

Brojevnii sistemi

— škola računara SystemPro —

Nemanja Mićović
nmicovic@outlook.com

Oktobar 2015

Abstrakt

Materijal pred Vama nastao je kao deo kursa Računarstva koji sam držao 2015. godine u školi računara za talentovane - SystemPro, i pokriva deo gradiva vezan za brojevne sisteme. Ukoliko je čitalac zainteresovan dublje za navedenu tematiku, može pogledati knjigu [1]. Takođe bih napomenuo da je tematika kojom ćemo se baviti u ovom tekstu, izuzetno široko dostupna i lepo ilustrovana na Veb-u, te svakako preporučujem da pogledate navedeni deo gradiva i tu.

Sadržaj

1	Uvod u brojevne sisteme	2
1.1	Prirodno pitanje o računarima	2
1.2	Osnovni pojmovi	2
1.3	Konverzija u dekadni brojevni sistem	2
1.3.1	Osnovni koncept za zapis broja	3
1.3.2	Binarni brojevni sistem	3
1.3.3	Oktalni brojevni sistem	5
1.3.4	Heksadekadni brojevni sistem	5
1.4	Ostale konverzije	6
1.4.1	Konverzija oktalni-binarni	6
1.4.2	Konverzija heksadekadni-binarni	6
2	Reprezentacija podataka u računaru	7
2.1	Osnovni pojmovi	7
2.2	Kodiranje karaktera	8
2.2.1	Binarni brojevi posmatrani kao kodiranje	8
2.2.2	ASCII tabela	9

1 Uvod u brojevne sisteme

1.1 Prirodno pitanje o računarima

Na časovima do sada, govorili smo uopšteno o računarima, i trudili smo se da dodemo do razumevanja njihovog celokupnog rada. Prvi korak koji načinismo na tom putu, jeste bio da računar podelimo na *hardver* i *softver*, odnosno na fizički deo koji se odnosio na komponente, i apstraktni deo koji se odnosio na programe.

Iz te podele, prirodno nam je proisteklo pitanje o samom načinu zapisa podataka u računarima, i ova skripta će na vrlo jednostavan i ilustrativan način predstaviti odgovor na to pitanje. Kako bismo došli do odgovora, najpre ćemo se malo pozabaviti brojevima samim po sebi, dok ćemo kasnije naučeno iskoristiti da damo konačan odgovor.

Na kraju poglavlja će biti navedeni primeri za vežbu, i čitalac nije u obavezi da ih uradi, ali neka ima na umu da time podsvesno nanosi sebi duševni bol i nemir.

1.2 Osnovni pojmovi

Kako bi načini prvi korak ka prosvetljenju naše duše, potrebno je razjasniti sledeće pojmove:

- Brojevni sistem
- Osnova brojevnog sistema
- Binarni brojevni sistem
- Oktalni brojevni sistem
- Dekadni brojevni sistem
- Heksadekadni (heksadecimalni) brojevni sistem

Brojevni sistem je sistem sa jasno definisanim pravilima za zapis broja po nekoj brojevnoj osnovi.

Binarni brojevni sistem je brojevni sistem sa osnovom 2 i ciframa iz skupa $\{0, 1\}$.

Oktalni brojevni sistem je brojevni sistem sa osnovom 8 i ciframa iz skupa $\{0, 1, 2, 3, 4, 5, 6, 7\}$.

Dekadni brojevni sistem je brojevni sistem sa osnovom 10 i ciframa iz skupa $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Heksadekadni brojevni sistem je brojevni sistem sa osnovom 16 i ciframa iz skupa $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

1.3 Konverzija u dekadni brojevni sistem

U navedenom delu, biće prikazane konverzije navedenih brojevnih sistema u nama prirodni, dekadni brojevni sistem. Najpe će biti ilustrovan način za zapis bilo kojeg broja na osnovu kojeg ćemo razviti način za konverziju navedenih sistema u kasnijem delu.

1.3.1 Osnovni koncept za zapis broja

Pogledajmo broj 421 i primetimo da važi jednakost:

Primer 1.1

$$(421)_{10} = 4 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0$$

Takođe, pogledajmo nešto veći broj:

Primer 1.2

$$(1234567)_{10} = 1 \cdot 10^6 + 2 \cdot 10^5 + 3 \cdot 10^4 + 4 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0$$

I nešto opštije, ukoliko su a_1, a_2, \dots, a_n cifre broja:

Primer 1.3

$$(a_1 a_2 a_3 \dots a_{n-1} a_n)_{10} = a_1 \cdot 10^{n-1} + a_2 \cdot 10^{n-2} + a_3 \cdot 10^{n-3} \dots + a_{n-1} \cdot 10^1 + a_n \cdot 10^0$$

Odnosno, neki dati broj možemo zapisati u obliku nekih sabiraka koje dobijamo množenjem cifara sa osnovom broja. U primerima 1.1, 1.2 i 1.3 smo koristili *dekadnu osnovu* broja, i osnova koju smo koristili bila je 10. Primetimo da je broj 10 u indeksu ispod zagrade sa leve strane navedenih jednakosti, služio da nam naznači koja je osnova u pitanju.

Naša sledeća opsesija je sledeća, hajde da navedeni princip primenimo na druge brojevne osnove kao što su, 2, 8 i 16.

1.3.2 Binarni brojevni sistem

Primetimo da smo u dekadnom brojevnom sistemu imali sledeće cifre: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, dok ćemo u binarnom sistemu koristiti osnovu 2, čime ćemo u optičaju imati cifre: $\{0, 1\}$.

Sledi prikaz stepena broja dva kao i ilustracija funkcije $f(x) = 2^x$ prikazana na slici 1.

$$\begin{aligned} 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &= 4 \\ 2^3 &= 8 \\ 2^4 &= 16 \\ 2^5 &= 32 \\ 2^6 &= 64 \\ 2^7 &= 128 \\ 2^8 &= 256 \\ 2^9 &= 512 \\ 2^{10} &= 1024 \\ 2^{11} &= 2048 \\ 2^{12} &= 4096 \\ 2^{13} &= 8192 \\ 2^{14} &= 16\,384 \\ 2^{15} &= 32\,768 \\ 2^{16} &= 65\,536 \\ 2^{17} &= 131\,072 \\ 2^{18} &= 262\,144 \\ 2^{19} &= 524\,288 \\ 2^{20} &= 1\,048\,576 \\ &\dots \end{aligned}$$

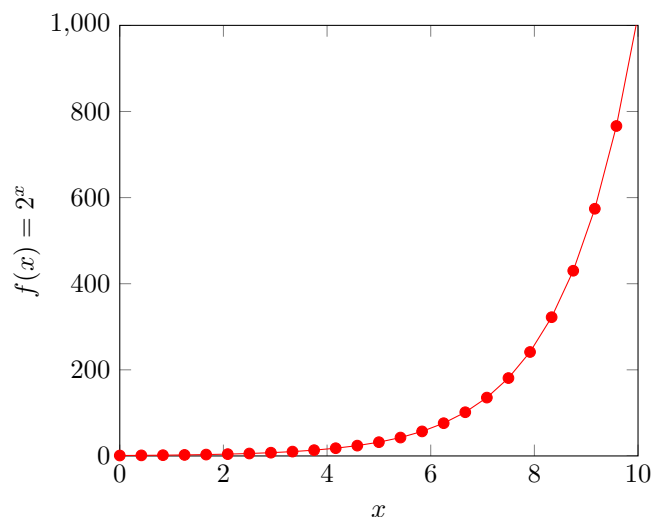
$$2^{32} = 4\,294\,967\,296$$

...

$$2^{64} = 18\,446\,744\,073\,709\,551\,616$$

...

$$2^{95} = 39\,614\,081\,257\,132\,168\,796\,771\,975\,168$$



Slika 1: Funkcija $f(x) = 2^x$

Primetimo da brojevi izuzetno brzo rasto, što je i očekivano, jer su stepeni broja dva podskup eksponencijalne funkcije $f(x) = a^x$, gde je $a = 2$, a x iz skupa prirodnih brojeva.

Navedimo par primera i primenimo ideju ilustrovanu ranije za dekadni brojevni sistem.

Primer 1.4

$$(1101)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (13)_{10}$$

Primer 1.5

$$(11111)_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (31)_{10}$$

I naravno opštije :

Primer 1.6

$$(a_1 a_2 \dots a_{n-1} a_n)_2 = a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \dots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0$$

Primetimo da možemo na brži način izračunati dekadnu vrednost binarnog broja dopisivanjem odgovarajućih stepena broja dva iznad cifara, od desne strane ka levoj:

Primer 1.7

$$\left(\overset{8}{\underbrace{1}} \overset{4}{\underbrace{1}} \overset{2}{\underbrace{0}} \overset{1}{\underbrace{1}} \right)_2 = 8 + 4 + 1 = (13)_{10}$$

Primer 1.8

$$\left(\overbrace{1}^{32} \overbrace{0}^{16} \overbrace{1}^8 \overbrace{1}^4 \overbrace{1}^2 \overbrace{0}^1 \right)_2 = 32 + 8 + 4 + 2 = (46)_{10}$$

1.3.3 Oktalni brojevni sistem

Oktalni sistem je brojevni sistem zasnovan na osnovi 8. Time su cifre koje koristimo: $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Pokažimo par primera sa konverzijama brojeva i oktalnog sistema u dekadni.

Primer 1.9

$$(312)_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 2 \cdot 8^0 = (201)_{10}$$

Primer 1.10

$$(1373)_8 = 1 \cdot 8^3 + 3 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0 = (761)_{10}$$

1.3.4 Heksadekadni brojevni sistem

Heksadekadni sistem je brojevni sistem zasnovan na osnovi 16. Do sada smo videli dekadni, binarni i oktalni brojevni sistem, i primetili smo da brojevni sistemi koriste cifre od 0 pa sve do $n-1$, gde je n osnova brojevnog sistema. Postavlja se pitanje, kako da reprezentujemo cifre 10, 11, 12, 13, 14 i 15 koje su potrebne za heksadekadni sistem?

Ako je skup cifara dat kao $\{0, 1, 2, \dots, 9, 10, 11, 12, 13, 14, 15\}$ ne bismo se usrećili jer bismo izgubili jednoznačnost zapisa. Na primer $(112)_{16}$. Da li je tu 11 cifra, ili je 12 cifra, ili su to cifre 1, 1, 2 ?

Ali ako pokušamo sa $\{0, 1, 2, \dots, 9, a, b, c, d, e, f\}$, nećemo izgubiti jednoznačnost zapisa, pri čemu novouvedene cifre definišemo kao:

$$a = 10$$

$$b = 11$$

$$c = 12$$

$$d = 13$$

$$e = 14$$

$$f = 15$$

Nije važno da li se koriste mala ili velika slova, i u drugim literaturama će se često sresti oba.

Primer 1.11

$$(F14)_{16} = 15 \cdot 16^2 + 1 \cdot 16^1 + 4 \cdot 16^0 = (3860)_{10}$$

Primer 1.12

$$(CD32)_{16} = 12 \cdot 16^3 + 13 \cdot 16^2 + 3 \cdot 16^1 + 2 \cdot 16^0 = (52530)_{10}$$

1.4 Ostale konverzije

U ovom delu ćemo prikazati još neke zanimljive načine da obavimo konverziju iz brojevni sistema u drugi. Posebno je zanimljiv slučaj kada je osnova jednog brojevnog sistema stepen osnove drugog brojevnog sistema kao što je to slučaj sa binarnim.

Odnosno, osnova binarnog sistema je 2, dok su osnove oktalnog i heksadekadnog, 8 i 16. Primetimo da je $8 = 2^3$ i $16 = 2^4$, odnosno da su osnove ova dva sistema, stepeni broja 2. Ovo izuzetno olakšava konverziju brojeva u takvim sistemima jer se mogu grupisati cifre na odgovarajući način, i potom se jednostavno pretvoriti u odgovarajuće cifre drugog brojevnog sistema. Sledeći deo nam pokazuje navedeno kroz primere.

1.4.1 Konverzija oktalni-binarni

Neka je dat broj sledeći oktalni broj:

$$(2471)_8 = (1337)_{10}$$

Primetimo da važi:

$$\left(\overbrace{2}^{010} \overbrace{4}^{100} \overbrace{7}^{111} \overbrace{1}^{001} \right)_8$$

Odnosno:

$$(2471)_8 = (010\ 100\ 111\ 001)_2 = (1337)_{10}$$

Kako izvršiti konverziju unazad? Ideja je slična, uzmemo po tri cifre krećući sa desna, i njih direktno konverzujemo u oktalne cifre.

$$\left(\overbrace{010}^2 \overbrace{100}^4 \overbrace{111}^7 \overbrace{001}^1 \right)_2 = (2471)_8$$

Zašto po 3? Prisjetimo se da je $2^3 = 8$, odnosno sa binarnim brojem dužine 3, možemo predstaviti sve oktalne cifre.

Šta ako je dat binarni broj koji nema dovoljno cifara za naše uparivanje dužine 3. Na primer:

$$\left(\overbrace{1}^? \overbrace{100}^4 \overbrace{111}^7 \overbrace{001}^1 \right)_2$$

Jednostavno dopunimo nulama sa leve strane:

$$\left(\overbrace{001}^1 \overbrace{100}^4 \overbrace{111}^7 \overbrace{001}^1 \right)_2$$

1.4.2 Konverzija heksadekadni-binarni

Logika je slična kao za oktalni sistem, jedino ćemo uzimati 4 cifre umesto 3.

$$\left(\overbrace{F}^{1111} \overbrace{A}^{1010} \overbrace{3}^{0011} \overbrace{8}^{1000} \right)_{16} = (1111\ 1010\ 0011\ 1000)_2$$

I obrnuto:

$$\left(\overbrace{1111}^F \overbrace{1010}^A \overbrace{0011}^3 \overbrace{1000}^8 \right)_{16}$$

I naravno, kada ne možemo da uparimo brojeve na odgovarajući način, možemo dopuniti nulama kao kod oktalnog sistema.

$$\begin{array}{c} \begin{array}{cccc} ? & A & 3 & 8 \\ \overbrace{10} & \overbrace{1010} & \overbrace{0011} & \overbrace{1000} \end{array} \\ \hline \begin{array}{cccc} 2 & A & 3 & 8 \\ \overbrace{0010} & \overbrace{1010} & \overbrace{0011} & \overbrace{1000} \end{array} \end{array} \Big)_{16}$$

2 Reprezentacija podataka u računaru

U poglavlju 1 dali smo ilustraciju različitih brojevnih sistema, ali ono što jeste bio najvažniji zaključak za nas, jeste da smo sve prirodne brojeve (odnosno ne baš sve jer ih ima beskonačno) mogli da zapišemo kao binarne brojeve. Binarni brojevni sistem je izuzetno važan u svetu računarstva i upravo su binarni brojevi osnova samih računara. Ne samo da možemo zapisati sve prirodne brojeve binarno, već možemo zapisati i negativne brojeve, kao i realne. Danas se negativni brojeve zapisuju koristeći *potpuni komplement*, dok se realni brojevni zapisuju po standardu *IEEE 754* izdatim 1985. No trenutno nam nije neophodno poznavanje ova dva navedena, te ih ostavljamo za neke bolje dane.

2.1 Osnovni pojmovi

Posmatrajmo sledeći binarni broj:

$$(0011\ 1100)_2$$

Razjasnimo najpre terminologiju koja prati računarstvo još od nas-tanka.

Bit je jedna binarna cifra u binarnom broju, odnosno dati broj ima 8 bita. *Bajt* je binarni broj koji sadrži 8 bita, odnosno gore navedeni broj je jedan bajt.

Primetimo da ako broj ima 32 bita, onda on ima 4 bajta, a ukoliko je 64-bitni, onda ima 8 bajta. Na dalje možemo izvesti sistem za meru količine podataka na sledeći način (pod B podrazumevamo bajt):

- Bajt - B
- Kilobajt - KB
- Megabajt - MB
- Gigabajt - GB
- Terabajt - TB
- Petabajt - PB
- Eksabajt - EB
- Zetabajt - ZB
- Jotabajt - YB

Gde važe jednakosti:

$1\text{KB} = 1024\text{B}$
 $1\text{MB} = 1024\text{KB}$
 $1\text{GB} = 1024\text{MB}$
 $1\text{TB} = 1024\text{GB}$
 $1\text{PB} = 1024\text{TB}$
 $1\text{EB} = 1024\text{PB}$
 $1\text{ZB} = 1024\text{EB}$
 $1\text{YB} = 1024\text{ZB}$

Danas su uobičajeni hard diskovi kapaciteta od po par terabajta, dok su se pre par godina najčešće sretali hard diskovi veličine od 100 do 500 gigabajta.

2.2 Kodiranje karaktera

Pitanje koje se prirodno javlja, je kako u računar u reprezentovati bilo šta što nije broj (videli smo u poglavlju 1 kako to da uradimo)? Odgovor nije komplikovan, i zasniva se na kodiranju. Kodiranje je proces dodele kodova (brojeva) nekim simbolima koje kasnije umemo da protumačimo na osnovu *kodne tabele* ili *kodne šeme*.

2.2.1 Binarni brojevi posmatrani kao kodiranje

Radi ilustracije koncepta kodiranja, uzmimo binarne brojeve dužine 4 bita i konstruišimo kodnu tabelu na osnovu njih (tabela 1).

kod	broj
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Tabela 1: Kodna tabela binarnih brojeva

I vrednosti u drugoj koloni tabele jesu upravo dekadne vrednosti odgovarajućih binarnih brojeva sa kojima smo se bavili u poglavlju 1. Primećimo još i da ako bismo u tabeli zamenili brojeve 10, 11, 12, 13, 14 i 15 sa A, B, C, D, E, F dobili bismo direktno kodiranje heksadekadnih brojeva.

Kodne tabele nam olakšavaju kodiranje i dekodiranje simbola jer sve što je potrebno da se učini jeste da se pogledaju odgovarajuća polja u tabeli, i dođe do potrebne informacije. Na primer potrebno je da kodiramo

broj 13 u binarni broj? Pogledamo tabelu pod broj 13 i vidimo da mu odgovara broj 1101.

2.2.2 ASCII tabela

ASCII (engl. *American Standard Code for Information Interchange*) je kodna tabela izdata 1963. godine, i od tada je doživljavala razne dopune i izmene. ASCII je 7-bitni kod kojim možemo reprezentovati 128 karaktera. U vreme kada je ASCII izdat, tadašnji računari su uglavnom bili 8-bitni (današnji su 32-bitni i 64-bitni), te se poslednji osmi bit najčešće koristio kao bit parnosti kojim se proveravala greška u bajtu.

Programski jezik C koristi ASCII tabelu kao osnovnu kodnu šemu za karaktere u svom radu, iako je ASCII tabele za današnje standarde zastarela i mnogo češće se sreću Unicode i UTF-8 kodiranja, koja dozvoljavaju mnogo širi opseg karaktera koji se mogu kodirati. Slika 2 prikazuje jednu ASCII tabelu.

Karakteristi kodirani sa:

- [0, 32] su kontrolni karakteri
- [48, 57] su cifre
- [65, 90] su velika slova
- [97, 122] su mala slova

Primer 2.1 *Neka je dat sledeći kod (dekadni brojevi) koji ćemo protumačiti pomoću ASCII tabele.*

122 100 114 097 118 111 032 115 118 101 116 101

Primetimo da na osnovu ASCII tabele važe jednakosti:

122 = z
100 = d
114 = r
097 = a
118 = v
111 = o
032 = _
115 = s
118 = v
101 = e
116 = t
101 = e

Odnosno: ZDRAVO SVETE

Naravno, računari rade sa binarnim brojevima, te se ASCII tabela često daje heksadekadno (prisetimo se brzog kodiranja u binarni sistem uzimanjem 4 cifre).

Kodiranje novog reda je zavisno od operativnog sistema, te se novi red kodira na sledeći način:

- Windows - CR (Carriage Return, kod: 013), LF (Line feed, kod: 010)
- Linux - LF
- Mac OS X - CR

0	<NUL>	32	<SPC>	64	@	96	`
1	<SOH>	33	!	65	A	97	a
2	<STX>	34	"	66	B	98	b
3	<ETX>	35	#	67	C	99	c
4	<EOT>	36	\$	68	D	100	d
5	<ENQ>	37	%	69	E	101	e
6	<ACK>	38	&	70	F	102	f
7	<BEL>	39	'	71	G	103	g
8	<BS>	40	(72	H	104	h
9	<TAB>	41)	73	I	105	i
10	<LF>	42	*	74	J	106	j
11	<VT>	43	+	75	K	107	k
12	<FF>	44	,	76	L	108	l
13	<CR>	45	-	77	M	109	m
14	<SO>	46	.	78	N	110	n
15	<SI>	47	/	79	O	111	o
16	<DLE>	48	0	80	P	112	p
17	<DC1>	49	1	81	Q	113	q
18	<DC2>	50	2	82	R	114	r
19	<DC3>	51	3	83	S	115	s
20	<DC4>	52	4	84	T	116	t
21	<NAK>	53	5	85	U	117	u
22	<SYN>	54	6	86	V	118	v
23	<ETB>	55	7	87	W	119	w
24	<CAN>	56	8	88	X	120	x
25		57	9	89	Y	121	y
26	<SUB>	58	:	90	Z	122	z
27	<ESC>	59	;	91	[123	{
28	<FS>	60	<	92	\	124	
29	<GS>	61	=	93]	125	}
30	<RS>	62	>	94	^	126	~
31	<US>	63	?	95	_	127	

Slika 2: Regularna ASCII tabela

Bibliografija

- [1] Nenad Mitić, *Uvod u organizaciju računara*, Matematički fakultet, 2009.