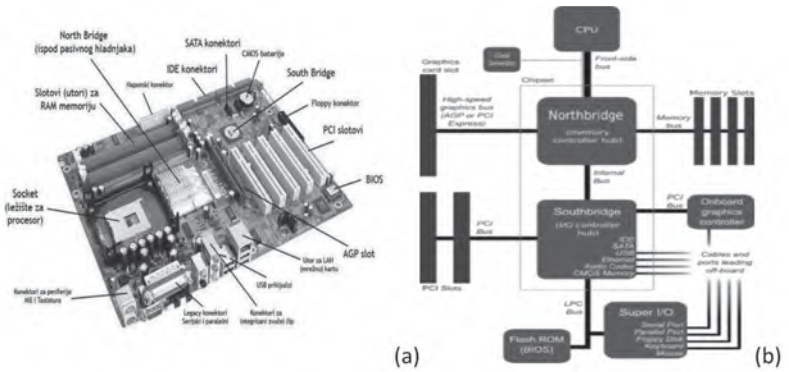
određeni faktorom forme glavne ploče, pošto je najveća komponenta računara. Danas je najpopularniji faktor forme kućišta desktop računara ATX kućište, mada microATX i mali faktori forme postaju vrlo popularni za različite namene.

Napajanje računara obezbeđuje jedinica za napajanje (power unit), koja vrši konverzija naizmeničnog (AC) u istosmerno (DC) napajanje – AC 220 V u DC 3,3 V, 5 V i 12 V. Ima sopstveno hlađenje, a glavna karakteristika jedinice za napajanje je njena snaga, koja se izražava u vatima (W).

1.4.2. Matična ploča

Glavna ploča je osnovni deo računara koja povezuje sve komponente računara u jednu funkcionalnu celinu. Memorijska kartica, grafička kartica i procesor su instalirani na glavnoj ploči: procesor preko socket utičnice, a RAM memorija preko ekspanzione slot utičnice. Slot 1 utičnica predstavlja fizičku i elekričku specifikaciju konektora za spajanje Intelovih mikroprocesora, uključujući Celeron, Pentium II i Pentium III, za procesore sa jednim i dva jezgra.

Matična ploča (motherboard), ili glavna, ili sistemska ploča povezuje sve komponente računarskog sistema i glavno je strujno kolo u PC računaru koje omogućava komunikaciju između svih delova računara. Više drugih komponenti je spojeno na glavnu ploču direktno ili indirektno. Od matične ploče zavise performanse i tip procesora koji ploča podržava (AMD, Intel). Matična ploča obično sadrži jedan ili više CPU, integrisanih kola (IC) za podršku koja obezbeđuju interfejs između CPU memorije i I/O periferijskih kola, primarne memorije i CMOS i BIOS kola za inicijalno podizanje (butovanje) sistema neposredno posle uključivanja računara. U većini prenosnih i ugrađenih personalnih računara matična ploča sadrži gotovo sve ključne komponente računara, a često može da sadrži jednu ili više periferijskih magistrala i fizičkih konektora za proširivanje hardverskih kapaciteta. Matična ploča može da uključuje integrisane komponente kao što kontroleri za zvuk, grafiku, LAN čip, USB priključak, flopy kontroler i dr. (Slika 1.9a). Blok šema savremene matične ploče prikazana je na slici 1.9b.



Slika 1.9. Sastavne komponente matične ploče i blok šema

1.4.3. Centralna procesorska jedinica

Centralna procesorska jedinica ili CPU je komponenta računara koja izvršava jednostavne programske zadatke koji se nazivaju instrukcije i transformaciju ulaznih podataka u izlazne informacije. Pojedinačne instrukcije izvršavaju male zadatke, npr., „pročitaj sadržaj memorijske lokacije x i dodaj mu broj y“. Većina CPU ima rečnik manji od 1000 različitih instrukcija. Svi računarski programi se sastoje od instrukcija napisanih iz ovog rečnika. Tipični računarski program sadrži milione instrukcija, a CPU može izvršiti milione instrukcija svake sekunde. Kada je program aktivan, brzo izvršavanje instrukcija stvara iluziju neprekidnog kretanja, na isti način na koji filmska tehnika simulira kretanje na bazi sekvenci mirnih slika.

Svaki računar ima barem jedan CPU za interpretaciju i izvršavanje instrukcija u svakom programu za aritmetičke i logičke manipulacije sa podacima i za indirektnu komunikaciju sa svim ostalim delovima računarskog sistema kroz primarnu memoriju (RAM). CPU je izuzetno kompleksan skup elektronskih kola (preko 100 miliona tranzistora i dioda) izgrađenih u jednom silicijumskom čipu koji se uobičajeno naziva mikroprocesor.

IBM kompatibilni PC računari koriste x86 kompatibilne procesore, koje proizvode Intel, AMD, VIA Technologies ili Transmeta. Apple Macintosh računari inicijalno su pravljeni sa Motorola 680x0 familijom procesora, zatim su prešli na PowerPC seriju sa RISC arhitekturom koju su zajedno razvili Apple Computer, IBM i Motorola. Od 2006, Apple je ponovo prešao na x86 kompatiblne procesore. Moderni CPU su opremljeni sa ventilatorom za hlađenje spojenog preko hladnjaka. Tipičan CPU sadrži nekoliko funkcionalnih jedinica: kontrolnu, aritmetičko logičku (ALU), dekoder, magistralu ili BIU (Bus Interface Unit) koja komunicira podatke, adrese i kontrolne informacije i prefetch jedinicu. Ove jedinice rade zajedno na izvršavanju programskih instrukcija, a često se grupišu u izvršnu jedinicu- EU (Execution Unit) koja sadrži kontrolnu, ALU dekodersku i prifeč jedinicu i BIU.

Sve programske instrukcije se skladište u primarnu memoriju (RAM), koja je obično u čipu izvan CPU. Prvi zadatak CPU je da očita instrukcije iz RAM-a. Prefetch jedinica instruiše bus jedinicu da očita instrukcije uskladištene u posebnu memorijsku adresu RAM-a. Ove jedinica uzima nekoliko sledećih instrukcija iz RAM-a da obezbedi da je sledeća instrukcija uvek spremna za CPU. Jedinica za dekodiranje uzima instrukciju koju je prefetch jedinica očitala i prevodi u oblik pogodan za CPU interno procesiranje. To radi istražujući korake potrebne za kompletiranje instrukcije u kontrolnoj jedinici. Posle operacije prefetch i jedinice za dekodovanja vrši se izvršna funkcija u EU jedinici. U toku ovog koraka, različiti delovi CPU tako se povezuju da mogu izvršavati željenu operaciju. Ako se, na primer, zahteva dodatna operacija ALU jedinica se spaja na set ulaza i izlaza. Ulazi sadrže brojeve koje treba dodati, a izlazi – konačnu sumu. Ako neka instrukcija zahteva da se informacija pošalje izvan memorije, npr., upiše u sekundarnu memoriju, onda je poslednja faza izvršavanja instrukcije povratno upisivanje (writeback), u kojem bus jedinica upisuje rezultate izvršene instrukcije natrag u memoriju, ili neki drugi uređaj.

U upotrebi su brojni tipovi CPU, a izbor CPU je važan deo odluke kod izbora računara. Iako ima mnogo razlika u dizajnu CPU, za odluku prosečnog korisnika računara, važna su samo dva faktora: kompatibilnost i brzina CPU.

Sa svakim CPU nisu kompatibilni svi programi, tj. program pisan jedan procesor ne mora raditi sa drugim procesorom. Svaki procesor (CPU) ima ugrađen set mikroinstrukcija – rečnik instrukcija koje procesor može izvršiti. CPU jedinice iz iste familije, generalno su dizajnirane tako da noviji procesor može procesirati sve instrukcije ranijih modela. Ova kompatibilnost CPU naziva se kompatibilnost unazad. Na primer, Pentium 4 procesor je kompatibilan sa prethodnim procesorima: Pentium III i II, Pentium Pro, Pentium 486, 386 i 286, tako da može raditi sa većinom programa pisanih za ove starije tipove CPU. Međutim, programi pisani za PowerPC familiju procesora u Macintosh računarima ne mogu raditi sa Intelovim procesorima, koji su ugrađeni u većinu IBM kompatibilnih PC računara i obrnuto – Intelovi procesori ne mogu čitati programe pisane za PowerPC procesor. Za prevođenje instrukcija programa pisanih za jedan CPU da bi mogli raditi sa drugim nekompatibilnim CPU, koriste se posebne vrste programa, tzv. emulacioni programi (emulatori).

Brzina kojom procesor manipuliše informacijama može veoma varirati. Većina aplikacija zahteva brže procesore, npr., statistički programi, programi za grafički dizajn, igre, pa čak i procesori teksta bolje rade sa bržim procesorima. Brzina računara delom je određena brzinom internog takta – tajmerskog uređaja koji proizvodi električne impulse za sinhronizaciju operacija računara. Brzina takta računara meri se u MHz (megahercima) i GHZ (gigahercima): 1MHz = 106Hz (milion ciklusa u sekundi), a 1GHz = 1000 MHz. Ovu veličinu proizvođači računara obično oglašavaju kao meru brzine rada računara, što nije sasvim tačno. To izgleda kao da brzinu automobila merimo sa brzinom obrtaja motora u minuti. Tako na primer, Celeron CPU od 700MHz nije obavezno brži od Pentium II procesora sa brzinom od 600 MHz, ili PowerPC procesor 4. G sa 500MHz - naprotiv, za neke zadatke je mnogo sporiji. Dakle, brzina takta ne opisuje adekvatno koliko brzo računar može procesirati reči, brojeve ili slike.

Glavne merne jedinice performansi CPU su ***MIPS*** i ***MEGAFLOPS***. *MIPS* označava brzinu takta CPU u milionima instrukcija u sekundi, a predstavljaju ukupan broj instrukcija koje se mogu obaviti u jedinici vremena. Razvoj broja instrukcija u sekundi (IPS/MIPS) i brzine CPU savremenih računara rastao je sa 1 MIPS (1971) -76.383 MIPS (2008). godine. Ova jedinica mere nije toliko od interesa za većinu korisnika, pošto računari obavljaju različite poslove. *Megaflopsi* su mera za milion operacija sa pokretnom decimalnom tačkom koje procesor može da izvrši za jednu sekundu.

1.5. RAČUNARSKE MEMORIJE

Glavni posao CPU je da sledi instrukcije kodirane u programima. Međutim, CPU u isto vreme može manipulisati samo sa jednom instrukcijom i nekoliko podataka. Zato

računar mora imati mesto gde će uskladištiti ostatak programa i podataka dok CPU ne bude spreman za njih. Ovo skladište treba da bude što bliže, ili u samom CPU i da omogućava dovoljno brzo učitavanje i **očitavanje programskih instrukcija i podataka.**

Brzina memorije, ili vreme pristupa CPU jedinice memoriji i očitavanje informacija iz nje zavisi od tipa memorije. Brzina pristupa registru CPU je najveća – praktično trenutna, a keš memoriji nešto manja. Vreme pristupa (*access time*) većini RAM memorija meri se u nanosekundama (1ns=10^{-9} s), a čvrstom disku u – milisekundama (1ms=10^{-6} s), tačnije oko 4-6 ms. Brzina memorije je drugi faktor koji utiče na ukupnu brzinu rada računara. Na slici 1.10 prikazan je odnos kapaciteta i brzine pristupa glavnim tipovima memorija računara.

Slika 1.10. Brzine pristupa i kapaciteti glavnih tipova memorija računara

Savremeni PC računari sadrže četiri osnovna tipa memorije, koje se u osnovi razlikuju po brzini pristupa i kapacitetu: *CPU registri, keš memorije, primarne (RAM) memorije i sekundarne memorije.*

1.5.1. Registri

CPU Registri su privremene lokacije za među rezultate i tekuće instrukcije u samom procesoru, velike brzine pristupa i malog kapaciteta. *Registri* su osnovni memorijski elementi u CPU računara i predstavljaju najbrže memorije kapaciteta od svega nekoliko bajta. Registri se normalno mere brojem bita koje mogu držati, npr., 8-bitni ili 32-bitni registri. Registri se obično implementiraju kao fajl registra, ali se mogu realizovati i hardverski korišćenjem individualnih *flip-flop* digitalnih elektronskih kola, memorije magnetnog jezgra (*core memory*) velike brzine pristupa, memorije tankoslojnog filma (*thin film*) i drugih mehanizama. CPU često sadrži nekoliko vrsta registara, koji se mogu klasifikovati prema sadržaju, ili instrukcijama koje se na njima izvršavaju:

Registri opšte namene- **GPR** (*General purpose registar*) privremeno čuvaju podatke i adrese;

Registri za instrukcije – **IP** (*Instruction pointer registar*), registar posebe namene koji sadrži adresu sledeće instrukcije koju BIU jedinica treba da prenese iz memorije u procesor;

Registar statusa binarne reči procesora – **PSW** (*Processor Status Word*), koji služi da zapamti nekoliko bitnih osobina poslednjeg rezultata dobijenog iz ALU i da zapamti određena setovanja procesora; sadrži tri tipa: *program counter, instruction registers i program status word*;

Korisnički dostupni registri – **UAR** (*User-Accessible Registers*) obuhvataju registre podataka i adresne registre;

Registri podataka – **DR** (*Data Registers*) se koriste za držanja numeričkih vrednosti kao što su celobrojne vrednosti (*Integer*) i vrednosti pokretnog zareza (*floating-point values*); u starijim CPU poznati su kao *akumulatori*;

Adresni registri- AR (*Address Registers***)** memoriše adrese i koriste ga instrukcije koje indirektno pristupaju memoriji;

Uslovni registri - CR (*Conditional registers*) drže tabele istine (*truth values*) koje se koriste za odlučivanje da li neku instrukciju treba ili ne treba izvršiti;

Registar pokretnog zareza - **FPR** (*Floating point registers*) skladišti brojeve pokretnih zareza u većini arhitektura;

Registar konstantnih vrednosti - CR (*Constant registers***)** drži vrednosti koje se samo čitaju kao što su 0, 1 ili *pi* vrednost;

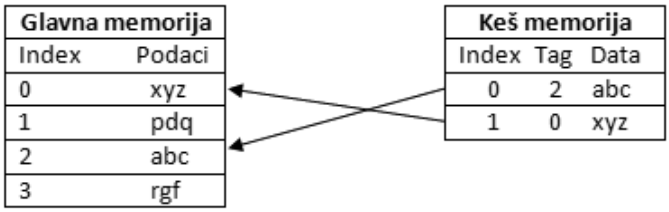
Registri vektora –VR (*Vector registers*) skladišti podatke za vektorsko procesiranje koje vrše SIMD (*Single Instruction Multiple Data*) instrukcije;

Registri koji se odnose na uzimanje informacija iz RAM memorije, čine kolekciju registara za skladištenje, lociranih na posebnim čipovima izvan CPU i generalno nisu arhitekturni registri: *Memory buffer register, Memory data register, Memory address register, Memory Type Range Registers*;

Hardverski registri su slični, ali se nalaze izvan CPU.

1.5.2. Keš memorije

Keš (*cashe****) memorija*** je blok memorija za privremeno skladištenje podataka koji će se ponovo koristiti (Slika 1.11). CPU i HD često koriste keš kao i web pretraživači i web serveri. Keš je sačinjen od skupa ulaza. Svaki ulaz je skup podataka koji su kopija podataka u nekom rezervnom (*backup*) skladištu. Svaki ulaz takođe ima indeks koji specificira identitet podataka u *bekap* skladištu čiji su podaci kopija. Kada se podaci upišu u keš, moraju se upisati i u *bekap* skladište. Sinhronizaciju vremena ovih upisivanja kontroliše politika upisivanja (*write policy*). Keš je nešto većeg kapaciteta, ali manje brzine pristupa od CPU registra i može biti formirana u dva tipa: L1 – interno u procesoru i L2 - brza memorija između CPU i RAM-a, za podatke koje CPU često traži; imaju kapacitet do ≈ 1MB.



Slika 1.11 Dijagram CPU keš memorije

1.5.3. Primarna memorija

Primarna RAM (*Random Access Memory*) memorija sa slučajnim pristupom je najčešća primarna memorija računara, koja privremeno skladišti programske instrukcije i podatke. RAM je poluprovodnička komponenta tipa r*ead/write* za učitavanje i očitavanje podataka i instrukcija programa koji se upravo izvršava. Računar deli svaki čip RAM-a u više memorijskih lokacija iste veličine. Memorijske lokacije RAM-a imaju jedinstvene adrese, tako da im računar (OS, ulazni uređaj) može odvojeno dati instrukciju, kada je primi od programa, da upiše/očita neku informaciju u/iz bilo koje lokacije RAM-a. Otuda dolazi i ime memorije sa slučajnim pristupom". Informacije uskladištene u RAM-u su, u stvari, obrazac električnog naboja koji teče kroz mikroskopska kola silicijumskog čipa. To praktično znači da računar trenutno zaboravi sve što je zapamtio u RAM-u, kada se isključi napajanje. Zato se RAM naziva nestalna ili promenljiva (*volatile)* memorija jer ne čuva permanentno uskladištene informacije. Naravno, ovaj ozbiljan problem računar prevazilazi sa drugim tipovima memorije za skladištenje informacija koje korisnik ne želi izgubiti posle isključivanja računara. Primer ovog tipa memorije su ROM (*Read Only Memory*) čipovi koji sadrže instrukcije za startovanje računara druge kritične informacije, koje se upisuju u procesu proizvodnje čipa i mogu se izmeniti samo izmenom ROM čipa.

RAM memorija može biti *dinamička*, koja zahteva osvežavanje sadržaja tokom rada, ili *statička,* koja zahteva stalno napajanje. Bajtovi memorije imaju jedinstvene adrese koje ih jednoznačno identifikuju i pomažu CPU da prati trag uskladištenih podataka. Zato svaki program kojeg treba izvršiti ili podatak kojeg treba modifikovati, moraju biti upisani u memoriju. Većina računarskih sistema koristi memorijski mapirane ulaze/izlaze gde se informacije za ulaz/izlaz skladište u specijalne oblasti memorije. Na primer, informacije koje treba prikazati na ekranu monitora, upisane su u posebnu oblast memorijskih adresa koje neprekidno skenira video grafička ploča.

1.5.4. Sekundarne memorije

Sekundarna memorija može biti *on line ili interna* masovna memorija sa podacima i programima, odnosno čvrsti disk (HD) kapaciteta ~ 750 GB za savremene PC, čiji se

deo može koristiti za virtualnu memoriju, flopi disk (FD) ili disketa, ili o*ff line (eksterna)* memorija koju čine optički spoljni diskovi i memorijski uređaji (CD, DVD, fleš memorije i dr.).

Masovna memorija (HD) je spojena na glavnu ploču preko kabla i može biti instalirana u kućištu računara ili odvojenom kućištu.

1.5.5. Ostale memorije računara

ROM (*Read Only Memory*) predstavlja stabilnu memoriju i često se naziva BIOS jer se koristi za BIOS čip, koji permanentno memoriše podatke (Slika 2.25). Po pravilu se u ovu BIOS memoriju upisuju *startup* instrukcije i drugi permanentni podaci. Postoji nekoliko tipova ROM memorije: PROM, čiji sadržaj se može upisati samo jedanput; EPROM, koji se može brisati ultraljubičastom svetlošću (najčešći BIOS); EEPROM, koji se može brisati električnim putem i Flash memorija koja dozvoljava višestruki upis podataka (BIOS čip).

CMOS (*Complementary Metal Oxide Semiconductor*) je specijalna vrsta ROM memorije. Troši malu energiju, napaja se baterijom na matičnoj ploči i može pamtiti malo podataka dugi vremenski period. CMOS skladišti osnovne postavke poslednje ispravne konfiguracije, vreme i datum, pasvord za uključivanje napajanja (*Power On*), CMOS pasvord, sekvencu za butovanje diska i tip HD (kod starijih PC), (Slika 2.26). U Macintosh računaru CMOS se naziva *Parametar RAM.*

Za regularno podizanje (*butovanje*) sistema do tačke kada operativni sistem preuzme upravljanje hardverom računara posebno su značajne dve hardverske komponente - CMOS i BIOS. Dakle, CMOS i BIOS nisu isto – ali rade veoma blisko! *BIOS* obezbeđuje pokretanje sistema, izvršava ključne funkcije u podizanju (*start up*) sistema – samotestiranju napajanja komponenti računara - **POST** (*Power On Self Test*) i testira i upoređuje sadržaj sa CMOS-a. CMOS trajno skladišti vreme i datum, sadrži konfiguraciju hardvera i drugih parametara podešavanja.

1.6. OSTALE HARDVERSKE KOMPONENTE CPU RAČUNARA

Grafička kartica (video kartica, grafički adapter, video adapter) procesira i renderuje grafičke izlaze iz računara za displej računara – monitor ili VDU (*visual display unit*) i bitan je deo savremenih računara. Grafičke kartice mogu biti realizovane u različitim oblicima:

Integrisana, ugrađena na matičnu ploču kao grafički memorijski čip;

PCI grafička kartica povezana preko PCI slot utičnice na matičnu ploču;

AGP grafička kartica koja se utiče u namenski AGP port veće brzine od PCI porta za 1x, 2x, 4x i 8x;

PCI Express grafička kartica, koja se utiče u namenski PCI slot, četiri puta brži od AGP8x

Čvrsti disk (Hard Disk - HD) je uređaji za masovno skladištenje programa i podataka ili interna sekundarna memorija. HD skladišti i čuva programe i podatke čak i kada je napajanje isključeno, ali zahteva napajanje za izvršavanje funkcija čitanja i upisivanja u toku upotrebe. Iako su poluprovodničke fleš memorije značajnije pojeftinile, preovlađujuća forma masovnih memorija u PC računarima još uvek je elektromehanički HD.

Slotovi i portovi obezbeđuju lakše povezivanje i dodavanje spoljnih uređaja – *periferijskih* uređaja na računarski sistem. Tako da CPU može komunicirati sa okruženjem i skladištiti informacije za kasniju upotrebu. Bez periferija CPU i primarna memorija izgledaju kao ljudski mozak bez tela. Neki periferijski uređaji, kao tastatura i miš predstavljaju komunikacione linkove između korisnika i računara, dok drugi periferijski uređaji povezuju računar sa drugim mašinama.

Uobičajeni slotovi i portovi u savremenom PC računaru su:

Memorijski slotovi: za prihvat RAM memorije, obično ih ima više vrsta;

PCI (*Peripheral Component Interconnect*) **slotovi**: ISA, EISA konektori za zvučne, TV, mrežne, grafičke kartice, ...;

AGP (*Accelerated Graphics Port***) port**: konektor isključivo namenjen za grafičke karte, karakteriše ga veća brzina od PCI-a;

IDE(*Integrated Drive Electronics*) **konektori**: služe za spajanje **PATA** hard diskova, optičkih uređaja (DVD/CD-ROM/RW); obično postoje dva konektora;

SATA (*Serial Advanced Technology Attachment*) **konektori**: je nešto novijeg datuma nego PATA, služi za konektovanje SATA hard diskova i ima bolje performanse;

USB (*Universal Serial Bus*) **priključci**: služe za priključivanje spoljnih uređaja (printera, memorijskih stikova itd.). Najnoviji standard je USB 3.1;

Nasleđeni konektori: prevaziđeni konektori (serijski i paralelni), koji podržavaju stare uređaje, a sve manje se koriste. Imaju malu brzinu prenosa podataka;

Konektori za periferije: konektori za miš i tastaturu su takođe veoma stari i nisu se previše menjali. Danas se sve više proizvode za USB standardni priključak;

Naponski konektori: preko njega matična ploča dobija napajanje iz naponske jedinice-AC/DC konvertora i raspodeljuje ga ostalim komponentama na ploči.

Umrežavanje računara se uobičajeno vrši u cilju korišćenja bogatih mrežnih, posebno Internet resursa. Glavni hardverski uređaji za računarske komunikacije su: mrežni adapter (*Ethernet*), eksterni modem, interna modemska kartica, *hub*, svičer (skretnica), most, ruter i kablovska infrastruktura. Sve ove hardverske komponente spadaju u kategoriju periferijskih uređaja.

Plug and Play funkcija kod PC računara predstavlja uključivanje ili isključivanje memorija, čipova, dodatnih pločica, *flash* memorija, uređaja itd., kojim se menja konfiguraciju računara, ali ne zahteva posebno podešavanje i setovanje. Ova funkcija po prvi put je viđena na *Apple Macintosh* računarima. Kod starijih računara ovakve akcije su zahtevale izmenu skretnica (*switches*), ili džampera (*jumpers*) .

Perspektivne tehnologije kao što su *EUVL* (*Extreme Ultraviolet Lithography*), *superprovodnici* i *kvantno optički* računari nude znatno bolje performanse budućih računara. *EUVL* je nova laserska tehnologija koja će drastično povećati performanse i smanjiti dimenzije čipova računara, a time i samog računara. *Superprovodnici* provode elektricitet bez zagrevanja i povećavaju brzinu računara za dva reda veličine u odnosu na tradicionalne bakarne provodnike. *Kvantno-optički računari* rade na principima kvantne optike, gde se signali se prenose *fotonima*, a ne električnim impulsima. Brzina ovih računara je neuporedivo veća i približava se brzini svetlosti u transparentnim medijima.

1.7. KLJUČNI TERMINI

Arhitecture (Arhitektura) – konceptualni dizajn i fundamentalna operativna struktura računarskog sistema.

ASCII kod (*American Standard Code for Information Interchange*) – standardni kôd koji ima 256 jedinstvenih znakova i predstavlja svaki karakter sa jedinstvenim 8-bitnim k*ô*dom.

Compatibility (Kompatibilnost) – kompatibilnost (unazad) sistemskih i drugih programa sa prethodnim verzijama iste familije, što omogućava da nove verzije programa podržavaju sva dokumenta napisana u starijim verzijama.

Binarni broj – deo digitalne informacije koji je predstavljen vrednostima sa samo dva stanja – 1 i 0.

Bit – (Binary Digit) je najmanji deo digitalne informacije i može imati vrednost samo dva stanja – 1 i 0.

Byte (Bajt) – jedinica u binarnom brojnom sistemu koja sadrži 8 bita.

CPU (*Central Processing Unit*) – centralna procesorska jedinica ili procesor; glavna komponenta računara koja izvršava jednostavne programske zadatke koji se nazivaju instrukcije i vrše transformaciju ulaznih podataka u izlazne informacije.

Input device (Ulazni uređaj) – interfejsni uređaj između korisnika i računara za unos ulaznih podataka.

Memory (Memorija) – privremena ili trajna skladišta za prihvat instrukcija i podataka programa koji se izvršava (malog kapaciteta) i masovno skladištenje informacija (velikog kapaciteta): CPU registri, keš memorija, RAM i sekundarna memorija (HD, spoljni diskovi, optički diskovi).

Motherboard (Matična ploča) – povezuje komponente računara i omogućava komunikaciju između delova računara; određuje performanse i tip procesora koje podržava i integriše brojne komponente: kontroler za zvuk, grafiku, LAN čip, USB priključak, *flopy* kontroler i dr.

Permanent memory (Permanentna memorija) – stalna memorija koja ne gubi podatke posle ukidanja napajanja (npr. čvrsti disk).

Volatile memory (Nestalna memorija) – memorija koja gubi podatke posle isključivanja napajanja (npr. RAM memorija)

Output device (Izlazni uređaj) – uređaji na izlazu računarskog sistema, kao što su monitor, zvučnik, printer, modemi i dr.

Periphery (Periferijski uređaj) – svi uređaji računarskog sistema izvan CPU.

Port (Port) – konektor za ulazno izlazne uređaje računarskog sistema, mogu biti serijski i paralelni.

Slot (Slot)– standardizovani konektori za zvučne, grafičke, mrežne i druge kartice.

Procesor – CPU računara koji se najčešće naziva procesor računara.

RAM (Random Access Memory) – nestalna memorija sa slučajnim pristupom koja privremeno skladišti podatke i instrukcije programa koji se izvršava tipičnog kapaciteta ~ 256, 512, 1024 MB.

ROM (Read Only Memory) – stabilna memorija koja sadrži kodove za butovanje na najnižem nivou, može se ažurirati sa "flashing" tehnikom; proverava da li sve komponente glavne ploče rade zajedno, obezbeđuje ulaze/izlaze na niskom nivou za OS i butuje računar do tačke gde OS preuzima.

Pitanja za ponavljanje

1. Navedite ulazne uređaje računara.
2. Koje uređaje obuhvata hardver unutar računara?
3. Kako se naziva grupa od 8 bita?
4. Koliko različitih podataka se može kôdovati sa 4 bita?
5. Napišite binarni zapis decimalnog broja 5?
6. Šta se kodira sa ASCII kodom i koliko ima karaktera?
7. Koji je heksadecimalni zapis binarnog broja 01010001b?
8. Koji je binarni zapis heksadecimalnog broja 42h?
9. Navedite osnovne delove funkcije i pravilne korake CPU?
10. U čemu se izražavaju performanse procesora?
11. Šta sadrže BIOS i CMOS i čemu služe?
12. Navedite karakteristike RAM memorije?
13. Navedite osnovne karakteristike ROM memorije?
14. Navedite glavne tipove i karakteristike keš memorija.
15. Šta označava *Plug and Play* tehnologija?

2. PERIFERIJSKI UREĐAJI

Razumevanjem sadržaja navedenog u ovom poglavlju studenti će biti u mogućnosti da navedu nekoliko primera ulaznih i izlaznih periferijskih uređaja, kao i da objasne kako isti omogućavaju korisniku lakši unos/dobijanje različitih tipova informacija u/iz računara. Takođe će moći moći i da objasne zašto klasični računar ima više različitih vrsta uređaja za skladištenje i kako sve komponente računaskog sistema funksionišu zajedno.

2.1. UVOD

Uspeh Apple II računara dolazi od činjenice da je računar pored procesora i memorije uključivao tastaturu, monitor, i disk i drajver za traku za sekundarno skladištenje podataka. Načini procesiranja informacija u savremenom računaru za običnog korisnika su skriveni; korisnik samo vidi ulaze i izlaze (I/O) računarskog sistema.

Ulazni uređaji su interfejs između korisnika i računarskog sistema. Tipično obuhvataju tastaturu, uređaje za pokazivanje (*pointing devices*), uređaje za očitavanje, skenere, audio i video uređaje i senzorske uređaje. Računari mogu primati ulazne signale iz različitih izvora, uključujući mašinsku opremu za proizvodnju, telefone, komunikacione mreže i druge računare. Kako se razvija tehnologija i menjaju ljudske potrebe, razvijaju se i novi ulazni uređaji za računar.

Računar može uraditi neverovatne stvari, ali za čoveka sve to ništa ne vredi dok ne dobije rezultat koji mu treba. „Po pravilu čovek više brine o onome što ne može videti nego što može videti“ (J. Cezar). Izlazni uređaji računarskog sistema konvertuju interne obrasce bita iz računara u oblik koji čovek može razumeti. [1]

2.2. ULAZNI UREĐAJI

Tastatura (*keyboard***)** je najčešći oblik ulaznog uređaja. Koristi se za unos slova, brojeva i specijalnih znakova. Nekada se unos podataka u računar vršio putem prekidača, čime se unosio po jedan bit sa svakom manipulacijom prekidača.

Savremeni korisnici računara imaju veliki izbor tastatura na raspolaganju. Tastatura računara se proizvodi u više različitih tipova, a najčešće se dele u sledeće kategorije:

- *Standardna tastatura*, koja ima QWERTY raspored slova klasične pisaće mašine;
- *Ergonomska tastatura*, sa rasporedom slova i znakova pod uglom, što smanjuje zamor ruku i zglobova i sprečava moguće zdravstvene probleme;

- *Bežična tastatura* (*wireless*) koristi radio talase za povezivanje na računar;
- *IC tastatura* koja koristi infracrveno zračenje za povezivanje na računar;
- *Rasklapajuća tastatura*, namenjena za mobilni rad, a koristi se za palm-računare;
- *Jednoruka tastatura* (*half*) namenjena za hendikepirane osobe;
- *Tastatura štampane na membrani* i
- *Holografska tastatura* (u razvoju), (Slika 2.1).

Slika 2.1. Neki od tipova tastastatura: ergonomska, jednoručna, standardna i holografska [1]

Standardni QWERTY tip tastature, nasleđen od mehaničke pisaće mašine, uprkos univerzalnoj prihvatljivosti i standardizaciji, nisu najbolji ulazni uređaji za savremene računare, iako su još standardna oprema gotovo svakog savremenog računara. Kucanje na standardnoj tastaturi sa pravolinijski poređanim tipkama zahteva neprirodno držanje ruku i ručnih zglobova, što u dužem vremenskom periodu može izazvati profesionalno oboljenje - RSI (*Repetitive stressinjures*) – povrede od ponovljenih stresova kao što su *tendonitus i carpal tuned sindrom. Ergonomske tastature* imaju tipke postavljene pod uglom, što olakšava kucanje i smanjuje profesionalna oboljenja. Standardna i ergonomska tastatura povezane su na računar žičnim kablom kroz koji šalju signal na ulazni port računara. *Bežična tastatura* šalje signal radio emisijom ili putem infracrvenih zraka, slično daljinskoj komandi TV prijemnika. Pored ovih, najčešće zastupljenih tastatura postoje specijalizovani i namenski tipovi, kao što su jednoručna, minijaturna rasklapajuća, membranska i holografska.

Uređaji za pokazivanje (*Pointing Devices*) obuhvataju razne tipove pokazivačkih uređaja kao što su: miš, *touchpad (*podloga osetljiva na dodir*), pointing stick* (pokazivački štapić)*, trackball* (kuglica za praćenje)*, joystick* (upravljačka ručica)*, graphics tablet* (grafička tabla) *i touch screen* (ekran osetljiv na dodir). Svi pokazivački uređaji su neefikasni za unose velike količine podataka. [1] Korisnici računara uglavnom koriste tastature za unos teksta i numeričkih podataka. Za druge funkcije, kao što su slanje komandi i pozicioniranje kursora tipično koriste *miš*. Miš je dizajniran za pomeranje pokazivača po ekranu i usmeravanje na specifične karaktere i objekte. Danas postiji više tipova miševa a proizvode se u većini slučajeva sa USB priključkom, dok su postoji i starija verzija koja se povezuje sa računarom uz pomoć PS/2 priključka. U periodu pre toga miš se povezivao serijskim priključkom.

Tipovi/vrste miševa:

- Miš sa kuglom
- Optički miš
- Bežični miš
  - RF miš
  - Bluetooth miš
- Laserski miš
- Biometrički miš

Miš sa kuglom je zastareo i prevaziđen , danas se najviše koriste optički miševi dok se laserski polako ali uspešno sve više i više probijaju na tržište. *Optički miš* koristi reflektovanu svetlost LED ili laserske diode i napredni optoelektronski senzor za detektovanje kretanja. Savremeni miš ima jedan dva, tri ili pet tastera za slanje signala računaru tipa: *„izvrši ovu komandu“*, *aktiviraj selektovani alat“* i *„selektuj sav tekst između ove dve tačke“*. Miš je standardni deo opreme desktop računara, ali je nepraktičan za prenosne računare.

Touchpad: mali ravni panel osetljivi na lagan pritisak. Korisnik pomera kursor povlačenjem prsta preko panela.

Pointing stick: tanko dugme slično gumici za brisanje na olovci, osetljivo na pritisak. Marker se kreće u pravcu pomeranja pritiska prsta (slično minijaturnom džoistiku). Obično se nalazi u sredini tastature Laptop računara.

Trackball: pokazivač funkcioniše kao obrnut miš koji ostaje miran, dok se pomera kuglica koja kontroliše kretanje kursora na ekranu.

Džoistik: upravljač sličan kontroli video igara ili menjaču automobila.

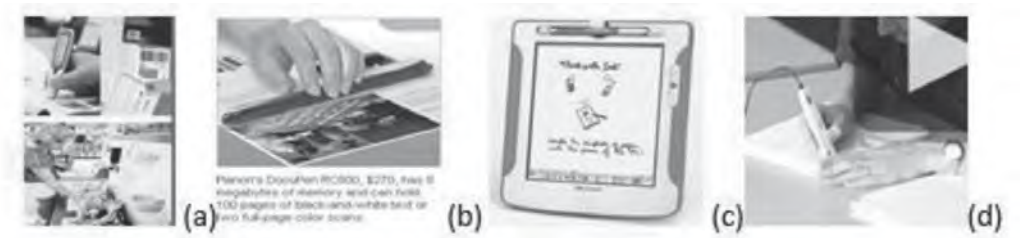
Grafička tabla popularna je kod umetnika i dizajnera; većina je osetljiva na pritisak, tako da šalje različite signale, zavisno od pritiska štapića (stila) koji izvršava klasičnu funkciju miša - *usmeri i klikni.*

Tuch screen: fotootporna površina koja odgovara na dodir prsta korisnika na različite oblasti ekrana. Ovi uređaji se istovremeno koriste kao ulazni i izlazni uređaji. Često se koriste u javnim ustanovama, aerodromima i sl. gde većina korisnika nije familijarna sa računarima. Takođe se koriste u priručnim i PDA uređajima. U nekim rešenjima se za pokazivanje i upisivanje podataka mora koristiti štapić (stilo).

Uređaji za pokazivanje nisu efikasni za unos velike količine teksta u računar, što je razlog zašto nisu zamenili standardne tastature. Alternativu kucanju na tastaturi pružaju specifični uređaji dizajnirani da prepoznaju određene oblike, karaktere, oznake i kodove i unose ih na ulaz računarskog sistema.

Uređaji za očitavanje podataka čitaju oznake koje prezentuju kodove, posebno dizajnirane za računarski unos podataka, (Slika 2.2): (a) optički čitač i POS terminal, (b) olovka (pen) skener, (c) *tablet* PC, (d) čitač znakova napisanih magnetnim mastilom Uređaji

za očitavanje uključuju: optičke čitače, čitače znakova napisanih magnetnim mastilom, čitače bar-kodova ili POS (*point-of-sale*) terminale, *Pen scanners* (priručne skenere), t*ablet PC (MS Win XP Tablet PC*), interaktivne pametne table (*Smart whiteboard,* 1991) i čitače radio identifikacionih kodova (*RIF reader*).



Slika 2.2. Uređaji za očitavanje [1]

Optički čitač oznaka koristi reflektovanu svetlost da odredi lokaciju markera na standardnom tekstu, tabeli i Slika Čitač karaktera pisanih magnetnim mastilom očitava karaktere neobičnog oblika napisane magnetnim mastilom na otpornoj površini sa slabim strujnim tokom. Čitač bar kodova koristi svetlost za čitanje univerzalnih kodova komercijalnih proizvoda, kodova inventara i drugih kodova kreiranih od obrazaca barova različite širine. Ovi čitači se standardno nalaze na prodajnim mestima (POS terminali), skeniraju bar kod informaciju i šalju je u *mainframe* računar, koji određuje cenu proizvoda, porez, ukupan iznos i registruje transakciju za kasnije potrebe inventarisanja, obračuna i revizije. Kako su oblici slova teksta, karakteri magnetnog mastila i bar kodovi dizajnirani za očitavanje sa računarom, ovi su uređaji ekstremno precizni.

Međutim, čitanje teksta iz knjiga sa velikom razlikom štampanih slova znatno je složenije Tehnologija za prepoznavanje individualnih karaktera na štampanim stranicama naziva se optičko prepoznavanje karaktera – OCR (*Optical Caracter Recognition*). Izlazni podaci iz OCR uređaja mogu se skladištiti i uređivati u računaru. Pre nego što računar može prepoznati rukopis ili štampan tekst, prvo mora kreirati digitalnu sliku stranice koju zatim može uskladištiti u memoriju. Ova se operacija obično izvršava uređajem koji se naziva skener. Skeneri prihvataju i digitalizuju štampane slike i druge štampane materijale u bit mapirane slike. Postoje brojni tipovi skenera koji u stvari ne čitaju niti prepoznaju slova i brojeve na skeniranoj stranici – oni samo prave digitalnu sliku te stranice, koja je kompatibilna za računarsko procesiranje. Zatim, računar koristi OCR softver za interpretaciju crno belih skeniranih obrazaca kao slova i brojeve.

Na raspolaganju je nekoliko specijalnih skenera koji direktno izvršavaju OCR funkciju. *Pen skener* izgleda kao svetlosni pokazivač, ali su, u stvari, bežični skeneri koji prepoznaju karaktere beskontaktnim prelaskom skenera preko skenirane stranice (površine). Prevlačenjem *Pen skenera* preko linija štampanog teksta, on kreira fajl teksta u ugrađenoj memoriji, gde se skladišti do prebacivanja u računar preko kabla ili IC zraka. Ovaj skener sadrži mali ugrađeni računar programiran za prepoznavanje teksta.

Prepoznavanje rukopisa je daleko teže i sa više grešaka od prepoznavanja štampanih karaktera, ali ima daleko više praktičnih aplikacija, posebno u pen- ugrađenim računarima koji nemaju tastaturu, a ulaze primaju preko štapića (*stilusa*) direktno povezanog na ravni panel displeja. Računar elektronski simulira olovku i papir, program za prepoznavanje rukopisa prevodi oblike slova rukopisa korisnika u ASCII karaktere. Većina ovih sistema zahteva da korisnik modifikuje svoj rukopis da bi se povećala tačnost prepoznavanja i dekodiranja. PDA (*Personal Digital Assistant*), džepni računar sa olovkom (pen) služi kao lični organizator, notes, knjiga za sastanke im uređaj za komunikacije. Ovaj uređaj, takođe, može programirati kodove za brojne praktične primene.

Program OCR može se primeniti i u *beloj smart tabli* u sobi za sastanke i prezentacije, koja služi kao ulazni uređaj za PC računar. Tabla ispisana rukopisom predavača prenosi se i skladišti na HD u PC računaru kao digitalizovana slika. Ako je rukopis jasan i korektna, OCR program ga može automatski prevesti u standardni štampani tekst i poslati e-poštom na udaljenu lokaciju. Generalno, broj periferijskih jedinica zavisi od broja korisnika.

2.3. DIGITALIZACIJA I SKENERI

Čovek živi u analognom svetu, gde vidimo i čujemo glatke obrise slika i neprekidne tonove. Računar skladišti sve informacije kao binarne veličine **1** i **0**. Da bi se analogne informacije uskladištile u računar potrebno ih je *digitalizovati* – pretvoriti ih iz analognog u digitalni oblik. Digitalizacija zahteva upotrebu nekog ulaznog uređaja, kao što je desktop ravni skener ili audio CD, koji uzimaju milione uzoraka analognog originala. Vrednost uzorka (može se predstaviti numerički (sa 1 i 0) i tako uskladištiti u računar. Originalne slike i ton mogu se rekonstruisati skupljanjem svih uzoraka u sekvenci.

Dakle, pre nego što računar može prepoznati rukopis ili štampani tekst, skener ili drugi ulazni uređaj mora *digitalizovati informacije* – konvertovati analognu informaciju u digitalni oblik.

Skener je ulazni uređaj koji može kreirati digitalnu prezentaciju štampane stranice/slike. Tipičan digitalni skener sadrži CCD video senzor, sličan senzoru u digitalnim video kamerama. CD senzor se kreće napred-nazad preko analogne skenirane površine, snimajući za svaki uzorak slike intenzitete crvene, zelene, plave (R,G,B) boje u svakoj tački (pikselu) slike. Ljudsko oko ima receptore za R,G,B boje, a sve druge boje se vide kao kombinacija ove tri. Jedan bajt uobičajeno predstavlja jednu boju, a kod od 3 bajta (24 bita) predstavlja boju za svaki uzorak. Skener šalje svaki digitalni kod u računar, gde se binarni podaci mogu skladištiti i procesirati. Skeneri se dele se u tri glavne kategorije: *ravne skenere, skenere negativa i slajdova, portabl skenere i doboš skenere*. Najčešći model je *ravni skener* (*Flatbed*), sličan kopir mašini i slično radi, osim što umesto papirne kopije kreira računarski fajl. Namenjeni su za individualnu kućnu upotrebu i male kancelarije. Imaju tipičnu rezoluciju od 3600 dpi, a dubinu boje od 48 bita, (Slika 2.3).

Skuplji modeli koje koriste profesionalci grafičari, mogu proizvesti reprodukciju većeg kvaliteta, a imaju i dodatke za skeniranje fotografskih negativa i slajdova. Neki skeneri se zovu *slajd skeneri* i mogu skenirati samo slajdove i negative filmova.

Slika 2.3. Ravni skener [2]

Skeneri negativa i slajdova su specijalna vrsta skenera namenjena samo za skeniranje transparentnih negativa i slajdova. *Portabl skeneri* (*Sheet-fed scanners*), bazirani na *CCD* prostornom senzoru *(Charged Cupled Devices Array*) namenjeni su za skeniranje pisanog materijala na terenu i to po jednu stranicu koja se uvlači u kućište skenera. *Dobošni skeneri* (*Dram scanners*), su skupi i velikog kapaciteta. Koriste za skeniranje tehnologiju optoelektronskih fotomultiplikativnih pojačavačkih cevi (*PMT*). Namenjeni su za izdavačke kuće gde je kritičan kvalitet slika. Svi skeneri u suštini konvertuju fotografije, crteže, dijagrame i druge štampane informacije u obrazac bita ili bit mapirane slike, koje se mogu uskladištiti u memoriju računara, obično upotrebom programa za računarsku grafiku.

Digitalna fotokamera je ulazni uređaj koji SD memoriju za skladištenje digitalnog signala fotosnimka, umesto na film. Može snimati sve što i klasičnim fotoaparat. Digitalna foto kamera digitalnu mirnu sliku ili kratke sekvence video snimka sa izlaza fotoosetljivog digitalnog senzora (CCD-*Charge Coupled Devices*), skladišti u fleš memoriju u formi obrasca bita. Ove se informacije kasnije mogu prebaciti u računar, po potrebi obrađivati i gledati na ekranu monitora računara.

Digitalna video kamera može snimati sve što i analogna, klasična filmska kamera, ili analogna video kamera samo što umesto na filmsku traku ili magnetnu traku snima sliku kao obrazac bita i skladišti na disk ili drugi medijum za skladištenje. Digitalna video kamera ne zahteva video digitalizator i šalje video signal već digitalizovan u računar. Upotrebom web video kamera, (Slika 2.4), koje ne zahtevaju video digitalizatore omogućavaju se servisi *videokonferencije,* gde se video slike prenose kroz računarsku mrežu, tako da se ljudi na različitim fizičkim lokacijama mogu čuti i videti i održavati telekonferencije i poslovne sastanke.

Slika 2.4. Digitalna video (web) kamera [1]

Video digitalizator je kolekcija elektronskih digitalnih kola koja primaju kao ulazne signale izlazne signale iz analogne video kamere, video kasetnog rekordera, TV ili drugog izvora analognog video signala i konvertuje ih u digitalne signale koji se može uskladištiti u memoriju i prikazati na ekranu monitora računara. Kao ulazni uređaj računarskog sistema, prihvata analogni video izlaz sa analogne video kamere, VCR (video kasetnog rekordera) ili TV prijemnika. Ovi uređaji vrše konverziju analognih video signala u digitalne video signale, koji se zapisuju u memoriju i prikazuju na ekranu monitora računara. Digitalni video se koristi za multimedijske aplikacije kao što su web stranice i razvoj audio CD ROM-ova. Funkciju video digitalizatora vrši video kartica u računaru.

Audio digitalizator je ulazni uređaj koji sadrži kola za digitalizaciju analognih audio signala sa mikrofona, diktafona i ostalih ulaznih audio uređaja u digitalne audio signale. Audio digitalizatori i digitalni audio su postali deo svakodnevnog života. Popularni CD plejer je u stvari računarski sistem dizajniran za prevođenje digitalne informacije na CD ROM-u u analogne audio signale koji se mogu pojačavati i reprodukovati na zvučniku/slušalici. Kod snimanja digitalnog audia pomoću računara, analogni zvučni talasi vibriraju membranu mikrofona povezanog na mikrofonsku utičnicu na zadnjoj strani panela računara. Analogni audio signal sa mikrofona se dovodi na zvučnu karticu u PCI slotu na matičnoj ploči, koja vrši funkciju audio digitalizatora u računaru. Pozicije mikrofonske membrane audio digitalizator, odnosno, zvučna kartica uzorkuje oko 44.000 puta u sekundi, odnosno sa frekvencijom od 44 kHz, a nivoi uzoraka se skladište kao binarni brojevi. Što je veća frekvencija uzorkovanja bolji je tonski snimak: jedan 8-bitni uzorak može predstaviti 256 različitih nivoa tona, a 16-bitni – 65.536 nivoa. Digitalizacija zvuka ili slike sa većom frekvencijom uzorkovanja i veću vernost reprodukcije, zahteva veći kapacitet medija za skladištenje.

Digitalni audio signali se mogu memorisati i naknadno procesirati specijalizovanim programskim alatima. Audio digitalizator može digitalizovati govorni signal, muziku i druge zvučne efekte, koji su obični binarni podaci za računar.

Program za prepoznavanje govora vrši konverziju analognog govornog signala u digitalni oblik. Digitalni govorni signali se, zatim, mogu editovati, memorisati i štampati. Ovaj program nalazi veliku primenu u medicini, vojnim programima, obuci kontrolora leta, *kol* centrima, za hendikepirana lica, za unos teksta u računar čitanjem, u robotici,

za kućnu automatiku (akustomate), radu sa računarom bez ruku, mobilnoj telefoniji itd. Program za prepoznavanje govora može konvertovati podatke glasa u reči koje se mogu skladištiti i štampati. Ovakav program je na raspolaganju više godina, ali nije bio dovoljno pouzdan za širu primenu. Čak i najnoviji programi za prepoznavanje govora imaju još uvek neka ograničenja da bi zamenili tastaturu kao ulazni uređaj.

Senzori su dizajnirani u cilju monitorisanja fizičkog životnog okruženja i radnih uslova u opasnim i za čoveka nepristupačnim sredinama. Senzori se koriste za neprekidno merenje i kontrolisanje temperature, vlažnosti, pritiska, zagađenosti vazduha, prisustva opasnih hemikalija, radioaktivnosti i Slika Senzori obezbeđuju podatke za potrebe brojnih disciplina: robotike, kontrole klime okoline, vremenske prognoze, medicinskog monitoringa, naučnih istraživanja itd. Čak se i ljudski osećaj mirisa može simulirati sa senzorima. Treba očekivati da se uskoro takvi uređaji koriste za detekciju pokvarenih namirnica u prodavnicama hrane, nagaznih mina u minskom polju, curenja opasnih hemikalija i slično.

2.4. IZLAZNI UREĐAJI RAČUNARSKOG SISTEMA

Većina savremenih računara daje izlazne rezultate kroz dva glavna tipa uređaja: *ekran monitora* za neposredni vizuelni pregled i printere za permanentni papirni izlaz.

Video izlazni uređaj računarskog sistema obuhvata tri ključne komponente: monitor, video adapter i video memoriju (VRAM). *Monitor* (video displej) koristi se kao jednosmerni prozor između korisnika i mašine. Savremeni monitor podjednako prikazuje numeričke i alfabetske karaktere, grafiku, fotografske slike, animaciju i video snimke. *Video adapter* ili video (grafička) kartica, integrisana ili preko PCI slota povezana na matičnu ploču, povezuje monitor sa računarom. VRAM ili video memorija je poseban deo RAM-a u kome se drže digitalne video slike. Što je više video memorije, to je moguće prikazati više detalja na jednoj slici na ekranu monitora računara.

Prvi monitori računarskog sistema bili su sa zelenim ekranom (1970-ih), zatim sa kolor grafičkim adapterom CGA (*Color Graphics Adapter*), sa 4 boje i rezolucijom 320x200 piksela (1981). Monitori sa poboljšanim grafičkim adapterom - EGA (*Enhanced Graphics Adapter*), sa 16 boja i rezolucijom od 640x350 piksela pojavili su se 1984. Prvi VGA (*Video Graphics Array*) monitori pojavili su se 1987, zatim XGA (*Extended Graphics Array*) monitori sa ravnim ekranom, 16.8 miliona boja i rezolucijom od 1024x768 piksela (1990). Savremeni monitori sa LCD panelom imaju u većini slučajeva 16.8 miliona boja i full HD (1920x1080 piksela) rezoluciju.

Konektori za povezivanje monitora na računar proizvode se u tri glavna tipa: VGA (*Video Graphics Addaptor*),DVI (*Digital Video Interface*) i HDMI (*High definition multimedia interface*) [3]. VGA konektori su starije generacije, dok su DVI i HDMI razvijeni da omoguće veliku rezoluciju na digitalnim ravnim ekranima.

Kako uloga monitora kao grafičkog izlaznog uređaja svakim danom raste, potrebno je da korisnici računara treba da znaju osnovne podatke o faktorima koji kontrolišu veličinu i kvalitet slike. Osnovne karakteristike monitora i kvaliteta slike su veličina ekrana, rezolucija i kvalitet slike (dubina boje). Veličina monitora, kao i TV ekrana, meri se dužinom dijagonale linije ekrana, tipično 19-24 inč, ali je stvarna vidljiva oblast obično manja. Slika na ekranu monitora sastoji se sitnih tačaka, koje se nazivaju ***pikseli*** (p*icture elements*) – elementi slike. Kvadratni inč neke slike monitora je tipično mreža piksela od oko 72x72 piksela. Kaže se da takav monitor ima rezoluciju od 72 tačke po inču ili – **dpi** (*dot per inch*). Što je rezolucija veća ovi pikseli su sve bliži jedan drugom. Tipična rezolucija od, na primer, 1024 x 768 sadrži 786.432 piksela.

Rezolucija nije jedini faktor koji određuje kvalitet slike. Kvalitet slike zavisi od rezolucije, ali i tzv. dubine boja, ili bitske dubine, što znači da veći opseg boja po pikselu zahteva više bita prostora u video memoriji. Dubinu boje (*color depth*) određuje broj različitih boja koje monitor može da prikaže u jednom trenutku, a izražava se u broju bita po pikselu. Što je veća bitska dubina to monitor može prikazati više nijansi boja po svakom pikselu.

Slika 2.5. Uticaj dubine boje na kvalitet slike [1]

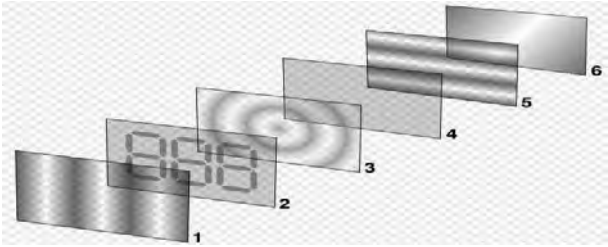
Ako se svaki piksel predstavlja sa 8 bita memorije, rezultirajuća slika može imati do 256 različitih boja na ekranu u istom trenutku (2^8=256). Drugim rečima, 8-bitni kolor, uobičajen kod starijih PC, ima dubinu boje od 256. Većina grafičkih profesionalaca koristi 24-bitni kolor monitor koji dopušta 2^{24}≈ 16 miliona boja po pikselu, što je više nego dovoljno za realističke slike. Uticaj dubine boje na kvalitet slike prikazan je na slici (Slika 2.5). Monohromatski monitor može prikazati samo monohromatske slike. Monitori sa sivom skalom mogu prikazati crno-belu sliku sa nijansama sivog i imaju veću dubinu boje od monohromatskih.

Savremeni PC i Macintosh monitori mogu na istom monitoru prikazati različite kombinacije rezolucije i dubine boje. Monitor se na računar spaja preko video adaptera ili grafičke (video) ekspanzione kartice, koja se utiče u PCI slot na matičnoj ploči. Svaka slika prikazana na ekranu monitora postoji u video memoriji (VRAM), posebnom delu RAM memorije, rezervisanom za čuvanje video slika. Veličinu VRAM-a određuju maksimalna rezolucija i dubina boje, koju računarski sistem može prikazati. Što je veća video memorija računar može prikazati više detalja u slici. Sa tehnološkog aspekta video displej

izlazni uređaji računarskog sistema dele se u četiri osnovne klase: CRT (*Catode Ray Tube*) monitore, LCD/TFT monitore, projekcione LCD panele i video projektore.

1. CRT (Catode Ray Tube) monitori su zasnovani na katodnim cevima. Ovi monitori su monitori stare generacije i u danas se veoma retko koriste. CRT je vakumska cev koja sadrži elektronski top kao izvor elektrona i fluorescentni ekran, sa internim i eksternim sredstvima za ubrzavanje i otklon elektronskog snopa, koji se koristi za formiranje emitovane svetlosti sa fluorescentnog ekrana. CRT se koristi u računarskim monitorima, TV prijemnicima, osciloskopima, radarima i sl. Glavne sastavne komponente kolor su: 1. Elektronski top; 2. Elektronski snop; 3. Namotaji za fokusiranje; 4. Namotaji za otklon; 5. Spajanje anode; 6. Maska za odvajanje snopa za crveni, zeleni i plavi deo prikazivane slike; 7. Fosforni sloj sa crvenim, zelenim i plavim elementima; 8. Presek fosfornog sloja na unutrašnjoj strani ekrana.

2. LCD (Liquid Crystal Displays) monitori je savremeni, popularni monitor sa ravnim ekranom na bazi tečnih kristala. Sam LCD je elektrooptički modulator realizovan kao tanki, ravni displej napravljen od nekog broja kolor ili monohromatskih piksela postavljenih ispred izvora svetlosti ili reflektora. Obično se koristi u baterijski napajanim uređajima jer zahteva malu snagu napajanja. Svaki piksel LCD tipično sastoji se od jednog sloja molekula postavljenog između dve transparentne elektrode i dva polarizaciona filtera, sa okomitom osom transmisije. Bez tečnog kristala između dva polarizaciona filtra svetlost koja prođe prvi filter bude blokirana na drugom filteru. Površine elektroda koje su u kontaktu sa materijalom tečnog kristala, tako su tretirane da poravnavaju molekule tečnog kristala u određenom pravcu. Ovaj tretman sastoji se od tankog polimerskog sloja koji se trljaju u jednom pravcu, korišćenjem, na primer, neke tkanine. Tako se pravac tečnih kristala poravna i definiše sa pravcem trljanja. Elektrode su sačinjene od transparentnih provodnika od *tankog indium oksida* – ITO (*Indium Tin Oxide*). Struktura refleksnog LCD displeja prikazana je na slici (Slika 2.6), gde je: 1. Polarizacioni filterski film sa vertikalnom osom za polarizaciju ulazne svetlosti; 2. Stakleni substrat sa ITO elektrodama, čiji oblik određuje formu kristala kada se uključi napajanje LCD; 3. Tečni kristal sa međusobno okomitim osama polarizacije molekula (*twisted nematic device*); 4. Stakleni nosilac sa filmom ITO elektrode sa horizontalnim ivicama za poravnavanje sa horizontalnim filterom; 5. Film polarizacionog filtera sa horizontalnom osom polarizacije za blokiranje prolaska svetlosti; 6. Refleksna površina za odbijanje svetlosti prema posmatraču.



Slika 2.6. Struktura LCD displeja [1]

Pre primene električnog polja, orijentacija molekula tečnog kristala određena je njihovim podešavanjem dovođenjem osa molekula u međusobno okomit položaj na površini. U ovom tipu LCD, molekule se aranžiraju u spiralnu strukturu, odnosno uvrću se (*twested*). Pošto materijal tečnog kristala deli svetlost, posle prolaska kroz jedan filter svetlost se rotira (menja osu polarizacije) sa spiralnom strukturom sloja tečnog kristala, omogućavajući joj prolaz kroz drugi polarizacioni filter. Polovina ulazne svetlosti se apsorbuje u prvom polarizacionom filteru, dok je ostatak strukture sasvim transparentan.

3. TFT monitori koriste tehnologiju tankog filma tranzistora za LCD displej, gde svakom pikselu odgovara jedna tranzistor. Tipičan 17 inča TFT monitor ima oko 1,3 miliona piksela slike, odnosno 1,3 miliona tranzistora. Takav monitor tipično može imati do 11 mrtvih piksela, odnosno tranzistora koji ne rade, a koji se na ekranu vide kao crvena, bela ili plava tačka i obično nisu kritične ako se ne nalaze u kritičnoj zoni ekrana. Ovi monitori imaju kraće vreme odziva i pogodni su za multimedijalne sadržaje.

4. Projekcioni LCD paneli (*Overhead projection panels*) projektuju video signale, ili računarske podatke na visećem zidnom ekranu. Ranih 1980s-1990's, LCD projektori tipa epidiaskopa (*overhead projectors*) dominirali su u školama i poslovnim prezentacijama. LCD panel u plastičnom okviru montiran na zidu i spojen na video izlaz računara, često razdvojen od normalnog monitorskog izlaza. Rashladni ventilator u panelu je sprečavao pregrevanje panela, koje je dovodilo do zamagljivanja slike. Prvi LCD paneli bili su monohromatski i za NTSC standardni video signal, kao što je iz *Apple* II računara ili VCR (*Video Recorder*). Kasnih 1980-tih pojavili su se 16-bitni kolor modeli za *Macintosh* i VGA PC računare. Displej je bio prilično inertan, sporo se osvežavao, pa su brzo pokretne slike ostavljale trag.

LCD projekcioni paneli su savremena zamena analognih diaskop, grafoskop i episkom projektora. Za prikazivanje slike, LCD projektor tipično šalje svetlost iz *metal-halidne lampe* kroz prizmu koja deli svetlost na tri poli-silicijumska panela, za crvenu, zelenu i plavu komponentu video signala. Kada polarizaciona svetlost prolazi kroz panel (kombinacija svetlosnog polarizatora, LCD panela i analizatora), individualni pikseli mogu biti otvoreni za propuštanje svetlosti, ili zatvoreni za blokiranje svetlosti. Kombinacija otvorenih i zatvorenih piksela može proizvesti širok opseg boja i nijansi sivog u projektovanoj slici. Metal-halidne lampe emituju snažnu usmerenu svetlost (2000-4000 lumena), ali su skupe i traju od 500-1000 **časova. Zahvaljujući ovoj lampi, o**vi projektori su manji i lakši, ali daju najbolji kvalitet slike na čisto beloj ili sivoj površini, ili namenskom ekranu.

5. *Video projektor* uzima video signal *video signal* i projektuje odgovarajuće slike na projekcionom ekranu koristeći kombinaciju sočiva. Svi video projektori koriste vrlo jak izvor svetlosti za projekciju slike, a savremeni projektori imaju mogućnost manuelne korekcije svetlosti, kontrasta i drugih parametara. Video projektori se često nazivaju *digitalni projektori*. Parametri koji određuju kvalitet i cenu video projektora su jačina svetlosti (od 1500-4000 lumena), koja je u određenom projektoru fiksna i veličina projektovane slike, pošto povećanje slike za 41% , smanjuje intenzitet svetla za 50%.

Štampač proizvodi *papirni izlaz* računara - čvrstu kopiju (*hard copy*) na papiru svake statičke informacije koja se može prikazati na ekranu monitora računara, koji prikazuje trenutan ili privremen izlaz računara na monitoru.

Stari modeli štampača su štampači sa dodirom i oni se svrstavaju u dve kategorije a to su linijski i matrični štampači. formiraju sliku fizičkim kontaktom papira preko trake (ribona), slično pisaćoj mašini.

Novije generacije štampača su laserski i inkdžet štampači (štampači bez dodira) i oni daju izlaz sa većom rezolucijom od matričnih štampača – obično 600 dpi i većom pa zbog toga i dominiraju tržištem.

Laserski štampači, najšire korišćeni savremeni štampači, koriste proces štampanja od 7 koraka.

Većina kolor štampača koristi subtehniku *subtraktivne sinteze* za proizvodnju boja piksela slike: mešaju različite količine svetlo plave (*cyan*), crveno-ljubičaste (magenta), žutih i crnih pigmenata za kreiranje neke boje. Tehnika subtraktivne sinteze boja može se demonstrirati bojenjem preklapajućih oblasti *cyan*, *magenta* i žute boje. Kombinacije sve tri daje crnu boju, samo dve daju crvenu, zelenu i plavu, koje su sekundarne boje subtraktivnog sistema. Većina štampača i monitora su rasterski uređaji, koji formiraju sliku sa sitnim tačkama. Rezolucija rastera štampača meri se u broju tačaka po inču, a kreće se 100 do preko 1000 dpi. Poređenje rezolucija monitora i štampača je teško, jer prvi koriste aditivnu, a drugi subtraktivnu sintezu boja. Monitori mogu prikazati više boja nego štampači, dok ovi mogu prikazati nekoliko boja koje monitori ne mogu. Oba uređaja mogu reprodukovati daleko više nijansi boja, nego što ljudsko oko može detektovati.

Multifunkcionalni izlazni uređaj računarskog sistema (*all-in-one*) kombinuje različite alate koji mogu koristiti slične tehnologije. Obično su kombinovani štampač *(MFP),* skener, štampač i kolor fotokopir mašina. *Ploteri* su veliki ***štampači koji*** proizvode velike, inženjerske šeme i mape koristeći specijalna pera tipa rapidografa za tehničko crtanje, a upotrebljavaju ih veliki projektni biroi i velike organizacije.

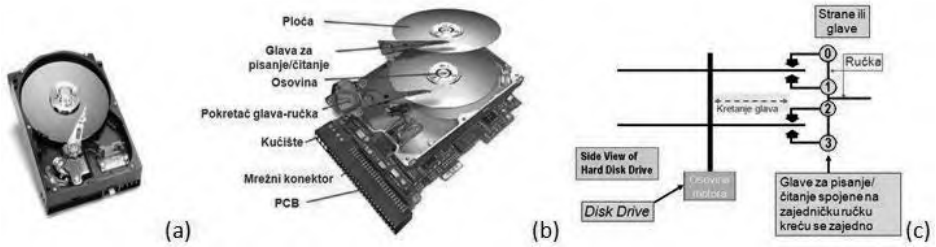
2.5. AUDIO ULAZ/IZLAZ RAČUNARA

Gotovo svi savremeni PC računari poseduju zvučnu karticu, koja prihvata mikrofonski ulaz, a reprodukuje muziku i druge tonove kroz zvučnik ili slušalicu i obrađuje zvuk na različite načine po želji korisnika. Većina zvučnih kartica takođe uključuje *sintesajzere* – specijalizovana kola dizajnirana za generisanje elektronskog zvuka - muzike, šuma i drugih zvučnih efekata. Sa odgovarajućim programom računar može generisati i govor.

2.6. ULAZNO/IZLAZNI UREĐAJI ZA SKLADIŠTENJE PODATAKA

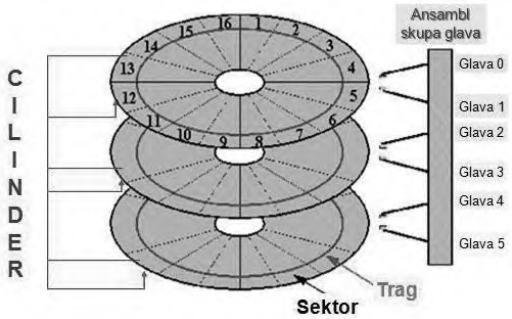
Neki periferijski uređaji računara mogu izvršavati funkcije ulaznih i izlaznih uređaja. U ovu kategoriju uređaja ubrajaju se uređaji za skladištenje podataka *(storrage devices),* koji se često nazivaju *sekundarne memorije*. Osnovni parametri ovih uređaja su lokacija, kapacitet, brzina i metod pristupa.

Čvrsti disk - ***HD*** (*Hard disk*) praktično ima svaki PC (Slika 2.7), kao glavnu sekundarnu (internu) memoriju. Uglavnom su neprenosivi, smešteni u hermetički zatvorena kućišta (a), gde se više ploča diskova obrće konstantnom velikom brzinom (oko 5000 RPM) oko centralne osovine (b), a ručica sa magnetnim glavama kreće se radijalno od spoljne strane diskova ka centru. Glave za upisivanje i čitanje, montirane na zajedničkoj ručci *skeniraju* sve diskove istovremeno sa obe strane radijalnim kretanjem napred/nazad od prvog traga prema osovini ansambla i lebde iznad ploča bez fizičkog kontakta (c).



Slika 2.7. Čvrsti disk, unutrašnja struktura asambla i bočni presek [1]

HD omogućavaju relativno brz pristup, a imaju standardne kapacitete: 500 GB, 1TB, 2 TB itd. Sa više ploča diska u asamblu HD povećava se kapacitet skladištenja bez ekvivalentnog povećanja cene. Ansambl HD mora biti *sterilan* i *hermetički zatvoren*, pošto zbog mikroskopskih dimenzija čestica prljavštine iz životnog i radnog okruženja, lako može doći do oštećenja površine ploča diska. Cilindri, glave, sektori i tragovi digitalnog zapisa na čvrstom disku ilustrovani su na slici (Slika 2.8).



Slika 2.8. Cilindri, glave i tragovi HD

Kapacitet čvrstih diskova računa se jednostavnom formulom množenjem broja cilindera, glava, sektora i kapaciteta jednog sektora (512 B):

Kapacitet FD: C*H*S*512 bajt

Za dati HD kapacitet se proračunava primenom formule: broj cilindera x broj strana/glava x broj sektora po tragu x broj bajta po sektoru, (Slika 28). Na primer, a HD sa 8 ploča (16 strana, 16 glava) biće broj:

Cilindera x Glava x Sektora/tragu x Bajta/sektoru = 6,304x 16x63 x 512 = 3,253,469,184 B ≈ **3.2GB**.

Ključni parametri za proračun kapaciteta HD obično se nalaze na kućištu HD.

Međutim, u poslednjih nekoliko godina, sve češće možemo čuti za SSD diskove, odnosno Solid State Drive diskove. Za razliku od HDD diskova, SSD diskovi nemaju pokretnih delova, i možda je najlakše da ih shvatite kao velike "USB memorije", mada treba imati u vidu da se memorijski čip u SSD disku razlikuje po tipu i brzini od onog u USB fleš memoriji. Ovakva konstrukcija SSD diskova omogućava znatno brže čitanje podataka sa njih što korisniku omogućava da se znatno brže učitava operativni sistem, startuju aplikacije i ukupno gledano ubrza rad računara. [4]

2.7. ERGONOMIJA I ZDRAVLJE

Upotreba računara u svakodnevnom životu i radu, pored brojnih prednosti može imati i posledice po zdravlje čoveka. *Ergonomija* ili humani inženjering je nauka o dizajniranju radnog okruženja koje omogućava ljudima i stvarima da interaktivno deluju efikasno i bezbedno po zdravlje ljudi. Ergonomske studije posledica profesionalne upotrebe računara sugerišu preventivne mere koje korisnici računara mogu preduzeti da zaštite svoje zdravlje radeći na računarima. Stručni saveti za pravilnu upotrebu računarskog sistema u profesionalnom radu i sprečavanje potencijalnih oboljenja uključuju sledeće osnovne preporuke:

- izabrati uređaje i opremu računarskog sistema koji su ergonomski dizajnirani, kao što su monitori sa niskom radijacijom, tastature pod uglom i sl;
- kreirati zdravo radno okruženje, kao što je držanje papirne kopije u visini ekrana monitora, pozicionirati monitor prema svetlu tako da se izbegnu refleksije i sedeti udaljenosti dužine ruke od monitora čime se smanjuje rizik od radijacije ekrana
- ugraditi fleksibilno radno mesto, kao što je stolica za podešavanje monitor sa podešavanjem ugla posmatranja i pokretna tastature; sugeriše se česta promena položaja tela pri radu;
- odmarati oči svakih 2 sata po 15 minuta;.

- skupljati i opuštati napete mišiće na svakom odmoru radi relaksacije ramena, ruku, zglobova i leđa;
- slušati *govor* svog tela i ako se osećate nelagodno, napraviti pauzu, promeniti položaj tela, a nikako sve to ignorisati;
- tražiti stručnu medicinsku pomoć kada je to potrebno.

2.8. ZAKLJUČAK

Računarski sistem sa CPU i internom memorijom ima ograničenu vrednost. Periferijske jedinice omogućavaju komunikaciju računara sa spoljašnjim svetom i memorisanje informacija. Neki periferijski uređaji su striktno ulazni, neki izlazni, a neki su ulazno/izlazni. U savremenom računaru tipične ulazne jedinice su tastatura i miš, a može se priključiti veliki broj različitih, alternativnih ulaznih jedinica, kao što su *trackball, tuchscreen, osetljivi pad* i *joistik*. Izlazni uređaji vrše obrnutu funkciju: prihvataju nizove bita iz računara i transformišu ih u formu koja je pogodna za korisnika. Za prepoznavanje i poređenje specijalizovanih štampanih obrazaca i karaktera, dizajnirani su bar kod čitači, optički čitači i čitači magnetnog mastila. Skeneri i digitalne kamere konvertuju fotografije, crteže i druge analogne slike u digitalne fajlove, koje računar može procesirati. Slično rade i digitalizatori zvuka (zvučne kartice) sa audio informacijama i grafičke (video) kartice za digitalizaciju analognih video signala. Svi ulazni uređaji su dizajnirani da konvertuju analogne signale iz spoljnog sveta u binarne signale koje računar može procesirati.

Izlazni uređaji izvršavaju obrnutu funkciju: binarne signale iz računara transformiše u oblik koristan za spoljne uređaje i korisničku primenu. Veliki broj različitih štampača koristi se za proizvodnju trajnog, papirnog izlaza računara. Tonski izlazi iz računara, uključujući muziku i sintetizovani govor, reprodukuju se zvučnicima ili slušalicama. Izlazni uređaji omogućavaju da računari kontrolišu druge mašine.

Za razliku od većine izlaznih i ulaznih periferijskih uređaja, uređaji za skladištenje kao što su HD, magnetne trake i optički diskovi obezbeđuju dvosmernu komunikaciju sa računarom i mogu biti istovremeno ulazno/izlazni uređaji. Zbog visoke brzine pristupa i slučajnog pristupa, magnetni čvrsti diskovi (HD) velikog kapaciteta, diskete i brojni tipovi pokretnih optičkih i drugih (fleš memorije) medija su najčešći oblici uređaja za skladištenje informacija u savremenim računarima. Sekvencijalni pristup podacima na magnetnoj traci čine je pogodnom za arhiviranje informacija kojima se ne pristupa tako često. Optički diskovi se uglavnom koriste kao mediji visokog kapaciteta za čitanje, ali i upisivanje informacija.

Hardver za kompletan računarski sistem uključuje najmanje jedan procesor, RAM memoriju i nekoliko ulazno/izlaznih uređaja za komunikaciju sa okruženjem. Povezivanje računara u mreže omogućava njihovo direktno međusobno komuniciranje. Računarske mreže šire granice računarskih sistema. Samo sa ispravnim hardverom i periferijskim uređajima računarski sistem je spreman da prima programske instrukcije i interaktivno komunicira sa fizičkim okruženjem.

2.9. KLJUČNI TERMINI

Keyboard (Tastatura računara) - najčešći oblik ulaznog uređaja; koristi se za unos slova, brojeva i specijalnih znakova.

Pointing device (Uređaj za pokazivanje) – obuhvata razne tipove pokazivačkih uređaja (miš, pokazivački štapić, dodirna površina i Slika).

Reading device (Uređaji za očitavanje podataka) – čitaju oznake koje prezentuju kodove, posebno dizajnirane za računarski unos podataka.

Digitalization (Digitalizacija) – proces konverzije analognog signala u digitalni.

Scaner (Skener) - ulazni uređaj koji može kreirati digitalnu prezentaciju štampane stranice/slike.

Bluetooth - ***bežični protokol koji koristi za bežično povezivanje velikog broja uređaja na računar.***

Printer (***Štampač***) - izlazni uređaj koji proizvodi papirni izlaz računara - čvrstu kopiju (*hard copy*) na papiru svake statičke informacije koja se može prikazati na ekranu monitora računara.

USB device (USB uređaj) – univerzalni serijski uređaj za sekundarnu memoriju masivnih podataka.

Video Display Unit (VDU- Video displej jedinica) - podsistem računara koji obuhvata tri ključne komponente monitor, video adapter i video memoriju.

Audio card (Zvučna kartica) – ekspanziona kartica za PCI slot na glavnoj ploči za digitalizaciju audio ulaznih signala sa mikrofona.

Pitanja za ponavljanje

1. Nevedite periferije računara.
2. Navedite ulazne uređaje računara.
3. Navedite izlazne uređaje računara.
4. Koji je standardan raspored slova na tastaturi?
5. Navedite vrste konektora za tastaturu.
6. Navedite uređaje za pokazivanje (pointing devices).
7. Koje su tandardne veličine ekrana (monitora) računara?
8. Navedite vrste monitora prema načinu funkcionisanja:
9. Kako se naziva osnovna tačka na monitoru koja se kontroliše?
10. Šta koriste laserski štampači za štampanje?
11. Navedite uređaje sekundarne memorije (*storrage devices*).
12. Koliki može biti kapacitet CD-ova?

13. Navedite osnovne karakteristike optičkih diskova?
14. Navedite osnovne karakteristike fleš memorija?

Literatura

[1] M. Milosavljević, M. Veinović and G. Grubor, Informatika, Beograd: Univerzitet Singidnum, 2017, p. 355.

[2] T. 4. Site, "That 401k Site," [Online]. http://that401ksite.com/. [Accessed 08 2018].

[3] https://www.nikon.com/, "https://www.nikon.com/," [Online]. https://www.nikon.com/. [Accessed 08 2018].

[4] https://electronics.howstuffworks.com/hdmi.htm. [Online]. https://electronics.howstuffworks.com/hdmi.htm. [Accessed 08 2018].

[5] HP, "ttps://store.hp.com/us/en/cv/printers," [Online]. [Accessed 08 2018].

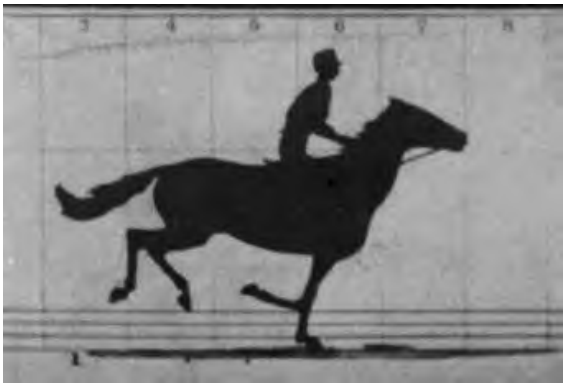
[6] https://www.2bi.me. [Online]. https://www.2bi.me. [Accessed 08 2018].

3. GRAFIKA I MULTIMEDIJA

3.1. UVOD

Pojam multimedija (multimedia) dolazi od latinskih reči multus (mnogi) i medium (medijum). Multimedija predstavlja integraciju različitih medijskih elementa koji su u osnovi samostalni zadržaji što znači da je informacija predstavljena ili sačuvana kao kombinacija teksta, grafike, zvuka, animacije i videa koji su objedinjeni pomoću računara. U interaktivnim multimedijalnim sadržajima korisnicima je omugućena kontrola nad aplikacijom, ako je dodatna sturktura hiperveza onda se radi o takozvanoj hipermediji. [1]

Prvi primer multimedije je "The horse in Motion", Eadweard Muybridge je 1878. godine napravio prvi uživo snimljen film. Eadweard je pokušao da sazna da li konj ikada ima sve 4 noge u vazduhu prilikom punog galopa.



Slika 3.1. Snimak konja prilikom galopiranja [2]

Itraživač je poređao paralelne kamere na svakih 27 inča i prekidanjem konca je konj trigerovao slike. Ekspozicija na kamerama je bila podešena na 1/1000 sekunde. Sve je ovo snimljeno pred novinarima u Palo Altu u Kaliforniji.

3.2. OSNOVNI POJMOVI I ISTORIJSKI PREGLED

Framerate je broj slika koje se prikažu u jednoj sekundi a skrećenica kojom se obeležava je fps. Ljudi mogu da individualno procesuiraju odvojene kadrove koji se krecu do 12 slika u sekundi, sve preko toga se prihvata kao pokretna slika. Prvi filmovi bez zvuka su

se puštali sa 16 do 24 fps-a, film se ručno premotavao prilikom snimanja i fps je zavisio od raspoloženja kamermana. U bioskopima su se puštali filmovi izmedju 22 i 26 *fps.* Ono o čemu su morali da brinu jeste da se oko opterećuje ukoliko je *fps* mali, u starim filmovima su pokušavali da rasterete oko podizanjem *fps-a* na 48 tako što su ponavljali frejmove. Čak dva ili tri puta su ponavljali svaki frejm. Krajem 1920-ih, *fps* za filmove bez zvuka su podigli na 20 do 26 slika u sekundi.

Video streaming je prelos različitih video sadržaja putem kompjuterske mreže. Danas je u svakodnevnoj upotrebi.

Bandwidth presdstavlja širinu pojasa ili drugačije rečeno količinu podataka koja se može poslati i/ili primiti u sekundi.

Veliki napredak u istoriji multimedije desio se 1936. godine na letnjim olimpijskim igrama. Tada se prvi put video prenosio uživo i to u dužini od 72 sata. Karakteristike ovog video prenosa su bile 180 linija i 25 frejmova u sekundi, a koristila se RCA oprema.

Slika 3.2. Snimane letnjih olimpijskih igara 1936. godine u Berlinu

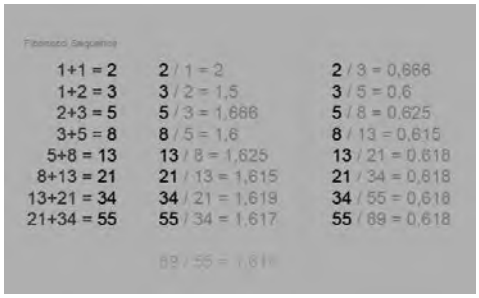
Takođe je bitno pomenuti da su se pre prenosa videa uživo pojavili i video materijali u boji. Prvi (crtani) film u boji se pojavio 1932. godine (Walt Disney – Flowers and trees). Prvi igrani film u boji je bio „La Cucaracha“ 1934. godine i trajao je 20 minuta. Do 1947. godine 12% Američkih filmova su bili u boji a krajem 1960-ih su svi filmovi bili snimani u koloru. Naravno, da bi snimanje u boji moglo biti iskorišćeno u potpunosti bilo je potrebno da se komercijalizuje kolor televizor. Prvi televizor u boji je pušten u prodaju 1951. godine sa cenom od 375$, ta vrednost e ekvivalentna današnjoj ceni od približno 10.000 evra.

Televizori i monitori su od tog doba dosta napredovali, tako da se danas televizori i monitori prikazuju sliku sa mnogo boljim rezolucijama i sa mnogo više frejmova u sekundi. U našem vremenu se tečnim prikazom smatra frame rate veći od 50Hz ili ti 50

frame-ova u sekundi. Moderni monitori prikazuju sliku na 60Hz a neki 4K monitori rade i na 120Hz. Današnji formati video snimaka se prave na 30 *fps* a monitor svaki frejm prikazuje dva puta.

3.3. PROPORCIJA I PRINCIPI KOMPOZICIJE

Proporcija je veoma bitna u mnogim oblastima i sferama života koje nas svakodnevno okružuju pa tako i u grafici i multimediji. Zlatni presek ili božanska proporcija predstavlja vizuelnu reprezentaciju broja koji se nazivi FI, a dobija se iz fibonačijevog niza koji je prikazan na slici (Slika 3.3)



Slika 3.3. Izračunavanje odnosa Fibonačijevih brojeva

Fibonačijev niz predstavlja niz brojeva u kome zbir prethodna dva broja daje vrednost narednog člana niza. Prva dva člana su mu 0 i 1, a dalje glasi:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144…

Fibonačijev niz se često povezuje i sa brojem **fi (ϕ)**, koji je dobio ime po starogrčkom vajaru i arhitekti Fidiji, graditelju Partenona i mnogih figura koje su izgrađene u skladu sa idealnim proporcijama. Upravo je ta dela Fidija smatrao najskladnijim oblicima arhitekture.

Ako uzmemo bilo koji deo Fibonačijevog niza, na primer 2, 3, 5, 8, i podelimo svaki sledeći broj s prethodnim, dobićemo uvek broj približan broju 1,618: (23=1,5, 35=1,66, 58=1,6). Vrednost broja fi iznosi upravo toliko. [3] Kada malo bolje pogledamo krivu koju možemo dobiti uz pomoć ovih brojeva vidimo da se takva kriva nalazi na mnogim mestima a da mi toga nismo ni svesni. Veliki broj starih građevina je imao idealnu proporciju ako se gleda po ovoj metodi, a isto tako nas i priroda okružuje stvarima koje su idealne u ovakvoj proporciji. Samo neki od mnogobrojnih primera su prikazani u sledećim slikama koje slede.

Slika 3.4. Primeri proporcije uz pomoć Fibonačijevih brojeva [4]

Kao što je pomenuto zlatni presek možemo naći i u prirodi što je prikazano na sledećim slikama.

Slika 3.5. Primeri zlatnog preseka u prirodi

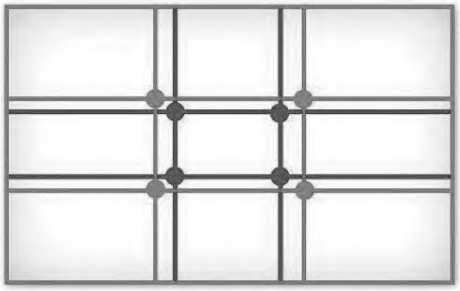
Kompozicija može biti simetrična i asimetrična. Da bi slika bila prijatna, skladna, a da bi izbegli cetralnu kompoziciju, stari majstori su primenjivali razne metode i matematičke proračune. Pošto su simetrične kompozicije katkad monotone i statične, zlatni presek je korišćen kao jedno od rešenja. Ovo pravilo je definisao rimski arhitekta Vitruvius koji pokušao da uspostavi nejednaku raspodelu i asimetriju prostora koje je prijatna i privlačna ljudskom vizuelnom doživljaju. To je klasična matematička formula za balansiranu raspodelu "težine" na slici. [5]

3.3.1. Kompozicija slike

Kompozicija je pojam koji označava raspored elemenata na slici. Slika ima dve dimenzije – dužinu i visinu, a njome se prikazuje trodimenzionalni prostor – dužina, visina i dubina. U kompoziciji slike ljudsko oko najpre uočava predmet koji je fokusiran, tj. izoštren. Međutim, ukoliko su svi predmeti na fotografiji izoštreni, posmatrač u svakom slučaju pronalazi glavni predmet slici. [6]

Pravilo trećina

Pravilo trećina je još jedno pravilo koje služi da izbegnemo centralnu kompoziciju – iako je ono u osnovi simetrično. Na slici ispod se vidi jednom linijom pravilo trećina, a drugom linijom su obeležene proporcije zlatnog preseka. Pejzaži su najbolji primer primene ovog pravila. Po ovom pravilu, liniju horizonta, ili objekte u kadru, najbolje je staviti na neku od linija koje presecaju format na trećine, ili u samom preseku horizontalne i vertikalne trećine formata.



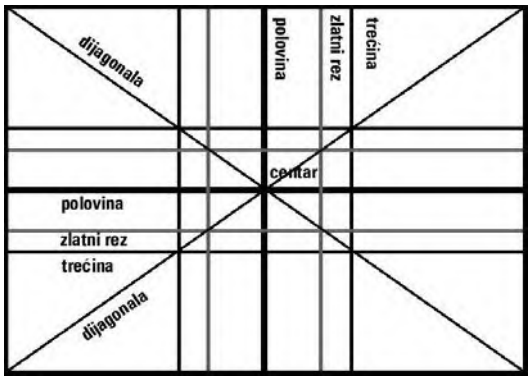
Slika 3.6. Pravilo trećine u poređenju sa zlatnim presekom

Tačka u kojoj se presecaju linije trećina je najsnažnija tačka u formatu, odnosno privlači pažnju i centar interesovanja bi trebalo da je postavljen na tom mestu. [5]

Vertikalne, horizontalne i dijagonalne linije

U kompoziciji fotografije veliki značaj mogu da imaju i vertikalne (uspravne), horizontalne (vodoravne) i dijagonalne linije. Vertikalna kompozicija se koristi kod snimanja pojedinačnih portreta, drveća, vodopada, visokih građevinskih objekata itd. Vertikalne linije naglašavaju visinu, veličinu i stabilnost. Horizontalna ompozicija se primenjuje uglavnom za pejzaže i panoramne snimke urbanih sredina. Horizontalne linije izražavaju spokojstvo, mirnoću. širinu, prostranost i stabilnost. Dijagonalnu kompoziciju koristimo uglavnom za fotografisanje određenih detalja u pejzažu ili predmeta. Pomoću dijagonalnih linija izražava se kretanje, dinamika i nestabilnost. [7]

Na sledećoj slici se najalakše mogu grafički prikazati pomenuti formati kompozicije.



Slika 3.7. Grafički prikazani formati kompozicije

3.4. KOMPRESIJA

Multimedijalni sadržaj zauzima puno mesta na računaru pa nam to i predstavlja problem prilikomemitovanja ili prenosa sadržaja. Do rešenja ovog problema dolazimo kompresijom.

Postoje dve vrste kompresije, kompresija gde gubimo na kvalitetu (JPG) i kompresija gde ne gubimo na kvalitetu (BMP). Kompresija nam omogućava bži prenos i emitovanje sadržaja ali moramo obratiti pažnju na balans između kvaliteta i brzine prenosa.

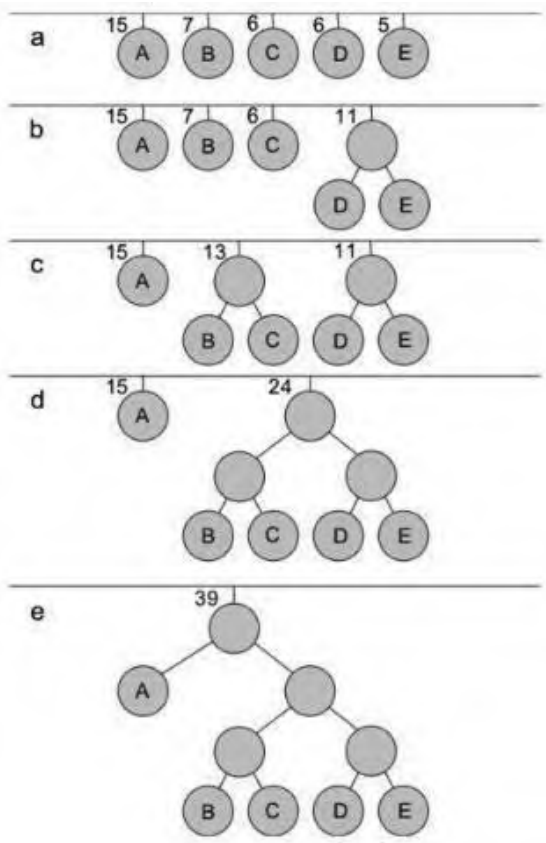
U računarstvu i teoriji informacije, Huffmanov kod je poseban prefiksni kod koji se najčešće koristi za sažimanje podataka bez gubitaka. Proces pronalaženja ili korištenja toga koda naziva se **Huffmanovo kodiranje**. Algoritam pomoću kojega se pronalazi kod, konstruirao je David Albert Huffman za vrijeme doktorskog studija na MIT-u. Svoje otkriće objavio je u radu "A Method for the Construction of Minimum-Redundancy Codes" 1952 godine. [8]

Huffmanov kod je neravnomeran kod. Algoritmom se dobija tablica kodova koja se bazira na verovatnosti pojave svakog simbola pojedinačno, što znači da će simboli koji imaju veću verovatnost pojavljivanja biti kodirani s kraćim kodnim rečima, a simboli koji imaju manju verovatnost pojavljivanja biti kodirani s dužim kodnim rečima. Takvim načinom kodiranja povećava se efikasnost koda.

Huffmanov algoritam je algoritam pomoću kojeg se na jednostavan način može kodirati neka informacija. [9]

Huffmanov algoritam kreće od šume **x** stabala sa po jednim eksternim čvorom. U svakoj iteraciji iz reda u kome se nalaze stabla vade se dva korena sa najmanjim težinama, zatim se pravi čvor sa težinom koja je jednaka zbiru težina izabranih čvorova. Sada je novonastali čvor koren novog stabla čiji su levo i desno podstablo izabrana dva čvora. Novonastalo stablo se sada vraća u red i u novim iteracijama konkuriše ravnopravno sa ostalim stablima. U svakoj novoj iteraciji se uzimaju nova dva korena od kojih se pravi

novo stablo sa korenom čija je težina jednaka zbiru težina korena izabranih stabala. Ovako se u svakoj iteraciji smanjuje broj stabala za jedan sve dok u redu ne ostane samo jedno stablo. Ako na početku imamo šumu od h stabala, potrebna nam je **x**-1 iteracija da bi dobili krajnje stablo. Ovakvo stablo ima koren čija je težina jednaka zbiru težina svih eksternih čvorova koji su se na početku nalazili u redu, a koji takođe sada čine listove novonastalog stabla. Ovako nastalo stablo ima minimalnu težinsku eksternu dužinu puta.



Slika 3.8. Ilustrovan primer Huffmanovog kodiranja

Huffmanovo kodiranje pronalazi optimalan sistem kodiranja zasnovan na relativnoj verovatnoći pojavljivanja svakog elementa. Predstavlja entropijsko kodiranje simbola bez gubitka informacija. Hafmanovo kodiranje je najefikasniji metod kompresije simbola koji nisu u tesnom odnosu, u svojoj klasi. On pripada klasi preslikavanja karaktera neke tekstualne kolekcije u bitove za poznate verovatnoće pojavljivanja svakog karaktera. Najduži Hafmanov kod se dobija kada se distribucija verovatnoća pojavljivanja odvija po pravilu generisanja Fibonačijevih brojeva. [10]

U udžbeniku će biti objašnjen još jedan primer kodovanja a to je Lampel - Ziv kodovanje. Ovaj tip kodovanja predstavlja jedinstven način kompresije podataka. Osnovna ideja Lempel-Ziv kodiranja podataka je kreirati rečnik od podataka koji su procesuirani i zamena simbola (ili celog niza simbola) sa indeksima iz toga rečnika. Tako u kompresovanom nizu podataka koristimo samo pokazivače prema frazama koje se nalaze u rečniku. Sama metoda kreiranja rečnika u Lempel-Ziv kodiranju je razlog što je ova vrsta kodovanja vrlo učinkovita. Svoju metodu Lempel i Ziv su predstavili u dve varijante koje su poznatije pod imenima LZ77 i LZ78. Iz tih medota kompresija razvile su se i druge varijante koje unosne male promene u sam algoritam i uglavnom opisuju tehničku stranu realizacije algoritma u praksi. Razvijene su sledeće varijante kodiranja LZ77 i LZ88 algoritmom: [11]

LZ77 varijante: LZR, LZSS, LZB, LZH

LZ78 varijante: LZW, LZC, LZT, LZMW, LZJ, LZFG

U daljem tekstu će biti prikazan jedan primer ove metode kodovanja I za to ćemo koristiti niz

AABABBBABAABABBBABBABB.

Potrebno je da napravimo rečnih malih stringova koji se koriste da bi predstavili ceo tekst. Prvo počinjemo da gradimo rečnik sa najkraćom frazom u tekstu. Ovo će na početku uvek biti samo jedno slovo a u ovom slučaju je to slovo A.

A|ABABBBABAABABBBABBABB

Sada uzmemo sledeću frazu koju nismo do sada videli, maksimalno za jedno slovo duže od najduže koju nismo do sada videli.

A|AB|ABBBABAABABBBABBABB

Sledeća serija karaktera koja nije jos viđena je ABB, pa onda samo B posle toga.

A|AB|ABB|B|ABAABABBBABBABB

Nastavljamo dalje da primenjujemo ista pravila u nizu.

A|AB|ABB|B|ABA|ABAB|BB|ABBA|BB

U ovom slučaju BB je već viđeno ali obzirom da su to poslednji karakteri u nizu možemo zanemariti da je to već viđeno i možemo započeti sa kreiranjem rečnika što će biti prikazano u sledećoj tabeli.

1	2	3	4	5	6	7	8	9
A	AB	ABB	B	ABA	ABAB	BB	ABBA	BB
A	1B	2B	0B	2A	5B	4B	3A	7

Tabela 3.1. Huffmanov rečnik i Huffmanov kod

Prvi i drugi red predstavljaju rečnik a treći je kod za rečnik iz prethodna dva reda koji se zapisuje u binarnom formatu. U binarnom kodu rečnika A=0 i B=1. Dakle naš niz možemo predstaviti na sledeći način:

011101001010010111001011001 11

Kada podelimo naš binarni zapis na "|" i "," dobijamo sledeće:

0|1, 1|10, 1|00, 1|010, 0|101, 1|100, 1|011, 0|0111

Prva | dolazi posle jednog bita, sledeća posle 2 bita, sledeće dve posle 3 bita, onda sledeće 2^2 od 4 bita, 2^3 od 5 bitova, 2^k od dužine k+2 bitova. Simbol „,“ ide pre poslednjeg simbola u svakom skupu. Na taj način smo završili sa kodovanjem našeg niza.

Još jedan vid formata (kompresovanog) koji je vema vredan pomena jeste JPEG (Joint Photographic Experts Group). Ovaj format predstavlja kompresiju fajlova sa gubicima a izveden je iz bitmape. Ovaj format je takođe najčešće korišćen u normalnom radu sa slikama zato što zauzima veoma skroman prostor memorije. Obzirom na to pogodan je za arhiviranje kao i za razmenu slika putem infomacionih mreža. Praktično svi programi za obradu slika ga podržavaju. Ovakav format postiže kompresiju 10:1 sa vrlo malo primetnog gubitka kvaliteta, a baziran je na ideji da se mogu izbaciti pikseli sa naglim promenama inteziteta (boja) u odnosu na okolinu. Funkcioniše tako što se slika podeli na kvadrate 8x8 i svaki se posebno kompresuje.

3.5. RAČUNARSKA GRAFIKA

Programi za tabele, statistički i drugi matematički programi kreiraju kvantitativnu grafiku. Ovi programi štede vreme poslovnim ljudima, naučnicima i inženjerima da ručno kreiraju visokokvalitetne crteže. Brojne grafičke aplikacije omogućavaju izradu jednostavnih crteža, slikarske alate i kompleksne programe za profesionalne umetnike i dizajnere.

Programi za bitmapiranu grafiku omogućavaju korisniku da „slika“ piksele na ekranu sa mišem. Tipični programi za slikanje prihvataju ulaze sa miša, *joystick*-a, *trackball*-a, *tuchpad*-a ili olovke , prevodeći kretanje pokazivača u linije i obrasce na ekranu. Program za slikanje tipično nudi paletu alata na ekranu. Programi za slikanje kreiraju bitmapiranu grafiku (ili rastersku grafiku) – slike koje su za računar jednostavno mapa, koja pokazuje kako pikseli slike treba da budu postavljeni na ekranu. Za najjednostavniju bitmapiranu grafiku, jedan bit memorije računara predstavlja svaki piksel. Pošto bit može imati vrednost 0 ili 1, svaki piksel može prikazivati jednu od dve moguće boje – crnu i belu.

Alociranjem više memorije po pikselu svaki piksel može prikazati više mogućih boja ili nijansi i proizvesti grafiku višeg kvaliteta. Grafika skale sivog omogućava da se svaki piksel pojavljuje kao crna, bela ili jedna od nijansi iz skale sivog. Neki program koji alocira 8 bita po pikselu omogućava da se na ekranu prikaže 256 različitih nivoa sivog crno bele slike. Realistična kolor grafika zahteva više memorije računara. 8-bitna

grafika omogućava prikazivanje slika, ali ne i tačnu reprodukciju većine kolor fotografija. Fotorealistična fotografija zahteva 24-bitnu ili 32-bitnu memoriju za svaki piksel slike na ekranu. Na primer, crno-bele slike koriste za svaki piksel dve boje - crnu ili belu, koje se mogu definisati korišćenjem samo jednog bita po pikselu. U ***RGB (****crvena, zelena, plava****) modelu boja*** za jednostavan crtež dovoljna je dubina piksela od 8 bita ($\mathbf{2^8}$ = **256** različitih boja); slike u punoj boji su sa dubinom piksela od 24 bita ($\mathbf{2^{24}}$≈**16,7** miliona različitih boja), što je više nego dovoljno za ljudsko oko. U ***CMYK (****cijan, magenta, žuta i crna*) modelu boja za svaku od četiri komponente po 8 bita po pikselu, čini $\mathbf{2^{32}}$ ili preko 4 milijarde različitih boja!

Broj bita namenjen za svaki piksel naziva se *dubina boje* ili *bitska dubina,* koja je jedan od dva tehnološka faktora koji ograničavaju mogućnosti umetnika da kreira realistične slike na ekranu sa bitmapiranim grafičkim programima. Drugi faktor kvaliteta bitmapirane slike na ekranu monitora je rezolucija – gustina piksela u broju tačaka po inču (dpi).

Međutim, neke grafičke slike, posle gledanja na ekranu, namenjene su za štampanje na štampaču, pa je potrebno poznavati i rezoluciju štampača. Tako slika prikazana na monitoru rezolucije 72 dpi koja izgleda da je visoko kvalitetna, odštampana na štampaču neće imati tako finu strukturu kvalitetne fotografije – dijagonalne linije, krivulje i slova teksta imaju sitne stepeničaste prekide u kojima se uočavaju pikseli slike. Programi za slikanje upravo se brinu za ove prekide omogućavajući korisniku da uskladišti sliku sa 300 dpi, ili većom rezolucijom, čak iako monitor ne može prikazati svaki piksel. Dakle, slike visoke rezolucije zahtevaju više memorijskog prostora na disku. Što je rezolucija štampača veća, teže je ljudskom oku da uočava piksele štampane slike.

Bitmapirane slike kreirane na računaru organizuju se u *grafičke fajlove***.** *Formati grafičkog fajla* su različiti, a predstavljaju način na koji se informacija o slici organizuje u fajlu. Poznatiji formati grafičkih fajlova su: Windows *bitmaps* (**.BMP),** *Device independent bitmaps, PC Paint*brush, *True Visions Targa* (**.TGA),** *Portable Document Format* (**.PDF),** *Graphics Interchange Format* (**.GIF),** *Joint Experts Photography Group* (**.JPG, .JEPG),** *Tagged Image File Format* (**.TIFF),** *Apple Macintosh* (**.PICT).**

Za skladištenje bitmapirane slike potreban je znatan memorijski prostor koji se računa jednostavnom formulom:

*potreban memorijski prostor = broj piksela **x** dubina piksela*

Primer 1: za otisak slike u boji veličine 10 x 8 inča (~ 25 x 20 cm) sa štampačom rezolucije 600 dpi, broj piksela je: (10 x 600) x (8 x 600) = 6000 x 4800 = **28,800.000**, dubina piksela je 32 bita=4 B, a potreban memorijski prostor više od 100 MB. Za smanjenje grafičkih fajlova i uštedu memorijskog prostora koriste se programi za *kompresiju*.

Za kvalitetan rad računarska grafika zahteva, minimalno personalni računar najboljih performansi ili namensku radnu stanicu, grafički korisnički interfejs (GUI), multimedijalne sisteme na Internetu, grafičke multimedijalne programe za prikaz sadržaja na optičkim diskovima, tipa CD-ROM i DVD, itd.

Namena računarske grafike je generisanje, obrada, modeliranje, vizuelizacija i animacija, statičnih i dinamičnih sintetičkih slika, u 2-D, ili 3-D prostoru uz pomoć računara i računarskih programa. Ima široku primenu u nauci, industriji, arhitekturi i dizajnu za stvaranje i obrada statične 2D sintetičke slike (vektorska i raster grafika), 3D vizuelizacija u realnom vremenu (video igre, modeli virtuelne realnosti), računarsku animaciju, video digitalizaciju i video vizualizaciju, obradu i dorađivanje specijalnih efekata (za potrebe filmske i televizijske industrije). Primenjuje se za potrebe grafičkog dizajna, industrijske štampe, itd; kreiranje i manipulaciju 3D računarskih modela (inženjerstvo, arhitektura, industrijski dizajn, hirurgija), itd; grafičke modele preko složenih algoritama (numerička mehanika za dinamičku simulaciju ponašanja fluida, određenih prirodnih fenomena, kretanja krvotoka, u meteorologiji, vulkanskoj geologiji, fizici, itd).

3.5.1. Rasterska grafika

Slika na ekranu monitora računara sačinjena je od piksela – najsitnijih crnih, belih i obojenih elemenata slike aranžiranih u redove. Reči, brojevi i slike koje vidimo na ekranu nisu ništa drugo do obrasci piksela koje kreira program. Većinu vremena korisnik ne kontroliše ove obrasce piksela; grafički programi automatski kreiraju ove obrasce kao odgovor na komande. Automatska grafika (npr. uzorci bar dijagrama) uobičajena je, ali može biti i restriktivna. Ako korisnik želi veću kontrolu nad detaljima na ekranu monitora, više odgovara vektorska grafika.

Dvodimenzionalna računarska grafika je npr., statična slika i animacija. Moguće su različite vrste statične slike: crteži, dijagrami, grafikoni, fotografije. 2-D grafika može trenutno da prenese informaciju i da bude hiperveza za drugi tip informacija. *Vizuelizacija* je važan deo komunikacionog procesa a obuhvata naglašavanje pojedinih delova, ilustrovanje pojmova, usmeravanje pažnje izbor podloge za sadržaj. Obezbeđuje rastersku (bitmapiranu) sliku, fotorealističnu sliku i složene crteže. Slika je matrica ***piksela*** (*picture element*). Bitmapirane slike su memorijski zahtevne. Nastaju problemi ako se povećava slika. Ako se rezolucija *smanjuje* -gube se detalji sa slike, a ako se *povećava*- treba izvršiti interpolaciju između tačaka.

Rasterska (bitmapirana) grafika (RG) je stalna 2-D mreža piksela. Svaki piksel ima svoju vrednost (svetlinu, boju, transparentnost, ili kombinaciju ovih). Rasterska grafika ima konačnu rezoluciju - ako se ona poveća najčešće se gubi kvalitet (gube se pikseli slike). Rasterske slike su slike prenesene iz stvarnog sveta: skenirani materijali, negativi filmova, snimci digitalnom foto kamerom, digitalizovani „okvir" video snimka, sintetizovana slika primenom odgovarajućeg programa, snimak (dela) ekrana, druga grafika pretvorena u bitmapiranu.

Kao i slike kreirane grafičkim programom visoke rezolucije, digitalna fotografija snimljena digitalnom kamerom je bitmapirana slika. Program za procesiranje digitalnih slika omogućava korisniku da manipuliše fotografijama i drugim slikama visoke rezolucije sa alatima sličnim programima za slikanje.

Bitmapirane slike (pikseli) imaju prednosti što bolje kontrolišu teksture, senčenja i fine detalje. Pogodne su za prikazivanje na ekranu, simulirajući prirodne slikarske podloge, kao i za ulepšavanje fotografija.

3.5.2. Vektorska grafika

Vektorska (objektno orijentisana) grafika koristi geometrijske oblike: linije, kvadrate, krugove, elipse, poligone i sl. Svaki oblik se matematički predstavlja koordinatama i rastojanjima. Vektor je linija opisana pozicijom svojih krajnjih tačaka, pravcem, smerom i intenzitetom. VG je najbolja je za dijagrame i crteže, ali je teža za rad.

Vektorska grafika sadrži tačne geometrijske podatke: topologiju, koordinatne tačaka, veze između tačaka (za linije i putanje), boju itd., a ne može se pregledati pomoću vanjskog programa (web browser npr.), odnosno, vektorsku grafiku prepoznaje program pomoću kojeg je nastala. Često različiti programi za VG mogu čitati druge formate (Freehand čita **.cdr** Corel Draw datoteku) zato je VG evoluirao u rastersku **(.jpg, .bmp** itd.). Sa povećanjem rezolucije VG ne gubi kvalitet slike zato što ima tačne podatke o geometriji i drugim parametrima slike. Postoje formati koji uključuju i RG i VG (.pdf, .swf).

Vektorska grafika obezbeđuje crtanje na računaru, upotrebom nekog softvera . Softver za crtanje memoriše slike, ne kao kolekcije tačaka, nego kao kolekciju linija i oblika. Kada korisnik crta liniju sa programom za crtanje, program ne snima promene u mapi piksela. Umesto toga program računa i pamti matematičku formulu za liniju. Program za crtanje skladišti oblike kao formule za oblike, a tekst kao tekst. Pošto su slike kolekcije objekata - linija i oblika, ova grafika se često naziva objektno orijentisana ili vektorska grafika. Drugim rečima računar pamti, na primer, „plava linija ide ovde, crveni krug tamo“, umesto“ovaj piksel je plav, onaj crven, a onaj beo“, kao kod rasterske grafike. Memorijski zahtevi nisu tako veliki kao kod rasterskih slika. Mnogi alati za crtanje linija i oblika, kao i alati za tekst, su slični slikarskim alatima u bitmapiranim programima, ali korisnici mogu manipulisati objektima i uređivati tekst bez uticaja na najbliže objekte, čak i kada se preklapaju sa tekstom. Na ekranu obe slike izgledaju slično, ali kada se crtež štampa, jasan je onoliko kolika je rezolucija štampača. Naravno svi crteži se ne moraju štampati, npr., za publikovanje na web stranici. Međutim, kako većina web pretraživača prepoznaje samo bitmapirane slike, verovatno treba konvertovati crteže u bitmapirane slike pre objavljivanja na web stranici.

PostScript programski jezik je standardni, tzv. *page-description language*, za opis tekstualnih fontova, ilustracija i drugih elemenata štampanih stranica. Koriste ga profesionalni programi za crtanje kao što *su Adobe Illustrator i Macromedia Freehand* i dr. Ugrađen je u mnoge laserske štampače i druge izlazne uređaje, tako da ovi uređaji prihvataju i izvršavaju *PostScript* naredbe. Program za crtanje zasnovan na *PostScript* jeziku konstruiše jedan *PostScript* program na osnovu tekućeg crteža. Ovaj program

obezbeđuje kompletan set instrukcija za rekonstrukciju slike na štampaču. Kada korisnik aktivira komandu *Print,* računar šalje *PostScript* instrukcije na štampač, koji ih koristi za konstrukciju matrice mikroskopskih piksela za štampanje na svakoj stranici. Većina desktop programa za publikovanje koristi *PostScript* program na isti način.

Objektno orijentisano crtanje i bitmapirano slikanje imaju određene prednosti i nedostatke za neke aplikacije. Programi za bitmapirano slikanje daju umetniku i uredniku fotografija neograničenu kontrolu nad teksturom, senkama i finim detaljima, što se mnogo koristi u video igrama, multimedijskim prezentacijama i web stranicama za simulaciju prirodnih slikarskih medija i obradu fotografija. Objektno orjentisani programi za crtanje su bolji za kreiranje štampanih grafikona, dijagrama i ilustracija sa jasnim linijama i glatkim oblicima. Neki integracioni programi, kao što su *Corel Draw* i *AppleWorks* sadrže module i za *crtanje i za slikanje*. Omogućavaju izbor pravog alata za svaki poseban zadatak. Neki programi objedinjavaju mogućnosti oba sistema u jednoj aplikaciji, čime se brišu razlike između tipova programa za slikanje i crtanje, čime se nude nove mogućnosti za amaterske i profesionalne ilustratore i slikare.

Za samostalno kreiranje umetničkog dela preporučuju se opuštena priprema, izbor pravih alata za dati posao, preuzimanje najboljih rešenja, ne preuzimanje bez dozvole i zaštita sopstvenog rada.

Pitanja za ponavljanje

1. Šta predstavlja pojam multimedija ?
2. Šta je framerate ?
3. Šta predstavlja video streaming ?
4. Šta je Bandwidth ?
5. Šta je proporcija ?
6. Kako objasniti zlatni presek ?
7. Šta predstavlja komoziciju slike ?
8. Objasniti pravilo trećina.
9. Objasniti šta je predstavlja kompresija.
10. Kako funksioniše Huffmanovo kodiranje ?
11. Šta je JPEG ?
12. Šta predstavlja rasterska grafika ?
13. Šta predstavlja vektorska grafika ?

Literatura

[1] Horić-Božić, “poincare.matf.bg.ac.rs,” [Online]. http://poincare.matf.bg.ac.rs/~vladaf/Courses/PmfBl%20I%20MNR/Predavanja/Predavanja02.pdf.

[2] wikimedia. [Online]. https://commons.wikimedia.org/wiki/File:The_Horse_in_Motion-anim.gif. [Accessed 08 2018].

[3] OK matematika, “matematika.edu.rs,” [Online]. http://matematika.edu.rs/fibonacijev-niz-u-prirodi-saznaj-gde-si-video-njegovu-pravilnost-da-nisi-ni-primeio/. [Accessed 08 2018].

[4] yogaesoteric, [Online]. http://www.yogaesoteric.net/content.aspx?lang=EN&item=13322. [Accessed 08 2018].

[5] https://www.tiyana.net, “https://www.tiyana.net,” [Online]. https://www.tiyana.net/sr/principi-kompozicije/zlatni-presek/.

[6] fotograf.co.rs, [Online]. http://fotograf.co.rs/fotografska-kompenzicija/. [Accessed 08 2018].

[7] D. Mustur, “http://digitalnasrbija.org,” [Online]. http://digitalnasrbija.org/fotografija-i-pravila-kompozicije/. [Accessed 08 2018].

[8] D. A. Huffman, “http://compression.ru,” 1952. [Online]. http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf. [Accessed 08 2018].

[9] D. Sofić, “http://denis-sofic.from.hr,” [Online]. http://denis-sofic.from.hr/huffmanov-kod. [Accessed 08 2018].

[10] wikipedia.org, “wikipedia.org,” [Online]. https://en.wikipedia.org/wiki/Huffman_coding. [Accessed 2018].

[11] P. Juroš, M. Srbiš and I. Grubišić, “http://pvprm.zesoi.fer.hr,” [Online]. http://pvprm.zesoi.fer.hr/2005-2006-web/studenti_rad/pjuros/seminar/lempel-ziv_metoda_vizualizacije.htm. [Accessed 08 2018].

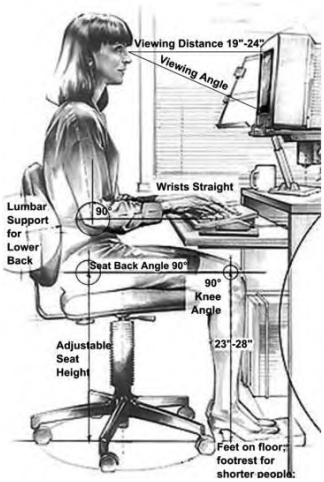
4. INTERAKCIJA ČOVEKA I KOMPJUTERA

Ne možemo da živimo bez tehnologije. Razmislite na nekoliko sekundi koliko različitih sprava korstite na dnevnom nivou. Glavne su, na primer, mobilni telefon i razne njegove aplikacije, kompjuter, laptop, tablet, daljinski za TV ili klimu, bankomat, šporet, mikrotalasna peć itd. Da li imate omiljenu spravu među njima, ili možda neku koja vas nervira čim je uzmete u ruke? Zbog čega vas nervira? Neke sprave se lakše koriste od drugih i u velikoj meri njihova upotrebljivost diktira naš nivo zadovoljstva kada ih koristimo. U ovom poglavlju postavljamo pitanje šta je upotrebljivost, i kako da napravimo proizvod koji ima visok nivo upotrebljivosti. Nije dovoljno rešiti problem, nego je jednako bitno, a možda i bitnije, da ga rešimo na način koji će učiniti da korisnik bude zadovoljan celokupnim postupkom rešavanja. Ovaj deo problema je često zaboravljen na našim prostorima i deo je razloga zbog čega skoro nemamo svetske poznate proizvode koji se povezuju pozitivnim reakcijama korisnika ni u jednoj industriji.

4.1. UVOD

Interakcije čoveka i kompjutera, ili Human Computer Interaction (HCI), naučno je polje koje je nastalo 1980-ih godina kao spoj kognitivne psihologije i inženjerstva ljudskih faktora. Kompjuteri pre tog perioda bili su ogromne mašine veličine jedne kuće, za koje je bilo potrebno vrlo usko specijalizovano znanje da bi se koristile. Upravljanje jednim kompjuterom bilo je skoro svedeno na inženjere i istraživače koji su radili u zgradi gde se i sam kompjuter nalazio. Da bi se omogućio rad na kompjuteru za široke slojeve stanovništva, morao se prilagoditi način njegovog korišćenja na nivo gde i dete može da ga upotrebi. U tom smislu, interfejs, ili način upravljanja, kritični je faktor u osposobljavanju ove interakcije.

Danas, interfejs može da se odnosi čak i na položaj i orijentaciju glave, ruke, noge i celog tela kada koristimo kompjuter, sve do dizajna i rasporeda dugmića i načina korišćenja programa na ekranu. Ergonomija je studija ovog prvog, i odnosi se na praksu projektovanja proizvoda, sistema, ili procesa, gde moraju da se uzmu u obzir interakcije između njih, i ljudi koji ih koriste. Slika 4.1 predstavlja jedan od prvih primera ergonomije, gde se vidi teoretski idealna pozicija sedenja prilikom korišćenja kompjutera. Ključni faktori u ovoj vrsti ergonomije su da ekran bude u ravni ili ispod nivoa očiju, da lakatovi budu na minimum 90 stepeni kada se kuca, da kičma bude podržana u lumbalnom delu, i da kolena takođe budu na 90 stepeni. Ovo nije jedini način pravilnog sedenja prilikom korišćenja kompjutera, ali je dobar referentni početak.



Slika 4.1. Dizajn kancelarijskog nameštaja [1]

Što se tiče korišćenja kompjutera, ili nekog njegovog softvera, pravi bum je napravljen 1968. godine, kada je Doug Engelbart demonstrirao skoro sve moderne funkcije kompjutera u jednoj prezentaciji od samo 90 minuta [2]. Tu je ceo svet prvi put video koncept „prozora“, ili windows za operativni sistem, hipertekst, grafiku, navigaciju i unos komandi, video-konferenciranje, miša, tekstualni editor, linkove za fajlove, kontrolu verzije dokumenta, pa čak i kolaborativno editovanje fajla preko Interneta, što je preteča Google Drive-a ili dropboksa. Ovaj spisak funkcija je i dan-danas osnova svih poslova koje obavljamo na kompjuteru na dnevnom nivou i koncept njihovog korišćenja, čak do nivoa opreme pomoću koje ih koristimo, ostao je nepromenjen već više od 50 godina. Ono što jeste promenjeno i unapređeno je lakoća, brzina i pristupačnost njihovog korišćenja, odnosno njihova upotrebljivost.

4.2. UPOTREBLJIVOST

Centralna tema, i najbitniji princip u polju HCI, upravo je upotrebljivost. Ona se definiše kao stepen do kojeg grupa korisnika može da upotrebi proizvod da bi postigli odgovarajuće ciljeve efektivno, efikasno i na zadovoljavajući način u specifičnom kontekstu. Ova definicija zaslužuje malo dublju analizu jer svaki njen deo treba da se rastavi i specifično definiše. Prvo, spominje se “grupa korisnika”, što znači da se odnosi na podskup korisnika, odnosno na glavne korisnike datog proizvoda. Dalje, ta grupa korisnika treba da postigne svoje ciljeve efektivno i efikasno, što znači da treba dati cilj da postignu u visokom procentu slučajeva (efektivno), i da to urade vrlo brzo (efikasno). To sve treba da se desi na zadovoljavajući način prvobitnoj grupi ciljnih korisnika. Poslednji aspekt definicije upotrebljivosti jeste da proizvod treba da radi u datom kontekstu. Znači

da proizvod može da bude toliko specifičan da optimalno funkcioniše pod određenim okolnostima. Na primer, softver kao što je Skajp može da se optimizuje da prikazuje izuzetno kvalitetan video-snimak kada su korisnici prikopčani na Internet sa slabim protokom. Takav sofver bi se najbolje pokazao u takvom kontekstu gde su korisnici na primer van grada.

Dodatni faktori koji utiču na upotrebljivost su estetika, zadovoljstvo korisnika, kao i celokupno korisničko iskustvo sa proizvodom. Slika 4.2 prikazuje dva daljinska upravljača za TV. Koji biste vi radije koristili? Koji je intuitivniji za korišćenje? Ovo su prosta pitanja, ali odgovori ne moraju da budu jednako prosti. Navika igra veliku ulogu u davanju odgovora na navedena pitanja. Način korišćenja takođe igra važnu ulogu jer neko ko nema iskustvo sa daljinskim koji ima mnogo dugmića mnogo teže bi se snašao sa takvim u mraku, pa bi u tom slučaju intuitivniji daljinski (slika 4.2 dole) bio prvi izbor. U ovom istom primeru je donji daljinski estetski privlačniji zbog svog minimalističnog dizajna, koji takođe rezultira povećanim zadovoljstvom korišćenja. Ako donji daljinski pouzdano i lako radi, drugim rečima - manje nervira korisnika, onda je taj upotrebljiviji.

Slika 4.2. Estetika, zadovoljstvo korisnika i iskustvo sa proizvodom

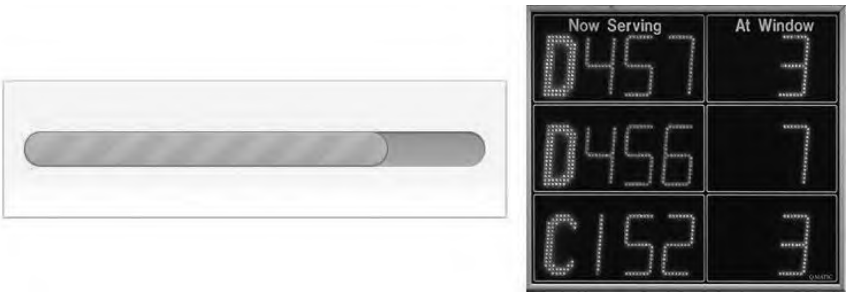
4.3. PRINCIPI I MERE UPOTREBLJIVOSTI

Upotrebljivost može da se razloži na razne tačke, koje individualno utiču na njen nivo. Ove tačke, ili principi, ne važe samo za upotrebljivost u smislu interakcije čoveka i kompjutera, nego generalno navedeno u širem smislu odnosa korisnika i bilo kog proizvoda. Nisu svi principi primenjivi van konteksta HCI, ali se vidi u kom smeru teče suština merenja upotrebljivosti.

4.3.1. Stanje sistema na vidnom mestu

Sistem mora da konstantno obaveštava korisnike o svom radu i napretku u realnom vremenu. Korisnici osećaju da je situacija pod kontrolom ako su obavešteni o stanju sistema, u suprotnom mogu da pomisle da je došlo do greške. Ako sistem reaguje na

datu komandu u roku od pola sekunde ili manje, korisnik nema potrebe da bude konstantno obavešten o stanju sistema jer vidi promene u realnom vremenu. Ako mora da se čeka više od toga, onda je poželjno da bude na neki način obavešten o razlogu zbog koga se čeka. Na slici 4.3 se sa leve strane vidi vodoravna crta koja vizuelno obaveštava koliko još ima nedovršenog posla zbog čega mora da se čeka. Sa takvom informacijom su manje šanse da korisnik pokuša da „popravi“ sistem ako mu se učini da se dogodila neka greška zbog predugog čekanja.



Slika 4.3. Progress bar (levo); brojilo u čekaonici (desno) [3]

U čekaonici je isto poželjno da se vidi na kom je mestu dati korisnik da bi mogao da proceni koliko dugo još mora da čeka. Kao što se vidi na slici 4.3 sa desne strane, ovakva vrsta informacije je korisna i za bolje funkcionisanje šaltera jer korisnici tačno znaju na koji šalter da pristupe kada dođe red na njih. Ovo su samo dva tipična primera obaveštavanja korsinika o stanju, ali ovaj problem mora da se reši u dizajnu bilo kog proizvoda za dati kontekst korišćenja.

4.3.2. Paralela između sistema i sveta oko nas

Moguće akcije koje korisnik može da obavi u sistemu treba da budu svedene na korake koji bi inače bili obavljeni i bez korišćenja sistema, ukoliko je to moguće. On mora da lako i intuitivno prepozna koje dugme ili komanda obavlja specifičnu funkciju u sistemu do te mere da čak i ako nikada nije video niti koristio sistem do tog trenutka, treba da se lako sam snađe bez ičije pomoći. Na primer, slika 4.4 prikazuje prost interfejs koji ima minus i plus dugme. Prozor se naziva „Magnifier“, i njegova ikonica je lupa preko ekrana. Logično je da će korisnik da pomisli da će nešto da se poveća na prozoru ili smanji ukoliko klikne na plus ili minus, iako možda nikada nije video ovaj prozor do sada. Bilo koja akcija će se brže i bolje razumeti u programu ako je može povezati sa nečim poznatim.

Slika 4.4. Povećavanje ili smanjivanje prozora [4]

4.3.3. Korisnička kontrola i sloboda

Korisnici često i slučajno izaberu sistemske funkcije koje ne obavljaju željenu operaciju, pa se nađu u situaciji gde žele da se vrate na prethodno stanje, ili na prethodni meni. Ovakve situacije zahtevaju jasno obeleženu opciju za izlaz ili povratak na prvobitno stanje, i one se kolektivno zovu navigacione opcije korisničkog interfejsa. Najpopularniji načini obavljanja osnovnih navigacionih funkcija se vide na slici 4.5. One su (1) povratak na prethodno stanje ili ponavljanje prethodne akcije, ili popularnije „undu“ i „redo“. Dozvoljavaju korisniku da ide nekoliko koraka unazad ili unapred, a (2) „close“ ili „exit“ se automatski razume kao ikonica za gašenje trenutnog prozora ili aplikacije i ne mora da bude crvene boje, jer je slovo „x“ sinonim za ovu funkciju. Konfigurisanje aplikacije u Android operativnom sistemu se tipično započinje ikonicom koja sadrži zupčanike (3). Donji red slike 4.5 predstavlja navigaciju bilo kog pametnog telefona, jer sadrži tri najpopularnije navigacione funkcije, a one su (4) „nazad“, kada korisnik želi da se vrati na prethodni meni, (5) „home“ ili direktni povratak na početni meni, i (6) „windows“, ili prozori, koja prikazuje sve otvorene aplikacije.

Slika 4.5. Otkaži i ponovi (1), zatvori (2), konfiguracija (3),
nazad (4), početni ekran „home“ (5), otvorene aplikacije (6)

Sve ikonice navedene na slici 4.5 su univerzalno prhvaćene u raznim operativnim sistemima i raznim aplikacijama i dozvoljavaju korisniku da intuitivno koristi aplikacije koje do sada nije video. Ne samo da je njihov lik poznat korisnicima, nego i relativna pozicija na ekranu. Donji red slike 4.5 se tipično pojavljuje na dnu svih pametnih telefona.

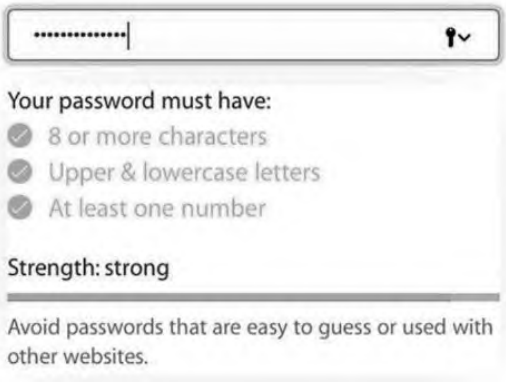
4.3.4. Doslednost i standardi

Korisnici ne smeju da se dvoume da li situacije, reči ili akcije imaju isto značenje u raznim kontekstima prilikom korišćenja programa. Ovde proširujemo dosledne univerzalne standarde navigacije iz prethodne sekcije, jer postoje dodatne standardne funkcije kao što je logovanje na sajt, gde korisnik očekuje da mu je to tipično omogućeno u gornjem desnom uglu, i da se to uvek zove „login“, ili „sign in“. Glavni navigacioni meni na bilo kom web sajtu mora da bude jednako formatiran, i da se pojavljuje na istom mestu na svim stranicama istog sajta radi lakšeg korišćenja. Ako dugmad imaju određen format, izbor fonta, ili specifičnu boju, onda moraju da budu isto formatirani na celom sajtu da bi ih korisnik lakše prepoznao.

Sajt mora da koristi koncepte i reči koji se univerzalno razumeju i prihvataju i na drugim sajtovima širom Interneta. Na primer, vrlo je teško uvesti nove reči koje menjaju sledeće vrlo dobro poznate akcije: sačuvaj ili “save”, otvori ili “open”, zatvori ili “close”, i lajkuj ili “like”. Jednostavno nema konkurencije u dominaciji ovih termina i bespotrebno i besmisleno je pokušati ih zameniti. U suprotnom će se korisnik zbuniti, i teže moći da se snađe na sajtu.

4.3.5. Prevencija grešaka

Bolje čak i od dobrih poruka koje javljaju korisniku o načinjenim greškama su metode koje mu ne dozvoljavaju da ih napravi. Ovo se postiže tako što se korisnik obavesti u realnom vremenu o potencijalnim problemima i pruži mu se mogućnost da ih reši sam pre nego što nastavi dalje. Na primer, na slici 4.6 se vidi jedna web forma koja mu dozvoljava da izabere, odnosno ukuca svoju lozinku.

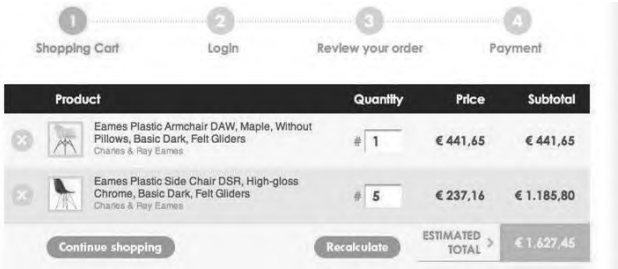


Slika 4.6. Obaveštavanje korisnika o bezbednosti lozinke [5]

Kada korisnik uspe da ispuni uslove za dužinu lozinke (najmanje 8 karaktera), ubaci barem jedno veliko i jedno malo slovo, i barem jedan broj, dozvoljeno mu je da pređe na sledeći korak. Na dnu forme je takođe obavešten o jačini lozinke, gde pozeleni horizontalna indikaciona linija kada se ispune svi navedeni uslovi. Bez ovakvih obaveštenja bi forma verovatno vraćala korisnika da ponovo izabere jaču lozinku, ali on ne bi znao na koji način da je popravi. Ukratko, prevencija grešaka je bolja nego korekcija, i postiže se kontrolom postupka korisnika, kao i smernicama kada je na putu da napravi grešku, i sugestijama kako da je izbegne.

4.3.6. Prepoznavanje je bolje od pamćenja

Minimizirajte pamćenje korisnika tako da objekti, akcije i opcije budu lako uočljivi. Korisnik ne treba da pamti informaciju sa jednog prozora kad prelazi na sledeći, i uputstva za korišćenje sistema treba da budu vidljiva ili barem da se lako pronađu kada mu zatrebaju. Na slici 4.7 se vidi tipiča forma za naručivanje proizvoda na Internetu. Korisnik je u ovom trenutku već izabrao dva prioizvoda, koja su stavljena u njegovu virtuelnu korpu, i on je sada u postupku plaćanja. Prvi korak plaćanja je da proveri šta se nalazi u korpi. Obratite pažnju da se ovde ne zahteva od korisnika da pamti ono što je naručio, nego već samo da prepozna i potvrdi već izabrano.



Slika 4.7. Forma za naručivanje proizvoda na Internetu

Web pretraživači imaju sličan pristup kada korisnik počne da ukucava narednu pretragu. Kada zaboravi ili permutuje neko slovo, ispravne reči se protumače i one se proslede pretraživaču. Google čak ume da nam “čita misli“, odnosno da nam nudi vrste pretrage na osnovu samo nekoliko upisanih reči, kao što se vidi na levoj strani slike 4.8. Sa desne strane iste slike se vidi spisak ponuđenih fontova iz Microsoft Word-a. Ovde primećujemo da je ime svakog fonta ispisano u svom fontu jer na ovaj način korisnik ne mora da pamti kako svaki izgleda pre nego što ga izabere.

Slika 4.8. Google pomoć u pretraživanju sa sugestijama (levo);
spisak fontova iz Microsoft Word-a (desno)

Bitno je uvek razmisliti kako korisnik upotrebljava određeni deo aplikacije, i osmisliti način da što manje mora da pamti.

4.3.7. Fleksibilnost i efikasnost korišćenja

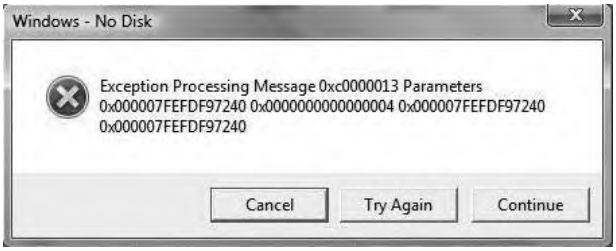
Potrebno je razmisliti o nivou korisnika, odnosno o njegovom iskustvu sa datim programom. Ekspertski korisnici se definišu kao oni koji često koriste određeni program i osećaju da to spretno rade. Ovoj grupi treba ponuditi opcje da brže odrade bilo koji deo posla uz pomoć skraćenica ili nekog alternativnog postupka. Dobar dizajn programa uključuje mogućnost da se isti posao obavi na razne načine. Na primer, u Microsoft Windows-u, bilo koji prozor može da se isključi na nekoliko načina (klikom na „x“ u gornjem desnom uglu, ALT + F4, CTRL + W itd.), jer svaki korisnik ima omiljen način rada. Neki načini rada više odgovaraju iskusnijem korisniku, i takvi načini moraju da postoje da se on ne bi nepotrebno nervirao, radeći stvari na isti spori način kao i početnici. Skraćenice kao za “copy” (CTRL + C) i “paste” (CTRL + V) su univerzalne, i korisnici očekuju da mogu da na ovaj način kopiraju i zalepe nešto u bilo kojoj aplikaciji, što znači da ne smeju da se koriste za bilo koju drugu funkciju, jer će na taj način sigurno doći do greške zbog neočekivanog ponašanja programa.

4.3.8. Estetika i minimalistički dizajn

Korisnički interfejs treba da jedino sadrži elemente koji su relevantni korisniku za obavljanje specifičnog posla. Prozori ne treba da sadrže nepotrebne ili retko korišćene podatke, i svaki nepotrebni detalj skreće pažnju sa korisnih informacija i umanjuje njihovu vidljivost. Najbolji primer minimalističnog dizajna na Internetu je upravo google.com, koji usmerava svu pažnju korisnika na funkciju pretraživanja, koja i jeste poenta celog sajta. Ako korisnik želi da sazna više, ili želi da obavi napredniju pretragu, koriste se neupadljivi linkovi koji su razbacani po ivicama stranice. Jedina stvar koja se menja na sajtu jeste povremeni dizajn logoa firme „Google“, ali to ne smeta korisnicima da obave pretragu.

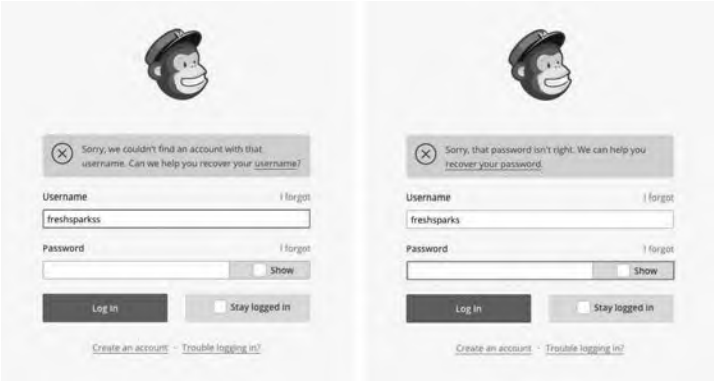
4.3.9.Prepoznati grešku, postaviti dijagnozu i oporavak od greške

Koliko god da je sistem isproban i testiran, korisnici će ga još dublje i temeljnije isprobati i sigurno pronaći određene greške. Dizajn dobrog sistema ili softvera podrazumeva da postoji mehanizam koji „hvata“ greške i na detaljan način ume da objasni korisniku zašto je došlo do njih, i šta može da učini da ih u daljem radu izbegne. Slika 4.9 predstavlja tipičan primer kako ne treba predstaviti grešku korisniku. Sa ovakvom porukom bi se čak i programeri teško snašli, pa možete da zamislite šta običan korisnik može da pomisli kada vidi nešto slično. Greške treba objasniti što prostijom terminologijom i pružiti korisniku mogućnost da se izvuče iz takve situacije sa što manje posledica. Glavna posledica od koje strepi svaki korisnik jeste da ne izgubi svoj rad, što znači da bilo koji oporavak od greške mora to da uzme u obzir.



Slika 4.9. Loš primer poruke sa greškom [6]

Dobar primer oporavka od greške se vidi na slici 4.10, gde se nalazi naslovna strana imejl servisa mailchimp.com. U ovoj situaciji je korisnik pogrešno uneo svoje korisničko ime (levo) i svoju lozinku (desno). Sistem u ovim situacijama obavesti korisnika da nešto nije u redu, i onda ponudi rešenje u vidu pomoći oko zaboravljene lozinke, ili korisničkog imena. Ne postoje suvišne informacije na stranici za logovanje, što dozvoljava korisniku da se lako snađe i reši problem.



Slika 4.10. Dobar primer poruke sa greškom [7]

4.3.10. Pomoć i dokumentacija

Uvek je bolje ako korisnici mogu da se snađu sa softverom bez uvida u dokumentaciju, ali postoje situacije u kojima nema alternative nego da se pročita uputstvo. Ono mora biti ispisano na način koji sledi već definisane principe dizajna, tako da korisnici lako mogu da pronađu to što ih zanima i da uspešno slede uputstva da bi postigli svoje ciljeve. Ne mora biti u vidu knjige ili opširnog teksta, već korisnik može da bude usmeren na video-snimak, ili čak na poruku koja iskoči preko ekrana kada se miš zadrži na mestu dileme. Bitno je da on može lako da vidi mesto gde treba da klikne na sekciju za pomoć, ili znak "?" po potrebi.

4.4. PRIMERI LOŠEG DIZAJNA

Nažalost, loš dizajn nije limitiran na softverske proizvode, nego se nalazi svuda oko nas, i pogotovo u našem delu sveta. Mi smo prilagodljiv narod pa se naviknemo na sve, ali ne znači da ne primetimo, i da ne pomislimo da nešto sigurno može da se promeni i poboljša. Upustićemo se prvo u analizu ambalaže nekih domaćih, i to vrlo ukusnih proizvoda koji se vide na slici 4.11.

Slika 4.11. Domaći proizvodi

Bez obzira na ukus, proizvođači imaju očiglednih problema sa dizajnom ambalaže jer se ne stave na mesto korisnika koji mora da se „bori" s njom. U gornjem levom uglu slike 4.11 nalazi se tečni sir "Tufi", koji je pakovan pod jakim pritiskom. Nema očigledan i elegantan način otvaranja, pa su potrošači ograničeni na to da probuše plastičnu ambalažu nožem, makazama, iglom, što često izazove neproporcionalnu kontrareakciju sira koji izlazi punom snagom (često i po zidovima...). Ako potrošač kojim slučajem ne pojede čitavu tubu sira, onda ostatak može da iscuri u frižideru jer ona nema ni

elegantan način zatvaranja. Sličan problem imamo i sa majonezom u donjem levom uglu, jer nakon otvaranja (opet makazama) nepotrošeni deo u otvorenoj ambalaži treba posebno pažljivo namestiti u frižideru da ne bi došlo do rasipanja. Trouglasti sir u gornjem desnom uglu ima dobru zamisao, ali lošu realizaciju. Često on može da se otvori povlačenjem crvenih plastičnih „konaca“ koji u teoriji seku ivice metalne ambalaže i dozvoljavaju nam da dođemo do željenog proizvoda. Realnost je nešto drugačija od teorije, i obično rezultira pokidanim crvenim tračicama, što vodi do mukotrpne hirurške intervencije koja optimizuje funkciju čuvanja što većeg dela sira uz što manje prljanja prstiju. U donjem desnem ugalu je popularni jogurt u kartonskoj ambalaži, sa neprobojnim plastičnim poklopcem ispod čepa. Da bi se došlo do lekovitog jogurta, često se žrtvuje barem jedan nokat u odmeravanju snage sa drugim slojem borbenog plastičnog poklopca koji mora da se svom snagom vuče i kida. Sigurno postoji način da se reše ovi problemi, ali samo ako proizvođači stave korisnika u jezgro procesa dizajna...

Problemi u dizajnu nisu ograničeni samo na proizvode, nego se mogu primetiti i na putevima. Na slici 4.13 se vide načini konstruisanja raskrsnice na raznim krajevima sveta. Počinjemo sa leve strane slike 4.13, gde se vidi tipična raskrsnica u Srbiji. Obratite pažnju na poziciju semafora - vozač mora da iskrivi vrat gledajući direktno iznad, ili desno, da bi video svetlo, ako ima tu nesreću da bude prvi u koloni i mora da stane na široku belu crtu. Taj problem je rešen u Americi, čija se tipična raskrsnica vidi na sredini slike 4.13. Tamo su semafori sa suprotne strane ulice, tako da se lako vidi svetlo bez obzira na poziciji u koloni, iako se saobraćaj zaustavlja na približno isto pozicioniranim belim crtama kao i u Srbiji.

Slika 4.12. Raskrsnica u Srbiji (levo) [8]; raskrsnica u Americi (sredina) [9]; raskrsnica u Japanu (desno) [10]

Sa desne strane slike 4.13 vidimo poznatu “Ginza” raskrsnicu u Tokiju, gde su semafori sa suprotne strane ulice, kao i u Americi, ali je dodata još jedna funkcija za pešake, a ona im dozvoljava da dijagonalno pređu raskrsnicu. Ovo je prilično popularan deo grada za pešake, pa je ovaj koncept uveden da bi bio što veći protok pešačkog saobraćaja.

4.5. PROCES PLANIRANJA NOVOG INTERFEJSA

Kada zamišljamo novi proizvod ili novi interfejs za proizvod, uvek treba da se pitamo šta on želi da postigne, a možda jednako bitna su pitanja ko/kako/gde i u kom kontekstu će ga koristiti. Reč interfejs znači način korišćenja, i tipično je povezan sa kompjuterskim proizvodima, ali se odnosi i na generalne proizvode. Počinjemo sa razjašnjavanjem ko će koristiti novi proizvod, odnosno koja demografska grupa, ili za koji profil ljudi je proizvod zamišljen. Da li je za decu, odrasle, tinejdžere, penzionere, invalide? Svaka demografska grupa ima svoje profesionalne i privatne navike, kao i preferencije, i, naravno, ograničenja, koja se moraju uzeti u obzir kada se osmišljava rešenje za njih. Pored toga, razjasni se gde se koristi proizvod, odnosno interfejs. Da li je unutra, napolju, u svemiru, na kiši, vetru, jakom suncu, na visokim temperaturama, niskim, uz wi-fi ili GPRS? Način i okolnosti korišćenja diktiraju funkcionalnost uređaja. Dodatni faktori koji mogu znatno da utiču na prihvaćenost novog rešenja, i stepen njegove upotrebe jesu razlike u kulturi između dizajnera i krajnjeg korisnika. Na primer, određene reči, znaci, boje, pa čak i brojevi imaju potpuno drugačije značenje ili konotaciju u drugim jezicima i kulturama, pa se moraju pažljivo birati da ne bi uvredili potencijalnog korisnika u drugom delu sveta.

Slika 4.13. Korisnik u jezgru dizajna

Faktori kao što su nivo privatnosti, stres na radnom mestu (bolnice), pa čak i dosada (portiri), mogu znatno da utiču na dizajn korisničkog interfejsa. Potrebno je analizirati podatke vezane za kontekst korišćenja, kao i za potrebe korisnika da bi upotrebljivost interfejsa zaista bila na visokom nivou. Na slici 4.12 se vidi ciklus dizajna bilo kod korisničkog interfejsa. Kada se razume kontekst korišćenja i zahtevi korisnika, osmisli se dizajn i testira se. Ako dizajn ne ispunjava očekivanja korisnika, on se popravi, pa se opet testira. Ovaj ciklus se ponavlja sve dok test grupa ciljnih korisnika ne bude zadovoljna.

4.5.1. Radno okruženje

Razni elementi spadaju u opis radnog okruženja, ali se tipično odnose na uticaj okruženja na uslove rada, i kategorišu se kao fizičke, tehničke i socijalne okolnosti. Fizičke okolnosti se odnose na lokaciju radnog mesta, i svode se na to da li je rad unutra ili napolje, na uticaj klime (vetar, kiša, temperatura) ili osvetljenja na rad. Primer fizičkog utijaca na rad je bankomat, koji može da radi i unutra i napolje. Da li treba da bude drugačije koncipiran ako je zamišljen da radi napolje, i na koje karakteristike tada treba obratiti pažnju?

Tehničke okolnosti opisuju tehnologije i mašinerije koje već učestvuju u radu, i mogu biti, na primer, desktop kompjuter, laptop, ekran na dodir, mobilni telefon, veličina ekrana itd. Ljudi se naviknu na tehničke okolnosti svog radnog mesta, i često bude teško da prihvate novo rešenje, čak i ako je superiorno. Radnici u vatrogasnim službama širom sveta su navikli da koriste voki-toki uređaje, koji imaju svoje mane: nemaju mogućnost ponavljanja komandi, svako na mreži čuje svaku komandu, što ometa u radu, kvalitet audio-zapisa nije na zavidnom nivou... Kada im je predstavljen novi uređaj koji rešava ove probleme, ali se koristi na potpuno drugačiji način, oni su odbili da ga koriste jer im je navika bila bitnija od tehnološkog napretka [11].

Socijalne okolnosti mogu da navedu ljude da odbiju određene proizvode. Broj 4 u Kini ima konotaciju jednaku smrti, pa se ljudi tamo trude da nemaju ovu cifru u svom broju telefona, da ne stanuju na 4. spratu, i slično. Ako im se ponudi proizvod koji sadrži broj 4 u imenu, manja je verovatnoća da će taj proizvod uspeti na tržištu. Slično važi za boje, pa čak i znakove. Popularna „vegeta“ ima logo sa kuvarom koji rukom pokazuje znak “ok” na svakoj kesici proizvoda, ali taj kuvar ne bi prodao nijednu u Brazilu tim gestom jer je tamo to vrlo uvredljiv znak. Dosada je još jedan socijalni faktor koji utiče na dizajn proizvoda, jer radnici koji moraju da gledaju skoro nepromenjenu sliku na ekranu po čitav dan (portiri, kontrolori nuklearnih pogona) mogu lako da zbog dosade propuste vrlo bitne događaje.

4.5.2. Korisnici

Nivo korisnika može da se odnosi na nivo spretnosti ili iskustva sa uređajima ili interfejsima, ali u ovoj situaciji se odnosi na uticaj date aplikacije na život korisnika. Na primer, zamislite aplikaciju koja pomaže korisnicima da se snađu u gradskom saobraćajnom prevozu. Prikazuje pozicije autobusa na mapi grada, ažurira rasporede autobusa, i obeležava buseve koji kasne. Primarni korisnik je taj koji direktno pristupa aplikaciji i spada u ciljnu grupu za koju je proizvod napravljen. Ciljna grupa za ovu aplikaciju uključuje sve putnike GSP-a, kao i potencijalno vozače autobusa. Sekundarna grupa korisnika se sastoji od osoba koje zavise od primarnih korisnika. U ovom primeru, oni mogu biti članovi porodice koji čekaju putnike autobusa (primarne korisnike). Takođe mogu biti

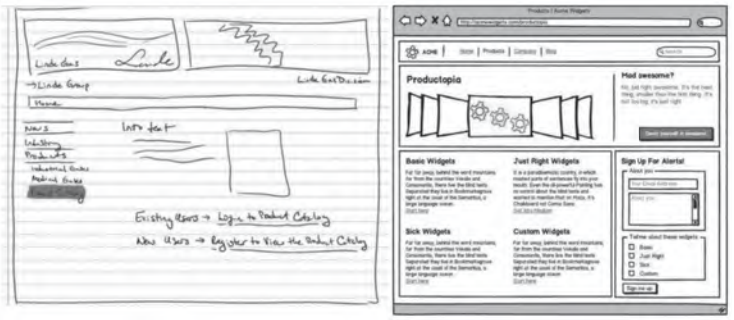
i menadžeri, programeri ili dizajneri prvobitne aplikacije. Tercijarni korisnici su entiteti koji zavise od uspeha projekta, i uključuju investitore i glavne menadžere projekta. Svako iz ovih grupa ima svoj interes, ali se projekat fokusira na to da primarni korisnici budu zadovoljni.

Kada se zamišlja kako da se napravi nova verzija softvera ili interfejsa, treba jasno konstruisati situaciju u kojoj trenutna verzija ne zadovoljava potrebe korisnika. Drugim rečima, treba jasno odgovoriti na pitanja: "Šta želi da postigne nova tehnologija, ili novi interfejs?". Na ovo pitanje se specifično odgovara tako što se konstruiše realan scenario, gde postoji tipičan primarni korisnik koji ne može da reši svoj problem uz pomoć trenutne tehnologije. Tipičan primarni korisnik, ili lice, izvuče se iz profila svih korisnika koji inače koriste datu aplikaciju, da bi se dizajneri, programeri i menadžeri osećali da rešavaju konkretan korisnički problem. Profil svih korisnika se opisuje manje-više kao i kriminalni profil kada policija traži osumnjičenu osobu. U našem primeru sa GSP aplikacijom, profil korisnika bi mogao da se svede na uzraste 13-18 godina, sa navikom da koristi mobilni telefon za skoro sve, koji živi na periferiji grada, i obično se vozi autobusom. Ovo opisuje grupu korisnika za koju se pravi aplikacija. Konkretno lice iz ove grupe može da ima sledeći opis: Petar Petrović, 16 godina, srednjoškolac, putuje autobusom do škole svaki dan, vikendom posećuje prijatelje. Petar živi na periferiji grada, i putuje sa mlađom sestrom i drugarima do škole. Voli da koristi kompjuter, laptop, mobilni svakodnevno, i mora da se budi rano da pomogne sestri da se spremi za školu. Autobusi ne cirkulišu često u njegovom kraju, i signal na telefonu mu često bude ispod tri crte.

Aplikacija mora da se dizajnira da bude od koristi Petru i njegovoj specifičnoj situaciji. Ovakvim detaljnim opisom jedne fiktivne osobe detaljnije se zađe u problem, što rezultira boljim rešenjem. Scenario bi opisao tipičnu situaciju Petrovog korišćenja trenutne aplikacije, naglašavajući gde ona ne uspeva. Na primer, trenutna aplikacija možda zahteva mnogo jači signal na mobilnom telefonu nego što Petar ima na raspolaganju, ili ne prikazuje rezultate na način na koji je navikao. Novo rešenje bi pomoglo Petru u njegovom kontekstu korišćenja aplikacije. Što više scenarija, i što više lica iz ciljne grupe vode do mnogo robustnije aplikacije, koju može da upotrebi širi spektar korisnika.

4.5.3. Prototip

Kada se definišu zahtevi korisnika i zabeleže razni scenariji u korišćenju proizvoda, faza pravljenja prototipa počinje. Proces obično započinje prototipom gde se ceo koncept korišćenja, sa svim prozorima, definiše na papiru u vidu crteža, sa post-it nalepnicama radi bržeg korigovanja. Ovaj način je najbrži za koncipiranje i brzo, idejno testiranje.



Slika 4.14. Papirnati prototip (levo); Balsamiq prototip (desno) [12]

Naredni korak, koji se vidi na slici 4.14 pravljenje je softverskog koncepta interfejsa, uz pomoć programa kao što je Balsamiq [12]. Balsamiq dozvoljava linkovanje stranica, relativno pozicioniranje elemenata, kao i uključivanje svih vrsta tipičnih kontrola na sajtu i dozvoljava kasnije ubacivanje grafika i ostalih elemenata da bi se dovršio interfejs. Poslednji korak u dizajnu interfejsa jeste implementacija pravog interfejsa sa svim funkcijama, i izgledom pravog, završenog proizvoda. Završeni interfejs može da pređe na fazu korisničkog testiranja pre nego što bude pušten u rad sa pravim korisnicima.

4.5.4. Testiranje

Interfejs se testira u laboratoriji sličnoj onoj na slici 4.15. Snimaju se lice, telo, pokreti, i beliži se položaj miša i očiju u svim trenucima. Subjekti koji testiraju novi softver dobiju seriju zadataka koje treba da reše koristeći novi interfejs. Meri se vreme do završetka svakog zadatka, kao i broj grešaka koje naprave na putu do cilja. Novi softver testiraju i početnici, kao i ekspertski korisnici, jer svaka grupa može da otrkije greške specifične za nju. Cilj ove vrste testiranja jeste da se skrati korisničko vreme do cilja, i da se minimizira broj grešaka. Na kraju testa subjekti dobiju upitnik gde ocenjuju svoje iskustvo sa upravo testiranim proizvodom, estetikom, kao i nivo zadovoljstva dok su ga koristili. Ova vrste povratne informacije dozvoljava dizajnerima i programerima da poboljšaju slabe tačke interfejsa pre nego što dođe do tržišta.

Slika 4.15. Laboratorija za testiranje korisničkih interfejsa

4.6. KLJUČNI TERMINI

Upotrebljivost – stepen do kojeg grupa korisnika može da upotrebi proizvod da bi postigli odgovarajuće ciljeve efektivno, efikasno i na zadovoljavajući način u specifičnom kontekstu.

Ergonomija – naučna oblast koja se bavi dizajnom proizvoda tako da oni najbolje budu prilagođeni ljudskom telu.

Prototip – prvi primerak interfejsa koji se dalje testira i rafiniše pre puštanja u serijsku proizvodnju za korišćenje na tržištu.

Interfejs – dodirna tačka korisnika i kompjutera.

Profil – demografski opis ciljne grupe.

Lice – izmišljeno fizičko lice iz skupa svih mogućih korisnika.

Scenario – priča o tipičnom korisniku koji koristi trenutnu tehnologiju/interfejs, i opisuje gde trenutna tehnologija/interfejs ne uspeva da postigne ciljeve.

Primarni korisnik – onaj sa direktnim pristupom interfejsu, za kojeg je i napravljen.

Sekundarni korisnik – onaj koji zavisi od primarnog korisnika ili učestvuje u izgradnji interfejsa.

Tercijarni korisnik – entitet koji zavisi/računa na uspeh projekta.

Pitanja za ponavljanje

1. Šta je upotrebljivost?
2. Gde se koristi ergonomija?
3.Koji su principi i mere upotrebljivosti?
4. Gde se koristi minimalistički dizajn?
5. Šta je prototip i koje vrste prototipa postoje u dizajnu korisničkog interfejsa?

6. Koji su nivoi korisnika i čemu služe?
7. Koja je razlika između profila i lica?
8. Šta znači „korisnik u jezgru dizajna”?
9. Koje su moguće okolnosti rada koje utiču na dizajn proizvoda ili interfejsa?
10. Kako utiču radne navike korisnika na dizajn novog interfejsa?
11. Navedite sve principe upotrebljivosti.
12. Zašto je minimalistički dizajn bitan u polju HCI?

Literatura

[1] S. Cochran, „Unusual Seating Options Complement Productivity,“ 9 6 2011. [Na mreži]. https://www.brighthub.com/office/home/articles/69470.aspx. [Poslednji pristup 7 2018].

[2] Wikipedia, „The Mother of All Demos,“ 1 8 2018. [Na mreži]. https://en.wikipedia.org/wiki/The_Mother_of_All_Demos. [Poslednji pristup 8 2018].

[3] D. Ogden, „The DMV.....,“ 20 3 2012. [Na mreži]. http://thehappynewslady.blogspot.com/2012/03/dmv-by-diane-ogden.html. [Poslednji pristup 8 2018].

[4] Microsoft, „Animations and Transitions,“ 30 5 2018. [Na mreži]. https://docs.microsoft.com/en-us/windows/desktop/uxguide/vis-animations. [Poslednji pristup 8 2018].

[5] N. Babich, „Golden Rules of User Interface Design,“ UX Planet, 10 2 2016. [Na mreži]. https://uxplanet.org/golden-rules-of-user-interface-design-19282aeb06b. [Poslednji pristup 8 2018].

[6] Steve, „Anatomy of an Error Message – A Windows Vista Example,“ Still Life, 2 4 2009. [Na mreži]. https://steveswanson.wordpress.com/tag/bad-error/. [Poslednji pristup 8 2018].

[7] S. Duggirala, „10 Usability Heuristics with Examples,“ Prototypr, 18 8 2016. [Na mreži]. https://blog.prototypr.io/10-usability-heuristics-with-examples-4a81ada920c. [Poslednji pristup 8 2018].

[8] SrbijaDanas, „PROVERA VOZAČA! Ko ima prvenstvo na raskrsnici ako semafor ne radi?,“ 12 09 2017. [Na mreži]. https://www.srbijadanas.com/auto/vesti/provera-vozaca-ko-ima-prvenstvo-na-raskrsnici-ako-semafor-ne-radi-video-2017-09-12. [Poslednji pristup 8 2018].

[9] M. Draper, „Intersections have county council’s attention,“ The Reporter, 3 7 2018. [Na mreži]. http://porthawkesburyreporter.com/intersections-have-county-councils-attention/. [Poslednji pristup 8 2018].

[10] Scoopy Web, „These Are the Safest Cities to Live in Around the World," 2 2018. [Na mreži]. http://www.scoopyweb.com/2018/02/these-are-safest-cities-to-live-in.html. [Poslednji pristup 8 2018].

[11] M. Stojmenovic i G. Lindgaard, „Probing PROBE: A field study of an advanced decision support prototype for managing Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) events," u *Proceedings of the 11th International ISCRAM Conference*, Pennsylvania, USA, 2014.

[12] Balsamiq, „Balsamiq," Balsamiq, 1 2008. [Na mreži]. https://balsamiq.com/. [Poslednji pristup 8 2018].

[13] Asteria Expeditions, „ONTDEK DE MEEST FASCINERENDE VOGEL TER WERELD," 25 11 2015. [Na mreži]. https://www.asteriaexpeditions.be/poolexpedities/antarctica-keizerspinguinsafari-9-daagse/keizerspinguin.asp. [Poslednji pristup 6 2018].

[14] R. B. Sherley, T. Burghardt, P. J. Barham, ,. N. Campbell i I. C. Cuthill, „Spotting the difference: towards fully-automated population monitoring of African penguins Spheniscus demersus," *Endang Species Research,* t. 11, br. 2, pp. 101-111, 2010.

[15] SENSIP, „Arizona State University," [Na mreži]. https://sensip.engineering.asu.edu/research/image-and-video-processing/. [Poslednji pristup July 2018].

[16] P. Viola i M. Jones, „Robust Real-time Face Detection," u *International Conference on Computer Vision*, 2001.

[17] The Guardian, „From Ale to Jail: facial recognition catches criminals at China beer festival," *The Guardian,* 2018.

[18] Wikipedia, „Integrated Automated Fingerprint Identification System," [Na mreži]. https://en.wikipedia.org/wiki/Integrated_Automated_Fingerprint_Identification_System. [Poslednji pristup 7 2018].

[19] W. Yang, J. Hu i M. Stojmenovic, „An alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbor structures," *Pattern Recognition,* t. 47, br. 3, p. 1309–1320, 2014.

[20] NeuroTechnology, „Verifinger," [Na mreži]. http://www.neurotechnology.com/verifinger.html. [Poslednji pristup 7 2018].

[21] Wikipedia, „Brandon Mayfield," [Na mreži]. https://en.wikipedia.org/wiki/Brandon_Mayfield. [Poslednji pristup 6 2008].

[22] S. L. Zabell, „Fingerprint Evidence,," *Journal of Law and Policy (Brooklyn College Law School),* t. 143, br. 77, 2005.

[23] J. Daugman, „How Iris Recognition Works," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,* t. 14, br. 1, 2004.

[24] Z. Wendt i J. S. Cook, „Saved by the Sensor: Vehicle Awareness in the Self-Driving Age," Machine Design, 17 1 2018. [Na mreži]. http://www.machinedesign.com/motion-control/saved-sensor-vehicle-awareness-self-driving-age. [Poslednji pristup 7 2018].

[25] Nvidia, „Autonomous Driving," Nvidia, [Na mreži]. https://www.nvidia.com/en-us/gtc/topics/autonomous-driving/. [Poslednji pristup 7 2018].

[26] Wikipedia, „Autonomous car," 7 2018. [Na mreži]. https://en.wikipedia.org/wiki/Autonomous_car. [Poslednji pristup 7 2018].

[27] MIcrosoft, „Color," 31 5 2018. [Na mreži]. https://docs.microsoft.com/en-us/windows/desktop/uxguide/vis-color. [Poslednji pristup 7 2018].

[28] SharkD, „HSL color solid dblcone," [Na mreži]. https://commons.wikimedia.org/wiki/File:HSL_color_solid_dblcone.png. [Poslednji pristup 7 2018].

[29] C. Blakemore i G. F. Cooper, „Development of the brain depends on the visual environment," *Nature,* t. 228, br. 5270, pp. 477-478, 1970.

[30] Wikipedia, „Sobel Operator," [Na mreži]. https://en.wikipedia.org/wiki/Sobel_operator. [Poslednji pristup 7 2018].

[31] L. Lagerqvist, „Camera Anatomy," 3 11 2010. [Na mreži]. https://lovisalagerqvist.wordpress.com/tag/anatomy/. [Poslednji pristup 6 2018].

[32] Rensselaer Polytechnic Institute, Department of the Arts - School of HASS, „Photographz Tips," [Na mreži]. http://www.arts.rpi.edu/~ruiz/idispring09/photography%20tips.htm. [Poslednji pristup 7 2018].

[33] Wikipedia, „Gustave Le Gray," [Na mreži]. https://en.wikipedia.org/wiki/Gustave_Le_Gray. [Poslednji pristup 7 2018].

5. OBRADA SLIKE I KOMPJUTERSKA VIZIJA

Da li kompjuteri mogu da gledaju i razumeju to što vide? Da li mogu da donose inteligentne odluke na osnovu onoga što vide? Kratak odgovor na ova pitanja je „naravno“, ali uz određene pretpostavke i ograničenja. Kompjuter može da se isprogramira da obavlja vrlo specifične zadatke u ovoj oblasti, ali ne može da vidi i razume sve u svim situacijama. Na primer, može da prepozna određen predmet ili klasu predmeta na slici ili video-snimku, ali vrlo retko može više od toga. Trenutno ne može da razume zašto su ti predmeti zajedno na toj slici, kako se koriste, niti koja je poruka koja se krije iza njihovog rasporeda na slici. Jednog dana će kompjuteri moći da dođu i do takvih zaključaka, ali sada je ova oblast na nivou na kojem može samo da nam pomogne u donošenju odluka, ne i da za nas odlučuje.

5.1. UVOD

Tema ove glave je spoj dve srodne oblasti, koje obavljaju komplementarne funkcije. Kompjuterska vizija je naučna oblast iz obrade signala, ili „signal processinga“, u kojoj je ulazni podatak slika ili video-snimak, a izlazni podatak slika, video-snimak ili skup otkrivenih karakteristika ili zaključaka iz ulaznih podataka. Na primer, na levoj strani slike 5.1 se vidi neobeležen skup pingvina. Sa desne strane je kompjuter uspeo da prepozna da su to pingvini na slici, pa ih je čak i individualno obeležio.

Slika 5.1. Prepoznavanje pingvina na slici [13]

Ne samo da kompjuter može da prepozna i obeleži životinje na slici, nego može u nekim okolnostima da raspozna koje je koji pingvin na osnovu rasporeda malih crnih tačaka na

njihovim stomacima. Jedan zoološki vrt u Engleskoj je prvi razvio ovu tehnologiju, ali se ona brzo proširila na sisteme posmatranja čitavih kolonija pingvina koji mogu da precizno i posebno prate kretanje čak 1.400 životinja [14]. Ova vrsta tehnologije pomaže etolozima u boljem razumevanju ponašanja životinja.

Obrada slike je oblast gde je ulazna informacija slika, a izlazna informacija opet slika, ali promenjena, ili na neki način poboljšana u odnosu na original. Slike se tipično obrađuju da bi bile lepše, razumljivije, ili da bi se nešto bolje videlo na njima nakon obrade, tj. manipulacije. Na primer, slika 5.2 predstavlja klasičan problem u obradi slike, a to je automatsko uklanjanje šuma, ogrebotina i nepoželjnih linija i regiona.

Slika 5.2. Uklanjanje nepoželjnih regiona slike [15]

Inače, obrada slike je jedan od prvih koraka ka bilo kom rešenju problema u kompjuterskoj viziji. Drugim rečima - slika mora da bude čistija i jasnije da bi ljudi mogli da je lakše razumeju, pa isto pravilo važi i za kompjutere.

5.2. PRIMENA KOMPJUTERSKE VIZIJE

Prva javna industrijska primena kompjuterske vizije pojavila se u oblasti bezbednosti, odnosno video-nadzora. Bezbednosne agencije zadužene za aerodrome početkom 2000-ih počele su da ulažu sredstva u razvoj tehnologije koja bi mogla da im pomogne da automatski prepoznaju osobe kojima je prilaz aerodromu strogo zabranjen. Kockarnice su u istom periodu imale sličnu ideju, ali je njihov cilj bio prepoznati i zabraniti prilaz kockarima koji uz pomoć prevara dolaze do dobitka, ili generalno previše zarađuju. Ova primena kompjuterske vizije može se podeliti na dva dela: pronalaženje lica na slici, i njegovo prepoznavanje. Pronalaženje lica na slici može se smatrati rešenim problemom [16], ali prepoznavanje lica i njegovo povezivanje sa osobom je aktuelna naučna tema. Softver je u poslednje vreme dovoljno napredovao da može sve preciznije da prepozna osobu na slici, u sve raznovrsnijim okolnostima, ali ipak najbolje funkcioniše

u kontrolisanim uslovima. Na primer, slike na Fejsbuku se lako automatski „taguju“ jer su osobe vrlo često okrenute ka kameri, stoje uspravno, slike su osvetljene, i osobe se fotografišu sa najbližim prijateljima, koji su često na njihovim spiskovima prijatalja. Ove kontrolisane okolnosti olakšavaju posao Fejsbuku da tačno obeleži, odnosno „taguje“ svaku osobu. Sa druge strane, policije širom sveta sve više koriste ovu vrstu tehnologije da bi identifikovale tražene osobe na javnim događajima, kao i na ulici. Na festivalu piva u Kvingdauu, u Kini, softver za prepoznavanje lica identifikovao je 25 traženih osoba, od kojih je jedna u bekstvu čak 10 godina [17].

5.2.1. Pronalaženje lica

Viola i Jones [16] su predstavili svoj sistem za pronalaženje lica 2001. godine, koji traži određene karakteristike lica na svakom mogućem regionu slike. Kada pronađe dovoljno karakteristika koje zajedno ukazuju na lice, sistem proglasi da je lice pronađeno. Na desnoj strane slike 5.3 se vidi primena ove ideje na jednom licu.

Slika 5.3. Skup „haar“ filtera (levo); primena „haar“ filtera radi pronalaženja lica [16] (desno)

Skup navedenih karakterisika koji pronalazi lice se vidi na levoj strani slike 5.3. Te karakteristike, odnosno „haar“ filteri, primenjuju se na specijalnim delovima slike kao što su oči, usta, nos itd. da bi proverili postojanje lica. Bez obzira na pol, rasu ili starost, lica imaju zajedničke karakteristike koje Violina metoda pokušava da identifikuje. Na primer, na desnoj strani slike 5.3 vide se dva primenjena „haar“ filtera koji ukazuju na to da je region oko naših očiju tamniji od regiona odmah ispod njih. Pored toga, region između očiju, odnosno malo iznad nosa, svetliji je od oba regiona koji opkoljavaju obrve. Ukoliko ova dva filtera ne ukazuju na lice, njihov region kandidat se odbacuje kao potencijalno mesto gde se nalazi lice. Ta dva filtera su dovoljna za čak 50 odsto odbacivanja potencijalnih lokacija lica na slici. Da bi se potvrdilo da se lice stvarno nalazi na slici u datom regionu, proverava se čak 6.000 sličnih filtera ove vrste. Predložen algoritam iz [16] je tačan u više od 99 odsto situacija i radi u realnom vremenu, kao što se vidi na našim telefonima, koji lako obeležavaju lica pre nego što ih mi fotografišemo.

5.2.2. Industrijske primene

Još jedna prirodna primena kompjuterske vizije je u vizuelnoj inspekciji komponenata u proizvodnji. Na primer, kada se proizvodi nova integrisana ploča za računar, vizuelno se može pregledati i pronaći bilo koje nesavršenosti pre mnogo skupljih i invazivnijih načina provere. Za štampani materijal, kao što je u dnevnom tiražu novina, koristi se kompjuterska vizija radi provere kvaliteta i grešaka u štampanju. U proizvodnji novih automobila koristi se automatska vizuelna inspekcija kvaliteta farbanja karoserije jer roboti su tipično zaduženi za taj deo posla. Čak i u proizvodnji eksploziva za miniranje starih zgrada koje su obeležene za demoliranje može se koristiti automatska vizuelna inspekcija u proizvodnji baruta za detonaciju. Kompanije „Google“ koristi tehnike iz obrade slike kada konstruiše skup podataka koji se koristi u njihovom „street view“. „Google“ je napredovao u tehnici spajanja slika koje se kreiraju u pokretu, jer njihovi „street view“ automobili se ne zaustavljaju da bi snimili okolinu u prolazu, nego kontinualnom vožnjom sastavljaju skupove slika.

U sportu se sve više koriste tehnike prikupljanja podataka koje zavise od kompjuterske vizije. Na fudbalskim, košarkaškim, odbojkaškim utakmicama kompjuter prati kretanje i akcije svakog igrača pojedinačno da bi treneri mogli da imaju više informacija o svojoj i protivničkoj ekipi. Prosti podaci koji se prikupljaju su broj kilometara koje je igrač pretrčao u fudbalu, broj uspešnih pasova... U košarci se tehnologija razvila do te mere da kompjuter može da „vidi“ i „razume“ individualne akcije ekipe, načine na koje ekipe igraju odbranu, kao i vrste šuteva svakog pojedinačnog igrača. Ovi podaci se koriste da bi se bolje razumela igra svake ekipe i da bi se pravio plan za narednu utakmicu. „Hawk-eye“ sistem, koji se koristi u tenisu, čista je primena kompjuterske vizije, gde kamere koja snimaju 1.000 slika u sekundi iz više uglova daju kompjuteru podatke koji služe da reše prostu matematičku jednačinu: da li aut linija ima presečnih tačaka sa elipsom koju formira loptica koja udara zemlju?

Kompjuterska vizija se koristi u poljoprivrednoj industriji u obliku alata koji analizira stanje biljaka na njivama koje se fotografišu satelitskim snimcima. Slika 5.4 predstavlja ovu ideju na terenu.

Slika 5.4. Infracrvena reprezentacija poljoprivrednih kultura iz satelita

Iz satelita se može prepoznati klasa kulture na njivi (kukuruz, soja, pšenica itd.), kao i njeno stanje, odnosno nivo oštećenja od poplave ili drugih nepogoda. Uz pomoć ovih informacija se može sprovesti procena prinosa svih kultura za datu godinu, što pomaže u odlukama o nivoima strateških rezervi.

5.2.3. Biometrija

Biometrija je nauka o statističkoj analizi bioloških podataka. Da bi se izvukli podaci iz biološke materije, često mora da se ona „vidi“ i „razume“, što spada u oblast kompjuterske vizije. Tipična primena biometrije je u analizi otisaka prstiju. Oni su najčešći forenzički dokazi koji se prikupljaju na bilo kom mestu zločina na svetskom nivou. Skoro svi sudski sistemi širom sveta su ih koristili radi identifikacije optuženih u proteklih 100 godina. U međunarodnim bazama otisaka prstiju, kao što su AFIS i IAFIS, postoje desetine miliona individualnih otisaka [18].

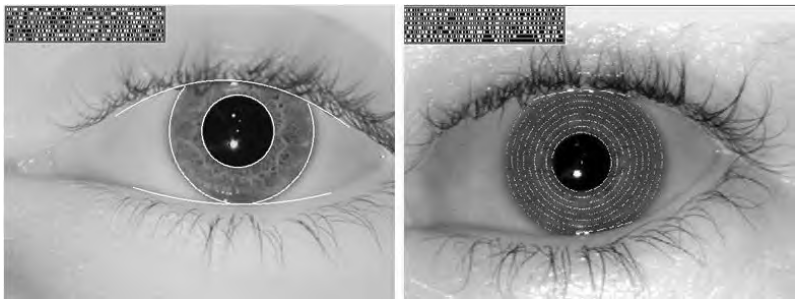
Slika 5.5. Upoređivanje otisaka prstiju [19]

Na slici 5.5 vidi se tipična analiza otiska prsta, iz koje se izvlače kontrolne tačke koje služe za upoređivanje otisaka. Softver kao što je Verifinger [20] koristi se da ih automatski pronađe i obeleži. Iz svakog otiska se može izvući između 20 i 40 kontrolnih tačaka. Slika 5.5 (sredina) takođe prikazuje jednu tehniku upoređivanja otisaka, gde se prvo Voronojev dijagram dobije od reprezentacije kontrolnih tačaka. Iz Voronojevog dijagrama se onda dobije planarni graf koji može da se upoređuje sa planarnim grafovima drugih otisaka radi dobijanja procene približnosti dva otiska [19].

Problem sa otiscima prstiju je to što se iz svakog otiska može izvući premalo kontrolnih tačaka, pa se dešava da kako raste broj otisaka sa kojima se uzorak upoređuje, tako raste i broj grešaka u identifikaciji. Slučaj Brandona Majfilda [21], advokata iz Amerike, daje nam razloga da razmislimo o pouzdanosti otisaka prstiju kao načina da se identifikuju optuženi u svim situacijama. Brandon je optužen da je učestvovao u terorističkom napadu na voz u Madridu 2004. godine jer je njegov otisak prsta, prema tvrdnji FBI, “apsolutno i nesporno”, mečiran sa otiskom na delu bombe koju je pronašla španska policija. On je proveo više od dve nedelje u pritvoru dok ga sudija nije pustio jer je

španska policija pronašla bolje umečiran otisak jednog Alžirca, koji je stvarno i počinio taj zločin. Brandon je tužio državu zbog raznih nepravilnosti u njegovom slučaju i dobio je presudu u svoju korist 2006. godine, kao i novčanu nadoknadu od dva miliona dolara. Njegov slučaj, gde se potkrala greška, nije izolovan - postoji mnogo visokoprofilnih slučajeva koji nas navode da zaključimo da ova tehnologije nije savršena, pogotovo kada se radi o velikim skupovima podataka. Informacija koja se dobije iz otisaka prstiju nije dovoljno diskriminantna da bi se nesporno zaključilo u svim situacijama da jedan otisak pripada tačno datoj osobi. U studiji urađenoj 1995. godine [22] testirana je sposobnost 156 eksperata da mečiraju otiske kandidata sa bazom otisaka. Ekspertima je data baza od samo četiri kompleta od svih 10 prstiju, tj. 40 otisaka, koja je sastavljena od strane "International Association for Identification", najveće međunarodne asocijacije forenzičara. Njihov zadatak je bio da umečiraju samo sedam nepoznatih otisaka sa bazom od 40 poznatih. Rezultati su bili razočaravajući. Samo 44 odsto ispitanika je uspelo da bez grešaka upari svih sedam nepoznatih prstiju. Svaki peti ispitani je napravio barem jednu grešku u identifikaciji, što bi vrlo verovatno rezultiralo zatvorskom kaznom za optuženog da se ovakva greška desi na sudu. Poražavajuća činjenica je da se u praksi upravo ljudi, tj. eksperti za analizu otisaka prstiju, a ne kompjuteri, pitaju kada je konačna reč o tome da li pronađen otisak pripada ili ne pripada osumnjičenoj osobi.

Postoje i novije tehnologije u oblasti biometrije koje su tačnije, ali ne ostavljaju vidljiv trag. Radi se o ostalim merljivim indikatorima koje naša tela proizvode, a to su naš glas, način hoda, otisak šake, otisak ušiju, oblik lica i izgled irisa naših očiju. Irisi su možda najzanimljiviji od navedenih biometrijskih indikatora jer na maloj površini prenose mnogo informacija. Levo i desno oko u bilo čijoj glavi, osim boje, imaju vrlo malo sličnosti kada se uporede uz pomoć digitalne forenzike. Čak se vrlo teško utvrđuje da dva oka pripadaju istom licu. Na slici 5.6 se vide koraci potrebni da bi se iz jednog oka dobio opis, odnosno kod, koji može da se upoređuje sa drugim očima (slično kao za otisak prsta u prethodnom delu teksta).



Slika 5.6. Identifikacija regiona irisa [23] (levo); izvlačenje koda iz irisa (desno)

Na slici 5.6 se isto primećuje da je uzorna fotografija oka napravljena u infracrvenom spektru, da bi bila nezavisna od dnevnog ili noćnog osvetljenja. Na taj način se izbegava

i boja očiju, koja nije bitna za ovu tehniku izvlačenja koda iz oka. Iris je obeležen na levoj strani slike 5.6, i vidi se kao region između zenice i beonjače, odnosno sklere. On je jedini vidljivi deo oka za koji može da se tvrdi da se razlikuje kada se oči upoređuju. Kada se izoluje iris na slici, koncentrični krugovi, kao oni što se vide na desnoj strani slike 5.6, koriste se radi izvlačenja koda. Kod se vidi u gornjim levim uglovima oba dela slike 5.6 i konstruiše se pomoću promena iz tamnih u svetle piksele iz svakog kruga. Zanimjliva osobina irisa je da se on skalira u okolnostima više ili manje svetlosti, jer se naše zenica šire i skupljaju. Bez obzira na to, kod koji se izvuče iz njega je uvek isti jer se iris ravnomerno skuplja i širi.

5.2.4. Autonomna vožnja

Jedna aktuelna primena kompjuterske vizije je u auto-industriji, gde kompanije pokušavaju, i uspevaju da naprave kompjuterski sistem koji sam upravlja automobilom. Cilj je da sistem prikuplja podatke iz raznih unutrašnjih senzora kao što su radari, GPS, lidari i kamere - slika 5.7 (levo), izračuna putanju automobila koja vodi do cilja, izdaje komande motoru, volanu i kočnicama, i za sve to vreme pazi da ne udari pešake i druge automobile, tj. poštuje saobraćajne propise - slika 5.7 (desno).

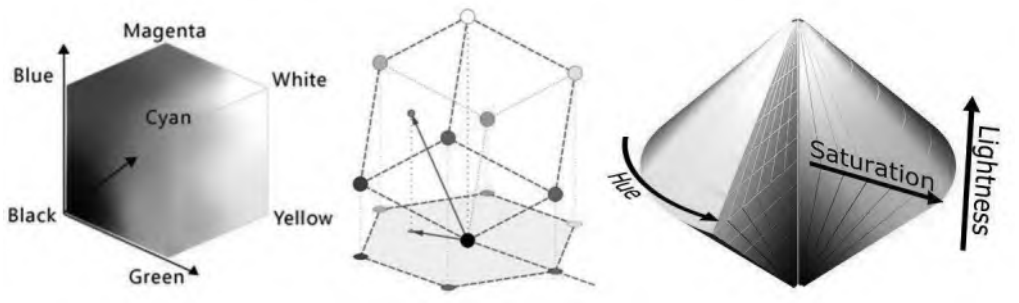
Slika 5.7. Senzori u autonomnoj vožnji [24] (levo); autonomna vožnja [25] (desno)

Tehnike kompjuterske vizije se koriste najviše za prepoznavanje trake i drugih saobraćajnih znakova na putu, kao i lokacije i brzine kretanja drugih automobila i pešaka u okolini. Radari služe da potvrde ili koriguju greške iz informacija koje kamera pronalazi. Postoji čitava naučna oblast koja pokušava da spoji podatke iz raznih uređaja koji posmatraju istu situaciju da bi se izveli pouzdaniji zaključci, a ona se naziva “data fusion“.

Bezbednost automobila koji se kreću bez vozača je tema koja nas sve više dotiče jer se ova tehnologija tek probija u raznim zemljama sveta. U 2018. godini, vodeće firme u proizvodnji autonomnih automobila su „Google“ („Waymo“), „Tesla“, „Uber“, koje su zajedno prešle više od 1,5 milijardi kilometara u autonomnim režimima vožnje. Broj grešaka koji ovi sistemi prave se smanjuje, a i broj preuzimanja kontrole nad vozilom od strane kontrolnog vozača se takođe drastično smanjuje. „Waymo“ je prijavio da njihovi autonomni automobili zahtevaju u proseku jedno preuzimanje kontrole vozila za svakih 9.000 kilometra vožnje [26].

5.3. BOJE, HISTOGRAMI I OPERATORI

Uspeh obrade slike, kao i kompjuterske vizije, u velikom delu zavisi od mogućnosti razumevanja boja na slici, odnosno video-snimku. Spektar svih boja se prikazuje na ekranu pomoću linearne kombinacije tri osnovne boje - crvene (R), zelene (G) i plave (B), kao što se vidi na slici 5.8 (levo). Intenzitet svake osnovne boje se opisuje kao diskretan broj u intervalu [0,255], što sve zajedno čini $256^3 \approx 16$ milona boja. RBG prostor boja u ovom formatu ne pokriva ceo spektar boja koje mi možemo da vidimo, ali pokriva njegov veći deo.

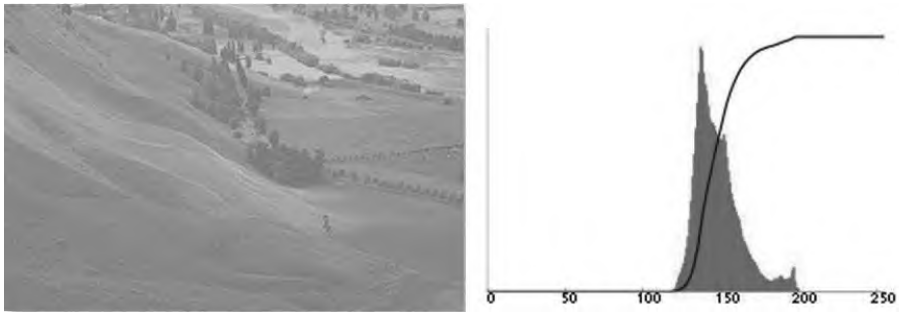


Slika 5.8. Kocka RGB boje [27] (levo);
projekcija u HSL prostor (sredina); HSL prostor boja [28] (desno)

RGB nije jedini prostor boja koji se koristi u obradi slike. Hue - saturation - lightness (HSL) je jedan prostor boja koji može da se izvede iz RGB, i ima određene osobine koje ga čine zgodnim za prostije manipulisanje bojama na slici. Na primer, ukoliko želimo da data slika bude svetlija, moramo matematičkom operacijom da izračunamo novu vrednost za svaki piksel pojedinačno. To bi bila prosta operacija da predstavimo boje na drugačiji način, a to se zapravo i dešava sa HSL prostorom. Zamislite da postavimo RGB kocku da stane na svoj najtamniji ćošak, i da njen najsvetliji ugao bude njena najviša tačka, kao što se vidi na sredini slike 5.8. Na ovaj način se prebacuju svetlosne vrednosti, ili količina svetlosti svake boje, na vertikalu prostora. Ovo objašnjenje aproksimira oblik HSL prostora koji se vidi na desnoj strani slike 5.8. U HSL prostoru boja se definiše pomoću boje (hue), zasićenja (saturation) i osvetljenja (lightness). Slično kao u RBG prostoru, boja se formira pomoću tri koordinate, ali ovde se one izražavaju kao ugao u opsegu (0-360°) koji bira boju, procenat (0-100%) za saturaciju i procenat (0-100%) za osvetljenje. U ovoj situaciji je lako pojačati ili smanjiti osvetljenje na slici jer je jedna dimenzija u HSL prostoru direktno kontroliše.

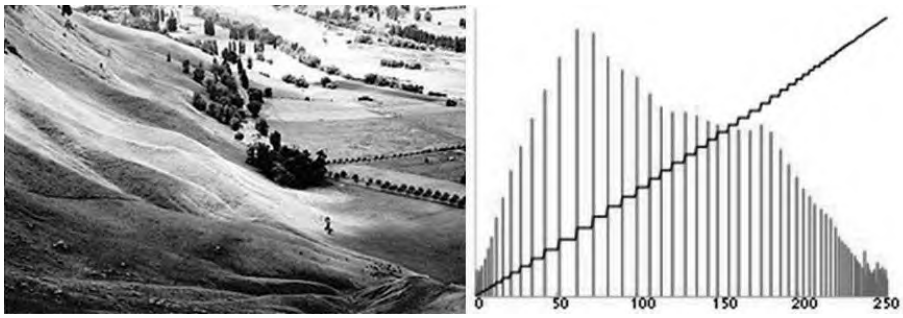
Histogram u ovom naučnom polju je distribucija intenziteta piksela na slici. Uzimajući u obzir sivu (grayscale) sliku čiji pikseli imaju opseg boja [0-255], gde je crna boja predstavljena intenzitetom 0 i bela intenzitetom 255, histogram jedne ovakve slike bi

predstavio distribuciju, ili broj piksela svake boje, poređan po rastućem rasporedu. Primer sive slike (levo) i njenog histograma (desno) možemo videti na slici 5.9. Scena na toj slici ima nizak kontrast, što znači da su joj boje relativno blizu jedna drugoj. Nizak kontrast ove fotografije se takođe oslikava u njenom histogramu, gde se vidi da praktično nema piksela koji su baš crni (intenziteti 0-100), kao što nema ni baš belih (intenziteti 200-255).



Slika 5.9. Siva (grayscale) slika (levo);
odgovarajući histogram i kumulativni zbir broja piksela (linija) (desno)

Crna linija na desnoj strani slike 5.9 predstavlja kumulativnu količinu svih boja. Njena vrednost za dati kumulativni intenzitet do boje *IB* je količina svih piksela čije su boje u intervalu [0-*IB*]. Za celu sliku, odnosno kumulativna količina svih piksela na slici je jednaka broju piksela na celoj slici. Crna linija se koristi u proceduri ekvalizacije slike, koja je predstavljena na slici 5.10.



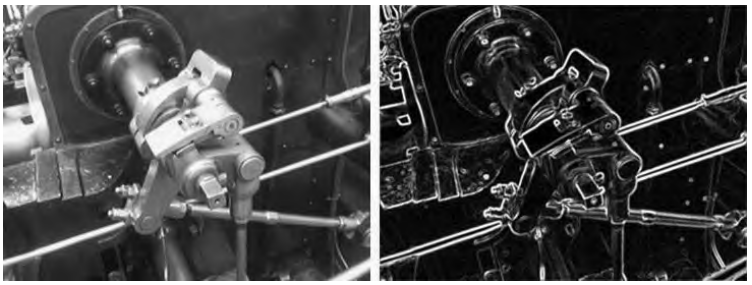
Slika 5.10. Ekvalizovana slika (levo); histogram ekvalizovane slike (desno)

Ekvalizacija slike je način uravnoteživanja boja po čitavom spektru, tako da je svaka nijansa ravnopravno zastupljena. Rezultat ekvalizacije je slika koja ima veći kontrast od originala, i na kojoj se jasnije vide detalji. Histogram ekvalizovane slike se vidi na desnoj strani slike 5.10, gde se primećuje prava linija kumulativnih količina boja, kao i

skoro potpuno iskorišćen opseg boja. Ekvalizacija se izvodi na slikama u boji na sličan način, osim što se R, G i B kanali ekvalizuju jedan po jedan, pa se rezultat spaja opet u RGB sliku. Ovo je standardna tehnika obrade slika jer njen rezultat često pojašnjava sliku, ili pročisti njene detalje, što vodi ka kvalitetnijem ulazu za bilo koju primenu u kompjuterskoj viziji.

Histogram takođe može da služi kao prosto pomagalo kada se upoređuju slike, jer različite slike obično imaju različite histograme. Problem sa korišćenjem histograma kada upoređujemo slike jeste to što se gubi informacija o poziciji boje na slici, što se beleži u pikselima, pa se samim tim smanjuje nivo informacija sa kojim radimo. Fotografi često pogledaju histogram svojih fotografija da bi proverili njihov nivo osvetljenja, kao i distribuciju boja, jer se lako vidi prevelika količina piksela u najmračnijem ili najsvetlijem delu histograma.

Jedna bitna operacija za razumevanja slike jeste da se prepozaju njene ivice. Uz pomoć ivica možemo da pokušamo da rekonstruišemo predmete na slici, ili da ih barem kvantitativno opišemo, što nas vodi ka boljem razumevanju same slike. Detekcija ivica je osnovni alat u kompjuterskoj viziji, pogotovu u polju prepoznavanja i karakterizovanju predmeta na slici. Spobnost za detekciju ivica, naročito horizontalnih i vertikalnih, razvija nam se u mozgu od rođenja, postaje ugrađena u našim vizuelnim sistemima. Naš mozak obavlja dekompoziciju predmeta u njegove osnovne linije i boje kada pamti [29], i u poslednje vreme kompjuterske neuralne mreže na isti način prepoznaju predmete na slici. Detekcija ivica se opisuje na slici 5.11, gde se vidi parna mašina, i njene ivice.

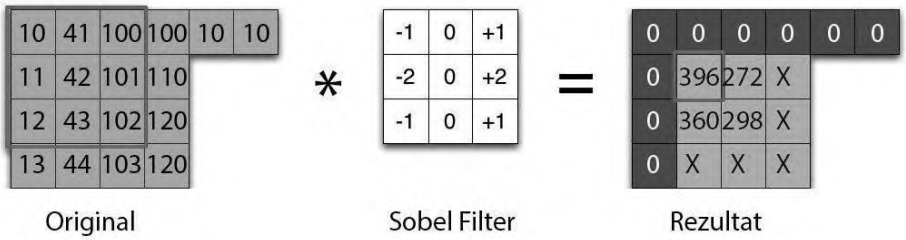


Slika 5.11. Parna mašina (levo); detektovane ivice mašine [30] (desno)

Postoje razni načini detektovanja ivica na slici, ali „sobel“ operator je možda najpopularniji i najprostiji od njih. „Sobel“ filter nalazi jake vertikalne i horizontalne ivice na slici uz pomoć dve proste 3 x 3 matrice L_x, i L_y, i originalne slike, koju ćemo da nazovemo L.

$$L_x = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * L \qquad L_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * L$$

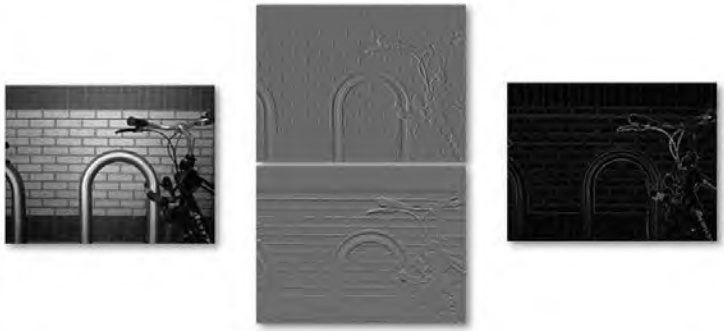
Matrica L_y nalazi vertikalne ivice, a matrica L_x pronalazi horizontalne. Na slici 5.12 se vidi primena matrice L_x na izvornu sliku. Tu je predstavljen gornji levi ugao jedne izvorne sive slike, i njeni intenziteti piksela. „Sobel“ filter se primenjuje na svaki 3 x 3 podskup piksela na čitavoj slici, ali u ovom primeru počinjemo iz ugla, gde je kvadratom obeležen podskup na koji se on prvo primenjuje. Množimo odgovarajuće ćelije uzorne slike sa matricom i sumiramo: -1*10 -2*11 -1*12 +0*41 +0*42 +0*43 +1*100 +2*101 +1*102 = 396.



Slika 5.12. Intenziteti originalne slike (levo);
L_y „sobel“ filter (sredina); reprezentacija uspravnih ivica (desno)

Primećujemo da je broj 396 iz našeg primera upisan u piksel koji je u centru kvadrata, ali se taj broj beleži u sliku koja predstavlja rezultat, i koja se vidi sa desne strane. Ona je istih dimenzija kao i uzorna, ali ne možemo da izračunamo vrednosti za njene okvirne piksele jer kvadrat koji bi ih računao izlazi iz dimenzija uzorne slike.

L_x, i L_y predstavljaju uspravne i vodoravne intenzitete promena ivica, ali ne predstavljaju završnu reprezentaciju ivične slike. Na slici 5.13 prikazani su potrebni koraci za sastavljanje ivične slike.



Slika 5.13. [30] Uzorna slika (levo); L_x i L_y (sredina); „sobel“ ivice uzorne slike (desno)

L_x, i L_y se računaju iz uzorne slike po već navedenoj metodi, a svaki piksel u ivičnoj, odnosno „sobel“ reprezentaciji ima dve vrednosti, a to su intenzitet ivice I, kao i ugao Θ

pod kojim se ta ivica nalazi. Ugao nije prikazan na slici 5.13, ali je korisna informacija kada se opisuje predmet na slici. Intenzitet *I*, i ugao Θ za dati piksel (*i,j*) se formiraju na sleceći način:

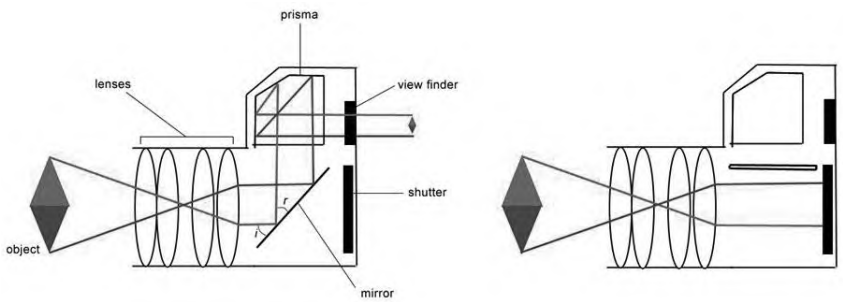
$$I(i,j) = \sqrt{L_x(i,j)^2 + L_y(i,j)^2}, \qquad \theta(i,j) = atan\left(\frac{L_y(i,j)}{L_x(i,j)}\right)$$

Odrediti ivice je obično prvi korak ka prepozvananju komplikovanijih struktura na slici, kao što su linije, krugovi, elipse, pa čak služi i za detektovanje teksture ili drugih karakteristika. Uz pomoć ovakvih matematičkih konstrukcija se konstruišu i komplikovaniji alati, koji rešavaju zadatke kao što su prepoznavanje lica na slici, ili čak autonomna vožnja automobila.

5.4. HDR SLIKE

„High dynamic range“ (HDR) je tehnika koja omogućava stvaranje fotografija koje predstavljaju veći opseg nijansi svetlosti od običnih tehnika fotografisanja. Naše oči se prilagođavaju na razne svetlosne okolnosti jer se zenice skupljaju i šire u zavisnosti od količine svetlosti sa kojom se susreću. Prečnik zenice može da varira od 1 do 9 mm, u zavisnosti od izloženosti dnevnoj svetlosti, ili gledanja u zvezde noću. Oči na ovaj način dinamično kontrolišu nivo svetlosti koji dopire do ćelija receptora, i stvaraju uslove da mozak na optimalan način dalje procesira sliku za najbolju reprezentaciju okoline u kojoj se nalazimo. One se kontinualno i samostalno prilagođavaju nivou svetlosti, pa čak i na određene delove scene na koje se koncentrišemo. Nivo ili količina svetlosti koja ulazi u oči ili se beleži na senzoru foto-aparata naziva se ekspozicija.

Foto-aparat takođe ima mogućnost da kontroliše ekspoziciju, ali ne radi to dinamično i kontinualno kao naše oči. Na slici 5.14 se vidi unutrašnji mehanizam moderni “digital single lens reflex“ (DSLR) foto-aparata.



Slika 5.14. Unutrašnji mehanizam DSLR aparata - spušteno ogledalo (levo); podignuto ogledalo [31] (desno)

Kada se pritisne okidač za snimanje fotografije, ogledalo koje se nalazi na sredini aparata se podigne. Svetlost onda prođe do senzora koji ga beleži, obrađuje i čuva u obliku slike. Ogledalo se vraća na početni položaj kada istekne izračunato vreme po kojem treba da bude podignuto. Vreme za podignuto ogledalo (otvarač), ili "shutter speed", može da varira od 1/8.000 dela sekunde sve do 30 sekundi, u zavisnosti od okolnosti i želje fotografa. Na slici 5.15 se vidi efekat slike koja je snimljena u različitim ekspozicijama. Kada je ekspozicija mala, vidi se svaka kapljica vode, a kada je velika, vidi se mlaz vode.

Slika 5.15. Fontana, ekspozicija 1/1.000 sekundi (levo); ekspozicija 1/25 sekundi [32] (desno)

Osim ogledala, nivo svetlosti se kontroliši pomoću blende, koja liči i ima istu funkciju kao i zenica. Za bilo koju situaciju fotografisanja, računica za vreme slikanja, otvorenosti blende, osetljivosti senzora itd. izvodi se samo jednom, što je problem kod foto-aparata. To rezultira slikom koja je u proseku osvetljena pravilno, ali se veliki broj detalja gubi u regionima koji zahtevaju više ili manje svetlosti da bi se bolje videli. Slika 5.16 prikazuje efekat manje, srednje i veće ekspozicije na slici. Srednja fotografija je tipičan rezultat autopodešavanja ekspozicije foto-aparata. Levo i desno se vide fotografije sa namerno spuštenim i podignutim nivoom osvetljenja. Spuštanjem ekspozicije se na primer otkrivaju detalji oko svakog prozora Fullerton hotela, a podizanjem ekspozicije se bolje osvetljavaju zgrade u pozadini, kao i odsjaj Fullertona u vodi.

Slika 5.16. Fullerton hotel, Marina bej, Singapur, 2015 - manja ekspozicija (levo); srednja ekspozicija (sredina); veća ekspozicija (desno)

Rezultat spajanja ovih ekspozicija, sa ciljem što boljeg čuvanja svih detalja originalne scene, vidi se na slici 5.17.

Slika 5.17. HDR slika Fullerton hotela, Marina bej, Singapur, 2015.

Ne postoji jedinstven način spajanja ekspozicija u univerzalan rezultat. Pravljenje HDR fotografija je naučno polje, koje koristi složene matematičke funkcije da bi stiglo do foto-realističnih rezultata. Mobilni telefoni, kao i neke DSLR kamere, imaju ugrađena HDR podešavanja, ali se najbolji rezultati dobijaju sa profesionalnom kamerom, stativom i specijalizovanim softverima za sastavljanje HDR fotografija, kao što su Fotomatiks i Fotošop. Ova tehnika najbolje funkcioniše kada postoji veliki kontrast na datoj slici, ili kada slika ima veće regione pune belih ili crnih piksela. Obično se u tim regionima kriju detalji koji nisu vidljivi bez HDR. Tipičan primer ovoga je kada je subjekat na slici osvetljen iz kontra pravca foto-aparata, tj. kada je "sunce iza leđa".

Ideja spajanja ekspozicija nije nova, i prvi put je njena realizacija pokušana čak 1850. Godine, kada je Gustav le Grej, zvanični fotograf Napoleona III, prvi put sastavio jednu sliku iz dve ekspozicije. Do tada je bilo teško fotografisati more I nebo u istom kadru zbog velikih razlika u osvetljenju [33]. HDR je zaslužan za način na koji kompjuteri mogu da generišu foto-realistične scene. Boje u RGB prostoru su standardno sačuvane u 8-bitnoj reprezentaciji, ali za veći raspon boja, koji je neophodan za foto-realistične slike, potrebna je 16-bitna ili čak 32-bitna reprezentacija, što dozvoljava 65.536 ili čak 4.294.967.296 boja po kanalu. Ovakve mogućnosti dozvoljavaju flimskoj industriji da generiše kompjuterski renderovane snimke koje teško možemo da raspoznamo od realnih.

5.5. KLJUČNI TERMINI

Kompjuterska vizija – naučna oblast koja pokušava da razume šta se dešava na ulaznoj slici ili video-snimku, i opisuje to što vidi (pretvara sliku ili video-snimak u skup karakteristika).

Obrada slike – naučna oblast koja uzima ulaznu sliku, i manipuliše njom tako da bude

bolja ili lakše razumljiva nakon manipulacije.

Biometrija – nauka o statističkoj analizi bioloških podataka. Najčešći biološki indikatori su otisci prstiju, šaka, ušiju, snimci irisa, glasa i hoda.

Autonomna vožnja – automobil koji može da se potpuno samostalno vozi na svim vrstama puteva i u svim vremenskim uslovima, poštujući saobraćajne propise.

Ekvalizacija slike – tehnika koja se koristi da bi se povećao kontrast na slici, samim tim i nivo detalja.

Kontrast – razlika u količini svetlosti ili boja u podskupovima piksela na slici koja omogućava prepoznatljivu razliku između njih.

Histogram – distribucija intenziteta piksela na slici, koja tipično predstavlja boje u rasponu [0-255].

"Sobel" filter – tehnika koja omogućava prepoznavanje jačine i pravca ivica na slici, gde je ulaz siva slika, a izlaz ivična reprezentacija.

Ekspozicija – količina svetlosti koja dospe do senzora foto-aparata, koja zavisi od sledećih faktora: otvorenosti blende, brzine zatvarača i osvetljenja scene.

„High dynamic range" (HDR) – tehnika koja se koristi za reprodukovanje većih raspona boja ili nijansi svetlosti nego što je moguće sa standardnim fotografijama.

DSLR kamera – foto-aparat koji koristi digitalni senzor svetlosti da bi sačuvao fotografiju, i koji ima ogledalo (otvarač) i prizmu koja dozvoljava fotografu da vidi šta slika pre nego što se scena prenesa na senzor.

Pitanja za ponavljanje

1. Šta je kompjuterska vizija?
2. Šta je obrada slike?
3. Koja je razlika između pronalaženja lica i prepoznavanja lica?
4. Šta je „haar" filter?
5. Koji su najčešće korišćeni biometrijski indikatori?
6. Ko je Brandon Majfild?
7. Šta je diskriminantnije - otisak prsta ili iris očiju?
8. Objasni tehnologije koje se koriste u autonomnoj vožnji.
9. Šta predstavlja histogram i gde se koristi?
10. Šta je HSL prostor?
11. Šta je RGB prostor?
12. Šta znači ekvalizacija boje?
13. Kako se obavlja funkcija detekcija ivica na slici i koji operator tu funkciju obavlja?

Literatura

[1] S. Cochran, “Unusual Seating Options Complement Productivity,” 9 6 2011. [Online]. https://www.brighthub.com/office/home/articles/69470.aspx. [Accessed 7 2018].

[2] Wikipedia, “The Mother of All Demos,” 1 8 2018. [Online]. https://en.wikipedia.org/wiki/The_Mother_of_All_Demos. [Accessed 8 2018].

[3] D. Ogden, “The DMV.....,” 20 3 2012. [Online]. http://thehappynewslady.blogspot.com/2012/03/dmv-by-diane-ogden.html. [Accessed 8 2018].

[4] Microsoft, “Animations and Transitions,” 30 5 2018. [Online]. https://docs.microsoft.com/en-us/windows/desktop/uxguide/vis-animations. [Accessed 8 2018].

[5] N. Babich, “Golden Rules of User Interface Design,” UX Planet, 10 2 2016. [Online]. https://uxplanet.org/golden-rules-of-user-interface-design-19282aeb06b. [Accessed 8 2018].

[6] Steve, “Anatomy of an Error Message – A Windows Vista Example,” Still Life, 2 4 2009. [Online]. https://steveswanson.wordpress.com/tag/bad-error/. [Accessed 8 2018].

[7] S. Duggirala, “10 Usability Heuristics with Examples,” Prototypr, 18 8 2016. [Online]. https://blog.prototypr.io/10-usability-heuristics-with-examples-4a81ada920c. [Accessed 8 2018].

[8] SrbijaDanas, “PROVERA VOZAČA! Ko ima prvenstvo na raskrsnici ako semafor ne radi?,” 12 09 2017. [Online]. https://www.srbijadanas.com/auto/vesti/provera-vozaca-ko-ima-prvenstvo-na-raskrsnici-ako-semafor-ne-radi-video-2017-09-12. [Accessed 8 2018].

[9] M. Draper, “Intersections have county council’s attention,” The Reporter, 3 7 2018. [Online]. http://porthawkesburyreporter.com/intersections-have-county-councils-attention/. [Accessed 8 2018].

[10] Scoopy Web, “These Are the Safest Cities to Live in Around the World,” 2 2018. [Online]. http://www.scoopyweb.com/2018/02/these-are-safest-cities-to-live-in.html. [Accessed 8 2018].

[11] M. Stojmenovic and G. Lindgaard, “Probing PROBE: A field study of an advanced decision support prototype for managing Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) events,” in *Proceedings of the 11th International ISCRAM Conference*, Pennsylvania, USA, 2014.

[12] Balsamiq, “Balsamiq,” Balsamiq, 1 2008. [Online]. https://balsamiq.com/. [Accessed 8 2018].

[13] Asteria Expeditions, “ONTDEK DE MEEST FASCINERENDE VOGEL TER WERELD,” 25 11 2015. [Online]. https://www.asteriaexpeditions.be/poolexpedities/antarctica-keizerspinguinsafari-9-daagse/keizerspinguin.asp. [Accessed 6 2018].

[14] R. B. Sherley, T. Burghardt, P. J. Barham, ,. N. Campbell and I. C. Cuthill, “Spotting the difference: towards fully-automated population monitoring of African penguins Spheniscus demersus,” *Endang Species Research,* vol. 11, no. 2, pp. 101-111, 2010.

[15] SENSIP, “Arizona State University,” [Online]. https://sensip.engineering.asu.edu/research/image-and-video-processing/. [Accessed July 2018].

[16] P. Viola and M. Jones, “Robust Real-time Face Detection,” in *International Conference on Computer Vision*, 2001.

[17] The Guardian, “From Ale to Jail: facial recognition catches criminals at China beer festival,” *The Guardian,* 2018.

[18] Wikipedia, “Integrated Automated Fingerprint Identification System,” [Online]. https://en.wikipedia.org/wiki/Integrated_Automated_Fingerprint_Identification_System. [Accessed 7 2018].

[19] W. Yang, J. Hu and M. Stojmenovic, “An alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbor structures,” *Pattern Recognition,* vol. 47, no. 3, p. 1309–1320, 2014.

[20] NeuroTechnology, “Verifinger,” [Online]. http://www.neurotechnology.com/verifinger.html. [Accessed 7 2018].

[21] Wikipedia, “Brandon Mayfield,” [Online]. https://en.wikipedia.org/wiki/Brandon_Mayfield. [Accessed 6 2008].

[22] S. L. Zabell, “Fingerprint Evidence,,” *Journal of Law and Policy (Brooklyn College Law School),* vol. 143, no. 77, 2005.

[23] J. Daugman, “How Iris Recognition Works,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,* vol. 14, no. 1, 2004.

[24] Z. Wendt and J. S. Cook, “Saved by the Sensor: Vehicle Awareness in the Self-Driving Age,” Machine Design, 17 1 2018. [Online]. http://www.machinedesign.com/motion-control/saved-sensor-vehicle-awareness-self-driving-age. [Accessed 7 2018].

[25] Nvidia, “Autonomous Driving,” Nvidia, [Online]. https://www.nvidia.com/en-us/gtc/topics/autonomous-driving/. [Accessed 7 2018].

[26] Wikipedia, “Autonomous car,” 7 2018. [Online]. https://en.wikipedia.org/wiki/Autonomous_car. [Accessed 7 2018].

[27] MIcrosoft, “Color,” 31 5 2018. [Online]. https://docs.microsoft.com/en-us/windows/desktop/uxguide/vis-color. [Accessed 7 2018].

[28] SharkD, “HSL color solid dblcone,” [Online]. https://commons.wikimedia.org/wiki/File:HSL_color_solid_dblcone.png. [Accessed 7 2018].

[29] C. Blakemore and G. F. Cooper, “Development of the brain depends on the visual environment,” *Nature,* vol. 228, no. 5270, pp. 477-478, 1970.

[30] Wikipedia, “Sobel Operator,” [Online]. https://en.wikipedia.org/wiki/Sobel_operator. [Accessed 7 2018].

[31] L. Lagerqvist, “Camera Anatomy,” 3 11 2010. [Online]. https://lovisalagerqvist.wordpress.com/tag/anatomy/. [Accessed 6 2018].

[32] Rensselaer Polytechnic Institute, Department of the Arts - School of HASS, “Photographz Tips,” [Online]. http://www.arts.rpi.edu/~ruiz/idispring09/photography%20tips.htm. [Accessed 7 2018].

[33] Wikipedia, “Gustave Le Gray,” [Online]. https://en.wikipedia.org/wiki/Gustave_Le_Gray. [Accessed 7 2018].

6. VISUAL BASIC

Posao programera je da izmisli algoritam koji će rešiti zadati problem, i taj algoritam prevesti u jezik koji kompjuter može da razume. Kao i sa jezicima koje govorimo, naš život je lakši ako ih znamo što više. Istom logikom se vode i programerski jezici jer su određeni pogodni u određenim situacijama, tako da je i život programera lakši ako zna što više programskih jezika. Srećom, najpopularniji programski jezici funkcionišu po istom, objektno orijentisanom principu, i njihova sintaksa je jedina bitna razlika. Visual Basic (VB) je ovde izabran kao jezik koji ćemo proučavati jer se uklapa u softverski paket koji najviše obrađujemo u ovom predmetu, a to je Microsoft Office. U pozadini svakog dela Office-a je Visual Basic, koji služi za jezik u kojem se prave skripte ili makroi za specifično definisano korišćenje ovih alata. On je jezik koji ima i širu industrijsku primenu, ali se mora priznati da je njegova popularnost u nepovratnom i kontinualnom padu. Bez obzira na to, primeri i logika koji se predstavljaju u ovom poglavlju su univerzalno primenjivi u bilo kom programskom jeziku, jer oslikavaju način „razmišljanja“ kompjutera.

6.1. UVOD

Posao kompjutera je da prima ulazne podatke, obrađuje ih i izbacuje rezultate, prema načinu i koracima koji su definisani u algoritmu. Algoritam je u programerskom smislu konačan niz nedvosmislenih instrukcija koje se izvršavaju jedna po jedna, u ograničenom vremenu radi rešavanja određenog problema. Softver je implementacija algoritma, u proizvoljnom kompjuterskom jeziku, koje obrađuje, i pretvara u rezultate.

U programiranju, određene konstrukcije se razlikuju u poređenju sa matematikom, a glavna od njih je da je A = B isto što i B = A. Ovo je matematički primer koji opisuje nepromenjivost relacije kada se uspostavi. Sa druge strane, u programiranju, A=B ima suštinski različito značenje od B = A. U programiranju, A = B znači da A preuzima vrednost od B, i slično time, B = A znači da B preuzima vrednost od A. U matematici je izraz A = A + 1 netačan, a u programiranju je standardni način da se naglasi da se A povećava za 1 nakon izvršavanja ove komande. Instrukcija A + 7 = 20 je dozvoljena, pa čak i lako rešiva jednačina u matematiciji, ali je nedopustiva komanda u programiranju jer bi značila da se dodeljuje vrednost 20 na „A + 7“ na levoj strani komande. To nije dozvoljeno jer na levoj strani jednakosti u programiranju mora da bude isključivo jedna varijabla.

Varijable su memorijske jedinice koje pamte određene vrste podataka, i mogu da menjaju vrednost u toku izvršavanja programa. One mogu da se zamisle kao virtuelni prostor koji pamti brojeve, tekst, logične (boolean) vrednosti, ili cele kompleksne strukture. Varijable se koriste kada se ne zna unapred vrednost jednog podatka, ali se taj

podatak koristi u izvršavanju programa i njegov rezultat zavisi od nje. Svaka varijabla ima svoj tip u VB-u, čiji je najbitniji podskup naveden u tabeli 6.1.

TIP VARIJABLE	VELIČINA	OPSEG
BOOLEAN	Zavisi od platforme	True ili False
BYTE	1 byte	0 do 255 (unsigned bez znaka)
CHAR	2 byte-a	0 do 65535 (unsigned bez znaka)
DATE	8 byte-ova	0:00:00 (ponoć), Januar 1, 0001 do 11:59:59 PM, Decembar 31, 9999
DOUBLE	8 byte-ova	≈ -1.797^{308} do -4.940^{324} za negativne vrednosti ≈ 4.940^{-324} do 1.797^{308} za pozitivne vrednosti
INTEGER	4 byte-a	-2,147,483,648 do 2,147,483,647 (sa znakom)
LONG (INTEGER)	8 byte-ova	-9,223,372,036,854,775,808 do 9,223,372,036,854,775,807 (sa znakom)
SHORT (INTEGER)	2 byte-a	-32,768 do 32,767 (sa znakom)
STRING	Zavisi od platforme	0 to 2 milijarde Unicode karaktera

Tabela 6.1. Tipovi varijable u Visual Basic-u [1]

Primere korišćenja skoro svih navedenih tipova varijabli ćemo videti u ovom poglavlju, ali ćemo obrazložiti svaku ovde. Boolean vrednosti su varijable koje imaju samo dva moguća stanja, a to su tačno ili netačno. Koriste se kada dalji tok programa zavisi od toga da li je određen uslov ispoštovan ili nije. Byte, Integer, Long, i Short su varijable koje pamte cele brojeve različitih veličina, koje su navedene u desnoj koloni tabele 6.1. Iz ovog podskupa, tip Integer se najčešće koristi u praksi. Varijabla Date je tip koji čuva datume, i sastoji se od godine, meseca, dana, sata, minuta, sekundi, pa čak i milisekundi. Polja Char i String su tekstualna polja, i mogu da čuvaju sva „Unicode“ slova. To znači da sve azbuke sveta dolaze u obzir kada se čuvaju u ovim tipovima varijabli. Varijabla Char može da čuva samo jedno slovo, ali varijabla String može da čuva proizvoljnu dužinu tekstualnog zapisa. Double je tip realnog broja, što znači da sadrži ne samo cele brojeve, nego celi skup brojeva sa decimalom.

Da bismo koristili bilo koju varijablu, moramo je prvo navesti, i dodetili joj ime. Na primer, ako želimo da pamtimo nečije ime, možemo navesti, ili pravilnije - deklarisati varijablu tipa String, koja će se zvati „ime“ na sledeći način: Dim ime As String. Ako želimo

da pamtimo cenu, možemo da deklarišemo: Dim cena as Double, uzimajući u obzir da su cene realni brojevi, pa ne mogu da budu tipa Integer. „Dim“ je početak deklaracije, i propraćen je imenom deklarisane varijable, gde na kraju sledi njen tip. U generalnom slučaju, bilo koja varijabla u VB-u se deklariše na sledeći način:

```
1.  Dim XX As YY
```

Svi proizvodi Microsoft Office-a imaju ugrađeni editor za pisanje, kompajliranje i izvršavanje VB programa. U Excel-u se, na primer, prvo dozvoli prikazivanje Developer Ribbon-a u glavnim menijima, pa se potom klikne na „Visual Basic“ dugme da bi se otvorio prozor za rad u VB-u. Pod menijem „Insert“ se izabere opcija „Module“, i pojavi se prozor za editovanje VB koda.

6.2. VISUAL BASIC KOD

U kodu 6.1 je ispisan prvi, i najosnovniji program u Visual Basic-u. Za naše potrebe će svaki VB kod početi sa Option Explicit, što se vidi u redu broj 1 u ovom primeru. Option Explicit diktira da svaka varijabla mora da ima striktno definisan tip, kao što je navedeno u tabeli 6.1. U trećem redu je deklaracija prve procedure, odnosno metode ako vam je Java već poznata, koja se zove „main“. Ona je početna tačka programa, i završava se na 17. redu komandom „End Sub“. Ovakav eksplicitno deklarisan početak i kraj svake procedure, petlje itd. je zamena za zagrade tipa {} u Javi.

Od 5. do 8. reda su deklaracije varijable *a, b, c*, i rezultati, koji su tipa Integer, odnosno celi brojevi. Varijable koje nemaju definisanu vrednost pri deklaraciji dobjiu standardnu vrednost jednaku nuli. Red 10 poziva ugrađenu VB funkciju InputBox, koja izbacuje prozor na ekranu sa tekstom “prvi broj”, i ima polje za unos podataka. Korisnik u tom polju ukuca broj, i taj broj se zapiše u varijabli *a*. Slično se dešava sa redovim 11 i 12, i varijablama *b* i *c*. U 13. redu varijabla „rezultat“ dobija vrednost zbira varijabli *a, b* i *c*. Konačno, u 15. redu se vidi poziv na funkciju MsgBox, koja je vrsta izlaza u VB-u. Ona izbacuje mali prozor na ekranu sa porukom “rezultat je“ zajedno sa vrednošću varijable „rezultat“ u tom trenutku. Znači, ovaj program pročita tri broja, sabere ih, i izbaci rezultat.

Kod 6.1. Prvi program u Visual Basic-u

```
1.  Option Explicit
2.
3.  Sub main()
4.
5.     Dim a As Integer
6.     Dim b As Integer
7.     Dim c As Integer
8.     Dim rezultat As Integer
9.
10.    a = InputBox (“prvi broj”)
11.    b = InputBox (“drugi broj”)
12.    c = InputBox (“treci broj”)
13.    rezultat = a + b + c
14.
15.    MsgBox (“rezultat je” & rezultat)
16.
17. End Sub
```

6.3. IF KLAUZULA

Kao u skoro svakom programskom jeziku, postoji If klauzula, koja određuje koji od dva bloka koda se izvršava u zavisnosti od logičke operacije. U kodu 6.2 je program koji učitava ocene tri kolokvijuma, kao i ocenu na završnom ispitu jednog studenta. Program računa završni saldo poena datog studenta koji je broj od 0 do 100, kao i završnu ocenu, koja je od 5 do 10 (kao na našem Univerzitetu). Maksimalni saldo bodova za svaki kolokvijum je 20, i maksimalni saldo brojeva na završnom ispitu je 25. Kolokvijumi zbirno vrede 70 odsto završne ocene, i završni ispit vredi preostalih 30 odsto.

Kod 6.2 deklariše sve potrebne varijable u 3. redu, i sve su tipa Double, odnosno realni brojevi. Dozvoljeno je deklarisanje više varijabli istog tipa u istom redu u VB-u, kao i u Javi.

Kod 6.2. Računica završne ocene

```
1.   Sub main()
2.     Dim kol1, kol2, kol3, zavrs, ocena, rezultat As Double
3.
4.     kol1 = InputBox (“kolokvijum 1”)
5.     kol2 = InputBox (“kolokvijum 2”)
6.     kol3 = InputBox (“kolokvijum 3”)
7.     Zavrs = InputBox (“zavrsni”)
8.
9.     rezultat = ((kol1 + kol2 + kol3) / 60) * 70 + (zavrs / 25) * 30
10.
11.    If (rezultat >= 91) Then
12.        ocena = 10
13.    Else If (rezultat >= 81) Then
14.        ocena = 9
15.    Else If (rezultat >= 71) Then
16.        ocena = 8
17.    Else If (rezultat >= 61) Then
18.        ocena = 7
19.    Else If (rezultat >= 51) Then
20.        ocena = 6
21.    Else
22.        ocena = 5
23.    End If
24.
25.      MsgBox (“rezultat je” & Math.Round
         (rezultat, 2) & “, oce  na = “ & ocena)
26.

27.   End Sub
```

Redovi 5 do 8 učitavaju ulazne podatke o kolokvijumima i učinku na završnom ispitu. Računica za ponderisano izvedenu završnu ocenu na osnovu učinka iz kolokvijuma i završnog ispita se nalazi u 10. redu. Kolokvijumi 1, 2 i 3 se ocenjuju do 20 bodova. Maksimalni učinak je dakle 60 bodova na kolokvijumima, i zbog toga zbir učinka na kolokvijumima delimo sa 60 da bismo dobili procenat uspeha. Taj procenat na kolokvijumima vredi 70 odsto završne ocene, pa ga zato množimo sa 70. Slično radimo sa ocenom na završnom ispitu, ali koristimo njenu ponderisanu vrednost od 30 odsto, i dobijemo jednačinu koja rezultira završnom ocenom u intervalu od 0 do 100.

Sintaksa If klauzule se vidi u redovima 12 do 24. Počinje slično kao u Javi, gde se navodi If (uslov), Then (operacija), Else (druga operacija), End If. Glavna razlika je što se eksplicitno završava sa „End If“, umesto „}“. U pomenutim redovima se izvršava vrsta

sita, gde varijabla „rezultat“ prolazi kroz sve „koševe“ mogućih ocena (10, 9,..., 5) dok ne upadne u jedan. Red 26 izbacuje rezultat na ekran u vidu prozora sa zbirom poena, i odgovarajućom zaključnom ocenom.

6.4. ZBIR CIFARA CELIH BROJEVA

Osnovna operacija u programiranju je zbir cifara celih brojeva. Ona se tipično poziva radi kontrole validnosti podataka, kao kod brojeva platnih kartica, ličnih karata itd. Poslednja cifra svake kreditne kartice se naziva „checksum“ i služi kao provera validnosti svih ostalih brojeva na celoj kartici. Kreditne kartice tipično imaju 16 cifara, i poslednja cifra zbira prvih 15 cifara se postavlja za šesnaestu. Postoje i drugi načini kreiranja checksum cifre, kao što su Luhn algoritam [2], ali se svode na neku operaciju prvih x cifara da bi se generisala poslednja. Da bi se započeo bilo koji algoritam za proveravanje ili generisanje cifara, moramo da iz proizvoljnog celog broja znamo da individualno pristupamo ciframa.

Kod 6.3. Zbir cifara dvocifrenih brojeva

```
1.  Option Explicit
2.  Sub dvocifren ()
3.
4.     Dim broj, jedinice, desetice, zbir As Integer
5.
6.     broj = InputBox (“dvocifreni broj”)
7.     jedinice = broj Mod 10
8.     desetice = (broj - jedinice) / 10
9.     zbir = jedinice + desetice
10.
11.    MsgBox (“zbir = “ & zbir)
12.
13. End Sub
```

Kod 6.3 predstavlja algoritam koji uzima dvocifreni broj kao ulaz, i generiše zbir njegovih cifara. Četvrti red služi za deklaraciju varijabli, gde su sve ulazne, pomoćne i izlazne varijable tipa Integer, ili celi brojevi. Broj se učitava u šestom redu, i počinje operacija izvlačenja cifara u redovima 7 i 8. Novina ovde je Mod, ili modulo operater, koji daje ostatak pri deljenju sa nekim brojem. Na primer, 15 mod 10 = 5, ili drugim rečima, 15 modulo 10, što znači da se traži ostatak pri deljenju 15 / 10, što je u ovom slučaju 5. Obično deljenje 15 / 10 bi dalo rezultat 1,5, ili ako se ne uzima u obzir ostatak, onda je to čak jednako 1. Kada se vrši operacija proizvoljnog broja x mod 10, uvek se dobija poslednja cifra od x.

Korišćenjem modulo operacije, 7. red računa ostatak pri deljenju sa 10 ulaznog broja, i pamti rezultat u varijabli „jedinice“. Cifra jedinice bilo kog ulaznog broja će biti skladištena u varijabli „jedinice“ nakon izvršetka ove komande. Desetice se računaju tako što se od originalnog broja oduzmu jedinice, pa se zatim taj rezultat prosto podeli sa 10, kao što se vidi u 8. redu datog koda. Zbir jedinica i desetica se izračuna u 9. redu, i rezultat se izbaci na ekran u 11.

Problem zbira trocifrenih brojeva se slično rešava kao problem zbira dvocifrenih brojeva. Kod 6.4 prikazuje rešenje u VB-u koje dosta liči na kod 6.3, jer se na isti način nalaze jedinice (4. red), i na sličan način nalaze desetice (5. red). Razlika između 5. reda u kodu 6.4 i kodu 6.3 jeste što postoji dodatni modulo 10 na kraju komande. Ovo ne menja vrednost varijable desetice jer je rezultat modulo operatera nad jednocifrenim brojem x sam broj x. Modulo operater u ovom redu ukazuje na buduće rešenja ovog problema uz pomoć For petlje, koju ostavljamo za kasnije.

Stotine računamo tako što prvo oduzmemo od ulaznog broja jedinice, pa čak i 10 * desetice. Ostaje nam trocifreni broj koji se završava sa 00, pa možemo da ga delimo sa 100 da dobijemo stotine. Primera radi, pretpostavljamo da je ulazni broj = 456. Jedinice bi bile jednake cifri 6, a desetice cifri 5. Da bismo izolovali cifru čija je vrednost na stotinama, moramo da oduzmemo od tog broja i 6, i 50, pa to podelimo sa 100. Odnosno, stotine = (456 – 6 – 50) / 100 = 4. Ova operacija se vidi u 6. redu, i zbir se računa u redu 7. Rezultat se izbacuje na ekran izvršavanjem reda broj 9.

Kod 6.4. Zbir cifara trocifrenih brojeva

```
1.  Dim broj, jedinice, desetice, stotine, zbir As Integer
2.
3.  broj = InputBox (“trocifreni broj”)
4.  jedinice = broj Mod 10
5.  desetice = ((broj - jedinice) / 10) Mod 10
6.  stotine = ((broj - desetice * 10 - jedinice) / 100) Mod 10
7.  zbir = jedinice + desetice + stotine
8.
9.  MsgBox (“zbir = “ & zbir)
```

Situacija bi se ponovila kada bismo uzeli četvorocifrene brojeve u obzir, kao što se vidi u kodu 6.5. Tu je dodanta linija koda ubačena da bi izvukla cifru za hiljade, na sličan način kao cifru za stotine. Ovde se već nazire šema, koja bi mogla da se ponavlja ovako u nedogled za bilo koju dužinu početnog broja, svaki put dodajući još jednu liniju koda za svako povećanje, ali je ovo prilično nepraktično kada uzmemo u obzir da kreditne kartice imaju 16 cifara. A šta ćemo da radimo ako ne znamo unapred dužinu ulaznog broja?

Kod 6.5. Zbir cifara četvorocifrenih brojeva

```
1.  jedinice = broj Mod 10
2.  desetice = ((broj - jedinice) / 10) Mod 10
3.  stotine = ((broj - desetice * 10 - jedinice) / 100) Mod 10
4.
5.  hiljade = ((broj – stotine *100 – desetice * 10 –
    jedinice) / 1000) Mod 10
```

6.5. FOR, WHILE PETLJE

For petlje nam pomažu da pokrenemo isti blok kod određen broj puta, ali nam While petlje pomažu da to isto uradimo neodređeni broj puta. I jedna i druga struktura su česte pojave u skoro svim modernim programskim jezicima, i kao što je slučaj sa If klauzulom, sintaksa im se vrlo malo razlikuje o odnosu na Javu, C/C++/C#. U kodu 6.6 (levo) se vidi prosta primena For petlje u VB-u. Ovde se ona koristi da nađe zbir svih brojeva od 1 do 10.

Kod 6.6. (levo); zbir brojeva od 1 do 10 (desno)

```
1.  Dim i, zbir As Integer
2.
3.  zbir = 0
4.  For i = 1 To 10
5.     zbir = zbir + i
6.  Next i
```

```
7.  Dim i, zbir As Integer
8.
9.  zbir = 0
10. For i = 1 To 10 Step 2
11.    zbir = zbir + i
12. Next i
```

Sa desne strane koda 6.6 se vidi vrlo slična petlja, ali u 10. redu ima dodatnu komandu „Step 2“. Ta komanda diktira da For petlja treba da se izvrši za svaku drugu vrednost u nizu od 1 do 10, počevši sa 1. Dakle, za vrednosti i = 1, 3, 5, 7, i 9 će biti izvršena petlja, i zbir će biti 25. Komanda „Step“, ili korak, može da koristi i negativne cele brojeve, što rezultira odbrojavanjem petlje unazad, na primer „For i = 10 To 1 Step -1“ bi izvršavalo kod u okviru petlje za vrednosti i = 10, 9, 8,…, 1.

Još jedna čista matematička primena For petlje je u računici operacije faktorijel broja n, ili *n!*. Faktorijel n je operacija koja računa proizvod svih pozitivih brojeva koji su manji ili jednaki n. Na primer, 3! = 3 * 2 * 1 = 6, a 7! = 7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040. U kodu 6.7 se vidi prosto rešenje za ovaj problem gde se koristi jedna For petlja.

Kod 6.7. N faktorijel uz pomoć For petlje

```
1.  Dim i, n, broj As Integer
2.
3.  n = InputBox("n?")
4.
5.  broj = 1
6.  For i = 2 To n
7.     broj = broj * i
8.  Next i
9.
10. MsgBox ("n= " & broj)
```

Kod počinje sa deklaracijom varijabli (1. red), i učitavanjem ulaznog broja (3. red). Pretpostavlja se da je učitan celi, pozitivan broj, koji je ≥ 2 u generalnom slučaju, i primera radi neka bude n = 5 u ovom objašnjenju. For petlja kreće sa odbrojavanjem od i = 2, i u svakoj iteraciji pomnoži trenutnu vrednost i sa varijablom *broj*. U prvoj iteraciji *broj* dobija vrednost 1 * 2 = 2, jer je početna vrednost *broj*-a 1, a i = 2. U drugoj iteraciji *broj* dobija vrednost 2 * 3 = 6. U trećoj *broj* = 6 * 4 = 24, i u petoj i konačnoj, *broj* = 24 * 5 = 120. Jednim prolazom kroz ovaj prost program je izračunata vrednost faktorijel n.

For petlja može da nam pomogne da rešimo problem zbira cifara broja bilo koje veličine. Kod 6.8 opisuje ovaj postupak, koji automatizuje ključne korake pronalaženja jedinične cifre, dodavanje iste na zbir, i pripremanje ulaznog broja za sledeću iteraciju.

Kod 6.8. Zbir cifara broja bilo koje dužine

```
1.  Dim broj, jedinice, i, zbir, duzina As Integer
2.  Dim str As String
3.
4.  broj = InputBox ("N-cifreni broj")
5.  str = CStr(broj)
6.  duzina = Len(str)
7.
8.  For i = 1 To duzina
9.     jedinice = broj Mod 10
10.    zbir = zbir + jedinice
11.    broj = (broj - jedinice) / 10
12. Next i
13.
14. MsgBox ("zbir = " & zbir)
```

Nakon učitavanja broja u 4. redu, 5. red služi da se varijabla *broj*, koja je tipa Integer, ili celi broj, pretvori u tekstualni zapis tipa String. U 6. redu se računa dužinu varijable *str*, koja je u stvari dužina tekstualnog zapisa ulaznog *broj*-a, i čuva se u varijabli *duzina*. Dužina ulaznog broja nam je neophodna, jer ona određuje broj puta koji treba da se For petlja izvršava. Sama For petlja ima tri komande koje uzimaju desno, jediničnu cifru broja, dodaju je na zbir, i skrate broj za desnu cifru. Ovaj redosled koraka dozvoljava petlji da prođe kroz sve cifre broja i nađe njihov zbir. Red broj 9 je poznata operacija uzimanja jedinične cifre ulaznog broja, koju smo videli u primerima koda 6.3, 6.4 i 6.5. U 10. redu dodajemo *jedinice* na tekući *zbir*. Jedanaesti red nam dozvoljava da ponovimo operacije u 9. i 10. redovima u narednoj iteraciji For petlje jer ona odstranjuje poslednju cifru sa varijable *broj*. Na primer, ako pretpostavimo da je vrednost *broj*-a 234, njegova *duzina* je 3, što znači da će se For petlja izvršavati tri puta. Za prvu iteraciju For petlje, *broj* = 234, i njegova jedinična cifra je dakle jednaka 234 Mod 10 = 4, po 9. redu programa. Zbir nakon prve iteracije dobija vrednost *zbir* = 0 + 4 = 4, po 10. redu programa. Ključni je 11. red, jer on određuje vrednost *broj*-a u narednoj iteraciji, a njegova vrednost postaje *broj* = (234 – 3) /10 = 230/10 = 23. For petlja se ponavlja gde je *broj* = 23, i trenutni *zbir* = 4.

U drugoj od tri iteracija se dobija *jedinice* = 23 Mod 10 = 3, *zbir* = 4 + 3 = 7, i *broj* = (23 – 3) / 10 = 20 / 10 = 2. U trećoj i poslednjoj iteraciji se dobija *jedinice* = 2 Mod 10 = 2, *zbir* = 7 + 2 = 9, i *broj* = (2 – 2) / 10 = 0 / 10 = 0. Konačno se izbacuje na ekran konačna vrednost zbira, koja je 9, u 14. redu.

While petlja ima sličnu strukturu kao i For petlja, ali se koristi kada se ne zna unapred vrednost određene varijable ili logičkog operatera. Ona se deklariše sa logičkim operaterom koji diktira kada će se završiti njen blok koda, i završava se sa Wend. While petlja nam je zgodna za rešavanje problema zbira cifara proizvoljnog *broj*-a jer možemo da je strukturišemo da se završava kada je ulazni *broj* jednak nuli. Kako uzimamo cifru po cifru varijable *broj* i dodajemo na *zbir*, svaki put skratimo *broj* za poslednju cifru, sve dok u nekom trenutku *broj* ne ostane bez cifara. Kada se to desi, onda smo sigurni da nemamo više cifara da dodajemo na tekući *zbir*, i možemo da izbacimo *zbir* na ekran. Kod 6.9 prikazuje implementaciju rešavanja problema zbira cifara broja proizvoljne dužine uz pomoć While petlje.

Kod 6.9. Zbir cifara broja bilo koje dužine sa While petljom

```
1.  Dim broj, jedinice, zbir As Integer
2.
3.  broj = InputBox("broj")
4.
5.  While broj > 0
6.     jedinice = broj Mod 10
7.     zbir = zbir + jedinice
8.     broj = (broj - jedinice) / 10
9.  Wend
10.
11. MsgBox ("zbir = " & zbir)
```

U 5. redu se vidi sintaksa korišćenja While petlje, gde se takođe proverava da li je ostalo još cifara kod ulaznog broja. Ukoliko jeste, onda se ulazi u još jednu iteraciju While petlje. Ako *broj* nema više cifara, ili drugim rečima *broj* = 0, While petlja se ne izvršava više, i program prelazi na poslednji, odnosno 11. red, gde se prikazuje rezultat.

6.6. ČITANJE I PISANJE U EXCEL ĆELIJE

Jedna od prednosti Visual Basic-a je njegova integrisanost u Microsoft Office pakete, što je najočiglednije u njegovom radu sa Excel-om. Možda je i najvažniji razlog programiranja u Visual Basic-u upravo kada treba raditi sa Excel-om. Komanda za pisanje u ćeliju je Cells(*x*,*y*) = *Z*, što u prevodu znači da ćelija u redu *x*, i koloni *y* dobija vrednost varijable *Z*. Varijabla *Z* može da bude bilo kog tipa koji može da se upiše u Excel ćeliju, što znači da brojevi, tekst i datumi mogu da se upotrebe. Kada se čita iz ćelije, onda se permutuje jednačina, i dobijemo *Z* = Cells(*x*,*y*), gde varijabla *Z* preuzme vrednost iz ćelije (*x*,*y*). U kodu 6.10 (redovi 1 do 5) se vidi kod koji ispisuje brojeve 1 do 10 u prvom redu tabele, ili drugim rečima, ćelija A1 = 1, B1 = 2, C1 = 3, …, I1 = 10. Redovi 6 do 8 predstavljaju malo drugačiju funkciju koja i čita i piše u ćelije, i pretpostavlja da su prethodni redovi već izvršeni.

Kod 6.10. Pisanje u Excel ćelije (redovi 1 do 5); čitanje i pisanje (redovi 6 do 8)

```
1.  Dim i As Integer
2.
3.  For i = 1 To 10
4.     Cells(1,i) = i
5.  Next i
6.  For i = 2 To 10
7.     Cells(1,i) = Cells(1,i-1) + i
8.  Next i
```

Kreće od druge kolone (*i* = 2), i svaka ćelija u redu dobije vrednost prethodne plus trenutne vrednosti varijable *i*. Ova petlja nema nekog praktičnog značaja, ali ilustruje koncept referenciranja ćelija i način na koji se one čitaju i pišu.

Kod 6.11 je nešto zanimljiviji jer rešava praktičan zadatak pronalaženja prvih 50 Fibonačijevih brojeva, i ispisivanja istih u prvoj koloni (A) Excel tabele. Redovi 3 i 4 diktiraju da će ćelije A1 i A2 dobiti vrednost 1. Petlja počinje od vrednost 3, i od 3. do 50. reda računa Fibonačijev broj za svaki naredni korak. Šesti red definiše da je vrednost trenutne ćelije jednaka zbiru vrednosti prethodne dve ćeljie, što je u skladu sa definicijom Fibonačijevih brojeva.

Kod 6.11. Fibonačijevi brojevi

```
1.  Dim i As Integer
2.
3.  Cells(1,1) = 1
4.  Cells(2,1) = 1
5.  For i = 3 To 50
6.      Cells(i,1) = Cells(i-1,1) + Cells(i-2,1)
7.  Next i
```

Naredni primer ilustruje kako možemo da izračunamo zlatni presek, odnosno Φ, na osnovu koda 6.11. Zlatni presek je jednak trenutnom broju u Fibonačijevoj seriji podeljeno sa prethodnim. Kod 6.12 predstavlja petlju koja prolazi od 2. do 50. reda, i u drugoj koloni tabele računa zlatni presek, pozivajući se na već izračunate vrednosti Fibonačijevih brojeva koji se nalaze u prvoj koloni. Četvrti red određuje da je ćelija u *i*-tom redu, druge kolone, jednaka podelom Fibonačijevog broja u *i*-tom redu sa prethodnim Fibonačijevim brojem, odnosno (*i*-1)-tom redu. Na primer, ako je vrednost i = 10, onda je ćelija (10,2) = ćelija (10,1) / ćelija (9,1). Deseti Fibonačijev broj je 55, a broj pre njega je 34, dakle ćelija (10,2) = 55 / 34 = 1.618 = Φ. Kako se varijabla i povećava, tako se vrednost razlomka približava i konvergira ka tačnoj vrednosti Φ, koja je $\left(1+\sqrt{5}\right)/2$, zato što se dele sve veći Fibonačijevi brojevi pa to rezultira preciznijom podelom.

Kod 6.12. Zlatni presek na osnovu Fibonačijevih brojeva

```
1.  Dim i As Integer
2.
3.  For i = 2 To 50
4.     Cells(i,2) = Cells(i,1) / Cells(i-1,1)
5.  Next i
```

6.7. SORTIRANJE

Jedna od najčešćih funkcija u informatici jeste sortiranje podataka, što znači ređanje od najmanjeg do najvećeg ili obrnuto. Postoje razne metode sortiranja, ali ćemo ovde krenuti od najprostije. Na primer, ako imamo tri varijable koje sadrže brojeve, koje želimo da poređamo od najmanje do najveće bez obzira u kom redosledu su nam predstavljene. Kod 6.13 nam predstavlja metodu sortiranja zvanu „bubble sort“, ili sortiranje pomoću „mehura“. Ovaj način sortiranja je dobio ime po mehuru koji se stavi oko svakog para sukcesivnih brojeva kada se oni upoređuju. Brojevi u „mehuru“ se permutuju u slučaju da je levi broj veći od desnog. Kod 6.13 počinje sa učitavanjem tri cela broja, *a*, *b*, i *c*, koji mogu biti uneti u bilo kom redosledu, i cilj koda je da posie završetka funkcije izbaci poređan spisak varijable, odnosno da je $a < b < c$.

Algoritam funkcioniše tako što prvo upoređuje varijable *a* i *b*. Ako je $a > b$, onda *a* preuzima vrednost *b*, i *b* preuzima vrednost *a*. Drugim rečima, njihove vrednosti se permutuju, i veća vrednost se efektivno približava desnom kraju spiska varijabli *a*, *b* i *c*. Permutovanje, ili zamena vrednosti *a* i *b* se vidi u redovim 7 do 9 u kodu 6.13. U 7. redu se koristi varijabla *temp*, koja služi da privremeno sačuva vrednost varijable *b*. Varijabla *b* onda preuzme vrednost varijable *a* (8. red), i konačno varijabla *a* preuzme vrednost varijable *temp*, koja inače čuva originalnu vrednost varijable *b*. Na ovaj način su *a* i *b* zamenile mesto, odnosno učinile „swap“ vrednosti.

Kod 6.13. „Bubble“ ili „mehur“ sortiranje tri cela broja

```
1.  Dim a, b, c, temp As Integer
2.  a = InputBox (“a”)
3.  b = InputBox (“b”)
4.  c = InputBox (“c”)
5.
6.  If (a > b) Then
7.      temp = b
8.      b = a
9.      a = temp
10. End If
11. If (b > c) Then
12.     temp = c
13.     c = b
14.     b = temp
15. End If
16. If (a > b) Then
17.     temp = b
18.     b = a
19.     a = temp
20. End If
21.
22. MsgBox (“a “ & a & “ b “ &
    b & “ c “ & c)
```

Ponavlja se slično upoređivanje u redovima 11 do 15, ali se ovog puta stavljaju varijable *b* i *c* u „mehur“. Opet se ponavlja procedura zamene za varijable *a* i *b* na kraju (redovi 16 do 20), i konačno se ispisuje rešenje u redu 22.

Slika 6.1 predstavlja primer primene koda 6.13, gde varijable *a*, *b* i *c* imaju ulazne vrednosti 11, 6 i 2. Korak 1 odgovara redovima 6 do 10 jer upoređuje i permutuje varijable *a* i *b*. Drugi korak odgovara redovima 11 do 15 jer upoređuje i permutuje varijable *b* i *c.* Treći korak odgovara redovima 16 do 20 jer upoređuje i permutuje varijable *a* i *b*, i neophodan je jer algoritam ne pravi sortiran niz posle drugog kora, kao što se i vidi iz slike 6.1. Nakon drugog koraka, najveća vrednost jeste na kraju niza varijabli, ali ostale varijable nisu sortirane. Potrebno je proći kroz preostale varijable (*a* i *b*) u trećem koraku da bi sve tri bile sortirane.

	a	*b*	*c*
Ulaz:	11	6	2
korak 1:	6	11	2
korak 2:	6	2	11
korak 3:	2	6	11

Slika 6.1. Sortiranje ulaznih brojeva 11, 6 i 2

Kako da sortiramo četiri ulazna broja umesto samo tri? Šta treba drugačije uraditi da bismo to postigli? Ako nastavimo sa logikom iz prethodnog primera, onda samo treba proširiti algoritam za nekoliko redova i postići ćemo cilj. Kod 6.14 prikazuje način proširenja prethodnog koda na soritranje četiri cela broja.

Kod 6.14. „Bubble“ sortiranje četiri cela broja

```
1.  Dim a, b, c, d, temp As I
    nteger
2.
3.  a = InputBox (“a”)
4.  b = InputBox (“b”)
5.  c = InputBox (“c”)
6.  d = InputBox (“d”)
7.
8.  If (a > b) Then
9.      temp = b
10.     b = a
11.     a = temp
12. End If
13. If (b > c) Then
14.     temp = c
15.     c = b
16.     b = temp
17. End If
18. If (c > d) Then
19.     temp = d
20.     d = c
21.     c = temp
22. End If
23. ‘ponovi proceduru za sortiranje
     3 broja
24.
25. MsgBox (“a “ & a & “ b “ & b &
    “ c “ & c & “ d « & d)
```

Počinjemo slično kao u prethodnom primeru, što znači da upoređujemo i permutujemo varijable *a* i *b*, pa *b* i *c*, zatim *c* i *d*, što se vidi u redovima 8 do 23. Ovo nam dozvoljava da najveći broj prebacimo na poziciju varijable *d*. Na ovaj način svedemo problem na prethodni, što znači da sada moramo da rešimo problem sortiranja tri varijable, jer smo već ovu četvrtu stavili na mesto koje joj pripada (poslednje). Komentar u 23. redu upravo to i opisuje, jer na tom mestu treba obaviti korake 6 do 20 iz koda 6.13 da bismo sortirali ulazne varijable. Slika 6.2 ilustruje sortiranje četiri varijable *a*, *b*, *c*, i *d*, koje imaju ulazne vrednosti 11, 15, 6 i 2.

	a	*b*	*c*	*d*
Ulaz:	11	15	6	2
korak 1:	11	15	6	2
korak 2:	11	6	15	2
korak 3:	11	6	2	15
korak 4:	6	11	2	15
korak 5:	6	2	11	15
korak 6:	2	6	11	15

Slika 6.2. Sortiranje ulaznih brojeva 11, 15, 6 i 2 koristeći „mehur“, ili “bubble sort“

Iz slike 6.2 se nazire šema za rešenje ovog problema. U prvom prolazu kroz varijable, najveća vrednost je gurnuta skroz na desni kraj spiska, odnosno u varjablu *d*. U tom trenutku, varijabla *d* je sortirana, ali treba sortirati i preostale varjable. Prolazi se drugi put kroz sve preostale varijable, gde je druga najveća vrednost gurnuta na pretposlednju poziciju, odnosno varijablu *c*. Isto se ponovi za varijable *a* i *b*, i ceo niz je sortiran. Iz priloženog se vidi da se ova procedura može automatizovati tako da se niz varijabli može sortirati bez obzira na njegovu dužinu. Ne želimo da dodajemo na kod svaki put kada je potrebno sortirati novu dužinu niza, tako da ćemo rešiti ovaj problem pozivajući se na pažljivo konstruisanu For petlju, i program koji uzima kao ulaz niz varijabli tipa Integer. Kod 6.15 prikazuje proceduru sortiranja brojeva koji se čitaju iz četvrtog reda Excel tabele, i ispisuje rezultate u 5. red iste tabele.

Kod 6.15. „Bubble sort“ niza proizvoljne dužine koristeći For petlju

```
1.  Sub BubbleSort()
2.
3.  Dim n, i, j, temp As Integer
4.  Dim brojevi() As Integer
5.  ReDim brojevi(n) As Integer
6.
7.  n = 10
8.  For j = 1 To n
9.      brojevi(j) = Cells(j, 4)
10. Next j
11.
12. For i = n To 1 Step -1
13.     For j = 1 To i-1
14.         If brojevi(j) > brojevi(j + 1) Then
15.             temp = brojevi(j + 1)
16.             brojevi(j + 1) = brojevi(j)
17.             brojevi(j) = temp
18.         End If
19.     Next j
20. Next i
21.
22. For j = 1 To n
23.     Cells(j, 5) = brojevi(j)
24. Next j
25.
26. End Sub
```

Prva novina se primećuje u redovima 4 i 5, gde se deklariše niz tipa Integer pod imenom *brojevi*. Komanda ReDim se koristi kada deklarišemo niz nepoznate dužine, koji se može odrediti kasnije, što se i dešava u 7. redu, gde se određuje da je $n = 10$. Niz dužine 10 se učitava iz 4. kolone Excel tabele zbog For petlje koja se vidi u redovima 8 do 10. Sortiranje ovog niza se vidi u redovima 12 do 20, i ispis rezultata se dešava u redovima 22 do 24.

Dve For petlje koje vrše sortiranje zaslužuju dodatno objašnjenje. Spoljna For petlja se odbrojava unazad od n, odnosno od 10 do 1. Ovo je neophodno jer po algoritmu „bubble sort“-a, prolazimo kroz ceo niz prvi put, pa kroz ceo niz bez poslednjeg člana drugi put, pa kroz ceo niz bez poslednja dva člana treći put itd., dok ne prođemo za sve moguće dužine (od 10 do 1) jedanput. Unutrašnja For petlja dozvoljava i-1 upoređivanja za svaku iteraciju spoljašnje petlje jer odbrojava od 1 do i-1 u svakoj iteraciji spoljašnje For

petlje. Spoljna odbrojava unazad, što znači da će unutrašnja da broji od 1 do 9 prvi put, pa od 1 do 8 drugi put itd., za svaku iteraciju spoljašnje. Ovu vrstu odbrojavanja smo i hteli jer koristimo ove brojeve za indeksiranje pozicija u nizu *brojevi* radi upoređivanja i permutovanja parova brojeva u svakom „mehuru“.

Četrnaesti red sprovodi upoređivanja trenutnog broja (*brojevi*(*j*)) sa sledećim brojem (*brojevi*(*j*+1)) i ako je prvi veći od drugog, njihove vrednosti menjaju mesta na isti način koji je već viđen u kodu 6.14. Najteži deo ovog koda je bio dizajn sistem za indeksiranje svih lokacija u nizu *brojevi*. Slične zagonetke u dizajnu se javljaju u programiranju sa nizovima. Duboko razumevanje logike indeksiranja je potrebno da bi se ovakvi zadaci rešili, a stiče se jedino vežbom!

6.8. NAJVEĆI ZAJEDNIČKI DELILAC I NAJMANJI ZAJEDNIČKI FAKTOR

Najveći zajedniči delilac (NZD) brojeva *a* i *b* je najveći broj *x* koji zadovoljava *a* mod *x* = 0 i *b* mod *x* = 0. Na primer, NZD od 8 i 12 je 4, ili pomoću pravilne notacije, NZD (8, 12) = 4. NZD se može koristiti kada želimo da uprostimo razlomak, jer je NZD (42, 56) = 14, pa dobijemo [3]:

$$\frac{42}{56} = \frac{3 * 14}{4 * 14} = \frac{3}{4}$$

Postoje i druge upotrebe NZD, ali nas ovde interesuje kako možemo da napišemo kratak program koji uzima dva ulazna broja (*a* i *b*) i nalazi njihov NZD. Kod 6.16 upravo pruža prosto rešenje ovog problema, koje prolazi kroz sve moguće brojeve dok se ne pronađe onaj pravi.

Kod 6.16. Najveći zajednički delilac – prosto rešenje

```
1.  Dim a, b, manji, i, broj As Integer
2.  Dim nađen As Boolean
3.  a = 24
4.  b = 20
5.
6.  If a < b Then
7.     manji = a
8.  Else
9.     manji = b
10. End If
11.
12. broj = -1
13. i = manji
14. nađen = False
15.
16. While Not nadjen
17.    If (a Mod i = 0) And (b Mod i = 0) Then
18.       broj = i
19.       nađen = True
20.    Else
21.       i = i - 1
22.    End If
23. Wend
24.
25. MsgBox (broj)
```

Nakon deklaracija varijabli (redovi 1 i 2), i određivanja početnih vrednosti varijabli *a* i *b* (redovi 3 i 4), određuje se koji je manji broj između *a* i *b* (redovi 6 do 10). Manji broj nam treba jer krećemo od njega i odbrojavamo unazad sve dok ne pronađemo prvi broj koji zadovoljava uslove NZD-a. U 12. redu označavamo da *broj*, odnosno izlazna varijabla, nije pronađena, tako što joj dodelimo vrednost -1. Varijabla *i* dobija vrednost manjeg od ulaznih brojeva u 13. redu, i Boolean varijabla *nađen* dobija vrednost *false*, jer *broj* nije još određen.

Sedamnaesti red While petlje proverava ključni uslov NZD algoritma, naime da li je broj i deljiv bez ostataka sa brojevima *a* i *b*. Ako su oba uslova zadovoljena, onda se i dodeljuje izlaznoj varijabli *zbir*, i varijabla *nađen* se setuje na True da bi se While petlja okončala. U suprotoj situaciji, uslovi nisu zadovoljeni i varijabla *i* se umanji za 1, i uslov se opet proverava. U najgorem slučaju, varijable *a* i *b* nemaju NZD koji je veći od 1, ali mora biti minimum 1 jer bilo koji celi brojevi su deljivi sa 1. Zbog ove osobine smo sigurni da će se While petlja okončati, i program će naći traženi rezultat.

Kod 6.17. Najmanji zajednički faktor

```
1.  Dim a, b, veci, i, broj As Integer
2.  Dim nađen As Boolean
3.  a = 12
4.  b = 8
5.
6.  If a > b Then
7.     veci = a
8.  Else
9.     veci = b
10. End If
11.
12. i = veci
13. nađen = False
14.
15. While Not nađen
16.    If (i Mod a = 0) And (i Mod b = 0) Then
17.       broj = i
18.       nadjen = True
19.    Else
20.       i = i + 1
21.    End If
22. Wend
23.
24. MsgBox (broj)
```

Najmanji zajedniči faktor (NZF) brojeva *a* i *b* je najmanji broj *x* koji zadovoljava *x* Mod *a* = 0 i *x* Mod *b* = 0. Na primer, NZF od 8 i 12 je 24, ili pomoću pravilne notacije, NZF(8, 12) = 24. Kao u prethodnom kodu, algoritam za računanje NZF je predstavljen u kodu 6.17 i slično funkcioniše. Ovog puta se kreće od većeg broja, i povećava se brojač za jedan u potrazi za brojem koji zadovoljava pomenute ulove u 16. redu. Ako su oba ulazna broja prosta, onda je njihov NZF u najgorem slučaju *a* * *b*, ali i tada postoji, što znači da će se While petlja okončati, i da će algoritam ipak naći barem neko rešenje. Poslednji red ispisuje rezultat na ekranu.

6.9. KLJUČNI TERMINI

Algoritam – u programerskom smislu, konačan niz nedvosmislenih instrukcija koje se izvršavaju jedna po jedna, u ograničenom vremenu radi rešavanja određenog problema.

Visual Basic – programski jezik okruženja Microsoft-a.

Iteracija – čin procesa ponavljanja bloka koda.

Blok koda – skup zajedničko grupisanog koda ili naredbi koje se izvršavaju uzastopno kao neprekidna celina. Smatra se logičnom radnom jedinicom u programu.

Petlja – niz ili blok komandi koji se definiše jednom, ali se izvršava određen ili neodređen broj puta, u zavisnosti od vrste petlje (For, While, Do While).

Varijabla – deklarisana ili imenovana memorijska lokacija koja je podobna za skladištenje određenog podatka [4].

Checksum – mali podatak „c" koji se izvede iz veće količine podataka „p", i proverava integritet i postojanja grešaka u „p" koje potencijalno nastaju u skladištenju ili slanju.

Pitanja za ponavljanje

1. Gde se koristi Visual Basic?
2. Šta je If klauzula?
3. Gde se koristi checksum?
4. Šta je iteracija?
5. Šta je For petlja?
6. Šta je While petlja, i zašto se koristi umesto For petlje?
7. Zašto se radi sortiranje podataka?
8. Kako se računa najveći zajednički delilac?
9. Kako se računa najmanji zajednički faktor?

Literatura

[1] Microsoft, „Data Type Summary (Visual Basic)," 20 7 2015. [Na mreži]. https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/data-types/data-type-summary. [Poslednji pristup 7 2018].

[2] Wikipedia, „Lunov Algoritam," 22 12 2016. [Na mreži]. https://sr.wikipedia.org/wiki/Lunov_algoritam. [Poslednji pristup 7 2018].

[3] Wikipedia, „Greatest common divisor," 8 2018. [Na mreži]. https://en.wikipedia.org/wiki/Greatest_common_divisor. [Poslednji pristup 8 2018].

[4] Wikipedia, „Variable (computer science)," 30 5 2018. [Na mreži]. https://en.wikipedia.org/wiki/Variable_(computer_science). [Poslednji pristup 8 2018].

7. BAZE PODATAKA

Baze podataka su danas široko rasprostranjene, predstavljaju osnovu informacionih sistema, omogućavaju izradu dinamičkih web sajtova i podržavaju sisteme elektronske trgovine. Za realizaciju baza podataka koriste se tehnike modelovanja kojima se realan svet (konkretan sistem) svodi na konačan broj povezanih objekata. Rad sa bazama podataka omogućavaju sistemi za upravljanje bazama podataka koji podržavaju kreiranje baze i njenih objekata, manipulaciju sa podacima i kontrolu prava pristupa. U ovoj lekciji su opisani osnovni koncepti u radu sa bazama podataka.

7.1. UVOD

Baze podataka (BP) se koriste za prikupljanje, organizovanje, čuvanje i manipulaciju podacima na osnovu kojih se dobijaju nove informacije. Svakodnevno ih koriste pojedinci putem ličnih računara, radne grupe putem mrežnih servera i svi zaposleni putem aplikacija iz različitih informacionih sistema. Bazama podataka pristupaju udaljeni korisnici korišćenjem tehnologija kao što su govorni automati, web čitači (browser-i), mobilni telefoni i sl. Zbog velike konkurencije u svim oblastima poslovanja, može se očekivati da tehnologija baza podataka dobije još veći značaj.

Izučavanje BP i sistema za upravljanje bazama podataka jesu osnova za izučavanje informacionih sistema. Stručnjak za informacione sisteme mora biti spreman da analizira potrebe preduzeća i da dizajnira i implementira baze podataka u okviru razvoja informacionog sistema jedne organizacije. Široko rasprostranjeno korišćenje BP vezanih za Internet sajtove, koji vraćaju dinamičke informacije korisnicima web sajta, zahteva od projektanta da razume ne samo kako da poveže BP sa sajtom već i kako da je obezbedi tako da se njenom sadržaju može pristupiti ali ne i izmeniti od strane spoljnih korisnika.

Moderne baze podataka su čuvaju na računaru, ali to nije bitno za samu definiciju. Na primer, adrese poznanika i prijatelja, kolekcija filmova, telefonski imenik itd. jesu baze podataka (mada ih većina ljudi tako ne zove). Međutim, smeštanje baze podataka na računar omogućava lakšu i bržu obradu podataka i dobijanje željene informacije. Karakterističan je primer sa telefonskim imenikom koji se nalazi na papiru. Jednostavno je pronaći telefonski broj željene osobe, ali je znatno teže pronaći ime osobe na osnovu telefonskog broja. Ako je telefonski imenik veći (više smeštenih podataka) prethodni problem se dodatno usložnjava. Pored osnovnih informacija, iz odgovarajuće baze podataka se mogu dobiti i posebne informacije. Na primeru telefonskog imenika mogu se izlistati podaci za sve osobe po željenom imenu, mogu se izlistati sve osobe kojima telefonski broj počinje npr. sa 2, osobe kojima se telefonski broj završava sa 45 i još mnogo toga.

Živimo u informacionom dobu i okružuje nas veliki broj podataka. Upravo baze podataka omogućavaju da se iz ogromne količine podataka mogu dobiti korisne i upotrebljive informacije. Što je broj podataka koji se čuvaju i obrađuju veći, to su prednosti upotrebe baza podataka izraženije.

7.2. OSNOVNI KONCEPTI I DEFINICIJE

Baze odataka se može definisati kao organizovani skup logički povezanih podataka. Ona može biti bilo koje veličine i kompleksnosti. Na primer, prodavac može da ima ličnu BP vezanu za kupce koja se sastoji od nekoliko megabajta podataka. Preduzeće koje zapošljava hiljadu i više ljudi može da ima veoma veliku BP od nekoliko terabajta podataka (jedan terabajt = 1012 bajtova). Veoma velika skladišta podataka imaju više od petabajta podataka (1 petabajt = 1015 bajtova). To je dobro struktuirana kolekcija podataka, koja postoji jedno određeno vreme, koja se održava i koju koristi više korisnika ili programa.

7.2.1. Podatak

Pod terminom podatak se podrazumeva činjenica o nekom predmetu ili događaju koja se može zabeležiti i sačuvati. Na primer, u bazi podataka nekog prodavca podaci bi bile činjenice kao što su ime, adresa i broj telefona kupca. Ovakav tip podatka se zove struktuirani podatak. Najvažniji struktuirani podaci su brojevi, karakteri i datumi (vreme). Današnje baze podataka pored struktuiranih podataka sadrže i druge vrste podataka kao što su razna dokumenta, mape, fotografije, zvuk, čak i video zapise. Na primer, u bazi podataka nekog prodavca mogla bi se naći i slika kupca. Takođe bi se mogao naći zvučni ili video zapis poslednjeg razgovora sa kupcem. Ova vrsta podatka se naziva nestruktuirani podatak ili multimedijalni podatak.

Pod podatkom se podrazumeva činjenica prihvaćena kakva jeste. Podatak sam po sebi nema značenje, tek kada se interpretira nekom vrstom sistema za obradu podataka poprima značenje i postaje informacija. Tipično, termin “podatak” se odnosi na ono što je u bazi podatak. Računar vrši obradu podataka, prema zadatom programu, te se na osnovu saznanja sadržanih u podacima, a kao rezultat algoritma obrade, stiču nova saznanja - informacije.

7.2.2. Informacija

Termini podatak i informacija su usko povezani i često se koriste kao sinonimi. Međutim, korisno ih je razlikovati. Informaciju definišemo kao podatak koji je bio obrađen na takav način da se znanje osobe koja koristi podatak povećalo. Na primer, razmotrimo sledeći spisak činjenica:

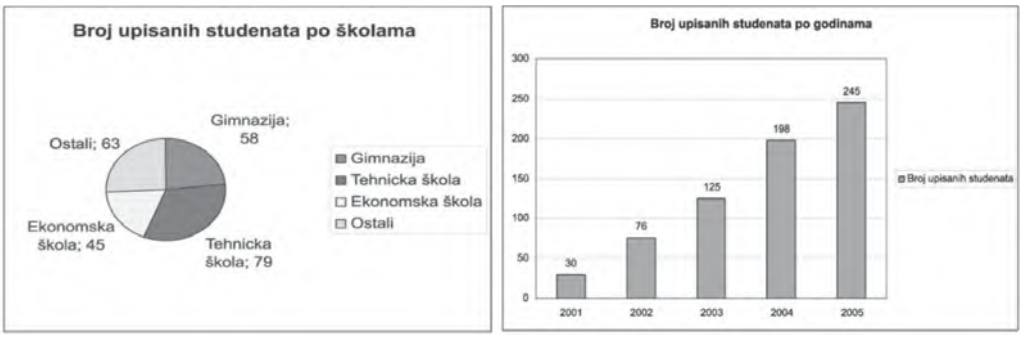
Petar Petrović	1506991710325
Marko Marković	0211998850123
Janko Janković	1112015830456
- - - - - - - - - - -	- - - - - - - - - - -

Prikazane činjenice po definiciji predstavljaju podatke, ali su ovi podaci u sadašnjoj formi beskorisni. Organizovanjem prethodnih podataka, dodavanjem još nekoliko novih i njihovim uređivanjem, prepoznaje se spisak upisanih studenata. Na ovaj način se dolazi do informacije koja je korisna npr. upravi fakulteta, profesorima, studentskoj službi i sl.

Ime i prezime	JMBG	Fakultet	Godina upisa
Petar Petrović	1506991710325	FIR	2017
Marko Marković	0211998850123	PFB	2015
Janko Janković	1112015830456	FTHM	2017
- - - - - - - - - - -	- - - - - - - - - - -	TF	2016

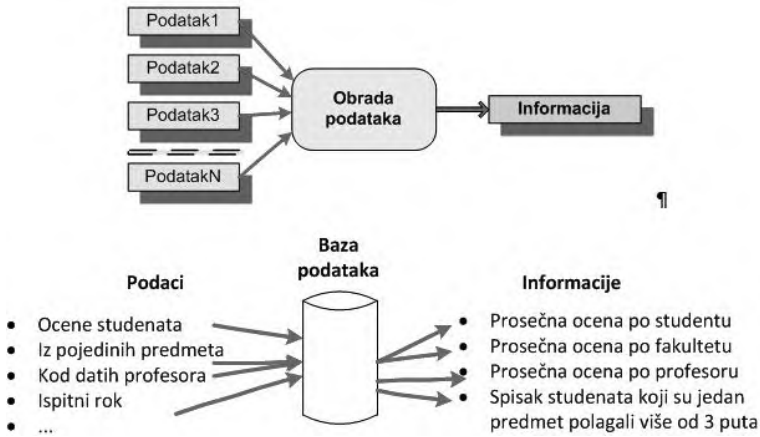
Tabela 7.1. Tabelarni prikaz podataka iz BP - informacija o upisu

Drugi način da se iz podataka dobiju informacije je da se podaci sumiraju ili na neki drugi način obrade i prezentuju. Na primer, na slici 7.1 se vide sumirani podaci o upisu studenata prikazani u vidu grafičke informacije. Ova informacija se može iskoristiti kao osnova za odlučivanje o dodavanju novih predavanja ili o zapošljavanju novog nastavnog kadra. Moderne baze podataka vrlo često sadrže i podatke i informacije. Podaci se često obrađuju i čuvaju u obrađenoj formi i služe za pomoć pri donošenju odluka, a takvim podacima (informacijama) se najbrže pristupa.



Slika 7.1. Grafički prikaz podataka iz BP - informacija o upisu

Podaci obrađeni tako da dobijaju značenje čine informaciju. Informacija koja je precizna, relevantna, i dobijena na vreme koristi se za donošenje odluka.



Slika 7.2. Obradom prikupljenih podataka nastaje informacija

7.2.3. Metapodaci - podaci o podacima (metadata)

Podaci koji se prikupljaju i čuvaju u bazi podataka često se nazivaju i podaci krajnjih korisnika (end user data). Metapodaci su podaci koji opisuju svojstva ili karakteristike podataka krajnjih korisnika i kontekst tih podataka. Neka tipična svojstva podataka su naziv (ime) podatka, definicija (tip), dužina (veličina) i dozvoljene vrednosti (ograničenja).

Field	Type	Null	Key	Default	Extra
Sifra	int(11)	NO	PRI	NULL	auto_increment
Naziv	varchar(45)	YES		NULL	
Cena	float	NO		NULL	
JedinicaMere	varchar(10)	NO		KG	
ZemljaPorekla	varchar(45)	YES		NULL	
Kategorija_Rb	int(11)	NO	MUL	NULL	

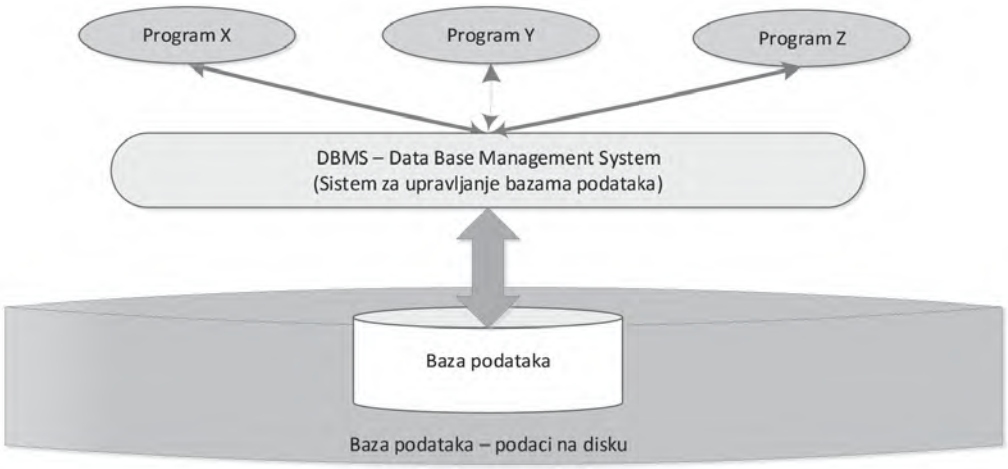
Slika 7.3. Primer metapodataka

Metapodaci opisuju svojstva podataka u bazi. Kao i sami podaci, metapodaci se takođe nalaze u bazi podataka. Metapodaci ne prikazuju ni jedan podatak. Slika 7.3 prikazuje metapodatke za tabelu Proizvod (Sifra, Naziv, Cena, JedinicnaMera, ZemljaPorekla, Kategorija_RB) iz MySQL softvera. Oni omogućavaju dizajnerima i korisnicima baza podataka da razumeju koji podaci postoje u bazi, šta oni znače i koja je razlika između podataka koji na prvi pogled izgledaju isto.

7.2.4. Sistem za upravljanje bazama podataka

Sistem za upravljanje bazama podataka - DBMS (Data Base Management System) je softver koji se koristi za kreiranje, održavanje i manipulisanje podacima, kao i za kontrolu prava pristupa bazi podataka. DBMS omogućava krajnjim korisnicima i programerima da dele podatke, tj. omogućava da se podaci koriste od strane više aplikacija, a ne da svaka aplikacija ima svoju kopiju podatka sačuvanu u posebnim datotekama. DBMS takođe pruža mogućnost kontrole pristupa podacima, osigurava integritet podataka, uspostavlja kontrolu kod konkurentnog pristupa i vrši oporavak baze podataka. Programeri aplikacija za rad sa bazama podataka ne moraju da poznaju detalje o načinu zapisa baze podataka na disku, ne moraju da formulišu algoritme za efikasan pristup podacima, niti su opterećeni bilo kakvim aspektima oko upravljanja podacima u bazi podataka.

Termini BP i upravljanje bazom podataka se ponekad mešaju. BP je uvek skup činjenica, a ne računarski program. DBMS je softver koji je uveden kao interfejs između korisnika (programa, aplikacija) i zapisa baze podataka na disku. Korisnički programi ne pristupaju podacima direktno, već komuniciraju sa ovim softverom. DBMS upravlja strukturom baze podataka: definiše objekte baze, njihova svojstva (atribute), dozvoljene vrednosti atributa, veze između objekata, ograničenja nad objektima i međusobnim vezama. Omogućava manipulaciju podacima u bazi: unošenje, brisanje i izmene, tj. omogućava njeno održavanje. Kontroliše pristup podacima: ko može da pristupi podacima, kojim podacima i šta može sa njima da radi. DBMS dozvoljava deljenje BP između više aplikacija i korisnika. Uobičajeno je da kada se govori o softveru za baze podataka, onda se misli upravo na DBMS.

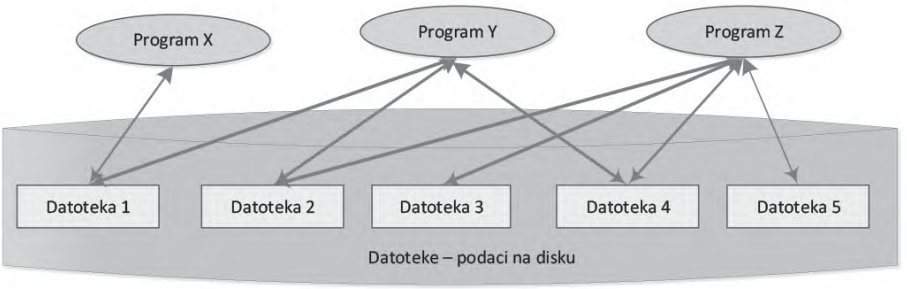


Slika 7.4. DBMS je interfejs između (aplikacija) korisnika i zapisa baze podataka na disku

7.3. KLASIČAN SISTEM ZASNOVAN NA DATOTEKAMA

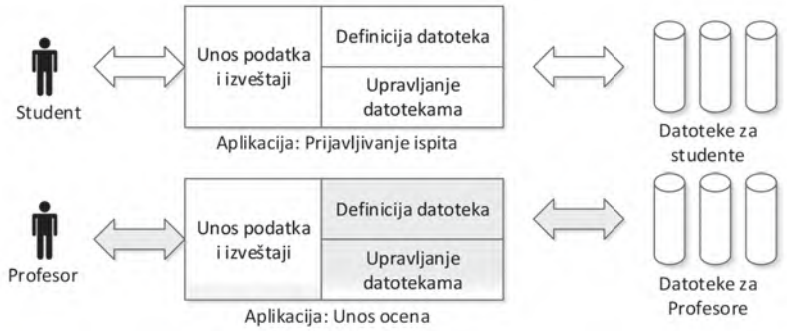
Kada su računari počeli da se koristiti za obradu podataka, nisu postojale baze podataka. Računari su u to vreme bili znatno slabiji nego današnji personalni računari, zauzimali su čitavu prostoriju i koristili su se skoro isključivo za naučna izračunavanja. Postepeno su računari uvođeni u poslovni svet. Da bi bili od koristi za poslovne aplikacije, računari moraju da skladište, manipulišu i preuzimaju velike datoteke podataka. Kako su aplikacije postajale sve kompleksnije, postalo je očigledno da klasični sistemi zasnovani na datotekama imaju veliki broj nedostataka i ograničenja. U većini poslovnih aplikacija danas se umesto klasičnog sistema zasnovanog na datotekama koriste baze podataka.

Klasičan sistem obrade podataka zasnovan na datotekama i programskim jezicima prikazan je principskom šemom na sledećoj slici. Programi su direktno povezani sa datotekama i svaki program mora da poznaje detaljan zapis podataka na disku. Da bi objasnili osnovne karakteristike sistema zasnovanog na datotekama, posmatrajmo informacioni sistem jednog univerziteta. Standardna funkcionalnost takvog sistema je da studenti preko interneta prijavljuju ispite, a profesori nakon ispita unose ocene. U klasičnoj obradi podataka sa datotekama postojao bi veliki broj datoteka za studente i za profesore. Neke od datoteka bi bile zajedničke.



Slika 7.5. Klasičan sistem obrade podataka zasnovan na programskim jezicima i datotekama

Aplikacije za prijavu ispita i za unos ocena bi bile opterećene velikim brojem preciznih definicija datoteka. Ove definicije (format i način pristupa datoteci) bi neminovno proširivale osnovni kod aplikacije, tako da bi sama funkcionalnost oko prijave ispita i unosa ocena bila praktično u drugom planu. Kako se u nekoj aplikaciji povećava broj linija koda, sama aplikacija postaje nepregledna i sklona greškama, od kojih su najteže tzv. logičke greške. Testiranje takvih aplikacija i ispravljanje logičkih grešaka je veoma složeno. Pošto postoji veliki broj datoteka, neminovno postoji i velika redudansa (ponavljanje) podataka. Isti podaci se mogu naći u različitim datoteka, a održavanje takvih podataka je teško i greške u radu su neminovne.



Slika 7.6. Klasična obrada podataka zasnovana na sistemu datoteka

7.3.1. Nedostaci sistema zasnovanog na datotekama

Postoji više mana koje su tipične za sistem koji je zasnovan na datotekama i klasičnim programskim jezicima. Na prvom mestu to je zavisnost između programa i podataka. Opisi datoteka se čuvaju u okviru svakog programa koji pristupa toj datoteci. Kao posledica ovoga, svaka promena koja se napravi u datoteci, a odnosi se na strukturu, momentalno podrazumeva da se mora menjati i opis datoteka u svakom programu koji pristupa tim podacima. Pretpostavimo da se veličina polja "Adresa studenta" menja sa 20 karaktera na 30 karaktera. Zbog toga se opis datoteke u svakom programu mora ažurirati. Često je teško i samo lociranje svih programa na koje je uticala ovakva promena. Što je još gore pri ažuriranju se često prave greške.

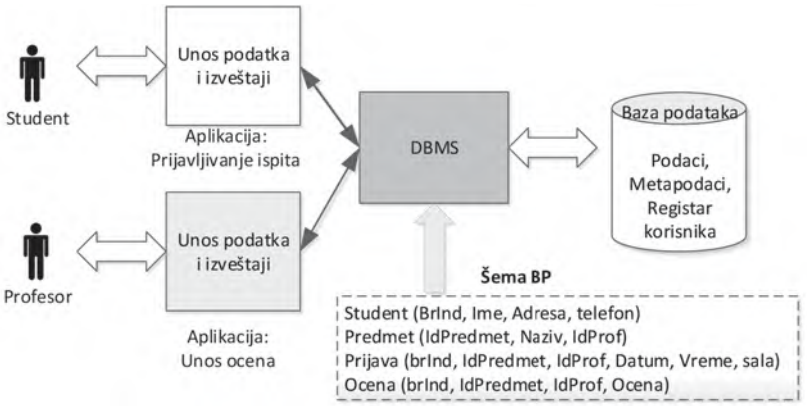
Redundansa podataka: Kako se u prikazanom sistemu procesi odvijaju nezavisno jedni od drugih, ponavljanje podataka nije izuzetak već je pravilo. Zbog nepotrebnih duplikata potreban je veći prostor za njihovo čuvanje kao i više truda i rada pri njihovom ažuriranju. Neplanirana redundansa podataka može da dovede do gubitka podataka. Na primer, isti podaci mogu se voditi pod različitim imenima atributa u različitim dokumentima, ili obrnuto, isto ime se može koristiti za različite vrste podataka.

Ograničenost deljenja podataka: Korišćenjem klasičnog sistema zasnovanog na datotekama, svaki proces ima svoje datoteke i korisnici nemaju šansu da međusobno dele podatke sa korisnicima iz drugih procesa. Dugo vreme za razvoj: Sa klasičnim sistemom zasnovanom na datotekama postoji mala šansa za korišćenje prethodnih razvojnih dostignuća. Svaka nova aplikacija zahteva od projektanta da krene od nule. Svaki put je neophodno definisati nove formate i opise podataka i pisati kod za pristup podacima za svaki program. Ovako veliko vreme za razvoj nije u skladu sa današnjim poslovnim potrebama, gde je svaki minut bitan da bi se postigao uspeh.

Teško održavanje programa: Skup svih prethodno navedenih nedostataka dovodi do preterane potrebe za održavanjem programa. Čak 80% budžeta predviđenog za razvoj sistema zasnovanog na datotekama odlazi na njegovo održavanje. Zbog toga, naravno, ostaje jako malo prostora za razvoj novih aplikacija.

7.4. PRISTUP ZASNOVAN NA BAZAMA PODATAKA

Pristup zasnovan na BP potencira integraciju i deljenje podataka između svih odeljenja jedne organizacije. Ovaj pristup zahteva potpunu promenu u načinu razmišljanja, počevši od najvišeg nivoa upravljanja. Takva promena načina razmišljanja za većinu organizacija je veoma teška.



Slika 7.7. Blok šema informacionog sistema zasnovanog na bazama podataka

Da bi objasnili pristup zasnovan na BP posmatrajmo prethodno razmatrani zastareli informacioni sistem koji se klasično zasnivao na datotekama. Koncept sistema zasnovanog na BP prikazan je na slici 7.7. Odmah se može uočiti da podaci koji su prethodno čuvani u više različitih datoteka, sada su integrisani u jedinstvenu BP. Metapodaci, koji opisuju ove podatke, nalaze se zajedno sa njima u BP. Sistem za upravljanje BP pruža mogućnost stvaranja različitih pogleda na istu (ili više) BP za različite korisnike u organizaciji. DBMS dozvoljava korisnicima da dele, pretražuju, pristupaju i ažuriraju integrisane podatke.

7.5. PRIMENE BAZE PODATAKA

Baze podataka imaju široku primenu. Karakteristični primeri su dati u nastavku.

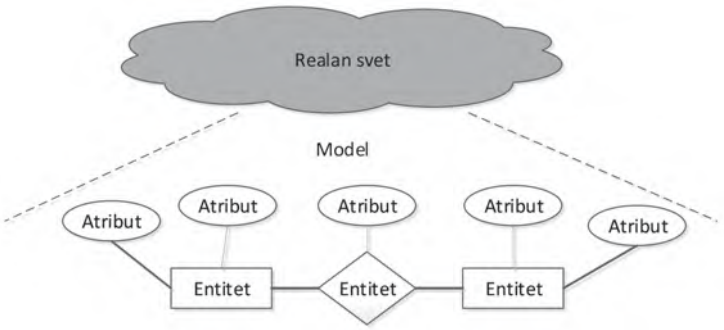
- Bankarstvo: Informacije o klijentima, računi, krediti, bankarske transakcije i sl.
- Avio-saobraćaj: Informacije o letovima, rezervacija karata i sl. Avio-kompanije su među prvima počele da koriste baze podataka kroz geografski distribuiran način – putem terminala i računara koji su geografski široko rasprostranjeni i koji pristupaju centralizovanim podacima putem telekomunikacionih kanala i Interneta.
- Univerziteti: Informacije o studentima, upis, ocene, kursevi itd.

- Transakcije putem kreditnih kartica: Plaćanje i generisanje mesečnih izveštaja.
- Telekomunikacije: Evidentiranje obavljenih razgovora, generisanje mesečnih računa, podrška za "pripejd" kartice, čuvanje informacija o komunikacionoj mreži i sl.
- Finansije: Prodaja i kupovina finansijskih instrumenata kao što su akcije i obveznice.
- Trgovina: Podaci o kupcima, proizvodima i različiti katalozi sa cenama.
- Proizvodnja: Upravljanje lancima snabdevanja, praćenje proizvodnje, stavki o zalihama u magacinima ili prodavnicama, generisanje otpremnica, faktura i sl.
- Ljudski resursi: Informacije o zaposlenima, plate, porezi na plate i beneficije i sl.

Iz navedene liste jasno je da su baze podataka danas suštinski deo skoro svih organizacija. U ranom periodu primene baza podataka mali broj ljudi je imao direktan kontakt sa bazama. Pristup se odnosio na posedovanje štampanih izveštaja ili se pristup ostvarivao posredno npr. preko službenika kod bakarskih transakcija ili rezervacije karata. Prvi masovni direktni pristupi bazama podataka su ostvarivani putem servisa preko telefonske linije. Razvoj Internet servisa krajem devedesetih godina povećao je direktan pristup korisnika bazama podataka. Web interfejsi su danas omogućili niz usluga koje se mogu realizovati u realnom vremenu (on-line). Korisnički interfejsi su sakrili detalje direktnog pristupa bazama, tako da većina ljudi danas nije ni svesna da radi sa bazama podataka koje su postale bitan deo njihovog života. Značaj baza podataka ogleda se i u tome što su softverske kompanije, kao što je Oracle postale jedne od najvećih na svetu, a softverske kompanije kao što su Microsoft i IBM poklanjaju veliku pažnju razvoju sopstvenih sistema za upravljanje bazama podataka.

7.6. MODELOVANJE

Informacioni sistemi pojedinih firmi omogućavaju upravljanje podacima koji su bitni za njeno poslovanje. Broj sopstvenih podataka i podataka iz okruženja je ogroman te je nemoguće sve podatke i sve uočene detalje opisati i sačuvati unutar informacionog sistema. Postupkom selekcije identifikuju se i čuvaju samo relevantni podaci. Time se dolazi do pojma modela podataka. Modeli su čovekovo sredstvo pojednostavljivanja problema i njegovo posmatranje samo sa stanovišta bitnih za ciljeve analize. Modeli sadrže konačan broj objekata, a svaki objekat je opisan sa konačnim brojem atributa. Takav model realnog sveta zasniva se na simbolima i zove se konceptualni model podataka.



Slika 7.8. Realan svet i njegov model

Modelovanje podataka se radi paralelno sa analizom potreba. Kako se informacije prikupljaju, objekti se identifikuju, dodjeljuju im se imena koristeći termine bliske krajnjim korisnicima. Objekti se onda modeluju i analiziraju korišćenjem dijagrama objekti-veze (ER dijagrami). Dijagram se može pregledati od strane dizajnera i krajnjeg korisnika da bi se osigurala njegova kompletnost i tačnost. Ako model nije tačan, modifikuje se, što ponekad zahteva da se prikupe dodatne informacije. Ciklus pregledanja i modifikovanja se nastavlja sve dok se ne dobije potvrda da je model korektan.

7.6.1. Razvoj konceptualnih modela

Objekti iz realnog sveta se u računarskoj primeni opisuju pomoću podataka. Podaci su zato apstrakcija realnosti, tj. sredstva za kodiranje osobina objekata iz realnog sveta. Modelovanje, kao postupak kojim se realni svet svodi na određeni broj podataka, predstavlja kompleksan posao i sastoji se iz više koraka:

U prvom koraku se mnoštvo objekata iz realnog sveta redukuje na manji skup objekata, koji će činiti objekte modela (izbor, selekcija). Npr. objekti mogu biti student, predmet, profesor, studentska služba, polaganje ispita i sl. U procesu selekcije ovaj broj objekata se može redukovati na manji broj, ako je cilj samo praćenje uspešnosti studiranja na fakultetu. Time se složenost realnog sistema smanjuje. Selekcija se ne odnosi samo na objekte nego i na njihove osobine, kao i na međusobne veze između objekata. Svakom objektu u realnom svetu, svakoj vezi između uočenih objekata, kao i svakom atributu uočenog objekta ili veze dodeljuje se ime (imenovanje objekata i njihovih atributa).

Rezultat navedenih koraka modelovanja zove se konceptualni model. On sadrži, za posmatrani problem iz realnog sveta, sve relevantne tipove objekata, njihove osobine i međusobne veze. Rezultati proučavanja modela podataka doveli su do saznanja da svaki model podataka ima tri neodvojive komponente:

- strukturu podataka,
- operacije nad podacima,
- ograničenja (constraints).

Struktura i ograničenja, za razliku od operacija, opisuju stanje realnog sistema, tj. predstavljaju statički opis stanja sistema. Strukturu modela čine objekti, njihova svojstva, veze između objekata i svojstava tih veza. Operacije nad podacima u modelu su, u stvari, operacije nad strukturom modela kojima se izražava dinamika realnog sistema. Operacije izražavaju kretanje i promene tj. dinamiku realnog sistema. Ograničenja su pravila koja razdvajaju dopuštena od nedopuštenih stanja realnog sistema i u svojoj prirodi deo su strukture modela podataka.

7.6.2. Entiteti

Modelima podataka nastoji se preslikati realan sistem. Realan sistem sastoji se od objekata iz realnog sveta i njihovih veza između kojih se uspostavljaju različiti odnosi. Pod entitetom se podrazumeva sve što se može jednoznačno odrediti, identifikovati i razlikovati. Tako široko postavljena definicija pokazuje da entitet može biti svaki "realan" ili "apstraktan" objekt o kojem u određenom trenutku razmišljamo. Entitet je realan ako fizički, stvarno postoji. Najopštije se može tvrditi da su granice entiteta u modelu podataka određene ljudskim pogledom i načinom razmišljanja.

Svaki entitet uočen u realnom sistemu ima svoje osobine koje ga čine složenim i njihove vrednosti omogućavaju razlikovanje entiteta. Svojstvo entiteta uključuje dva elementa - atribut i vrednost atributa (npr. entitet Student ima atribute: Ime, Prezime, Broj indeksa, Adresu, Telefon i sl. i vrednosti Marko, Marković, 123/17, Danijelova, 32, 011/376-543 respektivno). Svaki put kada se promeni vrednost atributa, potrebno je promenu evidentirati, tj. ažurirati tu vrednost atributa za dati entitet.

Atribut opisuje entitet. Jedno konkretno pojavljivanje atributa naziva se vrednost. Ako je atribut dovoljno složen, tako da ima svoje dodatne atribute, može se posmatrati kao novi entitet.

Domen atributa je skup svih mogućih vrednosti koje atribut može poprimiti.

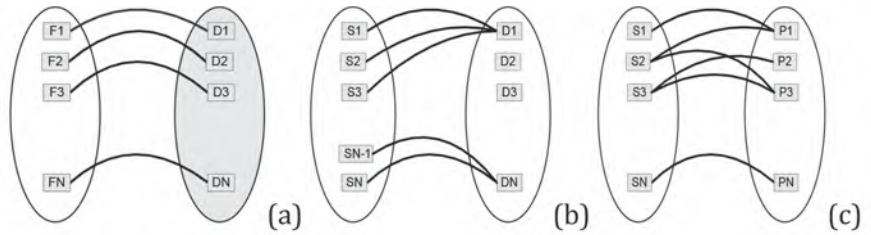
Primarni ključ je jedan ili više atributa čija vrednost jednoznačno određuje primerak entiteta. Na primer, za entitet Auto, primarni ključ je atribut "registarski broj". Dva različita člana ili primerka entiteta ne mogu imati isti primarni ključ. Primarni ključ je jedinstven za svakog člana entiteta. Na primer, za entitet Student primarni ključ bi mogao biti „broj indeksa".

7.6.3. Veze između entiteta

BP se ne odnosi samo na pojedinačne objekte nego i na odnose između objekata. U realnom sistemu objekti nisu međusobno izolovani, nego se nalaze u međusobnoj interakciji. Student se upisuje na fakultet, sluša predavanja iz pojedinih predmeta, prijavljuje polaganje ispita, polaže ispit itd. To su primeri logičkih i realnih veza između objekata, koje slede iz realnih odnosa u posmatranom sistemu studiranja na jednom

fakultetu. Istražimo jedan skup odnosa između studenata koji slušaju predavanja kod određenog profesora. Postavlja se pitanje šta su u takvim odnosima objekti, koje su njihove osobine (atributi) i kako prikazati njihove odnose.

(a) (b) (c)



Slika 7.9. Preslikavanje entiteta 1:1 (a), N:1 (b) i M:N (c)

Najjednostavniji odnos između dva objekta naziva se preslikavanje 1:1. Kod takvog preslikavanja svaki se element skupa X može preslikati na najviše jedan element skupa Y. Istovremeno, i svaki element skupa Y može biti preslikan na najviše jedan element skupa X. Karakterističan primer bi bio sa entitetima Fakultet i Dekan. Na jednom fakultetu može biti samo jedan dekan, a jedan dekan može biti dekan na samo jednom fakultetu. Takvi odnosi između entiteta su retki, a mogu se predstaviti slikom 7.9a.

Druga vrsta odnosa naziva se preslikavanje N:1 (ili 1:N). Više elementa skupa X može se preslikati na najviše jedan element skupa Y. Istovremeno jedan element skupa Y može se preslikati na više elemenata skupa X. Pogodan primer za ovu vrstu odnosa između entiteta je odnos između entiteta Student i Dekan. Više studenata na jednom fakultetu ima samo jednog dekana, a jedan dekan je dekan za više studenata na svom fakultetu.

Najsloženije preslikavanje je tipa M:N. Svaki element prvog skupa može se preslikati na više elemenata drugog skupa, ali se i svaki element drugog skupa može preslikati na više elemenata prvog skupa. Karakterističan primer ovakvih veza postoji ako se uoče entiteti Student i Profesor. Jednom studentu predaje više profesora, a ujedno jedan profesor predaje za više studenata.

7.7. RELACIONI MODEL BP

Kao i mnoge druge tehnologije u računarskoj industriji, koreni relacionih BP potiču iz IBM-a i njihovog istraživanja automatizovanja kancelarijskih operacija u 60-tim i 70-tim godinama XX veka. U srcu relacionog modela nalazi se koncept tabele (koja se naziva relacija) u kojoj su smešteni svi podaci. Svaka tabela je načinjena od slogova (redova u tabeli), a svaki slog ima svoja polja (atribute). Osnovne karakteristike relacionog modela podataka su sledeće:

- Sve se predstavlja relacijama (tabelama);
- Minimalna redundansa podataka;
- Jednostavno ažuriranje podataka;
- Izbegnute su anomalije ažuriranja;
- Redosled kolona i redova ne utiče na informacioni sadržaj tabele;
- Svaki red se može jednoznačno odrediti (postoji primarni ključ);

U relacionom modelu podataka klase objekata se predstavljaju tabelama. Na primer klasa Student se može opisati atributima Ime, Prezime, Broj indeksa, Mesto i Telefon. Klasa Predmet sa atributima Naziv predmeta, Oblast i Šifra predmeta. Neki od studenata su polagali neke predmete i dobili ocene na ispitu, što je prikazano u veznoj tabeli Ispit, koja se može opisati atributima Broj indeksa, Ocena, Šifra predmeta i Datum. Trenutno stanje studenata, predmeta i ocena sa ispita je uneseno u tabele i prikazano na slici 7.10.

STUDENT

Ime	Prezime	BrInd	Mesto	Telefon
Marko	Marković	123456/2016	Beograd	063-123123
Petar	Petrović	222333/2016	Valjevo	065-232323
Janko	Janković	111222/2015	Niš	062-121212

ISPIT

BrInd	Ocena	Šif_Predmeta	Datum
123456/2016	7	I7	21.01.2017.
123456/2016	8	UP15	27.01.2018.
222333/2016	10	I7	04.02.2017.

PREDMET

Naziv	Oblast	Šif_Predmeta
Baze podataka	Informatika	I7
Upravljanje projektima	Menadžment	UP15
Računovodstvo	Ekonomija	E2

Slika 7.10. Relacioni model podataka

Suština relacionog modela je da se i klase objekata i klase veza između objekata predstavljaju na jedinstven način, tj. preko tabela. U našem primeru postoje tri tabele: STUDENT, PREDMET i ISPIT. U relacionom modelu podataka tabela se definiše kao relacija, koja mora da ispuni odgovarajuće uslove. Svaka relacija mora da ima primarni ključ – jedan ili više atributa koji na jedinstven način opisuju svaki zapis u jednoj tabeli. Primarni ključ se pažljivo bira. Na primer u klasi studenata loš izbor primarnog ključa bi bio atribut Ime, zato što se mogu pojaviti dva studenta sa istim imenom. Dobar izbor primarnog ključa je atribut Broj indeksa, zato što ne postoje dva studenta sa istim

brojem indeksa. Za klase objekata Student i Predmet vrši se prevođenje u relacioni model na sledeći način (podvlačenjem su označeni atributi koji čine primarni ključ):

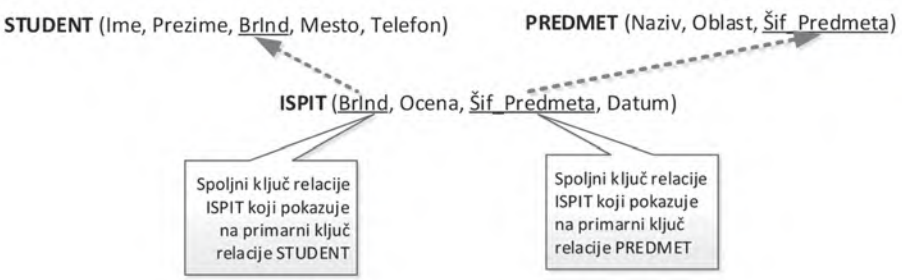
STUDENT (Ime, Prezime, BrInd, Mesto, Telefon),

PREDMET (Naziv, Oblast, Šif_Predmeta)

Za klasu veza ISPIT, može se definisati prirodan primarni ključ u odnosu na objekte koje povezuje. U našem primeru relacija ISPIT bi glasila:

ISPIT(BrInd, Ocena, Šif_Predmeta, Datum)

Dakle, za posmatrani realan slučaj gde su studenti polagali neke predmete i dobili ocene, izvršeno je modelovanje preko tri tabele tj. relacije. Sve tabele su povezane. Povezivanje se vrši preko vrednosti atributa u relacijama. na sledeći način:



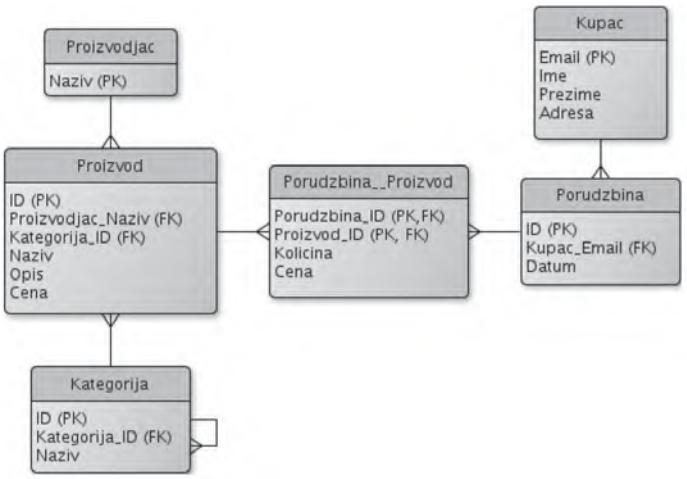
Slika 7.11. Relacije se povezuju vrednostima stranih i primarnih ključeva

Svaka relacija (tabela) se identifikuje jedinstvenim imenom. Redosled atributa u okviru jedne relacije nije od značaja. Nazivi atributa u okviru jedne relacije moraju da budu unikatni. Nema potrebe da se vodi računa o tome kako su podaci smešteni na disku. Ovo je različito od hijerarhijskog i mrežnog modela u kojima korisnik mora da razume kako su podaci struktuirani unutar BP da bi mogao da ih pretražuje, unosi nove, ažurira ili briše postojeće slogove.

Snaga relacionog modela podataka dolazi do izražaja kod održavanja, tj. ažuriranja podataka u relacijama. Relacije STUDENT i PREDMET su nezavisne relacije i zapisi u takve relacije mogu da se unose nezavisno. DBMS (sistem za upravljanje bazama podataka) vodi računa da u jednu relaciju nije moguće uneti dva identična zapisa (reda). Ova osobina se naziva identifikacioni integritet. Relacija ISPIT je zavisna, što znači da se u nju mogu uneti zapisi samo ukoliko vrednost atributa BrInd odgovara nekoj vrednosti atributa BrInd u relaciji STUDENT. Pored toga, vrednost atributa Šif_Predmeta mora da odgovara bar jednoj vrednosti u relaciji PREDMET. Ova osobina se naziva referencijalni integritet i o njemu takođe brine DBMS. Na kraju, nakon unosa podataka, u nezavisnim relacijama STUDENT i PREDMET nije moguće brisanje zapisa ukoliko jedna od vrednosti atributa BrInd i Šif_Predmeta odgovara bar jednoj od vrednosti tih atributa u relaciji ISPIT, zato što bi se brisanjem narušio tzv. referencijalni integritet. Navedena pravila se mogu ublažiti ukoliko se kod operacija ažuriranja navedu drugačije specifikacije refrencijalnog integriteta.

7.7.1. Model objekti veze

Projektovanje baze podataka u suštini znači da treba doći do kompleta naredbi kojima se definiše šema baze podataka - tabele, relevantni atributi, domeni, ograničenja, itd. U osnovi projektovanja je utvrđivanje preciznog modela organizacije za koju se radi informacioni sistem. Kao i u ostalim inženjerskim disciplinama, složenost ovakvog posla zahteva da proces kreiranja bude izveden dobro definisanom metodologijom i da bude testiran u skladu sa objektivnim kriterijumima. Jedan od najvećih problema u procesu razvoja BP je činjenica da projektanti, programeri i krajnji korisnici na potpuno različite načine shvataju podatke i načine njihove upotrebe, kao i procese iz posmatranog okruženja koje treba modelirati. Da bi se obezbedio precizan opis prirode podataka i načina na koji se oni koriste postoji jasan model koji nije striktno tehničke prirode. Najčešće korišćeni model u praksi je model objekti-veze (MOV).



Slika 7.12. Primer MOV dijagrama baze podataka za potrebe elektronske (web) prodavnice

Kreiranje baza podataka je praktično proces u dva koraka. Početna faza je bazirana na MOV modelu, a druga faza je implementacija. Rezultat MOV faze se redefiniše primenom teorije normalizacije posle koje se garantuje kvalitet šema relacija u skladu sa odgovarajućim kriterijumima. Tipičan dizajn baze podataka obuhvata definisanje skupova relacija, na stotine atributa i mnogo ograničenja koja proističu iz ograničenja u realnom svetu. Dizajniranje baze podataka zahteva dobar nivo kreativnosti, iskustva, tehničke ekspertize i razumevanje osnovnih pravila. MOV modelovanje obuhvata:

- Skup entiteta (objekti i veze), i njihovih ograničenja
- Definisanje ključeva i atributa
- Grafička predstava globalne šeme (ER dijagram)
- Svođenje globalne šeme na tabele (relacije)

7.8. SQL – JEZIK RELACIONE BAZE PODATAKA

SQL (Structured Query Language) je standardizovani programski jezik za rad sa relacionim bazama podataka. Ugrađen je u sve vodeće sisteme za upravljanje bazama podataka (Ingres, DB2, Oracle, SQL Server, MySQL, PostgreSQL i sl.). Koriste ga programeri aplikacija i administratori baza podataka. Postoji nekoliko ANSI[1] standarda za SQL, od 1986. godine do danas, a SQL se i dalje razvija. Standardima se postiže da programi koji sadrže SQL ne zastarevaju i prenosivi su na druge računarske platforme.

SQL je neproceduralan, deklarativni jezik visokog nivoa, koji ne govori kako će nešto biti urađeno već šta treba da se uradi. Svaki programski jezik obuhvata podatke i manipulaciju nad tim podacima. Kod SQL-a objekti manipulacije su relacije, a rezultat manipulacija su opet relacije. SQL podržava tri osnovne funkcije nad objektima baze podataka: definicije, manipulacije i kontrole. Naredbe za definisanje tj. kreiranje (CREATE) se koriste za kreiranje relacija, pogleda, indeksa itd. Manipulacija nad bazom podataka podrazumeva funkcije ažuriranja: unos (INSERT), brisanje (DELETE) i izmenu unetih podataka (UPDATE). Osnovna naredba kod SQL-a je naredba SELECT sa svojim klauzulama i logičkim operatorima, kojom se postavljaju upiti nad bazom podataka. Iako je SQL originalno namenjen za krajnje korisnike, mnogo češće se SQL upiti ugrađuju u softver koji omogućava lakši korisnički interfejs. Kao odgovor na upit, BP vraća skup podataka, koji je u stvari lista redova koji sadrže odgovor. Najjednostavniji upit je da se dobiju svi redovi iz tabele, ali češće, redovi se filtriraju na neki način da bi se dobio traženi odgovor. Često se podaci iz više tabela kombinuju u jednu, procesom udruživanja. Nad objektima baze podataka naredbama za kontrole mogu se dodeliti privilegije pojedinim korisnicima nad željenim objektima (GRANT). Jednom dodeljene privilegije se mogu oduzimati (REVOKE).

7.9. OPORAVAK BAZE PODATAKA

Oporavak BP (RECOVERY) predstavlja proces vraćanja BP u korektno stanje. Sasvim je realno, i dešava se, da usled otkaza sistema mora da se uradi oporavak BP. Uzroci otkaza mogu biti različiti: hardverske greške, greške u programiranju, greške u operativnom sistemu, mrežni problemi, nestanak napajanja i sl. Oporavak zavisi od tipa greške koja je nastala. Ako ni jedan podatak nije narušen ili izgubljen oporavak se često svodi na ponovnu inicijalizaciju sistema (reset). Ako su podaci izgubljeni, oporavak zahteva dodatne korake.

Proces oporavka u slučaju gubitka podataka se zasniva na redudansi podataka, tj. postojanje rezervnih kopija. U slučaju otkaza sistema, oštećena BP se rekonstruiše u ispravno stanje na osnovu poslednje kopije i na osnovu dnevnika transakcija, a nekonzistentno

1 American National Standards Institute, organizacija koja nadgleda razvoj standarda u SAD. To su najčešće korišćeni standardi, a neki od njih su usvojeni od strane ISO, kao međunarodni

stanje se rešava tako što se poništavaju nekonzistentne promene, a transakcije se ponavljaju. Trajnost podrazumeva da nijedna promena u bazi podataka koju je napravila započeta transakcija, ne sme biti izgubljena. Iz tog razloga, a pošto hard disk može da otkaže, BP se mora čuvati na različitim hard diskovima. Jednostavan primer postizanja trajnosti jeste čuvanje različitih kopija BP na različitim diskovima. Pošto je istovremeni pad oba diska malo verovatan, velika je verovatnoća da će barem jedna kopija hard diska biti uvek dostupna.

Jedini način da se obezbedi brzi oporavak baze podataka kod otkaza sistema jeste pažljivo napravljen plan oporavka sa detaljnim procedurama. To podrazumeva i sve aktivnosti koje preventivno mogu da utiču na smanjenje mogućnosti narušavanja i gubitka podataka u bazi.

7.10. KLJUČNI TERMINI

Atribut – Kolona u tabeli, koja opisuje entitet. Može biti tipa Integer, Double, Date, itd.

Primarni ključ – je jedan ili više atributa čija vrednost jednoznačno određuje primerak entiteta

Baza podataka – Skupo logično povezanih tabela

Rekord – jedan red u tabeli koji označuje jedan primerak entiteta

Entitet – objekat u sistemu koji želimo da modelujemo i skladištimo

Model – logična struktura baze podataka koja određuje konkretan način na koji se podaci skladište, organizuju i manipulišu

Structured Query Language – (SQL) jezik koji se koristi za kreiranje i manipulisanje relacionih baza podataka

Pitanja za ponavljanje

1. Šta se baza podataka?
2. Šta je primarni ključ?
3. Šta je SQL?
4. Šta je DBMS?
5. Šta je model baze podataka?
6. U kojim situacijama se koristi SELECT komanda SQL-a?

Literatura

[1] M. Veinović, M. Šimic, A. Jevremović i I. Franc, Baze Podataka, Beograd: Univerzitet Singidunum, 2016.

[2] M. Milosavljević, M. Veinović i G. Grubor, Informatika, Beograd: Univerzitet Singidunum, 2017.

[3] G. Beekman i B. Beekman, Digital Planet: Tomorrow's Technology and You, Upper Saddle River: Prentice Hall, 2012.

[4] A. Silberschatz, H. Korth i S. Sudarshan, Database Systems Concepts, New York: Mc Graw Hill, 2011.

[5] H. Garcia-Molina, J. Ulman i J. Widom, Database Systems - The Complete Book, Upper Saddle River: Prentice Hall, 2008.

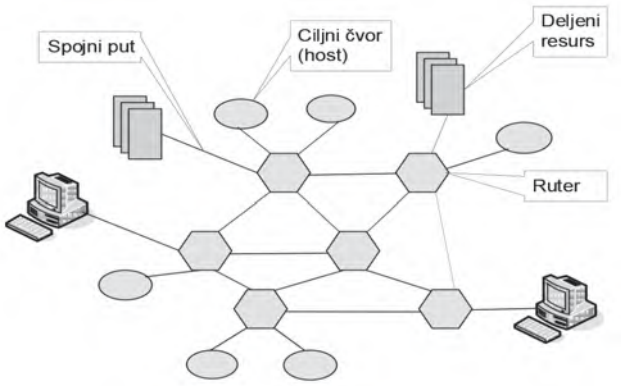
[6] B. Lazarević, Z. Marjanović, N. Ančić i S. Babrogić, Baze Podataka, Beograd: FON, 2008.

8. RAČUNARSKE MREŽE

Potreba za informacijama naterala je čoveka da uspostavlja veze sa raznim izvorima informacija i da stvara mreže preko kojih će sebi olakšati prikupljanje, prenos, skladištenje i obradu podataka. Stalnim razvojem računarskih tehnologija (povećanje performansi uz pad cena) i sa razvojem Interneta, broj korisnika računara i računarskih mreža raste velikom brzinom. Sa sve moćnijom računarskom opremom svakodnevno se uvode novi servisi. Vremenom su se mrežni sistemi razvijali da bi danas dostigli nivo praktičnog efikasnog okruženja za razmenu podataka. U ovoj lekciji razmatraju se koncepti i standardi umrežavanja neophodni za razumevanje računarskih mreža.

8.1. UVOD

Počeci umrežavanja vezuju se za prve telegrafske i telefonske linije kojima su se prenosile informacije do udaljenih lokacija. Dostupnost i fleksibilnost tehnologija današnjih savremenih računarskih mreža omogućava efikasno povezivanje na Internet sa svih lokacija, što uključuje i pokretne korisnike. Cena eksploatisanja današnjih mreža je sve niža. Računarske mreže su postale nezamenjivi deo poslovne infrastrukture kako malih, tako i velikih organizacija. Poznavanje tehnologije i korišćenje mreža nije samo stvar opšte kulture.



Slika 8.1. Osnovna arhitektura mreže

Računarska mreža u svojoj osnovi je skup dva ili više računara, koji su povezani medijumom za povezivanje i koji međusobno mogu da komuniciraju i dele resurse. Koristi se za prenos kako digitalnih tako i analognih podataka, koji moraju biti prilagođeni

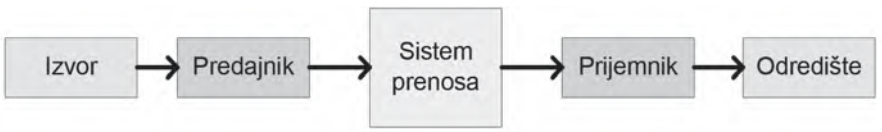
medijumima za prenos. Mrežom se prenose računarski podaci, govor, slika, video, a aplikacije na stranama korisnika mogu biti takve da se zahteva prenos podataka u realnom vremenu (govor, video i sl.) ili to ne mora biti uslov (elektronska pošta, prenos datoteka i sl.). Mreža se sastoji od računara, medijuma za prenos (žica, optičko vlakno, vazduh) i uređaja kao što su čvorišta, komutatori, ruteri itd. koji čine infrastrukturu mreže. Neki od uređaja, kao što su mrežne kartice, služe kao veza između računara i mreže.

Svaka mreža se može svesti na sledeće dve osnovne celine: hardversku i softversku. Hardversku celinu sačinjavaju mrežni čvorovi (nods) u kojima se vrši obrada informacija, fizičkih spojnih puteva i deljenih resursa. Čvorovi su delovi mreža u kojima dolazi do obrade podataka. Postoje dve vrste čvorova: čvorovi u kojima se vrši stvarna obrada i oni predstavljaju ciljne čvorove (hosts), i čvorovi kojima je uloga da usmeravaju informacije (routers). Deljeni resursi su hardverski (štampači, ploteri, diskovi i sl.) ili softverski elementi (datoteke, baze, aplikacije i sl.). Softversku celinu mreže čine protokoli – pravila po kojima se vrši komuniciranje (razmena podataka) u mreži i operativni sistemi koji su u direktnoj komunikaciji sa hardverom računara i imaju podršku za mrežni hardver i mrežne protokole.

8.2. KOMUNIKACIONI SISTEM

Računarska mreža se može posmatrati kao komunikacioni sistem, gde se informacija generisana na predajnoj strani (izvorište poruke) dostavlja željenom odredištu. Osnovni elementi komunikacionog sistema su:

- Izvor (source) – generisanje podataka za prenos,
- Predajnik (transmitter) – prevođenje generisanih podataka u oblik pogodan za prenos (npr. mrežna kartica),
- Prenosni sistem (tramission sistem) – može biti jednostavna linija ili kompleksna mreža koja spaja izvor i odredište (žično, bežično),
- Prijemnik (receiver) – prihvata signal iz prenosnog sistema i transformiše ga u oblik pogodan za prijem,
- Odredište (destination) – prihvata prenete podatke.



Slika 8.2. Model komunikacionog sistema

Ključni poslovi u komunikacionom sistemu su:

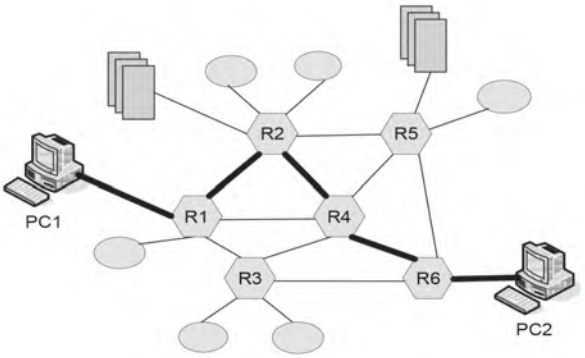
- Povezivanje (interfacing) uređaja na komunikacioni sistem
- Generisanje signala (signal generation) – propagacija, regeneracija, domet itd.
- Sinhronizacija (synchronization) predajnika i prijemnika
- Razmena podataka (exchange management) – prema odgovarajućem protokolu
- Otkrivanje i ispravljanje grešaka (error detection and correction) npr. kod slanja datoteka
- Kontrola toka (flow control) usaglašavanje brzine slanja i brzine prijema podataka
- Adresovanje i usmeravanje (addressing and routing) – čim postoje više od dva učesnika
- Oporavak (recovery) – mogućnost da se transfer podataka nastavi od mesta prekida
- Formatiranje podataka (message formatting) dogovor učesnika
- Zaštita (security) na prenosnom putu, autentifikacija učesnika i integritet podataka
- Upravljanje mrežom (network management) – mreža je kompleksan sistem, koji ne radi sam po sebi. Neophodno je mrežu konfigurisati, monitorisati, intervenisati i inteligentno planirati za buduću namenu.

8.3. VRSTE PRENOSA PODATAKA

U računarskim mrežama postoje dva dijametralno suprotna načina prenosa podataka. Kod prvog načina, koji je stariji, veza između izvorišta poruke i odredišta uspostavlja se kroz čvorove mreže, na način da se zauzima kompletan spojni put. Karakterističan primer je javna telefonska komutirana mreža (PSTN). Drugi tip je paketski način prenosa, gde se poruka deli u manje celine – pakete (okvire), a kroz mrežu se paketi mogu preusmeravati po različitim spojnim putevima. Ovakav način prenosa je karakterističan kod Interneta.

8.3.1. Prenos podataka sa komutacijom veza (*Circuit Switched*)

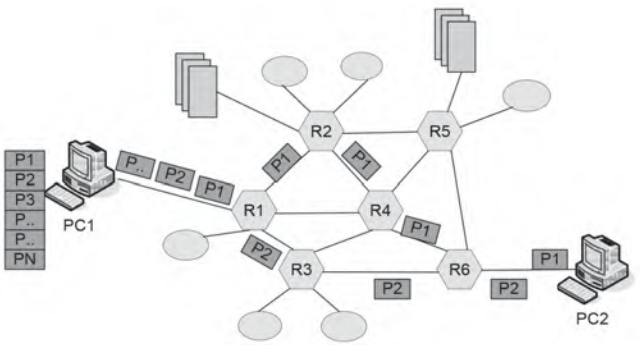
U ovom tipu prenosa podataka između dva učesnika u komunikaciji uspostavlja se čvrsta direktna veza, a ukupna informacija se prenosi putanjom koja je utvrđena u toku uspostave veze. Na primer, ako računar PC1 želi da komunicira sa računarom PC2 prvo se uspostavlja veza između ova dva računara i ta veza postoji samo za dati prenos podataka. Ako neki treći računar poželi da komunicira sa računarom PC2 u tom trenutku, to neće biti moguće po istom spojnom putu. Osnovna karakteristika ovakvog načina prenosa podataka je da se podaci mogu prenositi uspostavljenom vezom maksimalnom brzinom koja je moguća, tj. u potpunosti se može koristiti kompletan frekvencijski opseg uspostavljenog spojnog puta (komunikacionog kanala) za prenos podataka.



Slika 8.3. Prenos podataka sa komutacijom veza

8.3.2. Prenos podataka sa komutacijom paketa (*Packett Switched*)

Kod ovog načina prenosa podataka između dva učesnika, informacija koja se razmenjuje se deli u pakete, odgovarajuće strukture (dužina paketa, redni broj, adresa odredišta, prioritet i sl.). Paketi se upućuju do prvog čvora u mreži (rutera), a u svakom ruteru se vrši nezavisno usmeravanje paketa. Izbor putanje u ruterima se vrši na osnovu više kriterijuma koji važe u datom trenutku. Paketi prolaze različite putanje od izvorišta do odredišta. Na odredištu se vrši slaganje paketa u prvobitan redosled da bi se dobila potpuna informacija. Ovakav način prenosa podataka je karakterističan za računarske mreže gde većinu mrežnog saobraćaja čine kratki naleti podataka, nakon kojih u dužem periodu nema prenosa. U ovim tzv. "prazninama" mogu se slati paketi koje šalje neki treći učesnik. Dakle, podaci od različitih izvorišta mogu prolaziti istim spojnim putem. Ovo je daleko žilaviji način prenosa, zato što paketi najčešće mogu da nađu bar jedan slobodan spojni put. Mana je što je efektivna brzina slanja podataka na ovaj način manja od maksimalne koju dozvoljava propusni opseg kanala, zato što ga koriste više učesnika u komunikaciji.



Slika 8.4. Prenos podataka sa komutacijom paketa

Danas kada su računari dostupni svakom i uz to su izuzetno moćni, umrežavanje povećava efikasnost i smanjuje troškove poslovanja. Mreže omogućavaju da se deli računarski hardver i da se time bolje iskoristi računarska oprema (smanjenje troškova, bolje performanse opreme i sl.). Sa stanovišta softvera, mreže omogućavaju da se dele podaci i da softverske aplikacije sarađuju što dovodi do povećanja efikasnosti i produktivnosti. Mreže omogućavaju ljudima da rade zajedno, da komuniciraju, koriste zajedničke resurse i da se razviju usluge bez kojih se ne može zamisliti današnji život.

Mreže se mogu upotrebiti i za zajedničko i standardizovano korišćenje aplikacija, kao što su programi za obradu teksta, programi za tabelarne proračune ili baze podataka, u situacijama kada je bitno da svi koriste iste aplikacije i iste verzije tih aplikacija. Dalje, jednostavnije je i bolje da ljudi potpuno savladaju jedan program, nego da moraju da rade sa četiri ili pet različitih programa. Kada su računari umreženi, to značajno pojednostavljuje i njihovu podršku. Za jednu kompaniju je daleko efikasnije kada tehničko osoblje održava jedan operativni sistem i kada su svi računari identično podešeni prema konkretnim potrebama te kompanije.

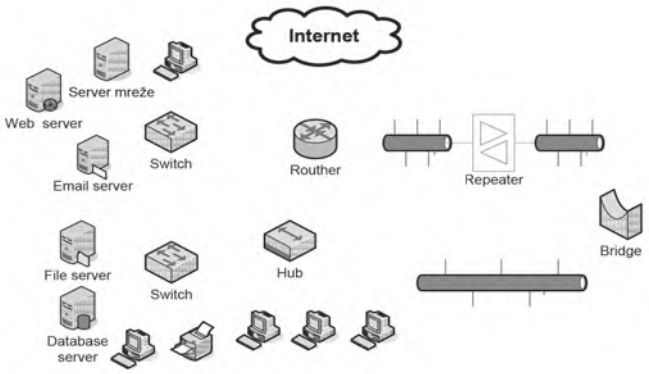
8.4. VRSTE RAČUNARSKIH MREŽA PREMA PROSTORU KOJI OBUHVATAJU

Prema prostoru koji obuhvataju, računarske mreže se mogu podeliti na

- lokalne (LAN)
- regionalne računarske mreže (WAN) – mreže šireg područja

8.4.1. Lokalna računarska mreža (*Local Area Network, LAN*)

Predstavlja osnovu svake mreže. Ona može biti jednostavna kada imamo dva računara koja komuniciraju ili složena kada su povezani na stotine računara i periferijskih uređaja u jednoj velikoj organizaciji. Osnovno obeležje lokalne računarske mreže je to što je ona prostorno ograničena. Računari su blizu jedni drugima, najčešće u istoj zgradi. Svaki računar ili periferija su nezavisni čvorovi na mreži, a spojeni su sa komunikacionim čvorovima (habovi, svičevi, …) koji omogućavaju da bilo koji čvor na mreži može da komunicira sa nekim drugim čvorom.



Slika 8.5. Lokalna računarska mreža (LAN)

Najpoznatiji način povezivanja čvorova mreže je preko eternet kabla koji je povezan na eternet port na računaru ili uređaju (periferiji). Eternet je popularna arhitektura za umrežavanje koja je danas postala industrijski standard. Svaki novi PC računar sadrži eternet port na matičnoj ploči. Preko eternet porta se kontroliše tok podataka između memorije računara i mrežnog kabla i vrši se fizičko prilagođavanje signala iz računara sa mrežom.

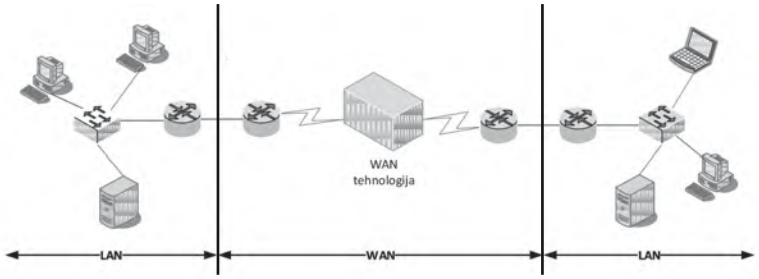
Slika 8.6. LAN mreže omogućavaju žično i bežično povezivanje uređaja i periferija

Za povezivanje čvorova u LAN mrežama danas je najveći trend u primeni bežičnih tehnologija. U ovom slučaju svaki čvor sadrži primo-predajnik koji omogućava prenos podataka putem radio signala, kroz vazduh, a ne kroz kabl. Bežični prenos je pogodan za manja rastojanja, u kućama i manjim preduzećima. Omogućava povezivanja bez kopanja zidova i provlačenja kablova, a posebno je pogodan za povezivanje na mrežu povremeno i u pokretu. Brzine prenosa podataka u bežičnim mrežama su po pravilu sporije od žičnih, ali je bežično povezivanje daleko fleksibilnije. LAN mreže omogućavaju kombinovano – žično i bežično povezivanje.

Računari koji se koriste u LAN mreži ne moraju da imaju isti operativni sistem. Na primer u jednoj LAN mreži se mogu naći računari sa MAC ili Windows operativnim sistemom ili serveri pod Linux-om. Postavljanje LAN mreže u kućnim uslovima je jednostavno i pravolinijsko. Sem povezivanja opreme, gde je industrijskim standardima obezbeđeno da se ne može učiniti greška, zahteva se minimalno znanje za instalaciju. Za veće mreže, zahteva se znanje mrežnog administratora za konfiguraciju i obezbeđivanje funkcionalnosti mreže i kako bi svi ostali koristili samo usluge date mreže.

8.4.2. Regionalna računarska mreža (*Wide Area Network, WAN)*

Nije prostorno ograničena. Ona može da poveže računare i uređaje širom sveta. Najveća i najpoznatija WAN mreža je Internet. WAN mrežu čini veliki broj povezanih lokalnih mreža. Za povezivanje se koriste usluge telekomunikacionih operatera. Neke od tehnologija za povezivanje LAN-ova su: E1(T1), E3(T3), ATM, ISDN, ADSL, Frame Relay, radio veze i slično. Ove tehnologije se nazivaju i okosnice ili kičma-mreže (backbone).



Slika 8.7. Regionalna računarska mreža (WAN)

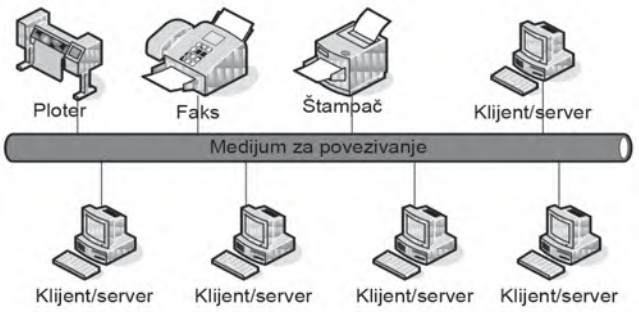
8.5. PRISTUP UMREŽAVANJU

8.5.1. Mreže ravnopravnih računara

Kod mreža ravnopravnih računara (peer-to-peer mreže) ne postoje namenski serveri niti hijerarhija računara. Svi računari su jednaki, odnosno ravnopravni. Nude jednostavan pristup povezivanju računara radi zajedničkog korišćenja resursa i međusobne komunikacije. Svaki računar funkcioniše i kao klijent i kao server, pa ne postoji administrator koji bi bio odgovoran za celu mrežu. Korisnik svakog računara sam određuje koji se podaci sa njegovog računara mogu deliti preko mreže.

Mreže ravnopravnih računara se često nazivaju i radne grupe. Ovakvu mrežu najčešće čini do 10 računara. Takve mreže su relativno jednostavne. U situaciji kada svaki računar funkcioniše i kao klijent i kao server, ne postoji potreba za moćnim centralnim

serverom, ili drugim karakterističnim komponentama mreža velikog kapaciteta. Stoga su ove mreže jeftinije od serverskih mreža. Bezbednost, sprečavanje neovlašćenog pristupa računarima i podacima, podrazumeva definisanje lozinke za resurs. U mreži ravnopravnih računara, svaki korisnik sam podešava sopstvenu bezbednost, pa je zato teško sprovesti centralnu kontrolu. Ovaj nedostatak kontrole ima značajne posledice na bezbednost mreže, jer pojedini korisnici mogu da ne primenjuju mere bezbednosti. Stoga, ukoliko je bezbednost bitan faktor, bolje rešenje predstavlja serverska mreža.



Slika 8.8. U mreži ravnopravnih korisnika računari funkcionišu i kao klijenti i kao serveri

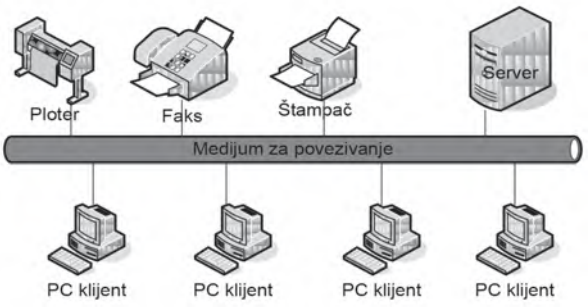
8.5.2. Serverske mreže

U mreži sa više korisnika, mreža ravnopravnih računara u kojoj se računari ponašaju i kao klijenti i kao serveri, nije pravo rešenje. U takvim situacijama postoje namenski serveri. Namenski server je računar čija je jedina uloga opsluživanje mreže i ne koristi se kao klijent ili radna stanica. Za servere se kaže da su „namenski" zato što oni sami ne mogu biti klijenti, već su optimizovani da brzo opsluže zahteve mrežnih klijenata i osiguraju bezbednost datoteka i direktorijuma.

Kako se mreža povećava sa brojem računara, njihove međusobne udaljenosti i saobraćaja između njih, nastaje potreba za većim brojem servera. Podela poslova na nekoliko servera obezbeđuje da se svi poslovi obavljaju na najefikasniji mogući način. Raznovrsnost i složenost poslova koje serveri treba da obave je velika. Mnoge velike mreže imaju različite vrste servera.

Iako su instaliranje, konfigurisanje i upravljanje kod serverskih mreža znatno složeniji nego kod mreža ravnopravnih korisnika, one imaju brojne prednosti. Server je napravljen tako da omogući pristup brojnim periferisjkim uređajima, datotekama, bazama podataka, uz odgovarajuće performanse i bezbednost. Kod serverskih mreža je moguće administriranje i kontrolisanje zajedničkog korišćenja resursa iz jednog centra. Ovako se resursi lakše pronalaze i čine dostupnijim nego kod mreža ravnopravnih korisnika. Bezbednost je najčešće osnovni razlog opredeljivanja za serversku mrežu. U ovakvom okruženju jedan administrator može da definiše bezbednost i to, onda, važi za svakog

korisnika mreže. U zavisnosti od važnosti podataka, moguće je praviti rezervne kopije više puta dnevno ili nedeljno. Kako su najbitniji podaci centralizovani na jednom ili nekoliko servera, ovaj proces je vrlo jednostavan.



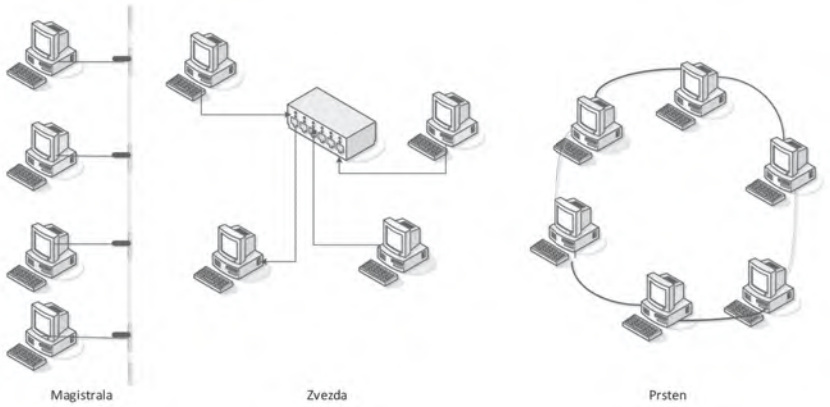
Slika 8.9. Serverska mreža

Serverske mreže mogu imati hiljade korisnika. Takvom mrežom se ne bi moglo upravljati kada bi se primenio princip ravnopravnih korisnika, ali savremeni alati za nadgledanje i upravljanje mrežama omogućavaju da serverska mreža normalno funkcioniše i sa ogromnim brojem korisnika.

8.6. STANDARDNE TOPOLOGIJE

LAN topologija definiše način povezivanja računara i ostalih uređaja u mreži. Lokalna mrežna topologija može se opisati sa stanovišta fizičke ili logičke perspektive. Fizička topologija opisuje geometrijsko uređenje komponenti koje sadrži LAN, odnosno proučava i objašnjava kako se mrežne komponente stvarno, fizički povezuju odgovarajućim medijumom. Logička topologija opisuje mogućnost veze između dve krajnje tačke mreže koje komuniciraju, odnosno objašnjava kako informacija putuje kroz mrežu.

Postoje tri glavne LAN topologije: magistrala (bus), prsten (ring) i zvezda (star). Izbor i specifikacija topologije LAN mreže zavisi od: fizičkih lokacija na kojima se nalaze korisnici sistema, količine podataka u lokalnim bazama podataka, od frekvancije pristupa bazama na drugim lokacijama i zahteva za komuniciranjem između dve korisničke lokacije.



Slika 8.10. Osnovne topologije mreža

Topologija magistrale (bus) je linearna LAN arhitektura, kod koje se prenos podataka obavlja celom dužinom fizičkog mediuma. Kod ove toplogije lako je dodati novi mrežni uređaj i zahteva daleko manje kabla nego ostale topologije. Mane je što cela mreža može biti u prekidu ako negde postoji prekid na glavnom kablu. Eventualni problemi se teško otkrivaju.

Ring topologija ili topologija prstena predstavlja način na koji su uređaji međusobno logički povezani. Ovakva vrsta mreže se sastoji od više uređaja povezanih jedan sa drugim tako da se obrazuje zatvorena kružna putanja. IBM mreže Token Ring/IEEE 802.5 koriste implementaciju ring topologije.

Topologija zvezde predstavlja takav oblik arhitekture gde su krajnji čvorovi na mreži povezani preko posebne veze na centralni hub ili svič. Logičke bus ili ring topologije su često fizički implementirane kao star topologije. Prednosti ove topologije: lako se instalira i povezuje, nema prekida u mreži pri dodavanju novog uređaja ili uklanjanja, lako je otkriti greške i zameniti delove i sl. Mane ovo topologije: podložna je zagušenjima saobraćaja, zahteva više kabla nego linearna topologija, ako se hub ili switch pokvari svi čvorovi su ugašeni, mnogo je skuplja topologija od npr. topologije magistrale.

8.7. PASIVNA MREŽNA OPREMA

Pasivna mrežna oprema predstavlja najjednostavniju komponentu računarskih mreža. Atribut “pasivna” potiče od ciljne karakteristike komponenti ove kategorije da nad mrežnim saobraćajem ne izvrše nikakvu izmenu. Pasivne komponente mreže čine:

- utičnice
- kablovi
- paneli za prespajanje i za završavanje kablova (patch panel)

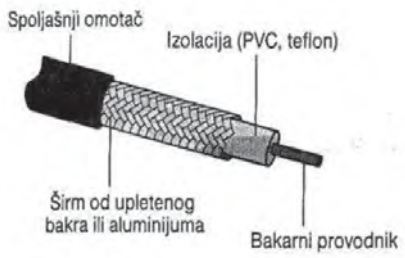
- kablovi za prespajanje (patch cabel)
- rek ormani
- kanalice za vođenje kabla
- itd.

Za prenos signala između računara većina današnjih mreža koristi kablove koji se ponašaju kao mrežni prenosni medijumi. Postoji mnogo različitih tipova kablova koji mogu da se primene u različitim situacijama. Većina današnjih mreža koristi tri osnovne vrste kablova: koaksijalne kablove, kablove sa upredenim paricama (twisted pair) ioptičke kablove. Kroz koaksijalni kabl i upredene parice prenose se električni signali, dok se kroz optička vlakna prenose signali u vidu svetlosnih impulsa. Izbor kabla za konkretnu mrežu zavisi od više parametara, kao što su: brzina prenosa, pouzdanost, maksimalno rastojanje između čvorova, električne smetnje, podužno slabljenje, tolerancija u otežanim uslovima rada, cene, lakoće povezivanja, troškova održavanja itd.

8.7.1. Koaksijalni kabl

U jednom trenutku ovo su bili najrasprostranjeniji mrežni kablovi, i to iz više razloga: relativno su jeftini, laki, fleksibilni i jednostavni za rad. U svom najjednostavnijem obliku, koaksijalni kabl se sastoji od bakarne žice u sredini, oko koje se nalazi najpre izolacija, a zatim sloj od upletenog metala – širm, i na kraju spoljašnji zaštitni omotač .Svrha ovog oklopa je da apsorbuje elektromagnetne smetnje ili šum, i time spreči njihovo mešanje sa podacima koji se prenose. Postoje koaksijalni kablovi sa višestrukom zaštitom (dva sloja izolacije i dva sloja širma), koji se primenjuju u sredinama sa jakim elektromagnetnim smetnjama.

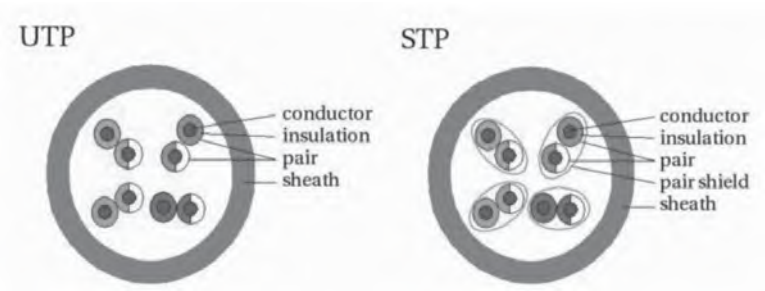
Bakarni provodnik (žica) u sredini kabla prenosi elektromagnetne signale koji predstavljaju kodirane računarske podatke. Ovaj provodnik može biti od punog metala, ili u obliku više upletenih žica. Ukoliko je od punog metala, onda je to obično bakar. Provodnik je obložen dielektričnim izolacionim slojem koji ga odvaja od širma.



Slika 8.11. Slojevi koaksijalnog kabla

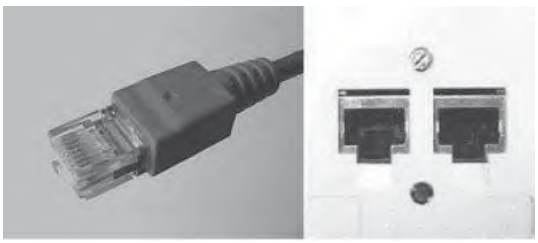
8.7.2. Kabl sa upredenim paricama (*Twisted Pair*)

Kabl sa upredenim paricama se sastoji od dve izolovane bakarne žice koje su obmotane (upredene) jedna oko druge. U osnovi, razlikujemo kabl sa neoklopljenim (Unshielded Twisted-Pair, UTP) i oklopljenim paricama (Shielded Twisted-Pair, STP). Grupe parica se obično nalaze grupisane u zaštitnom omotaču i zajedno sa njim čine kabl. Upredanjem se poništava električni šum od susednih parica, ili ostalih izvora, kao što su motori, releji, transformatori i energetska instalacija.



Slika 8.12. Presek: kablovi sa neoklopljenim i oklopljenim paricama

S obzirom da je problem elektromagnetne zaštite veoma ozbiljan, neki proizvođači su razvili tzv. oklopljene kablove, koji oko parica imaju određenu električno provodnu strukturu koja pruža znatno veći nivo zaštite. Kablovi sa upredenim paricama za povezivanje sa računarima koriste RJ-45 konektore.

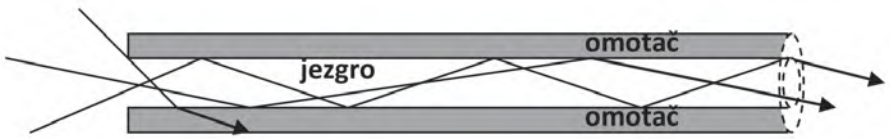


Slika 8.13. Konektor RJ45 i odgovarajuća utičnica

8.7.3. Optički kablovi

Kod ove vrste kablova, optička vlakna prenose digitalne signale u obliku modulisanih svetlosnih impulsa. Kablovi od optičkih vlakana ne podležu električnim smetnjama, imaju najmanje slabljenje signala duž kabla i podržavaju izuzetno velike brzine prenosa podataka. Koriste se i u slučajevima kada LAN mreža treba da poveže više objekata, gde se sa bakarnim kablovima mogu očekivati problemi sa uzemljenjem i atmosferskim

pražnjenjima. Optičke veze osim velike brzine prenosa obezbeđuju i potrebno galvansko razdvajanje instalacija. Često se postavljaju u objektima, u slučajevima kada se predviđa veliki mrežni saobraćaj između spratnih razvoda u odnosu na centar mreže.



Slika 8.14. Totalna refleksija kod prenosa kroz optičko vlakno

Sistemi prenosa sa optičkim kablovima se sastoje iz tri osnovna funkcionalna dela, a to su predajnik (izvor svetlosti – LED ili laserska dioda), optičko vlakno i prijemnik (foto senzor). Standardni električni signal se dovodi na LED ili lasersku diodu koje vrše konverziju u svetlost, zatim se svetlost "ubacuje" u optičko vlakno na čijem drugom kraju je prijemnik koji vrši opto-električnu konverziju posle koje se dobija standardni električni signal. Princip po kome se informacija prenosi po optičkom vlaknu bazira se na fizičkom fenomenu pod nazivom totalna refleksija. Svako optičko vlakno se sastoji iz jezgra koga čini staklo određenog indeksa prelamanja i omotača presvučenog preko jezgra. Ovaj omotač je takođe od stakla, ali ono ima drugu vrednost indeksa prelamanja. Svetlost se ubacuje u jezgro pod određenim uglom potrebnim da dođe do totalne refleksije, zbog koje se svetlosni zrak neprestalno odbija od granične površine jezgro/omotač putujući tako kroz vlakno do prijemnika.

Optičkim kablovima su danas povezana glavna komunikaciona čvorišta. Mnoga velika preduzeća i državne institucije su povezane na Internet preko optičkih kablova. Međutim, većina manjih preduzeća, kao i domovi običnih korisnika i dalje zavise od bakarne žice (UTP ili koaksijalnog provodnika). Uobičajen termin za vezu krajnjih korisnika je "last mile" i ona se najčešće ostvaruje preko bakarnog provodnika ili bežično. U budućnosti se očekuje da će i veze do krajnjih korisnika biti preko optičkih kablova, što će omogućiti brze dvosmerne veze i potpuno nove usluge za telefoniju, televiziju, radio, računare, video nadzor itd.

8.8. AKTIVNA MREŽNA OPREMA

U nastavku su opisani uređaji koji se koriste u realizaciji mreža, počev od onih neophodnih za realizaciju svih vrsta mreža pa sve do uređaja potrebnih za realizaciju WAN mreža. Tu spadaju ripiteri, habovi, mostovi, svičevi, ruteri i firewall-ovi.

8.8.1. Ripiter (*Repeater*)

Ripiteri su jednostavni uređaji sa dva porta, koji rade na fizičkom nivouNa jednom portu (priključku) ripiter prima signal i prenosi na drugi port. Pritom ripiteri vrše tzv. 3R funkcionalnost - Reamply, Reshape i Retime, tj. obnavljaju amplitudu, oblik i vremenske reference primljenog signala pre nego što ga proslede. Radi u prvom sloju OSI modela.

8.8.2. Hab (*Hub*)

Hab je mrežni uređaj koji takođe funkcioniše na prvom OSI nivou (fizičkom nivou). Na habu postoji više konektora (obično su to RJ-45 konektori). Na svaki konektor se priključuje po jedan kabl, preko kojeg se povezuje jedan računar ili server. Omogućava povezivanje više segmenata mreže u jedan segment. Hab funkcioniše slično kao ripiter: ono što primi na jednom svom portu hab emituje na svim ostalim portovima. Može se posmatrati kao višeportni ripiter. Može se koristiti kao centralna tačka u topologiji zvezde. Habovi sadrže između 6 i 24 porta. Svaki hab ima još jedan dodatni port koji se naziva uplink port. On služi za međusobno povezivanje dva haba. Povezivanje se vrši tako što se spaja uplink port jednog haba sa običnim portom drugog haba.

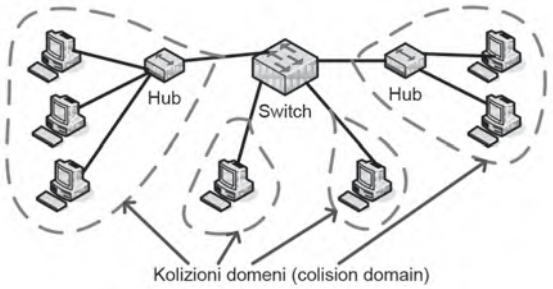
Slika 8.15. Različite veličine habova

8.8.3. Mrežni most (*Bridge*)

To je uređaj koji povezuje udaljene mrežne segmente. Radi u drugom sloju osi modela, tj. u sloju veze podataka. Mrežnim mostom se može izvršti segmentacija mreže. Spolja je sličan ripiteru, a funkciono ima sve njegove osobine uz dodatak nekoliko novih koje su veoma značajne. Most proverava sadržaj zaglavlja primljenog paketa da bi saznao MAC (fizičku) adresu izvora i odredišta. Na osnovu toga, on formira tabelu MAC adresa za svaki port. Pojedini segmenti mreže se nazivaju kolizioni domeni. Kada dobije broadcast paket (paket za sve računare u mreži), mrežni most ga samo prosleđuje i ne pamti MAC adresu iz njegovog zaglavlja.

8.8.4. Komutator – svič (*Switch*)

Komutator je za mrežni most isto što je i hab za ripiter. Dakle, na sebi ima veći broj portova. Svaki port, kao i kod mosta, ima izvestan stepen inteligencije, odnosno ne vrši samo retransmisiju paketa, već upisuje MAC adrese u odgovarajuću tabelu. Veoma značajna mogućnost koju svič poseduje je da se na svaki port sviča može priključiti stanica, a ne segment mreže. Kolizioni domen u ovom slučaju čini stanica sa odgovarajućim portom. U ovom slučaju, saobraćaj koji vidi stanica je samo onaj koji je direktno upućen za nju, kao i broadcast poruke.



Slika 8.16. Svič omogućava podelu LAN-a na više kolizionih domena

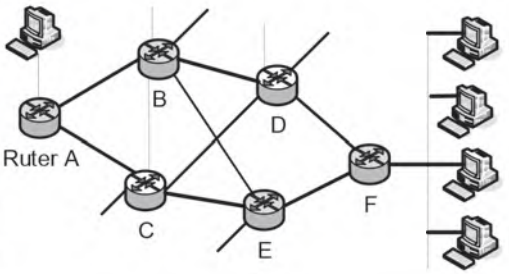
Problem koji se javlja kod upotrebe sviča je preopterećenje. Može se desiti da veliki broj paketa bude upućen na neki od portova koji treba da ih prosledi dalje i koji to nije u stanju da uradi jer kapacitet odlazne veze to ne može da podrži. Paketi koji pristižu mogu da se baferuju do izvesne granice, posle koje se odbacuju. Svičevi se bolje ili lošije nose sa ovim problemom u zavisnosti od njihovog kvaliteta (veličine bafera - memorije i brzine obrade).

8.8.5. Ruter - usmerivač (*Router*)

Za razliku od mrežnih uređaja koje smo do sada videli i koji su radili na prvom i drugom OSI nivou, ruteri rade na trećem nivou, odnosno mrežnom sloju. Glavna uloga rutera u mreži je da rutiraju (usmeravanje) pakete kako bi oni stigli do svog odredišta. Informacija koja se koristi za ovu funkciju je odredišna adresa smeštena u paketu. Ruter obavlja ovu funkciju tako što po prispeću paketa izvuče odredišnu adresu, zatim nađe odgovarajući zapis u tabeli rutiranja gde su smešteni podaci na koji port treba paket da se prosledi i odredi adresu sledećeg rutera na putu ka kojem se paket usmerava. Kada se dobije ova informacija vrši se proces komutacije (switching) gde se paket komutira sa ulaza na odgovarajući izlazni port odakle se šalje dalje.

Tabela rutiranja za ruter A

Odredište	Sledeći skok
A	.....
B	B
C	C
D	B
E	C
F	C



Slika 8.17. Ruteri usmeravaju pakete na osnovu tabele rutiranja

Pored ovih osnovnih funkcija ruteri vrše i druge funkcije kao npr. provera ispravnosti paketa, obrada kontrolnih paketa itd. Najnoviji trendovi su da ruteri treba da obavljaju i dodatne funkcije kao npr. Protokoli zaštite, kvalitet servisa i sl. koji nameću dodatne zahteve ruterima. Takođe, broj korisnika računarskih mreža je u stalnom porastu tako da je saobraćaj koji generišu korisnici sve veći. Saobraćaj se takođe uvećava usled sve novijih aplikacija koje zahtevaju veoma velike propusne opsege (npr. prenos videa u realnom vremenu). Da bi se zadovoljili zahtevi za povećanim saobraćajnim realizuju se linkovi sve većeg kapaciteta (do nekoliko desetina gigabajta po sekundi) sa tendencijom da se ti protoci podignu na terabitske brzine. To znači da obrada paketa mora biti veoma brza i efikasna. Postoji više algoritama rutiranja koji treba ovaj proces da načine što efikasnijim.

Svaki protokol rutiranja koristi različiti algoritam za utvrđivanje kada su dostupne nove rute i koja je ruta najbolja. Prosleđivanje paketa do mreža sa kojima ruter nije u direktnoj vezi može da se vrši na dva načina:

- Statičke putanje - Reč je o putanjama koje administrator ručno ustanovljava. Kada god topologija mreže iziskuje ažuriranje (na primer, prilikom kvara na vezi), administrator mreže ovakvu putanju mora da ažurira.
- Dinamičke putanje - Ove putanje ruter automatski saznaje nakon što administrator konfiguriše protokol rutiranja. Za razliku od statičkih putanja, čim mrežni administrator uključi dinamičko rutiranje, informacije o rutiranju se samim procesom rutiranja automatski ažuriraju svaki put kada se od nekog rutera u okviru mreže primi informacija o novoj topologiji.

8.8.6. Mrežni prolaz (*Gateway*)

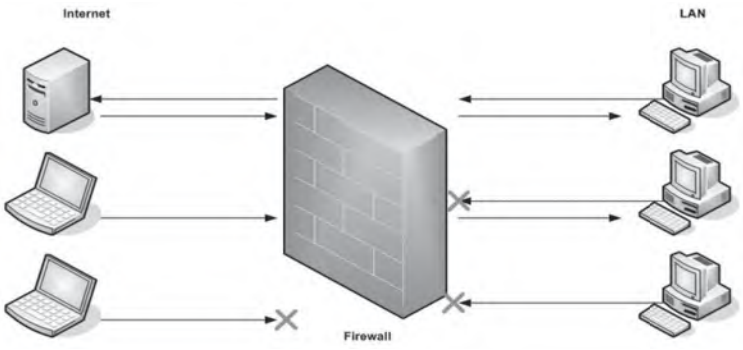
Mrežni prolaz je hardverski uređaj ili softverski paket koji povezuje dva različita mrežna okruženja. Vrši prepakivanje i pretvaranje podataka koji se razmenjuju između potpuno drugačijih mreža, tako da svaka od njih može razumeti podatke iz one druge. Zahteva značajne količina RAM memorije za čuvanje i obradu podataka. Kako povezuje različite

mreže, mrežni prolaz menja format poruka da bi ih prilagodio krajnjim aplikacijama kojima su namenjene, vrši prevođenje podataka (iz ASCII u EBCDIC kod, na primer) kompresiju ili ekspanziju, šifrovanje ili dešifrovanje, i drugo. Dakle, osnovna namena mrežnih prolaza je konverzija protokola. Radi između transportnog i aplikativnog sloja OSI modela.

8.8.7. Bezbednosna barijera (*Firewall*)

Firewall je bezbednosni hardverski ili softverski uređaj, najčešće smešten između lokalne mreže i javne mreže (Interneta), čija je namena da štiti podatke u mreži od neautorizovanih korisnika (blokiranjem i zabranom pristupa po pravilima koje definiše usvojena bezbednosna politika). Služi za sprečavanje komunikacije zabranjene određenom mrežnom polisom.

Vrlo često ne moraju svi korisnici u LAN-u da imaju jednaka prava pristupa mreži. Postavljanjem firewall uređaja između dva ili više mrežnih segmenata mogu se kontrolisati i prava pristupa pojedinih korisnika pojedinim delovima mreže. Firewall može biti softverski ili hardverski. Osnova rada firewall-a je u ispitivanju IP paketa koji putuju između klijenta i servera, čime se ostvaruje kontrola toka informacija za svaki servis po IP adresi i portu u oba smera.



Slika 8.18. Princip rada firewall-a

Firewall je odgovoran za više važnih stvari u okviru jednog informacionog sistema:

- implementira bezbednosnu politiku
- beleži sumnjive događaje
- upozorava administratora na pokušaje napada i pokušaje kompromitovanja bezbednosne politike
- u nekim slučajevima obezbeđuje statistiku korišćenja

8.8.8. Mrežna kartica

Mrežna kartica je uređaj koji povezuje računar sa računarskom mrežom. Često se naziva: mrežni adapter, mrežni interfejs, NIC... Jedan od važnijih elemenata svake mrežne kartice je MAC adresa koja čini da ovaj uređaj radi na 2. sloju OSI modela. MAC adresa predstavlja 48-bitni serijski broj iz opsega koji IEEE (Institute of Electrical and Electronics Engineers) dodeljuje proizvođaču.

Mrežne kartice su se ranije u računarima mogle naći najčešće u vidu zasebnih kartica dok se danas uglavnom integrišu u matične ploče računara. U jednom računaru se može naći i više mrežnih kartica, bilo na matičnoj ploči, bilo u vidu zasebnih kartica. U tom slučaju kartice mogu funkcionisati zasebno ili udruženo. Mrežne kartice uglavnom imaju RJ-45 za UTP kablove. Takođe, na mrežnim karticama se uglavnom nalaze i LED diode koje služe za praćenje aktivnosti kartice. Najčešće brzine na kojima rade mrežne kartice su 10, 100 ili 1000 Mbit/s. Glavni proizvođači mrežnih kartica su 3Com, Intel, Realtek, Marvell, VIA...

8.9. ETERNET (*ETHERNET*)

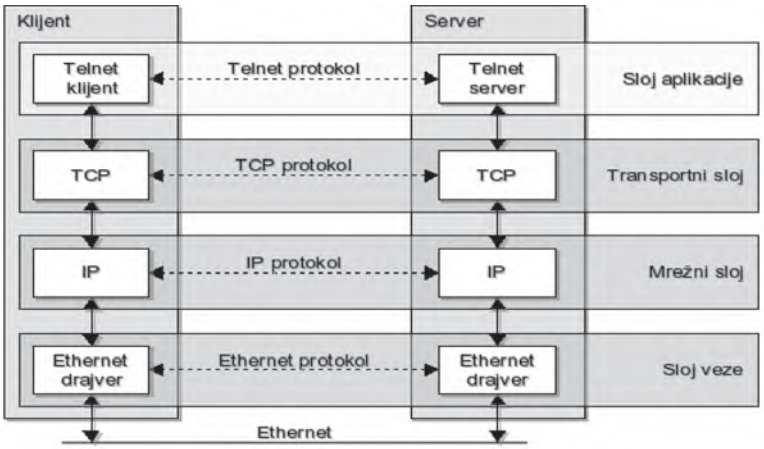
Eternet je najviše korišćena mrežna tehnologija u LAN mrežama. IEEE je uveo 1983. godine službeni standard za Eternet i nazvao ga IEEE 802.3 po imenu radne grupe odgovorne za njegov razvoj. Eternet je preživeo niz godina, u dosta velikoj meri zahvaljujući svojoj velikoj fleksibilnosti i relativnoj jednostavnosti za implementaciju i razumevanje.

Eternet sam po sebi ne obezbeđuje nikakvu sigurnost – on je jednostavan i predstavlja otvorenu fizičku sredinu za prenos podataka. Nije imun na prisluškivanje i špijuniranje. Slabosti Eterneta su:

- otvorena arhitektura gde svaki čvor može da šalje ili da prima;
- koristi širokodifuzne (broadcast) komunikacije;
- lako ga je prisluškivati;
- nema nikakav hardver za obezbeđenje;
- lako je onesposobiti mrežu.

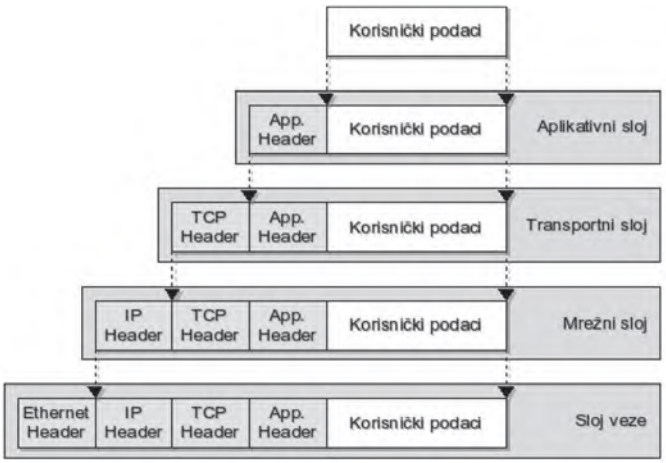
8.9.1. Internet model (*TCP/IP*)

Nasuprot OSI modelu koji je formalno standardizovan Internet model (TCP/IP) je de facto standard. Ovaj model je razvijan za potrebe Interneta i jednostavniji je od OSI modela. Jednostavnost ovog modela se ogleda u apstraktnom gledanju na tri najviša sloja OSI modela tako da Internet model propisuje samo sloj aplikacije naspram slojeva aplikacije, prezentacije i sesije kod OSI modela. Takođe, usled nedostatka formalne standardizacije Internet modela u nekim izvorima se ovaj model definiše sa 5 a u nekim sa 4 sloja. Današnje implementacije mrežnog softvera uglavnom koriste Internet model.



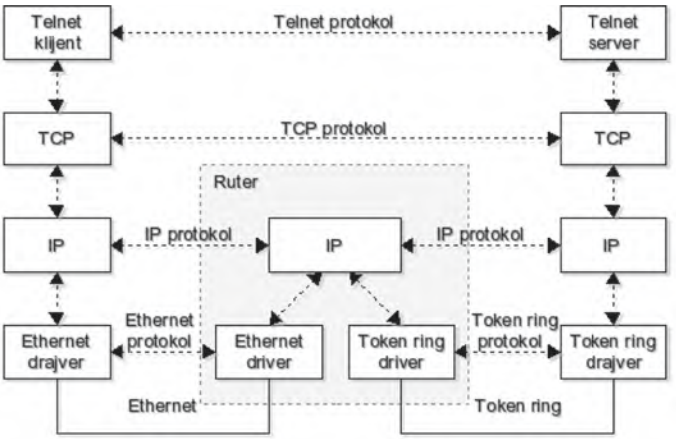
Slika 8.19. Vertikalna i horizontalna komunikacija u TCP/IP modelu

Jedna od glavnih karakteristika kod korišćenja slojevitih modela jesu horizontalna i vertikalna komunikacija. Horizontalna komunikacija predstavlja mogućnost projektovanja protokola jednog sloja na isti način kao da on direktno komunicira sa udaljenim sagovornikom tj. bez korišćenja protokola koji se koriste ispod njega. Vertikalna komunikacija je osobina slojeva da komuniciraju sa višim/nižim slojevima. Vertikalna komunikacija podrazumeva enkapsulaciju.



Slika 8.20. Vertikalna komunikacija (enkapsulacija)

Principom enkapsulacije protokol nižeg nivoa pri preuzimanju podataka dobijenih od protokola višeg nivoa dodaje sopstvene kontrolne parametre (zaglavlje) i takav skup podataka prosleđuje protokolu nižeg nivoa (koji primenjuje isti princip).



Slika 8.21. Premošćavanje razlika putem slojevite arhitekture

Slojevita arhitektura omogućava jednostavno premošćavanje problema koji se javljaju na komunikacionom putu kao i nekompatibilnosti koje postoje između učesnika. Na primer, ukoliko je jedan učesnik povezan na Ethernet a drugi na Token Ring mrežu, korišćenjem adekvatnog uređaja za premošćavanje pomenutih razlika moguće je ostvarivanje komunikacije.

Literatura

[1] M. Milosavljević, M. Veinović and G. Grubor, Informatika, Beograd: Univerzitet Singidunum, 2017.

[2] M. Veinović and A. Jevremović, Računarske Mreže, Beograd: Univerzitet Singidunum, 2018.

[3] R. J. Kurose, Computer Networking A Top Down Approach, Pearson, 2017.

[4] A. Tanenbaum, Computer Networks, Pearson, 2010.

9. KRIPTOVALUTE

9.1. UVOD

Elektronsko poslovanje (e-poslovanje), elektronska trgovina (e-trgovina), elektronsko bankarstvo i drugi oblici elektronskih transakcija koriste zajednički segment Internet tehnologija - intranet i ekstranet mreže i servise, koji su odlučujući faktor za iskorak organizacije iz okvira elektronskog poslovanja u granicama same organizacije na otvoreno i neograničeno elektronsko tržište preko Interneta umreženih korisnika u celom svetu.

Na razvoj svih oblika elektronskih transakcija, odlučujuće je uticao razvoj Internet tehnologija i svetske mreže (weba) sa gotovo neograničenim *mogućnostima izbora, pogodnostima pristupa i konkurencijom cena ponuda svake vrste.* Poseban porast imaju svi oblici e-trgovine. Razvijene su jedinstvene tehnologije, modeli trgovine, markentiški modeli, modeli komunikacija i plaćanja u oblasti e-trgovine.

Uporedo sa razvojem korisnih elektronskih servisa na Internetu i web-u, rastu i brojne zloupotrebe i napadi na elektronske servise, podatke i informacije. Značajno je povećana potreba za zaštitom svih vrsta e-transakcija na web-u preko globalne komuniacione Internet mreže. [1]

9.2. RAZVOJ ELEKTRONSKOG POSLOVANJA I ELEKTRONSKE TRGOVINE

Brojni autori tvrde da e-trgovina obuhvata ceo svet elektronski baziranih aktivnosti organizacija koje podržavaju razmenu na tržištu, uključujući cele IKT sisteme organizacija, a drugi da e-poslovanje obuhvata sve elektronski bazirane aktivnosti, uključujući i e-trgovinu. Iako oba koncepta imaju podjednako prihvatljive argumente, potrebno ih je razlikovati, pošto postoje razlike između ta dva fenomena.

Elektronsko poslovanje (*e-busines*) su digitalno omogućene transakcije i procesi u okviru jedne organizacije uz pomoć i pod kontrolom njenog informacionog sistema. E-poslovanje ne obuhvata komercijalne transakcije izvan granica organizacije gde se vrši razmena vrednosti.

Elektronska trgovina (*Electronic Commerce* ili *e-Commerce*) je razmena poslovnih informacija, održavanje poslovnih veza i vođenje poslovnih transakcija između različitih organizacija putem telekomunikacionih mreža i predstavlja nešto više od skupa Internet tehnologija. E-trgovina je kupovina i prodaja informacija, proizvoda i usluga putem

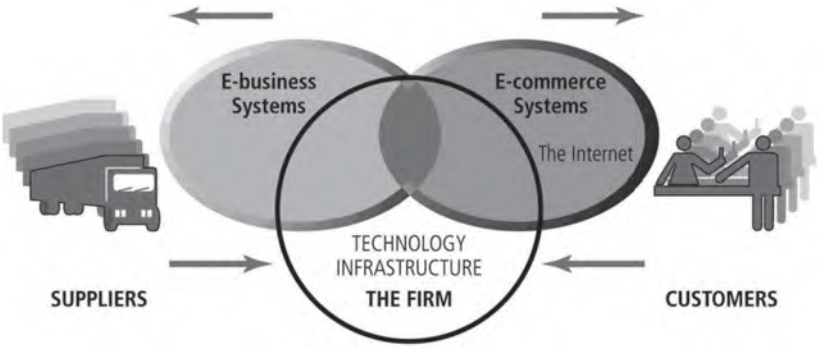
računarske mreže i podrška za bilo koju vrstu poslovnih transakcija putem digitalne infrastrukture. Omogućava digitalno omogućene komercijalne transakcije između organizcija (B2B), organizacija i pojedinaca (B2C), između pojdinaca (C2C), organizacija i vladinih agencija (B2G), vladinih agencija (G2G) itd. Digitalno omogućene transakcije obuhvataju sve transakcije zasnovane na digtalnoj tehnologiji računara, računarskih mreža i Interneta. Komercijalne transakcije se odnose na razmenu novčane vrednosti između organizacija, ili individualnih granica poslovnih sistema u zamenu za proizvode i/ili servise.

B2B je skraćenica od *Business-to-business*. U takvoj firmi, proizvodi i usluge se prodaju drugim firmama. Na primer, *Maersk Line*, svetska dostavljačka firma koja poseduje preko 600 brodova, vodeća je firma u B2B biznisu. Osnovu njenih klijenata čine firme koje se bave uvozom i izvozom. B2B nastaje kada su firmi potrebni sirovi materijali za proizvodnju ili kada joj je potrebna operativna pomoć. Na primer, proizvođač peciva sarađuje sa firmom koja prozivodi emulgatore neophodne u proizvodnji peciva. Što se tiče operativne pomoći, maloprodajna kompanija može da kupi softver koji pojednostavljuje proces zapošljavanja - HRM / *human resource management software*. [2]

B2C ili *business-to consumer* firma je ona koja **prodaje proizvode i usluge direktno potrošačima**. Restorani, maloprodajni lanci, usluge vođenja domaćinstva su neki od primera B2C poslovanja. Naziv B2C je postao popularan u kasnim devedesetim, kada je onlajn prodavcima posao krenuo usled dotcom balona. Zahvaljujući mreži, ljudi su mogli da kupe šta god su hteli za svega nekoliko minuta. ,,Dotcom boom“ se završio, ali onlajn maloprodajne firme, poput *Amazon.com* i *eBay*-a, postale su izuzetno popularne u B2C svetu. [2]

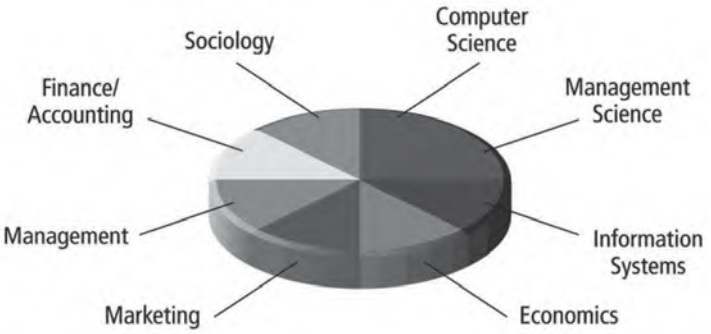
C2C (*Consumer-to-Consumer*) je model e-trgovine u kojem potrošači direktno prodaju robu ili usluge drugim potrošačima *on-line* putem. Uključuje i prodaju nekretnina, automobila i dr. preko *on-line* oglasa. Najpoznatiji primer C2C je web sajt *eBay.com***,** koji obezbeđuje reklamiranje ličnih usluga na Internetu ili ponudu konsalting usluga. Neki aukcijski sajtovi nude mogućnost C2C elektronske trgovine, omogućavajući korisnicima da ponude robu na *on-line* aukcijama. Ovaj oblik transakcija podrazumeva učešće i treće strane u *online* trgovini. Treća strana može biti neki aukcijski sajt, kao što su: *eBay, Gnutella* za prodaju muzike, *Monsters* za *online* poslove, *Owners.com* za klasifikovane reklame, itd. Poseban oblik C-C transakcija predstavljaju transakcije u direktnoj komunikaciji računara (*peer-to-peer*) komunikaciji bez posrednika, koju omogućavaju programi, kao što su *KazaA*, *Tornet* i dr. *eBay.com*, kreira *prostor za trgovinu* gde potrošači mogu dati robu na aukciju ili direktno prodati drugim potrošačima. [1]

Često se e-trgovina poistovećuje sa e-poslovanjem, što suštinski nije tačno iako je u krajnjem slučaju i e-trgovina samo jedan vid e-poslovanja. E-trgovina i e-poslovanje preklapaju se na granici poslovnog sistema organizacije u tački u kojoj se, na primer, interni sistem organizacije povezuje sa snabdevačima i kupcima. Aplikacije e-poslovasnja prelaze u e-trgovinu kada se vrši razmena vrednosti (Slika 9.1).



Slika 9.1. Razlike između e-poslovanja i e-trgovine [1]

U oblast e-trgovine uključene su brojne naučne discipline kao što su ekonomija, informacioni sistemi, menadžment, računarstvo, sociologija, računovodstvo i finansije, menadžment tržišta i marketing (Slika 9.2)



Slika 9.2. Discipline uklju;ene u oblast e-trgovine [1]

Brojni faktori u razvoju Internet tehnologija i *web*-a uticali su na razvoj tehnologija e-trgovine, ali su presudna četiri ključna razloga za kupovinu preko web-a:

- *mogućnost selekcije* iz obilja ponuda od reklama do proizvoda i zabavnih sadržaja,
- *pogodnost* pristupa izabranom sadržaju bez obzira gde se kupac nalazi i u koje vreme zahteva kupovinu i
- Cena koju velika konkurencija na otvorenom e-tržištu, veliki popusti i druge povlastice na pojedinim artiklima (npr. bestselerima) mogu učiniti prihvatljivom za brojne kategorije online kupaca

Svi navedeni faktori doveli su do razvoja jedinstvene *tehnologije e-trgovine*, koja ima sedam ključnih dimenzija:

1. *Sveprisutnost:* Internet/web dostupan od kuće, sa posla bilo odakle.
2. *Globalna dostupnost:* prelazi nacionalne granice.
3. *Univerzalni standardi:* rad zasnovan na setu Internet standarda omogućava razmenu transakcija preko web-a.
4. *Informaciona bogatstvo:* video, audio, tekst itd.
5. *Interaktivnost:* omogućava interaktivni rad sa korisnikom/kupcem.
6. *Informaciona gustina:* bogatstvo informacija na web-u smanjuje troškove informacija i povećava kvalitet.
7. *Personalizacija/kastomizacija:* omogućava da se poruke personalizuju za slanje pojedincu/grupi.

Ove dimenzije tehnologije e-trgovine značajno utiču na ukupno poslovanje organizacije, a svaka specifično na oblast e-trgovine.

Sveprisutnost: Tržište se proširuje izvan tradicionalnih granica i pomera sa privremenih i geografskih lokacija. Kreira se novi "prostor za trgovinu"- e-tržište, a kupovina se može izvršiti sa bilo gog mesta i bilo gde. su Povećane su pogodnosti za kupce, a troškovi kupovine smanjeni. Dolazi do promene industrijske strukture kreiranjem novih tržišnih kanala i proširivanjem veličine ukupnog tržišta. Takođe, se povećava ukupna efikasnost i smanjuju cene troškova prodaje. Stvaraju se mogućnosti za razvoj nove strategije diferencijacije. [1]

Globalna dostupnost: Trgovina je omogućena kroz gotovo sve kulturološke i nacionalne granice nesmetano i bez modifikacija. E-tržište uključuje potencijalno milijardu kupaca i milione poslova širom sveta. Dolazi do promena industrijske strukture smanjivanjem barijera za ulazak na tržište uz njegovo istovremeno znatno proširivanje. Smanjuju se troškovi industrijske proizvodnje i poslovanja povećanjem efikasnosti proizvodnje i prodaje. Omogućava se veća konkurencija na tržištu globalnih razmera.

Univerzalni standardi: Postoji jedinstven skup standarda tehničkih medija usvojen ***širom sveta***, koji omogućava interaktivni rad na globalnoj svetskoj mreži-Internetu. Dovode do promena industrijske strukture u pogledu smanjivanja barijera za ulazak na tržište, povećanja konkurencije unutar jedne industrijske grane i diferencijacije strategije prodaje. Smanjuju se troškovi industrijske proizvodnje i ukupnog poslovanja snižavanjem komunikacionih i računarskih troškova i omogućava delovanje široke strategije.

Informaciono bogatstvo: Video, audio i tekstualni sadržaji za marketing su integrisani u jednu marketinšku poruku i jedinstveno potrošačko iskustvo. Dovodi do promena industrijske strukture redukovanjem jačine uticaja snažnih distribucionih kanala na efekte trgovine. Dolazi do promene troškova industrijskog i kompanijskog poslovanja smanjivanjem potrebe za radom klasičnih prodavaca. Poboljšava se strategija podrške nakon obavljene prodaje.

Interaktivnost: Potrošači su angažovani u dijalogu koji dinamički prilagođava iskustvu pojedinca i čini ga saučesnikom u procesu isporuke robe na tržište. Dovodi do promene industrijske strukture redukovanjem rizika od zamene prodatog proizvoda kroz veće prilagođavanje potrebama kupca/grupe kupaca -poboljšanje personalizacija *(kastomizacije)*. Smanjuje troškove industrijskog i kompanijskog poslovanja smanjivanjem potrebe za rad klasičnih prodavaca i omogućava nove diferencijalne strategije.

Personalizacija (kastomizacija): Personalizacija markentiških poruka i proizvoda zasniva se na njihovom prilagođavanju individualnim potrebama i karakteristikama kupaca/grupe kupaca. Dovodi do promene industrijske strukture redukovanjem rizika od zamene prodatog proizvoda i smanjenjem barijera za ulazak na tržište. Takođe, smanjuje vrednosti troškova lanaca snabdevanja u industriji i prodaji smanjivanjem potrebe za radnom snagom klasičnih prodavaca.

Informaciona gustina: Dovodi do promena industrijske strukture slabljenjem značaja snažnih distribucionih i prodajnih kanala, pomeranjem pregovaračke snage direktno na kupce. Dolazi do redukovanja industrijskih i kompanijskih troškova snižavanjem troškova pristupa, obrade i distribucije informacija o dobavljačima i potrošačima. [1]

9.3. POSLOVNI MODELI E-TRGOVINE

Poslovi u organizaciji bilo kojeg tipa izvršavaju se kroz izvršavanje poslovnih procesa i aktivnosti u tim procesima. Model poslovanja ili poslovni model se definiše kao skup aktivnosti u cilju ostvarivanja dobiti (profita) na tržištu. Aktivnosti se planiraju u formi *biznis plana*, kojim se opisuje poslovni model. Poslovni model e-trgovine je specifičan u smislu da koristi jedinstvene kvalitete tehnologija i prednosti Interneta i Web-a.

Poslovni modeli na Internetu mogu biti **opšti** (*horizontalno tržište*) ili **specijalizovani** (*vertikalno tržište*). Pregled poslovnih modela e-trgovine na Internetu prikazan je u tabeli (Tabela 9.1)

Generički *poslovni model* e-trgovine sadrži 8 ključnih komponenti: predlog vrednosti, model prihoda, šanse na tržištu, konkurencija (konkurentsko okruženje), konkurentska prednost (konkurentnost), tržišna strategija, organizacioni razvoj i tim za upravljanje.

U tabeli (Tabela 9.2) prikazani su zbirno ključne komponente poslovnih modela i pitanja na koje te komponente moraju dati odgovore.

Oblik	Karakteristike	WWW
Elektronske aukcije	Elektronska ponuda cene za proizvode ili usluge	www.letsbuyit.com www.eBay.com
Elektronske prodavnice	B-B e-trgovina B-C e-trgovina B-C e-trgovina	www.cisko.com www.tehnicom.net www.amazon.com
Elektronska grupna kupovina	Elektronska prodaja robe grupama kupaca ili malim poslovnim sistemima po diskontnim cenama	www.volumebuy.com www.etrana.com
Elektronska nabavka	Lanac nabavke na Internetu	www.sap.com
Provideri Internet usluga	Omogućavanje pristupa Internetu	www.EUnet.com www.aol.com www.AT&Tworldnet.com
Elektronska razmena	Omogućavanje elektronske razmene između kupaca i prodavaca	www.patents.ibm.com www.Chem.Connect.com www.rooster.com
Elektronski posrednici	Elektronska maloprodaja	www.stephenking.com www.moster.com
Elektronski provajderii sadržaja	Informacione usluge	www.nytimes.com www.cnn.com
Elektronski portali	Informacione usluge, link-ovi sa drugim sajtovima, elektronska pošta itd.	www.Lycos.com www.Infoseek.com www.Excite.com

Tabela 9.1. Pregled poslovnih modela e-trgovine

	Komponente biznis modela	Ključna pitanja
1.	*Predlog vrednosti*	Zašto kupac treba da kupi od vas?
2.	*Model prihoda*	Kako ćete zaraditi novac?
3.	***Šansa na tržištu***	Šta je *prostor za trgovinu (tržište)* koji želite opsluživati i kolika je njegova veličina?
4.	*Konkurentsko okruženje*	Ko još zauzima vaš planirani *prostor za trgovinu*?
5.	*Konkurentska prednost (konkurentnost)*	Koje posebne prednosi vaša firma donosi na *prostor za trgovinu?*
6.	*Tržišna strategija*	Kako planirate da promovišete vaše proizvode, ili servise da privučete ciljne grupe potrošača?
7.	*Organizacioni razvoj*	Koji je tip organizacione strukture u okviru firmepotreban za realizaciju biznis plana?
8.	Tim za upravljanje	Koje vrste iskustava i obrazovanja je vežno da imaju lideri kompanije?

Tabela 9.2. Ključne komponente poslovnih modela

Predlog vrednosti definiše kako proizvodi ili servisi jedne kompanije zadovoljavaju potrebe potrošača. Ova komponenta daje odgovor na ključna pitanja zašto bi potencijalni kupac izabrao vašu kompaniju, a ne neku drugu, odnosno, šta nudi vaša kompanija, šta ne nude ili ne mogu da ponude druge konkurentske kompanije?

Model prihoda opisuje na koji način će kompanija ostvarivati prihod, profit i superiorno povratiti investirani kapital. *Model prihoda* u poslovnom modelu e-trgovine obuhvata:

- *model reklamiranja,*
- *model pretplate (u*učlanjivanja*),*
- model provizije za transakciju,
- model prodaje i
- model partnerskih odnosa (afilacioni model)

Model prihoda od reklamiranja uključuje forum za reklamiranje koji kompanija za e-trgovinu uspostavlja, objavljuje reklame i naplaćuje odgovarajuću proviziju. Poznat primer ovog modela prihoda je web lokacija Yahoo.com. Model prihoda od pretplate obuhvata prostor za reklamiranje ili druge web servise koje e-kompanija nudi korisnicima, a usluge naplaćuje kroz različite vidove učlanjivanja i pretplate za pristup nekim, ili svim ponuđenim servisima. Primeri su Consumereports.org, Sportsline.com i WallStreetJournal.com. Model prihoda zasnovan na transakcionim provizijama podrazumeva ostvarivanje prihoda e-kompanije kroz naplatu provizije za omogućene, ili ostvarene e-transakcije. Primeri ovog modela prihoda su eBay i E-Trade kompanije. Model prihoda od prodaje ostvaruje prihod e-kompanije kroz prodaju proizvoda, informacija, ili servisa. Primeri primene ovog modela prihoda su Amazon.com i DoubleClick.net, Salesforce.com. Model prihoda kroz partnerske odnose usmerava poslovanje kompanije na partnere i dobija fiksni, ili procentualni prihod od svake ostvarene prodaje, ili proviziju za davanje biznis refernci. Primer primene ovog modela prihoda je MyPoints.com.

O*čekivano tržište* (šanse na tržištu) odnosi se na procenu tržišta na koje organizacija računa, kao i na ukupnu potencijalnu finansijsku dobit ostvarljivu na tom tržištu. ***Šanse na očekivanom tržištu*** definisane su potencijalnim prihodom u svakom od segmenata tržišta na kojima kompanija ima nameru da konkuriše. Veličina tržišta predstavlja obim stvarne ili potencijalne komercijalne vrednosti u okviru koje kompanija ima nameru da posluje.

Konkurentsko okruženje ili konkurencija odnosi se na druge kompanije koje deluju na istom tržištu prodajući slične proizvode ili usluge. Zavisi od toga koliko je konkurenatskih kompanija aktivno na istom tržištu, koliko su velike konkurentske aktivnosti, koliko je udeo svakog od konkurenata na zajedničkom tržištu, koliko su profitabilne konkurentske kompanije i kako konkurentske kompanije cene svoje proizvode?

Konkurentska prednost ili konkurentnost postiže se onda kada kompanija proizvodi superiorne proizvode/usluge i/ili postiže za njih cenu na tržištu nižu od većine, ili svih konkurentskih kompanija. Postiže se na osnovu diferencijacije pristupa faktorima proizvodnje, čime se stiče prednost u odnosu na konkurenciju – makar i na kratak vremenski rok. Konkurentnost se stiče kreativnim korišćenjem:

- *asimetrije tržišta*, koja nastaje uvek kada jedan od učesnika na tržištu ima više resursa ili stekne drugu prednost (lokacija, privilegija i Slika) u odnosu na ostale i
- *prednosti prvog poteza*, gde se konkurentnost ostvaruje na tržištu, zahvaljujući prvenstvu stečenom iznošenjem proizvoda i/ili servisa.

Nepoštena konkurentska prednost (nelojalna konkurencija) dešava se onda kada jedna kompanija ostvaruje prednost na osnovu tržišnih faktora koji nisu dostupni konkurentskim kompanijama. Primera nelojalne konkurencije na klasičnom tržištu su brojni.

Savršeno (perfektno) tržište je ono na kome nema asimetrije tržišta i nelojalne konkurencije, budući da sve kompanije imaju jednak pristup svim proizvodnim/ tržišnim faktorima. Kompanija koristi svoju konkurentsku prednost za postizanje dodatne prednosti u odnosu na tržišta u okruženju.

Strategija nastupa na tržištu obuhvata viziju, inicijative i kontrolne tačke u dugoročnom detaljnom planu nastupa kompanije na novom tržištu i načina privlačenja novih kupaca. Potencijalnim kupcima treba preneti ključne elemente strategije nastupa. Poznato je da i najbolji biznis plan neće imati uspeha ukoliko nije na odgovarajući način plasiran potencijalnim kupcima.

Organizacioni razvoj opisuje kako će se kompanija organizovati za zahtevani posao. Organizacija posla se uobičajeno deli na funkcionalne celine – odeljenja, odseke, sektore i Slika, zavisno od veličine i tipa organizacije. Uobičajeno se razvoj hijerarhijske strukture organizacije kreće od opštenih ka specijalizovanim poslovima, u skladu sa rastom i razvojem organizacije.

Upravljački tim zaposlenih u organizaciji odgovoran je za realizaciju biznis modela.

Dobro je uspostaviti jak menadžerski tim, koji pruža trenutni kredibilitet investitorima, partnerima i klijentima. Međutim, ni jak menadžerski tim često ne može da spasi slab i neadekvatan biznis model. Zato je neophodno da menadžerski tim ima mogućnost promene i redefinisanja biznis modela ukoliko je to neophodno.

9.4. KRIPTOVALUTE KAO MODERINI NAČIN E-TRGOVINE

9.4.1. Bitcoin

Kada pričamo o kriptovalutama neophodno je pomenuti prvu nastalu i za sada jednu od najvrednijih kriptovaluta a to je Bitcoin.

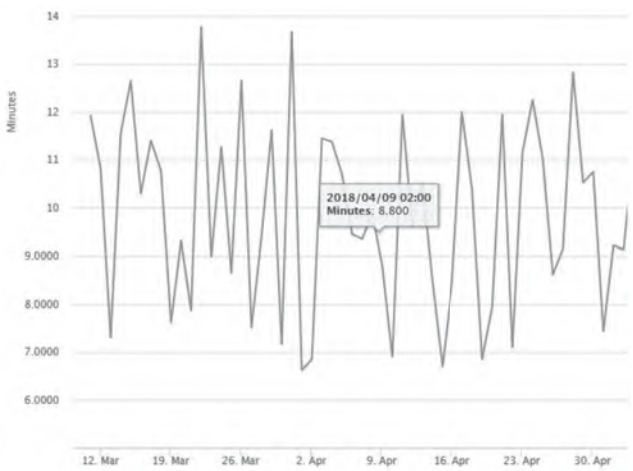
Bitkoin je prvo predstavljen kao open-source/ otvoreni softver od strane anonimnog programera ili grupe programera pod pseudonimom Satoši Nakamoto 2009. godine. Postojalo je mnogo glasina o pravom identitetu kreatora bitkoina, međutim, svi ljudi koji su pominjani u tim glasinama javno su ih negirali.

Sam Nakamoto je jednom prlikom tvrdio da je tridesetsedmogodišnji muškarac koji živi u Japanu. Međutim, zbog njegovog savršenog engleskog i softvera koji nije označen na

japanskom, postoje osnovane sumnje u istinitost ove tvrdnje. Sredinom 2010. Nakamoto je nastavio ka drugim stvarima, ostavljajući bitkoin u rukama nekoliko istaknutih članova BTC zajednice. Satoši je imenovao Gevina Andersena (Gavin Andersen) za vodećeg programera.

Za većinu ljudi, osnovna prednost bitkoina jeste njegova nezavisnost od svetskih vlada, banaka i korporacija. Nijedna vlast se ne može mešati u BTC transakcije, uvesti takse na njih ili uzeti ljudima novac. Štaviše, bitkoin pokret je krajnje transparentan - svaka pojedinačna transakcija se čuva u masovno distribuiranom javnom registru zvanom blokčein (*eng. Blockchain*).

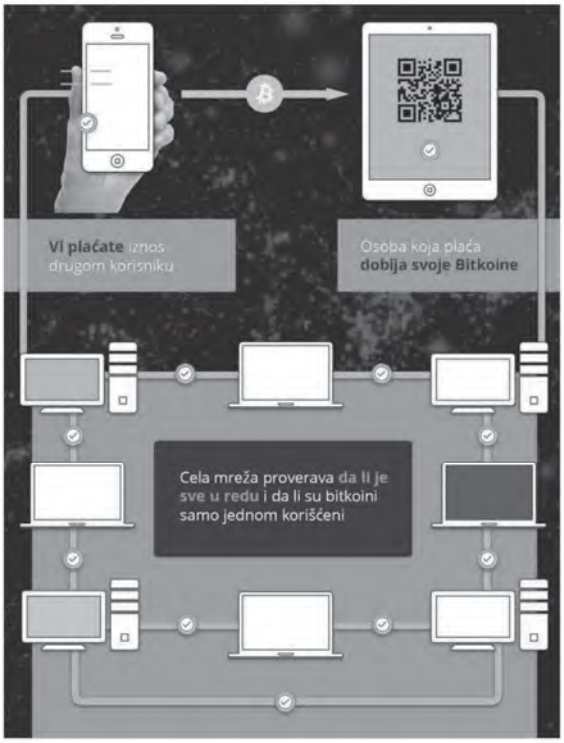
Suštinski, dok bitkoin nije kontrolisan kao mreža, svojim korisnicima pruža potpunu kontrolu nad svojim finansijama. Korisnik vidi samo količinu bitkoina na svom novčaniku i rezultate transakcije. Iza scene, bitkoin mreža učestvuje u javnom registru zvanom "blokčein". Ovaj registar sadrži svaku transakciju koja se obrađuje. Digitalni zapisi transakcija se spajaju u "blokove". Ako neko pokuša da promeni samo jedno slovo ili broj u bloku transakcija, uticaće i na svaki sledeći blok. Zbog toga što je registar javan, greška ili pokušaj prevare se može lako uočiti i ispraviti od strane bilo koga. Korisnički novčanik može da potvrdi ispravnost svake transakcije. Autentičnost svake transakcije je zaštićena digitalnim potpisima koji odgovaraju adresama slanja. Zbog procesa potvrđivanja (verifikacije), i u zavisnosti od platforme trgovanja, može biti potrebno do nekoliko minuta da BTC transakcije budu završene. Bitkoin protokol je dizajniran tako da svaki blok traje 10 minuta. [3] Bez obzira sto je blok dizajniran da traje 10 minuta to ne znači da će se poslata količina novčića sa jedne adrese na drugu pojaviti u roku od 10 minuta. Obično zbog verifikacija to potraje malo duže a taj period nije uvek isti. Potrebno je da se svaki blok obradi kako bi transakcije bile izvršene u potpunosti a to zahteva nekada period do 30 minuta.



Slika 9.3. Prosečno vreme prihvatanja bloka i
dodavanja istog u blokčein u navedenim mesecima 2018. godine [4]

U periodu kada je bitcoin nastao i kada je bilo znatno manje transakcija nego danas, vreme potrebno za izvršavanje ovog procesa je bilo mnogo manje. Do danas se u velikoj meri povećao broj transakcija pa je i vreme potrebno za obradu dosta veće. Obradu transakcije vrše takozvani rudari (*eng. miners*).

Za sada kako bi vizuelno stvorili sliku celog procesa pogledajte sledeću sliku (Slika 9.4 Proces kroz koji prolazi svaka transakcija [3]).



Slika 9.4. Proces kroz koji prolazi svaka transakcija [3]

Bitcoin je stvoren sa idejom da sadrži sledeće osobine:

- Decentralizovanost
- Anonimnost
- Transparentnost
- Brzinu
- Nepovratnost

Decentralizovanost

Jedan od glavnih Satošijevih ciljeva prilikom stvaranja bitkoina bio je nezavisnost mreže od bilo kog organa vlasti. Dizajniran je tako da svaka osoba, posao, kao i svaka mašina

koja učestvuje u rudarenju i potvrdama transakcija postane deo prostrane mreže. Čak iako neki deo mreže padne, novac će nastaviti da se kreće.

Transparentnost

Anonimnost je relativna, jer se svaka BTC transakcija, koja je ikada obavljena, čuva na blokčeinu. U teoriji, ako je adresa vašeg novčanika javno korišćena, svako može da kaže koliko novca ima u njemu tako što pažljivo prouči javni registar (blokčein). Međutim, praćenje određene bitkoin adrese i otkrivanje osobe je i dalje skoro nemoguće.

Oni koji žele da ostanu anonimni mogu preduzeti mere da prođu neprimećeni/"ispod radara". Postoje određeni tipovi novčanika kojima je prioritet neprimećenost/neprozirnost i sigurnost, ali najjednostavnije mere bezbednosti bi bile: upotreba više različitih adresa i neprenošenje velikih količina novca na jedan novčanik.

Brzina

Bitkoin mreža obrađuje isplate gotovo trenutno. Potrebno je svega nekoliko minuta da neko na drugoj strani sveta primi novac, dok bankovni transferi zahtevaju dane. [3] Kao što je već pomenuto izvršavanje transakcije za par minuta je sada malo duži period ali i dalje znatno kraći od bankovnih trasfera.

Nepovratnost

Kad jednom pošaljete BTC nekom, ne postoji način da ga vratite, osim ako primalac želi da vam ih pošalje nazad. Ovo osigurava primanje isplate, što znači da vas onaj sa kojim trgujete ne može prevariti tvrdnjom da nije primio novac.

Slika 9.5. Bitcoin logo [5]

Zbog svih ranije navedenih karakteristika, korišćenje bitcoin-a je postalo masovno i danas se dosta stvari može kupiti ovom valutom.

Godine 2009. kad je BTC plasiran, nije bilo sasvim jasno kako i gde se može trošiti. Sada se može kupiti praktično sve. Na primer, velike kompanije kao što su Microsoft i Dell

primaju uplate u BTC-u za niz svojih proizvoda i digitalnog sadržaja. Možete leteti sa aviokompanijama kao što su AirBaltic i Air Lithuania, kupiti karte za pozorište putem engleskog Theatre Tickets Direct, uzeti nekoliko boca zanatskog piva od Honest Brew pivare itd.

Bitkoin je relativno nov i prilično složen proces plaćanja. Prirodno je što su potrošačke opcije i dalje ograničene, ali svakog dana sve više preduzeća - od malih, lokalnih kafića do industrijskih giganata - prihvata plaćanje u bitcoin-u. [3]

9.4.2. Prednosti i mane bitcoin-a

Kao prednosti bitcoina možemo uzeti sledeće:

- Sloboda
- Prenosivost
- Provizija
- Ne koriste se kartice
- Sigurnost i kontrola
- Transparentno i neutralno
- Nemogućnost Falsifikovanja

Sloboda

BTC je dizajniran sa slobodom na umu. Najvažnije, sloboda od vlasti koja kontroliše transakcije, uvodi takse i raspolaže novcem ljudi. Kada je u pitanju kupovina stvari, kriptovalute su postale legitimni način plaćanja poslednjih godina. Uzimajući u obzir postojanje brojnih 'deep-web' tržišta koja primaju samo bitkoin, neke stvari možete da kupite lakše sa bitkoinom nego bilo kojom drugom valutom.

Prenosivost

Jedna od osnovnih karakteristika novca je prenosivost, što znači da se lako može nositi i koristiti. S obzirom da je BTC potpuno digitalan, praktično svaka suma novca se može poneti na usb-u ili se čak može čuvati onlajn.

Kriptovalute pružaju ljudima slobodu slanja i primanja novca samim skeniranjem QR koda ili klikom na onlajn novčaniku. Ne treba mnogo vremena, nema visokih naknada i novac prelazi od jedne osobe do druge bez nepotrebnih posrednika. Pristup internetu je sve što vam treba.

Provizija

Još jedna neosporna prednost bitcoin-a mreže je mogućnost biranja visine naknade transakcije ili čak neplaćanje provizije. Naknadu prima rudar, nakon što se kreira novi blok uz uspešan heš (hash). Obično pošiljalac plaća celu naknadu, dok se odbijanje naplate primaocu može smatrati nepotpunom isplatom.

Naknade su dobrovoljne prirode i služe rudarima kao podsticaj i kao osiguranje da će određena transakcija biti uključena u stvaranje novog bloka. Ovaj podsticaj funkcioniše i kao izvor prihoda rudarima, što im često donosi više novca nego rudarenje, naročito kad se uzme u obzir da će se rudarenje u budućnosti zaustaviti u potpunosti, kad se dostigne limit bitkoina. Stoga tržište kriptovaluta traži od korisnika da izaberu između troškova i vremena čekanja. Viša naknada za transakciju značila bi bržu obradu, dok korisnici bez vremenskog ograničenja mogu uštedeti novac.

Ne koriste se kartice

Industrija platnih kartica obuhvata debitne, kreditne i pripejd kartice, elektonski novčanik, ATM i POS kartice i druge povezane poslove. Sačinjena je od svih organizacija koje čuvaju, obrađuju i prenose podatke o vlasnicima kartica, postoje strogi sigurnosni propisi i većina glavnih brendova kartica je deo toga.Dok su objedinjena pravila i propisi dobri za velike kompanije, oni baš i ne uzimaju u obzir potrebe svakog pojedinca.Kada koristite BTC, ne treba da pristajete na sigurnosne standarde ove industrije, koji omogućuju ljudima da se odvoje na nova tržišta gde kreditne kartice nisu dostupne ili su nivoi prevara neprihvatljivo visoki.Kao rezultat, korisnici dobijaju niže provizije, priliku da prošire svoja tržišta i snize svoje administrativne troškove.

Sigurnost i kontrola

Korisnici bitkoina mogu da kontrolišu svoje transakcije: niko ne može povući novac sa važeg računa bez vašeg znanja i pristanka, kao što se ponekad dešava sa drugim vidovima plaćanja, i niko ne može ukrasti vaše informacije o plaćanju od trgovaca.

BTC korisnici mogu i da zaštite svoj novac rezervnim kopijama i šifrovanjem. Njihovi identiteti i lične informacije su uvek zaštićeni, jer se nijedna od njih ne mora otkriti da bi se plaćanje izvršilo.

Transparentno i neutralno

Svaka pojedinačna transakcija, kao i svaka informacija u vezi sa njom, uvek je dostupna svakome u blokčeinu, može se proveriti i koristiti u realnom vremenu. BTC protocol je šifrovan i zbog toga ga nijedan čovek ili organizacija ne može kontrolisati i manipulisati njime. Mreža je decentralizovana tako da je niko nikadan ne bi u potpunosti kontrolisao. Zato će BTC uvek da bude neutralan, transparentan i predvidljiv.

Nemogućnost falsifikovanja

Jedan od najpopularnijih načina falsifikovanja u digitalnom svetu je upotreba istog novca dva puta, čineći obe transakcije prevarantskim. Naziva se „dupla potrošnja“. BTC, kao i mnoge druge kriptovalute, koristi se blokčein tehnologijom i sa još nekoliko različitih konsenzusnih mehanizama ugrađenih u sve BTC algoritme, kako bi se suprotstavio ovome. [3]

Kao mane bitcoina možemo uzeti sledeće:

- Pravna pitanja
- Nivo priznavanja
- Izgubljena šifra
- Nestabilna cena-
- Kontinuirani razvoj

Pravna pitanja

Pravni status bitkoina drastično varira od zemlje do zemlje. U nekim se njegovo korišćenje i upotreba podstiče, dok je u drugima nelegalan i zabranjen. Dosta se brinulo o privlačnosti bitkoina ljudima koji vrše nelegalne radnje, neke vesti ukazuju da se njegova popularnost u potpunosti oslanja na mogućnost trošenja na nelegalne stvari. Zaista, kada se zloglasno internet crno tržište Silk Road zatvorilo, bitkoinu je instant smanjena vrednost.

Nivo priznavanja

BTC je priznat i legalan u mnogim zemljama, međutim, neke od svetskih vlada još uvek ne poseduju propise za njega,dok su ga druge otvoreno zabranile.Većina poslova, bilo malih ili velikih, još uvek ga je potpuno nesvesna. Skoro je nemoguće napustiti sve ostale valute i koristiti isključivo BTC.

Izgubljena šifra

Šifa je jedinstvena alfanumerička lozinka neophodna za pristup bitcoin novčaniku. Gubljenje te šifre suštinski znači gubljenje novčanika. Međutim, većina novih novčanika ima rezervni kopije i obnavljajuće mehanizme, s tim što korisnik treba da ih podesi pre korišćenja.

Nestabilna cena

Cena bitkoina je imala svoje uspone i padove, prolazeći razne cikluse drastičnih skokova i padova, koje neki nazivaju “mehurima” i krahovima. Tokom svoje istorije BTC je osvajao nove visine, samo da bi održao veliki pad odmah potom. Vrednost mu je nepredvidljiva,drastično i brzo se menja, što može da uzrokuje značajnu finansijsku štetu nesmotrenom investitoru.

Kontinuirani razvoj

Budućnost bitkoina je krajnje nejasna. Trenutno, vlade i banke nisu u mogućnosti da kontrolišu BTC, gotovo je neregulisan. Međutim, što postaje veći i popularniji, to će se više vlada postarati da ga stavi pod kontrolu. Regulisan BTC bio bi sasvim drugačija vrsta valute. [3]

9.5. KLJUČNI TERMINI

Kriptovalute – Isključivo elektronski novac koji je zasnovan na blokčein tehnologiji. Danas postoji preko 1500 različitih kriptovaluta a prva baluta je bila bitcoin.

Rudari – Ljudi koji poseduju jake računare uz pomoć čijeg hardvera se obrađuju blokovi u blokčein tehnologiji.

Blokčejn tehnologija – Tehnologija na kojoj su zasnovane kriptovalute a koja funkcioniše po principu blokova koji se na kraju smeštaju u jedan veliki fajl koji je svima dostupan. Na taj način svako ima uvid u sve transakcije ali ne i u lične podatke pošiljaoca ili primaoca.

Literatura

[1] M. Milosavljević, M. Veinović and G. Grubor, Informatika, Beograd: Univerzitet Singidnum, 2017, p. 355.

[2] Small Business trends, "smallbiztrends.com," [Online]. https://smallbiztrends.com/2016/03/difference-between-a-b2b-and-b2c.html. [Accessed 08 2018].

[3] cointelegraph.com, [Online]. https://rs.cointelegraph.com/bitcoin-for-beginners/what-is-bitcoin#kako-bitkoin-funkcionie. [Accessed 08 2018].

[4] www.blockchain.com, "www.blockchain.com," [Online]. https://www.blockchain.com/en/charts/median-confirmation-time?timespan=180days.

[5] K. Nikolaev, "https://hacked.com," [Online]. https://hacked.com/trade-recommendation-bitcoin-33/.

[6] W. Yang, J. Hu and M. Stojmenovic, "An alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbor structures," *Pattern Recognition,* vol. 47, no. 3, p. 1309–1320, 2014.

[7] S. L. Zabell, "Fingerprint Evidence,," *Journal of Law and Policy (Brooklyn College Law School),* vol. 143, no. 77, 2005.

10. BEZBEDNOST I RIZICI

Digitalna bezbednost može da se definiše kao zaštita našeg digitalnog identiteta, ili sve softverske i hardverske komponente koje koristimo ili koje sadrže naše privatne podatke. Ova oblast podrazumeva razumevanje svih rizika koji nas prate na internetu, kao i svesnost da upotrebimo odgovarajuće tehnike i alate kako bi minimizirali eventualne opasnosti. Postoji veliki broj aplikacija koje se izvršavaju preko računarskih mreža, koriste se Internet servisi, podaci se smeštaju na serverima baza podataka, daje se podrška informacionim sistemima i vrše se složene elektronske transakcije. U ovom poglavlju se govori o rizicima upotrebe savremenih informacionih tehnologija, razmatraju se različiti oblici računarskog kriminala, analiziraju se moguće tehnike zaštite, a na kraju su date etičke norme ovog problema.

10.1. UVOD

Moderni računari upravljaju našim novcem, zdravstvom, železnicom, avio-saobraćajem i sl. Od nas se očekuje da verujemo u informacione tehnologije. Mnoge koristi našeg partnerstva sa mašinama su jasne. Međutim, potpuno poverenje u modernu računarsku tehnologiju može da bude apsurdno i, u dosta slučajeva, opasno. U savkom ozbiljnom sistemu danas se razmatraju bezbednosni izazovi primene savremenih informacionih tehnologija: pravne dileme, etička pitanja i rizici pouzdanosti. Ovi aspekti navode nas na značajnije pitanje: Kako možemo da učinimo računare bezbednijim tako da se možemo osećati bezbednije u našem svakodnevnom poslovanju?

Računari su jedna od najznačajnijih i najrevolucionarnijih tekovina razvoja tehničko - tehnološke civilizacije. Međutim, pored svih prednosti i koristi koje sa sobom nose oni su brzo postali i sredstvo zloupotrebe nesavesnih pojedinaca, grupa, organizacija, čak i država. Računari se odavno ne posmatraju izolovano. Oni su povezani sa drugim računarima u mreže većeg ili manjeg obima, direktno ili preko Interneta. Internet je oduvek bio pogodan poligon za isprobavanje raznih zlonamernih programa. Zahvaljujući ogromnim procesorskim i memorijskim karakteristikama savremenih računara, koje se neprestano uvećavaju, automatizovani informacioni sistemi postali su nezamenjivi deo celokupnog društvenog života od proizvodnje, prometa, vršenja usluga pa do nacionalne odbrane i bezbednosti u najširem smislu.

Raznovrsne forme primene računara odavno su interesantne pojedincima i grupama koje ne birajući sredstva i način protivpravnim ponašanjem pokušavaju da sebi ili drugom pribave imovinsku korist. Tako računar postaje sredstvo vršenja različitih oblika nedozvoljenih, protivpravnih i društveno opasnih delatnosti. Računarski kriminal (ili

visokotehnološki kriminal, sajberkriminal) obuhvata aktivnosti tokom kojih se računari, računarske mreže ili računarski podaci koriste kao izvori, sredstva, objekat ili mesta izvršenja određenog krivičnog dela.

10.2. RAČUNARSKI KRIMINAL

Razvoj informacionih tehnologija ima uticaja na mnoge profesije. Ministarstvo unutrašnjih poslova koristi specijalizovane baze podataka u kojima se skladište podaci o građanima, motornim vozilima, prekršajima, zločinima, kriminalcima i sl. i na zahtev se putem raličitih upita nad bazom podataka dobijaju pravovremene informacije u mnogim kompleksnim slučajevima. U okviru Kreditnog biroa, banke skladište podatke o kreditnoj zaduženosti građana, tako da se na svaki novi zahtev brzo može proveriti dalja kreditna sposobnost građana i time se brže odgovara na postavljene zahteve za odobrenjem novog kredita. Bolnice i druge zdravstevne ustanove koriste namenske baze podataka sa DNK uzorcima koje se mogu brzo pretražiti u mnogim kritičnim trenucima (brzo reagovanje). Apotekarske ustanove su integrisane u informacioni sistem kojim se može pratiti izdavanje recepata i podizanje odgovarajuće količine lekova u datom vremenskom intervalu. Opštine i drugi organi uprave koriste specifične informacione sisteme kako bi pravovremeno pružili uslugu građanima.

Kao i kod upotrebe oružja, ljudi koriste računare kao podršku svom poslu, ali vrlo često i za kršenje zakona. Računari su moćni alati u rukama kriminalaca, a računarski kriminal je rapidno rastući problem. Danas je računar u mnogim situacijama kod kriminalaca zamenio klasično oružje. Računarski kriminal se često definiše kao bilo koji zločin učinjen zahvaljujući znanju ili korišćenjem računarske tehnologije.

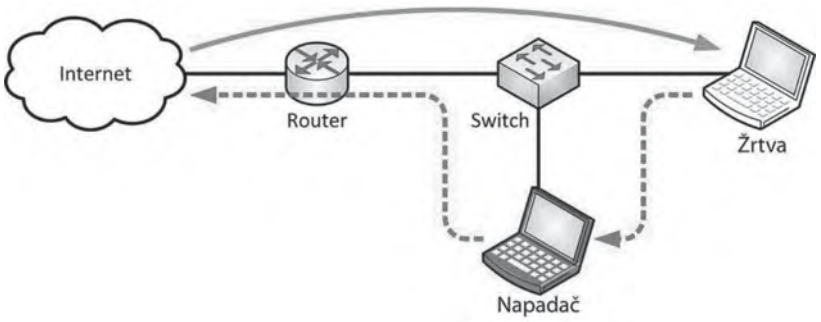
Mnogi računarski zločini prođu neopaženi. Oni koji se otkriju, često prođu neprijavljeni zato što se kompanije plaše da mogu izgubiti više zbog negativnog publiciteta nego od stvarnog zločina.

Većinu računarskih zločina počine insajderi kompanije (zaposleni unutar kompanije), a njih ne prijavljuju vlastima, čak i kada su uhvaćeni na delu. Da bi se izbegle nezgode, mnoge kompanije prikrivaju računarske zločine koje su počinili njihovi zaposleni i rukovodioci. Ove zločine tipično počine referenti, blagajnici, programeri, operateri na računarima i rukovodioci koji nemaju izraženu tehničku genijalnost. Tipičan računarski kriminalac je službenik od poverenja koji nije osuđivan, koji je došao u iskušenje zahvaljujući prilici kao što je otkriće rupe u bezbednosti sistema. Pohlepa, finansijske brige i lični problemi motivišu ovu osobu da padne u iskušenje. Naravno, ne uklapaju se svi računarski kriminalci u ovaj profil. Neki su bivši službenici koji žele da se osvete svojim bivšim šefovima. Neki su visoko tehnološke šaljivdžije koji traže izazov. Mali broj njih su korporacijski ili međunarodni špijuni koji traže poverljive informacije. Organizovani kriminal je odavno prešao na računarsku tehnologiju. Eksplozivni razvoj Interneta i servisa za trgovanje i plaćanje putem Interneta postali su meta računarskog kriminala, kako Intenet veze, tako i računarski sistemi koji su povezani na Internet.

Bezbednosni proboji preko Interneta su u porastu, interni bezbednosni proboji su u porastu i računarski kriminal uopšteno je u porastu. Ovaj porast se događa uprkos povećanoj bezbednosti i naporima bezbednosnih službi. Vredno je pomenuti da se ponekad napadi dešavaju upravo zbog propusti nekih špijunskih agencija, poput Američkog NSA, koja traži i koristi slabosti pojedinih sistema, umesto da ukaže na iste proizvođačima opreme ili softvera. U 2017. godini je doslo do objavljivanja u javnosti velikog dela špijunskog alata kojeg koristi ova agencija. Ovaj alat je iskorišćen za do sada barem dva velika napada na više stotina hiljada kompjutera koji koriste Microsoft Windows operativni sistem. U prvom napadu, zvanom "WannaCry", je blokirano 300,000 računara, iznuđivajući pare od žrtava.

10.2.1. Krađa preko računara

Krađa je najčešći oblik računarskog zločina. Računari se koriste za krađu novca, robe, informacija i računarskih resursa. Neke vrste računarskih zločina su toliko česte da su ovi zločini dobili i svoja imena. Česta prevara koristi proces koji se naziva *spoofing* (obmana) za krađu lozinki. Tipičan obmanjivač pokreće program koji na nekom računaru imitira ekran za logovanje stranice koja žrtva tipično posećuje. Kada žrtva otkuca ID i lozinku, program reaguje npr. porukom o grešci a lozinka se pamti. Slika 10.1 opisuje šemu zvanu 'Man in the Middle', ili MITM napad, gde napadač presretne saobraćaj žrtve, i pozira kao autentična stranica koju posećuje žrtva.



Slika 10.1. Spoofing Napad [1]

Spoofing, kao vrsta napada, ima i svoj pasivni oblik, gde napadač može da pošalje email žrtvi, navodeći je da unese svoju lozinku na sajtu napadača koji liči na originalan sajt. Postoji i manje tehnički zahtevan napad, gde se napadač telefonom [2] predstavlja kao Američka porezka služba, i ubeđuje žrtvu da nije plaćen porez u iznosu od nekoliko hiljada dolara.

Ponekad napadači koriste računare i drugi alat da ukradu kompletne identitete. Sakupljanjem ličnih podataka: brojeva kreditnih kartica, brojeva vozačkih dozvola, brojeva

socijalnog osiguranja i drugih podataka – lopov može efikasno da se izdaje kao neko drugi, čak i da počini zločine u ime te osobe. Kradljivcu identiteta (*identity theft*) često nije potreban računar, zato što se mnoge osetljive informacije mogu naći na otpadu i u ličnom smeću. Jedan od dobro poznatih problema danas je i npr. uništavanje nepotrebnih CD-ova, ili dokumenata, na kojima može biti zaostalih važnih informacija. Vrlo često oni se ne uništavaju, već se jednostavno bacaju.

10.2.2. Softverska piraterija i zakon o intelektualnoj svojini

Softverska piraterija – nezakonito umnožavanje zaštićenog softvera – je izuzetno rasprostranjena. Milioni računarskih korisnika prave kopije programa koje ne poseduju legalno i dobijene kopije distribuira, često besplatno. Iz razloga što samo mali broj softverskih kompanija koristi fizičke metode za zaštitu od kopiranja, kao što je hardverski ključ za zaštitu, kopiranje softvera je jednostavnije od umnožavanje audio diskova ili fotokopiranje poglavlja knjiga. Na nesreću, mnogi ljudi nisu svesni da kopiranje softvera, snimljene muzike i knjiga može da prekrši zakone koji štite intelektualnu svojinu. Drugi jednostavno to ignorišu, ubeđeni da softverske kompanije, muzičke kuće i izdavači već zarađuju dovoljno novca.

Softverska industrija tvrdi da gubi ogromne novce svake godine zbog softverskih pirata. Procenjuje se da je više od jedne trećine kompletnog softvera u upotrebi ilegalno kopirano. Piraterija može da bude posebno problematična za male softverske kompanije. Razvijanje softvera je jednako teško za njih kao i za velike kompanije kao što su Microsoft i Oracle, ali njima često nedostaju finansijski i pravni resursi da pokriju svoje gubitke od piraterije. Sa druge strane, pojedinci koji koriste kopiran softver često to rade da bi naučili kako se koristi isti. Ta njihova želja za znanjem često i ide u korist firmama čiji se softver kopira, jer služi kao reklama softvera u pitanju. Korisnici se naviknu načinu rada jednog softvera koji je njima dostupan u datom trenutnku, i onda iz navike ne žele kasnije da prelaze na konkurentske softvere. Kao primer služi firma Microsoft, i njihova praksa davanja besplatne license studentima širom sveta za korišćenje svog Office paketa. Studenti se naviknu na njega, i kasnije u radnom okruženju, traže od poslodavca da kupi isti, jer već znaju da ga koriste.

Iako softverska industrija tvrdi da je piraterija krađa, lako se vidi da situacija nije tako prosta. Krađa, u klasičnom smislu, je kada lopov otuđi svojinu vlasnika. Softverska piraterija je kada se napravi kopija softvera, gde prvobitni imalac ne gubi ništa. Softverske kuće tvrde da za svaku neovlašćenu kopiju softver gube iznos novca jednak ceni kopiranog dela. To ne može da se izjednači jer je u pitanju funkcija potražnje i ponude. Ako pojedinac može besplatno da dođe do nečega, u ovom slučaju softvera, to ne znači da bi koristio to isto da je njegova cena van budžetskog dometa.

10.2.3. Intelektualna svojina i pravo

Pravna, definicija intelektualne svojine obuhvata rezultate intelektualne aktivnosti u umetnosti, nauci i industriji. Zakoni o autorskom pravu imaju tradicionalno zaštićene forme literarnog izražavanja, patentno pravo štiti mehaničke pronalaske i ugovorno pravo pokriva privredne tajne. Softver se ne uklapa u bilo koju od ovih pravnih kategorija. Zakoni o autorskom pravu štite većinu komercijalnog softvera, ali samo nekoliko kompanija ima finansijsku moć da uspešno brani patentna prava za zaštitu softverskih proizvoda.

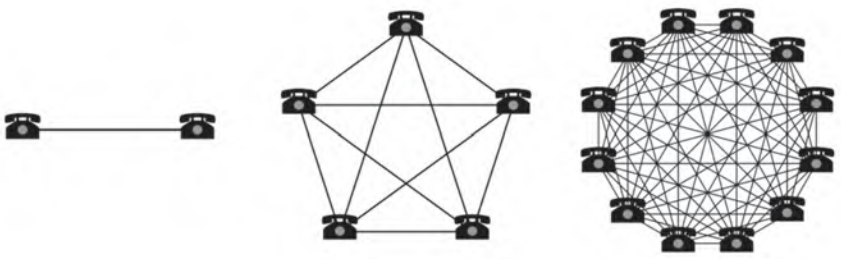
Svrha zakona o intelektualnoj svojini je obezbeđivanje da umni rad bude pravedno nagrađen i da podstakne inovaciju. Programeri, pronalazači, naučnici, pisci, urednici, filmski radnici i muzičari zavise od ideja i izražavanja tih ideja radi ostvarivanja prihoda. Ideje su informacija, a informaciju je lako kopirati – posebno u ovom elektronskom dobu. Zakoni o intelektualnoj svojini su napravljeni da zaštite ove profesionalce i da ih podstaknu da nastave sa kreativnim naporima tako da društvo može da izvuče korist od njihovog budućeg rada.

Zakoni pomažu autorima da postignu svoje ciljeve. Romanopisac može da posveti dve ili tri godine svog života za pisanje remek dela, uveren da neće pronaći nelegalnu kopiju u slobodnoj prodaji. Filmski studio može da investira ogroman novac u film, znajući da će se investicija postepeno vratiti, preko prodaje ulaznica i iznajmljivanja. Pronalazač može da dugo radi da bi stvorio bolji uređaj i zna da niko neće ukrasti njegovu ideju.

Međutim, ponekad se zakoni o intelektualnoj svojini primenjuju na način koji može da uguši inovacije i kreativnost umesto da ih štite. 1999. godine Amazon.com je zaštitio patent za "*one-click shop*ping", sprečivši druge sajtove e-trgovine da pruže svojim klijentima slično. Slično tome, *SightSound* je patentirao sva plaćena preuzimanja željenih digitalnih video ili digitalnih audio sadržaja, *RealNetworks* je patentirao audio i video striming, a *British Telecom* tvrdi da drži patent iz 1976. godine koji pokriva svaki Web hiperlink! Većina stručnjaka se slaže da su ove ideje isuviše jednostavne i široko primenjive da bi ih posedovala jedna kompanija. U većini slučajeva, vlasnik patenta nije pronalazač koncepta – Daglas Engelbart je demonstrirao hiperlink veze još davne 1967. godine na Istraživačkom institutu Stanford, a nije ga patentirao. Takvi obimni patenti obično završe u sudnici, gde pravni i tehnološki stručnjaci raspravljaju vrednosti i polje primene ideja i zakone napravljene da ih štite. U međuvremenu, zakonodavci pokušavaju da ažuriraju zakone radi usklađivana sa stalnim tehnološkim promenama.

Većina postojećih zakona o autorskom pravu i patentu, koji su pravljeni u eri štampe i mehaničkih pronalazaka, su zastareli, kontradiktorni i neadekvatni za današnju informacionu tehnologiju. U tome i jeste problem, jer ne može direktno da se primene zakoni namenjeni na materijalnu svojinu na ne materijalnu, kao što je softver. Jedna prosta obzervacija zbog čega ne mogu zakoni biti prosto preslikani jeste u činjenici da kada se automobil, mobilni telefon, garderoba, ili bilo koja materijalna svojina troši, ona vredi manje. Kontra je za ne materijalnu svojinu, jer što je neki softver popularniji, i više se

koristi, on vredi više. Ovaj fenomen se zove mrežni efekat, i objašnjen je na slici 10.2. Kako raste broj korisnika jednog servisa, ili posetioca jedne web stranice, taj servis ili stranica vredi više. Ta vrednost može se definiše kao popularnos autora čije se delo proširilo po internetu, ili može da dođe do mnogo direknije finansijke nagrade. Na primer, koliko vredi Youtube klip koji ima 10,000 pregleda? 100,000? 1,000,000,000? Što više pogleda klip ima, više zaradi od reklame.



Slika 10.2. Mrežni Efekat [3]

Prema većini zakona, nije legalno pisati program koji zaobilazi šeme za zaštitu od kopiranja, bez obzira da li se taj program koristi ili ne koristi za ilegalno kopiranje DVD diskova, elektronskih knjiga ili drugog zaštićenog materijala. Zločin je i deljenje informacija o načinu razbijanja zaštite od kopiranja. Kritičari govore da zakon onemogućava slobodu govora, akademsku slobodu i princip fer korišćenja – tradicionalnog prava pravljenja kopija zaštićenog materijala za lično i akademsko korišćenje i za druge nekonkurentne svrhe. Po pitanju softvera pravni sistem se još uvek dograđuje. Bilo da se uhvate u koštac sa aspektima piraterije ili monopola, zakonodavci i sudije moraju da se bore sa teškim pitanjima o inovaciji, svojini, slobodi i napretku. Ova pitanja će verovatno ostati uz nas prilično dugo.

10.3. ZLONAMERNI SOFTVER

Još jedan vid računarskog kriminala je sabotaža hardvera ili softvera. Reč sabotaža dolazi iz početaka industrijske revolucije, kada su buntovni radnici isključili nove mašine udarajući drvenim cipelama, zvanim “saboti“, u zupčanike. Moderni računarski *saboteri* za izvršenje destruktivnog dela obično koriste zlonameran softver (*malware, malicious software*). Zlonamerni softver je skup instrukcija koji se pokreću na korisnikovom računaru i čine da korisnički računar radi ono što napadač želi. Nazivi koji su dati sabotirajućim destruktivnim programima – virusi, crvi i trojanski konju – zvuče više nalik biologiji nego tehnologiji, a većina programa čak i imitira ponašanje živih organizama.

Tipovi	Karakteristike	Primeri
Virus	Inficira host fajl, samo-kopira se, u većini slučajeva mu je potreban ljudski faktor da bi se samo-kopirao (otvaranje fajla, čitanje *maila*, boot-ovanje sistema, ili izvršavanje inficiranog programa).	Stuxnet, Slammer
Crv	Širi se putem mreže, samo-kopira se, u većini slučajeva nije mu potrebna ljudska interakcija da bi se širio.	Sobig.F, Morris Worm, Code Red, MyDoom
Trojanski konj	Izgleda kao koristan program, ima prikrivenu malicioznu svrhu	ILOVEYOU
Adware, Spyware	Spyware -špijunski softver, *Adware*-reklamni špijunski softver, često se sadrže u drugim softverima	Gator, Save
Maliciozni mobilni kod	Čine ga mali programi skinuti sa nekog udaljenog sistema i pokrenuti lokalno sa minimalnim, ili bez učešća korisnika. Tipično pisani u: *Javascript, VBScript, Java,* ili *ActiveX*.	Cross Site Scripting
Backdoor	Zaobilazi sigurnost sistema da bi omogućio pristup napadaču	Netcat, VNC
Rootkit	Manipuliše sa srcem operativnog sistema, kernelom, skriva i stvara backdoorse	Adore, Kernel Intrusion System
Kombinovan *malware*	Kombinacija više različitih tehnika prethodno prikazanih da bi se stvorio bolji *malware*	Lion, Bugbear.B
Ransomware	Vrsta malicioznog softvera koja zarobi, i šifruje fajlove ili čitav kompjuter korisnika i zapreti da će sve izbristati, ili sve objaviti ako se na plati otkup	WannaCry, CryptoLocker, Petya

Tabela 10.1 Tipični primeri zlonamernog softvera sa osnovnim karakteristikama

Mnogobrojni su problemi koje može da proizvede zlonamerni softver, a neki od njih su:

- brisanje osetljivih fajlova sa hard diska,
- inficiranje korisničkog računara na način da on postaje “odskočna daska“ za dalje širenje,
- krađa podataka (lične i finansijske prirode),
- praćenje aktivnosti na tastaturi,

- praćenje promena na Desktopu,
- sakupljanje podataka o navikama korisnika,
- skrivanje fajlova, procesa i mreže,
- korišćenje zaraženog računara kao skladište za dodatne zlonamerne kodove, ukradene informacije, slanje spam emailova, itd.

U svakodnevnom govoru obično se ne pravi razliku između trojanskih konja, virusa i crva - svi se nazivaju računarski virusi. Bez obzira kako se nazivaju, ovi programi čine život komplikovanim i skupim za ljude koji zavise od računara. Istraživači su identifikovali više od 18000 vrsta virusa, sa pojavljivanjem 200 novih svakog meseca.

10.3.1. Virusi

Biološki virus ne može da se sam reprodukuje, ali može da osvoji ćelije drugog organizma i upotrebi reproduktivni mehanizam svake ćelije domaćina da napravi svoju kopiju. Nove kopije napuštaju domaćina i traže nove domaćine radi ponavljanja procesa. Softverski virus funkcioniše na isti način. Širi se sa programa na program, ili sa diska na disk, i koristi svaki zaraženi program, datoteku ili disk da napravi što više svojih kopija. Softver virusa je obično skriven u operativnom sistemu računara ili u aplikativnim programima. Neki virusi ne rade ništa osim reprodukcije, drugi prikazuju poruke na ekranu računara, dok ostali uništavaju podatke ili brišu diskove.

Kao i većina softverskog koda, virus je obično napravljen za određen operativni sistem. Virusi za Windows osvajaju samo diskove sa Windows-om, virusi za Apple osvajaju samo Apple diskove, itd. Postoje izuzeci: Makro virusi kače sebe na dokumenta koja sadrže makroe – usađene programe za automatizovanje zadataka. Makro virusi mogu da se šire preko računarskih platformi ako su dokumenta napravljena i širena korišćenjem međuplatformskih aplikacija – najčešće Microsoft Office aplikacija. Makro virusi mogu da se šire preko email priloga bezazlenog izgleda. Virusi raširenі preko email-a se ponekad nazivaju email virusi.

Jedan od najpoznatijih email virusa bio je virus Melissa iz 1999. godine. Melissa metod rada je tipičan za email viruse: Korisnik računara primi "Važnu poruku" od prijatelja: *"Here is that document you asked for . . . don't show it to anyone else".* Na primer, prikačen dokument u MS Word-u sadrži spisak lozinki za pornografske Internet sajtove. Međutim, on sadrži nešto drugo: makro virus napisan u Visual Basic skript jeziku ugrađenom u MS Office. Nakon što se dokument otvori, makro virus šalje kopije email poruke i zaraženog dokumenta na prvih 50 imena iz korisnikovog Outlook adresara. Za par minuta, još 50 potencijalnih žrtava virusa Melissa prima poruke očigledno od nekoga poznatog – korisnika nedavno zaraženog računara. Melissa se raširio kao požar među Windows sistemima, zarazivši 100000 sistema za samo nekoliko dana. Melissa nije projektovana da ošteti sisteme, ali je nagli nalet poruka oborio neke email servere.

Uskoro posle virusa Melissa, sličan ali destruktivniji virus nazvan Černobil zarazio je više

od 600000 računara širom sveta. Samo Južna Koreja je pretrpela 300000 napada; virus je oštetio oko 15% njenih PC računara, a šteta je bila 250 miliona USD. U maju 2000. godine, virus nalik Melisi pod imenom Love Bug raširio se sa PC računara na Filipinima širom sveta preko bezazlenog priloga email poruke "I Love You". Za samo par sati, Love Bug je napravio milijarde dolara štete zbog izgubljene proizvodnje i oštećenja računarskih sistema.

10.3.2. Trojanski konji

Trojanski konj je program koji može da radi nešto korisno dok istovremeno sprovodi neko tajno destruktivno delo. Kao u staroj priči o drvenom konju koji je uveo Grčke vojnike kroz kapije Troje, softverski trojanski konj krije pravog neprijatelja. Ovi programi često imaju imena nalik igricama ili korisničkim programima. Kada neiskusni pojedinac preuzme i pokrene takav program, on može obrisati fajlove, promeniti podatke ili prouzrokovati neki drugi vid štete. Neki saboteri mreže koriste trojanske konje da proslede tajne podatke drugim neovlašćenim korisnicima.

Jedna vrsta trojanskog konja - logička bomba je programirana da napadne kao reakcija na određen događaj ili redosled događaja. Na primer, programer može da ugradi logičku bombu koja je projektovana tako da uništi fajlove sa podacima ako se programer ikada pojavi na spisku otpuštenih u fajlu personalne službe kompanije. Logička bomba može da se aktivira kada se uloguje određeni korisnik, unese specijalna šifra u polje baze podataka ili korisnik izvrši određen redosled radnji. Ako se logička bomba aktivira događajem povezanim sa časovnikom, ona se zove tempirana bomba. Dobro je poznat virus sa logičkom bombom, koji je programiran da uništi PC fajlove sa podacima na Mikelanđelov rođendan.

Trojanski konji mogu da prouzrokuju ozbiljne probleme u računarskim sistemima svih veličina. Da stvar bude gora, mnogi trojanski konji prenose softverske viruse.

10.3.3. Crvi

Nalik virusima, zlonamerni softveri poznati pod nazivom crvi koriste računare domaćine da se reprodukuju. Za razliku od virusa, programi crva prenose se nezavisno preko računarskih mreža, tražeći nezaražene radne stanice u kojima će se reprodukovati. Crv može da se reprodukuje sve dok se računar ne blokira usled nedostatka slobodne memorije ili prostora na disku. Tipičan segment crva nalazi se u memoriji pre nego na disku, tako da se crv može eliminisati potpunim isključivanjem svih radnih stanica na mreži.

Prvi poznati crv napravio je student Cornell-a u okviru eksperimenta, 1988. godine. Crv je slučajno pušten na Internet i blokirao je 6000 računara širom SAD. U leto 2001. godine pojavio se crv nazvan Code Red. Ovaj crv nije napadao PC računare. Njegova

meta su bili Internet servisi koji rade pod Microsoft serverima. Vlada SAD i Microsoft su poslali upozorenja o crvu i napravili besplatne softverske zakrpe radi zaštite servera. Čak i tada, mnogi serveri su bili onesposobljeni usled ponavljanih napada od strane crva, uključujući servere koje poseduje i koristi Microsoft.

10.3.4. Antivirusna zaštita

Savremeni virusi mogu da se raširе brže i naprave veću štetu od virusa od pre par godina iz nekoliko razloga. Prenos podataka na Internetu se neprekidno ubrzava, a samim tim se ubrzava i prenošenje virusa. Web strane, makroi i druge tehnologije daju autorima virusa nova mesta za skrivanje štetnih programa. Pored toga, povećana standardizacija Microsoft aplikacija i operativnih sistema olakšala je širenje virusa. Kao što su prirodno mešovite šume otpornije na bolesti od jednovrsnih farmi drveća, mešovita računarska okruženja su manje podložna na virusne napade od organizacija u kojima svi koriste isti hardver i softver.

Antivirusni programi (poznati kao AV softver) su projektovani da traže viruse, obaveste korisnike kada ih nađu i uklone ih sa zaraženog diska ili fajla. Većina antivirusnih programa neprekidno nadgleda aktivnosti sistema, prateći i izveštavajući o sumnjivim radnjama nalik virusima. Međutim, ne postoji antivirusni program koji može da otkrije sve viruse. I ovi programi, sa pojavom novih virusa, moraju stalno da se poboljšavaju. Većina antivirusnih programa može automatski da preuzme novi programski kod sa Web-a. Međutim, ponekada je softverskim kućama potrebno nekoliko dana da razviju i distribuiraju zakrpe za nove viruse. Za to vreme destruktivni virusi mogu da načine veliku štetu.

Antivirusni softveri rade na osnovu baze kôdova virusa koje su proizvođaci ovog softvera uspeli da otkriju i definišu. Ovakvi softveri jednostavno pretražuju kôdove koje imaju u svojoj bazi. U slučaju pojave novog virusa antivirusi neće moći da ga otkriju. Eventualno se takav kôd definiše kao “mogući virus” ali ga nije moguće ukloniti.

Borba protiv virusa, kao i razvijanje novih virusa se neprestano razvija. Većina korisnika je naučila da ne otvara nedefinisane email priloge, a prodavci softvera su počeli da modifikuju email aplikacije da bi sprečili ovu vrstu napada. Međutim, crv nazvan *Bubble-Boy d*emonstrirao je da sistem može da se zarazi email-om čak i ako mail nije otvoren. Neki virusi su čak razvijeni da zaraze HTML kôd na Web stranama ili HTML email porukama. HTML virusi ne mogu (trenutno) da zaraze računar kada se pregleda zaražena Web strana. Zaraženi HTML kod mora da se preuzme download-uje na računar.

Softverske kompanije neprekidno testiraju njihove proizvode na bezbednosne propuste i pokušavaju da ih učine otpornijim na viruse, crve i druge napade. Iz razloga što je Microsoft Windows meta velike većine zlonamernih softvera, Microsoft periodično pušta bezbednosne zakrpe – programe koji ispravljaju (krpe) potencijalne bezbednosne proboje u operativnom sistemu. Ove zakrpe su dostupne za slobodno preuzimanje

ili automatsko ažuriranje svim korisnicima OS. Nekada se i preventivne bezbednosne mere mogu zloupotrebiti. U leto 2003. godine pojavio se crv nazvan *MS Slammer*. Nakon što je Microsoft objavio bezbednosnu zakrpu koja popravlja taj problem MS Slammer je ponovo aktiviran. Publikovanjem ranjivosti, Microsoft je nenamerno inspirisao zlonamerne programere da naprave crv. Oni su iskoristili činjenicu da mnogi korisnici računara nisu instalirali bezbednosne zakrpe, ostavljajući tako svoje sisteme otvorene za napad. Kao odgovor na MS Slammer, dobronamerni programer je objavio crva-pomagača projektovanog da potraži na Internetu računare koji su zaraženi crvom *MS Slammer* i primeni Microsoft-ovu bezbednosnu zakrpu na te mašine. Međutim, ovaj crv je prouzrokovao druge probleme, usporio je veliki broj računara (do blokiranja) neprekidnim proverama na postojanje bezbednosnih problema.

Svi moderni antivirusni programi imaju nekoliko komponenata. Ove komponente su najčešće deo za proveru fajlova (*scan*), deo za dezinfekciju - odnosno čišćenje zaraženih programa (*clean*) i stalno aktivni deo koji nadgleda ulazno-izlazne operacije na računaru i proverava da li se možda tu kreće i neki virus (*monitor*). Scan programima se proverava sadržaj diska u potrazi za virusima. Sprečava se zaraza tako što se skeniraju fajlovi, direktorijumi ili celi drajvovi. Ukoliko se otkrije virus, scan program će automatski pokrenuti clean deo i pokušati da dezinfikuje fajl. Čišćenje se vrši tako što se unutar zaraženog fajla briše kod virusa. Nekad je jedino rešenje brisanje zaraženog fajla. Štaviše, to je najbolje rešenje koje treba primenjivati kad god je moguće. Što se tiče monitor programa, većina antivirusnih programa ovakve programe instalira da se automatski startuju po podizanju računara.

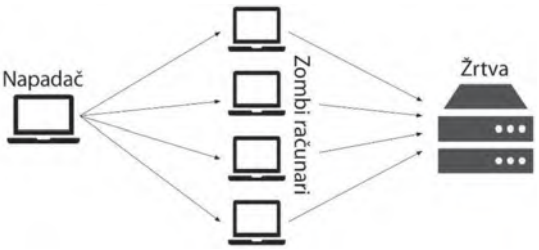
10.4. HAKOVANJE I ELEKTRONSKI UPADI

Davnih 70-tih godina prošlog veka, haker je bila osoba koja je uživala u učenju detalja o računarskim sistemima i pisanju pametnih programa koji su se nazivali hakovi. Hakeri su pretežno bili entuzijaste i radoznale osobe sa ličnim osobinama: inteligentan, idealističan, ekscentričan i bezopasan. Većina tih ranih hakera bili su, u stvari, arhitekte mikroračunarske revolucije.

Termin hakerisanje danas ima novo, zloslutnije značenje. Iako puno ljudi i dalje koristi ovaj termin da opiše softverskog genijalca, on se češće odnosi na neovlašćen pristup računarskim sistemima. Mnogi hakeri su motivisani jedino radoznalošću i intelektualnim izazovom. Kada budu otkriveni pravdaju se izgovorom da rade za dobrobit društva ukazivanjem na bezbednosne probleme u komercijalnim softverskim proizvodima. Drugi, zlonamerni, hakeri koriste trojanske konje, logičke bombe i druge metode da nanesu štetu pojedincima ili celim organizacijama. Rastući broj računarskih provalnika su usmereni na krađu brojeva kreditnih kartica i drugih osetljivih i dragocenih informacija. Ova vrsta krađa je teška za otkrivanje i praćenje zato što originalna informacija ostaje nepromenjena kada je kopija ukradena.

Napad poznat kao odbijanje usluga ili *Distributed Denial of Service* (DDoS) dešava se

na serverima i Web sajtovima, a manifestuje se sa ogromnim lažnim saobraćajem. Posledica je da serveri postaju isključeni, odbijajući usluge legitimnim korisnicima i klijentima. Na slici 10.3, se vidi šema pomenutog napada, gde poplava zahteva dolazi od puno kompromitovanih sistema distribuiranih širom Interneta. Zombi računari su zaraženi virusom koji dozvoljava napadaču da ih kontroliše. Kada napadač odluči da pokrene napad, naredi zombi računarima da zahtevaju pristup sajtu žrtve, koja nema dovoljno kapaciteta da zadovolji svim zahtevima. Rezultat uspešnog napda jeste da sajt žrtve bude nedostupan.



Slika 10.3. DDOS napad [4]

Ova vrsta napada je popularna kada napadač ima kao cilj uništavanje konkurencije, ili zaustavljanje širenja informacija. Dokazano je čak da ne mora običan zahtev otvaranja glavne stranice sajta da bude jedina dodirna tačka napada. Napadač može da pokuša da otvori stranice na sajtu žrtve koje koriste puno rezursa žrtve, kao što su pozivi ka bazi podataka u pozadini sajta, da bi brže oborili žrtvu, i došli do cilja.

10.5. BEZBEDNOST – SMANJENJE RIZIKA

Sa porastom računarskog kriminala, računarska bezbednost je postala važna briga za administratore sistema i računarske korisnike. Bezbednost se odnosi na zaštitu računarskih sistema i informacija protiv neželjenog pristupa, oštećenja, modifikacije ili uništavanja. Računari imaju dve unutrašnje karakteristike koje ih ostavljaju otvorenim za napade ili greške u radu:

1. Računar radi tačno ono za šta je programiran, uključujući otkrivanje poverljivih informacija. Svaki sistem koji može da se programira, može i da se reprogramira.
2. Svaki računar može da uradi samo ono za šta je programiran. On ne može da zaštiti samog sebe od otkazivanja ili namernih napada osim ako su takvi događaji posebno pretpostavljeni, obrađeni i odbijeni odgovarajućim programiranjem.

Vlasnici i administratori računara koriste različite bezbednosne tehnike da zaštite svoje sisteme, obuhvatajući svakodnevne jednostavne fizičke blokade i ograničenja do naprednih tehnika koje koriste kriptološke sisteme (šifrovanja) i biometrijske podatke.

10.5.1. Ograničenja fizičkog pristupa

Jedan od načina za smanjenje rizika od bezbednosnih proboja je osiguranje da samo ovlašćeno osoblje ima pristup računarskoj opremi. Organizacije koriste brojne proizvode i tehnike da identifikuju ovlašćeno osoblje. U zavisnosti od sistema bezbednosti pristup računaru može biti odobren na osnovu:

- Posedovanja dokumenta ili uređaja: ključ, ID kartica sa fotografijom ili smart kartica koja sadrži digitalno šifrovanu identifikaciju u ugrađenom čipu;
- Poznavanja i pamćenja – lozinka, ID broj, kombinacija ili deo lične istorije, kao što je npr. devojačko prezime majke;
- Načina vršenja neke radnje – potpis, brzina kucanja,
- Biometrijskih podataka: glas, otisak prsta, skeniranje mrežnjače, skeniranje osobina lica ili druga merenja pojedinih telesnih karakteristika.

Zbog toga što većina ovih bezbednosnih kontrola mogu da se kompromituju – ključevi mogu da se ukradu, potpisi da se falsifikuju, itd. – puno sistema koristi kombinaciju kontrola. Na primer, od zaposlenog može da se zahteva da pokaže bedž, otključa vrata ključem i otkuca lozinku za korišćenje zaštićenog računara.

U doba kada su mainframe računari bili izolovani u zasebnim prostorijama, fizički pristup je bio dovoljan za odbranu od napadača. Ali u modernim kancelarijama, računari i podaci su skoro svuda i mreže povezuju računare sa spoljnim svetom. U distribuiranom, mrežnom okruženju, bezbednost je mnogo problematičnija. Nije dovoljno ograničiti fizički pristup glavnim računarima kada personalni računari i mrežne konekcije nisu ograničene. Potrebne su dodatne bezbednosne mere.

10.5.2. Lozinke

Lozinke (*paswords*) su najčešći alat koji se koristi za ograničenje pristupa računarskim resursima. Lozinke su efektivne pod uslovom da su pažljivo izabrane. Većina korisnika računara bira lozinke koje je lako pogoditi: imena partnera, dece ili kućnih ljubimaca, reči vezane za poslove ili hobije i uzastopne karaktere na tastaturama. Mnogi bezbednosni sistemi odbijaju da omoguće korisnicima izbor bilo koje govorne reči ili imena kao lozinku tako da hakeri ne mogu koristiti softverske rečnike za njihovo sistematsko pogađanje. Čak i najbolje lozinke treba često menjati.

Softver za kontrolu pristupa ne treba da tretira sve korisnike na isti način. Mnogi sistemi koriste lozinke da ograniče korisnike tako da oni mogu da otvore samo fajlove vezane za njihov posao. U dosta slučajeva, korisnicima je dato pravo pristupa samo za čitanje fajlova tako da mogu da ih vide ali ne i da ih menjaju.

Preporučuje se da lozinke budu najmanje 16 karaktera dužine, gde se koriste mala i velika slova, kao i brojevi i interpukcijski znaci. Korišćenjem lozinki dužine 8 karaktera, bez brojeva i velikih slova je nebezbedan pristup pitanju bezbednosti.

10.5.3. Mrežne barijere

Krađa računarskih podataka se često ne obavlja u samom računaru. Moguće je da se presretnu poruke – računarski podaci koji se prenose računarskom mrežom. Na primer, lozinke su od male koristi za skrivanje email poruka. Štite korisnika od pristupa porukama na samom računaru, ali se mail-ovi prenose standardizovanim TCP/IP protokolima preko računarske mreže.

Vrlo često pristup Internetu je najznačajniji zahtev, a bezbednost je odmah iza ovog zahteva. Mnoge organizacije koriste mrežne logičke barijere (*firewalls*) za bezbednost internih mreža dok omogućavaju komunikaciju sa ostatkom Interneta. Tehnički detalji mrežnih barijera značajno variraju, a u njihovoj suštini je zaštita od neovlašćenog pristupa unutrašnjoj mreži. Efektivno, mrežna barijera se otvora samo za pakete informacija koji prolaze jednu ili više bezbednosnih provera. Mrežne barijere su namenjene kako velikim korporacijama tako i pojedinačnim korisnicima.

Logička barijera (*Firewall*) je poseban hardver ili softver koji u okviru računarske mreže ima mogućnost da spreči nepropisni ili neželjeni prenos podataka preko mreže, koji je zabranjen od strane sigurnosne politike postavljene na mreži. Komunikaciona efikasnost koju Internet omogućava je prouzrokovala masovno priključenje privatnih mreža direktno na Internet. Direktne Internet konekcije olakšavaju hakerima da eksploatišu privatne mrežne konekcije. Pre postojanja Interneta, jedini način koji je omogućavao hakerima da se povežu od kuće na privatnu mrežu bio je direktno biranje telefonskog broja modemom preko javne telefonske mreže. Pitanju bezbednosti daljinskog pristupa nije posvećivano mnogo pažnje.

Firewall se koristi za kreiranje kontrolnih tačaka bezbednosti (*chekpoints*) na granicama privatnih mreža. Na ovim kontrolnim tačkama firewall-i ispituju sve pakete koji prolaze između privatne mreže i Interneta, u zavisnosti od toga da li odgovaraju pravilima politike programirane na *firewall*-u. Ako je *firewall* propisno konfigurisan, u mogućnosti je da ispita svaki protokol kome je dozvoljen prolaz. Ovakav koncept graničnog obezbeđenja veoma je bitan - bez njega svaki računar (host) na privatnoj mreži morao bi sam da obavlja funkciju *firewall*-a. Time bi se bespotrebno trošili računarski resursi. *Firewall*-ovi omogućavaju centralizaciju svih bezbednosnih servisa na spoljnim mašinama koje su optimizovane i posvećene zadatku zaštite. Ispitivanje saobraćaja na graničnim mrežnim prolazima je takođe korisno u sprečavanju hakerisanja propusnog opsega na privatnoj tj. unutrašnjoj mreži.

Po prirodi, *firewall*-i kreiraju "uska grla" (*bottlenecks*) između unutrašnjih i spoljnih mreža. Razlog za to je što sav saobraćaj između ovih mreža mora proći kroz jednu tačku kontrole. Ovo je mala cena za bezbednost. S obzirom na to da su spoljne zakupljene linije relativno spore u poređenju sa brzinama modernih računara, zastoj prouzrokovan *firewall*-ima može biti zanemarljiv. Većini korisnika su relativno jeftini *firewall* uređaji više nego dovoljni da se povežu sa Internetom. Za poslovne potrebe i potrebe davaoca Internet usluga - ISP (*Internet Service Provider*), čiji je Internet saobraćaj na mnogo

višem nivou, razvijena je nova vrsta ekstremno brzih (skupih) *firewall*-a, koji su u mogućnosti da opsluže i najzahtevnije privatne mreže.

Firewall-i primarno funkcionišu koristeći tri osnovna metoda:

- Filtriranje paketa - odbacuje TCP/IP pakete neautentifikovanih hostova kao i pokušaje povezivanja na neautentifikovane servise.
- *Network Address Translation* (NAT) - prevodi IP adrese unutrašnjih hostova i tako ih skriva od spoljašnjeg praćenja. Ovaj metod se naziva i maskiranje IP adrese (IP *address masquerading*).
- *Proxy* servisi - uspostavljaju konekcije na visokim aplikativnim nivoima za unutrašnje host-ove u cilju da se kompletno prekine konekcija mrežnog sloja između unutrašnjih i spoljnih hostova.

Takođe, neki *firewall*-i obezbeđuju dodatne servise zasnovane na pretplati i nisu striktno povezani sa bezbednošću, ali će mnogi korisnici uvideti da su krajnje korisni:

- Skeniranje virusa - pretražuje dolazeće nizove podataka u potrazi za virusima. Ažuriranje servisa liste virusa zahteva pretplatu kod proizvođača *firewall*-a.
- Filtriranje prema sadržaju – mogućnost da se unutrašnjim korisnicima blokira pristup određenim tipovima sadržaja po kategorijama, kao na primer pornografiji, sadržaju koji propagira govor mržnje ili informacije o hakerisanju. Trenutno aktuelne liste blokiranih sadržaja takođe zahtevaju pretplatu.

10.5.4. Kriptografske tehnike – šifrovanje/dešifrovanje

Kriptografija je nauka koja se bavi metodama očuvanja tajnosti informacija. Osnovni elementi kriptografije su:

- Šifrovanje - postupak transformacije originalnih podataka (otvoren tekst) u šifrovani oblik za onoga kome taj tekst nije namenjen;
- *Dešifrovanje* - postupak vraćanja šifrata u originalne (otvorene) podatke;
- *Algoritam* – matematička transformacija – automat konačnih stanja, kojim se vrši šifrovanje/dešifrovanje;
- *Ključ* – definiše početne vrednosti algoritma kojim se vrši šifrovanje.

Šifrovanje tajnim ključem (simetrično šifrovanje) jeste šifarski sistem kod koga je ključ za šifrovanje identičan ključu za dešifrovanje. To znači da i pošiljalac i primalac poruke koriste isti tajni ključ. Ako pošiljalac želi da pošalje primaocu poruku (fakturu, račun, podatke o kreditnoj kartici) on će je šifrovati izabranim tajnim ključem. Kada primalac primi poruku može je dešifrovati samo ako poseduje kopiju tog ključa. Ako se pošiljalac i primalac nalaze na fizički udaljenim lokacijama javlja se problem obezbeđenja sigurnog kanala za distribuciju tajnog ključa. Svako ko na bilo koji način sazna njegovu vrednost mogao bi da čita i modifikuje sve poruke koje međusobno razmenjuju pošiljalac i primalac a da to ostane neprimećeno. Postoji još jedan problem, ako pošiljalac želi

da komunicira sa više poslovnih partnera mora da obezbedi različit ključ za svakog primaoca, kako bi se izbegla mogućnost da bilo koji primalac čita poruke koje mu nisu namenjene.

Rešenje ovih problema je pronađeno u vidu sistema šifrovanja javnim ključem (asimetrični šifarski sistem). U njemu svaki učesnik u komunikaciji koristi dva ključa. Jedan ključ je javni i može se slobodno distribuirati, dok je drugi privatni (tajni) i dostupan je samo njegovom vlasniku. Iako su različiti, ključevi su međusobno povezani matematičkim transformacijama. Poznavanje jednog ključa i algoritma transformacije ne omogućava dobijanje drugog ključa. Najbitnije je da se privatni ključ u celom postupku komunikacije nigde ne šalje jer ne postoji potreba da bilo ko sem njegovog vlasnika bude upoznat s njim. Šifrovanje se vrši primenom javnog ključa osobe kojoj se šalje informacija, a na prijemu ovakvu poruku može dešifrovati samo onaj ko poseduje upareni privatni ključ.

Obezbeđenje autentičnosti informacija tj. definisanje i provera identiteta pošiljaoca postiže se upotrebom digitalnih potpisa i digitalnih sertifikata. Svrha digitalnog potpisa je da potvrdi autentičnost sadržaja poruke (dokaz da poruka nije promenjena na putu od pošiljaoca do primaoca), kao i da obezbedi garantovanje identiteta pošiljaoca poruke. Osnovu digitalnog potpisa čini sadržaj same poruke. Pošiljalac primenom jednosmernih kriptografskih funkcija od svoje poruke koja je proizvoljne dužine stvara zapis fiksne dužine, koji u potpunosti odslikava sadržaj poruke. To praktično znači da svaka promena u sadržaju poruke dovodi do promene potpisa. Ovako dobijen zapis se dalje šifruje sopstvenim tajnim ključem i tako formira digitalni potpis koje se šalje zajedno sa porukom.

Ako se koriste sistemi šifrovanja sa javnim ključem neophodno je prvo utvditi javni ključ primaoca. Digitalnim sertifikatima se potvrđuje verodostojnost javnog ključa, tj. onemogućava se zamena javnog ključa na proizvoljan način. Oni se drugačije mogu nazvati digitalnom ličnom kartom, jer oni zaista to i jesu - digitalna lična karta u *cyber* prostoru, tj sredstvo kojim će se dokazati identitet na Internetu. Sertifikaciona tela, koja su 'treće strana' u komunikaciji, tj. strana od poverenja (*CA Certificate Authority*) imaju ulogu da na Internetu provere i utvrde nečiji identitet i nakon toga mu izdaju digitalni sertifikat.

10.5.5. Bekapi I druge bezbednosne mere

Čak i najjači bezbednosni sistem ne može da garantuje apsolutnu zaštitu podataka. Strujni udar ili nestanak struje može trenutno da obriše ili na neki drugi način da uništi čak i najpažljivije zaštićene podatke. Izvor neprekidnog napajanja (*uninterruptible power supply* - UPS) može da zaštiti računare od gubitka podataka usled nestanka struje. Zaštita od strujnog udara nije od pomoći kada nema struje, ali može da zaštiti elektronsku opremu od opasnih strujnih udara, sprečavajući skupe kvarove hardvera.

Katastrofalne situacije dolaze u mnogim oblicima. Sabotaža, ljudska greška, kvarovi mašine, požar, poplava, grom i zemljotresi mogu da oštete ili unište računarske podatke

zajedno sa hardverom. Svaki kompletan bezbednosni sistem treba da sadrži plan za oporavak od katastrofa. Najbolje i najviše korišćeno osiguranje podataka je preko rezervnih kopija (*backup*). Za mnoge sisteme, podaci i softver se bekapuju automatski na diskove ili trake, obično na kraju svakog radnog dana. Za maksimalnu bezbednost, mnogi korisnici računara čuvaju kopije poverljivih podataka na nekoliko različitih mesta. Tehnologija za smeštanje pod imenom RAID (redundatna grupa nezavisnih diskova) omogućava da višestruki hard diskovi rade kao jedna logička jedinica. RAID sistemi mogu, pored ostalog, automatski da prave duplikat podataka na višestrukim diskovima.

10.6. BEZBEDNOST, PRIVATNOST, SLOBODA I ETIKA

Računarska bezbednost je ljudski problem koji ne može samostalno da reši tehnologija. Bezbednost je pitanje menadžmenta, a dejstva i politike menadžera su kritične za uspeh bezbednosnog programa. Alarmantno veliki broj kompanija ne poklanja dovoljno pažnje računarskoj bezbednosti. Mnogi menadžeri ne razumeju probleme i ne misle da su ugroženi. Važno je da menadžeri shvate praktična, etička i pravna pitanja koja okružuju bezbednost. Menadžeri moraju da ukažu svojim zaposlenima na važnost bezbednosnih pitanja i bezbednosnih rizika.

Ponekada mere računarske bezbednosti mogu da stvore svoje probleme. Kompleksne procedure pristupa, programi za zaštitu od virusa, zakoni o intelektualnoj svojini i druge bezbednosne mere mogu, ako se uvode isuviše rigorozno, da ugroze ljude u obavljanju njihovog posla. U ekstremnim slučajevima, bezbednost može da ugrozi individualna čovekova prava.

Kao i u drugim segmentima života, računari ugrožavaju našu ličnu privatnost na nekoliko frontova. Korporacijske i vladine baze podataka skupljaju i dele ogromne količine informacija o nama protiv naše volje i bez našeg znanja. Programi za nadgledanje Interneta mogu da prate naše pretraživanje Web-a i da čitaju našu elektronsku poštu. Menadžeri mogu da koriste softver za nadgledanje rada, da mere produktivnost radnika i da posmatraju njihove aktivnosti na ekranu. Vladine bezbednosne agencije prate telefonske pozive i prenose podataka.

Bezbednosne mere koje se koriste za sprečavanje računarskog kriminala istovremeno pomažu u zaštiti prava na privatnost. Kada haker osvoji računarski sistem on može da prati privatnu komunikaciju legitimnih korisnika sistema. Ako neko spolja upadne u bazu podataka banke, ugrožena je privatnost svakog klijenta banke. Isto se odnosi i na vladine računare, računare u kreditnim biroima i ostale računare koji sadrže podatke o građanima. Bezbednost ovih sistema je važna za zaštitu privatnosti čoveka.

Aktivni bedž je jedan od najboljih primera nove tehnologije koja istovremeno može da poboljša bezbednost i ugrozi privatnost. Aktivni bedževi se koriste za identifikaciju, pronalaženje i podsećanje zaposlenih. Tehnologija aktivnog bedža ima potencijal da se iskoristi na razne načine, dobre i loše. Aktivni bedževi, nalik drugim bezbednosnim uređajima i tehnikama, postavljaju važna pravna i etička pitanja o privatnosti.

Vlade država reaguju na rastući problem računarskog kriminala donošenjem novih zakona protiv elektronskih napada. Otvaraju se pitanja o tome kako se građanska prava primenjuju u "elektronskoj oblasti." Da li se sloboda štampe odnosi na *on-line* magazine na isti način kao i za časopise? Da li uređivač elektronskog biltena ili Internet provajderi mogu da se smatraju odgovornim za informacije koje drugi postavljaju na server? Da li su Internet provajderi odgovorni kada njihovi korisnici ilegalno trguju muzikom on-line?

10.7. ZAKLJUČAK

Računari imaju rastuću ulogu u borbi protiv kriminala. Istovremeno, bezbednosne službe se suočavaju sa povećanjem računarskog kriminala. Većina računarskih zločina prođu neopaženi, a dosta onih koji su otkriveni često prođu neprijavljeni. Prema svim procenama računarski kriminal svake godine nanosi ogromne finansijske gubitke.

Računarski kriminalci koriste računare i računarsku opremu da ukradu robu, novac, informacije, softver i usluge. Drugi koriste trojanske konje, viruse, crve, logičke bombe i druge zlonamerne softvere da sabotiraju sisteme. Računarske zločine izvršavaju mladi, bistri računarski geniji zvani hakeri. Međutim, istraživanja pokazuju da su hakeri odgovorni samo za mali deo računarskih zločina. Tipičan računarski kriminalac je službenik od poverenja koji poznaje računarski sistem i ima lične ili finansijske probleme. Najčešći računarski kriminal, softversku pirateriju, izvršavaju milioni ljudi, često iz neobaveštenosti.

Zbog porasta računarskog kriminala i drugih rizika, organizacije su razvile brojne bezbednosne tehnike da zaštite svoje sisteme i podatke. Neki bezbednosni uređaji projektovani su da ograniče fizički pristup računarima. Međutim, ti alati postaju manje efikasni u doba personalnih računara i mreža. Lozinke, šifrovanje, postavljanje zaštite i softver za kontrolu se koriste da zaštite poverljive podatke u različitim organizacijama. Kada sve ostalo otkaže, sigurnosne kopije važnih podataka se koriste za rekonstrukciju sistema nakon pojave oštećenja.

Normalno, bezbednosne mere služe da zaštite našu privatnost i druga individualna prava. Ali, povremeno, bezbednosne procedure prete time pravima. Ustupci između računarske bezbednosti i slobode podižu važna pravna i etička pitanja.

Računarskim sistemima ne predstavljaju pretnju samo ljudi, njima takođe prete softverski bagovi i hardverski kvarovi.