

Predstavljanje različitih tipova podataka

Računari danas čuvaju velike količine podataka, brojeve, slike, zvuk, tekst. Kako se svi ti podaci beleže u računaru.

Kodiranje podatak korišćenjem binarnog brojnog sistema

Računari sve podatke zapisuju korišćenjem brojeva. Memorija računara je niz bistabilnih elementata, elemenata koji mogu biti u jednom od dva različita stanja. Na primer elektronsko kolo može biti pod naponom ili bez napona, deo trake može biti namagnetisan ili razmagnetisan, emulzija kompak diska može biti ogrebana ili ne. Ako svako stanje obeležimo različitim ciframa moguće je koristiti samo dve različite cifre, odnosno binarni sistem zapisa brojeva.

Zamislimo memoriju kao niz uključenih (1) i neuključenih sijalica (0). Ako imamo 8 sijalica kako zapisujemo broj 1, 5, 13. Da li je moguće zapisati brojeve 123, 256?

Ako imamo 3 sijalice onda imamo sledeće kombinacije: 000, 001, 010, 011, 100, 101, 110, 111, tačno 8 kombinacija. Ako imamo k sijalica onda imamo 2^k različitih zapisa.

Brojevni sistemi i binarni brojni sistem

432 je 4 stotine 3 desetice i 2 jedinice

$$432 = 4 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$$

U binarnom sistemu skup cifara je 0, 1 osnova je 2

$$13 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (1101)_2$$

$$(10101)_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 4 + 1 = (21)_{10}$$

$$2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}$$

$$1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$$

binarni sistem osnova 2 cifre 0, 1

oktalni sistem osnova 8 cifre 0, 1, 2, 3, 4, 5, 6, 7

heksadekaski sistem osnova 16 cifre 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Ako je osnova b i cifre $c_0, c_1, c_2, c_3, \dots, c_k$ $0 \leq c_i < b$ zapis

$$(c_k c_{k-1} \dots c_1 c_0)_b \text{ označava broj } c_k \cdot b^k + c_{k-1} \cdot b^{k-1} + \dots + c_1 \cdot b^1 + c_0 \cdot b^0$$

Prevođenje celih brojeva

432 prvu cifru možemo dobiti kao količnik pri deljenju sa 100

4321 prvu cifru dobijamo kao količnik pri deljenju sa 1000

Ne postoji uniforman način da dobijemo prvu cifru

4321 poslednju cifru dobijamo kao ostatak pri deljenju sa 10

432 poslednju cifru dobijamo kao ostatak pri deljenju sa 10

U zašisu $c_k \cdot b^k + c_{k-1} \cdot b^{k-1} + \dots + c_1 \cdot b^1 + c_0 \cdot b^0$ tj. $c_k \cdot b^k + c_{k-1} \cdot b^{k-1} + \dots + c_1 \cdot b^1 + c_0$ svi sabirci osim poslednjeg deljivi su sa b $c_k \cdot b^k + c_{k-1} \cdot b^{k-1} + \dots + c_1 \cdot b^1 + c_0 \cdot b^0 = (c_k \cdot b^{k-1} + c_{k-1} \cdot b^{k-2} + \dots + c_1) \cdot b + c_0$

$$(c_k c_{k-1} \dots c_1 c_0)_b = (c_k c_{k-1} \dots c_1)_b + c_0$$

Ostatak pri deljenju broja osnovom b daje poslednju cifru u zapisu tog broja u osnovi b, dok celobrojni količnik pri deljenju broja osnovom b predstavlja broj čiji se zapis u osnovi b dobija brisanjem poslednje cifre iz zapisa polaznog broja u osnovi b.

Prikažimo postupak određivanja cifra u zapisu broja 4321 u osnovi 10

$$4321=432*10+1$$

$$432=43*10+2$$

$$43=4*10+3$$

$$4=0*10+4$$

4321	432	43	4	0
1	2	3	4	



Prikažimo postupak određivanja cifra u zapisu broja 13 u osnovi 2

$$13=2*6+1$$

$$6=2*3+0$$

$$3=2*1+1$$

$$1=0*2+1$$

$$(13)_{10}=(1101)_2$$

13	6	3	1	0
1	0	1	1	



Prevedi broj $(3F5)_{16}$ u binarni sistem, kako je $16=2^4$, pokazat ćemo da je dovoljno svaku cifru prevesti u binarni sistem i to sa 4 binarne cifre.

$$(3F5)_{16}=3*16^2+15*16^1+5*16^0=3*2^8+15*2^4+8*2^0=(2^1+2^0)2^8+(2^3+2^2+2^1+2^0)*2^4+(2^2+2^0)*2^0=2^9+2^8+2^7+2^6+2^5+2^5+2^2+2^0=(1111110101)_2$$

Prevođenje prirodnih brojeva iz heksadekadnog u binarni sistem vršimo tako što prevodimo svaku cifru ponaosob korišćenjem sledeće tablice

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

Prevedi broj $(1010101110)_2$ u heksadekadni sistem:

$$(1010101110)_2=1*2^9+0*2^8+1*2^7+0*2^6+1*2^5+0*2^4+1*2^3+1*2^2+1*2^1+0*2^0=$$

$$(1*2^1+0*2^0)*2^8+(1*2^3+0*2^2+1*2^1+0*2^0)2^4+1*2^3+1*2^2+1*2^1+0*2^0=$$

$$2*16^2+A*16^1+E=(2AE)_{16}$$

Prevođenje prirodnih brojeva iz binarnog u heksadekadni sistem vršimo tako što od kraja broja izdvajamo po 4 binarne cifre i na osnovu tablice prevodimo svaku grupu od 4 cifre u odgovarajuću heksadekadnu cifru.

Slično se vrši prevođenja iz binarnog u oktalni sistem, i obrnuto, tada se uzimaju po 3 cifre i koristimo sledeću tablicu:

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

Binarni sistem se koristi jer je njegov zapis tehnički jednostavno realizuje korišćenjem bistabilnih elemeneta. Brojevi zapisani u binarnom sistemu izuzetno su dugački, pa je ljudim teško da rade direktno sa njima, a s obzirom na jednostavnost prevođenja iz binarnog u heksadekadni sistem i obrnuto, heksadekadni sistem se koristi da bi se skratio zapis dugačkih binarnih brojeva. Na primer umesto 32 binarne cifre potrebno je samo 8 heksadekadnih cifara, umesto 17 binarnih cifara potrebno je 5 heksadekadnih cifara. Iako ima slična svojstva oktalni sistem se manje koristi jer je step skraćivanja zapisa malo manji.

Prevođenje razlomljenih brojeva

$$0,1253=1/10+2/100+5/1000+3/1000$$

$$0,1253*10=1,253=1+2/10+5/100+3/100$$

$$0,1253 \cdot 10 = 1,253$$

$$0,253 \cdot 10 = 2,53$$

$$0,53 \cdot 10 = 5,3$$

$$0,3 \cdot 10 = 3 + 0$$

$$X = (0, c_{-1} c_{-2} \dots c_{-k} \dots)_b$$

Neka je $X_1 = X$

$$X_1 \cdot b = (0, c_{-1} c_{-2} \dots c_{-k} \dots) \cdot b = \left(\frac{c_{-1}}{b} + \frac{c_{-2}}{b^2} + \dots + \frac{c_{-k}}{b^k} + \dots \right) \cdot b$$

$$= c_{-1} + \frac{c_{-2}}{b} + \dots + \frac{c_{-k}}{b^{k-1}} + \dots = c_{-1} + (0, c_{-2} \dots c_{-k} \dots)$$

Cifra c_{-1} dobija se kao ceo deo proizvoda $X_1 \cdot b$, oznacimo razlomljen deo proizvoda sa X_2

$$X_2 = (0, c_{-2} c_{-3} \dots c_{-k} \dots)_b$$

$$X_2 \cdot b = (0, c_{-2} \dots c_{-k} \dots) \cdot b = \left(\frac{c_{-2}}{b} + \dots + \frac{c_{-k}}{b^{k-1}} + \dots \right) \cdot b = c_{-2} + \frac{c_{-3}}{b} + \dots + \frac{c_{-k}}{b^{k-2}} + \dots$$

$$= c_{-2} + (0, c_{-3} \dots c_{-k} \dots)$$

$$X_3 \cdot b = (0, c_{-3} \dots c_{-k} \dots) \cdot b = c_{-3} + (0, c_{-4} \dots c_{-k} \dots)$$

....

$$X_k \cdot b = (0, c_{-k} \dots) \cdot b = c_{-k} + (0, c_{-k-1} \dots)$$

...

Ako postoji q tako da je X_q jednako 0 dobijamo „tačno“ prevođenje $X = (0, c_{-1} c_{-2} \dots c_{-q})_b$

Ako je za svako q $X_q \neq 0$, dobijamo „približno“ prevođenje $X \approx (0, c_{-1} c_{-2} \dots c_{-k})_b$

$$(0,84375)_{10} = (?)_2$$

$$0,84375 \cdot 2 = 1,68750 = 1 + 0,6875$$

$$0,6875 \cdot 2 = 1,3750 = 1 + 0,3750$$

$$0,375 \cdot 2 = 0,75 = 0 + 0,75$$

$$0,75 \cdot 2 = 1,5 = 1 + 0,5$$

$$0,5 \cdot 2 = 1,0 = 1 + 0$$

$$(0,84375)_{10} = (0,11011)_2$$

0,84375	0,6875	0,3750	0,750	0,50	0
0	1	1	0	1	1



$$(0,4)_{10} = (?)_2$$

$$0,4 \cdot 2 = 0 + 0,8$$

$$0,8 \cdot 2 = 1,6 = 1 + 0,6$$

$$0,6 \cdot 2 = 1,2 = 1 + 0,2$$

$$0,2 \cdot 2 = 0,4 = 0 + 0,4$$

...

$$(0,4)_{10} = (0,011001100110\dots)_2$$

0,4	0,8	0,6	0,2	0,4	
0	0	1	1	0	

Prilikom prevođenja razlomljenih brojeva iz binarnog sistema u heksadekadni sistem, polazimo od decimalnog zareza na desno izdvajamo grupe od 4 cifre, po potrebi kod poslednje grupe dodamo nule, i prevedemo svaku grupu ponaosob.

$$(0,0101100111)_2 = (0,010110011100)_2 = (0,59C)_{16}$$

Prevođenje razlomljenog broja iz heksadekadnog zapisa u binarni vršimo tako što svaku cifru prevedemo pomoću 4 binarne cifre (na osnovu tablice).

$$(0,A4B)_{16}=(0,101001001011)_2$$

Prevođenje mešovityh brojeva

prevedemo posebno ceo deo posebno razlomljen deo a zatim ih saberemo (spojimo)

$$(67,875)_{10}=(?)_2$$

Prevodimo ceo deo (delimo sa 2 i zapisujemo ostatak)

67	33	16	8	4	2	1	0
1	1	0	0	0	0	1	

$$(67)_{10}=(10000011)_2$$

Prevodimo razlomljeni deo (mnozimo sa 2 i zapisujemo ceo deo, nastavljamo sa razlomljenim delom)

0,875	0,75	0,5	0
0	1	1	1

$$(0,875)_{10}=(0,111)_2$$

Prema tome

$$(67,875)_{10}=(10000011,111)_2$$

$$(A2,7B)_{16}=(?)_2$$

$$(A2,7B)_{16}=(10100010,01111011)_2$$

$$(10110111,011010)_2=(?)_{16}$$

$$(1011011,0110101)_2=(01011011,01101010)_2=(5B,6A)_{16}$$

O načinu registrovanja prirodnih, celih i decimalnih brojeva biće reči kasnije.

Jedinice za merenje količine podataka

Sve informacije se zapisuju u obliku binarnih brojeva, pa je osnovna jedinica mere količine podataka broj binarnih cifara ukupno upotrebljenih za zapis. Umesto termina binarna cifra koristi se termin **bit** binary digit. Bit je mala količina informacija, jednim bitom se mogu predstaviti samo broj 0 i broj 1. Već za predstavljanje broja 2 potrebno je dve cifre $(10)_2$.

Količina informacija koja se može registrovati jednim bitom je mala, bitovi se udružuju u grupe od 8, 16, 32, 64 bita. Grupa od 8 bita zove se bajt (**Byte**). Jednim bajtom možemo registrovati 2^8 (256) različitih vrednosti. Memorija računara je organizovana kao niz bajtova i svaki bajt ima svoj redni broj, tzv. adresu u memoriji. Svaki bajt ima svoju **adresu**, koja se koristi prilikom čitanja informacija iz memorije i prilikom zapisa informacija u memoriju.

Osnovna jedinica za merenje količine podataka je bit. Osnovna jedinica mere količine podataka u današnjim računarima je bajt.

Kapacitet memorije izražava se brojem bajtova koje računar ima.

$$2^{10} \text{ Byte} = 1024 \text{ Byte} = 1\text{KB (Kilo Bajt)}$$

$$2^{10} \text{ KB} = 1024 \text{ KB} = 1\text{MB (Mega Bajt)}$$

$$2^{10} \text{ MB} = 1024 \text{ MB} = 1\text{GB (Giga Bajt)}$$

$$2^{10} \text{ GB} = 1024 \text{ GB} = 1 \text{ TB (Tera Bajt)}$$

$$2^{10} \text{ TB} = 1024 \text{ TB} = 1 \text{ PB (Peta Bajt)}$$

Ako se svaki znak u tekstu predstavlja jednim bajtom, onda 1KB sadrži 1000 znakova. Ako jedan red sadrži 80 znakova, onda je za 12 redova čistog teksta bez dodatnih informacija o njegovom izgledu potreban 1KB.

Svaka tačkica slike (piksel) najčešće se zapisuje pomoću 3 bajta, pa slika koja ima 1000*1000 tačkica zauzima 3MB. Međutim postoje različite metode kompresije slike (npr JPEG) kojima se slika zapisuje sa manje bajtova uz neznatan gubitak kvaliteta, tako slika veličine 1000*1000 zauzima nekoliko stotina kilobajta.

Slično jedan minut zvuka zauzima 10MB, a kad se kompresuje može da zauzima i manje od 1MB.

Film koji traje oko 2 sata, snimljen sa manjom kompresijom (kada se koristi DVD) zauzima nekoliko gigabajta, a snimljen sa većom kompresijom (koristi se DivX) zauzima manje od jednog gigabajta.

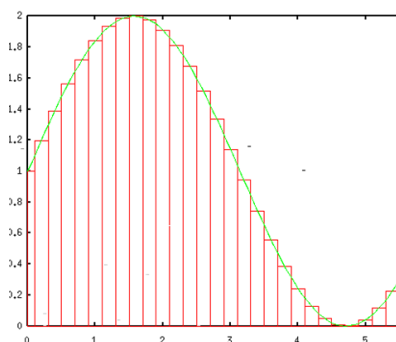
Svaku memoriju karakteriše njen kapacitet – količina podataka koji mogu u njoj da se zapišu. Danas se ove veličine izražavaju u gigabajtima.

Digitalni zapis podataka

Digitalni zapis podrazumeva da su podaci zapisani u obliku brojeva (digitalno doba).

Podaci su zapisani analogno ako se kontinulane promene signala zapisuju odgovarajućim kontinualnim promenama medijuma.

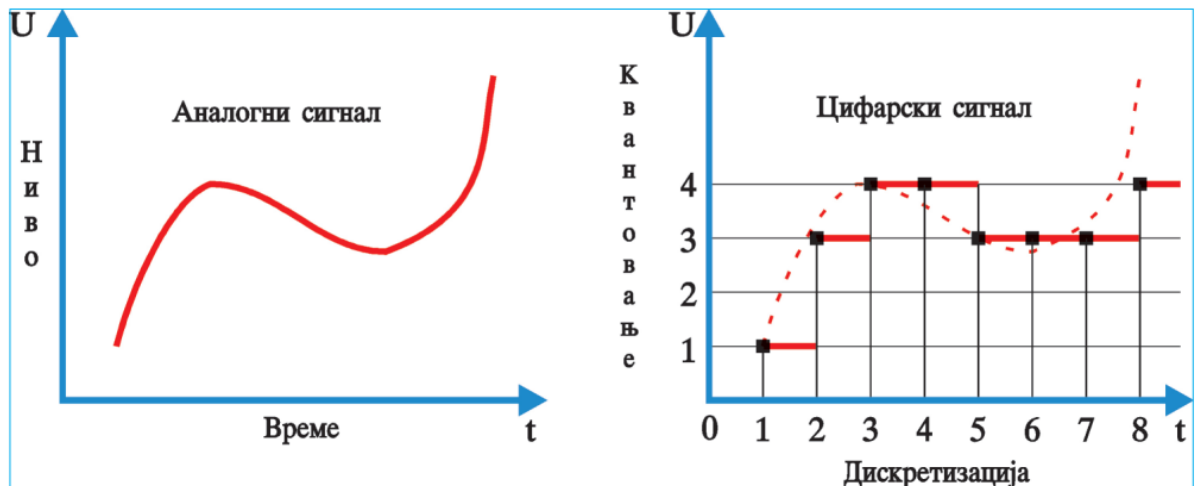
Primer analognog zapisa informacija je korišćenjem starog analognog foto aparati sa filmom premazanem emulzijom koja je osetljiva na svetlost. Slika koja bi se dobila na filmu nakon njegovog razvijanja (razne hemikalije), doduše u negativu, odgovarala bi sceni koja se nalazila ispred fotoaparata u trenutku fotografisanja. I nakon izrade fotografije (ponovo osvetljavanje i hemikalije) na osnovu negativa dobila bi se fotografija koja potpuno odgovara sceni koju smo fotografisali. Postojala je odgovarajuća analogija između onoga što smo slikali, slike na negativu i konačne slike na papiru.



Digitalni zapis zvuka

Zvuk čujemo zahvaljujući talasima koji se prostiru kroz vazduh. Zvuk je kontinualan (neprekidan) i usvakom trenutku definisan je njegov intezitet (svakom realnom broju iz nekog intervala možemo dodeliti vrednost funkcije inteziteta zvučnog talasa). Prilikom predstavljanja zvuka na računara nije moguće zapisati njegov intezitet u proizvoljnom vremenskom trenutku (trenuci su kontinulani i ima ih beskonačno mnogo, a memorija računara je ograničena). Audio snimci se danas zapisuju tako što uređaj za snimanje meri intezitet u pravilnim vremenskim intervalima (više od 40 000 puta u sekundi) i brojeve koji

iskazuju intezitet (semplove - uzorke) beleže se u memoriju. Ovako dolazi do diskretizacije vremena (vremenski trenuci više nisu prikazani kontinualnim decimalnim vrednostima već diskretnim decimalnim vrednostima). I vrednost inteziteta zvuka je realna broj i ne može se zapisati savršeno ispravno, već se zapisuje približno. Pa imamo i diskretizaciju inteziteta zvuka, i on se predstavlja korišćenjem konačnog broja dopuštenih vrednosti (2 bajta što dopušta oko 6500 različitih vrednosti). Bolji kvalitet se dobija ako se zvuk snima sa dva različita mikrofoba i reprodukuje se na dva zvučnika (stereo – zvuk).



Kvalitet analogno-digitalne transformacije karakterišu:

- rezolucija – broj nivoa kvantovanja, odnosno broj bitova (8, 16, 24) koji se koriste za regitrovanje jednog merenja.
- frekvenca diskretizacije – broj merenja u jednoj sekundi.

Na opisan način se evidentiraju informacije u .wav formatu. Najpopularniji kompresovani zvučni format je .mp3.

Digitalni zapis slike

Prostor slike je diskretizovan izdelfjen na kvadartiće, koji se zovu pikseli (pixel –picture element). Što je piksela više kvalitet slike je bolji. Stepjen jasnosti slike zavisi od broja linija na ekranu i broja piksela u liniji, koji predstavljaju rezoluciju monitora.

Svetlost svake boje može da se razloži na svetlost crvene, zelene i plave boje. Na primer ljubičasta je ombinacija crvene i plave, žuta kombinacija crvene i zelene. Kombinacija sve tri boje daje belu, a odsustvo svetlosti daje crnu boju. Za zapis inteziteta svake komponente koristi se konačan broj dopuštenih vrednosti (najčešće 256 jer za zapis se koristi 1bajt), pa je i zapis inteziteta svetlosti diskretizovan. RGB model koristi tri bajta za Red – Green – Blue boje. (to znači da za lik dimenzija 8x8 piksela treba 8x8x3 bajta=192bajta).

255,0,0 – crvena; 0,255,0 – zelena; 0,0,255 – plava; 255,255,255 –bela; 0,0,0 – crna;

Broj bitova koji se koriste za kodiranje boje jednog piksela naziva se **dubina boje**.

Ako sliku posmatramo kao niz piksela, a svaki piksel se predstavljaj brojevima, onda se slika predstavlja nizom brojeva. Za zapis slike na opisan način kaže se da je rasterski ili bitmapiran.

Slika po RGB modelu se čuva u formatu .bmp (prg. Paint).

Kod .gif formata je ograničena paleta boja na 256, tj. za čuvanje boje je dovoljan jedan bajt. Kompresovani .jpg ili .jpeg formati umesto informacije o svakom pikselu formiraju informacijom o grupi piksela smanjujući na taj način dimenzije 5-6 puta.

Za zapis crteža i ilustracija koriste se vektorski zapis, slika je predstavljanja matematičkim opisom geometrijskih likova koji je sačinjavaju (prave, krugovi, ...)

Kodiranje karaktera

Tekst je na neki način već diskretan pa ga je relativno lako pretvoriti u niz brojeva.

Tekst je niz karaktera. Karakteri su svi elementi od kojih se sastoji tekst: slova, cifre, interpukcijski znaci, specijani znaci (razmak, prelazak u nov red). Da bi se tekst predstavio dovoljno je kodirati karaktere. Kodiranjem se svakom karakteru pridruži jedan broj kod. Na primer ako se dogovorimo da je kod slova A 1, a slova B 2, onda se reč BABA kodira sa 2121. Da bi se tekst unet na jednom računaru video ispravno na drugom računaru važno je da dogovo bude standardizovan i da svi koriste standardne kodove. Iz tog razloga sastavljeni su standardne tabele u kojima su popisani karakteri i njihovi kodovi. Ovakve tablice nazivaju se **kodne strane** ili kodne šeme.

U kodnoj strani za svaki kod je naznačeno koji je to karakter ali nije preciziran njegov izgled. Numerički kod karaktera je njegova unutrašnja reprezentacija, dok je ono kako ga mi vidimo na ekranu njegova spoljna reprezentacija. Spoljna reprezentacija karaktera nije određena u zapisu čistog teksta, već nju određuju programi koji čitaju i prikazuju karakter. Za spoljnu reprezentaciju koriste se sličice - glifovi koji se prikazuju na ekranu kada se prikazuje karakter. Pojedinačni glifovi sakupljaju se u grupe, koje se nazivaju fontovi.

Prilikom rada sa tekstem uvek se koriste dva postupka:

- dok se kuca tekst se pretvara u niz brojeva koristeći pri tome neku kodnu stranu
- kad se prikazuje ovaj niz brojeva pretvara se natrag u karaktere opet korišćenjem neke kodne strane.

Sam niz brojeva ne prati uvek informacija koja kodna strana je korišćena. Može se desiti da program u koji se unosi tekst koristi jednu kodnu stranu a program koji prikazuje tekst drugi kodnu stranu, tako se na ekranu prikazuju karakteri koje autor nije uneo.

Najznačajnije kodne strane

ASCII - America Standard Code for Information Interchange – za kodiranje 128 karaktera koji se koriste na engleskom govornom području. Prvih 32 karaktera su kontrolni karakteri koji nemaju grafičku interpretaciju. CR i LF prelazak u novi red.

slovo A ima kod $(42)_{16}=65$ slovo B ima kod $(43)_{16}=66$

Velika slova se razlikuju tačno u jednoj cifri, tako i mala slova i cifre.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Unicode – jedinstveni standard koji obuhvata sve karaktere svih jezika. Cilj je da bude :

- univerzalan – obuhvata sve jezike
- jedinstven – nema dupliranja karaktera
- uniforman – svi kodovi iste dužine.

Unicode sadrži 65536 karaktera, čiji kodovi se zapisuju pomoću dva bajta. Prvih 128 karaktera se poklapa sa ASCII tabelom, sledećih 128 čine karakteri zapadno evropske latinice, zatim ostali latinični karakteri (među kojim i naši š,ć...), zatim ćirilica, grčka slova itd. Kodiranje se vrši pomoću dva bajta UCS-2 (Universal Character Set). Međutim ako se tada

kodira tekst koji na engleskom jeziku koriste se samo 7 bita, tada svaki drugi bajt ima vrednost 0. Ovo je neracionalno i predstavlja problem starim programima. Zato se često koristi modifikacija Unicoda UTF-8 (Unicode Transformation Format) - znak kodiramo jednim bajtom ako sadrži samo 7 najnižih bitova (reč je o ASCII znaku), ostali latinični karakteri, grča slova i ćirilični karakteri kodiraju se sa 2 bajta, dok se kineski karakteri kodiraju u obliku 3 bajta. Pri UTF-8 karakteri se kodiraju sa promenljivim brojem bajtova, postoji precizan postupak za kodiranje i dekodiranje.