Kódovanie znakov

Na to, aby medzi sebou mohli komunikovať nielen uzavreté skupiny, ale aj široké masy používateľov, bol potrebný jednotný spôsob kódovania znakov. Prvým univerzálnym a rozšíreným kódom sa stal 7 – bitový **kód ASCII** (*American Standard Code for Information Interchange*), publikovaný v roku 1963. Bol určený na zabezpečenie vzájomnej komunikácie medzi rôznymi (počítačovými) systémami. ASCII presne určuje, akou kombináciou bitov sú reprezentované jednotlivé znaky (napr. A má kód 1000001 (v prepočte na DPS číslo predstavuje 65), B – 1000010(66), a – 1100001 (97) atď.).

ASCII kóduje 128 rôznych hodnôt, pričom prvých 32 je nezobraziteľných – popisujú / riadia, ako bude vyzerať výstup, potom nasledujú rôzne špeciálne znaky (!, “, #), číslice, veľké a malé písmená bez diakritiky.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Kód sa pomerne dlho používal, no v čase, keď anglická abeceda prestala byť jedinou používanou a počítače začali využívať i národné abecedy (napr. znaky s dĺžňami a mäkčeňmi), prestalo byť 7 bitov reprezentujúcich 128 rôznych hodnôt, ktoré ASCII kód využíval, postačujúcich.

Na kódovanie znakov národnej abecedy sa začalo využívať ďalších 128 znakov, ktoré sa získali pridaním ďalšieho bitu k ASCII kódu. Na základe tejto filozofie vzniklo pre každú národnú abecedu niekoľko tabuliek, v ktorých bolo popísané, aký znak sa na ktorej pozícii nachádza. Napr. slovenskú abecedu podporujú *Latin 2 (ISO 8859-2), Windows Latin 2 (Windows 1250), Kamenický (895)* atď. Bohužiaľ, rozdiely medzi jednotlivými kódovaniami spôsobovali a ešte i dnes niekedy spôsobujú (najmä pri starších stránkach), že znaky uložené korektne v jednej kódovej stránke sa pri prepnutí na inú správne nezobrazia – najväčšie problémy sú so znakmi „č“, „š“ a „ž“.

Po niekoľkých pokusoch o vylepšenie bola filozofia kódovania znakov postavená do inej roviny. Zatiaľ čo v prípade 7 a 8 – bitového kódovania bolo zvykom pre každú abecedu (latinka, azbuka, cyrilika, arabské písmo atď.) používať osobitné znakové sady, v prípade **Unicode** sa stalo hlavným cieľom zakódovať do jednej abecedy čo najviac znakov vo svete používaných. Hoci pôvodná idea (zakódovať do jednej abecedy všetky

znaky) sa postupne zredukovala, **Unicode** sa presadil a stal sa kódovacou schémou, ktorá sa v súčasnosti používa prakticky všade. **Unicode** je medzinárodný štandard, ktorého cieľom je definovať kódovaciu schému (každému znaku priradené jednoznačné číslo) schopnú reprezentovať väčšinu znakov používaných v písaných jazykoch spolu s inými symbolmi.

V praxi sa môžeme stretnúť s troma verziami: *UTF – 8 (web)*, *UTF – 16 (Java a Windows)* a *UTF – 32 (Linux)*. Líšia sa spôsobom kódovania aj počtom bajtov, ktoré využívajú na kódovanie. Napr. *UTF – 8* využíva na kódovanie jeden až štyri bajty, UTF – 16 používa jednu alebo dve dvojbajtové časti, no prevod medzi nimi je bezproblémový.

Ako sme si povedali, *UTF – 8* využíva jeden až štyri bajty. To znamená, že rôzne **Unicode** znaky zaberajú rôzny počet bajtov. Pre ASCII znaky (A, B, C, ..., Z atď.) používa *UTF – 8* jeden bajt na znak. V skutočnosti využíva presne ten istý bajt, ako ASCII, a teda prvých 128 znakov (0 – 127) sa v *UTF – 8* nedá rozlíšiť od ASCII. Znaky z „rozšírenej latinky“, ako sú ň, ď, ä, atď. budú zaberať dva bajty. Čínske znaky zaberajú tri bajty. Väčšina bežných emojis zaberajú štyri bajty.

Nevýhody *UTF – 8*: Pretože každý znak zaberá rôzny počet bajtov, je nájdenie n – tého znaku lineárnou zložitosťou, tzn. čím je reťazec dlhší, tým dlhšie budeme znak na určenej pozícii vyhľadávať. Pri kódovaní znakov na bajty a dekódovanie bajtov na znaky sa musíme navyše zaoberať ďalšími manipuláciami s bitmi.

Výhody *UTF – 8*: Kódovanie bežných ASCII znakov je extrémne efektívne. Pri kódovaní znakov z rozšírenej latinky nie je horšie ako UTF – 16. Pre čínske znaky lepšie ako *UTF – 32*. A vďaka jednoznačnému spôsobu manipulácie s bitmi tu neexistujú žiadne problémy s poradím bajtov. Dokument kódovaný v *UTF – 8* používa na každom počítači presne rovnakú postupnosť bajtov.