



Univerza v Mariboru

Fakulteta za gradbeništvo

Računalništvo in informatika za inženirje

Andrej TIBAUT, Danijel REBOLJ

marec, 2013

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 1 od 219

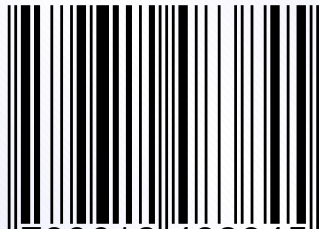
[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

ISBN 978-961-248-384-5



9 789612 483845 >

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor
004:62(075.8)

TIBAUT, Andrej, 1968- Računalništvo in informatika za inženirje
[Elektronski vir] / avtorja Andrej Tibaut, Danijel Rebolj. - 1. izd. - El.
učbenik. - Maribor : Fakulteta za gradbeništvo, 2013

Sistemske zahteve: osebni računalnik, bralnik za format PDF

ISBN 978-961-248-384-5

1. Rebolj, Danijel

COBISS.SI-ID 73834497

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



[Stran 2 od 219](#)

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Naslov: Računalništvo in informatika za inženirje

Avtorja: Andrej Tibaut, Danijel Rebolj

Vrsta publikacije: skripta za predavanja (e-publikacija)

Založnik: Univerza v Mariboru, Fakulteta za gradbeništvo, Maribor

Kraj založbe: Maribor

Datum izida: marec, 2013

Vrsta izdaje: 1. izdaja

Različica: R1 (20130325)

URL: <http://dkum.uni-mb.si>

Sistemske zahteve: osebni računalnik, bralnik za format PDF

Kazalo

1	Uvod	19
1.1	Motiv	20
1.2	Iz vsebine	21
2	Zgodovina računalništva in informatike	23
2.1	Prvi korak k mehanizaciji računanja	24
2.2	Prvi avtomatski mehanski računalnik	26
2.3	Prvi logični demonstrator	28
2.4	Prvi mehanski stroj na preluknjane kartice	30
2.5	Prvi elektromehanski stroj na preluknjane kartice	31
2.6	Prva avtomatska naprava za računanje aritmetičnih problemov	33
2.7	Prvi elektronsko-mehanski računalnik	34
2.8	Prvi elektronsko-digitalni računalnik	36
2.9	Prvi (resnični) hrošč	37
2.10	Prvi velik elektronski digitalni računalnik	38
2.11	Prvi pomnilnik	40

Spletna stran

Naslovnica

Kazalo



Stran 3 od 219

Nazaj

Full Screen

Zapri

Končaj

2.12	Prvi tranzistorski splošno namenski računalnik	41
2.13	Prvo integrirano vezje	42
2.14	Prvi računalniški program za interaktivno grafiko v realnem času	43
2.15	Prvi miniračunalnik	45
2.16	Še nekaj zgodovinskih zanimivosti	46
2.17	Generacije računalnikov	47
3	Računalništvo in informatika v inženirstvu	49
3.1	Računalništvo in informatika v gradbeništvu	50
3.2	Prihodnost gradbene informatike	55
4	Zgradba in delovanje digitalnih računalnikov	57
4.1	Računalniška aritmetika	59
4.1.1	Pretvorba desetiškega števila v dvojiški zapis	59
4.1.2	Pretvorba dvojiškega števila v desetiško	60
4.1.3	Seštevanje dvojiških števil	60
4.1.4	Odštevanje dvojiških števil	61
4.2	Merski sistemi v računalništvu - decimalni (SI) in dvojiški mnogo- kratniki	62
4.3	Predstavitev znakov	67
4.4	Osnovni model digitalnega računalnika	69
4.4.1	Delovanje von Neumannovega modela računalnika	70
4.4.2	Procesna enota	72
4.4.2.1	Aritmetične operacije	73
4.4.2.2	Logične operacije	74
4.4.3	Pomnilnik	75

Spletna stran

Naslovnica

Kazalo



Stran 4 od 219

Nazaj

Full Screen

Zapri

Končaj

4.4.4	Nadzorna enota	75
4.4.5	Vhodne in izhodne enote	76
4.5	Vrste računalniških sistemov	79
4.5.1	Namizni računalniški sistemi	79
4.5.2	Paralelni računalniški sistemi	79
4.5.3	Porazdeljeni računalniški sistemi	80
4.6	Moore-ov zakon	82
4.6.1	Diskusija o Moore-ovem zakonu	84
4.7	Izbrani primeri	85
4.7.1	Izvajanje programa v enoprosorskem računalniku	85
4.8	Več o zgradbi digitalnih računalnikov...	89
4.9	Izbirne naloge	90
5	Operacijski sistemi	91
5.1	Primer iz prakse	92
5.2	Umestitev operacijskega sistema v računalniški sistem	93
5.3	Definicija in cilji operacijskega sistema	96
5.4	Delitev operacijskih sistemov	97
5.4.1	Delitev OS glede na uporabniški vmesnik	97
5.4.2	Delitev OS glede na število centralnih procesnih enot, ki jih OS zmore izkoriščati	98
5.4.3	Delitev OS glede na število opravil, ki jih OS zmore podpirati “sočasno”	98
5.4.4	Delitev OS glede na število uporabnikov, ki jih OS podpira	98
5.4.5	OS za osebne računalnike	98
5.4.6	OS za strežnike in delovne postaje	99

Spletna stran

Naslovnica

Kazalo



Stran 5 od 219

Nazaj

Full Screen

Zapri

Končaj

5.4.7	OS za dlančne računalnike	101
5.5	Osnovni koncepti in funkcije sodobnih operacijskih sistemov	103
5.5.1	Podpora uporabnikom	104
5.5.2	Servisiranje programov	104
5.5.2.1	Preprosta paketna obdelava	105
5.5.2.2	Izboljšana paketna obdelava	107
5.5.2.3	Interaktivna obdelava	107
5.5.3	Upravljanje pomnilnika	110
5.5.3.1	Predpomnilnik	111
5.5.3.2	Navidezni pomnilnik	112
5.5.4	Upravljanje perifernih naprav	113
5.5.5	Upravljanje datotečnega sistema	114
5.6	Nekateri sodobni operacijski sistemi	121
5.6.1	Operacijski sistem GNU/Linux	121
5.6.1.1	Kako je nastal Linux?	122
5.6.1.2	Značilnosti Linuxa	126
5.7	Operacijski sistemi malo drugače	131
5.8	Izbirne naloge	132
6	Računalniške komunikacije	133
6.1	Osnovni koncepti računalniških komunikacij	135
6.1.1	Analogni in digitalni signali	135
6.1.2	Multipleksiranje signalov	136
6.2	Računalniška omrežja	139
6.2.1	Linjsko preklopna omrežja	139
6.2.2	Paketno preklopna omrežja	139

Spletna stran

Naslovnica

Kazalo



Stran 6 od 219

Nazaj

Full Screen

Zapri

Končaj

6.2.3	Struktura omrežij	140
6.2.3.1	LAN - lokalno omrežje	141
6.2.4	Parametri računalniškega omrežja	142
6.2.4.1	Hitrost komunikacije	142
6.2.5	Komunikacijski protokoli	144
6.2.6	Standardi za komunikacijo	144
6.2.6.1	ISO/OSI model	144
6.3	Medmrežje (internet)	145
6.3.1	Naslavljanje v medmrežju	145
6.3.2	Medmrežni servisi	145
6.3.3	Nekateri “omrežni” ukazi	145
6.4	Tehnologije za medmrežje	147
6.4.1	HTML	151
6.4.2	CSS	152
6.4.2.1	Zunanji (eksterni) slog	152
6.4.2.2	Notranji slog	154
6.4.2.3	Vrstični slog	155
6.5	Povzetek	157
6.6	Izbirne naloge	158
7	Računalniška grafika	159
7.1	Svetloba in barve	161
7.1.1	Elektromagnetni spekter	161
7.1.2	Predstavitev barv z računalniškimi napravami	164
7.1.2.1	Barvni model HSB	167
7.1.2.2	Barvni model RGB	168

Spletna stran

Naslovnica

Kazalo



Stran 7 od 219

Nazaj

Full Screen

Zapri

Končaj

7.1.2.3	Barvni model CMYK	169
7.2	Naprave za prikaz slike	173
7.2.1	Katodna cev	173
7.2.1.1	Rastrski prikaz	173
7.2.1.2	Vektorski prikaz	173
7.2.1.3	Ločljivost	173
7.2.2	Tiskalniki in risalniki	173
7.2.3	Vhodne interaktivne naprave	173
7.3	Rastrska in vektorska grafika	174
7.4	Nekateri uporabni programi	175
8	Napredna raba uporabniških programov	177
8.1	Umestitev uporabniške programske opreme	178
8.2	Kje se začne napredna raba uporabniških programov	180
8.3	Napredna raba urejevalnika besedil	183
8.3.1	Makro in Visual Basic za aplikacije	183
8.3.1.1	Izdelava makroja s snemanjem	183
8.3.1.2	Ročna izdelava makroja v Visual Basicu za aplikacije	188
8.3.1.3	Izdelava forme za vnos podatkov	193
8.3.1.4	Povezava urejevalnika in elektronske preglednice	193
8.4	Sistem za izdelavo dokumentov L ^A T _E X	199
8.4.1	L ^A T _E X in PDF za inženirsko rabo	203
8.5	Napredna raba elektronske preglednice	206
8.5.1	Uporaba funkcij v e-preglednici	207
8.5.1.1	Analiza stroškov gradbišča	207

Spletna stran

Naslovnica

Kazalo



Stran 8 od 219

Nazaj

Full Screen

Zapri

Končaj

8.5.1.2	Preoblikovanje seznama študentov	207
8.5.1.3	Izračun dopustne obtežbe temeljnih tal	208
8.5.2	Uporaba Visual Basica za aplikacije v elektronski preglednici	209
8.5.2.1	Izdelava študentske evidence	209

Bibliografija	217
----------------------	------------

Stvarno kazalo	219
-----------------------	------------

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Stran 9 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **10** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Slike

2.1	Napierjeve tablice	25
2.2	Pascaline	27
2.3	Logični demonstrator	29
2.4	Mehanska naprava krmiljena s preluknjanimi karticami	30
2.5	Elektromehanska naprava krmiljena s preluknjanimi karticami	32
2.6	Avtomska naprava za računanje aritmetičnih problemov	33
2.7	Z1 - elektromehanski računalnik	34
2.8	Konrad Zuse in njegov Z1	35
2.9	Elektronsko-digitalni računalnik	36
2.10	Prvi resnični bug	37
2.11	ENIAC	38
2.12	Pomnilnik s feritnimi obročki	40
2.13	TRADIC	41
2.14	Integrirano vezje	42
2.15	Interaktivna grafika	44
2.16	Miniračunalnik	45
2.17	Generacije	47

Spletna stran

Naslovnica

Kazalo



Stran 11 od 219

Nazaj

Full Screen

Zapri

Končaj

3.1	Otoki avtomatizacije v gradbeništvu (nekoč), [1]	53
3.2	Otoki avtomatizacije v gradbeništvu (danes), [1]	54
3.3	študentka FG oprtana z nosljivim računalnikom na gradbišču	56
4.1	ASCII tabela	68
4.2	koncept shranjenega programa – von Neumanov model računalnika	71
4.3	Logični binarni seštevalnik [2]	78
4.4	Simulacija prometa na porazdeljenem računalniku	81
5.1	OS – sistemska programska oprema	94
5.2	Umestitev OS v računalniški sistem	95
5.3	Windows	100
5.4	Linux	101
5.5	namizje Mac OS X v10.7 (Lion)	102
5.6	Ponavljjanje konceptov OS	103
5.7	Kontrolna struktura v pomnilniku pri paketni obdelavi	106
5.8	Preprosta in izboljšana paketna obdelava	108
5.9	Predpomnilnik	119
5.10	Navidezni pomnilnik	120
5.11	Richard Stallman - začetnik gibanja za zastonsko programsko opremo, vir	128
5.12	Linus Torvalds - začetnik Linuxa, vir	129
5.13	Maskota Linuxa, vir	130
6.1	Razlika med analognimi in digitalnimi signali	136
6.2	Vzorčenje analognih signalov	137
6.3	Multipleksiranje optičnih signalov	138

Spletna stran

Naslovnica

Kazalo



Stran 12 od 219

Nazaj

Full Screen

Zapri

Končaj

6.4	Primer lokalnega omrežja (LAN)	141
6.5	Komunikacijsko omrežje ARNES, 2003 [3]	142
6.6	Komunikacijsko omrežje ARNES, 2011 [4]	143
6.7	Komunikacijsko omrežje T-2, 2009 [5]	144
7.1	Elektromagnetni spekter	162
7.2	Elektromagnetni spekter	163
7.3	Slika zapisana v barvah modela RGB	164
7.4	Slika zapisana v barvah modela CMYK	165
7.5	Določitev barve z uporabo barvnega modela RGB ali HSV (HSL)	166
7.6	Določitev barve po RGB s programom Paintbrush (Mac OS X)	167
7.7	Program za pretvorbo RGB in CMYK	170
7.8	Barvna prostor HSB (HSV) ponazorjen s šeststrano piramido ([6])	171
7.9	Barvna modela RGB in CMYK ponazorjena s kocko	172
7.10	Primerjava RGB in CMYK	173
8.1	Uporabniška programska oprema	178
8.2	Ekran zapolnjen z večimi programi	181
8.3	Tematska organizacija diska	182
8.4	Delovno okolje Visual basica v Wordu	186
8.5	Dialogno okno, ki ga prikaže makro AutoOpen	189
8.6	Seznam makrojev v urejevalniku	190
8.7	Lažni virus	192
8.8	Forma v urejevalniku	194
8.9	Villa Kniggeweg v Berlinu, arhitekta K.Pogačar in A.Žižek	205
8.10	Obstoječi seznam študentov	215

Spletna stran

Naslovnica

Kazalo



Stran 13 od 219

Nazaj

Full Screen

Zapri

Končaj

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 14 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Tabele

4.1	Nekateri primeri desetiških števil v binarni obliki	63
4.2	Predlogi IEC za predpone binarnih mnogokratnikov	65
4.3	Vrste procesorjev	73
4.4	Tabelica za operacijo AND	74
4.5	Tabelica za operacijo OR	74
4.6	Tabelica za operacijo XOR	75
4.7	Primer pomnilnika	76
4.8	Cena 1MB pomnilnika (delovni pomnilnik in trdi disk) skozi leta	77
4.9	Moore-ov zakon in procesorji Intel	82
4.9	Moore-ov zakon in procesorji Intel	83
4.9	Moore-ov zakon in procesorji Intel	84
4.10	Binarne kode mnemonikov za operacije	86
4.11	Binarne kode registrov	86
4.12	Oblika zapisa ukaza za aritmetične operacije	86
4.13	Oblika zapisa za operacije za dostop do pomnilnika	87
4.14	Program v zbirnem in strojnem jeziku	87
4.15	Koraki izvajanja ukaza za seštevanje	88

Spletna stran

Naslovnica

Kazalo



Stran 15 od 219

Nazaj

Full Screen

Zapri

Končaj

5.1	Tipi zbirk	115
5.1	Tipi zbirk	116
6.1	Spletne tehnologije	148
6.1	Spletne tehnologije	149
6.1	Spletne tehnologije	150
8.1	šumniki in njihovi ekvivalenti v starih dokumentih	183
8.2	Seznam auto-makrojev	192

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 16 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Programi

6.1	Dokument CSS z definicijami slogov	152
6.2	Dokument HTML z zunanjim slogom CSS	153
6.3	Primer uporabe notranjega sloga	154
6.4	Vrstični slog	155
8.1	Makro za zamenjavo znakov v VBA	184
8.2	Izboljšan makro za zamenjavo znakov v VBA	185
8.3	Dialog v VBA	188
8.4	Dialog v VBA	189
8.5	Vrstični slog	190
8.6	Vrstični slog	191
8.7	Vrstični slog	191
8.8	Vrstični slog	191
8.9	Dialog v VBA	191
8.10	program v VBA za evidenco dela v urejevalniku MS Word	193
8.11	Dokument v zapisu \LaTeX	200
8.12	Obstoječa oblika seznama študentov	207
8.13	Zahtevana oblika seznama študentov	207

Spletna stran

Naslovnica

Kazalo



Stran 17 od 219

Nazaj

Full Screen

Zapri

Končaj

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 18 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Poglavje 1

Uvod

Vsebina pričujoče učne publikacije je namenjena bodočim inženirjem tehniških ved. Publikacija ima ambicijo postati svež in pregleden učbenik v elektronski obliki, ki bralca vabi in vzpodbuja k usvajanju in izpopolnjevanju znanja na področju računalništva in informatike na inženirskem področju. Dobro in praktično utrjeno poznavanje konceptov računalništva in informatike je ključno za strokovno rast in samozavest vsakega bodočega inženirja. Motiviran bralec bo hitro ugotovil, da računalništvo in informatika ni le “sestavljanje računalnikov”, “nameščanje programov”, “pošiljanje elektronske pošte” ali “risanje v Autocadu”, temveč veda, katerih temelji so zgrajeni iz zanimivih konceptov, ki izhajajo iz realnega sveta.

Ta učbenik je izdelan v sistemu \LaTeX in zapisan v Adobe PDF. ■

Spletna stran

Naslovnica

Kazalo



Stran 19 od 219

Nazaj

Full Screen

Zapri

Končaj

1.1. Motiv

Motivacija in cilji za bralca te publikacije naj bodo naslednji:

- razumeti ključne koncepte s področja računalništva in informatike,
- spoznati princip delovanja in potenciale računalnikov
- spoznati sodobna računalniška okolja in se jih naučiti učinkovito uporabljati,
- spoznati smisel napredne uporabe računalnikov na inženirskem področju.

Predpogoj za učinkovito razumevanje vsebine publikacije od bralca zahteva znanje uporabe računalnika, osnovne rabe pisarniških programov (urejevalnik, elektronske preglednice) in programov za komunikacijo.

Preden nadaljujete si odgovorite na naslednja vprašanja

1. Kako dobro poznate področje računalništva in informatike? (precej, malo, nič)
2. Ali čutite potrebo po boljšem poznavanju računalništva in informatike? (zelo, malo, nič)
3. Na katerih področjih RI bi želeli biti boljši? (operacijski sistemi, računalniške komunikacije, računalniška grafika, pisarniški programi, izdelava programov)
4. Ali napredna uporaba računalništva in informatike prinaša prednosti v inženirski praksi? (precej, malo, nič)

Če ste na prvi dve vprašanji odgovorili z *nič*, ste vzeli v roke napačno publikacijo... Za ostale je še dovolj upanja...kar pogumno naprej.

Spletna stran

Naslovnica

Kazalo



Stran 20 od 219

Nazaj

Full Screen

Zapri

Končaj

1.2. Iz vsebine

Sosledje poglavij pokriva najpomembnejša področja računalništva in informatike.

- Zgodovina računalništva in informatike
 - Kratek oris najpomembnejših zgodovinskih dejstev
 - Razvoj uporabe računalnikov na področju gradbeništva, prometa, arhitekture
- Zgradba digitalnih računalnikov
 - računalniška aritmetika, standardi za predstavitev znakov, model zgradbe računalnika
- Operacijski sistemi
 - obravnava vrste in značilnosti operacijskih sistemov
- Računalniške komunikacije
 - umestitev računalniških komunikacij, komunikacijski protokoli, najpomembnejši servisi, izdelava spletnih dokumentov
- Računalniška grafika
 - teorija barv, barvni modeli, grafični gradniki
- Napredna raba uporabniških programov

Spletna stran

Naslovnica

Kazalo



Stran 21 od 219

Nazaj

Full Screen

Zapri

Končaj

- predvsem napredna raba pisarniških programov: urejevalniki besedil, elektronske preglednice, predstavitvena orodja; izdelava makrojev, kratkih programov

Spletna stran

Naslovnica

Kazalo



Stran 22 od 219

Nazaj

Full Screen

Zapri

Končaj

Poglavje 2

Zgodovina računalništva in informatike

Za lažje razumevanje področij računalništva in informatike in njene vloge na področju gradbeništva, prometa in arhitekture je potrebno poznati nekatera zgodovinska dejstva. Vsebina podglavij, ki sledijo je povzeta po [7].

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **23** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.1. Prvi korak k mehanizaciji računanja

Napierjeve pomične tablice (angl. Napier's bones, John Napier, 1617, slika 2.1) za množenje vgravirane v slonovino predstavljajo prvi sodoben korak k mehanizaciji računanja.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 24 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

	I	2	3	4	5	6	7	8	9	0
2	2	4	6	8	10	12	14	16	18	0
3	3	6	9	12	15	18	21	24	27	0
4	4	8	12	16	20	24	28	32	36	0
5	5	10	15	20	25	30	35	40	45	0
6	6	12	18	24	30	36	42	48	54	0
7	7	14	21	28	35	42	49	56	63	0
8	8	16	24	32	40	48	56	64	72	0
9	9	18	27	36	45	54	63	72	81	0
0	0	0	0	0	0	0	0	0	0	0

Slika 2.1: Napierjeve tablice

Spletna stran

Naslovnica

Kazalo



Stran 25 od 219

Nazaj

Full Screen

Zapri

Končaj

2.2. Prvi avtomatski mehanski računalnik

Prvi avtomatski mehanski računalnik (Pascaline, Blaise Pascal, 1642, slika 2.2), ki ga je izumil Blaise Pascal, da bi svojemu očetu poenostavil obračunavanje davkov. Napis na tablici: Vsebuje mehanske sklope za računanje z 10 mestnimi števili, zadnji dve mesti predstavljata enoto za kovance, ostala predstavljajo enice, stotice, tisočice, itd. Mehanizem za računanje sestavljajo zobata kolesca, ki so nameščena po principu stožčastega kolesa. Naprava omogoča tudi prenose med številskimi mesti.

Več o tem: [1](#), [2](#).

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **26** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 2.2: *Pascaline*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **27** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.3. Prvi logični demonstrator

Prvi “logični demonstrator” (logic machine, Charles Mahon, 1777, slika 2.3), izum, s katerim je bilo možno rešiti enostavne numerične probleme v logični obliki. Napravo lahko smatramo kot predhodnico mehanizmov odločanja in logike v kasnejših računalnikih (npr. ekspertni sistemi). Kaj je ekspertni sistem? Za lažjo predstavbo si predstavljajmo zdravnika (eksperta), ki se na podlagi znanih pravil odloča o bolezenski diagnozi pacienta.

[Spletna stran](#)[Naslovnica](#)[Kazalo](#)[Stran 28 od 219](#)[Nazaj](#)[Full Screen](#)[Zapri](#)[Končaj](#)



Slika 2.3: *Logični demonstrator*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran **29** od **219**

[Nazaj](#)

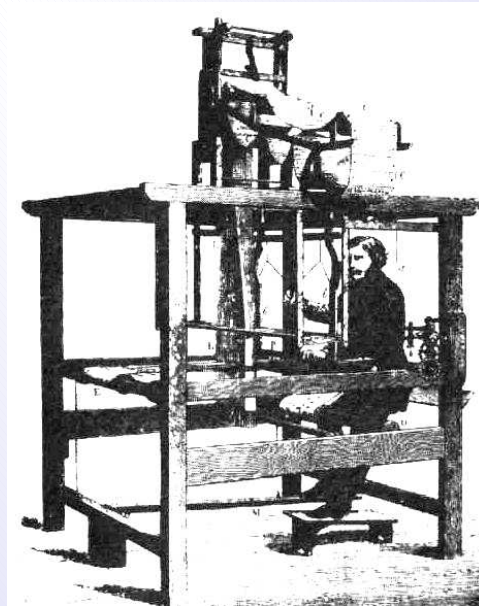
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.4. Prvi mehanski stroj na preluknjane kartice

Prvi mehanski stroj na preluknjane kartice (punched-card machine, Joseph-Marie Jacquard, 1804, slika 2.4), ki se je uporabljal v tekstilni industriji za krmiljenje pri tkanju svile.



Slika 2.4: Mehanska naprava krmiljena s preluknjanimi karticami

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran 30 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.5. Prvi elektromehanski stroj na preluknjane kartice

Prvi elektromehanski sistem na preluknjane kartice (dr. Herman Hollerith, 1886, slika 2.5). Naprava za štetje in naprava za sortiranje sta prvi primer uporabe kartic za razvrščanje. Hollerith je ustanovil firmo Tabulating Machines Company iz katere je kasneje nastal IBM.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 31 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 2.5: Elektromehanska naprava krmiljena s preluknjanimi karticami

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **32** od **219**

[Nazaj](#)

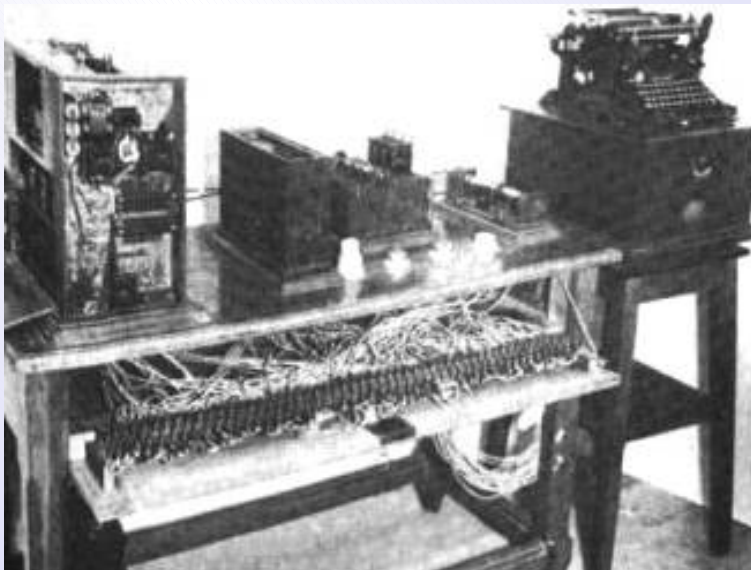
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.6. Prva avtomatska naprava za računanje aritmetičnih problemov

Prva avtomatska naprava za računanje aritmetičnih problemov (automatic calculating machine, Leonardo Torres y Quevedo, 1920, slika 2.6), ki je za vhod in izhod uporabljala avtomatski pisalni stroj. Torres je razvil tudi napravo za igranje šaha.



Slika 2.6: Avtomatska naprava za računanje aritmetičnih problemov

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **33** od **219**

[Nazaj](#)

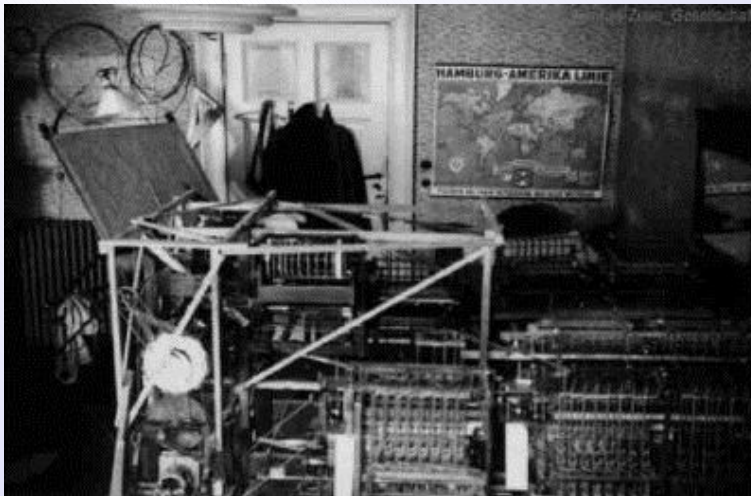
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.7. Prvi elektronsko-mehanski računalnik

Prvi elektronsko-mehanski digitalni računalnik Z1 je leta 1938 izumil **Konrad Zuse** (slika 2.7). Osnovni elementi računalnika Z1 so bili releji in vakuumske cevi. Računalnik Z1 je bil kasneje rekonstruiran (slika 2.8). Konrad Zuse je bil **gradbeni inženir**, ki je znan tudi po razvoju računalniških naprav za gradbeništvo (robotizirana gradnja).



Slika 2.7: Z1 - elektromehanski računalnik

Iz Z1 je kasneje nastal Z2, ter nato še Z3

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran 34 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 2.8: *Konrad Zuse in njegov Z1*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **35** od **219**

[Nazaj](#)

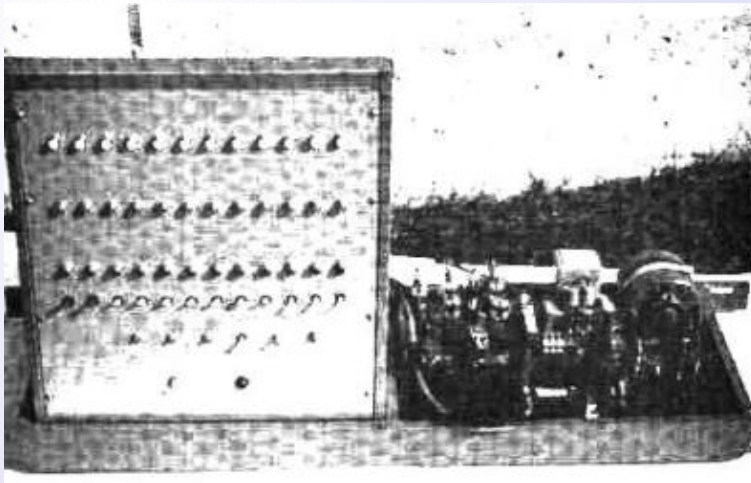
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.8. Prvi elektronsko-digitalni računalnik

Prvi elektronsko-digitalni računalnik (Dr. John V. Atanasoff, 1939, slika 2.9). Za logično vezje je uporabljal vakuumske cevi.



Slika 2.9: Elektronsko-digitalni računalnik

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 36 od 219

[Nazaj](#)

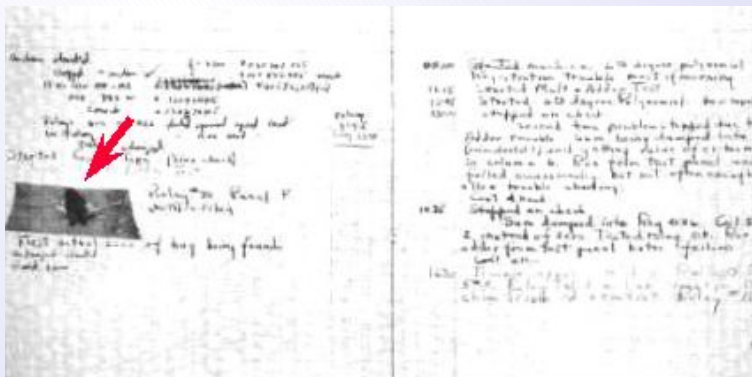
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.9. Prvi (resnični) hrošč

Prvi resnični hrošč (angl. bug), najden v računalniku Mark II (1945, slika 2.10), ki je prekinil delovanje računalnika. Izraz "bugža napako v sistemu je sicer znan že od prej (Edison), kasneje pa so nastale še izpeljanke, kot npr. "debugging"(slovenski izraz je razhroščevanje!) za odkrivanje napak v programu.



Slika 2.10: Prvi resnični bug

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 37 od 219

Nazaj

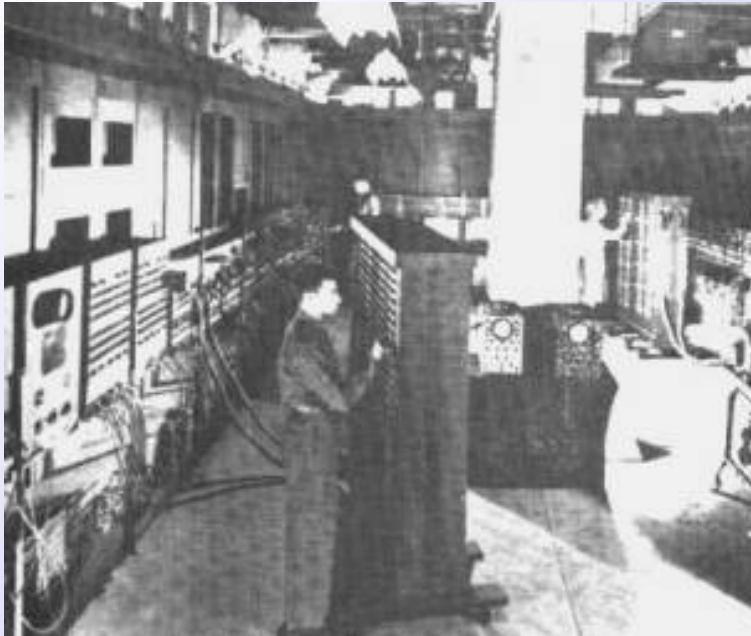
Full Screen

Zapri

Končaj

2.10. Prvi velik elektronski digitalni računalnik

Prvi velik elektronski digitalni računalnik (ENIAC - Electronic Numerical Integrator and Calculator, 1946, slika 2.11).



Slika 2.11: ENIAC

Značilnost elektronsko digitalnih računalniških sistemov v letih okoli 1950 je bilo upravljanje sistema s pomočjo konzole. To so bili *enouporabniški* sistemi, kjer

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran **38** od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

je bil operater hkrati uporabnik in programer. Preluknjane kartice so predstavljale programe in podatke za procesiranje (vhod), rezultate je tiskalnik izpisal na papir (izhod). Prvi programi na teh sistemih so bili zbirniki, prevajalniki, povezovalniki, nalagalniki in gonilniki naprav. Slaba lastnost teh sistemov je bila zelo majhna izkoriščenost centralne procesne enote zaradi dolgotrajnih nastavitvev sistema pred izvajanjem programov. Dobra lastnost teh sistemov je bila visoka stopnja varnosti (v vsakem trenutku je bil na sistemu največ eden uporabnik).

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **39** od **219**

[Nazaj](#)

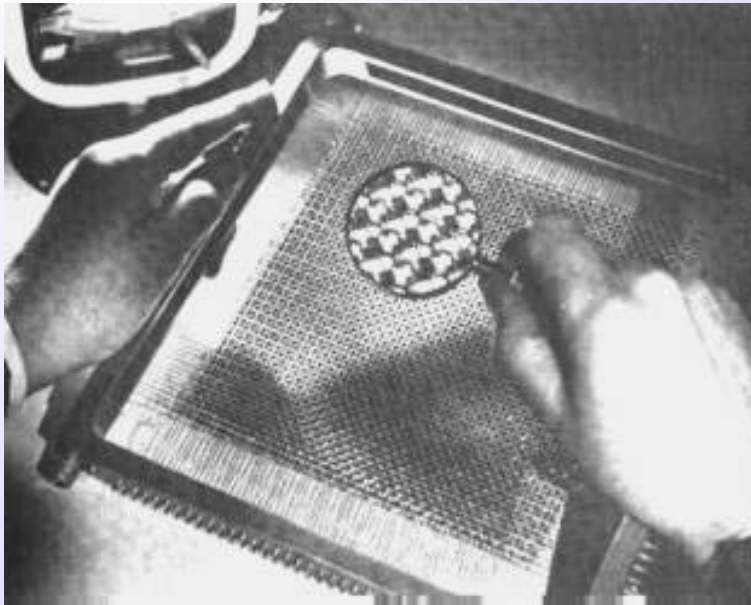
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.11. Prvi pomnilnik

Prvi pomnilnik s feritnimi obročki (core memory, Jay Forrester, 1949, slika 2.12), ki je odpravil pomanjkljivosti nezanesljivih vakuumskih cevi. Štiri leta kasneje so ga vgradili v računalnik Whirlwind.



Slika 2.12: Pomnilnik s feritnimi obročki

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran 40 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.12. Prvi tranzistorski splošno namenski računalnik

Prvi tranzistorski splošno-namenski računalnik (TRADIC - TRansistored Airborne Digital Computer, Bell Laboratories, 1954, slika 2.13) z 800 tranzistorji. Bil je tudi prvi računalnik, ki je uspešno deloval na letalu.



Slika 2.13: TRADIC

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Stran 41 od 219

[Nazaj](#)

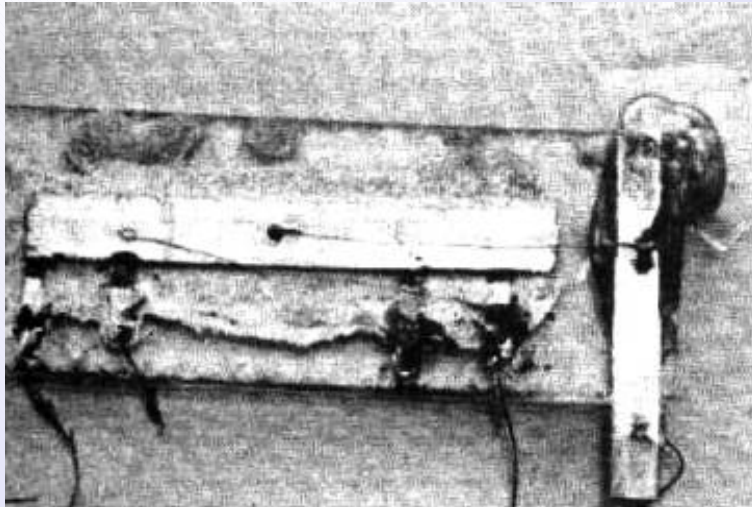
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.13. Prvo integrirano vezje

Prvo integrirano vezje (integrated circuit, Jack S. Kilby, Texas Instruments, 1958, slika 2.14). Osnovna ideja je bila izdelati vezje v enem koraku in iz enega materiala. Komponente so bile iz silicija.



Slika 2.14: Integrirano vezje

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 42 od 219

Nazaj

Full Screen

Zapri

Končaj

2.14. Prvi računalniški program za interaktivno grafiko v realnem času

Prvi program, ki je omogočal interaktivno grafično delo v realnem času (Interactive real time graphic, Ivan Sutherland, 1963, slika 2.15). Računalniški program Sketchpad je uporabniku omogočal interaktivno risanje na CRT (Cathode Ray-Tube) zaslon.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 43 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 2.15: *Interaktivna grafika*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



[Stran 44 od 219](#)

[Nazaj](#)

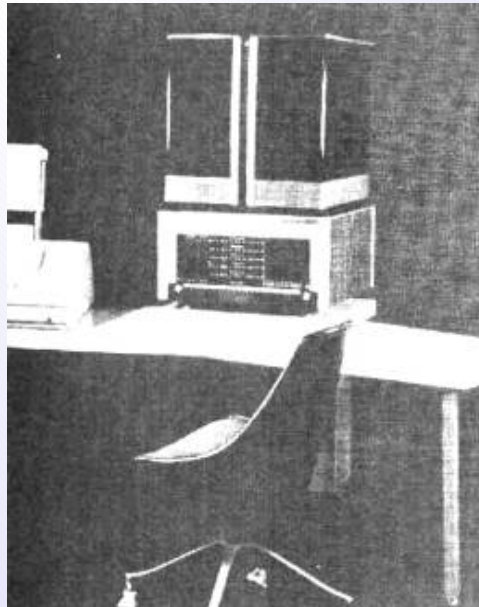
[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.15. Prvi miniračunalnik

Prvi miniračunalnik (minicomputer, DEC - Digital Equipment Corporation, 1963, slika 2.16). Izdelani so bili moduli z logičnim vezjem, ki so bili leta 1959 osnova za računalnik PDP-1 (12 bit). Leta 1963 so izdelali prvi komercialno uspešen miniračunalnik PDP-8.



Slika 2.16: Miniračunalnik

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 45 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.16. Še nekaj zgodovinskih zanimivosti

- Prvi komercialni superračunalnik CDC 6600 (CDC - Control Data Corporation, 1964). CDC 6600 je bil mnogo let najhitrejši in naj-zmogljivejši superračunalnik. Izraz “superračunalnik” je bil že nekeje let predtem uporabljen za računalnika IBM Stretch in Univac LARC.
- Prvi 16 bitni miniračunalnik Nova (Data General Corporation, 1969).
- Prvi mikroprocesorski čip imenovan 4004 (microprocessor chip, Intel Corporation, 1971). Ted Hoff je izdelal načrt za čip, ki bi imel logično in aritmetično vezje na enem čipu. V kombinaciji z ROM in RAM je čip predstavljal mikro-računalnik.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 46 od 219

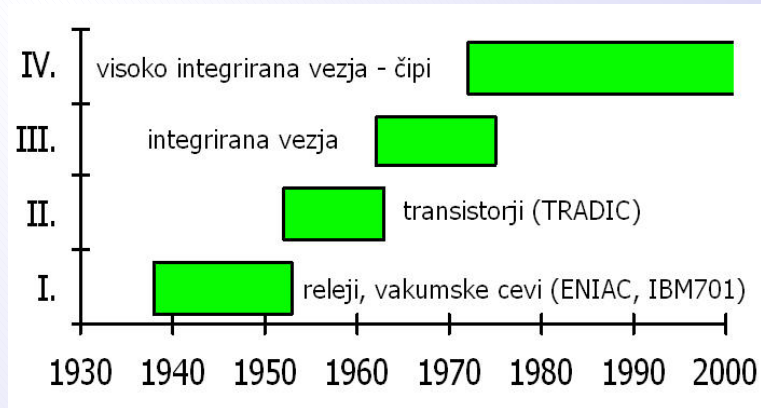
[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

2.17. Generacije računalnikov



Slika 2.17: Generacije

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

◀◀ ▶▶

◀ ▶

Stran 47 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **48** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Poglavje 3

Računalništvo in informatika v inženirstvu

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 49 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

3.1. Računalništvo in informatika v gradbeništvu

Računalništvo in informatiko na področju gradbeništvu poimenujmo **gradbena informatika** in jo definirajmo:

Gradbena informatika (angleško *information technology in construction - ITC, construction information technology*; nemško *Bauinformatik*) je znanstvena disciplina, ki izhaja iz gradbeništvu in ima za cilj sistematično in dosledno reševanje praktičnih in teoretičnih problemov v gradbeništvu s pomočjo metod in orodij računalništva in informatike.

Razvoj gradbene informatike se je prepletal z razvojem računalništva:

- Konrad Zuse (glej 2.7 - **Prvi elektronsko-mehanski računalnik**) je bil gradbeni inženir
- Steven Fenves (STRESS)
- Charles Eastman (BIM)

V Sloveniji so se začetki gradbene informatike izoblikovali na Univerzi v Ljubljani in Univerzi v Mariboru:

- Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo (1971): IKPIR
- Univerza v Mariboru, Fakulteta za gradbeništvo in Fakulteta za strojništvo (1979): CETES

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 50 od 219

Nazaj

Full Screen

Zapri

Končaj

- Univerza v Mariboru, Fakulteta za gradbeništvo (1981): LRG (Lutar),
- Univerza v Mariboru, Fakulteta za gradbeništvo (1995): Center za gradbeno informatiko
- Univerza v Mariboru, Fakulteta za gradbeništvo (2000): Katedra za gradbeno in prometno informatiko

Jedro raziskav gradbene informatike so informacijski modeli gradbenih objektov in procesov ter informacijski sistemi, ki omogočajo integrirano uporabo takšnih modelov ter s tem optimalnejšo uporabo razpoložljivih informacij v več fazah življenjskega cikla gradbenega produkta. Gradbene informatike torej ne enačimo le z uporabo računalnika v gradbeništvu, temveč gre za sistemsko uporabo informacijskih tehnologij v širšem kontekstu gradbenih procesov in produktov. Opis bi lahko strnili v naslednjo definicijo prof. dr. Žige Turka (v angleščini):

Construction information technology is equipment, applications, and services that are used by organisations to assist human communication, commitment negotiation, problem solving and decision making, and spans over several civil engineering disciplines.

Na področju izhajajo tudi priznane revije, utečene so znanstvene konference, mednarodni raziskovalni in razvojni projekti in druge aktivnosti, ki utemeljujejo področje gradbene informatike v Evropi in v svetu.

Pozitivni učinki razvoja gradbene informatike so se najprej pokazali na naslednjih področjih gradbeništva:

- 1955 - FORTRAN, programski jezik za razvoj inženirskih programov (npr. program za analizo konstrukcij)

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 51 od 219

Nazaj

Full Screen

Zapri

Končaj

- 1956 - COBOL, programski jezik za razvoj poslovnih programov (npr. program za finančne kalkulacije)
- 1970. - parametrično načrtovanje (primer: parametri za izračun notranjih statičnih količin prostoležečega nosilca z zvezno obtežbo: dolžina nosilca l , obtežba nosilca q , maksimalni moment $M_{max} = ?$), metoda končnih elementov (MKE, angl. FEM), metoda robnih elementov (redkeje), projektno upravljanje
- 1980. - risanje, CAD - CAE - CIM
- 1990. - integracija sistemov s standardi (STEP/EXPRESS-G - standard za izmenjavo modelov produktov, IFC, CAD = industrijski standard)
- 2000. - računalniški sistemi upravljajo z znanjem, mediacija med sistemi namesto integracije sistemov, programski agenti

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



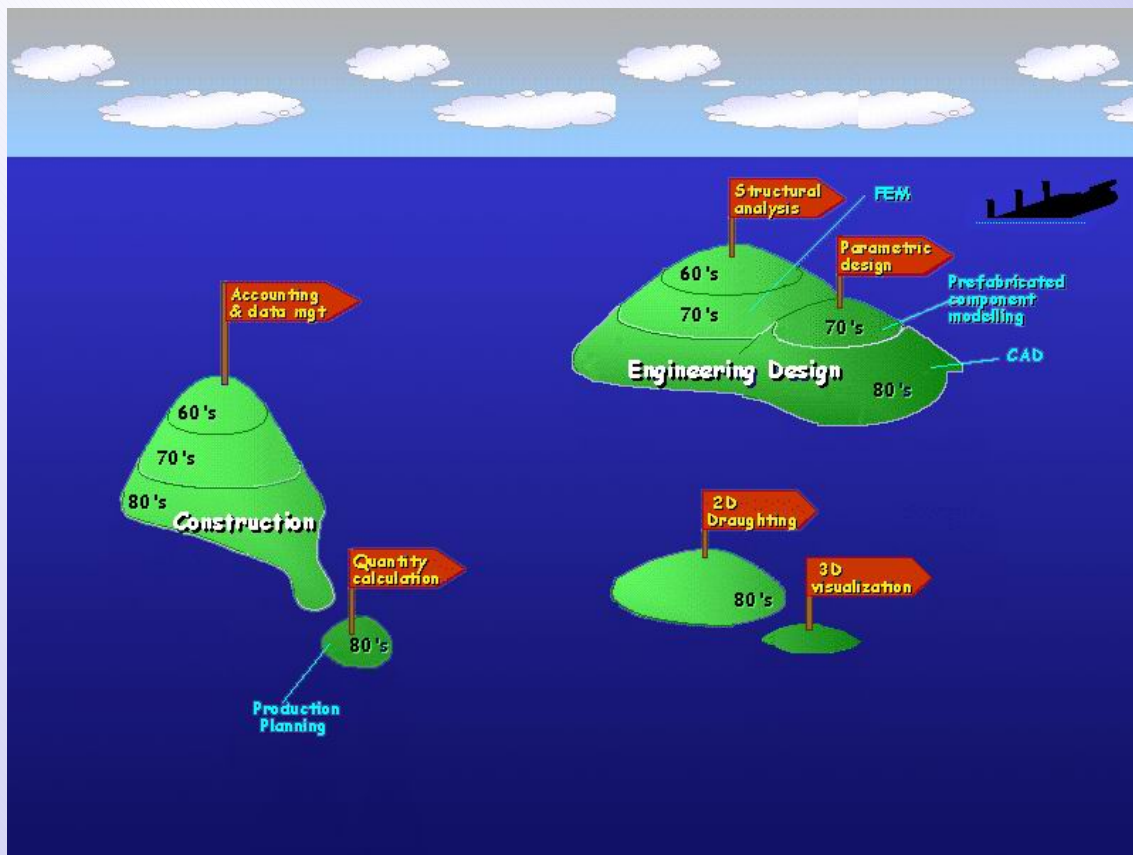
Stran 52 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 3.1: Otoki avtomatizacije v gradbeništvu (nekoč), [1]

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

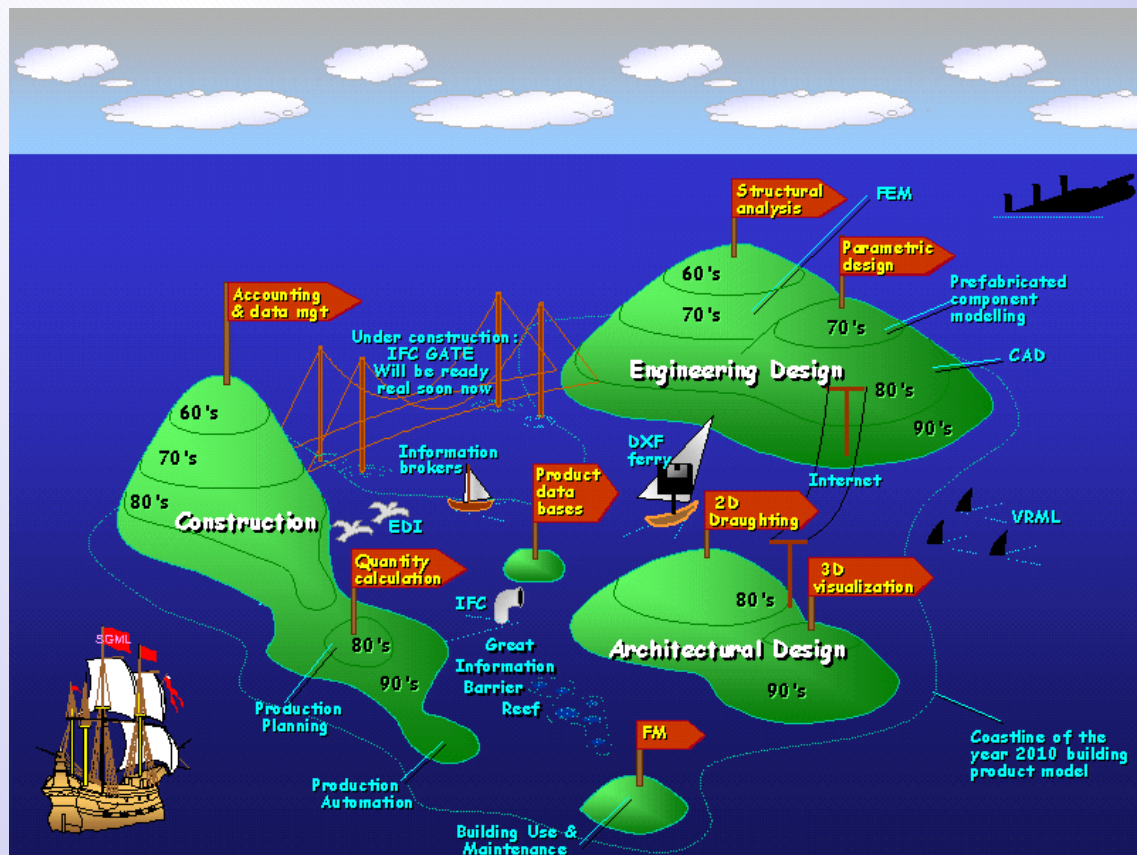
Stran 53 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 3.2: Otoki avtomatizacije v gradbeništvu (danes), [1]

Spletna stran

Naslovnica

Kazalo

Stran 54 od 219

Nazaj

Full Screen

Zapri

Končaj

3.2. Prihodnost gradbene informatike

Razvoj računalništva in informatike bo najbolj intenziven na naslednjih področjih:

- Brežžične komunikacije in multimedijски mobilni računalniki
- Modularnost, povezljivost in prilagodljivost programov
- Totalna integracija sistemov

To bo posledično vplivalo na naslednja področja uporabe v gradbeništvu:

- Popolna integracija dokumentacije
- Inteligentni gradbeni stroji
- Popolna komunikacijska vključitev vseh izvajalcev v realnem času (slika 3.3)
- Inteligentne zgradbe (konstrukcije, klimatizacijski sistemi, komunikacije)

Spletna stran

Naslovnica

Kazalo



Stran 55 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 3.3: študentka FG oprtana z nosljivim računalnikom na gradbišču

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **56** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Poglavje 4

Zgradba in delovanje digitalnih računalnikov

Področje zgradbe ali arhitekture računalnika (arhitektura računalnika zveni posebej všečno za arhitekto!) je sila zahtevno za učenje, količina informacij o novostih je preobsežna za uporabnika inženirja (npr. inženirja gradbeništva, prometa, strojništva). Razlogi so naslednji [8]:

- dinamična narava področja; razvojni cikli tehnologije so krajši od treh let,
- potrebno je poznati različna področja, elektronska vezja, računalniška aritmetika, zbirni jeziki, načrtovanje sistemov, diskretno matematiko, analiza zmogljivosti, itd.
- nivo abstrakcije se večja in zakriva podrobnosti.

[Spletna stran](#)[Naslovnica](#)[Kazalo](#)

Stran **57** od **219**

[Nazaj](#)[Full Screen](#)[Zapri](#)[Končaj](#)

Kaj storiti, da se bolje znajdemo? Posameznik mora razumeti osnovne koncepte delovanja računalnikov in znati uporabiti razpoložljive vire informacij, ko je to potrebno (npr. spletni viri, tiskana literatura, primerjalne študije, poznavanje glavnih proizvajalcev). Za inženirje, čigar domicil ni računalniška stroka, zadostuje spoznati in razumeti osnove zgradbe enoprocesorskih računalnikov.

- razumeti računalniško aritmetiko in standarde za predstavitev znakov
- razumeti von Neumannov model enoprocesorskega računalnika
- poznati osnovne komponente, ki sestavljajo današnje mikroračunalnike
- poznati parametre, ki vplivajo na zmogljivost računalniškega sistema

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

◀◀

▶▶

◀

▶

Stran **58** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

4.1. Računalniška aritmetika

Digitalne računalnike sestavljajo elektronski elementi, ki električno napetost digitalizirajo, t.j. pretvarjajo v eno od dveh stanj 0 in 1: vklop/izklop, +5V/-5V ustreza 1/0.

Operacije v digitalnih sistemih potekajo v dvojiškem (*binarnem*) številskem sestavu, ki ga sestavlja nabor dveh znakov (0 in 1). Binarna števila imajo svoje desetiške ekvivalente (nekaj primerov prikazuje tabela 4.1). Za računalništvo je pomemben še šestnajstniški (HEX) številski sistem s simboli 0, 1,...9, A, B, C, D, E, F in nekoliko manj tudi osmiški (OCT) številski sistem s simboli 0, 1,..., 7.

4.1.1. Pretvorba desetiškega števila v dvojiški zapis

Kakšen je postopek pretvorbe desetiškega števila 77 v dvojiški (binarni) zapis?

$$\begin{array}{r} 77 / 2 = 38 + 1 \\ \swarrow \\ 38 / 2 = 19 + 0 \\ 19 / 2 = 9 + 1 \\ 9 / 2 = 4 + 1 \\ 4 / 2 = 2 + 0 \\ 2 / 2 = 1 + 0 \\ 1 / 2 = 0 + 1 \uparrow \\ \hline 77_{(10)} = 1001101_{(2)} \\ \longrightarrow \end{array}$$

Spletna stran

Naslovnica

Kazalo



Stran 59 od 219

Nazaj

Full Screen

Zapri

Končaj

4.1.2. Pretvorba dvojiškega števila v desetiško

Kako binarno število 1100101 pretvorimo v desetiško?

$$\begin{array}{r} 1 * 2^0 = 1 \\ 0 * 2^1 = 0 \\ 1 * 2^2 = 4 \\ 0 * 2^3 = 0 \\ 0 * 2^4 = 0 \\ 1 * 2^5 = 32 \\ 1 * 2^6 = 64 \\ \hline 1 + 0 + 4 + 0 + 0 + 32 + 64 = 101_{(10)} \end{array}$$

4.1.3. Seštevanje dvojiških števil

Seštejmo dve binarni števili 1010 in 1111.

$$\begin{array}{r} 1010_{(2)} \\ + 1111_{(2)} \\ \hline 11001_{(2)} \end{array}$$

Pri seštevanju upoštevamo prenose čez 1. Primer: $1+1 = 10$ ($1+1=0$ in 1 prenesemo naprej), $1+1+1=11$. Kako bi sešteli dve osmiški, šestnajstiški, n-tiški števili?

Spletna stran

Naslovnica

Kazalo



Stran 60 od 219

Nazaj

Full Screen

Zapri

Končaj

4.1.4. Odštevanje dvojiških števil

Odštejmo binarno število 1010 od 1111, torej $1111 - 1010$. $1111 - 1010$ lahko pretvorimo v operacijo seštevanja $1111 + (-1010)$. Vendar računalniška aritmetika ne pozna predznaka »-«, ampak *dvojiški komplement*. Pokažimo celoten postopek odštevanja:

1. pretvorba 1010 v dvojiški komplement

$$\begin{array}{r} 1010 \\ \hline 0101 \text{ (eniški komplement, zamenjamo 0 in 1)} \\ + 0001 \text{ (prištejemo 1)} \\ \hline 0110 \text{ (dvojiški komplement)} \end{array}$$

2. operacija seštevanja $1111 + 0110$

$$\begin{array}{r} 1111 \\ + 0110 \text{ (prištejemo dvojiški komplement)} \\ \hline 10101 \end{array}$$

3. ignoriramo prvi bit z leve

$$\begin{array}{r} (1)0101 \text{ (ignoriramo prvi bit z leve)} \\ \hline 0101 \text{ (rezultat odštevanja)} \end{array}$$

Računalniško aritmetiko lahko dodatno utrdimo s pomočjo nekaterih (izobraževalnih programov).

Spletna stran

Naslovnica

Kazalo



Stran 61 od 219


Nazaj

Full Screen

Zapri

Končaj

4.2. Merski sistemi v računalništvu - decimalni (SI) in dvojiški mnogokratniki

V dvojiškem sistemu posameznemu znaku pravimo *bit* (angl. BInary digiT). Bit je tudi najmanjša enota za mero količine podatkov v računalništvu. Osem bitov predstavlja *besedo* ali *bajt* (angl. byte) ). Računalniški inženirji v praksi najpogosteje izražajo »prostornino« (kapaciteto, velikost) pomnilnika (npr. delovni pomnilnik) kot mnogokratnik byte-a:

- 1 bit (b) (0 ali 1)
- 1 Byte (B) = 8 bitov
- 1 KiloByte (KB) = 1024 b = 2^{10} B
- 1 MegaByte (MB) = 1024 KB = 2^{20} B
- 1 GigaByte (GB) = 1024 MB = 2^{30} B
- 1 TeraByte (TB) = 1024 GB = 2^{40} B
- 1 PetaByte (PB) = 1024 TB = 2^{50} B
- 1 ExaByte (EB) = 1024 PB = 2^{60} B
- 1 ZettaByte (ZB) = 1024 EB = 2^{70} B
- 1 YottaByte (YB) = 1024 ZB = 2^{80} B

Spletna stran

Naslovnica

Kazalo



Stran 62 od 219

Nazaj

Full Screen

Zapri

Končaj

desetiški številski sestav	dvojiški številski sestav
1	1
2	10
3	11
4	100
5	101
10	1010
512	1000000000
1024	10000000000
65535	1111111111111111

Tabela 4.1: Nekateri primeri desetiških števil v binarni obliki

- ☺ 1 BogračByte (BB) = 1024 YB = 2^{90} B ☺

Pogosta napaka je zamenjava bit in Byte, npr. 1KB ni enako 1Kb.

Proizvajalci ostalih pomnilniških naprav (diski) »presenetljivo« večinoma uporabljajo decimalne predpone, torej merske predpone mednarodnega standarda SI:

- 1 bit (b) (0 ali 1)
- 1 Byte (B) = 8 bitov
- 1 KiloByte (KB) = 1000 B = 1000^1 B
- 1 MegaByte (MB) = 1000 KB = 1000^2 B
- 1 GigaByte (GB) = 1000 MB = 1000^3 B

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 63 od 219

Nazaj

Full Screen

Zapri

Končaj

- 1 TeraByte (TB) = 1000 GB = 1000^4 B
- 1 PetaByte (PB) = 1000 TB = 1000^5 B
- 1 ExaByte (EB) = 1000 EB = 1000^6 B
- 1 ZettaByte (EB) = 1000 ZB = 1000^7 B
- 1 YottaByte (EB) = 1000 YB = 1000^8 B


Primer: proizvajalec diska navede podatek o velikost diska 137 GB in pri tem uporabi decimalni mnogokratnik: $137 * 10^9$ B. Če uporabimo dvojiški mnogokratnik je dejanska velikost diska ~ 128 GB: $137000000 \div 2^{30} \sim 128$.

Da bi bila zmeda popolna navedimo, da načrtovalci računalniških omrežij uporabljajo enoto KiloBit/s (1024 bitov/s) ali MegaBit (1048576 bit/s) za merjenje hitrosti prenosa (pasovne širine), medtem ko telekomunikacijski inženirji z MegaBit označujejo 10^6 bitov/s.

Odkod takšna (ne)doslednost...?

Razlogi so predvsem zgodovinski, saj so mnogokratnik 2^{10} uporabljali izključno računalniški inženirji. Vendar računalništvo kmalu ni bilo več samo v domeni računalničarjev, ampak tudi ostalih inženirjev (gradbeni inženirji, strojni inženirji, fiziki,...), ki pa so »govorili« v SI predponah po katerem mnogokratnik *kilo* pomeni 1000.

Ali obstaja rešitev?

Decembra 1998 je International Electrotechnical Commission (IEC) potrdila predloge drugačnih predpon za mnogokratnike na področju procesiranja in prenosa podatkov (tabela 4.2, ). Predlog čaka, da ga potrdi ustanova za standardizacijo.

- 1 kibibit = 1 Kibit = 2^{10} bitov = 1024 bitov

Spletna stran

Naslovnica

Kazalo



Stran 64 od 219

Nazaj

Full Screen

Zapri

Končaj

Mnogokratnik	Predpona	Znak	Opis
1024^1	kibi	KiB	Kilobinary
1024^2	mebi	MiB	Megabinary
1024^3	gibi	GiB	Gigabinary
1024^4	tebi	TiB	Terabinary
1024^5	pebi	PiB	Petabinary
1024^6	exbi	EiB	Exabinary
1024^7	zebi	ZiB	Zettabinary
1024^8	yobi	YiB	Yottabinary

Tabela 4.2: Predlogi IEC za predpone binarnih mnogokratnikov

- 1 kilobit = 1 Kbit = 10^3 bitov = 1000 bitov
- 1 mebibyte = 1 MiB = 2^{20} B = 1.048.576 B
- 1 megabyte = 1 MB = 10^6 B = 1.000.000 B
- 1 gibibyte = 1 GiB = 2^{30} B = 1.073.741.824 B
- 1 gigabyte = 1 GB = 10^9 B = 1.000.000.000 B

Predlogi IEC čakajo na potrditev mednarodnega urada za standardizacijo. Binarno lahko zapišemo tudi vse ostale podatke, ki jih lahko predstavimo z računalnikom:

- znake: npr. črka A = 1000001 (65 desetiško, glej tabelo ASCII),

Spletna stran

Naslovnica

Kazalo



Stran 65 od 219

Nazaj

Full Screen

Zapri

Končaj

- slike: npr. 1 pika (angl. pixel) na ekranu lahko predstavimo z osmimi biti (torej eno od 256 možnih vrednosti, vsaka vrednost predstavlja svojo barvo),
- zvok: npr. CD zapis, DVD zapis, MP3 zapis, DivX zapis.

Spletna stran

Naslovnica

Kazalo



Stran 66 od 219

Nazaj

Full Screen

Zapri



Končaj

4.3. Predstavitev znakov

Ste se kdaj vprašali kako so v računalniku predstavljeni *znaki*; števila, črke in drugi simboli (npr. »+«,»-«,»*«,»!«, itd.)? Za predstavitev znakov v računalniških programih uporabljamo *kodne tabele*. Kodna tabela vsebuje preslikavo med znakom in njegovo računalniško predstavitevjo. Računalniška predstavitev je izražena z numerično *kodo* v dvojiškem (osmiškem, šestnajstiškem) zapisu. Koda je *enoličen* zapis znaka. Kaj pomeni enoličen? Opišimo enoličnost z negacijo: ni mogoče, da bi dva različna znaka imela enako kodo. Prva kodna tabela z imenom *ASCII* (=American Standard Code for Information Interchange) je vsebovala kode za 128 znakov (male in velike črke angleške abecede, arabske številke in nekateri drugi kontrolni znaki):

- črke abecede: »A« (»a«),...,»Z« (»z«)
- številke: »0«,...,»9«
- druge znake: »@«, »#«, »€«,
- kontrolne znake: »RET« (tipka ENTER), »SPACE« (preslednica), »ESC« (tipka za prekinitev)

Za predstavitev vseh znakov zadostuje 7 bitov (2^7). Izhajajoč iz tega lahko ugotovimo, da bi bilo možno z osmimi biti (oz. bajtom) predstaviti 256 različnih znakov (2^8). Kode v tabeli so minimalni skupni imenovalci za izmenjavo besedil za vse današnje računalnike, v urejevalnikih besedil boste ASCII zapis spoznali kot *raw text*, *plain text*, *ASCII text*, *text only*.

Problem posebnih znakov »vseh« svetovnih pisav (npr. šumniki, prevoji, kitajske pisanke, idr.) je rešen z nadgradnjami ASCII tabele: kodne tabele *UNICODE* , *UTF-8*. Kodo znaka po *UNICODE* lahko poiščete z iskalnikom .

Spletna stran

Naslovnica

Kazalo



Stran 67 od 219

Nazaj

Full Screen

Zapri

Končaj

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	€#32;	Space	64	40	100	€#64;	@	96	60	140	€#96;	`
1	1	001	SOH (start of heading)	33	21	041	€#33;	!	65	41	101	€#65;	A	97	61	141	€#97;	a
2	2	002	STX (start of text)	34	22	042	€#34;	"	66	42	102	€#66;	B	98	62	142	€#98;	b
3	3	003	ETX (end of text)	35	23	043	€#35;	#	67	43	103	€#67;	C	99	63	143	€#99;	c
4	4	004	EOT (end of transmission)	36	24	044	€#36;	\$	68	44	104	€#68;	D	100	64	144	€#100;	d
5	5	005	ENQ (enquiry)	37	25	045	€#37;	%	69	45	105	€#69;	E	101	65	145	€#101;	e
6	6	006	ACK (acknowledge)	38	26	046	€#38;	&	70	46	106	€#70;	F	102	66	146	€#102;	f
7	7	007	BEL (bell)	39	27	047	€#39;	'	71	47	107	€#71;	G	103	67	147	€#103;	g
8	8	010	BS (backspace)	40	28	050	€#40;	(72	48	110	€#72;	H	104	68	150	€#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	€#41;)	73	49	111	€#73;	I	105	69	151	€#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	€#42;	*	74	4A	112	€#74;	J	106	6A	152	€#106;	j
11	B	013	VT (vertical tab)	43	2B	053	€#43;	+	75	4B	113	€#75;	K	107	6B	153	€#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	€#44;	,	76	4C	114	€#76;	L	108	6C	154	€#108;	l
13	D	015	CR (carriage return)	45	2D	055	€#45;	-	77	4D	115	€#77;	M	109	6D	155	€#109;	m
14	E	016	SO (shift out)	46	2E	056	€#46;	.	78	4E	116	€#78;	N	110	6E	156	€#110;	n
15	F	017	SI (shift in)	47	2F	057	€#47;	/	79	4F	117	€#79;	O	111	6F	157	€#111;	o
16	10	020	DLE (data link escape)	48	30	060	€#48;	0	80	50	120	€#80;	P	112	70	160	€#112;	p
17	11	021	DC1 (device control 1)	49	31	061	€#49;	1	81	51	121	€#81;	Q	113	71	161	€#113;	q
18	12	022	DC2 (device control 2)	50	32	062	€#50;	2	82	52	122	€#82;	R	114	72	162	€#114;	r
19	13	023	DC3 (device control 3)	51	33	063	€#51;	3	83	53	123	€#83;	S	115	73	163	€#115;	s
20	14	024	DC4 (device control 4)	52	34	064	€#52;	4	84	54	124	€#84;	T	116	74	164	€#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	€#53;	5	85	55	125	€#85;	U	117	75	165	€#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	€#54;	6	86	56	126	€#86;	V	118	76	166	€#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	€#55;	7	87	57	127	€#87;	W	119	77	167	€#119;	w
24	18	030	CAN (cancel)	56	38	070	€#56;	8	88	58	130	€#88;	X	120	78	170	€#120;	x
25	19	031	EM (end of medium)	57	39	071	€#57;	9	89	59	131	€#89;	Y	121	79	171	€#121;	y
26	1A	032	SUB (substitute)	58	3A	072	€#58;	:	90	5A	132	€#90;	Z	122	7A	172	€#122;	z
27	1B	033	ESC (escape)	59	3B	073	€#59;	;	91	5B	133	€#91;	[123	7B	173	€#123;	{
28	1C	034	FS (file separator)	60	3C	074	€#60;	<	92	5C	134	€#92;	\	124	7C	174	€#124;	
29	1D	035	GS (group separator)	61	3D	075	€#61;	=	93	5D	135	€#93;]	125	7D	175	€#125;	}
30	1E	036	RS (record separator)	62	3E	076	€#62;	>	94	5E	136	€#94;	^	126	7E	176	€#126;	~
31	1F	037	US (unit separator)	63	3F	077	€#63;	?	95	5F	137	€#95;	_	127	7F	177	€#127;	DEL

Source: www.LookupTables.com

Slika 4.1: ASCII tabela

Spletna stran

Naslovnica

Kazalo

◀ ▶

◀ ▶

Stran 68 od 219

Nazaj

Full Screen

Zapri

Končaj

4.4. Osnovni model digitalnega računalnika

Sredi 20. stoletja so se matematiki ukvarjali z idejo, da bi spisek navodil za računalniško obdelavo (računalniški program) »shranili v računalniku«. Iz poglavja 2 - **Zgodovina računalništva in informatike** se spomnimo kako in kje je bil shranjen program (oz. navodilo za izvajanje operacij računalnika):

- Statve krmiljene s preluknjanimi karticami (1804): program \approx preluknjane kartice,
- ENIAC (1943): program \approx nastavljena kombinacija stikal,
- EDVAC (1944): program \approx zaporedje ukazov shranjeno v pomnilniku

John von Neumann (kakor tudi **Konrad Zuse** v tistem obdobju) je leta 1945 razvil idejo, da bi *računalniški program in podatki lahko bili shranjeni v pomnilniku* (koncept shranjenega programa - von Neumannov model računalnika). Predlagal je naslednjo osnovno zgradbo računalnika (slika 4.2):

- *pomnilnik* za podatke in programe
- *centralno procesno enoto* za izvajanje aritmetičnih in logičnih operacij
- *nadzorno enoto* za interpretacijo ukazov
- *vhodne naprave* (tipkovnica, miška, idr.)
- *izhodne naprave* (ekran, tiskalnik, disk, idr.)

Spletna stran

Naslovnica

Kazalo



Stran 69 od 219

Nazaj

Full Screen

Zapri

Končaj

Z definicijo modela digitalnega računalnika si je von Neumann prislužil sloves »očeta modernega računalnika«. Nicholas Kaldor je o njem izjavil:

»Bil je nedvomno najbližje geniju od vsega na kar sem v življenju nalezel«.

4.4.1. Delovanje von Neumannovega modela računalnika

Delovanje dandanašnjih (digitalnih) računalnikov je še vedno skladno z izvajanjem računalniškega programa kot si ga je zamislil von Neumann. Osnovno delovanje von Neumannovega modela računalnika lahko opišemo na sledeč način:

1. Zagon programa

- Ob zagonu se program (npr. urejevalnik besedil) naloži iz vhodne naprave (npr. disk) v pomnilnik.
- Analogija: Iz knjižne police vzamemo knjigo in jo položimo na delovno mizo.

2. Izvajanje programa: pridobivanje podatka

- CPE zahteva podatek na določenem naslovu pomnilnika, ki je zapisan v programskem števcu nadzorne enote.
- Analogija: Poiščemo določeno stran v knjigi.

3. Izvajanje programa: obdelava podatka

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 70 od 219

Nazaj

Full Screen

Zapri

Končaj

Spletna stran

Naslovnica

Kazalo



Stran 71 od 219

Nazaj

Full Screen

Zapri


Končaj

Slika 4.2: koncept shranjenega programa – von Neumanov model računalnika

- Procesor obdela podatek, ki ga je pridobil iz pomnilnika. Programski števec se poveča.
- Analogija: Preberemo stran in določimo naslednjo stran.

4. Program zaključi z delovanjem

- Podatki se iz pomnilnika zapišejo na izhodno napravo (npr. disk)
- Analogija: Knjigo smo prebrali, zato jo odložimo nazaj na knjižno police.

Animiran prikaz [9] poenostavlja razumevanje zgoraj opisanega delovanja von Neumannovega modela računalnika . V nadaljevanju je opisan pomen in vloga posameznih komponent von Neumannovega modela digitalnega računalnika.

4.4.2. Procesna enota

Centralna procesna enota (CPE, obdelovalna enota, procesor, angl. CPU - central processing unit) izvaja operacije, ki jih zahtevajo ukazi zapisani v programu. Specifičen del procesne enote je *aritmetično-logična enota*, ki izvaja dva tipa *binarnih operacij*:

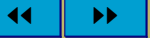
- aritmetične operacije
- logične ali Boolove operacije

Značilnosti najpogostejših procesorjev so opisani v tabeli **4.3**.

Spletna stran

Naslovnica

Kazalo



Stran 72 od 219

Nazaj

Full Screen

Zapri

Končaj

Tabela 4.3: Vrste procesorjev

Oznaka procesorja	Opis
Core Duo	Intelov procesor z dvema procesorskima jedroma, ki deluje v 32-bitnem načinu
Core 2 Duo	Intelov procesor z dvema procesorskima jedroma, ki deluje v 32-bitnem in 64-bitnem načinu
Intel Quad-Core Xeon	Intelov procesor s štirimi procesorskimi jedri, ki delujejo v 64-bitnem načinu
Intel Core i3	Intelov procesor, ki deluje v 64-bitnem načinu
Intel Core i5	Intelov procesor, ki deluje v 64-bitnem načinu
Intel Core i7	Intelov procesor, ki deluje v 64-bitnem načinu

4.4.2.1. Aritmetične operacije

Primer aritmetične operacije je seštevanje (glej **seštevanje binarnih števil**).

Spletna stran

Naslovnica

Kazalo



Stran 73 od 219

Nazaj

Full Screen

Zapri

Končaj

4.4.2.2. Logične operacije

operacija NOT — negacija

- NOT 1 = 0
- NOT 0 = 1

operacija AND — logično seštevanje

Tabela 4.4 prikazuje pravilo operacije AND.

AND	0	1
0	0	0
1	0	1

Tabela 4.4: Tabela za operacijo AND

Operacija OR — logično množenje

Tabela 4.5 prikazuje pravilo operacije OR.

OR	0	1
0	0	1
1	1	1

Tabela 4.5: Tabela za operacijo OR

operacija XOR — ekskluzivni ALI

Tabela 4.6 prikazuje pravilo operacije XOR.

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 74 od 219

Nazaj

Full Screen

Zapri

Končaj

XOR	0	1
0	0	1
1	1	0

Tabela 4.6: Tabela za operacijo XOR

Uporabnost logičnih operatorjev lahko demonstriramo s pomočjo binarnega seštevalnika (slika 4.3), ki sešteje dve binarni števili. Seštevalnik deluje kot avtomat čigar logiko tvorijo operatorji *AND*, *OR* in *XOR*. Seštevalnik lahko uporabimo za seštevanje dveh štiribitnih vrednosti, lahko pa ga poljubno razširimo.

4.4.3. Pomnilnik

Pomnilnik (tudi *delovni pomnilnik*, angl. RAM - Random Access Memory) je matrika velikosti $k * m$ (k = število vrstic, običajno 2^n , m = število stolpcev, običajno 8 bitov), ki predstavlja zaporedje bitov grupiranih v besede po 8 bitov (8 bitov = 1 byte, izg. »bajt«). Vsaka beseda v pomnilniku ima svoj *naslov*. Beseda na tem naslovu pomnilnika predstavlja torej *vsebinsko* pomnilnika na naslovu. Velikost pomnilnika izražamo v byte-ih (B) oz. mnogokratnikih le-tega (npr. MegaByte, GigaByte). Primer: za naslavljanje 256MB pomnilnika potrebujemo naslove dolžine 28 bitov ($256 = 2^8$ in $1\text{MB} = 2^{20}$ torej $256\text{MB} = 2^{28}$ ali drugače 2^{31} bitov).

4.4.4. Nadzorna enota

Nadzorna enota vodi evidenco o stanju izvajajočega se programa in stanju trenutnega ukaza. V nadzorni enoti najdemo register *programski števec*, ki beleži lokacijo v po-

Spletna stran

Naslovnica

Kazalo



Stran 75 od 219

Nazaj

Full Screen

Zapri

Končaj

naslov	vsebina
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	1 1 0 0 1 0 1 1
0 1 0 0	
.....
1 1 1 0	0 1 0 0 1 0 0 0
1 1 1 1	

Tabela 4.7: Primer pomnilnika

mnilniku od koder je bil pridobljen trenutni ukaz (in se nahaja v ukaznem registru). Ukaz se prenese v *ukazni register* (IR, angl. instruction register), ki beleži trenutni ukaz v izvajanju.

4.4.5. Vhodne in izhodne enote

Vhodno/izhodne naprave predstavljajo velik delež cene računalnika. Uporabnost današnjih računalnikov je tesno povezana s pestrostjo razpoložljivih vhodno/izhodnih naprav.

Trdi disk je naprava za shranjevanje podatkov in spada med vhodno izhodne naprave. Tudi pomnilnik je naprava za shranjevanje, vendar kratkotrajnejše. Ali ste se kdaj vprašali kakšna je cena enote pomnilnika trdega diska v primerjavi z delovnim pomnilnikom? Poglejmo primerjavo skozi leta, ki temelji na povprečni ceni trdega diska in delovnega pomnilnika. (tabela 4.8).

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 76 od 219

Nazaj

Full Screen

Zapri

Končaj

leto	naprava	velikost (MB)	cena (EUR)	cena/1MB
11/2004	trdi disk	163840	96,00	0,0006
10/2005		163840	89,00	0,0005
10/2006		262144	92,00	0,0003
10/2007		512000	115,00	0,0002
10/2008		786000	117,00	0,0002
10/2010		1048576 (1TB)	64,00	0,00006
10/2011		1048576 (1TB)	56,00	0,00005
11/2004	delovni pomnilnik	512	84	0,16
10/2005		512	54,00	0,11
10/2006		512	58,00	0,11
10/2007		1024	30,00	0,03
10/2008 (400 MHz)		4096	85,00	0,021
10/2010 (800 MHz)		4096	97,00	0,024
10/2011 (1333 MHz)		4096	30,00	0,007

Tabela 4.8: Cena 1MB pomnilnika (delovni pomnilnik in trdi disk) skozi leta

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

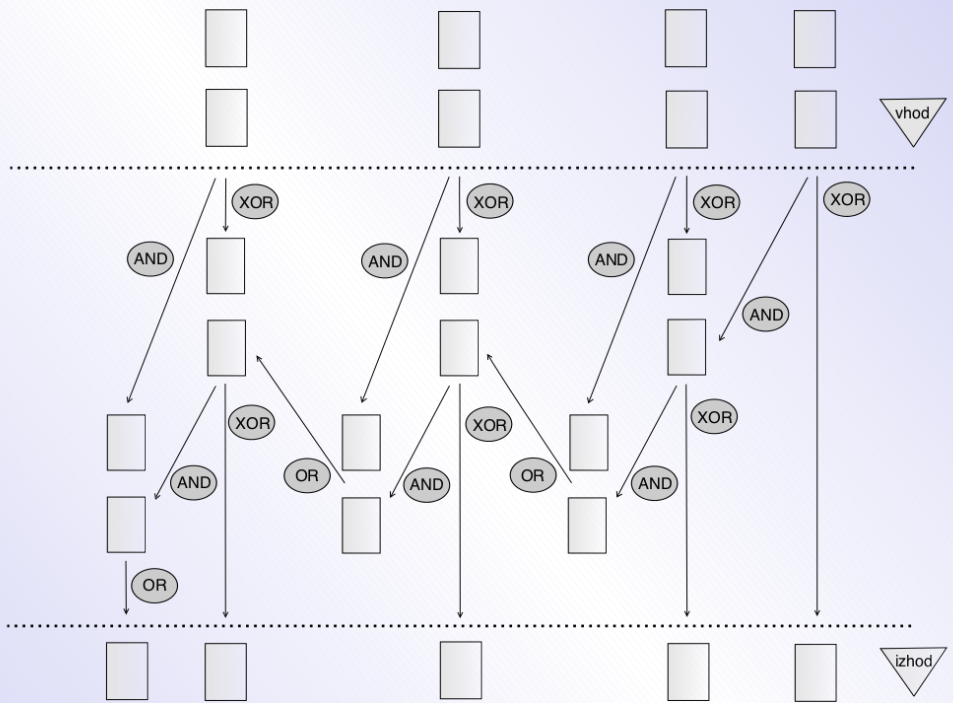
Stran 77 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 4.3: Logični binarni seštevalnik [2]

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 78 od 219

Nazaj

Full Screen

Zapri

Končaj

4.5. Vrste računalniških sistemov

4.5.1. Namizni računalniški sistemi

Namizni računalniški sistemi (angl. *desktop systems, desktop computers*) ali osebni računalniki (angl. *personal computers*) so bili, po svoji zasnovi, v začetku namenjeni enem samemu uporabniku (odtod tudi naziv osebni računalnik). Iz zgodovine se spomnimo, da so bili prvi elektronsko digitalni računalniki prav tako namenjeni le enem samemu uporabniku.

Značilnosti namiznih računalnikov so:

- dobra odzivnost celotnega računalniškega sistema,
- zahteve za učinkovito upravljanje s strojnimi napravami so manjše kot pri velikih računalniških sistemih,
- uporabnik namiznega računalnika je največkrat ena oseba, zato napredni koncepti za čimboljši izkoristek CPE niso v ospredju snovalcev OS za te sisteme

Osebni računalniki so splošno namenski zato jih lahko uporabljamo z različnimi (nameščenimi) OS: MS Windows, Mac OS X, Linux, Solaris.

4.5.2. Paralelni računalniški sistemi

Paralelne računalniške sisteme sestavlja več procesorjev, ki so v tesni medsebojni komunikaciji. Pravimo tudi, da so to *tesno sklopljeni sistemi*, angl. *tightly coupled system*. Procesorji v takšnih sistemih imajo skupen ali deljen pomnilnik (angl. *shared memory*) in uro za takt. Komunikacija običajno poteka s pomočjo skupnega pomnilnika. Prednosti paralelnih računalniških sistemov so:

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 79 od 219

Nazaj

Full Screen

Zapri

Končaj

- velika procesorska moč
- ekonomičnost (včasih vprašljiva!)
- povečana zanesljivost delovanja
- modularnost

Primer uporabe paralelnega računalnika je *mikrosimulacija prometa* (slika 4.4) . To je simulacija premikanja vozil v cestnem omrežju. Gre za računalniško intenziven problem saj želimo na ekranu prikazovati zvezne pomike vozil. Paralelizacijo izvedemo z delitvijo posameznih delov omrežja (karte) in pripadajočega prometa med sodelujoče računalnike. Reševanje problemov z uporabo paralelnega računalnika, torej s sočasno uporabo vsaj dveh procesorjev, imenujemo tudi *multiprocesiranje* .

4.5.3. Porazdeljeni računalniški sistemi

Porazdeljeni računalniški sistem (angl. distributed computer system) sestavljata vsaj dva aktivna računalnika na različnih lokacijah. Pravimo tudi, da so to *rahlo sklopljeni sistemi*, angl. *loosely coupled system*. Računalniki v porazdeljenem sistemu imajo svoj lokalni pomnilnik in medsebojno komunicirajo z uporabo različnih komunikacijskih kanalov (lokalno računalniško omrežje, telefonsko omrežje). Prednosti porazdeljenih računalniških sistemov so:

- združevanje virov
- pohitritev procesiranja - delitev podatkovnega bremena
- velika zanesljivost

Spletna stran

Naslovnica

Kazalo



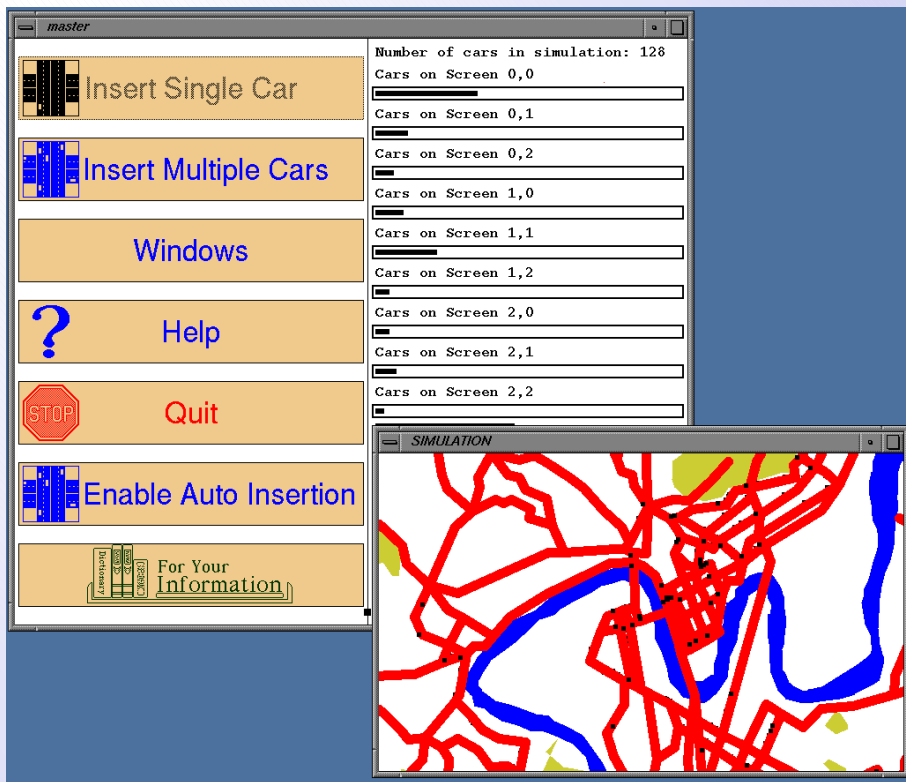
Stran 80 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 4.4: Simulacija prometa na porazdeljenem računalniku

- uporaba obstoječih komunikacijskih kanalov

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

◀◀ ▶▶

◀ ▶

Stran 81 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

4.6. Moore-ov zakon


Moore-ov zakon (angl. Moore's Law, Gordon Moore, 1965) pravi, da se število tranzistorjev na silikonskem čipu podvoji vsakih 18-24 mesecev. Povečanje števila tranzistorjev v procesorju posledično poveča zmogljivost procesorja .

Tabela 4.9: Moore-ov zakon in procesorji Intel

Intel CPU	Datum	Benchmark	Velikost besede	Velikost vodila	Taktura (MHz)	Tranzistorji (10^6)	Opombe
8088	1979	1.0	16	8	4.77	0.029	1MB RAM limit
80286	1982	3.1	16	16	8	0.134	16MB RAM limit
80386DX	1985	15	32	32	16	0.275	4GB RAM limit
80486DX	1989	54	32	32	25	1.2	8KB L2 cache in FPU
Pentium	1993	190	32	64	60	3.1	16KB L1 cache in FPU

nadaljevanje na naslednji strani

Spletna stran

Naslovnica

Kazalo



Stran 82 od 219

Nazaj

Full Screen

Zapri

Končaj

Tabela 4.9: nadaljevanje

Intel CPU	Datum	Benchmark	Velikost besede	Velikost vodila	Takt ure (MHz)	Tranzistorji (10^6)	Opombe
Pentium II	1997	570	32	64	233	7.5	64GB RAM limit, 32KB L1 cache, FPU, MMX instruction set
Pentium III (Katmai)	1999	1200	32 (8 128-bit floating point registers)	64	450	9.5	64GB RAM limit, 32 KB L1 cache, FPU, MMX and SIMD instruction sets

nadaljevanje na naslednji strani

Spletna stran

Naslovnica

Kazalo



Stran 83 od 219

Nazaj

Full Screen

Zapri

Končaj

Tabela 4.9: nadaljevanje

Intel CPU	Datum	Benchmark	Velikost besede	Velikost vodila	Takt ure (MHz)	Tranzistorji (10^6)	Opombe
Pentium III (Coppermine)	2000	2500	32 (8 128-bit floating point registers)	64 (256-bit L1 cache bus)	850	28.1	64 GB RAM limit, 32 KB L1 cache, FPU, MMX in SIMD instruction sets

4.6.1. Diskusija o Moore-ovem zakonu

- Ali Moore-ov zakon v praksi še vedno drži?
- Zmogljivost se podvoji vsaka 3 leta...
- Ali lahko povečanje zmogljivosti procesorja posplošimo na povečanje zmogljivosti računalnika?
- Stroški podjetij, ki se ukvarjajo z izdelavo čipov se z vsako novo generacijo čipov ekponentno povečajo!

Spletna stran

Naslovnica

Kazalo



Stran 84 od 219

Nazaj

Full Screen

Zapri

Končaj

4.7. Izbrani primeri

4.7.1. Izvajanje programa v enoprocorskem računalniku

Računalniški program (zaporedje ukazov za računalniško obdelavo) in podatki so po von Neumannovem modelu shranjeni v pomnilniku. Pojasnimo izvajanje preproste računalniškega programa, ki se lahko izvede v naslednji praktični situaciji:

- seštevanje dveh števil (npr. $C=A+B$) s programom Računalo ali
- vrstični pomik pri urejevalniku besedil (pomik s smernimi tipkami iz trenutne vrstice A za B vrstic naprej, rezultat je utripanje značke v novi vrstici C.

Za izvajanje računalniškega programa potrebujemo nek hipotetični računalnik z mikroprocesorjem in registri ter pomnilnikom. Vhodno izhodne naprav ne bomo potrebovali. Izberimo hipotetični računalnik, ki vsebuje:

- centralno procesno enoto s štirimi registri (A,B,C,D) in naborom le treh operacij (ADD, LOAD, STORE)
- pomnilnik z 8 bitnim naslavljanjem

Na kratko lahko naš program zapišemo: vrednosti v A prištej vrednost iz B in rezultat shrani v C.

Vsaka operacija ima svoj binarni ekvivalent (tabela 4.10). Računalnik namreč ne razume angleških besed, ampak le binarni zapis.

Tudi imenom štirih registrov priredimo binarni zapis (tabela 4.11).

Celoten ukaz zapišemo kot kombinacijo binarnih kod za mnemonike operacij in registre v točno določenem zapisu:

Spletna stran

Naslovnica

Kazalo



Stran 85 od 219

Nazaj

Full Screen

Zapri

Končaj

Mnemonic operacije	Binarna koda
ADD	0000
LOAD	0001
STORE	0010

Tabela 4.10: *Binarne kode mnemonikov za operacije*

Register	Binarna koda
A	00
B	01
C	10
D	11

Tabela 4.11: *Binarne kode registrov*

- ukazna vrstica za **aritmetične operacije** (tabela 4.12)
- ukazna vrstica za **operacije za dostop do pomnilnika** (tabela 4.13)

Byte 1		
Binarna koda operacije	ciljni register	izvorni register

Tabela 4.12: *Oblika zapisa ukaza za aritmetične operacije*

Zapišimo zdaj primer celotnega ukaza za seštevanje:

ADD A, B = 00000001

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 86 od 219

Nazaj

Full Screen

Zapri

Končaj

Byte 1			Byte 2
Binarna koda operacije	register	00	naslov

Tabela 4.13: *Oblika zapisa za operacije za dostop do pomnilnika*

Še nekaj dodatnih primerov ukazov z aritmetičnimi operacijami:

```
ADD C, D = 00001011
```

```
ADD D, B = 00001101
```

Zapišimo še ukaz za dostop do pomnilnika (12 je 1100 oz. 00001100 v binarnem 8-bitnem zapisu):

```
Load A, #12 = 00010000 00001100
```

Celoten program za seštevanje dveh števil izgleda takole (tabela 4.14):

Vrstica	Zbirni jezik	Strojni jezik
1	LOAD A, #12	00010000 00001100
2	LOAD B, #13	00010100 00001101
PŠ⇒ 3	ADD A, B	00000001
4	STORE A, #14	00100000 00001110

Tabela 4.14: *Program v zbirnem in strojnem jeziku*

Posamezni vrstici pravimo *ukaz*, ki ga sestavljajo (*koda operacije* (ADD) in (*operandi* (A,B,C). Mnemoniki (npr. ADD je mnemonik ukaza za seštevanje) služijo lažjemu (človeškemu) pomnjenju ukazov (inštrukcij). Mnemoniki tvorijo t.i. *zbirni*

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 87 od 219

Nazaj

Full Screen

Zapri

Končaj

Cikel	ukaz	Opis
FETCH	ADD A,B	Programski števec se poveča za 1
DECODE	ADD	Dekodiraj pomen mnemonika ADD (seštej)
EVALUATE ADDRESS	A, B	ovrednoti naslov
FETCH OPERANDS	LOAD R1, A	pridobi operande
EXECUTE	ADD R1,B	izvedi ukaz
STORE	STORE R1, A	shrani rezultat na naslovu A

Tabela 4.15: Koraki izvajanja ukaza za seštevanje

jezik, ki ga programerji še dandanes uporabljajo za izvajanje časovno kritičnih delov programske kode (npr. programi za krmiljenje telefonskih central).

Pri tem velja, da niti *A* niti *B* ne predstavljata seštevancev, ampak *naslov* na katerem se seštevanca nahajata. Pravi vrednosti obeh seštevancev mora procesna enota šele pridobiti. *C* predstavlja naslov kamor se bo shranil rezultat. Da bi bolje razumeli uporabimo analogijo...*C* je številka bančnega računa na katerega želimo prenesti denar iz bančnih računov *A* in *B*. Seštevamo torej vrednost v denarju, ki se nahaja na računih in ne številki bančnih računov. Čeprav posamezna vrstica našega programa predstavlja 1 korak programskega števca, je za izvedbo posamezne vrstice potrebnih več korakov (ali *ciklov*). V našem primeru procesiranje (npr. `ADD A, B`) ukaza obsega šest ciklov.

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 88 od 219

Nazaj

Full Screen

Zapri

Končaj

4.8. Več o zgradbi digitalnih računalnikov...

1. Biografija John-a von Neumanna [21.10.2005]
2. Računalniška aritmetika [21.10.2007] in binarni odštevalnik [21.10.2007]
3. Mooreov zakon [04.11.2005]

Spletna stran

Naslovnica

Kazalo



Stran 89 od 219

Nazaj

Full Screen

Zapri

Končaj

4.9. Izbirne naloge

Naloga 4.1

Izdelajte interaktivno spletno stran, ki prikaže delovanje binarnega seštevalnika. Spletna stran naj uporabniku omogoča vnos dveh štiribitnih vrednosti. Rezultat naj se s postopnim prikazom vmesnih logičnih operacij izpiše na dnu seštevalnika.

Naloga 4.2

Izdelajte animacijo delovanja digitalnega (enoprosesorskega) računalnika. Za izdelavo animacije uporabite Scalable Vector Graphics (SVG); rezultat naj bo datoteka v zapisu SVG. Animacija naj bo grafično in funkcionalno identična kot jo prikazuje slika 4.2 in animacija [9].

Spletna stran

Naslovnica

Kazalo



Stran 90 od 219

Nazaj

Full Screen

Zapri

Končaj

Poglavje 5

Operacijski sistemi

Imeti dober operacijski sistem (OS) je želja vsakega uporabnika računalnikov. Kaj pa je dober OS? Kakšne so njegove značilnosti? Zaradi pomanjkanja znanja večina uporabnikov o primerjalnih lastnostih operacijskih sistemov težko sodi. Pri izbiri operacijskega sistema bo lažje, če poznamo osnovne koncepte OS.

To poglavje opisuje zgodovino, teorijo in prakso operacijskih sistemov. Cilji poglavja so, spoznati in razumeti:

- povezavo med zgradbo računalnika, OS in računalniškimi programi
- osnovne koncepte OS
- delitev in funkcije OS

Vse zgoraj naštetu vodi k razvoju kritičnega razmišljanja o današnjih operacijskih sistemih.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **91** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

5.1. Primer iz prakse

Družba CM Celje d.d. se je leta 1997 odločila, da bo informacijski sistem obvladovala tako, da bo na področju informacijskega sistema delovala z minimalnim številom zaposlenih ter ne bo razvijala lastnih produktov programske opreme ter bo uvajale tako imenovane tipske rešitve, ki bodo čim bolj kompatibilne z obstoječim IT sistemom CM Celje d.d. ter zunanjim okoljem CM Celje d.d. Naloga zaposlenih skupne informatika je zagotavljati podporo uporabnikom programske opreme v procesih pri postavljanju zahtev za programsko opremo do izdelovalcev programske opreme, zagotavljati tehnično kompatibilnost novih rešitev z obstoječim IT sistemom, skrbeti, za nemoteno delovanje informacijskega sistema z opravljanjem enostavnih vzdrževalnih posegov, ter načrtovanje razvoja na področju informacijskih tehnologij in telekomunikacij. Od 450 zaposlenih delajo v skupini za informatiko 4 zaposleni. (vir: seminarska naloga “Obvladovanje stroškov informacijske tehnologije z možnostjo uvedbe programske opreme odprte kode”, Marjan Vengust)

Vprašanja za razmislek:

- Na kakšen način lahko podjetja zmanjšajo stroške za informacijski sistem?
- Ali *zamenjava OS* lahko zmanjša stroške za informacijski sistem podjetja?
- V katerih segmentih je realno pričakovati zmanjšanje stroškov za informacijski sistem podjetja?
- Ali lahko znižamo stroške pri osebem (domačem) računalniku?

Spletna stran

Naslovnica

Kazalo



Stran 92 od 219

Nazaj

Full Screen

Zapri

Končaj

5.2. Umestitev operacijskega sistema v računalniški sistem

Obravnava operacijskih sistemov v kontekstu programske opreme, ki je potrebna za oživitev arhitekture računalnika, je logično nadaljevanje poglavja o arhitekturi računalnika.

Operacijske sisteme uvrščamo med sistemsko programsko opremo računalnika (slika 5.1).

OS je tesno povezan z arhitekturo računalniškega sistema, zato ga umestimo v računalniški sistem na sledeči način: (slika 5.2):

- Strojna oprema: osnovni računalniški viri (CPE, pomnilnik, vhodne/izhodne enote)
- Operacijski sistem: nadzira in koordinira uporabo strojne opreme med različnimi aplikativni programi in za različne uporabnike (v primeru večuporabniškega operacijskega sistema)
- Aplikativni programi: definirajo različne načine za uporabo sistemskih virov z namenom reševanja uporabniških problemov (inženirski programi, poslovni programi, pisarniški programi, prevajalniki, podatkovne baze)
- Uporabniki: ljudje, stroji, ostali računalniki

Spletna stran

Naslovnica

Kazalo



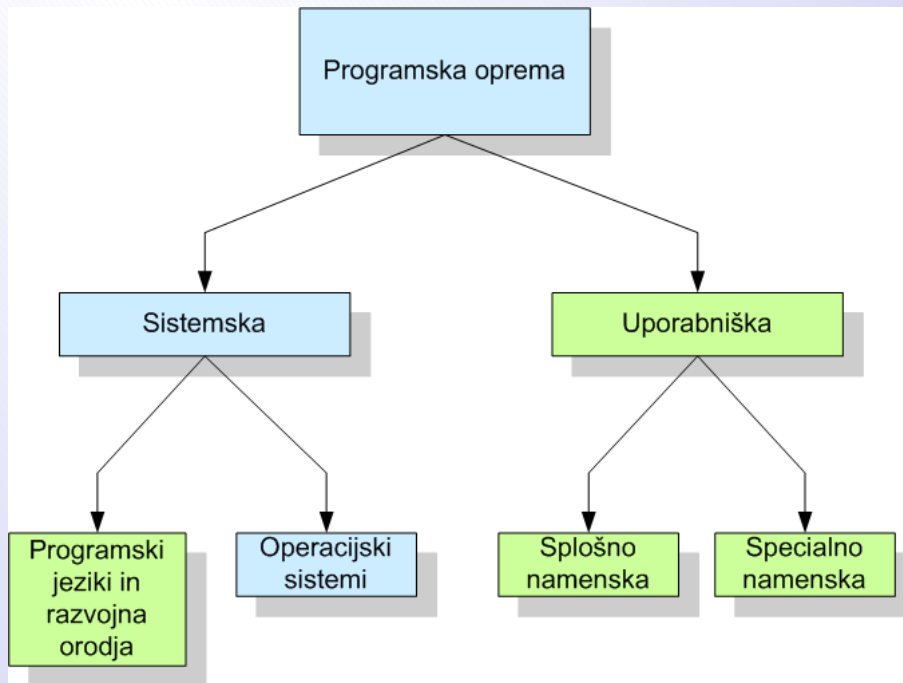
Stran 93 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 5.1: OS – sistemska programska oprema

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

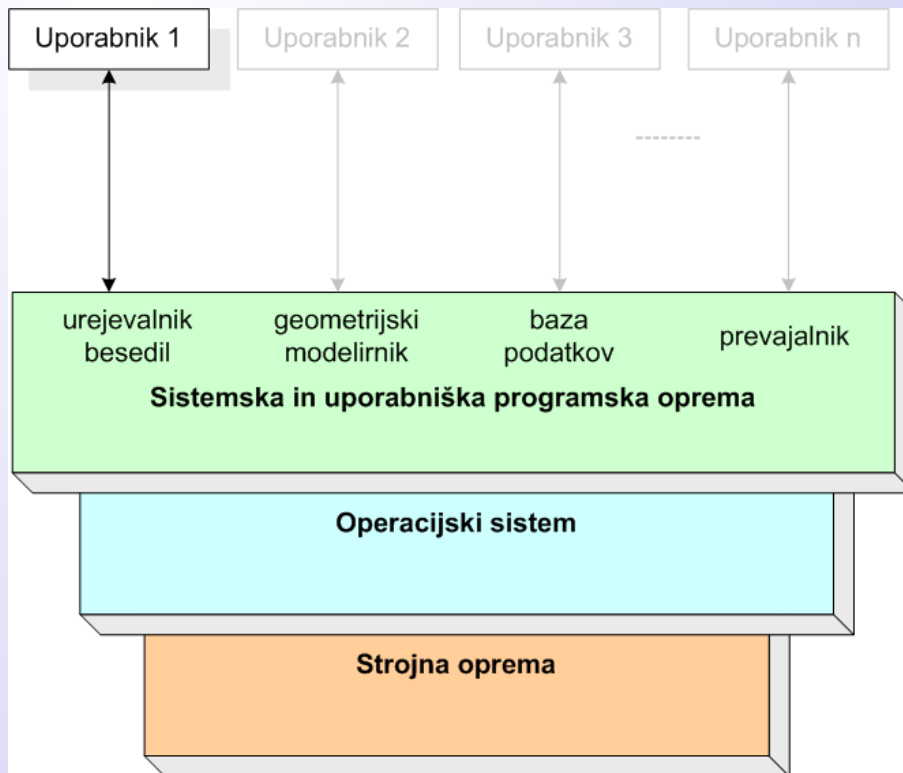
Stran 94 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 5.2: Umestitev OS v računalniški sistem

Navigation and content controls:

- Spletna stran
- Naslovnica
- Kazalo
- Navigation arrows: << >>
- Navigation arrows: < >
- Page information: Stran 95 od 219
- Nazaj
- Full Screen
- Zapri
- Končaj

5.3. Definicija in cilji operacijskega sistema

Definicija operacijskega sistema: OS je uigrana množica *sistemskih programov*, ki delujejo kot posrednik med *uporabnikovimi programi ali razvijalcem uporabniških programov* in *računalnikovimi strojnimi komponentami*, ter omogočajo njihovo učinkovito uporabo.

Zgornja definicija velja, če operacijski sistem uresničuje naslednje **cilje**:

1. maksimizirati *izkoriščenost* računalnikove *strojne opreme* (procesor, pomnilnik, vhodne in izhodne naprave) za potrebe uporabnikovih programov, t.p. delitev in dodeljevanje računalniških virov različnim uporabnikovim programom
2. maksimizirati produktivnost uporabnika, t.p. zagotoviti enostaven vmesnik za izvajanje uporabnikovih programov in upravljanje s podatki z namenom lažjega reševanja uporabnikovih problemov
3. zagotoviti vmesnik do sistemskih virov za razvijalce aplikativne programske opreme

Spletna stran

Naslovnica

Kazalo



Stran 96 od 219

Nazaj

Full Screen

Zapri

Končaj

5.4. Delitev operacijskih sistemov

V tem poglavju odkrivamo značilnosti operacijskih sistemov. Te značilnosti so inženirju v pomoč pri izbiri operacijskega sistema za različne strokovne probleme (primer: izbira zanesljivega operacijskega sistema za računalniški program za merjenje temperature vozišča, rezultate meritev uporablja zimska služba za vzdrževanje cest).

5.4.1. Delitev OS glede na uporabniški vmesnik

Uporabniški vmesnik (UV, angl. UI - User Interface) je vidni del OS preko katerega uporabnik komunicira z OS. Zgodovinsko gledano poznamo:

- *Uporabniški vmesnik z ukazno vrstico* (angl. command line user interface): uporabnik mora ukaze tipkati v ukazni vrstici. Primer: DOS, VMS
- *Menujski uporabniški vmesnik* (angl. menu driven user interface): nadgradnja ukaznega UV, kjer so ukazni vrstici dodani preprosti grafični menuji iz katerih uporabnik izbira ukaze iz menujev (najpogosteje na zgornjem delu ekrana).
- *Grafični uporabniški vmesnik* (angl. graphical user interface): nadgradnja ukaznega in menujskega, kjer “ukazujemo” s pomočjo miške, tipkovnice in grafičnih objektov na ekranu (okna, gumbi, drsniki, tekstovni okvirji, ikone, slike). Primeri grafičnih uporabniških vmesnikov: Windows (slika 5.3), MAC OS X (slika 5.5) ter Unix in Linux (Okna X (angl. X-Windows) so *grafični uporabniški vmesnik* za operacijski sistem Linux, Unix (slika 5.4))

Spletna stran

Naslovnica

Kazalo



Stran 97 od 219

Nazaj

Full Screen

Zapri

Končaj

5.4.2. Delitev OS glede na število centralnih procesnih enot, ki jih OS zmore izkoriščati

- OS za podporo *enoprocesorski* arhitekturi računalnika (DOS, Mac OS, Linux, OS/2)
- OS za podporo *večprocesorski* arhitekturi računalnika (Unix, Windows 2000 server, Windows XP server)

5.4.3. Delitev OS glede na število opravil, ki jih OS zmore podpirati “sočasno”

- Enopravilni (DOS)
- Večopravilni (VMS, UNIX, LINUX, OS/2, Windows od verzije 95 dalje). Pri večopravilnem OS procesor izmenično izvaja programe naložene v pomnilnik.

5.4.4. Delitev OS glede na število uporabnikov, ki jih OS podpira

- Enouporabniški (DOS, ..., namizne različice Windows OS, OS/2)
- Večuporabniški (UNIX, VMS, Linux, strežniške različice Windows OS - npr. Windows 2003 server)

5.4.5. OS za osebne računalnike

- MS DOS (ni več aktualen)

Spletna stran

Naslovnica

Kazalo



Stran 98 od 219

Nazaj

Full Screen

Zapri

Končaj

- OS/2 (ni več aktualen)
- različice Windows (3.11, 95, 98, NT, 2000, XP, Vista, 7)
- Linux (več kot 200 različic)
- Mac OS X (X = deset, angl. ten)
 - Mac OS X verzija 10.0 (Cheetah), marec/2001
 - Mac OS X verzija 10.1 (Puma)
 - Mac OS X verzija 10.2 (Jaguar)
 - Mac OS X verzija 10.3 (Panther)
 - Mac OS X verzija 10.4 (Tiger), april/2005
 - Mac OS X verzija 10.5 (Leopard), oktober/2007
 - Mac OS X verzija 10.6 (Snow leopard), avgust/2009
 - Mac OS X verzija 10.7 (Lion), julij/2011, slika 5.5
 - Mac OS X verzija 10.8 (Mountain Lion), julij/2012

5.4.6. OS za strežnike in delovne postaje

OS za delovne postaje:

- Unix (Solaris, AIX, HP-UX)
- VMS / VAX

Spletna stran

Naslovnica

Kazalo



Stran 99 od 219

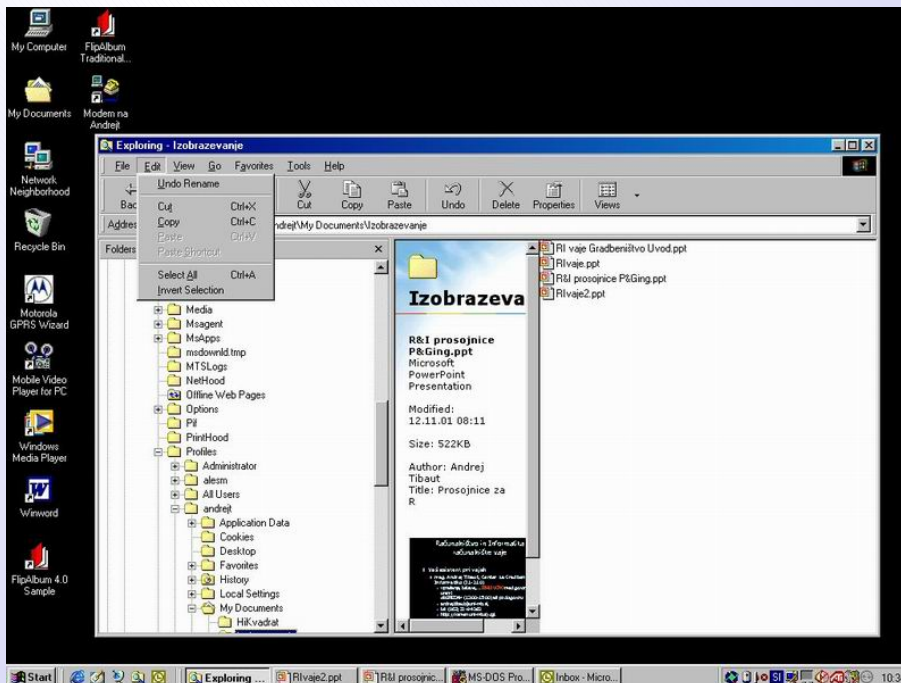
Nazaj

Full Screen

Zapri

Končaj

- Mac OS X Server
- Windows Server



Slika 5.3: Windows

Spletna stran

Naslovnica

Kazalo

◀ ▶

◀ ▶

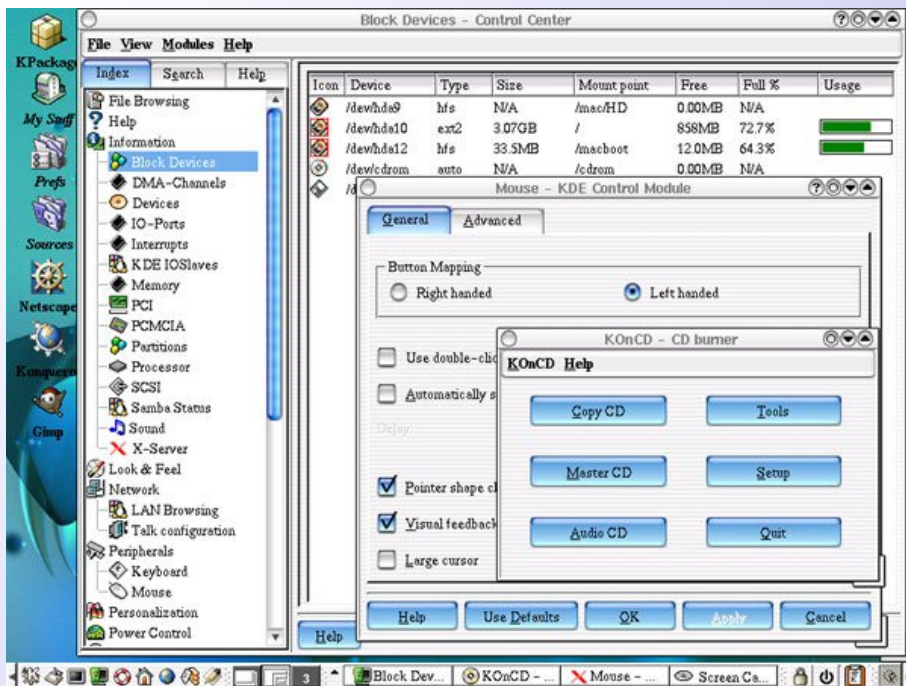
Stran 100 od 219

Nazaj

Full Screen

Zapri

Končaj

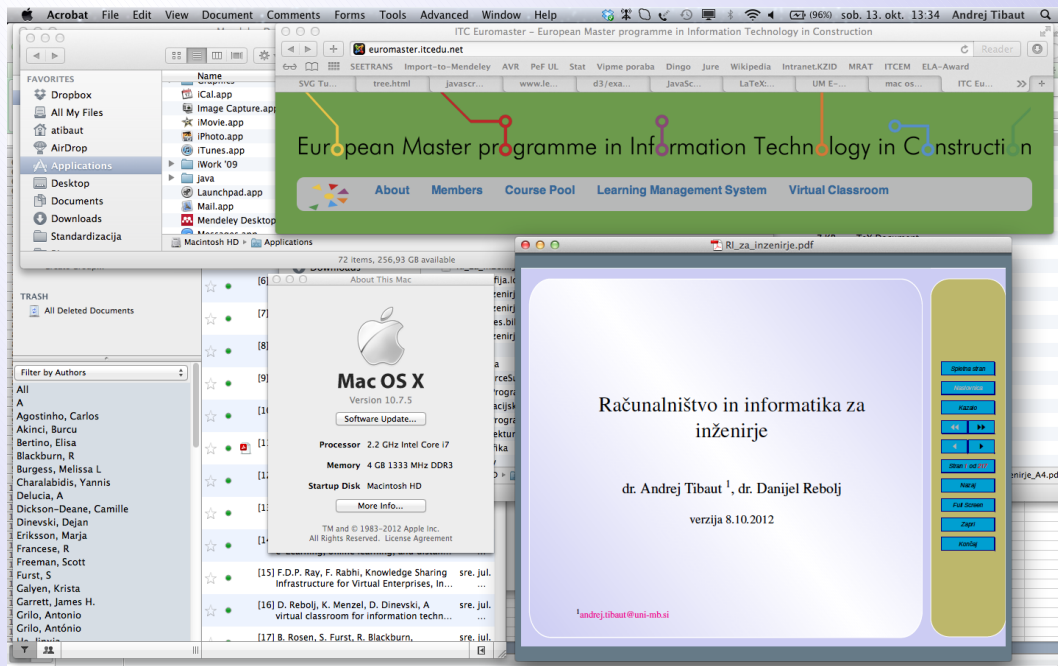


Slika 5.4: Linux

5.4.7. OS za dlančne računalnike

Dlančnim računalniškimi (angl. *PDA - Personal Digital Assistants*) napravam pravimo t.i. dlančniki. Sem spadajo tudi pametni mobilni telefonski aparati (angl. smart phones). Značilnosti:

- [Spletna stran](#)
- [Naslovnica](#)
- [Kazalo](#)
- ◀◀
▶▶
- ◀
▶
- [Stran 101 od 219](#)
- [Nazaj](#)
- [Full Screen](#)
- [Zapri](#)
- [Končaj](#)



Slika 5.5: namizje Mac OS X v10.7 (Lion)

- majhen pomnilnik
- manj zmogljivi procesorji
- majhen ekran

OS za dlančnike podpirajo interaktivnost in večopravnost.

Spletna stran

Naslovica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 102 od 219

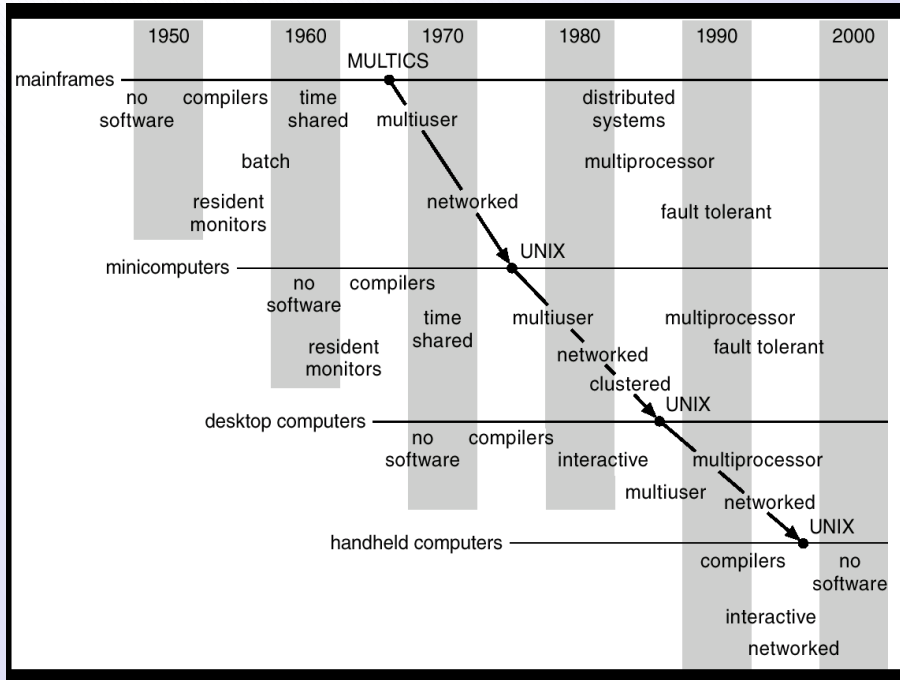
Nazaj

Full Screen

Zapri

Končaj

5.5. Osnovni koncepti in funkcije sodobnih operacijskih sistemov



Slika 5.6: Ponavljanje konceptov OS

Spremljanje razvoja operacijskih sistemov z nekaj desetletno časovno razdaljo omogoča “pogled od daleč”. Prvi operacijski sistemi so se razvili v poznih 1950-tih

Spletna stran

Naslovnica

Kazalo



Stran 103 od 219

Nazaj

Full Screen

Zapri

Končaj

letih (upravljanje tračnih naprav). Sredi 1960-tih so operacijski že upravljali delo z diski.

Tako lahko ugotovimo, da je razmišljanje o operacijskih sistemih pravzaprav razmišljanje o organizaciji delovanja računalnika. Takšno razmišljanje je polno analogij iz vsakdanjega življenja.

Računalnik je primerljiv z veliko tovarno: velika tovarna mora imeti zelo dobro organizacijo delovnega procesa, da bi bili stroji kar najboljše izkoriščeni, ljudje večji upravljana s stroji in s tem proizvodni rezultati čimboljši.

Manjša podjetja se konceptov organizacije učijo od velikih, prav tako kot koncepti operacijskih sistemov za velike računalniške sisteme (angl. *mainframes*) slej ko prej veljajo tudi za operacijske sisteme naslednje generacije (namiznih, prenosnih in dlančnih) računalniških sistemov.

5.5.1. Podpora uporabnikom

Kot sledi iz uvodne definicije je eden izmed ciljev operacijskega sistema tudi i učinkovito delo uporabnika. V ta sklop uvrščamo tudi mehanizme za zagotavljanje varnosti uporabniku OS.

5.5.2. Servisiranje programov

Operacijski sistem nudi podporo programom v izvajanju od trenutka, ko se program naloži iz sekundarnega pomnilnika (npr. disk) v glavni pomnilnik. OS bdi nad izvajanjem programa vse do konca izvajanja, ko se glavni pomnilnik sprosti, kontekst programa (npr. spremenjen dokument) pa zapiše nazaj na sekundarni pomnilnik.

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 104 od 219

Nazaj

Full Screen

Zapri

Končaj

5.5.2.1. Preprosta paketna obdelava

Eden prvih konceptov OS je bil *paketno procesiranje* ali *paketna obdelava* (angl. *batch, offline processing*). Gre za koncept obdelave, kjer se posli izvajajo v ozadju (kot nasprotje interaktivnemu izvajanju), brez možnosti uporabnikovega vpliva. Uporabnik pripravi podatke in izbere programe s katerimi bo te podatke obdelal, vse skupaj spravi v paket in ga pošlje v obdelavo. Takšen OS je zagotavljal zaporedno (*sekvenčno*) obdelavo paketov v pomnilniku. Sekvenčno je bilo tudi izvajanje posameznih faz v življenju paketa: *vhod podatkov in programov, obdelava, prikaz rezultatov*. Za razpoznavanje in obdelavo paketov (proženje obdelave naslednjega paketa po zaključku trenutnega) je skrbel operater. O prvem preprostem operacijskem sistemu govorimo šele, ko je operaterja zamenjal čitalec prelučnjanih kartic (avtomatizacija), ki je bral in izvajal ukaze zapisane na karticah, ter skrbel za prenos obdelave od paketa do paketa.

Ni bilo časovnega prekrivanja posameznih faz (npr. CPE in vhodnih/izhodnih naprav).

Primer analogije za paketno obdelavo je *oddaja fotografskih filmov in izdelava slik*. Uporabljeno rolo fotografskega filma lahko oddamo tudi na mestih izven fotografskih trgovin (veliki trgovinski centri). Na teh mestih moramo sami pripraviti paket za oddajo (zapakirati film v predpisan paket in ga opremiti z naslovom). Fotograf nato film obdela in nam vrne rezultat (slike).

Primer paketne obdelave: imamo dva kratka programa (A in B), ki izračunata sinusno krivuljo (A) in eksponentno funkcijo (B) in rezultate zapišeta v ločene datoteke. Makro v elektronski preglednici (C) prebere rezultate iz obeh datotek, jih

Spletna stran

Naslovnica

Kazalo



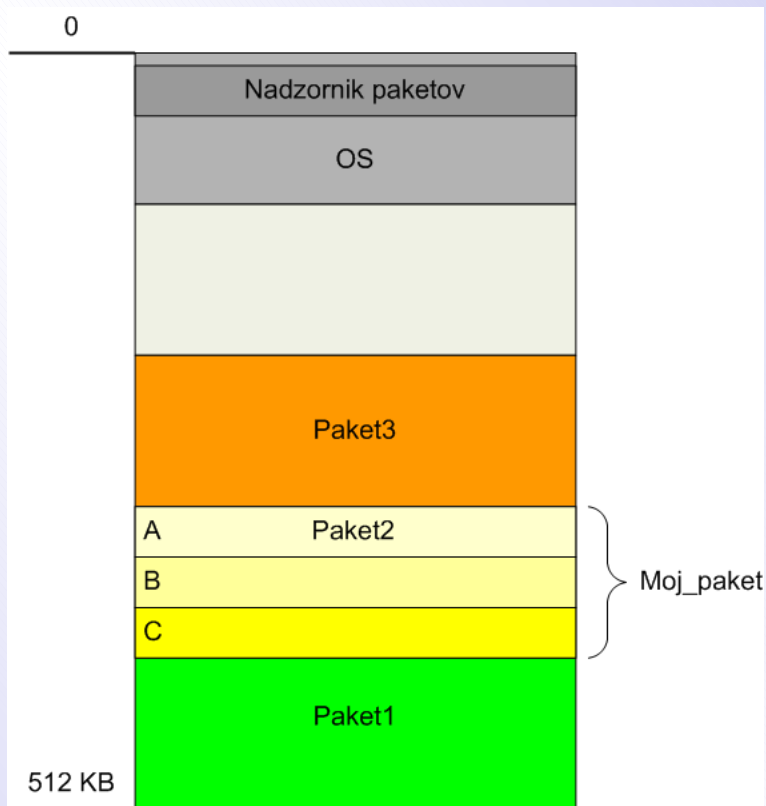
Stran 105 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 5.7: Kontrolna struktura v pomnilniku pri paketni obdelavi

zapiše v preglednico, kjer se rezultati iz A in B pomnožijo in prikažejo v obliki grafa. Pripravimo lahko paket (*Moj_paket*), ki vsebuje naslednje zaporedje izvajanja

Spletna stran

Naslovnica

Kazalo



Stran 106 od 219

Nazaj

Full Screen

Zapri

Končaj

programov: A, B, C (slika 5.7). Paket (*Moj_paket*) pošljemo v obdelavo s posebnim ukazom.

5.5.2.2. Izboljšana paketna obdelava

Izboljšava paketne obdelave je šla v smeri sočasnega delovanja (prekrivanja) večih naprav računalnika (angl. *spooling*, *SPOOL = Simultaneous Peripheral Operation On-Line*, slika 5.8). OS je med nalaganjem paketa iz diska v pomnilnik, hkrati tudi izpisoval rezultate prejšnjega paketa na izhodno napravo.

Prvič se zgodi, da vsaj dva paketa sobivata v pomnilniku (slika 5.7), procesor pa jim izmenično (*multipleksiranje*) namenja svoj procesorski čas, kar imenujemo *večprogramska paketna obdelava*.

Operacijski sistem je za potrebe izboljšane paketne obdelave moral imeti naslednje lastnosti:

- podpora vnosu paketov
- upravljanje s pomnilnikom (dodeljevanje pomnilnika paketom)
- dodeljevanje CPE med različne pakete
- dodeljevanje strojne opreme (npr. izpis na tiskalnik)

5.5.2.3. Interaktivna obdelava

Interaktivna ali pogovorna obdelava (angl. *interactive, online processing*) je učinkovitejša alternativa paketni obdelavi. Značilnosti interaktivne obdelave so:

Spletna stran

Naslovnica

Kazalo



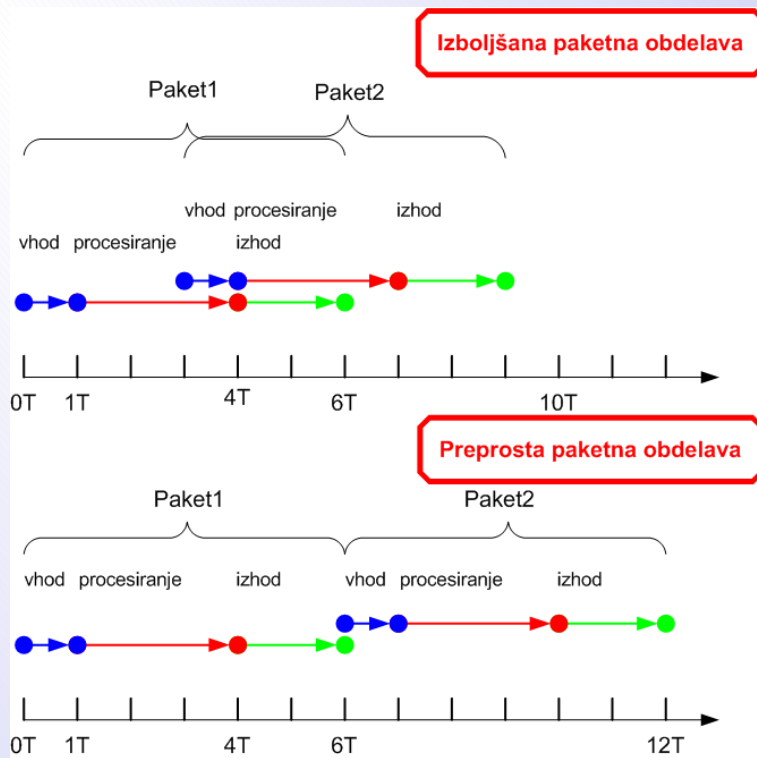
Stran 107 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 5.8: Preprosta in izolirana paketna obdelava

- komunikacija med uporabniki in operacijskim sistemom poteka "v živo", torej interaktivno
- operacijski sistem po zaključku nekega ukaza čaka na naslednji ukaz, ki ga

Spletna stran

Naslovnica

Kazalo

◀ ▶

◀ ▶

Stran 108 od 219

Nazaj

Full Screen

Zapri

Končaj

uporabnik vtipka s pomočjo tipkovnice

- uporabnik od operacijskega sistema pričakujejo kratek odzivni čas
- ker delo poteka interaktivno mora OS zagotoviti datotečni sistem kjer uporabnik shrani podatke in naloži programe.

Interaktivna obdelava je mogoča zaradi časovnega dodeljevanja procesorskega časa, kjer je vsakemu uporabniku (procesu) dodeljena določena količina procesorskega časa (*časovna rezina*).

Primer analogije za obdelavo s časovnim dodeljevanjem je *igranje šahovske simultanke*. Pri šahovski simultanki se šahovski mojster pomeri z večimi šahovskimi izzivalci sočasno. To izvede tako, da se določen čas posveti prvemu izzivalcu, nato usmeri svojo pozornost drugemu, nato tretjemu, itd. Medtem, ko šahovski mojster razmišlja o pravi potezi pri drugem izzivalcu, prvi izzivalec sočasno razmišlja o svoji novi potezi, nato se šahovski mojster pomakne k naslednjemu izzivalcu in s tem spet zaposli prejšnjega izzivalca, itd. Analizirajmo časovni potek simultanke... Predpostavimo naslednje poenostavitve: vsak igralec (šahovski mojster, izzivalec) za eno potezo potrebuje natanko 1 minuto, vsak igralec lahko potegne natanko 30 potez, nato se igra zaključi. Če bi šahovski mojster z vsakim izzivalcem odigral neprekinjeno partijo bi partija trajala 60 minut. Pri petih izzivalcih, bi se zadnja partija zaključila po 300 minutah (ali 5 urah). Če gre za simultanko se vse partije zaključijo po 150 minutah (ali 2 urah in 30 minutah). Prihranek v času je torej dvakratni. V računalniškem sistemu

Spletna stran

Naslovnica

Kazalo



Stran 109 od 219

Nazaj

Full Screen

Zapri

Končaj

lahko šahovskega mojstra enačimo s CPE, izzivalce pa z uporabniki, ki s svojimi procesi nalagajo delo CPE.

5.5.3. Upravljanje pomnilnika

Osvežimo spomin o vrstah pomnilnika... Govorimo o *glavnem* (npr. RAM, DRAM) in *sekundarnem* pomnilniku (npr. disk). Glavni pomnilnik je edino veliko pomnilniško področje do katerega lahko CPE dostopa direktno. Glavni pomnilnik ima dve slabosti:

- je kratkotrajen (angl. *volatile*)
- nikoli ga ni dovolj

Sekundarni pomnilnik (npr. disk) je razširitev glavnega pomnilnika, ki predstavlja velik in dolgotrajen (angl. *non-volatile*) pomnilnik.

Naloga upravljalnika pomnilnika je zagotoviti hiter dostop do čim večjega naslovnega prostora, po možnosti neodvisno od velikosti glavnega pomnilnika.

Pri upravljanju pomnilnika sta najpomembnejša dva koncepta:

- predpomnilnik (angl. *cache*)
- navidezni pomnilnik (angl. *virtual memory*)

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 110 od 219

Nazaj

Full Screen

Zapri

Končaj

5.5.3.1. Predpomnilnik

Primer analogije za predpomnilnik. Vaša shramba z živili je predpomnilnik trgovine v kateri nakupujete. Če potrebujete konzervo fižola jo boste najprej poiskali v shrambi. Šele če konzerve ne najdete v shrambi (ni zadetka), boste pohiteli v trgovino, nakupili nekaj konzerv in napolnili svojo shrambo.

Predpomnjenje je vzdrževanje kopije nekaterih podatkov, iz počasnejšega pomnilnega medija, na hitrejšem pomnilnem mediju. Podatki, ki se nahajajo v predpomnilniku (angl. *cache*) se dejansko trenutno tudi uporabljajo za procesiranje. Naloga predpomnilnika je zagotavljati preslikavo med kopijo in originalnimi podatki, ter skrbeti za njihovo konsistenco. Predpomnilnik je vrsta pomnilnika, ki je hitrejši od glavnega pomnilnika (in počasnejši od registrov). Pri večini današnjih računalnikov je vgrajen v CPE. Pomembno je vedeti, da je predpomnilnik majhen v primerjavi z glavnim pomnilnikom in da ga želimo narediti tako, da vsebuje tisti del vsebine glavnega pomnilnika, ki sestavlja trenutno delovno množico naslovov. To pomeni, da mora predpomnilnik skupaj z informacijo, ki ustreza vsebini dela glavnega pomnilnika, vsebovati tudi informacijo, ki pove kateremu delu glavnega pomnilnika pripada trenutna vsebina. Z drugimi besedami, predpomnilnik mora vsebovati tudi naslove, ki povedo katerim besedam glavnega pomnilnika ustreza njegova trenutna vsebina. Predpomnilnik (slika 5.9) je sestavljen iz dveh delov: iz kontrolnega dela in iz pomnilniškega dela. Pomnilniški del je razdeljen v enote enake velikosti, ki jim pravimo bloki, Kontrolni del vsebuje kontrolno informacijo, ki enolično identificira vsak blok. Ta informacija vsebuje trenutni pomnilniški naslov bloka ter še nekatere dodatne kontrolne bite kot so umazani bit, veljavni bit, ipd. Predpomnil-

Spletna stran

Naslovnica

Kazalo



Stran 111 od 219

Nazaj

Full Screen

Zapri

Končaj

nik (slika 5.9) uporablja direktno naslavljanje. To pomeni, da se vsak blok glavnega pomnilnika preslika v točno določen (vedno isti) blok predpomnilnika. Poleg tega poznamo še asociativno naslavljanje in set asociativno naslavljanje.

Delovanje predpomnilnika je sledeče: procesor generira pomnilniški naslov, če se ta naslov nahaja v bloku predpomnilnika bo podatek vrnjen kar iz predpomnilnika (trajanje: 1-2 cikla), če se naslov ne nahaja v predpomnilniku (pogrešek, angl. *cache miss*) je potrebno ta naslov najprej naložiti iz glavnega pomnilnika v blok predpomnilnika (trajanje: 8-100 ciklov). Statistika pravi, da je pogreškov do 10%.

5.5.3.2. Navidezni pomnilnik

Na mnogih računalnikih se lahko zgodi, da je glavni pomnilnik premajhen za današnje velike programe. Tudi, če je pomnilnik dovolj velik za nek program, je pri večopravilnih OS razpoložljivi prostor v pomnilniku potrebno deliti z drugimi uporabniki oz. programi. Del, ki je na voljo enemu programu je lahko premajhen za dano velikost programa. Vendar se ne zgodi, da bi zaradi premajhnega glavnega pomnilnika, programi ne delovali. Kako je to mogoče? Rešitev je *navidezni pomnilnik* (angl. *virtual memory*), ki omogoča, da se v pomnilnik naloži le nekaj delov programa, ostali deli pa ostanejo na disku. Ko se nek del programa v glavnem pomnilniku ne uporablja več, se ga zamenja z nekim drugim delom iz diska. Navidezni pomnilnik uporabniku omogoča, da navidezno poveča velikost glavnega pomnilnika z uporabo diska. Zakaj je navidezni pomnilnik sploh potreben? Če enkrat...omogoča, da več programov deli omejen fizični prostor glavnega pomnilnika, omogoča delo z večjim glavnim pomnilnikom kot ga je dejansko na razpolago, samodejno skrbi za polnjenje navideznega pomnilnika s podatki iz diska (v primeru da ni zadetka). Navidezni pomnilnik lahko razumemo tudi kot predpomnilnik za sekundarni pomnilnik

Spletna stran

Naslovnica

Kazalo



Stran 112 od 219

Nazaj

Full Screen

Zapri

Končaj

(disk). Delovanje navideznega pomnilnika prikazuje slika 5.10. Začetek dogajanja, ki pojasnjuje delovanje navideznega pomnilnika moramo začeti pri CPE. Le-ta v fazi izvajanja neke trenutne operacije generira naslov (primer: na tem naslovu se nahaja eden od operandov pri operaciji seštevanja) na katerem se nahaja nek podatek. Na sliki vidimo, da se prisotnost naslova, ki ga generira CPE, najprej primerja s seznamom naslovov zapisanih v *tabeli strani*, ki se nahaja v glavnem pomnilniku. V odvisnosti od *veljavnostnega bita* lahko vodi pot do tega naslova v glavni pomnilnik (veljavnostni bit = 1) ali v sekundarni pomnilnik (npr. disk, veljavnostni bit = 0). »e se željeni podatek nahaja v glavnem pomnilniku je to idealna situacija (zadetek, angl. *hit*, trajanje: 40-100 ciklov), če ga moramo naložiti iz sekundarnega pomnilnika je to manj ugodno (pogrešek, angl. *page fault*, trajanje: 1 do 6 milijonov ciklov). Sekundarni pomnilnik je namreč veliko počasnejši kot glavni pomnilnik. Statistika pravi, da je pogreškov do 0.001%.

5.5.4. Upravljanje perifernih naprav

Periferne enote (vhodno/izhodne naprave, kot npr. sekundarni pomnilniki, tiskalniki, digitalizatorji, itd.) se po svojem namenu, zgradbi, delovanju zelo razlikujejo. Kljub raznolikosti je potrebno zagotoviti povezavo teh naprav med seboj, s CPE in pomnilnikom. Prve periferne naprave so bili tiskalnik, naprava za lociranje (npr. miška), sekundarni pomnilnik in prikazovalnik. Zaradi majhnega števila je lahko imela vsaka naprava svoj lasten priključek. S porastom števila perifernih naprav so se pojavili univerzalnejši priključki, ki jih lahko uporablja več naprav (npr. USB, COM, Bluetooth). Periferne naprave so tako priključene, toda kako njihovo delovanje uskladiti z delovanjem CPE? CPE namreč ne ločuje med različnimi perifernimi napravami. Kot že vemo, CPE le izvršuje logično-aritmetične operacije in pri tem generira pomnil-

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 113 od 219

Nazaj

Full Screen

Zapri

Končaj

niške naslove za branje/zapis podatkov. Vsako periferno napravo krmili t.i. *gonilnik* (angl. *driver, controller*), ki pri tem uporablja lokalni vmesni pomnilnik (angl. *buffer*). Komunikacija med CPE in periferno napravo poteka tako, da imata tako CPE kot periferna naprava bralno/pisalni dostop do vmesnega pomnilnika.

5.5.5. Upravljanje datotečnega sistema

Osnovni gradnik datotečnega sistem predstavlja *datoteka* ali *zbirka*. Zbirke so grupirane v organizacijski strukturi, ki jo imenujemo mapa (angl. *directory*). Mape predstavljajo vozlišča v grafu. Ta graf je običajno urejen v drevesno hierarhijo. Zbirka je logično povezan naslovni prostor (v glavnem ali sekundarnem pomnilniku). Glede na tip ločimo:

- podatkovne zbirke
 - alfanumerične (znaki, števila)
 - binarne
- programe

Struktura zbirke je lahko:

- brez urejene strukture
- preprosta struktura (zapis urejen v vrstice, fiksna dolžina)
- kompleksna struktura (formatirani dokumenti)

Atributi zbirke so naslednji:

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 114 od 219

Nazaj

Full Screen

Zapri

Končaj

- Ime (v človeško berljivi obliki)
- Tip (za OS, ki podpirajo različne tipe)
- Lokacija (kazalec do lokacije zbirke na periferni napravi, npr. disk)
- Velikost (trenutna dolžina zbirke)
- Zaščita (nadzor nad branjem, pisanjem in izvajanjem zbirke)
- Ura, datum in identifikacija uporabnika (podatki za zaščito, varnost in nadzor uporabe)

Podatki o zbirkah so shranjeni v strukturi map, ki se vzdržuje na disku. Tipične operacije OS nad zbirkami so preimenovanje in brisanje.


Najpogostejši tipi zbirk (datotek) so zapisani v tabeli 5.1. Pri ugotavljanju povezave med tipom zbirke in pripadajočim programom si lahko pomagamo s spletnim servisom Filext.com .

Tabela 5.1: Tipi zbirke

Tip zbirke	Končnica	Pomen
besedilo	txt	tekstovni podatki
urejevalnik, oblikovalnik besedil	doc in docx (Ms Word), rtf (večina urejevalnikov), odt (OpenOffice.org Writer, LibreOffice Writer), pages (iWork Pages), tex ($\text{L}^{\text{T}}_{\text{E}}\text{X}$)	različni formati urejevalnikov besedil

nadaljevanje na naslednji strani . . .

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 115 od 219

Nazaj

Full Screen

Zapri

Končaj

Tabela 5.1: nadaljevanje

Tip zbirke	Končnica	Pomen
preglednice	xls in xlsx (MS Excel), ods (OpenOffice.org Calc, Libre-Office Calc)	različni formati programov za delo s preglednicami
arhiviranje	zip, rar, tar, gzip, 7z	več zbirk stisnjenih v eno zbirko, za arhiviranje
zvok, video	mpeg, mov, mp3, m4a (AAC)	binarna zbirka z audio ali audio/video zapisom
slika	bmp, gif, jpg, png, gif, svg	binarna zbirka za zapis slike
geometrijski modeli	dwg, dxf, pln, 3ds, stl, wrl, obj	zapis 2D, 3D geometrijskih modelov
izvorna, programska koda	c, java, pas, bas	izvorna (programska) koda v različnih programskih jezikih
knjižnica	dll, lib	knjižnice funkcij za programerje
izvršna koda	exe, com, bin, bat	računalniški program pripravljen za izvajanje
paket (batch)	bat, sh	paket ukazov za ukazni interpreter

Spletna stran

Naslovnica

Kazalo



Stran 116 od 219

Nazaj

Full Screen

Zapri

Končaj

Kreator/lastnik zbirke lahko odloča o tipu dostopa do zbirke (kdo in kaj lahko počne z zbirko):

- branje (*angl. read*)
- pisanje (*angl. write*)
- izvajanje (*angl. execute*)
- dodajanje (*angl. append*)
- brisanje (*angl. delete*)

Operacije nad mapami so naslednje:

- kreiranje mape
- preimenovanje
- kopiranje
- premikanje
- prikaz vsebine mape

Logična organizacija vsebine mape naj bo takšna, da zagotavlja:

- učinkovitost pri iskanju zbirk
- poimenovanje zbirk, ki je smiselno za uporabnika (dva uporabnika lahko različno poimenujeta enako zbirko, vendar enako poimenujeta dve različni zbirki)

Spletna stran

Naslovnica

Kazalo



Stran 117 od 219

Nazaj

Full Screen

Zapri

Končaj

- grupiranje zbirk po skupnih lastnostih (namenu in vsebini; npr. projektne zbirke, vse igre)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



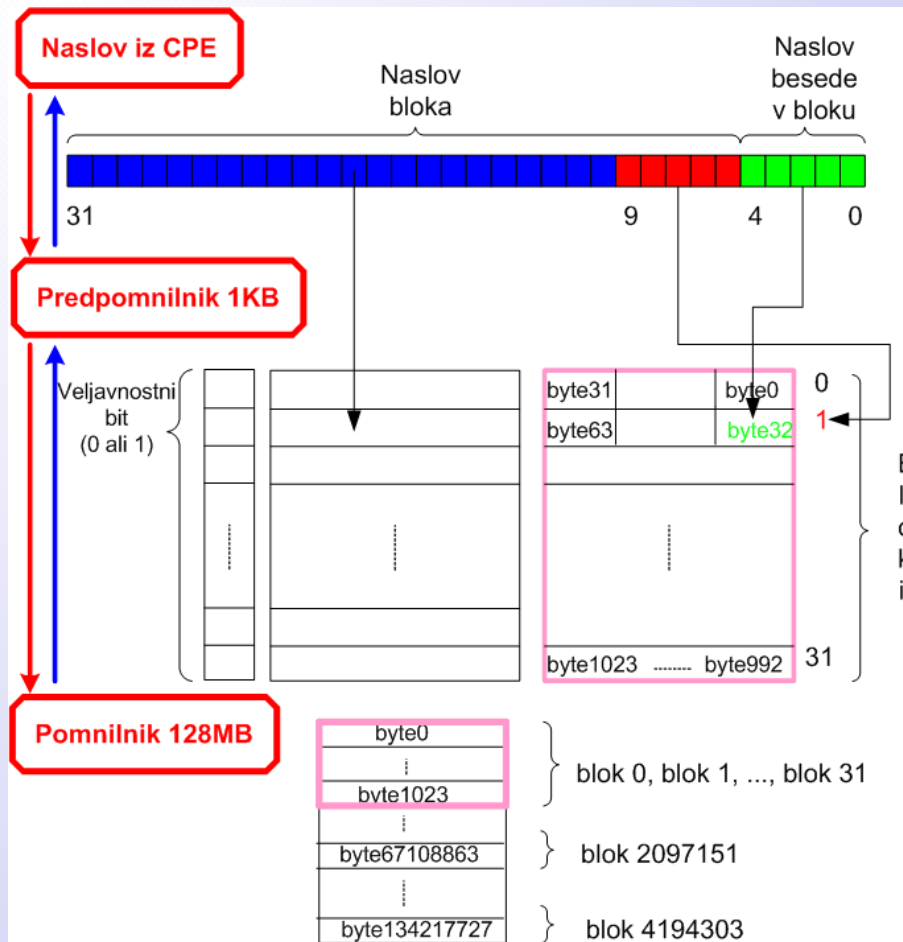
Stran **118** od **219**

[Nazaj](#)

[Full Screen](#)

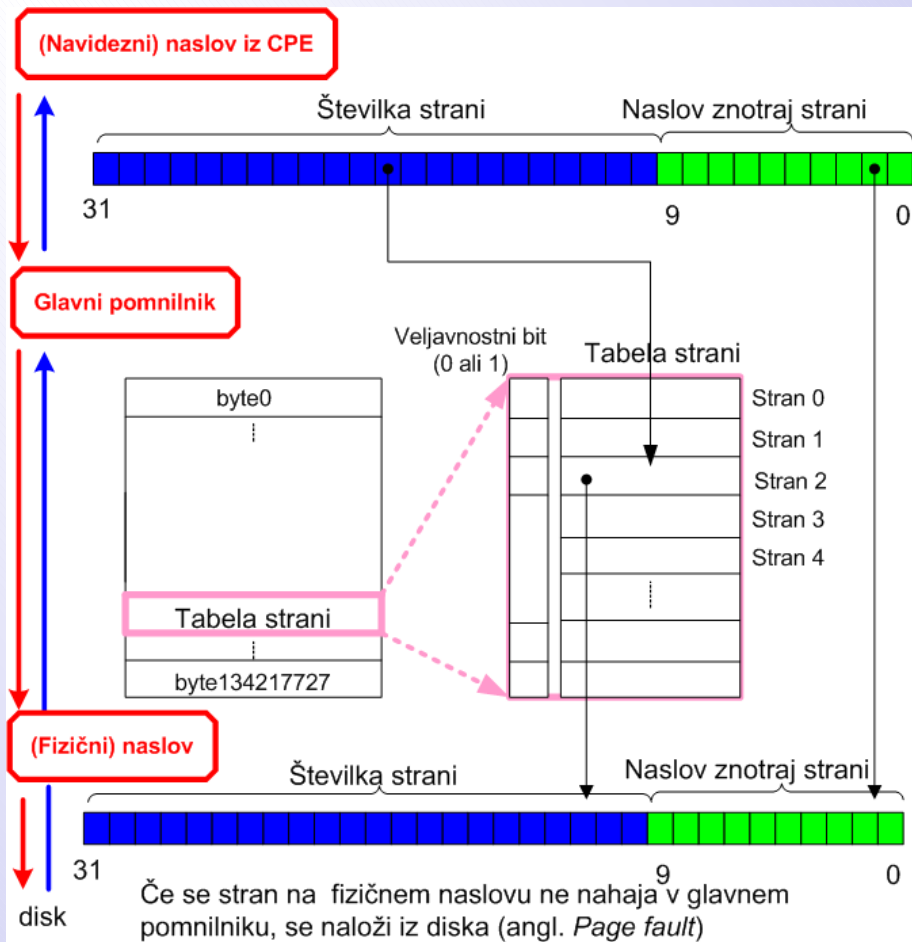
[Zapri](#)

[Končaj](#)



Slika 5.9: Predpomnilnik

- [Spletna stran](#)
- [Naslovnica](#)
- [Kazalo](#)
- [◀](#) [▶](#)
- [◀](#) [▶](#)
- Stran 119 od 219
- [Nazaj](#)
- [Full Screen](#)
- [Zapri](#)
- [Končaj](#)



Slika 5.10: Navidezni pomnilnik

- Spletna stran
- Naslovnica
- Kazalo
- ⏪ ⏩
- ⏴ ⏵
- Stran 120 od 219
- Nazaj
- Full Screen
- Zapri
- Končaj

5.6. Nekateri sodobni operacijski sistemi

V naslednjem podglavju bomo spoznali operacijski sistem GNU/Linux (povzeto po [10]).

5.6.1. Operacijski sistem GNU/Linux

Povprečnemu bralcu se morda zastavlja vprašanje zakaj obravnavamo operacijski sistem Linux? Preprosto zato, ker:

- je o njem čedalje več govora,
- čedalje več podjetij vedoželjno tipa kje v svoji organizacijski shemi bi lahko ta zastonski operacijski sistem uporabili,
- inženirji pri svojih odločitvah morajo znati primerjati različne tehnične rešitve in se odločati z argumenti

Odgovor na vprašanje, kaj je Linux, bi se marsikomu resnično lahko zdel zapleten, saj Linux lahko pomeni več stvari. Edini popolnoma ustrezen pomen je, da je **jedro (angl. kernel) operacijskega sistema**. To pomeni, da je nekakšna **vez med strojno opremo in programi**, ki jih uporabnik potrebuje pri vsakdanjem delu. Samo jedro operacijskega sistema ne vsebuje nobenega žanimivega "programa, zato z njim ne moremo napisati niti preprostega besedila, kaj šele risati, poskušati glasbo, gledati filme ali deskati po internetu. Morda se vam je že zastavilo vprašanje, jedro katerega operacijskega sistema torej je Linux. In ne boste verjeli, Linux je jedro Linuxa. Operacijskemu sistemu kot celoti namreč tudi pravimo Linux, čeprav bi bilo ustrežneje, če bi tak sklop programja imenovali GNU/Linux. A več o tem pozneje. Matematično

Spletna stran

Naslovnica

Kazalo



Stran 121 od 219

Nazaj

Full Screen

Zapri

Končaj

gledano je enako ime za jedro in za celoten sistem vsekakor nenavadno, vendar se je obdržalo iz zgodovinskih razlogov. Zdaj torej že vemo, kaj je Linux in kaj Linux, kljub temu pa nekateri govorijo o “Linuxih”. Kaj pa je to? “Linuxi” ali “Linuksi” ne obstajajo, gre samo za napačno poimenovanje Linuxa, kot posledica poimenovanja operacijskega sistema Windows. Ker je beseda Windows (slov. okna) v angleščini v množini, jo pogovorno včasih postavimo še v slovensko množinsko obliko in dobimo “Windowse”. In ker se tudi Linux fonetično konča s s-jem, kar zveni množinsko, nas to pomotoma napelje v izgovorjavo “Linuksi”. Pomembno je torej, da se zavedamo, da je edina ustrezna beseda Linux in ne “Linuksi” ali kaj podobnega.

5.6.1.1. Kako je nastal Linux?

Kdaj se je Linux pojavil in kako? Okrog leta 1960 so nekatera velika podjetja že imela prve računalnike, ki so bili bistveno večji od današnjih. Med seboj so se zelo razlikovali, saj je imel vsak popolnoma drugačno zgradbo in svoj operacijski sistem. Tako so bili računalniški strokovnjaki usposobljeni za delo samo z enimi takim računalnikom, kar ni bilo pretirano gospodarno.

Leta 1969 pa je iz Bellovih laboratorijev prišel operacijski sistem Unix, napisan v novem programskem jeziku C. Ta je omogočil prenosljivost jedra in tako je Unix kmalu našel svoj dom na večini velikih računalnikov, kjer je ostal dolga leta. Ker si je Unix začel ustvarjati monopol, si je lahko vedno bolj privoščil manipulacijo s cenami. Postajal je vedno dražji, njegovo delovanje in izvorna koda pa čedalje večja skrivnost. Temu se je v začetku 80. uprl talentiran Richard M. Stallman (slika 5.11) in zapustil prestižen oddelek Artificial intelligence laboratories na MIT (Massachusetts Institute of Technology). Nadarjeni programerji so bili namreč v tistem času prisiljeni podpisati stroge pogodbe o varovanju podatkov, saj je bil eden ključnih

Spletna stran

Naslovnica

Kazalo



Stran 122 od 219

Nazaj

Full Screen

Zapri

Končaj

interesov podjetij varovanje izvorne kode njihovih izdelkov.

Stallman je temu močno nasprotoval in leta 1984 začel s projektom zbiranja prostega programja (freeware) za Unixu podoben sistem ter mu nadel ime GNU (GNU's Not Unix!). Napisal je GCC - prevajalnik za C, urejevalnik Emacs in dodal še pravno osnovo prostega programja - GNU GPL (General Public Licence). Ta daje uporabnikom prostega programja štiri temeljne pravice:

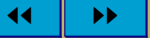
- pravica do poganjanja programa,
- pravica do preučevanja, kako program deluje, in prilagajanja svojim potrebam,
- pravica do razširjanja izvodov,
- pravica do izboljšave programa in javna izdaja izboljšave, tako da pridobi vsa skupnost.

Seveda je zdaj jasno, da je pogoj za zadoščanje licenci GPL, ki jo večkrat označujemo *copyleft* (kot nasprotje od *copyright*), odprtost izvorne kode (angl. *open source*). Vsakdo jo lahko spremeni, kar prispeva k hitrejšemu razvoju programja, ki se v praksi ponavadi izkaže tudi za bolj inovativnega od klasičnega. Pojavile so se tudi drugi izpeljanke *copy...* kot npr. *copydown* (=download a copy, upload an idea). Razvoj prostega programja se je tako lahko začel, saj so bili na voljo prevajalnik, urejevalnik besedil (za programiranje) in pravila igre. Leta 1985 je bila ustanovljena še FSF (Free Software Foundation), ki je s prostovoljnimi sredstvi spodbujala razvoj in število programov je hitro naraščalo. Danes skupina GNU skrbi za skoraj 3000 programov. Tako je bilo v začetku 90. let proste programske opreme že zelo veliko, manjkal pa je ključni del - jedro. Sodelavci projekta GNU so imeli v načrtih modularno jedro z imenom Hurd, a se je njegov razvoj zelo zavlekel, tako da je izšel

Spletna stran

Naslovnica

Kazalo



Stran 123 od 219

Nazaj

Full Screen

Zapri

Končaj

šele pred kratkim časom in še ni pripravljen za stabilno delo in verjetno niti nikoli ne bo. Zazdalo se je že, da GNU nikoli ne bo postal celovit sistem. A ne za dolgo. V tem času je profesor Andrew S. Tanenbaum na Free University of Amsterdam študente učil razvoja operacijskih sistemov, pri čemer si je pomagal z izpeljanko takrat že zaprtega Unixa, z Minixom, ki ga je napisal sam. Operacijski sistem je bil relativno preprost in nič posebnega, imel pa je prosto dostopno izvorno kodo in je bil prilagojen za osebne računalnike.

Spoznaval ga je tudi Linus B. Torvalds (slika 5.12) z univerze v Helsinkih, ki pa nad njim ni bil pretirano navdušen, zato ga je želel izboljšati. Ker je imel pri 21. že precej znanja, je začel iz nič in napisal svojo različico Unixa (Minixa), ki mu je dal ime Linux. 25. avgusta 1991 je na takrat še priljubljeno novičarsko skupino poslal naslednje sporočilo:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict
      Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: 1991Aug25.205708.9541@klaava.Helsinki.FI
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

```
Hello everybody out there using minix
```

```
I'm doing a (free) operating system (just a hobby, won't
  be big and professional like gnu)
for 386(486) AT clones. This has been brewing since
```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 124 od 219

Nazaj

Full Screen

Zapri

Končaj

```
april, and is starting to get ready. I'd like any
feedback on things people like/dislike in minix, as my
OS resembles it somewhat
(same physical layout of the file-system (due to
practical reasons)
among other things).
I've currently ported bash(1.08) and gcc(1.40), and
things seem to work. This implies that I'll get
something
practical within a few months, and I'd like to know
what features most people would want. Any suggestions
are welcome, but I won't promise I'll implement them
:-)
Linus (torvalds@kruuna.helsinki.fi)
```

In kmalu se je začelo nekaj, kar se Linusu (po zgornjem sporočilu sodeč) niti sanjalo ni. Ko je izdelek čez nekaj dni postal prosto dostopen, je GNU dobil tisto, kar mu je manjkalo, torej jedro. Jedro na samem začetku ni bilo pretirano zmogljivo, a to je kmalu postala zgodovina, saj so na pomoč priskočili številni računalniški zanesenjaki in sprožili enega največjih fenomenov v zgodovini računalništva. Vsak je Linuxu dodal, kar je želel, in kvaliteta je rasla iz dneva v dan. Pa tudi priljubljenost. K temu je svoje dodal še Tux (Torvald's UniX), pingvin (slika 5.13), ki je postal maskota Linuxa.

Novi operacijski sistem upravičeno nosi ime otrok interneta, saj je bil internet edini medij, preko katerega je potekala izmenjava vrstic izvorne kode. Danes ga srečamo na najrazličnejših računalnikih, od posebnih ur in mobilnikov do najzmogljivejših strežnikov.

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 125 od 219

Nazaj

Full Screen

Zapri

Končaj

5.6.1.2. Značilnosti Linuxa

Obstaja veliko inačic (distribucij) Linuxa; po Distrowatch.com jih je kar 256, od tega 225 takih, ki se še aktivno razvijajo. Katera distribucija je torej najboljša? To vprašanje nima jasnega odgovora, ravno zato ne, ker se distribucije močno razlikujejo med sabo. Nekatere imajo preprosto namestitve, druge največ pozornosti posvečajo varnosti, spet tretje so prirejene za strežnike, nekatere so poslovenjene, nekatere so plačljive. Izbira je vsekakor velika, kar je dobro, saj si distributerji vsaj na določenih področjih konkurirajo. Za prvo srečanje z Linuxom je primerna distribucija, ki ne zahteva namestitve in jo je možno naložiti iz CDja (angl. boot CD). Komercialni distributerji svoje izdelke prodajajo, saj je treba plačati dokumentacijo, medij (ponavadi CD ali DVD), tehnično podporo ipd. Pravico pa imajo priložiti tudi plačljivo programsko opremo in zanjo zahtevati dodaten denar, zato nekaterih izmed njih ne smemo prosto deliti naokoli. Tipični komercialni distributerji so Red Hat, SuSE, Mandrake, Lindows.com, Xandros. Vendar kljub temu večina komercialnih distributerjev plačljivih programov ne prilaga osnovni distribuciji in razvija ločene izdaje, katerim prilaga plačljive programe. Vse bolj priljubljene pa postajajo tudi posebne distribucije Linuxa, ki jih poženemo z zgoščenke in jih ni potrebno namestiti na trdi disk. Služijo kot izjemen pripomoček za demonstracijo Linuxa ali pa za reševanje sistemov, ki se jih ne da več zbuditi. Najbolj znan primer je Knoppix, ki temelji na Debianu, pri nas pa njegovi izpeljanki, Slo-Tech GNU/Linux in Slix. V Linuxu boste zaman iskali C: A: D: in datoteke s končnico .exe. Datotečni sistem sicer ima drevesno strukturo, vendar so oznake drugačne kot npr. v okolju Windows. Na vrhu sistema je / ali koren (root), pod njim pa glavni imeniki (mape). Imenik bin hrani osnovne programe za ukazno vrstico, v dev se nahajajo naprave, v etc privzete nastavitve, home vsebuje podimenike z imeni uporabnikov, ki predstavljajo njihov

[Spletna stran](#)[Naslovnica](#)[Kazalo](#)[Stran 126 od 219](#)[Nazaj](#)[Full Screen](#)[Zapri](#)[Končaj](#)

dom (podobno kot My Documents v Windows), v usr je večina programov.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **127** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 5.11: Richard Stallman - začetnik gibanja za zastonjsko programsko opremo, *vir*

Spletna stran

Naslovnica

Kazalo



Stran 128 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 5.12: *Linus Torvalds - začetnik Linuxa, vir*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **129** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 5.13: *Maskota Linuxa, vir*

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 130 od 219


[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

5.7. Operacijski sistemi malo drugače

Preberite sproščujočo primerjavo operacijskih sistemov in letalskih družb, ki jo je zapisal neznan avtor. Kaj bi se zgodilo, če bi značilnosti delovanja posameznih operacijskih sistemov preslikali na delovanje letalske družbe? S katero letalsko družbo bi najraje leteli? Presodite sami...().

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **131** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

5.8. Izbirne naloge

Naloga 5.1

Na osebni računalnik namestite eno izmed različic operacijskega sistema Linux. Namestitev naj bo izvedena na osebem računalniku, ki ima že nameščen kak drug operacijski sistem. Napišite natančno navodilo o poteku namestitve in opišite tipične težave, ki se lahko pojavijo med namestitvijo.

Naloga 5.2

Opravite primerjavo med obstoječo uporabniško programsko opremo (pisarniški in inženirski programi) na operacijskem sistemu Windows in Linux. Zanima nas naslednje:

- Kateri so najpogosteje uporabljeni pisarniški programi (urejevalniki besedil, elektronske preglednice, orodja za predstavitve) in kakšna je njihova povezljivost med obema OS?
 - Kateri so najpogosteje uporabljeni inženirski programi na obeh OS?
 - Opravite primerjavo cen pisarniških in inženirskih programov.
 - Ali lahko inženir svoje projekte izvaja na obeh OS, torej neodvisno od OS?
-

Spletna stran

Naslovnica

Kazalo



Stran 132 od 219

Nazaj

Full Screen

Zapri

Končaj

Poglavje 6

Računalniške komunikacije

Potreba po komunikaciji je ena največjih potreb človeka. Nekoč so na daljavo komunicirali izbranci, danes, z zlitjem komunikacijske tehnologije in računalniških sistemov, je elektronska komunikacija dostopna vsakomur. Elektronska komunikacija osvobaja, opogumlja, razveseljuje, pomaga raziskovati in se (žal!) tudi bolje vojskovati. Računalniške komunikacije danes najbolje promovira medmrežje (angl. internet) in njegovi servisi, ki pa ne bi bili možni brez računalniških sistemov (npr. strežniških računalnikov in osebnih računalnikov) kakor tudi ne brez komunikacijskih omrežij (npr. javnega telefonskega omrežja, satelitske komunikacije, optičnih povezav). Zanima nas kako vse to učinkovito deluje (primer: komunikacijska hrbtenica ARNES: slika 6.5, slika 6.6, T-2: slika 6.7)?

To poglavje opisuje teorijo in prakso današnjih računalniških komunikacij.

Cilji poglavja so, spoznati in razumeti:

- osnovne koncepte računalniške komunikacije (analogni/digitalni signali, ko-

Spletna stran

Naslovnica

Kazalo



Stran 133 od 219

Nazaj

Full Screen

Zapri

Končaj

munikacijski protokoli in kontrola napak)

- topologija in arhitektura računalniških omrežij,
- internet in njegovi servisi, komunikacijski protokol TCP/IP
- internetne tehnologije (HTML, CSS)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **134** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

6.1. Osnovni koncepti računalniških komunikacij

6.1.1. Analogni in digitalni signali

Tradicionalni sistemi za komunikacijo (telefon, radio, TV) so za komunikacijo uporabljali analogne signale. Naprave v teh sistemih niso bile predvidene za sprejem digitalnih signalov, ki jih uporabljajo računalniki. Zato je za uporabo npr. telefonskega omrežja za računalniško komunikacijo potrebno vršiti pretvorbo analognih signalov v digitalne in obratno. Razlika med analognimi in digitalnimi signali (slika 6.1) je sledeča:

- Analogni signal lahko opišemo kot zvezno funkcijo z neskončno napetostnimi nivoji (analogni signal v naravi nikoli ni idealno gladek, kljub temu, da se pogosto prikazuje kot sinusna krivulja).
- Digitalni signal opišemo z diskretnimi signali oziroma s končnim številom nivojev (poenostavljen zapis analognega signala). Digitalni signal, ki ga prikazuje slika 6.1 bi lahko zapisali tudi kot zaporedje 1010.

Pretvorbi analognega signala v digitalni signal pravimo *vzorčenje*. Postopek vzorčenja prikazuje slika 6.2.

Analogni signal vzorčimo v določenih časovnih korakih. Digitalni signal na sliki ima 7 napetostnih nivojev. Takšno vzorčenje poteka med komunikacijo iz smeri analogne naprave (npr. modem) do računalnika.

Spletna stran

Naslovnica

Kazalo



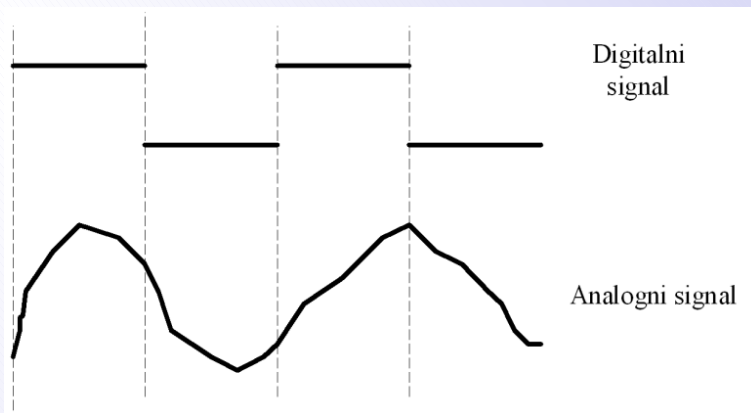
Stran 135 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 6.1: Razlika med analognimi in digitalnimi signali

6.1.2. Multipleksiranje signalov

Pri prenosu analognih in digitalnih signalov lahko prihaja do »mešanja« ali multipleksiranja signalov znotraj skupnega komunikacijskega kanala. To pomeni, da se po skupnem prenosnem mediju (npr. optični kabel, koaksialni kabel) prenašajo podatki iz večih komunikacijskih virov. Povedano preprosteje: multipleksiranje omogoča, da lahko več uporabnikov hkrati komunicira po skupnem kablu. Za boljše razumevanje si oglejmo primer multipleksiranja dveh analognih signalov (zvoč in slika). V eksperimentu na sliki (slika 6.3) imamo dva vira podatkov: kamera je vir za video signal, glasbeni predvajalnik pa je vir za zvočni signal. Škatla zgoraj levo pretvori zvočni signal v rdečo lasersko svetlobo, video pa v zeleno svetlobo. Optična prizma na sredini najprej zlije, nato pa razkloni svetlobo na obe izvorni barvi. Na koncu se video slika prikaže na zaslonu, zvok pa predvaja na zvočniku.

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

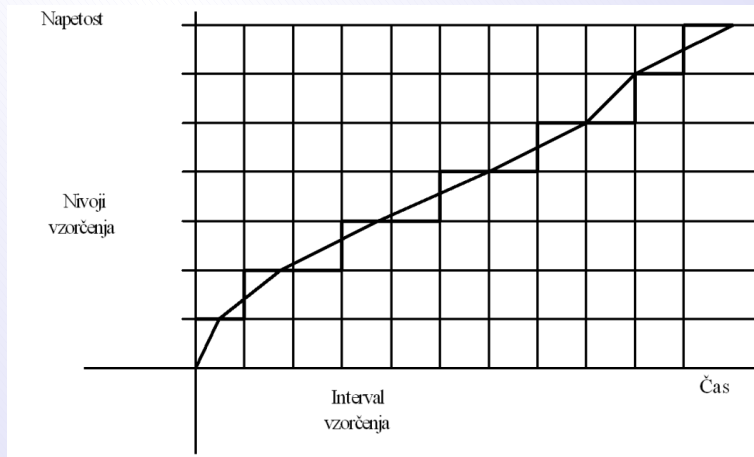
Stran 136 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 6.2: Vzorčenje analognih signalov

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Stran 137 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 6.3: *Multiplexiranje optičnih signalov*

Spletna stran

Naslovnica

Kazalo



Stran 138 od 219

Nazaj

Full Screen

Zapri

Končaj

6.2. Računalniška omrežja

Računalniško omrežje omogoča dostop do programov, podatkov in naprav drugega računalnika. Za vključevanje računalnikov v lokalna računalniška omrežja potrebujemo komunikacijski medij ter ustrezno mrežno strojno in programsko opremo.

6.2.1. Linijsko preklopna omrežja

Med linijsko preklopna omrežja uvrščamo telefonsko omrežje. Telefonsko omrežje uporablja bakreno (žico) parico po kateri poteka telefonska govorna povezava. Za delovanje povezave med uporabnikom A in B je potreben namenski komunikacijski kanal. Telefonsko omrežje lahko uporabimo za vzpostavitev povezave po tehnologiji xDSL. Izvedbe tehnologije xDSL (npr. ADSL, VDSL, SHDSL) se razlikujejo glede na hitrost prenosa. Ponudniki ponujajo pakete VDSL s hitrostjo do 60/25 Mb/s in pakete ADSL s hitrostjo do 20/1 Mb/s. Take hitrosti so mogoče le na lokacijah, ki so oddaljene od centrale manj kot 500 m, z večjo oddaljenostjo pa najvišje hitrosti prenosa padajo. Na redko poseljenih območjih so uporabniki pogosto zelo oddaljeni od centrale. S tem je onemogočena uporaba tehnologij xDSL, ostane le še možnost povezave s podatkovno povezavo ISDN [11].

6.2.2. Paketno preklopna omrežja

Med paketno preklopna omrežja prištevamo digitalna računalniška omrežja. Digitalna računalniška omrežja uporabljajo optična vlakna (najvišja hitrost in največja zanesljivost pri prenosu podatkov) in koaksialni kabel (omogoča združljivost z obstoječim omrežjem kableske televizije). Za paketno preklopna omrežja je značilno,

Spletna stran

Naslovnica

Kazalo



Stran 139 od 219

Nazaj

Full Screen

Zapri

Končaj

da se podatki večji od določene mejne vrednosti pred pošiljanjem "razbijejo" na manjše pakete. Za delovanje povezave med uporabnikom A in B ni potreben namenski komunikacijski kanal. Delovanje paketno preklopnega omrežja ponazarja naslednja **animacija**.

6.2.3. Struktura omrežij

Računalniška omrežja so urejena hierarhično, od manjših do velikih:

- LAN (angl. Local Area Network)
omrežje znotraj neke organizacije, npr. LAN na UNI MB, slika 6.4
- WLAN (angl. Wireless Local Area Network)
brezžično lokalno omrežje
- MAN (angl. Metropolitan Area Network)
omrežje na večjem področju, npr.: omrežje na področju metropole, omrežje mobilne telefonije
- WAN (angl. Wide Area Network)
omrežje med mesti
- GAN (angl. Global Area Network)
omrežje med kontinenti

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

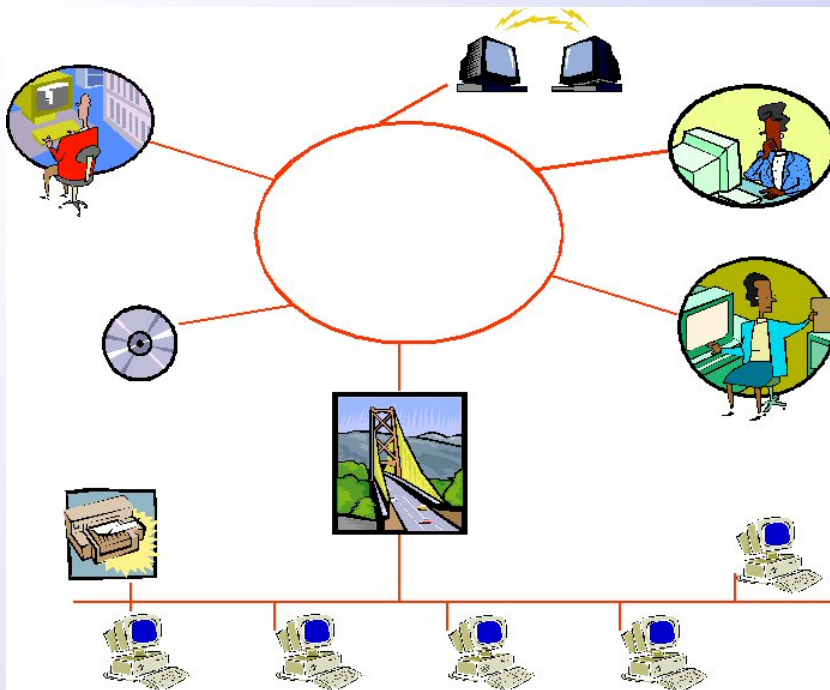
Stran 140 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 6.4: Primer lokalnega omrežja (LAN)

6.2.3.1. LAN - lokalno omrežje

Topologija LAN omrežij:

- krožna
- linijska

Spletna stran

Naslovnica

Kazalo



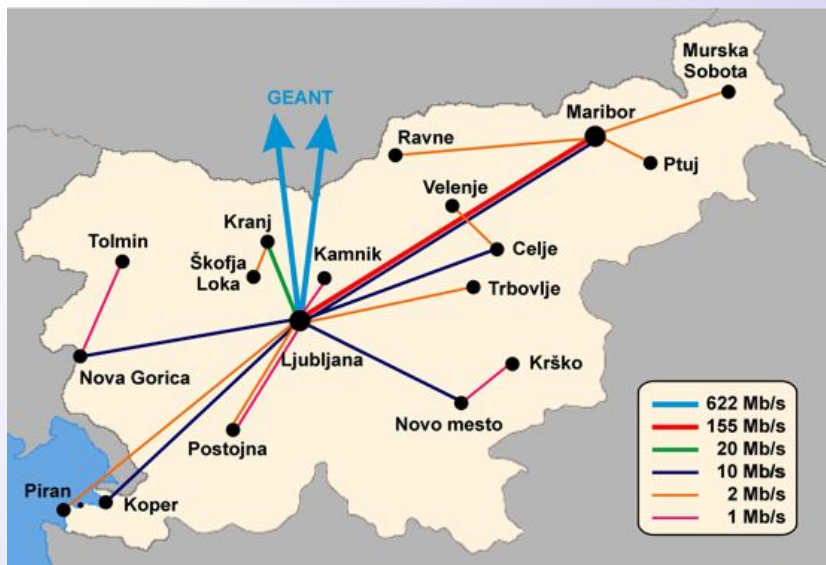
Stran 141 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 6.5: Komunikacijsko omrežje ARNES, 2003 [3]

- zvezdasta

6.2.4. Parametri računalniškega omrežja

6.2.4.1. Hitrost komunikacije

$$\text{Minimalna Pričakovana Hitrost} = \frac{\text{Hitrost Komunikacijske Hrbtenice}}{\text{Število Uporabnikov}} \quad (6.1)$$

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

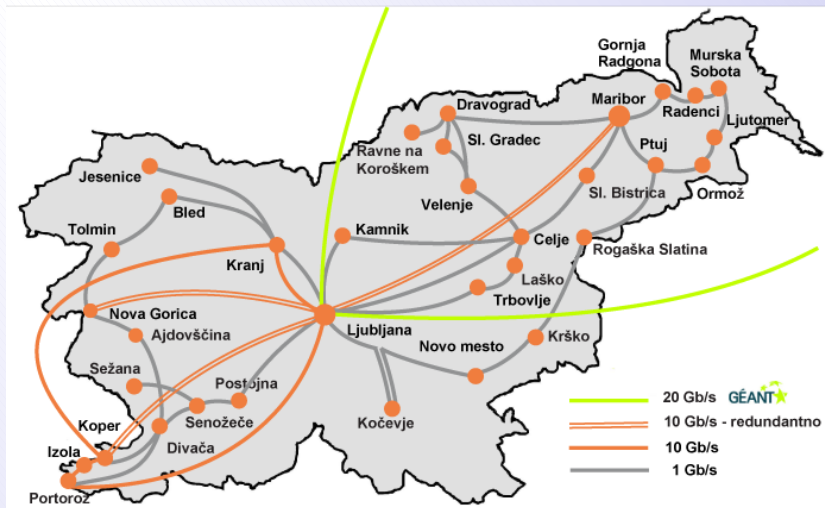
Stran 142 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 6.6: Komunikacijsko omrežje ARNES, 2011 [4]

Primer :

- omrežna hrbtnica s hitrostjo 100 Mb/s
- 24 uporabnikov
- Minimalna pričakovana hitrost = 4.1 Mb/s

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 143 od 219

Nazaj

Full Screen

Zapri

Končaj

T2 Shema omrežja
(na dan 30. 6. 2009)



Slika 6.7: Komunikacijsko omrežje T-2, 2009 [5]

6.2.5. Komunikacijski protokoli

6.2.6. Standardi za komunikacijo

6.2.6.1. ISO/OSI model

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 144 od 219

Nazaj

Full Screen

Zapri

Končaj

6.3. Medmrežje (internet)

Medmrežje ali omrežje vseh omrežij (angl. *internet*).

6.3.1. Naslavljanje v medmrežju

6.3.2. Medmrežni servisi

- Svetovni splet (WWW, WAP)
- Terminalsko delo (Telnet)
- Prenos datotek (FTP)
- Elektronska pošta (SMTP, POP3, IMAP)
- Direktna komunikacija (IRC, Sporočilni sistemi, Videokonference)
- Novičarski sistemi (NNTP)...nadomešča jih WWW

6.3.3. Nekateri “omrežni” ukazi

- Moj IP naslov: IPCONFIG (OS Windows) ali IFCONFIG (OS Linux)
- IP naslov \Leftrightarrow IP ime: NSLOOKUP
- IP naslov dostopen: PING

Spletna stran

Naslovnica

Kazalo



Stran 145 od 219

Nazaj

Full Screen

Zapri

Končaj

- Sledenje IP povezavi: TRACERT (OS Windows) ali TRACEROUTE (OS Linux)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **146** od **219**

[Nazaj](#)

[Full Screen](#)

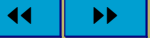
[Zapri](#)

[Končaj](#)

6.4. Tehnologije za medmrežje

Brez tehnologij, ki omogočajo izdelavo vsebin, bi bila računalniška omrežja le skuppek računalniške in komunikacijske opreme. Te tehnologije inženirjem omogočajo objavo rezultatov projektov na medmrežju, *povezovanje* inženirjev in *sodelovanje* inženirske programske opreme. Učinkovito sodelovanje programske opreme s pomočjo tehnologij za medmrežja odpravlja otoke avtomatizacije (slika 3.2).

Poglejmo primer, ki pa tokrat ni iz inženirskega področja. Pri brskanju po internetu velikokrat naletimo na spletno stran, ki nas tako navduši, da bi želeli nekaj podobnega narediti tudi sami. Vendar kaj kmalu ugotovimo, da naše znanje jezika HTML ni dovolj bogato, da bi takšni nalogi lahko bili kos. Kako kljub temu priti do navdušujočega rezultata? Od inženirjev ne-računalniške stroke se običajno ne pričakuje, da bi bili večji v izdelavi programskih rešitev za računalnik. Vendar pa bi moralo vsakega dobrega inženirja odlikovati solidno razumevanje področja tehnologij za računalniška omrežja. Razumevanje mora biti na nivoju praktične iznajdljivosti na posameznih področjih računalništva. Če se pogovarjamo o naprednem oblikovanju spletnih dokumentov, moramo razumeti katere tehnologije (tabela 6.1) lahko uporabimo za izdelavo spletnih dokumentov, kako in kje jih lahko uporabimo, ter kako strma je krivulja učenja za posamezno tehnologijo.

[Spletna stran](#)[Naslovnica](#)[Kazalo](#)

Stran 147 od 219

[Nazaj](#)[Full Screen](#)[Zapri](#)[Končaj](#)

Tabela 6.1: Spletne tehnologije

Oznaka tehnologije	Opis	Licenca
HTML (HyperText Markup Language)	označevalni jezik za opis spletnih strani (dokument HTML, .html)	W3C standard
CSS (Cascading Style Sheets)	jezik za opis sloga dokumenta HTML	W3C standard
Javascript	skriptni jezik, ki dokument HTML obogati z interaktivnimi elementi	-
DHTML (Dynamic HyperText Markup Language)	DHTML ni jezik, ampak pristop s katerim kombiniramo HTML, CSS in Javascript, da bi izdelali dinamične in interaktivne spletne strani	W3C standard

nadaljevanje na naslednji strani . . .

Spletna stran

Naslovnica

Kazalo



Stran 148 od 219


Nazaj

Full Screen

Zapri

Končaj

Tabela 6.1: nadaljevanje

Oznaka tehnologije	Opis	Licenca
Adobe Flash	sistem za izdelavo večpredstavnostnih vsebin v zapisu SWF (ShockWave Flash, .swf), ki jih lahko vključimo v spletne strani, predvajanje vsebin zahteva uporabo predvajalnika Adobe Flash Player	
PHP	PHP je splošno namen-ski skriptni jezik, ki je posebej primeren za razvoj spletnih aplikacij. Lahko ga vključujemo v dokumente HTML. 	
ASP (Active Server Pages)		Microsoft

nadaljevanje na naslednji strani . . .

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 149 od 219


[Nazaj](#)




[Full Screen](#)

[Zapri](#)

[Končaj](#)

Tabela 6.1: nadaljevanje

Oznaka tehnologije	Opis	Licenca
Perl	Stabilen programski jezik za več operacijskih sistemov (platform). Uporablja se za izdelavo programskih rešitev in je pogosta izbira za izdelavo spletnih aplikacij. 	odprta koda
XML (eXtended Markup Language)	razširljiv označevalni jezik	W3C standard

Denimo, da vam je pri brskanju po internetu oko zastalo na spletni strani na kateri po ekranu poplesujejo snežinke  za **MS Internet Explorer**. Zato bi jih želeli dodati k svojemu spletnemu dokumentu . Kako bi jih lahko dodali na lastno spletno stran? Najprej je potrebno pridobiti nekaj znanja o samih internetnih tehnologijah, da bi lahko potem lažje ocenili na kakšen način lahko kaj takega sploh naredimo. Oboroženi s takšnim praktičnim znanjem (kar ne pomeni nujno tudi razumevanjem) velikokrat ugotovimo, da je možno nek problem rešiti z uporabo že izdelane rešitve (npr. programske kode). Tako bo tudi v primeru naših snežink, ki so izdelane v programskem jeziku *Javascript*. če vemo, da je rešitev izdelana v jeziku Javascript, nam postane jasno, da lahko takšno kodo tudi vidimo, če pogledamo v notranjost dokumenta HTML .

V naslednjem koraku bomo to kodo (med komentarjem ZACETEK in KONEC)

Spletna stran

Naslovnica

Kazalo




Stran 150 od 219

Nazaj

Full Screen

Zapri

Končaj

dodali k našemu spletnemu dokumentu. Dobimo zanimiv in dinamičen spletni dokument .

6.4.1. HTML

Ideja jezika HTML (Hyper Text Markup Language) je bila na začetku zelo skromna. Avtor prve različice jezika HTML, Tim Berners-Lee, je oblikoval enostaven označevalni jezik. Za osnovo mu je služil jezik SGML (Standard Generalized Markup Language), s katerim je moč opisati poljuben označevalni jezik.

Pričakoval je, da se bo v prihodnosti razvilo več različnih označevalnih jezikov, primernih za različne namene, in da se bodo ti prenačali s pomočjo "pametnih" odjemalcev in strežnikov preko omrežja. Tak mehanizem bi deloval na poljubnem sistemu (platformi) in s poljubnim pregledovalnikom.

Značilnost jezika HTML je enostavnost. Ker je bil jezik HTML zasnovan tekstovno z minimalnim številom oznak, se ga lahko vsakdo hitro nauči in s poljubnim urejevalnikom besedila napiše lasten dokument.

Naenkrat so vsi začeli objavljati lastne HTML dokumente in splet se je razcvetel. Z rastjo spleta so tudi naraščale potrebe po dodatnih oznakah v jeziku HTML. Ker seveda Tim Berners-Lee ni bil pripravljen prevzeti nadaljnega razvoja jezika HTML, so to storili proizvajalci pregledovalnikov. Z izdajo novih različic pregledovalnikov so proizvajalci dodajali nove oznake. Te oznake so se obdržale oz. izginile glede na okus in potrebe uporabnikov.

Takšen razvoj se nadaljuje in postaja vse bolj kaotičen. Veliki proizvajalci pregledovalnikov so razvijali lastne različice jezika HTML, vse dokler se v dogajanje ni vmešal konzorcij W3C (World Wide Web Consortium) in oblikoval enoten HTML standard (HTML 3.2) in tako prevzel skrb za razvoj HTML standarda.

Spletna stran

Naslovnica

Kazalo



Stran 151 od 219


Nazaj

Full Screen

Zapri



Končaj

6.4.2. CSS

Jezik HTML je v izvorni različici vseboval le oznake (značke) za opis vsebine (npr. `<head>`, `<body>` in `<p>`). Kasneje so prizvajalci dveh največjih medmrežnih brskalnikov uvedli tudi oznake za opis oblike (predstavitve) dokumenta HTML (npr. `<color>` za barvo in `` za pisavo). Konzorcij World Wide Web Consortium (W3C, ) je zato predlagal koncept *ločitve oblike od vsebine*, ki bi omogočal večjo fleksibilnost dokumentov HTML, in definiral standard CSS (angl. Cascading Style Sheets). Standard CSS je tehnologija za opis oblike dokumentov HTML. V nadaljevanju si oglejmo tri različne načine uporabe slogov:

- zunanji slog
- notranji slog
- vrstični slog

6.4.2.1. Zunanji (eksterni) slog

Zunanji slog, v obliki ločenega dokumenta CSS (npr. `zunanjiSlog.css`) je najboljša izbira, če želimo slog uporabiti za več dokumentov HTML ali če želimo pogosto menjati obliko (izgled) dokumenta HTML. Inženir, ki objavlja rezultate projekta na spletnem strežniku, bo poročilo zapisal v dokumentu HTML in ga povezal z dokumentom CSS. Dokument CSS lahko izdelamo “iz nič” ali pa ga poiščemo na medmrežju (). Spodnji primer () prikazuje vsebino `zunanjiSlog.css`, ki definira slog naslova (h1), odstavka (p) in ozadja telesa dokumenta HTML (body):

Program 6.1: Dokument CSS z definicijami slogov



Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 152 od 219

Nazaj

Full Screen

Zapri

Končaj


```

1  body { background-image: url(snezinka.gif);
    }
3
5  h1 { font-size: 16pt;
    background-color: rgb(238, 238, 238);
    }
7
9  p { background-color: rgb(238, 238, 238);
    color: rgb(255, 0, 0);
    line-height: 20pt;
11 font-size: 12pt;
    }

```

V dokument HTML smo dodali element z oznako *<link>*, ki vsebuje pot do datoteke z definicijami slogov (zunanjiSlog.css):

Program 6.2: Dokument HTML z zunanjim slogom CSS

```

1 <html>
2 <head>
    <link rel="stylesheet" href="zunanjiSlog.css"
4         type="text/css">
    <title>CSS: uporaba zunanjega sloga</title>
6 </head>
    <body>
8     <h1>Oblikovanje vsebine dokumenta HTML z uporabo
        slogov CSS</h1>
10    <p>Zunanji slog: definicije sloga so zapisane v
        zunanjem dokumenta CSS</p>

```

Spletna stran

Naslovnica

Kazalo



Stran 153 od 219

Nazaj


Full Screen

Zapri

Končaj

```
12 </body>
</html>
```

6.4.2.2. Notranji slog

Notranji (interni) slog zapišemo v glavi dokumenta HTML (uporabimo *<style>*). Tako zapisan slog je “zapečen” v dokument HTML. Uporaba notranjega sloga je smiselna, če zahtevamo enkratno (unikatno) obliko, ki je značilna samo za ta dokument HTML. Vsak posamezni element vsebine (npr. odstavek, *<p>*) v dokumentu je oblikovan kot je opisan v slogu, t.j. oblika je definirana na nivoju celotnega dokumenta HTML. Vsaka ponovitev uporabe tega sloga v drugem dokumentu zahteva prepis celotnega sloga kar je zamudno opravilo, ki obenem poveča tudi možnost napak. Brskalnik bo med interpretiranjem dokumenta HTML prebral definicije sloga in vsebino ustrezno prikazal. Primer prikazuje uporabo notranjega sloga (

Program 6.3: Primer uporabe notranjega sloga

```
1 <html>
  <head>
3     <style type="text/css">
      body { background-image: url("snezinka.gif");}
5     h1 { font-size: 16pt;
        background-color : rgb(238, 238, 238);}
7     p { background-color : rgb(238, 238, 238);
        color: rgb(255, 0, 0);
9         line-height: 20pt;
        font-size: 12pt;}
11    </style>
```

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 154 od 219

Nazaj

Full Screen

Zapri

Končaj

```


13 <title>CSS: Uporaba notranjega sloga</title>
    </head>
    <body>
15     <h1>Oblikovanje vsebine dokumenta HTML z uporabo
        slogov CSS</h1>
17     <p>Notranji slog: definicije sloga so zapisane v
        glavi dokumenta HTML</p>
19 </body>
</html>

```

6.4.2.3. Vrstični slog

Uporaba vrstičnega sloga popolnoma

pomeša

vsebino in obliko dokumenta HTML. Na ta način lahko vsak element dokumenta oblikujemo na svoj način, t.j. oblikovanje je na nivoju posameznega elementa (npr. odstavka, *<p>*). Vrstični slog uporabljamo izjemoma, njegova uporaba namreč poveča nepreglednost med urejanjem dokumenta. Spreminjanje oblike takšnega dokumenta je zelo zamudno. Primer prikazuje uporabo vrstičnega sloga (

Program 6.4: Vrstični slog

```

1 <html>
2   <head>
3     <title>CSS: Uporaba vrstičnega sloga</title>
4   </head>
5   <body style="background-image: url(snezinka.gif);">

```

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 155 od 219

Nazaj

Full Screen

Zapri

Končaj

6.5. Povzetek

- Računalniške komunikacije temeljijo na enakih konceptih kot človeška komunikacija
- Komunikacijski kanali imajo kapaciteto in so lahko multipleksirani
- Protokoli morajo biti definirani
- Standardi poenostavijo uporabo komun. sistemov

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 157 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

6.6. Izbirne naloge

Naloga 6.1

Predpostavite, da imate doma 3 računalnike, ki jih želite povezati v lokalno omrežje in internet. Omrežje naj omogoča delitev virov med tremi računalniki (tiskanje, dostop do podatkov). Opišite čimveč možnih načinov vzpostavitve lokalnega omrežja z dostopom do interneta. Vsak posamezni način naj bo dokumentiran z grafično shemo, ki prikazuje vse naprave, ki jih potrebujemo.

Naloga 6.2

Spletne strani (zapis HTML) prištevamo k internetnim tehnologijam. Izdelajte spletno fotogodbo pri čemer boste demonstrirali uporabo poljubnega programa za izdelavo spletnih strani, programa za obdelavo slik, itd. Pojasnite katere HTML gradnike ste uporabili (npr. naslove, tabele, itd.) za oblikovanje spletne strani. Izdelajte predstavitev naloge.

Spletna stran

Naslovnica

Kazalo



Stran 158 od 219

Nazaj

Full Screen

Zapri

Končaj

Poglavje 7

Računalniška grafika

Računalniška grafika je zastavonoša računalništva in informatike. To je področje, ki vključuje programerje in uporabnike, ter grafične aparaturne naprave. Dober programer mora dobro poznati teorijo računalniške grafike, da bi lahko izdelal dobre programe za inženirje. Dober uporabnik mora za učinkovito delo razumeti osnovne koncepte računalniške grafike. Grafične aparaturne naprave so nepogrešljivi del za shranjevanje in prikaz računalniške slike.

Cilji poglavja so, spoznati in razumeti:

- koncept barve in njene značilnosti,
- barvne modele,
- rastrsko in vektorsko grafiko,
- osnovne grafične gradnike,

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 159 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

- prikaz računalniške slike.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **160** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

7.1. Svetloba in barve

Teorija barv je znanstvena disciplina z začetki okoli leta 1666, ko je Isaac Newton izvedel prve eksperimente z barvami. Bil je prvi, ki je razumel mavrico. Z **eksperimentom** in dvema zaporednima prizmama je najprej belo svetlobo s prizmo razcepil na nekatere njene barvne sestavine (rdečo, oranžno, rumeno, zeleno, modro in vijoličasto) in jih nato spet zliil. Tako je Newton ovrgel prepričanje, da so barve kombinacija "dneva in noči".

7.1.1. Elektromagnetni spekter

Elektromagnetni spekter (slika 7.1) obsega valovne dolžine od stotinke nanometra (gama žarki) do enega metra ali več (radijski valovi). Človek brez pomoči naprav ni sposoben zaznati celotnega spektra. S pomočjo spektrofotometričnih naprav, ki imajo ločljivost 1-3nm, lahko izmerimo intenzivnost posameznih valovnih dolžin v spektru (npr. elektromagnetno valovanje, ki prihaja iz vesolja). Edino področje, ki ga človek lahko zazna, je področje vidne svetlobe med valovnimi dolžinami od 400 (**modra svetloba**) do 700 (**rdeča svetloba**) nanometrov. Svetloba je elektromagnetno valovanje. Barvo svetlobe določa njena valovna dolžina.

V naravi je najpogosteje prisotna bela svetloba (za razliko od npr. rdeče, modre in zelene svetlobe, ki jih najpogosteje ustvarimo z barvastimi svetili). Odkod torej predmetom v naravi barva, če so obsijani z belo svetlobo. Barva objekta nastane z absorpcijo določenih valovnih dolžin svetlobe. Natančneje, objekt absorbira vse barve razen tiste, ki jo vidimo kot barvo objekta (slika 7.2). Moder objekt osvetljen z belo svetlobo absorbira vse valovne dolžine razen tiste, ki predstavlja modro barvo (400nm). Modra barva (oz. elektromagnetno valovanje z valovno dolžino 400nm) se

Spletna stran

Naslovnica

Kazalo



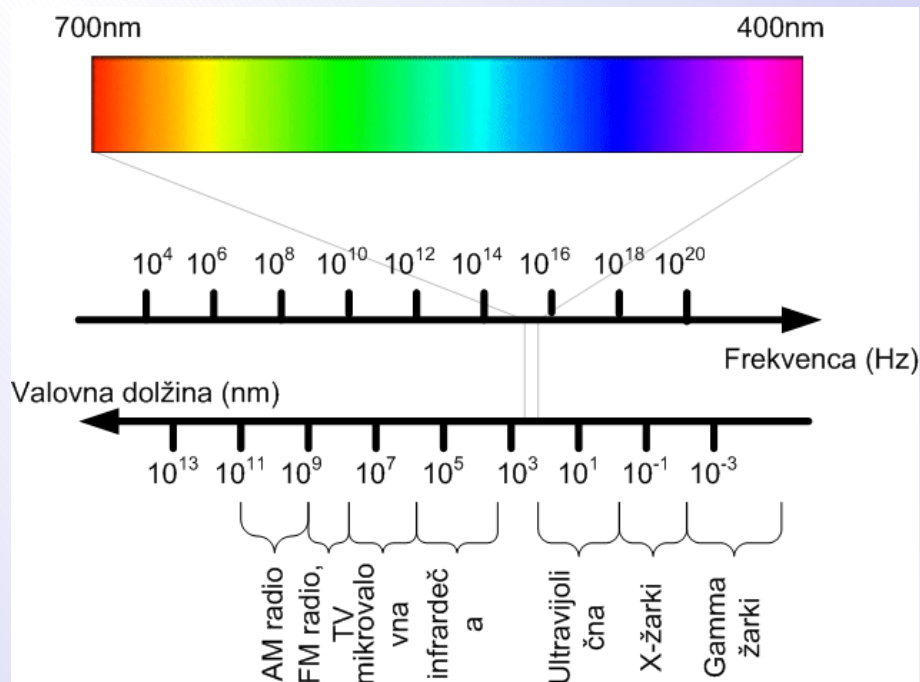
Stran 161 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.1: Elektromagnetni spekter

odbije od objekta.

Človeško oko lahko zazna okoli 10 milijonov barv. Oko ima tri tipe receptorjev v zadnjem delu očesa, ki se imenuje *retina*. Ti receptorji se odzivajo na svetlobo različnih valovnih dolžin. Možgani dobijo torej tri signale za vsak spekter, ki ga vidimo. Spekter se nato kot dražljaj na treh tipih receptorjev (prvi so občutljivi na oranžno-rdečo

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

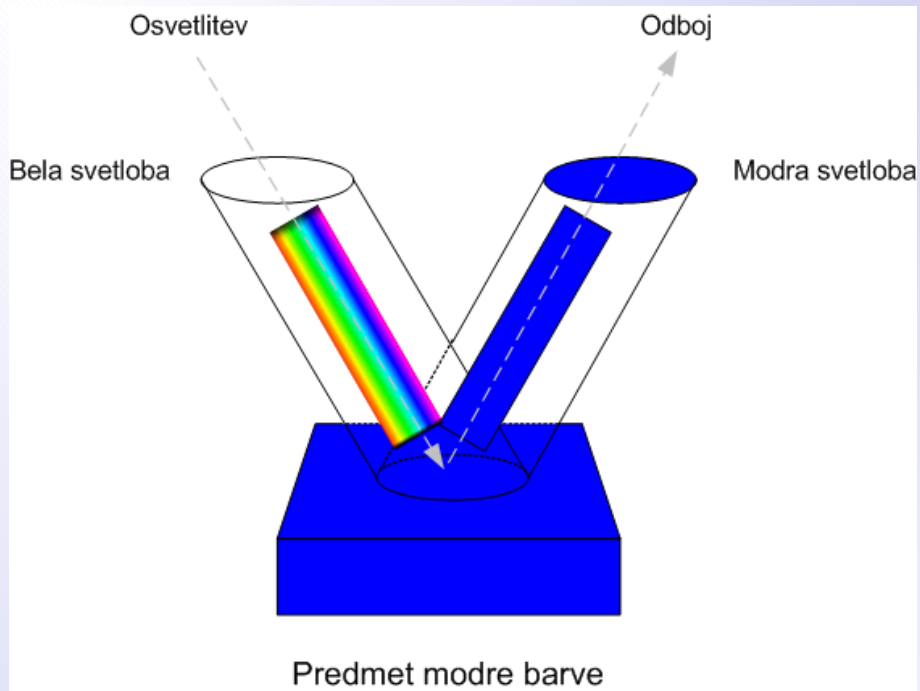
Stran 162 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.2: Elektromagnetni spekter

barvo, drugi na zeleno-rumeno in tretji na modro) prenese v možgane, ki dražljaj ovrednotijo kot barvo. Torej lahko zaznano barvo predstavimo kot trojico števil, npr. rdeča, zelena in modra.

Spletna stran

Naslovnica

Kazalo



Stran 163 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.3: Slika zapisana v barvah modela RGB

7.1.2. Predstavitev barv z računalniškimi napravami

Z računalniško grafiko se ukvarjajo nezahtevni uporabniki, programerji, arhitekti, gradbeniki, prometniki, idr. Pri svojem delu uporabljajo različne naprave za prikaz računalniške grafike (npr. zasloni, projektorji, tiskalniki, digitalizatorji, digitalne kamere). Želja vsakega od naštetih skupin uporabnikov je ustvariti lep in prepričljiv

Spletna stran

Naslovnica

Kazalo



Stran 164 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.4: Slika zapisana v barvah modela CMYK

prikaz. Vsaka slika ni primerna za vsako napravo, zato je lahko računalniško sliko optimiramo za napravo na kateri jo želimo prikazati. Spodnji dve sliki se razlikujeta v podrobnostih; slika 7.3 je svetlejša kot slika 7.4. Prva slika je zapisana v barvah modela RGB, druga pa v barvah modela CMYK. Poznamo pa še tretji barvni model HSB.

Spletna stran

Naslovnica

Kazalo



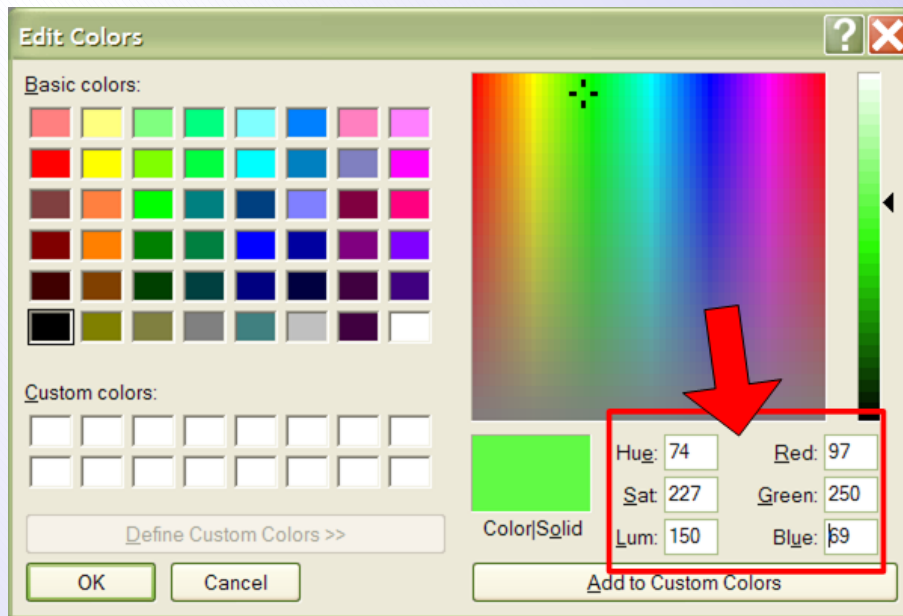
Stran 165 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.5: Določitev barve z uporabo barvnega modela RGB ali HSV (HSL)

Kje se v praksi srečamo z barvnimi modeli? Pri določanju barv v računalniških programih. V OS Windows lahko to preverimo s programom Risar (angl. Paint, slika 7.5) ali s priloženim programom **CMYK Calculator** (slika 7.7). V Mac OS X lahko uporabimo program Paintbrush (slika 7.6).

Kaj so torej barvni modeli? Spoznajmo njihove značilnosti.

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

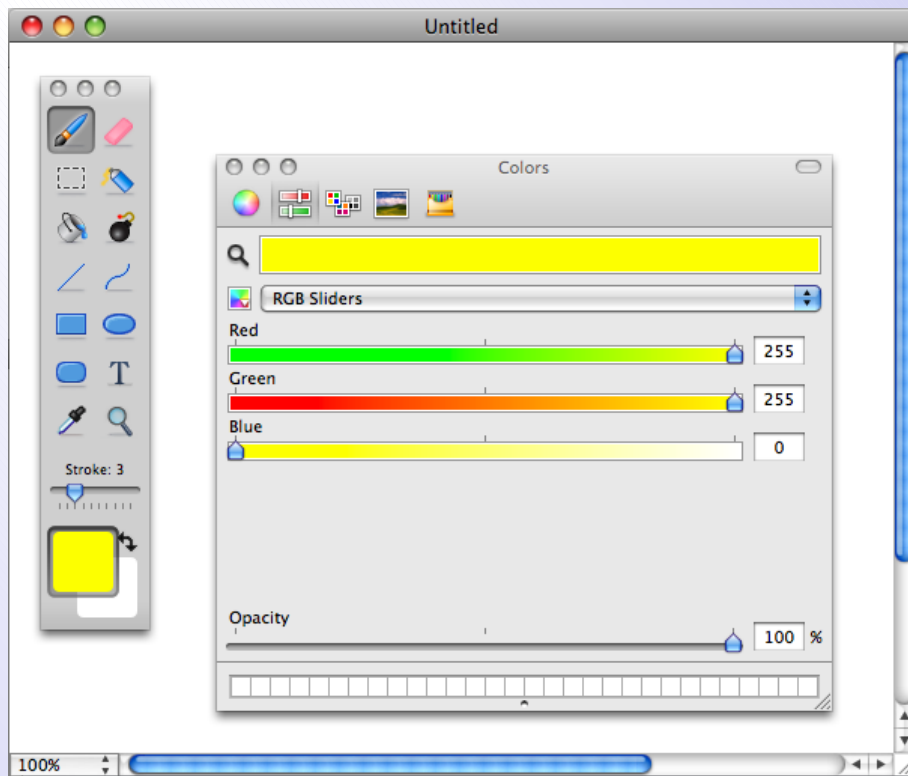
Stran 166 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.6: Določitev barve po RGB s programom Paintbrush (Mac OS X)

7.1.2.1. Barvni model HSB

Barvni model HSB definira barvni prostor s tremi značilnostmi:

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Stran 167 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

- Hue (v prevodu: barvni ton, barvni odtenek)
- Saturation (v prevodu: nasičenost)
- Brightness (v prevodu: svetlost)

Barvni model HSB označujemo tudi kot HSL (angl. Hue Saturation Lightness) ali HSV (angl. Hue Saturation Value). Za boljše razumevanje lahko barvni model HSB predstavimo s šeststrano piramido (slika 7.8).

7.1.2.2. Barvni model RGB

RGB je okrajšava za angl. **Red**, angl. **Green** in angl. **Blue** (pravimo jim tudi trije barvni kanali - rdeč, zelen in moder). Barve modela RGB prikazujejo naprave, ki za prikaz slike na prikazovalni površini mešajo različno intenzivne svetlobne curke rdeče, zelene in modre barve. Barvam modela RGB pravimo tudi primarne *seštevne barve*. Med naprave s to tehnologijo prikazovanja barv prištevamo prikazovalnike s katodno cevjo (angl. CRT - Cathode Ray Tube): televizorji, starejši računalniški ekrani. Ker tovrstne naprave prikazujejo barve s svetlobo, se slika s povečevanjem intenzivnosti rdeče, zelene in modre svetlobe spreminja od temnejše (črne) do svetlejšje (bele).

Z barvnim modelom RGB lahko na prikazovalniku prikažemo 16.777.216 barv. Zakaj? Vsako posamezno seštevno barvo lahko prikažemo v 256 odtenkih (od 0 do 255), kar ustreza zapisu posamezne barve z 8 biti ($2^8 = 256$). Skupno število barv je enako številu vseh možnih kombinacij odtenkov primarnih barv, $256 * 256 * 256 = 256^3 = 16.777.216$. Zapisu barve v barvnem modelu RGB pravimo tudi 24-bitni zapis barve (2^{24}). Barvni prostor modela RGB lahko predstavimo kot kocko (slika 7.9).

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 168 od 219

Nazaj

Full Screen

Zapri

Končaj

7.1.2.3. Barvni model CMYK

Naprave, ki uporabljajo barvni model CMYK, sliko ustvarijo z nanašanjem osnovnih (pigmentov) barv (C kratica za angl. **Cyan** - cianovo modra, M kratica za angl. **Magenta** - vijolična, Y kratica za angl. **Yellow** - rumena in K kratica za angl. **Black** - črna), zato z dodajanjem barv slika postaja temnejša.

Tudi barvni prostor modela CMYK lahko (tako kot RGB) predstavimo kot kocko (slika 7.9). Glede na razporeditev osnovnih barv v ogliščih kocke lahko ugotovimo, da sta barvna modela RGB in CMYK komplementarna (oziroma so komplementarne njune osnovne barve). To pomeni, da s seštevanjem barv modela RGB dobimo barve modela CMYK:

- Rumena = Rdeča + Zelena,
- Cianovo modra = Modra + Zelena,
- vijolična = Modra + Rdeča.

Tudi barvni prostor modela CMYK lahko, tako kot RGB, predstavimo kot kocko (slika 7.9).

Vsi opisani barvni modeli delujejo nad skupnim barvnim prostorom. Zaradi tega lahko določeno barvo zapišemo v vseh barvnih modelih. Primerjavo barvnih modelov RGB, CMYK in HSB in njihovih barvnih prostorov v 3D (kocka, šeststrana piramida) lahko preizkusimo s **programom**. Analogija barvnih prostorov z geometrijskimi telesi nam olajša razumevanje barvnih modelov.

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

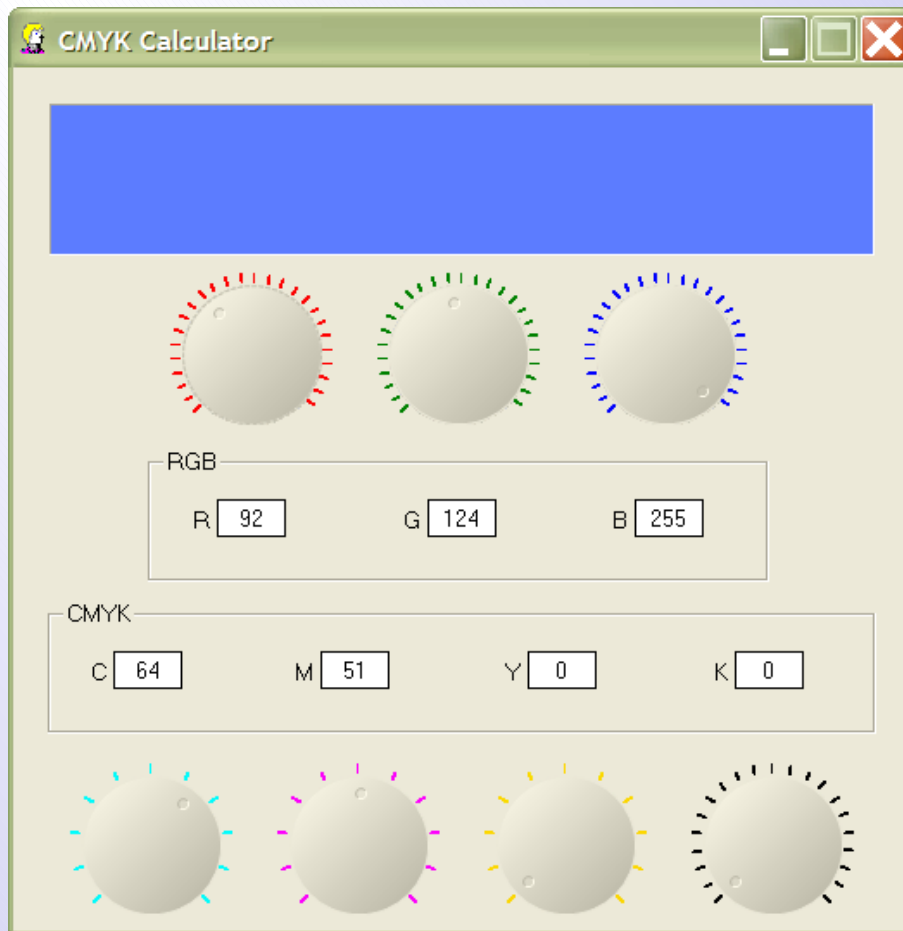
Stran 169 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 7.7: Program za pretvorbo RGB in CMYK

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



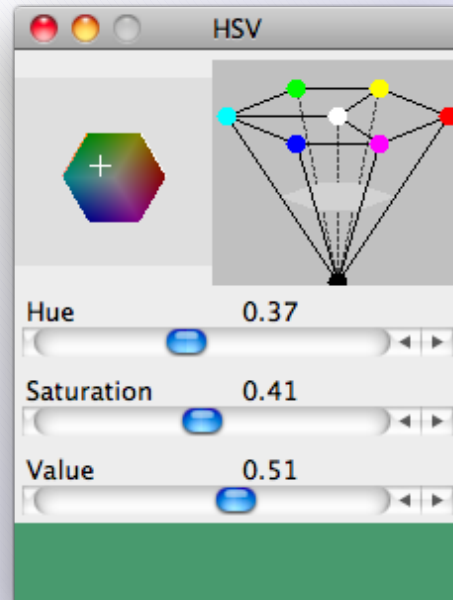
Stran 170 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 7.8: Barvna prostor HSB (HSV) ponazorjen s šeststrano piramido ([6])

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



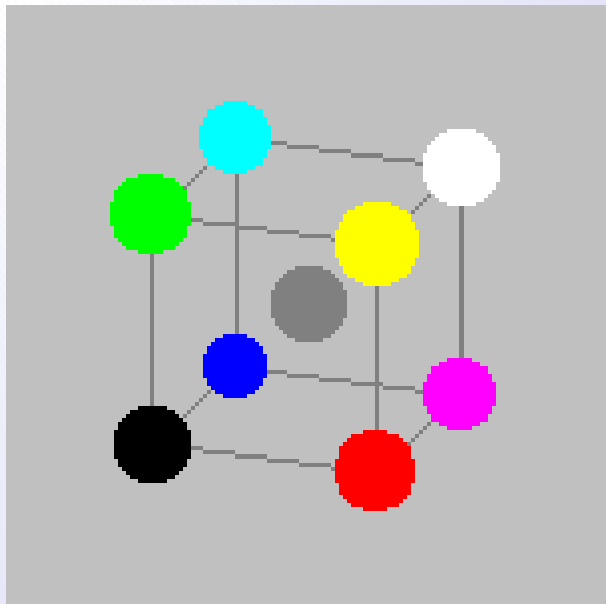
Stran 171 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)



Slika 7.9: Barvna modela RGB in CMYK ponazorjena s kocko

Spletna stran

Naslovnica

Kazalo



Stran 172 od 219

Nazaj

Full Screen

Zapri

Končaj

7.2. Naprave za prikaz slike

7.2.1. Katodna cev

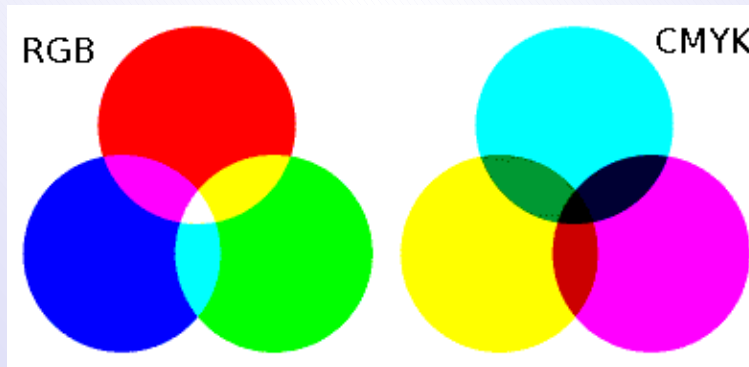
7.2.1.1. Rastrski prikaz

7.2.1.2. Vektorski prikaz

7.2.1.3. Ločljivost

7.2.2. Tiskalniki in risalniki

7.2.3. Vhodne interaktivne naprave



Slika 7.10: Primerjava RGB in CMYK

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 173 od 219

Nazaj

Full Screen

Zapri

Končaj

7.3. Rastrska in vektorska grafika

Primerjava rastrske in vektorske grafike.

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran 174 od 219

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

7.4. Nekateri uporabni programi

Program za zajemanje barve (angl. "picker"): **Pixie**

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **175** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **176** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

Poglavje 8

Napredna raba uporabniških programov

Cilji poglavja so:

- motivirati (bodočega) inženirja za *kreativno razmišljanje* pri reševanju (vsakdanjih inženirskih) problemov z računalniškimi programi,
- pridobiti občutek, da so *uporabniški programi* zmogljiva orodja v rokah spretnega uporabnika,
- razumeti in uporabiti *mehanizme avtomatizacije uporabniške programske opreme* (splošno namenska in specialno namenska)

Spletna stran

Naslovnica

Kazalo



Stran 177 od 219

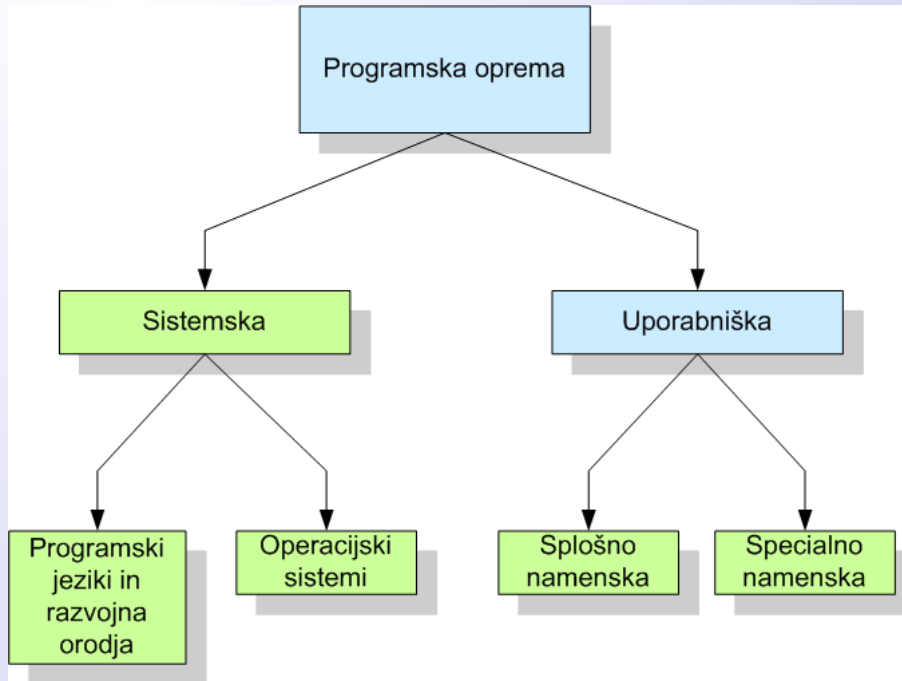
Nazaj

Full Screen

Zapri

Končaj

8.1. Umestitev uporabniške programske opreme



Slika 8.1: *Uporabniška programska oprema*

Uporabniško programsko opremo delimo na splošno in specialno namensko (slika 8.1). Med splošno namensko programsko opremo prištevamo:

- pisarniške programe kamor spadajo urejevalniki besedil, elektronske pregle-

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 178 od 219

Nazaj

Full Screen

Zapri

Končaj

dnice, programi za predstavitve, elektronsko pošto, upravljanje s slikami.

Med specialno namensko programsko opremo lahko prištevamo:

- programsko opremo za oblikovanje spletnih dokumentov, inženirsko programsko opremo (npr. programi CAD), podatkovne baze, varnost, virusi.

Spletna stran

Naslovnica

Kazalo



Stran 179 od 219

Nazaj

Full Screen

Zapri

Končaj

8.2. Kje se začne napredna raba uporabniških programov

Predpogoj za napredno rabo programov v današnjih grafičnih operacijskih sistemih je obvladovanje dela z okni. Trdimo lahko naslednje:

- *pokaži mi kako delaš z okni in povem ti kakšen uporabnik računalnika si.*

Večletno gledanje pod prste uporabnikom v računalniški učilnici me je prepričalo, da večina bodočih inženirjev ni dovolj spretnih pri delu z večimi okni v večopravilnih operacijskih sistemih. Najpogostejša nerodnost je:

- nepotrebno zapiranje/odpiranje programov namesto preklapljanja med aktivnimi programi

Sodobni operacijski sistemi so večopravilni, zato lahko uporabnik dela z večimi opravili (okni) "hkrati". Uporabnikova spretnost pri navigaciji med večimi okni lahko prispeva k učinkovitejši rabi teh programov. Zato za vajo odprimo nekaj različnih programov in jih razporedimo tako, da zapolnimo ekran (slika 8.2). Zelo blizu resnice je tudi naslednja trditev:

- *pokaži mi svojo organizacijo diska in povem ti kakšen uporabnik računalnika si*

Ampak, to je lahko tudi filozofska debata, kjer imajo svoje argumente tudi zagovorniki kaosa (*ali drugače...vsi dokumenti v eni mapi*), ki uporabljajo dobro orodje za iskanje (nenazadnje Google na internetu nam pomaga iskati v kaosu spletnih dokumentov). Nasprotje kaotični organizaciji diska je vpeljava večnivojske strukture z uporabo tematskih map (slika 8.3).

Spletna stran

Naslovnica

Kazalo



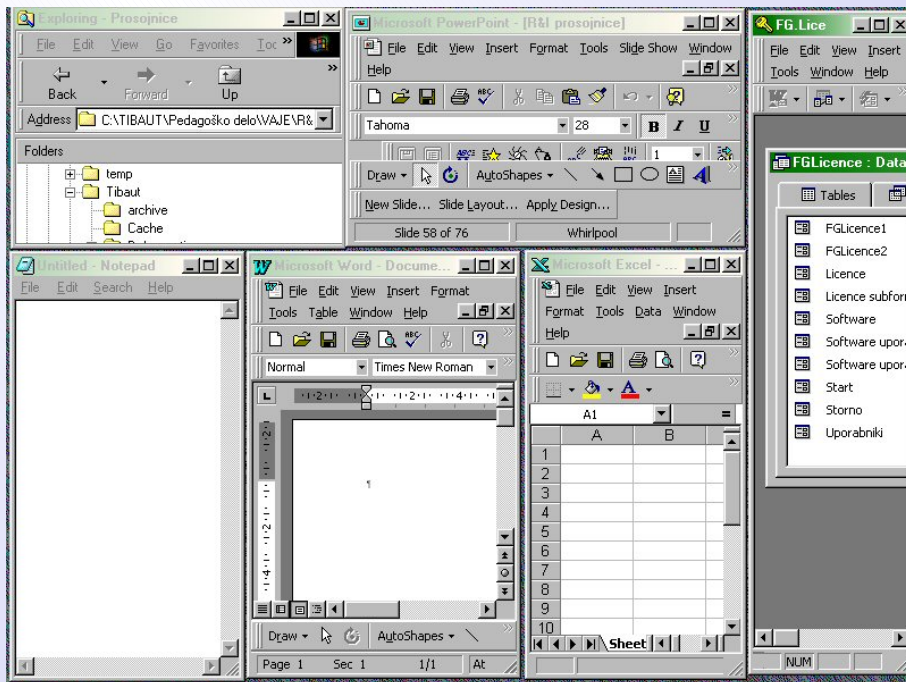
Stran 180 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.2: Ekran zapolnjen z večimi programi

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

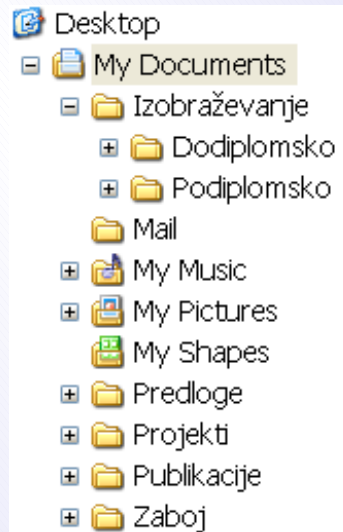
Stran 181 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.3: Tematska organizacija diska

[Spletna stran](#)

[Naslovnica](#)

[Kazalo](#)



Stran **182** od **219**

[Nazaj](#)

[Full Screen](#)

[Zapri](#)

[Končaj](#)

8.3. Napredna raba urejevalnika besedil

Začnimo na zabaven način. Zagotovo si ne želite pisati takšnih **dokumentov**. Ta neugleden dokument je bil izdelan z urejevalnikom MS Word.

V nadaljevanju nas bo zanimala napredna uporaba urejevalnika besedil MS Word, ki jo bomo spoznali skozi primere.

8.3.1. Makro in Visual Basic za aplikacije

8.3.1.1. Izdelava makroja s snemanjem

neustrezen znak	ustrezen znak
{	š
[Š
‘	ž
@	Ž
~	č
^	Č
,	ć
]	Ć

Tabela 8.1: šumniki in njihovi ekvivalenti v starih dokumentih

Starejši dokumenti (glej **primer seznama študentov iz leta 1994**), izdelani pred urejevalnikom iz Office 97, so v novejših verzijah pogojno berljivi, ko gre za pravi prikaz znakov č, Č, š, Š, ž, Ž, ć in Ć. Ker kodne tabele vse do verzije Windows

Spletna stran

Naslovnica

Kazalo



Stran 183 od 219

Nazaj

Full Screen

Zapri

Končaj

98 niso vsebovale teh znakov, je bilo potrebno za pisanje besedil uporabiti posebne pisave, ki so nekatere znake iz ASCII tabele nadomestili z “našimi”. V starih dokumentih je zato potrebno neustrezne znake (tabela 8.1) zamenjati s pravimi.

Izdelajmo *makro* v Wordu s katerim bomo vse neustrezne znake zamenjali z ustreznimi. Posneli bomo makro *ZamenjavaZnakov* za zamenjavo prvega znaka, nato pa makro ročno dopolnili še za zamenjavo ostalih znakov. Po izvedbi snemanja makra za zamenjavo znakov [v Š, nastane zanimiv, a zaenkrat nerazumljiv, zapis v *programskem jeziku Visual Basic za aplikacije* (angl. VBA - Visual Basic for Applications), ki izgleda takole:

```
1 Sub ZamenjavaZnakov()  
2 ' Procedura ZamenjavaZnakov  
3 ' Namen: izvede zamenjavo znaka v besedilu  
4  
5 ' Ta del je bil izdelan s 'snemanjem makroja'  
6 ' Ta del zamenja vse znake [ s "S  
7 Selection.HomeKey Unit:=wdStory  
8 Selection.Find.ClearFormatting  
9 Selection.Find.Replacement.ClearFormatting  
10 ' ZACETEK ZAMENJAVE  
11 With Selection.Find  
12 .Text = "[" ' -- stari znak  
13 .Replacement.Text = "S" ' -- novi znak  
14 .Forward = True  
15 .Wrap = wdFindContinue  
16 .Format = False
```

Spletna stran

Naslovnica

Kazalo



Stran 184 od 219

Nazaj

Full Screen

Zapri

Končaj


```

17 .MatchCase = False
18 .MatchWholeWord = False
19 .MatchWildcards = False
20 .MatchSoundsLike = False
21 .MatchAllWordForms = False
22 End With
23 Selection.Find.Execute Replace:=wdReplaceAll
24 ' KONEC ZAMENJAVE
25 End Sub

```

Program 8.1: Makro za zamenjavo znakov v VBA

Zdaj moramo makro dopolniti, da bo opravil še zamenjavo preostalih znakov. To lahko storimo tako, da podvajamo kodo med vrsticama ZAČETEK ZAMENJAVE in KONEC ZAMENJAVE tolikokrat kot je znakov za zamenjavo. Zdaj samo še spremenimo znake v vrsticah stari znak in novi znak.

Spremembe opravimo v okolju urejevalnika za *Visual basic za aplikacije* (slika 8.4), ki ga odpremo v meniju Orodja/Makri/Visual Basic (angl. Tools/Macros/Visual basic) ali s kombinacijo tipk ALT-F11.

Nastala programska koda brežhibno opravi svojo nalogo (*zamenjava znakov v šumnike in ščičnike*) vendar bo nekoliko izkušenejši uporabnik ugotovil, da je rešitev s *podvajanjem kode* nekoliko okorna. Kako bi torej spremenili programsko kodo makra, če bi zaradi “krajšega programa in boljšega programerskega sloga”, ne želeli tolikšnega podvajanja? Rešitev je v uporabi dveh dodatnih programskih struktur *polja in zanke*:

```

1 Sub ZamenjavaZnakov ()

```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

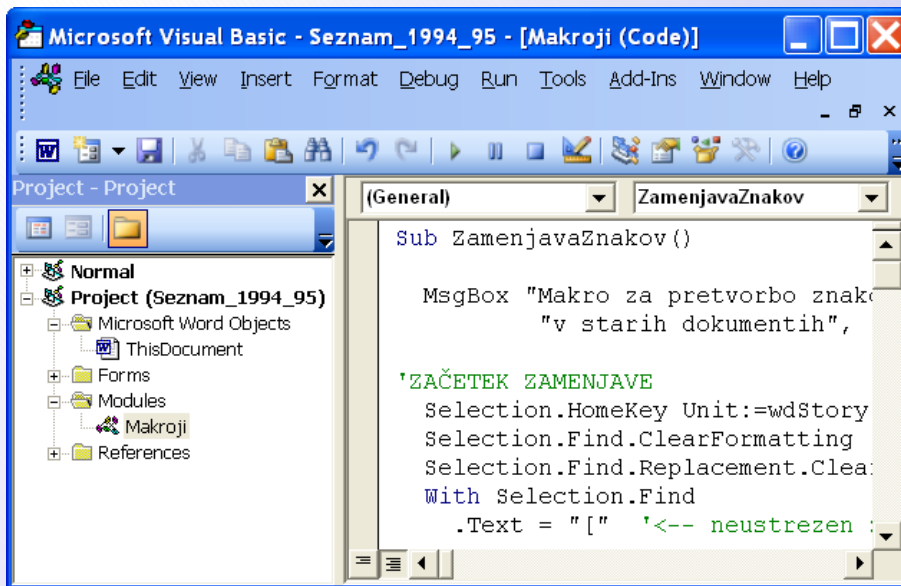
Stran 185 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.4: Delovno okolje Visual basica v Wordu

```

2 ' Procedura ZamenjavaZnakov
3 ' Namen: izvede zamenjavo znaka v besedilu
4
5 ' Naslednje vrstice so bile dodane ročno
6 Dim IskaniZnaki(4) As String
7 Dim Sumniki(4) As String
8
9 IskaniZnaki(1) = "["

```

Spletna stran

Naslovnica

Kazalo

◀

▶

◀

▶

Stran 186 od 219

Nazaj

Full Screen

Zapri

Končaj

```

10 IskaniZnaki(2) = "^"
11 IskaniZnaki(3) = "@"
12 IskaniZnaki(4) = "]"
13
14 Sumniki(1) = "S"
15 Sumniki(2) = "C"
16 Sumniki(3) = "Z"
17 Sumniki(4) = "\u0106"
18
19 ' Ta del je bil izdelan s snemanjem makra
20 ' Ta del zamenja vse znake [ s \u0160, ^ s \u010c,
21 ' @ z "Z in ] s \u0106
22 Selection.HomeKey Unit:=wdStory
23 Selection.Find.ClearFormatting
24 Selection.Find.Replacement.ClearFormatting
25 ' ZACETEK ZAMENJAVE
26 For ponovitev = 1 To 4
27   With Selection.Find
28     .Text = IskaniZnaki(ponovitev) '-- stari znak
29     .Replacement.Text = Sumniki(ponovitev) '-- novi
        znak
30     .Forward = True
31     .Wrap = wdFindContinue
32     .Format = False
33     .MatchCase = False
34     .MatchWholeWord = False

```

Spletna stran

Naslovnica

Kazalo



Stran 187 od 219

Nazaj

Full Screen

Zapri

Končaj

```

35     .MatchWildcards = False
36     .MatchSoundsLike = False
37     .MatchAllWordForms = False
38     End With
39     Selection.Find.Execute Replace:=wdReplaceAll
40 Next ponovitev
41 ' KONEC ZAMENJAVE
42 End Sub

```

Program 8.2: Izboljšan makro za zamenjavo znakov v VBA

Programski strukturi *polje* in *zanka* na tej točki podajamo kot zanimivost, ki je ne bomo podrobneje obrazložili. Zaključimo le, da so makroji izdelani s snemanjem zmogljivo orodje v rokah inženirja.

8.3.1.2. Ročna izdelava makroja v Visual Basicu za aplikacije

Zastavimo si naslednjo nalogo: kako uporabnika presenetiti z dialognim oknom ob odpiranju/zapiranju dokumenta v urejevalniku MS Word?

Poskusimo brez razlage...tukaj je kratki *program*, poimenujmo ga *Pozdrav uporabniku*, ki uporabnika prijazno pozdravi ob odpiranju dokumenta (slika 8.5):

```

1 Sub AutoOpen ()
2     MsgBox "Dober dan uporabnik! Ura je " & Now() & "!", ,
        "RI"
3 End Sub

```

Program 8.3: Dialog v VBA

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

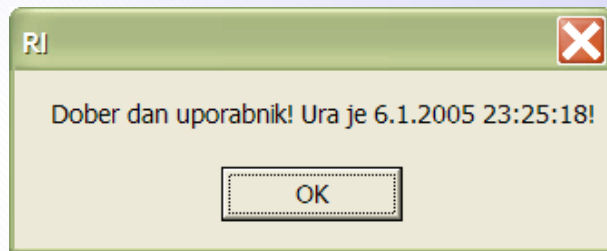
Stran 188 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.5: Dialogno okno, ki ga prikaže makro *AutoOpen*

in se prav tako prijazno poslovil od uporabnika ob zapiranju dokumenta, če dodamo še naslednji del *programske kode*:

```
1 Sub AutoClose()  
2   MsgBox "Nasvidenje uporabnik! Ura je " & Now() & "!",  
   , "RJ"  
3 End Sub
```

Program 8.4: *Dialog v VBA*

Prenesimo zgornjo programsko kodo v urejevalnik MS Word na sledeči način (slika 8.6):

- izberimo **Orodja**→**Makro**→**Makri** in v polje ime makra vpišimo *AutoOpen* nato izberimo **Kreiraj**. Vpišimo vrstico, ki se začne z *MsgBox*.

Kaj je pravzaprav *makro*? Makro (**glej definicijo**) je zaporedje ukazov za računalniško obdelavo. Ukazi se izvedejo (obdelajo), če izvedemo makro. S pomočjo makroja lahko izvedemo avtomatizacijo izvajanja pogostih opravil. Zapomnimo si,

Spletna stran

Naslovnica

Kazalo



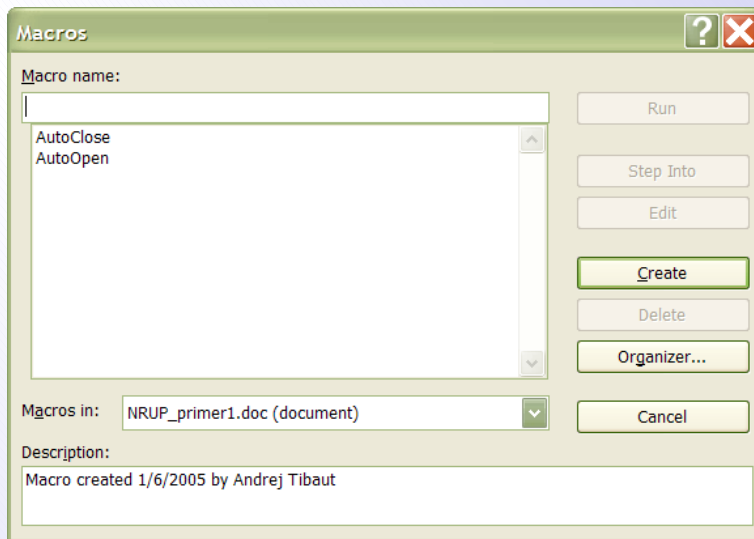
Stran 189 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.6: *Seznam makrojev v urejevalniku*

da smo za izdelavo programa *Pozdrav uporabniku* uporabili makro, katerih izvajanje omogoča urejevalnik MS Word.

Program *Pozdrav uporabniku* nas uči, da je možno z določenim zaporedjem ukazov zapisati *programsko kodo*, ki predstavlja *računalniški program*. Če pogledamo natančneje ugotovimo, da je program sestavljen iz dveh sklopov. Vsak sklop se začne z besedo

```
1 Sub ... ()
```

Program 8.5: *Vrstični slog*

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 190 od 219

Nazaj

Full Screen

Zapri

Končaj

in zaključí z besedo

```
1 End Sub
```

Program 8.6: Vrščni slog

Takšen sklop imenujemo *procedura*. Naš program *Pozdrav uporabniku* vsebuje torej dve proceduri (`AutoOpen` in `AutoClose`). Poglejmo si поблиže vsebino obeh procedur. Dve besedi izstopata:

```
1 MsgBox
```

Program 8.7: Vrščni slog

in

```
1 Now ( )
```

Program 8.8: Vrščni slog

Prva beseda `MsgBox` je *ukaz* za prikaz dialognega okna, druga beseda `Now ()` je ukaz za prikaz datuma in ure v dialognem oknu. Oba ukaza pripadata *programskemu jeziku Visual Basic za aplikacije* (angl. Visual Basic for Applications, VBA). Naš program *Pozdrav uporabniku* je torej zapisan v programskem jeziku Visual Basic za aplikacije.

Predstavlajte si uporabnika, ki bi ob odpiranju dokumenta zagledal naslednji dialog (slika 8.7):

```
1 Sub AutoOpen()  
2   MsgBox "Virus: ali naj pobrišem celoten disk?", _  
3     vbYesNo + vbCritical, "RI"
```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 191 od 219

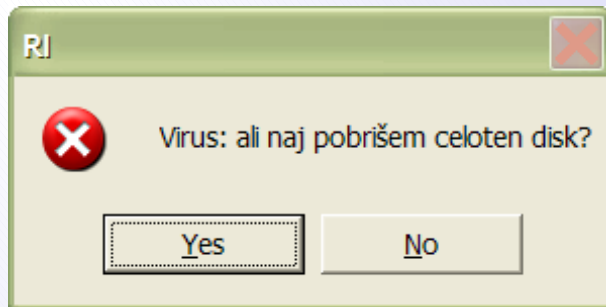
Nazaj

Full Screen

Zapri

Končaj

Program 8.9: Dialog v VBA



Slika 8.7: Lažni virus

Če bi imena procedur poimenovali nekoliko drugače (tabela 8.2), bi se dialogno okno prikazalo ob drugem dogodku.

Ime makroja	Kdaj se izvede (dogodek)
AutoExec	vsakič, ko zaženemo Word ali naložimo globalno predlogo
AutoNew	vsakič, ko kreiramo nov dokument
AutoOpen	vsakič, ko odpremo obstoječ dokument
AutoClose	vsakič, ko zapremo dokument
AutoExit	vsakič, ko zapremo Word ali zapremo globalno predlogo

Tabela 8.2: Seznam auto-makrojev

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 192 od 219

Nazaj

Full Screen

Zapri

Končaj

Bo zadostovalo za pripravo presenečenja na domačem računalniku?

8.3.1.3. Izdelava forme za vnos podatkov

Nadgrajevanje uporabe makrov in Visual Basica za aplikacije bi lahko šlo v smeri izdelave bolj zanimivih dialognih oken, takšnih, ki poleg ukaznih gumbov in besedila vsebujejo še kakšen kombinirani seznam, vnosno polje in še kaj. Takšna okna se običajno uporabljajo za vnos podatkov, imenujemo jih tudi *forme* za vnos. V urejevalniku Word ima uporabnik možnost, s pomočjo Visual basica za aplikacije, izdelati tudi forme (slika 8.8).

8.3.1.4. Povezava urejevalnika in elektronske preglednice

V nekem podjetju želijo spremljati čas, ki ga zaposleni porabijo za pisanje dokumentov v urejevalniku. V ta namen potrebujejo rešitev, ki bi zabeležila čas ob odpiranju in zapiranju dokumenta v urejevalniku. Za vsak dogodek odpiranja in zapiranja dokumenta naj se shrani podatek o imenu dokumenta in trenutnem času. Podatki naj bodo shranjeni v Excelovi preglednici kot seznam, ki ga bo možno kasneje pregledati. V ta namen bomo uporabili t.i. auto-makre (tabela 8.2), ki smo jih že spoznali. Spomnimo...to so posebni makri, ki se samodejno izvedejo ob določenem dogodku: zagonu urejevalnika, odpiranju novega dokumenta, odpiranju obstoječega dokumenta, zapiranju dokumenta ali izhodu iz urejevalnika.

Poglejmo kako izgleda **rešitev** v Visual basicu za aplikacije v Wordu:

```
1 Sub AutoOpen()  
2 'Opis: Poklice funkcijo za pisanje v Excelov dokument.  
3 ' Izvede se ob odpiranju Excelovega dokumenta.
```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

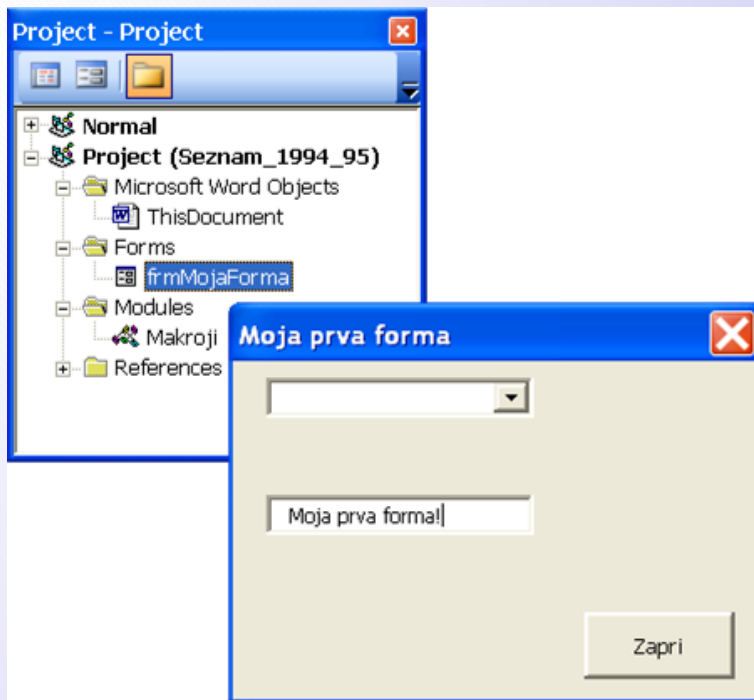
Stran 193 od 219

Nazaj

Full Screen

Zapri

Končaj



Slika 8.8: Forma v urejevalniku

```
4  
5 Call Zapisi_tekst_na_list_v_Excelu( _  
6     Application.ActiveDocument.Name & "/zacetek ")  
7  
8 End Sub
```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 194 od 219

Nazaj

Full Screen

Zapri

Končaj

```

9 Sub AutoClose()
10 'Opis: Poklice funkcijo za pisanje v Excelov dokument.
11 ' Izvede se ob zapiranju Excelovega dokumenta.
12
13 Call Zapisi_tekst_na_list_v_Excelu( _
14     Application.ActiveDocument.Name & "/konec ")
15
16 End Sub
17
18 Sub Zapisi_tekst_na_list_v_Excelu(mojTekst As String)
19 'Opis: Odpre dokument v Excelu in vanj zapise tekst
20 ' glej tudi komentar v kodi
21 'Vhod:
22 ' - mojTekst: tekst, ki ga zelimo vpisati
23 ' na list Excelovega dokumenta
24 'Izhod: nova vrstica v Excelovem dokumentu
25
26
27 Dim xlApp As Excel.Application
28 Dim xlWB As Excel.Workbook
29 Dim i As Integer
30
31 ' odpri Excel (nov Excelov objekt)
32 Set xlApp = CreateObject("Excel.Application")
33 'ce dokument RI_Evidenca_Dela_v_Wordu.xls obstaja
34 If Dir(Application.ActiveDocument.Path + _

```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 195 od 219

Nazaj

Full Screen

Zapri

Končaj

```

35  "/RI_Evidenca_Dela_v_Wordu.xls") <> vbNullString Then
36  ' ...ga odpri
37  Set xlWB = _
38  xlApp.Workbooks.Open(Application.ActiveDocument.Path +
39  _
40  _           "/" + "RI_Evidenca_Dela_v_Wordu.xls")
41 Else
42 ' sicer kreiraj nov dokument
43 xlApp.Visible = False
44 Set xlWB = xlApp.Workbooks.Add
45 End If
46
47 ' izvedi pisanje v Excelov dokument
48 With xlWB.Worksheets(1)
49 ' poisci prvo prazno vrstico
50 i = Najdi_prazno_vrstico(xlWB.Worksheets(1))
51 ' in v prvi stolpec vpisi zeljen tekst
52 .Cells(i, 1) = mojTekst
53 ' v drugega pa trenutni cas
54 .Cells(i, 2) = Now()
55 ' ce dokument ze obstaja
56 If Dir(Application.ActiveDocument.Path + _
57 _       "\RI_Evidenca_Dela_v_Wordu.xls") <> vbNullString
58 Then
59 ' ga samo shrani
60 xlWB.Save

```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 196 od 219

Nazaj

Full Screen

Zapri

Končaj

```

59 Else
60 ' sicer ga poimenuj in shrani = Save as
61   .SaveAs (Application.ActiveDocument.Path + _
62             "\RI_Evidenca_Dela_v_Wordu.xls")
63 End If
64 End With
65
66 ' Zapri Excelov dokument
67 xlWB.Close False
68 ' Zapri Excel
69 xlApp.Quit
70 ' Zbrisi kazalce objektov
71 Set xlWB = Nothing
72 Set xlApp = Nothing
73
74 End Sub
75
76 Function Najdi_prazno_vrstico(sSht)
77 'Opis: Poisce prvo prazno vrstico v Excelovem dokumentu
78 ' glej tudi komentar v kodi
79 'Vhod:
80 ' - sSht: Excelov list
81 'Izhod: stevilka prazne vrstice, npr. C=3
82
83 Dim StevVrstic As Integer
84

```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 197 od 219

Nazaj

Full Screen

Zapri

Končaj

```

85 ' stevec vrstic postavimo na zacetek
86 StevVrstic = 0
87 ' zanka v kateri iscemo prazno vrstico
88 Do
89 ' v vsakem koraku povecamo stevec
90   StevVrstic = StevVrstic + 1
91 ' ponavljamo dokler ne naletimo na prazno celico
92 Loop While Not sSht.Cells(StevVrstic, 1).Value =
    vbNullString
93
94 ' vrnemo stevilko celice
95 Najdi_prazno_vrstico = StevVrstic
96
97 End Function

```

Program 8.10: program v VBA za evidenco dela v urejevalniku MS Word

Na koncu tega poglavja si pogledjmo še seznam makrojev v meniju Orodja/Makro-/Makroji (angl. Tools/Macro/Macros), ki je nastal v tem podpoglavju (slika 8.6).

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 198 od 219

Nazaj

Full Screen

Zapri

Končaj

8.4. Sistem za izdelavo dokumentov \LaTeX

Vprašajmo se kako kritični znamo biti uporabniki računalniških programov do današnjih urejevalnikov besedil? Nikoli dovolj! Delež časa, ki ga inženirji, v profesionalnem delu življenja, porabimo za pisanje dopisov, navodil, poročil, analiz, idr. je prevelik, da ne bi razmišljali o učinkovitosti urejevalnika besedil, ki ga uporabljamo. Vprašanja, ki si jih moramo zastavljati niso: ali je moj urejevalnik boljši od sosedovega (navsezadnje večina itak uporablja takega kot sosed), ali ima moj urejevalnik dovolj funkcij (povprečen uporabnik MS Word-a itak uporablja le 40% vseh njegovih funkcij). Odločilen je odgovor na vprašanje: ali je pri izkušenem uporabniku uporaba urejevalnika besedil dovolj transparentna (neopazna). Transparentna v tem smislu, da je delo udobno in nekonfliktno, da je rezultat na izhodni napravi takšen kot si ga želi avtor, da je življenska doba dokumenta vsaj enaka življenski dobi avtorja, da so besedila v digitalni obliki enostavno prenosljiva. Razočarani boste...vaš urejevalnik tem kriterijem ne zadostuje! Zakaj ne? Ker najverjetneje uporabljate MS Word.

S sistemom za izdelavo dokumentov \LaTeX (izg. lateh) sem se srečal po dobrih petih letih uporabe urejevalnika MS Word. Besedilo je bilo potrebno opisovati, besedam dodajati dodatne znake. Poudarjen tekst je bilo potrebno napisati kot `\textbf{poudarjeno}`. Hej, komplicirano, ker si je bilo potrebno zapomniti vse te “okraske”. Se mi je pa zdelo dologčasno, ker si besedilo pisal v navadnem ASCII urejevalniku besedil, in zanič ker končnega izgleda besedila ni bilo moč videti sproti (kot je to pri MS Wordu). Za predogled je bilo potrebno besedilo “prevesti”. Končni izdelek je bil zapisan v formatu Postscript ali PDF. Proces publikacije je bil tako daljši kot pri ostalih urejevalnikih, ki delujejo po načelu “to kar vidiš, to dobiš” (angl. WYSIWYG - What You See Is What You Get). Kljub mojemu nezadovolj-

Spletna stran

Naslovnica

Kazalo



Stran 199 od 219

Nazaj

Full Screen

Zapri

Končaj

stvu se način dela v \LaTeX -u ni spremenil in dokumenti iz leta 1990 so berljivi v vseh naslednjih distribucijah \LaTeX -a. Pridih drugačnega pristopa k oblikovanju besedil pa je ostal popolnoma nespremenjen. Po dobrih desetih letih uporabe urejevalnika MS Word sem tudi sam predsedal na \LaTeX . Doba uporabe urejevalnika MS Word ni bila odločilna, odločilna je bila potreba po urejevalniku, ki bi omogočal dolgo- ročno in neodvisno rešitev za izdelavo velikih dokumentov kot je npr. to študijsko gradivo. Resno publiciranje zahteva verodostojnost končnega izgleda, prenosljivost dokumentov med različnimi operacijskimi sistemi in nenazadnje tudi spletno pove- zljivost. Ravno zaradi teh lastnosti \LaTeX slovi kot sistem za izdelavo dokumentov, ki mu lahko rečemo “to kar vidiš, to si želel” (angl. WYSIWYM - What You See Is What You Mean).

\LaTeX temelji na \TeX -u, sistemu za izdelavo (pisanje, postavitev) visoko kvalite- tnih dokumentov, ki ga je leta 1979 razvil **Donald Knuth**.

Oglejmo si primer **dokumenta** izdelanega v \LaTeX u:

```
1 %***** glava *****
2 \documentclass[a4paper,11pt]{letter}
3 \usepackage[slovene]{babel}
4 \usepackage[utf8]{inputenc}
5 \usepackage[T1]{fontenc}
6
7 %***** novi ukazi *****
8 \newcommand{\bi}{\begin{itemize}}
9 \newcommand{\vei}{\end{itemize}}
10 %*****
11
12 \name{Miha Zavarovanec}
13 \address{Miha Zavarovanec\
14          Zavarovanceva 33\
```

Spletna stran

Naslovnica

Kazalo



Stran 200 od 219

Nazaj

Full Screen

Zapri

Končaj


```

15      3333 NeVemKje }
16
17 \signature{Miha Zavarovanec}
18 \date{Oktober 28, 2003}
19 %***** zacetek dokumenta *****
20 \begin{document}
21 \begin{letter}{Zavarovalnica PridiKNam\\
22              Zavarovalniska pot 13\\
23              6666 Velike Zavarujce}
24 \opening{Spostovani g. direktor!}
25
26 Prejel sem Vas dopis v katerem me obvescate o poteklem
27 avtomobilskem zavarovanju. Hvala za Vaso skrb,
28 zal je le—ta bila izrazena nekoliko prepozno.
29 Vaso zavarovalnico sem se odlocil zamenjati za konkurencno.
30 Dovolite mi, da nekoliko natančneje pojasnim razlog za to:
31 \bi
32 \item nedoslednost vasih agentov za avtomobilsko zavarovanje;
33      nobeden od vasih dveh agentov me v zadnjih dveh letih ni
34      pravocasno opozoril o poteklem zavarovanju
35      (dasiravno mora to biti tudi moja skrb).
36      Taksne nedoslednosti zaradi dnevne uporabe avtomobila
37      ne morem spregledati.
38 \item v dobi, ko informatika ponuja toliko resitev,
39      elektronski opomniki ne bi smeli biti problem.
40 \ei
41
42 Pohvaliti pa zelim strokovni odnos vasih kolegov
43 v skodnih primerih.
44
45 \closing{V dobronamerni zelji, da postanete se boljji

```

Spletna stran

Naslovnica

Kazalo



Stran 201 od 219

Nazaj


Full Screen

Zapri

Končaj

```
46     Vas lepo pozdravljam,}
47
48 \end{letter}
49 \end{document}
```

Program 8.11: Dokument v zapisu \LaTeX

Izvorno besedilo napisano v \LaTeX -u je shranjeno v dokumentu tipa *.tex*. Vendar to ni predstavitevni format, ki bi ga želeli pošiljati okoli, torej izmenjevat. Dokument napisan v \LaTeX -u moramo prevesti, t.j. ga procesirati s programom \TeX , da dobimo zapis v formatu DVI (angl. DeVice Independent), ki je "od naprave neodvisen" zapis, ki vsebuje opis dokumenta, pisave, itd. Datoteko v zapisu DVI lahko pretvorimo v ostale predstavitvene formate, npr. PDF, Postscript, HTML in navadno besedilo. \LaTeX omogoča avtorju vključevanje že izdelanih makrojev neposredno v besedilo. Avtor ima na ta način neposreden nadzor nad tokom besedila in končnim izgledom dokumenta. Končna oblika dokumenta je rezultat zanesljivega programa za procesiranje besedila "okrašenega" z makroji \LaTeX -a, ki jih je avtor ročno dodal k vsebini besedila. O zanesljivosti programa za prevajanje besedila napisanega v \LaTeX -u pričča denarna nagrada, ki jo je Donald Knuth razpisal za vsako napako, ki bi jo kdo našel v programu . Najpogostejši predstavitevni format za izmenjavo besedil generiranih v \LaTeX -u je PDF (angl. Portable Document Format). Zakaj je PDF dobra izbira? Adobe PDF je izdelalo in standardiziralo podjetje Adobe Systems. Podjetje izdeluje zastojnski program za prikaz dokumentov PDF. Nastanku formata PDF je botrovala potreba založniške industrije po konsistentnem prikazu in tiskanju elektronskih dokumentov v formatu, ki bi bil neodvisen od operacijskega sistema. Konsistentnost prikaza originalnega dokumenta je dosežena zato, ker PDF format temelji na jeziku Postscript. Jezik Postscript je optimiran za tiskanje teksta in grafike in je neodvisen

Spletna stran

Naslovnica

Kazalo



Stran 202 od 219

Nazaj

Full Screen

Zapri

Končaj

od naprave na kateri dokument tiskamo. Postscript je leta 1985 predstavilo podjetje Adobe.

8.4.1. \LaTeX in PDF za inženirsko rabo

Razvoj \LaTeX -a sledi razvoju formata PDF. Format PDF ima tri podmnožice, ki jih razvija mednarodna organizacija za standardizacijo ISO:

- PDF/A je inačica formata PDF (ISO 19005-1:2005). Format je namenjen trajnejšemu arhiviranju digitalnih dokumentov (predvsem besedilnih) s ciljem, da bi dokumenti v tem zapisu bili samozadostni in bi tako tudi v prihodnosti izgledali enako kot ob stvaritvi. Format PDF/A zato zahteva posebno obravnavo v dokumentu uporabljenih pisav (morajo biti zapisane v dokumentu), barv in komentarjev.
- PDF/X je inačica formata PDF (ISO 15930-8:2008). Format je namenjen predvsem izmenjavi grafičnih vsebin in njihovemu tiskanju zato ne podpira vključevanja obrazcev, digitalnih podpisov, vključevanja zvoka in videa. Družina formatov PDF/X ima 8 različnih podverzij. Razlike med posameznimi podverzijami so predvsem v podpori barvnih modelov za vključene slike (CMYK in RGB).
- PDF/E je inačica formata PDF (ISO 24517-1:2008). Črka E v kratici PDF/E pomeni *Engineering*, torej PDF/Engineering. To je odprt zapis (specifikacija formata je javno dostopna), ki je namenjen neodvisni (nevtralni) izmenjavi tehnične dokumentacije v inženirstvu.

Spletna stran

Naslovnica

Kazalo



Stran 203 od 219

Nazaj

Full Screen

Zapri

Končaj

Za uporabo v inženirstvu je posebej zanimiv format PDF/E, ki omogoča vključevanje modelov 3D v dokument PDF. Vključevanje modelov 3D je omogočeno preko formata U3D (Universal 3D). Format U3D je standardiziran format (je del ISO PDF/E) za opis računalniške grafike v 3D. Z uporabo \LaTeX -a lahko izdelamo dokumente PDF z vključenimi interaktivnimi modeli 3D (slika 8.9).

Spletna stran

Naslovnica

Kazalo



Stran 204 od 219

Nazaj

Full Screen

Zapri

Končaj

Spletna stran

Naslovnica

Kazalo



Stran 205 od 219

Nazaj

Full Screen

Zapri

Končaj

Slika 8.9: *Villa Kniggeweg v Berlinu, arhitekta K.Pogačar in A.Žižek*

8.5. Napredna raba elektronske preglednice

Napredna raba elektronske preglednice (v nadaljevanju *e-preglednica*) predvideva, da uporabnik obvlada pretvorbo realnih problemov iz področja statistike (obdelava in analiza podatkov po statističnih zakonitostih), matematike (tabeliranje funkcij, prikaz funkcijskih grafov) in inženirskih ved (tabeliranje in analiza meritev, izvajanje inženirskih izračunov) v obliko primerno za reševanje v e-preglednici. Pri spoznavanju napredne rabe e-preglednice bomo uporabljali MS Excel, ki pa ga lahko v večini situacij ekvivalentno zamenjamo z e-preglednico OpenOffice.org Calc ali LibreOffice.org Calc. V okvir napredne rabe e-preglednice bomo uvrstili:

- uporabo vgrajenih funkcij s poudarkom na sestavljenih (gnezdenih) funkcijah iz vseh kategorij:
 - matematične in trigonometrične funkcije: SUM, SUMIF, AVERAGE, COUNT, COUNTIF, (*,/, +, -), POWER (^), SIN, COS, TAN, PI, ROUND, TRUNC, SQRT
 - delo z besedilom: CONCATENATE, LEN, LEFT, RIGHT, EXACT, LOWER, UPPER
 - logične funkcije: IF, AND, OR, NOT
 - iskalne funkcije, delo s podatki: MIN, MAX, HLOOKUP, VLOOKUP, Pivot tables (vrtilne tabele)
- uporaba funkcij za inženirske izračune
- izdelava makrojev in njihova nadgradnja s programskim jezikom Visual Basic za aplikacije

Spletna stran

Naslovnica

Kazalo



Stran 206 od 219

Nazaj

Full Screen

Zapri

Končaj

8.5.1. Uporaba funkcij v e-preglednici

Uporaba funkcij je eno (naj)močnejših “orožij” pri delu s preglednicami. Pokazali bomo primer napredne uporabe funkcij na nekaj primerih.

8.5.1.1. Analiza stroškov gradbišča

Primer prikazuje uporabo vrtilne tabele za pregled stroškov gradbišča.

8.5.1.2. Preoblikovanje seznama študentov

Pri svojem vsakdanjem delu je avtor te publikacije naletel na sledeč problem. Na začetku študijskega leta je bilo potrebno v spletno učilnico (Moodle) vključiti 15 študentov. Podatki o študentih so bili pridobljeni v preglednici. Spletna učilnica omogoča ročni vnos posameznega študenta in masovni vnos seznama študentov. Ob pregledu navodil za masovni vnos seznama študentov se je izkazalo, da bo obstoječi seznam potrebno preoblikovati. Možnosti sta dve: ročno preoblikovanje ali preoblikovanje z uporabo funkcij preglednice. Seveda smo izbrali preoblikovanje z uporabo funkcij, saj je ročni vnos počasen. Podatki v obstoječem seznamu študentov (slika 8.10) so zapisani v sledeči obliki:

1 priimek in ime, e–naslov

Program 8.12: *Obstoječa oblika seznama študentov*

Podatke iz obstoječega seznam študentov je potrebno preoblikovati v naslednjo obliko (slika 8.11):

1 uporabnisko ime, geslo, ime, priimek, e–naslov

Spletna stran

Naslovnica

Kazalo



Stran 207 od 219

Nazaj

Full Screen

Zapri

Končaj

Program 8.13: Zahtevana oblika seznama študentov

Za rešitev naloge smo uporabili funkcije za delo z besedilom (CONCATENATE, FIND, LEFT, LEN, MID). Bistvo naloge je v pravilni sestavi funkcij. Poglejmo primer. V celici A8 je kot besedilo zapisano "McKone, David."; priimek in ime študenta z vmesno vejico in piko na koncu. Izluščiti je potrebno ime in priimek:

- priimek zapišemo v celico D25 s formulo:

$$= LEFT(A8; FIND(", "; A8) - 1) \quad (8.1)$$

- ime zapišemo v celico C25 s formulo:

$$= MID(A8; FIND(" "; A8) + 1; LEN(A8) - LEN(D25) - 3) \quad (8.2)$$

- uporabniško ime sestavimo iz imena in priimka s formulo:

$$= CONCATENATE(C25; ". "; D25) \quad (8.3)$$

Za geslo smo uporabili kar elektronski naslov. Celotna rešitev naloge je v preglednici .

8.5.1.3. Izračun dopustne obtežbe temeljnih tal

Za reševanje osnovnih problemov temeljenja objektov je v strokovni literaturi in v veljavnih tehničnih normativih podano večje število pretežno empiričnih obrazcev za

Spletna stran

Naslovnica

Kazalo



Stran 208 od 219


Nazaj

Full Screen

Zapri

Končaj

vrednotenje dopustnih obtežb temeljnih tal, posedanje objektov, stabilnostnih analiz pobočij ipd. Zemljine so praviloma heterogene sestave in so velikokrat na mikrolokacijah posameznih objektov bistveno drugačne od tistih ugotovljenih s sondažnimi in laboratorijskimi preiskavami. Zato je v geomehaniki v večini primerov potrebno iskati primerne rešitve takoj na terenu, upoštevajoč dejansko ugotovljene sestave temeljnih tal v izvršenih izkopih in ob tem podati dokončno mnenje glede pogojev temeljenja objekta.

V tem primeru bomo s pomočjo Excela obdelali vrednotenje dopustne obtežbe temeljnih tal. Osnovna enačba je obrazec Brinch - Hansena, ki jo v preglednici zapišemo kot večdelno formulo :

$$P_a = \frac{\gamma}{2} \times B \times N_\gamma \times S_\gamma \times i_\gamma + (c_m + q \times \tan \phi_m) \times N_c \times S_c \times d_c \times i_c + q \quad (8.4)$$

8.5.2. Uporaba Visual Basica za aplikacije v elektronski preglednici

8.5.2.1. Izdelava študentske evidence

Elektronsko preglednico želimo uporabiti za seznam študentov. Zelo enostavno! Poimenujemo stolpce, npr. ime, priimek, vpisna številka, ocenah, itd. Preglednica je uporabna če lahko po njej enostavno iščemo. Iskanje je zares enostavno, saj je to ena od vgrajenih funkcionalnosti preglednice. V primeru zadetka lahko hitro lociramo iskano celico. Pa vendarle naletimo na eno nerodnost. Če je stolpcev toliko, da jih vseh ne vidimo hkrati na zaslonu, je lahko iskanje zaradi tega nerodno. Pomikanje drsnika levo in desno postane ščasoma “naporno”. Če vseh stolpcev pri iskanju niti

Spletna stran

Naslovnica

Kazalo



Stran 209 od 219

Nazaj

Full Screen

Zapri

Končaj

ne potrebujemo, potem velja to še toliko bolj. Predstavljajmo si predavatelja, ki želi iz seznama dobiti le tiste stolpce, ki jih potrebuje za vpis ocen v indeks. Običajno je to vpisna številka, ime in priimek, ter ocene. Kako naj dosežem, da mi bo iskanje prikazalo le te štiri stolpce, od katerih so prvi trije stolpci na začetku preglednice, stolpec ocene pa v zadnjem stolpcu.

Rešitev lahko zapišemo kot funkcijo v Visual basicu, ki jo bo izvedel makro. Funkcijo poimenujmo *Poizvedba*.

```
1 Option Explicit
2 Option Base 1
3
4 Function Poizvedba() As Boolean
5
6 ' Definicija spremenljivk
7 ' trenutna celica z zadetkom
8 Dim mojaCelica As Variant
9 ' celica s prvim zadetkom
10 Dim prviZadetek As Variant
11
12 ' vrstica in stolpec za izpis rezultata
13 Dim rVrstica As Integer
14 Dim rStolpec As Integer
15
16 ' vrstica in stolpec zadetka
17 Dim iVrstica As Integer
18 Dim iStolpec As Integer
19 ' vrstica predhodnega zadetka
```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 210 od 219

Nazaj

Full Screen

Zapri

Končaj

```

20 Dim ipVrstica As Integer
21
22 ' sstevílo zadetkov
23 Dim zadetki As Integer
24 ' naslov celice zadetka, npr. R1C2
25 Dim mojNaslov As String
26
27 zadetki = 0
28 rVrstica = 4: rStolpec = 1
29 ipVrstica = -1
30
31 With Worksheets("Ocene").Range("a1:af500")
32 ' zacni iskanje
33 Set mojaCelica = .Find(Sheets("Poizvedbe").Cells(2, 1)
34     ' -
35     LookIn:=xlValues, LookAt:=xlWhole)
36 ' ce je iskana beseda najdena nadaljuj
37 If Not mojaCelica Is Nothing Then
38     prviZadetek = mojaCelica.Address
39     ' isci dokler ni vec zadetka
40 Do
41     ' povecaj stevílo zadetkov
42     zadetki = zadetki + 1
43     mojNaslov = mojaCelica.Address(ReferenceStyle:=
44         xlR1C1)
45     iVrstica = CInt(Mid(mojNaslov, 2, InStr(1,

```

Spletna stran

Naslovnica

Kazalo

◀◀

▶▶

◀

▶

Stran 211 od 219

Nazaj

Full Screen

Zapri

Končaj

```

    mojNaslov, "C") - 2))
44
45 If ipVrstica <> iVrstica Then
46     rStolpec = 1
47     ' vstavi novo vrstico na zacetek izpisa zadetkov
48     Worksheets("Poizvedbe").rows(rVrstica).Insert
49     ' Indeks
50     Sheets("Poizvedbe").Cells(rVrstica, rStolpec) = _
51     Sheets("Ocene").Cells(iVrstica, 1)
52     ' Priimek
53     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 1)
54     = _
55     Sheets("Ocene").Cells(iVrstica, 2)
56     ' Ime
57     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 2)
58     = _
59     Sheets("Ocene").Cells(iVrstica, 3)
60     ' Predmet
61     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 3)
62     = _
63     Sheets("Ocene").Cells(iVrstica, 4)
64     ' Leto
65     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 4)
66     = _
67     Sheets("Ocene").Cells(iVrstica, 5)
68     ' datum izpita

```

Spletna stran

Naslovnica

Kazalo



Stran 212 od 219

Nazaj

Full Screen

Zapri

Končaj

```

65     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 5)
        = _
66     Sheets("Ocene").Cells(iVrstica, 6)
67     'ocena izpita
68     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 6)
        = _
69     Sheets("Ocene").Cells(iVrstica, 22)
70     'ocena vaj
71     Sheets("Poizvedbe").Cells(rVrstica, rStolpec + 7)
        = _
72     Sheets("Ocene").Cells(iVrstica, 30)
73     ipVrstica = iVrstica
74     End If
75
76     Set mojaCelica = .FindNext(mojaCelica)
77     Loop While Not mojaCelica Is Nothing And mojaCelica.
        Address _
78     <> prviZadetek
79     End If
80 End With
81
82 ' ni bilo zadetka, izpisi obvestilo
83 If zadetki = 0 Then _
84     MsgBox "Iskanje ni dalo rezultatov!", vbExclamation, "
        Opozorilo"
85

```

Spletna stran

Naslovnica

Kazalo



Stran 213 od 219

Nazaj

Full Screen

Zapri

Končaj

Program 8.14: *Poizvedba po seznamu študentov*

Spletna stran

Naslovnica

Kazalo



Stran 214 od 219

Nazaj

Full Screen

Zapri

Končaj

The screenshot shows an Excel spreadsheet with the following data:

	A	B
1	priimek in ime	e-naslov
2	Bartley, Eoin.	eoin.bartley@student.uci.si
3	Gallagher, Fergal.	fergal.gallagher1@student.uci.si
4	George, Jansi.	jansi.george@student.uci.si
5	Grimes, Enda.	enda.grimes@student.uci.si
6	Langridge, Graham.	graham.langridge@student.uci.si
7	Martin, Gerard.	gerard.martin@student.uci.si
8	McKone, David.	david.mckone@student.uci.si
9	McLoughlin, David.	david.mcloughlin5@student.uci.si
10	Molloy, Fergus.	fergus.molloy@student.uci.si
11	Ogbebor, Paul.	paul.ogbebor@student.uci.si
12	O'Reilly, Kieran.	kieran.oreilly1@student.uci.si
13	O'Neill, James	james.oneill3@student.uci.si
14	Sheedy, Paul.	paul.sheedy@student.uci.si
15	Sopel, Marek.	marek.sopel@student.uci.si
16	Woods, Trevor.	trevor.woods1@student.uci.si

Slika 8.10: Obstoječi seznam študentov

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 215 od 219

Nazaj

Full Screen

Zapri

Končaj

NaprednaRabaPreglednice-Funkcije-resitev.xlsx

Search in Sheet

Home Layout Tables Charts SmartArt Formulas Data Review

Font: Arial, 11

Alignment: General

Number: General

Format: Conditional Formatting, Styles, Actions, Themes

	A	B	C	D	E	F	G
18	uporabniško ime	geslo	ime	priimek	e-naslov		
19	Eoin.Bartley	eoin.bartley@student.uci.si	Eoin	Bartley	eoin.bartley@student.uci.si		
20	Fergal.Gallagher	fergal.gallagher1@student.uci.si	Fergal	Gallagher	fergal.gallagher1@student.uci.si		
21	Jansi.George	jansi.george@student.uci.si	Jansi	George	jansi.george@student.uci.si		
22	Enda.Grimes	enda.grimes@student.uci.si	Enda	Grimes	enda.grimes@student.uci.si		
23	Graham.Langridge	graham.langridge@student.uci.si	Graham	Langridge	graham.langridge@student.uci.si		
24	Gerard.Martin	gerard.martin@student.uci.si	Gerard	Martin	gerard.martin@student.uci.si		
25	David.McKone	david.mckone@student.uci.si	David	McKone	david.mckone@student.uci.si		
26	David.McLoughlin	david.mcloughlin5@student.uci.si	David	McLoughlin	david.mcloughlin5@student.uci.si		
27	Fergus.Molloy	fergus.molloy@student.uci.si	Fergus	Molloy	fergus.molloy@student.uci.si		
28	Paul.Ogbebor	paul.ogbebor@student.uci.si	Paul	Ogbebor	paul.ogbebor@student.uci.si		
29	Kieran.O'Reilly	kieran.oreilly1@student.uci.si	Kieran	O'Reilly	kieran.oreilly1@student.uci.si		
30	Jame.O'Neill	james.oneill3@student.uci.si	Jame	O'Neill	james.oneill3@student.uci.si		
31	Paul.Sheedy	paul.sheedy@student.uci.si	Paul	Sheedy	paul.sheedy@student.uci.si		
32	Marek.Sopel	marek.sopel@student.uci.si	Marek	Sopel	marek.sopel@student.uci.si		
33	Trevor.Woods	trevor.woods1@student.uci.si	Trevor	Woods	trevor.woods1@student.uci.si		

Seznam Kopija

Slika 8.11: Zahtevan seznam študentov

Spletna stran

Naslovnica

Kazalo

◀◀ ▶▶

◀ ▶

Stran 216 od 219

Nazaj

Full Screen

Zapri

Končaj

Literatura

- [1] Matti Hannus. Evolution of it in construction over the last decades, 1998. <http://cic.vtt.fi/hannus/islands/index.html>.
- [2] Danijel Rebolj. Binarni seštevalnik. slika, 1998.
- [3] ARNES. Komunikacijsko omrežje arnes. slika, 11 2003. <http://www.arnes.si/infrastruktura.html>, 2003.
- [4] ARNES. Komunikacijsko omrežje arnes. slika, 10 2011. <http://www.arnes.si/infrastruktura.html>, 2011.
- [5] T-2. Komunikacijsko omrežje t-2. slika, 11 2006. <http://www.t-2.net/>, (Omrežje), 2009.
- [6] Eugene Vishnevsky. The color spaces conversion applet. program, 11 2010. http://www.cs.rit.edu/~ncs/color/a_spaces.html.
- [7] Robert H. Blissmer. Introducing computers: Concepts, systems, and applications 1995/96 edition. Wiley, 03 1995.

Spletna stran

Naslovnica

Kazalo



Stran 217 od 219

Nazaj

Full Screen

Zapri

Končaj

- [8] W. Yurcik and E. F. Gehringer. A survey of web resources for teaching computer architecture. *Workshop on Computer Architecture Education*, 4 2002. <http://www.ncsu.edu/wcae/ISCA2002/submissions/yurcik.pdf>, 15.5.2007.
- [9] T. Matjašec and A. Tibaut. Animacija delovanja von neumannovega modela digitalnega računalnika. animacija, 01 2012. <http://www.fg.uni-mb.si/predmeti/RI/Slike/DelovanjeVonNeumannRacunalnika.swf>.
- [10] Matjaž Horvat. Uvod v linux. *Slo-Tech.com*, 4 2004. <http://slo-tech.com/clanki/04004/04004.shtml>, 10.11.2004.
- [11] ARNES. Komunikacijske tehnologije za telefonsko omrežje. <http://www.arnes.si/storitve/dostop/adsl-in-xdsl.html>, 10 2010.

Spletna stran

Naslovnica

Kazalo



Stran 218 od 219

Nazaj

Full Screen

Zapri

Končaj

Stvarno kazalo

ALE, *glej* aritmetično-logična enota
aritmetično-logična enota, **72**
ASCII, **67**

e-preglednica, **206**
enota

aritmetično-logična, **72**

gonilnik, **114**

grafični uporabniški vmesnik, **97**

LibreOffice.org Calc, *glej* e-preglednica

mikrosimulacija prometa, **80**

MS Excel, *glej* e-preglednica

multiprocesiranje, **80**

notranji slog, **154**

OpenOffice.org Calc, *glej* e-preglednica

programski števec, **75**

računalniški program, **85**

ukazni register, **76**

UNICODE, **67**

uporabniški vmesnik, **97**

von Neumann

animacija delovanja digitalnega računalnika, **72**

model digitalnega računalnika, **69**

vrstični slog, **155**

Zuse, Konrad

prvi elektronsko-mehanski digitalni računalnik, **34**

začetnik gradbene informatike, **50**

Spletna stran

Naslovnica

Kazalo



Stran **219** od **219**

Nazaj

Full Screen

Zapri

Končaj