Temat : Systemy liczenia. Kod ASCII (czytaj „aski”).

Dlaczego w informatyce zajmujemy się systemami liczenia ? Otóż komputery wykorzystują inny system liczenia niż ten, z którego korzystamy na co dzień. Na co dzień posługujemy się systemem dziesiątkowym. Korzystamy z dziesięciu cyfr : 0,1,2,3,4,5,6,7,8 i 9. Liczba dziesięć jako podstawa liczenia, liczba dziesięć a więc również jej wielokrotności - sto, tysiąc, milion - czyli 10^2, 10^3, 10^6.

Dlaczego na co dzień posługujemy się systemem dziesiątkowym (decymalnym)? Można próbować się domyślić. Człowiek posiada dziesięć palców i zanim powstały jakiekolwiek urządzenia do liczenia, palce zawsze były "pod ręką". Podstawowe urządzenie do liczenia (liczydło) nosiliśmy przy sobie.

Dlaczego zaczęliśmy w takim razie używać w technice komputerowej systemu dwójkowego (binarnego)?

Sprawa związana jest z techniką przesyłania informacji. Przy przesyłaniu informacji systemem dziesiątkowym, gdzie musimy rozpoznać dziesięć różnych stanów np. wartości napięcia elektrycznego, informacja jest bardziej narażona na zniekształcenia (przy przesyle np. przewodem miedzianym rezystancja, pojemność, indukcyjność powoduje zmiany napięcia). W systemie dwójkowym mamy dwa stany 0 i 1, bo takie cyfry w tym systemie funkcjonują. Nawet przy spadku wartości napięcia łatwo jest rozpoznać brak sygnału od sygnału, przy zapisie magnetycznym jest to namagnesowanie lub jego brak. Przy obecnym stanie techniki system binarny jest najbardziej odporny na zniekształcenia. Dlatego telewizja cyfrowa, radio cyfrowe, gdzie przy przesyle sygnału wykorzystywany jest system dwójkowy.

Wszystkie systemy liczenia są systemami pozycyjnymi. Cyfra w zależności od pozycji na jakiej stoi ma odpowiednią wagę. Czyli mamy w systemie dziesiątkowym pozycję jedności, dziesiątek, setek, tysięcy itd. Dla przykładu liczba 3679 to 9 jedności, 7 dziesiątek, 6 setek i 3 tysiące i to jest dla nas oczywiste.

Jak by wyglądała ta liczba gdybyśmy zapisali ją w postaci sumy iloczynów? $3679=3\cdot1000+6\cdot100+7\cdot10+9\cdot1$ lub w odwrotnej kolejności (prawo przemienności) $9\cdot1+7\cdot10+6\cdot100+3\cdot1000$ a to można zapisać $9 \cdot 10^0 + 7 \cdot 10^1 + 6 \cdot 10^2 + 3 \cdot 10^3$ w dalszym ciągu jest to liczba 3679 powstaje więc ogólny wzór na wartość liczby zapisanej w systemie dziesiątkowym

$\mathbf{w=b_0 \cdot 10^0 + b_1 \cdot 10^1 + b_2 \cdot 10^2 + \cdots + b_n \cdot 10^n}$ gdzie b_n - cyfra stojąca na n-tej pozycji

Po co ten wzór? Żeby lepiej zrozumieć, że można używać dowolnego systemu liczenia. Liczba to ciąg cyfr używanych w danym systemie, wartość każdej z nich zależy od pozycji, na której stoi, a waga pozycji zależy od kolejnej potęgi podstawy systemu. W systemie dziesiątkowym podstawą systemu jest **dziesięć** i tyloma cyframi też system się posługuje.

Co będzie podstawą w systemie dwójkowym ? Oczywiście - **dwa** i będą tylko dwie cyfry (0,1). A w piątkowym ? - **pięć** i cyfry (0,1,2,3,4), w szesnastkowym (heksadecymalnym) - **szesnaście** i cyfry (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) – to, że tutaj pojawiły się litery zdecydowała konieczność pojawienia się znaków graficznych opisujących kolejne cyfry czyli 10,11,12,13,14 i 15 a najłatwiej było przyjąć znane znaki graficzne istniejące poza tym na klawiaturze. Jak widzimy można liczyć w dowolnym systemie.

SYSTEM DWÓJKOWY (BINARNY)

System dwójkowy równie dobrze nadaje się do zapisu liczb i operacji na nich jak system dziesiątkowy. Naszej wyobraźni przyzwyczajonej do dziesiątkowego jest tylko to trudniej przyswoić.

System dwójkowy czyli każda liczba zapisana jest przy pomocy cyfr 0 i 1 Weźmy np. taką liczbę **10011** na pierwszy rzut oka nie znamy jej wartości, ale znamy przecież wzór, trzeba tylko zamienić w nim podstawę potęgi z 10 na 2 i już mamy nowy wzór : $W=b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 + \cdots + b_n \cdot 2^n$ trzeba tylko pamiętać o tym, że jest to system pozycyjny i wpisując cyfry z konkretnej liczby musimy je wpisać na odpowiedniej pozycji (zaczynając od prawej - od pozycji najmniej znaczącej), a więc jaką wartość w systemie dziesiątkowym ma liczba **10011**
$W=1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4$ a więc
$W = 1 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 0 \cdot 8 + 1 \cdot 16$
W = 19 czyli jesteśmy w stanie zrozumieć już teraz wartość liczby **10011**

A zatem liczba zapisana jako **10011** w systemie dwójkowym to liczba **19** w systemie dziesiątkowym. Proste - prawda ? Przy odpowiedniej wprawie zauważymy, że **11111** to liczba **31** a **11111111** to liczba **255** trzeba tylko znać kolejne potęgi liczby **2** i już.

Żeby ułatwić sobie przeliczanie korzystamy z tabelki, w tabelce osiem pozycji nie jest przypadkiem, każda pozycja to jeden bit informacji a 8 bitów stanowi najmniejszą jednostkę pamięci jaką jest bajt.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Mając dowolną liczbę w systemie binarnym podpisujemy ją pod tabelkę w odpowiednim miejscu, a później już tylko sumujemy te potęgi liczby dwa gdzie występuje cyfra 1.

Np. 10110110

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	0

czyli 128+32+16+4+2=182

Podsumowując, żeby przeliczyć system dwójkowy na dziesiątkowy korzystamy bezpośrednio ze wzoru $W=b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 + \cdots + b_n \cdot 2^n$ lub z tabelki, która jest ilustracją tego wzoru.

Jeszcze wniosek : jak rozpoznać na pierwszy rzut oka liczbę parzystą od nieparzystej, w parzystej ostatnią cyfrą jest zero a w nieparzystej jeden. W sumie wszystkich potęg liczby 2 tylko 2^0 jest wartością nieparzystą czyli jeżeli nie występuje to suma liczb parzystych jest zwasze liczbą parzystą.

A teraz w drugą stronę. Mam liczbę w systemie dziesiątkowym i chcę ją zapisać w systemie dwójkowym. Np. 137 , pierwsza metoda to dzielenie przez podstawę systemu czyli tutaj przez 2

137 : 2 = 68 r 1
68 : 2 = 34 r 0
34 : 2 = 17 r 0
17 : 2 = 8 r 1
8 : 2 = 4 r 0
4 : 2 = 2 r 0
2 : 2 = 1 r 0
1 : 2 = 0 r 1

Teraz tylko uzyskany wynik musimy zapisać w linii poziomej, która w takim razie to cyfra b_0.
Najbardziej skrajna cyfra z prawej strony to pierwsza reszta z dzielenia w więc wynik to 10001001 bo (128+8+1 = 137)

Można skorzystać też bezpośrednio z tabelki próbując przez dodawanie w pamięci wybrać, które wartości trzeba dodać, czyli gdzie postawić 1. Lub myśląc w drugą stronę, gdzie postawić zero.
Np. liczba 243 : rozumowanie – gdybyśmy postawili osiem jedynek 11111111 to mamy liczbę 255 czyli o 12 za dużo, jak wyeliminować 12 ? , wstawiając zero na pozycji 8 i 4 a więc 243 to

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	1	1	1	0	0	1	1

<u>SYSTEM SZESNASTKOWY (HEKSADECYMALNY – HEX)</u>

Wzór będzie wyglądał w ten sposób

$\mathbf{w=b_0 \cdot 16^0 + b_1 \cdot 16^1 + b_2 \cdot 16^2 + \cdots + b_n \cdot 16^n}$ gdzie b_n - cyfra stojąca na n-tej pozycji

trzeba jeszcze pamiętać, że występują cyfry 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F a więc istnieje liczba np. FF jaka to wartość? Korzystając ze wzoru (F to 15) $\mathbf{w=15 \cdot 16^0 + 15 \cdot 16^1 = 255}$

Zapiszmy liczbę 255 w różnych systemach:

11111111 – system dwójkowy
255 – system dziesiątkowy
FF – system szesnastkowy
WNIOSEK : czym większa podstawa systemu, tym krótszy zapis dużych liczb

Z systemem szesnastkowym mamy do czynienia np. w kodowaniu koloru na stronach WWW, w html-u chcąc ustawić kolor czcionki piszemy **color = # ff 00 00** jest to ustawienie koloru w systemie **color = # R G B**. W przykładzie jest to maksymalnie czerwony bo ff to liczba 255 (największa liczba jaką możemy zapisać w systemie szesnastkowym przy wykorzystaniu dwóch pozycji) a zielony i niebieski ustawione na zero.

INNE SYSTEMY LICZENIA

Możemy liczyć w dowolnym systemie i przeliczać je według przedstawionych metod na system dziesiątkowy. Zmiana we wzorze i w tabelce jest logiczna.

Np. system trójkowy - cyfry 0,1,2 wzór $\mathbf{w=b_0 \cdot 3^0 + b_1 \cdot 3^1 + b_2 \cdot 3^2 + \cdots + b_n \cdot 3^n}$

Liczba $112_{(3)}$ –system trójkowy to wg wzoru $w=2 \cdot 3^0 + 1 \cdot 3^1 + 1 \cdot 3^2 = 2+ 3 + 9 =14_{(10)}$-system dziesiątkowy

Np. system czwórkowy - cyfry 0,1,2,3 wzór $\mathbf{w=b_0 \cdot 4^0 + b_1 \cdot 4^1 + b_2 \cdot 4^2 + \cdots + b_n \cdot 4^n}$

Komputer działa w oparciu o system dwójkowy (binarny)

Podstawowym działaniem wykonywanym przez procesor komputera jest dodawanie. Zobaczmy jak wygląda dodawanie w systemie dwójkowym.

Dodajemy dwie liczby 1101 i 110 . Zasada jest taka sama jak w systemie dziesiątkowym chcąc dodać podpisujemy w słupku odpowiednio pozycja pod pozycją.

1101	13
+ 110	6
10011	19

Kod ASCII (czytaj „aski”)

Kod ASCII (American Standard Code for Information Interchange) czyli Amerykański Standardowy Kod do Wymiany Informacji. Ponieważ komputer posługuje się tylko liczbami w systemie binarnym więc zastosowano kodowanie wszystkich znaków graficznych za pomocą liczb. W jednym bajcie (najmniejsza jednostka pamięci komputerowej składająca się z 8 bitów – czyli kombinacja ośmiu zero-jedynek) możemy zapisać liczby od 0 do 255 czyli mamy do dyspozycji 256 kodów.

Od 0 do 32 i 127 to kody sterujące np. zmiana wiersza, sygnał dźwiękowy itd.

Kody od 33 do 126 to kody cyfr, liter i znaków specjalnych, np. Litera A -65 a dla komputera to oczywiście ta liczba w systemie dwójkowym czyli 01000001

Kody od 128 do 255 to tzw. „rozszerzony zestaw znaków” pozwalający kodować znaki specyficzne dla poszczególnych krajów.

	32	!	33	"	34	#	35	$	36	%	37
&	38	’	39	(	40	)	41	*	42	+	43
,	44	-	45	.	46	/	47	0	48	1	49
2	50	3	51	4	52	5	53	6	54	7	55
8	56	9	57	:	58	;	59	<	60	=	61
>	62	?	63	@	64	A	65	B	66	C	67
D	68	E	69	F	70	G	71	H	72	I	73
J	74	K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84	U	85
V	86	W	87	X	88	Y	89	Z	90	[	91
\	92	]	93	^	94	_	95	‘	96	a	61
b	98	c	99	d	100	e	101	f	102	g	103
h	104	i	105	j	106	k	107	l	108	m	109
n	110	o	111	p	112	q	113	r	114	s	115
t	116	u	117	v	118	w	119	x	120	y	121
z	122	{	123	|	124	}	125	~	126		

Jeżeli przytrzymamy klawisz „ lewy Alt” i z klawiatury numerycznej wybierzemy liczbę reprezentującą dany znak i puścimy Alt wtedy na ekranie pojawi się znak.