



1

on-line

Google Classroom (stuba.sk)

Len 114 študentov! ☹

Kód triedy

sk2n3o3o

Základy počítačov '25 1. Roč (2025)

Kopírovať odkaz na pozvánku

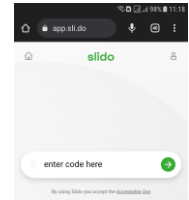
cvičenie v priestore - so sebou ale nepo

Interakcia:

slido

Joining as a participant?

ZPOC



2

Imitation game / Enigma



3

Kódy a šifry



4

Kódy a šifry

[en] Code
[de] Kode
[ua] Код
[sk,cz] Kód

- + súbor znakov určených na zaznamenávanie informácií určitým spôsobom a pravidlá na ich používanie
- + predpis na prevádzkanie jednej sústavy znakov do inej

[en] Cipher
[de] Verschlüsselungsverfahren
[ua] Шифр
[sk,cz] Šifra

- + kryptografický algoritmus, ktorý transformuje čitateľnú správu (prstý text) do nečitateľnej podoby (šifrový text).

Šiframi sa nebudeme zaoberať, na to máme iný študijný program

5

Kódy

S pevnou dĺžkou

- Binárny
- BCD
- ASCII

S premenlivou dĺžkou

- Unicode
- Morse



Grafické

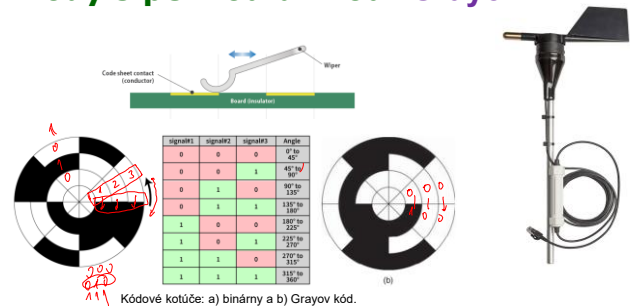
6

Kódy s pevnou dĺžkou: Binárny a iné

Kladné čísla	Inverzný kód	Doplňkový kód	Grayov kód
0 0 0 0 0 0	1 0 0 0 0 -7	1 0 0 0 0 -8	0 0 0 0 0 0
1 0 0 0 0 1	1 0 0 0 1 -6	1 0 0 0 1 -7	1 0 0 0 0 1
2 0 0 0 1 0	1 0 0 1 0 -5	1 0 0 1 0 -6	2 0 0 0 1 1
3 0 0 1 0 1	1 0 0 1 1 -4	1 0 0 1 1 -5	3 0 0 1 0 1
4 0 0 1 0 0	1 1 0 0 0 -3	1 1 0 0 0 -4	4 0 0 1 0 0
5 0 0 1 1 1	1 1 0 0 1 -2	1 1 0 0 1 -3	5 0 0 1 1 1
6 0 0 1 1 0	1 1 1 0 0 -1	1 1 1 0 0 -2	6 0 0 1 1 0
7 0 0 1 1 1	1 1 1 1 1 -0	1 1 1 1 1 -1	7 0 0 1 1 1

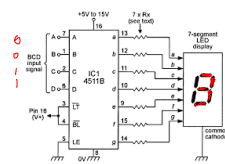
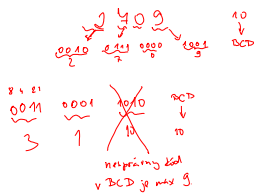
7

Kódy s pevnou délkou: Grayov



8

BCD – binary coded decimal



BCD sa používal v starších počítačoch, pretože výkon skorých procesorov neumožňoval efektívne binárne operácie, a BCD zjednodušoval aritmetiku na číslíce.

9

Ako zakódovať texty v počítači?

26	/	A, B, C, D, E, F, G,...	5b (32)	
+26=	52	/	a, b, c, d, e, f...	6b (64)
+10=	62	✓	0,1,2,3,4,5,6,7,8,9	6b (64)
+30=	92	⊥	+ - / * . , ? ! ; : % @ “ () ,	7b (128)
+??			a, á, ä, b, c, č, d, đ, e, é,...	8b (256)

10

Koľko bitov potrebujeme na zobrazenie daného čísla?

K rôznym objektom zodpovedá K rôznym kódov vyjadriteľných pomocou usporiadanej m -tice **binárnych** symbolov

$$m = \log_2 K + 1$$



```
matlab> floor( log2(15) +1 )
ans = 4

matlab> floor( log2(16) +1 )
ans = 5

matlab> floor( log2(1000) +1 )
ans = 10
```

11

ASCII

American Standard Code for Information Interchange



Telex

12

ASCII

American Standard Code for Information Interchange

first edition of the standard was published in 1963

USASCII code chart

0 1 2 3 4 5 6 7

0 0 0 0 0 0 0 0 NUL DLE SP

0 0 0 0 1 0 0 0 SOH DC1 !

0 0 0 0 1 0 0 1 STX DC2 "

0 0 0 0 1 0 1 0 ETX DC3 #

0 0 0 0 1 0 1 1 EOT DC4 \$

0 0 0 0 1 1 0 0 ENQ NAK %

0 0 0 0 1 1 0 1 ACK

0 0 0 0 1 1 1 0 BEL

0 0 0 0 1 1 1 1

0 0 0 1 0 0 0 0

0 0 0 1 0 0 0 1

0 0 0 1 0 0 1 0

0 0 0 1 0 0 1 1

0 0 0 1 0 1 0 0

0 0 0 1 0 1 0 1

0 0 0 1 0 1 1 0

0 0 0 1 0 1 1 1

0 0 0 1 1 0 0 0

0 0 0 1 1 0 0 1

0 0 0 1 1 0 1 0

0 0 0 1 1 0 1 1

0 0 0 1 1 1 0 0

0 0 0 1 1 1 0 1

0 0 0 1 1 1 1 0

0 0 0 1 1 1 1 1

0 0 1 0 0 0 0 0

0 0 1 0 0 0 0 1

0 0 1 0 0 0 1 0

0 0 1 0 0 0 1 1

0 0 1 0 0 1 0 0

0 0 1 0 0 1 0 1

0 0 1 0 0 1 1 0

0 0 1 0 0 1 1 1

0 0 1 0 1 0 0 0

0 0 1 0 1 0 0 1

0 0 1 0 1 0 1 0

0 0 1 0 1 0 1 1

0 0 1 0 1 1 0 0

0 0 1 0 1 1 0 1

0 0 1 0 1 1 1 0

0 0 1 0 1 1 1 1

0 0 1 1 0 0 0 0

0 0 1 1 0 0 0 1

0 0 1 1 0 0 1 0

0 0 1 1 0 0 1 1

0 0 1 1 0 1 0 0

0 0 1 1 0 1 0 1

0 0 1 1 0 1 1 0

0 0 1 1 0 1 1 1

0 0 1 1 1 0 0 0

0 0 1 1 1 0 0 1

0 0 1 1 1 0 1 0

0 0 1 1 1 0 1 1

0 0 1 1 1 1 0 0

0 0 1 1 1 1 0 1

0 0 1 1 1 1 1 0

0 0 1 1 1 1 1 1

0 1 0 0 0 0 0 0

0 1 0 0 0 0 0 1

0 1 0 0 0 0 1 0

0 1 0 0 0 0 1 1

0 1 0 0 0 1 0 0

0 1 0 0 0 1 0 1

0 1 0 0 0 1 1 0

0 1 0 0 0 1 1 1

0 1 0 0 1 0 0 0

0 1 0 0 1 0 0 1

0 1 0 0 1 0 1 0

0 1 0 0 1 0 1 1

0 1 0 0 1 1 0 0

0 1 0 0 1 1 0 1

0 1 0 0 1 1 1 0

0 1 0 0 1 1 1 1

0 1 0 1 0 0 0 0

0 1 0 1 0 0 0 1

0 1 0 1 0 0 1 0

0 1 0 1 0 0 1 1

0 1 0 1 0 1 0 0

0 1 0 1 0 1 0 1

0 1 0 1 0 1 1 0

0 1 0 1 0 1 1 1

0 1 0 1 1 0 0 0

0 1 0 1 1 0 0 1

0 1 0 1 1 0 1 0

0 1 0 1 1 0 1 1

0 1 0 1 1 1 0 0

0 1 0 1 1 1 0 1

0 1 0 1 1 1 1 0

0 1 0 1 1 1 1 1

0 1 1 0 0 0 0 0

0 1 1 0 0 0 0 1

0 1 1 0 0 0 1 0

0 1 1 0 0 0 1 1

0 1 1 0 0 1 0 0

0 1 1 0 0 1 0 1

0 1 1 0 0 1 1 0

0 1 1 0 0 1 1 1

0 1 1 0 1 0 0 0

0 1 1 0 1 0 0 1

0 1 1 0 1 0 1 0

0 1 1 0 1 0 1 1

0 1 1 0 1 1 0 0

0 1 1 0 1 1 0 1

0 1 1 0 1 1 1 0

0 1 1 0 1 1 1 1

0 1 1 1 0 0 0 0

0 1 1 1 0 0 0 1

0 1 1 1 0 0 1 0

0 1 1 1 0 0 1 1

0 1 1 1 0 1 0 0

0 1 1 1 0 1 0 1

0 1 1 1 0 1 1 0

0 1 1 1 0 1 1 1

0 1 1 1 1 0 0 0

0 1 1 1 1 0 0 1

0 1 1 1 1 0 1 0

0 1 1 1 1 0 1 1

0 1 1 1 1 1 0 0

0 1 1 1 1 1 0 1

0 1 1 1 1 1 1 0

0 1 1 1 1 1 1 1

1 0 0 0 0 0 0 0

1 0 0 0 0 0 0 1

1 0 0 0 0 0 1 0

1 0 0 0 0 0 1 1

1 0 0 0 0 1 0 0

1 0 0 0 0 1 0 1

1 0 0 0 0 1 1 0

1 0 0 0 0 1 1 1

1 0 0 0 1 0 0 0

1 0 0 0 1 0 0 1

1 0 0 0 1 0 1 0

1 0 0 0 1 0 1 1

1 0 0 0 1 1 0 0

1 0 0 0 1 1 0 1

1 0 0 0 1 1 1 0

1 0 0 0 1 1 1 1

1 0 0 1 0 0 0 0

1 0 0 1 0 0 0 1

1 0 0 1 0 0 1 0

1 0 0 1 0 0 1 1

1 0 0 1 0 1 0 0

1 0 0 1 0 1 0 1

1 0 0 1 0 1 1 0

1 0 0 1 0 1 1 1

1 0 0 1 1 0 0 0

1 0 0 1 1 0 0 1

1 0 0 1 1 0 1 0

1 0 0 1 1 0 1 1

1 0 0 1 1 1 0 0

1 0 0 1 1 1 0 1

1 0 0 1 1 1 1 0

1 0 0 1 1 1 1 1

1 0 1 0 0 0 0 0

1 0 1 0 0 0 0 1

1 0 1 0 0 0 1 0

1 0 1 0 0 0 1 1

1 0 1 0 0 1 0 0

1 0 1 0 0 1 0 1

1 0 1 0 0 1 1 0

1 0 1 0 0 1 1 1

1 0 1 0 1 0 0 0

1 0 1 0 1 0 0 1

1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 1

1 0 1 0 1 1 0 0

1 0 1 0 1 1 0 1

1 0 1 0 1 1 1 0

1 0 1 0 1 1 1 1

1 0 1 1 0 0 0 0

1 0 1 1 0 0 0 1

1 0 1 1 0 0 1 0

1 0 1 1 0 0 1 1

1 0 1 1 0 1 0 0

1 0 1 1 0 1 0 1

1 0 1 1 0 1 1 0

1 0 1 1 0 1 1 1

1 0 1 1 1 0 0 0

1 0 1 1 1 0 0 1

1 0 1 1 1 0 1 0

1 0 1 1 1 0 1 1

1 0 1 1 1 1 0 0

1 0 1 1 1 1 0 1

1 0 1 1 1 1 1 0

1 0 1 1 1 1 1 1

1 1 0 0 0 0 0 0

1 1 0 0 0 0 0 1

1 1 0 0 0 0 1 0

1 1 0 0 0 0 1 1

1 1 0 0 0 1 0 0

1 1 0 0 0 1 0 1

1 1 0 0 0 1 1 0

1 1 0 0 0 1 1 1

1 1 0 0 1 0 0 0

1 1 0 0 1 0 0 1

1 1 0 0 1 0 1 0

1 1 0 0 1 0 1 1

1 1 0 0 1 1 0 0

1 1 0 0 1 1 0 1

1 1 0 0 1 1 1 0

1 1 0 0 1 1 1 1

1 1 0 1 0 0 0 0

1 1 0 1 0 0 0 1

1 1 0 1 0 0 1 0

1 1 0 1 0 0 1 1

1 1 0 1 0 1 0 0

1 1 0 1 0 1 0 1

1 1 0 1 0 1 1 0

1 1 0 1 0 1 1 1

1 1 0 1 1 0 0 0

1 1 0 1 1 0 0 1

1 1 0 1 1 0 1 0

1 1 0 1 1 0 1 1

1 1 0 1 1 1 0 0

1 1 0 1 1 1 0 1

1 1 0 1 1 1 1 0

1 1 0 1 1 1 1 1

1 1 1 0 0 0 0 0

1 1 1 0 0 0 0 1

1 1 1 0 0 0 1 0

1 1 1 0 0 0 1 1

1 1 1 0 0 1 0 0

1 1 1 0 0 1 0 1

1 1 1 0 0 1 1 0

1 1 1 0 0 1 1 1

1 1 1 0 1 0 0 0

1 1 1 0 1 0 0 1

1 1 1 0 1 0 1 0

1 1 1 0 1 0 1 1

1 1 1 0 1 1 0 0

1 1 1 0 1 1 0 1

1 1 1 0 1 1 1 0

1 1 1 0 1 1 1 1

1 1 1 1 0 0 0 0

1 1 1 1 0 0 0 1

1 1 1 1 0 0 1 0

1 1 1 1 0 0 1 1

1 1 1 1 0 1 0 0

1 1 1 1 0 1 0 1

1 1 1 1 0 1 1 0

1 1 1 1 0 1 1 1

1 1 1 1 1 0 0 0

1 1 1 1 1 0 0 1

1 1 1 1 1 0 1 0

1 1 1 1 1 0 1 1

1 1 1 1 1 1 0 0

1 1 1 1 1 1 0 1

1 1 1 1 1 1 1 0

1 1 1 1 1 1 1 1

ASCII

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	00000000	NUL	32	0x20	040	01000000	space	64	0x40	100	10000000	@	96	0x60	140	11000000	`
1	0x01	001	00000001	SOH	33	0x21	041	01000001	!	65	0x41	101	10000001	A	97	0x61	141	11000001	a
2	0x02	002	00000010	STX	34	0x22	042	01000010	"	66	0x42	102	10000010	B	98	0x62	142	11000010	b
3	0x03	003	00000011	ETX	35	0x23	043	01000011	#	67	0x43	103	10000011	C	99	0x63	143	11000011	c
4	0x04	004	00000100	EOF	36	0x24	044	01000100	\$	68	0x44	104	10000100	D	100	0x64	144	11000100	d
5	0x05	005	00000101	ENQ	37	0x25	045	01000101	%	69	0x45	105	10000101	E	101	0x65	145	11000101	e
6	0x06	006	00000110	ACK	38	0x26	046	01000110	&	70	0x46	106	10000110	F	102	0x66	146	11000110	f
7	0x07	007	00000111	BEL	39	0x27	047	01000111	'	71	0x47	107	10000111	G	103	0x67	147	11000111	g
8	0x08	010	00010000	BS	40	0x28	050	01010000	(	72	0x48	110	10010000	H	104	0x68	150	11010000	h
9	0x09	011	00010001	TAB	41	0x29	051	01010001	)	73	0x49	111	10010001	I	105	0x69	151	11010001	i
10	0x0A	012	00010010	LF	42	0x2A	052	01010010	,	74	0x4A	112	10010010	J	106	0x6A	152	11010010	j
11	0x0B	013	00010011	VT	43	0x2B	053	01010011	;	75	0x4B	113	10010011	K	107	0x6B	153	11010011	k
12	0x0C	014	00010100	FF	44	0x2C	054	01010100	<	76	0x4C	114	10010100	L	108	0x6C	154	11010100	l
13	0x0D	015	00010101	CR	45	0x2D	055	01010101	=	77	0x4D	115	10010101	M	109	0x6D	155	11010101	m
14	0x0E	016	00010110	SO	46	0x2E	056	01010110	.	78	0x4E	116	10010110	N	110	0x6E	156	11010110	n
15	0x0F	017	00010111	SI	47	0x2F	057	01010111	/	79	0x4F	117	10010111	O	111	0x6F	157	11010111	o
16	0x10	020	00100000	DLE	48	0x30	060	01100000	0	80	0x50	120	10100000	P	112	0x70	160	11100000	p
17	0x11	021	00100001	DC1	49	0x31	061	01100001	1	81	0x51	121	10100001	Q	113	0x71	161	11100001	q
18	0x12	022	00100010	DC2	50	0x32	062	01100010	2	82	0x52	122	10100010	R	114	0x72	162	11100010	r
19	0x13	023	00100011	DC3	51	0x33	063	01100011	3	83	0x53	123	10100011	S	115	0x73	163	11100011	s
20	0x14	024	00100100	DC4	52	0x34	064	01100100	4	84	0x54	124	10100100	T	116	0x74	164	11100100	t
21	0x15	025	00100101	NAK	53	0x35	065	01100101	5	85	0x55	125	10100101	U	117	0x75	165	11100101	u
22	0x16	026	00100110	SYN	54	0x36	066	01100110	6	86	0x56	126	10100110	V	118	0x76	166	11100110	v
23	0x17	027	00100111	ETB	55	0x37	067	01100111	7	87	0x57	127	10100111	W	119	0x77	167	11100111	w
24	0x18	030	00110000	CAN	56	0x38	070	01110000	8	88	0x58	130	10110000	X	120	0x78	170	11110000	x
25	0x19	031	00110001	EM	57	0x39	071	01110001	9	89	0x59	131	10110001	Y	121	0x79	171	11110001	y
26	0x1A	032	00110010	SUB	58	0x3A	072	01110010	:	90	0x5A	132	10110010	Z	122	0x7A	172	11110010	z
27	0x1B	033	00110011	ESC	59	0x3B	073	01110011	;	91	0x5B	133	10110011	[	123	0x7B	173	11110011	{
28	0x1C	034	00110100	FS	60	0x3C	074	01110100	<	92	0x5C	134	10110100	\	124	0x7C	174	11110100	\\
29	0x1D	035	00110101	GS	61	0x3D	075	01110101	=	93	0x5D	135	10110101	]	125	0x7D	175	11110101	}
30	0x1E	036	00110110	RS	62	0x3E	076	01110110	?	94	0x5E	136	10110110	^	126	0x7E	176	11110110	~
31	0x1F	037	00110111	US	63	0x3F	077	01110111	@	95	0x5F	137	10110111	_	127	0x7F	177	11110111	DEL

Radiace znaky (nevizobraziteľné)

American Standard Code for Information Interchange

13

14

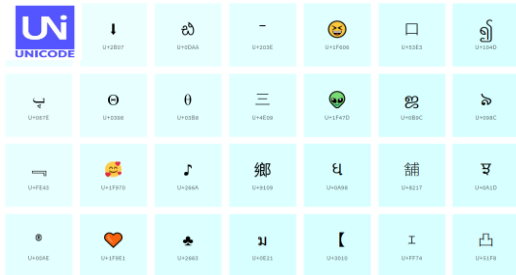
ASCII

ID	Description	Relationship to ISO 8859 or other established encodings
1025 (P)	Latin 2 / Central European	Similar to ISO 8859-2 but moves several characters, including multiple letters.
1026 (R)	Cyrillic	Incompatible with both ISO 8859-2 and ISO 8859-4.
1027 (T)	Latin 1 / Western European	Superset of ISO 8859-1 (without C1 controls). Letter repertoire accordingly similar to CP1252.
1028 (G)	Greek	Similar to ISO 8859-7 but moves several characters, including a letter.
1029 (H)	Turkish	Superset of ISO 8859-9 (without C1 controls).
1030 (I)	Hebrew	Almost a superset of ISO 8859-8, but with two incompatible punctuation changes.
1031 (J)	Arabic	Not compatible with ISO 8859-4; rather, OEM Code page 708 & an ISO 8859-4 (ASMO 708) superset.
1032 (K)	Baltic	Not ISO 8859-4; the later ISO 8859-13 is closely related, but with some differences in available punctuation.
1033 (L)	Vietnamese (also OEM)	Not related to VISCII or VISCII, uses fewer base characters with combining diacritics.

Code page 437																ISO/IEC 8859-2 (Latin-2)															
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

Unicode

http://www.unicode.org/



19

Unicode

http://www.unicode.org/

Štandard kódovania znakov vyvinutý **Unicode Consortium** (1991)Paralelne **ISO 10646** – definuje **UCS** (Universal Character Set)

– došlo k zjednoteniu kódovania znakov

snaha obsiahnuť takmer všetky znaky používané
v jazykoch na svete (japončina, čínština – tisíce znakov)

- prvých 256 znakov je zhodných s rozšíreným ASCII
- pôvodne bol pre každý znak pridelený 16-bitový jedinečný kód
- v súčasnosti 31-bitové, štandard je navrhnutý tak, že všetky možné znaky rozdeľuje do sedemnástich skupín po 65536 znakov.
- To znamená môžeme definovať až $1\,114\,112\,(\,17 \times 2^{16}\,)$ znakov.

20

Štandard Unicode sa oproti ISO 10646 navyše zaoberá:

- algoritmami pre písmo písané smerom doľava (arabština)
- podporou obojstranných textov (napr. zmes hebrejštiny a latinky)
- algoritmami pre usporiadavanie a porovnávanie textov

Nevýhodou je dĺžka znakov, prítomnosť kódov v znakoch ako nulový bajt, znak „\“, atď.

Vznik systémov kódovania UTF

Unicode Transformation Format

ako sú UTF-8, UTF-16 a UTF-32.

Väčšina rozhraní systému Windows používa formu **UTF-16**.

21

UTF-8

Kóduje do 1 až 6 bajtov: Do 1 bajtu kóduje prvých 128 znakov zo znakovkej sady US-ASCII (U+0000 až U+007F),

Unicode hodnota	UTF-8 sekvencia
U-00000000 - U-0000007F	0xxxxxxx Znaky s číslami od 0 do 127 sa kódujú ako jeden bajt
U-00000080 - U-000007FF	110xxxxx 10xxxxxx Znaky s číslami od 128 do 2 047 sa kódujú ako dva bajty
U-00000800 - U-0000FFFF	1110xxxx 10xxxxxx 10xxxxxx Znaky s číslami od 2 048 do 65 535 sa kódujú ako tri bajty
U-00010000 - U-001FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx Znaky s číslami od 65 536 do 1 114 111 sa kódujú ako štyri bajty
U-00200000 - U-03FFFFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
U-04000000 - U-7FFFFFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Staršie webové prehliadače (**HTML** dokumenty) podporovali len znakovú sadu ASCII, moderné majú prednastavenú sadu ISO-8859-1, inak musia mať v položke <meta> uvedenú znakovú sadu buď ISO-8858-xx, UTF-8 alebo UTF-16.

22

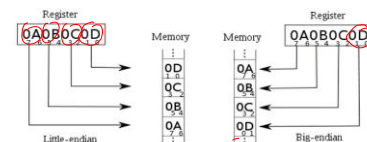
UTF-8 : Príklady

	A	Á	€	😄
Unicode	U+41	U+C1	U+20AC	U+1F600
description	Latin Capital Letter A	Latin Capital Letter A with Acute	Euro Sign	Grimacing Face
UTF-8	41hex	C3 81	E2 82 AC	F0 9F 98 80
UTF-16	0041hex	00C1	20AC	D83D DE00
UTF-32	00000041hex	000000C1	000020AC	0001F600

23

Ukladanie viacbajtových údajov do pamäte

Problém ukladania informácií je, ak veľkosť údajového typu nie je zhodná s veľkosťou strojového slova.



Niekedy sa tento problém nazýva aj **NUXI** problém. (režiac „UNIX“ môže počítať s opačnou orientáciou zobrazí ako „NUXI“)



24

Please, make a table for various encodings of the capital letter A in ascii, utf-8, utf-16 and utf-32 and stored as big endian and little endian.

Here is a table showing the encodings for the capital letter "A" in ASCII, UTF-8, UTF-16, and UTF-32, with byte orders for the latter two.

Encoding	Bytes (Hex)	Description
ASCII	41	A single-byte character encoding. "A" is character 65 in decimal.
UTF-8	41	A variable-width encoding. For characters in the ASCII range, it uses a single byte, making it backward compatible.
UTF-16BE (Big Endian)	00 41	Uses two bytes. Big endian places the most significant byte first. The "00" indicates a single-byte ASCII character.
UTF-16LE (Little Endian)	41 00	Little endian places the least significant byte first, reversing the order of the big endian bytes.
UTF-32BE (Big Endian)	00 00 00 41	Uses a fixed four bytes per character. Big endian places the most significant bytes first.
UTF-32LE (Little Endian)	41 00 00 00	Little endian places the least significant bytes first, reversing the order of the big endian bytes.

25

Dátum

Európa: little - endian
Day, Month, Year (24th, May, 2006 = 24/ 5/2006)

USA: middle - endian
Month, Day, Year (May, 24th,2006 = 5/24/2006)

ISO8601 (+Čína, Japonsko): **big - endian**
Year, Month, Day (2006, May, 24th = 2006-05-24)

Poznámky k endianom:
Nedá sa povedať, že niektorý endian je výhodnejší voči inému, ale ak zapíšeme dátum v tvare big endian – ľahšie sa **triedia** položky. Ak prenášame súbory medzi počítačmi s rôznymi endianmi, treba robiť vykonať transformáciu.

27

27

Kódy

Detekčné

- **Paritný bit** ✓
- **CRC** ✓
cyklický redundantný kód
- **Checksum** ✓
kontrolný súčet

Samoopravné

- **Hammingov**

PRST	≡	PR ¹ ST
PRST	≡	K ¹ RST
PRST ¹	≡	KRS4
PPRRSS ¹ TT	≡	P ¹ KRRS ¹ TT
PPRRSS ¹ TTT	≡	P ¹ K ¹ RRRS ¹ TTT

Zjednotené príklady, nesúvisia s názvami vetí.



29

29

And now, please, the same table for euro sign.

I am unable to create a table for the euro sign's encoding in ASCII because **ASCII does not support the euro sign (€)**. The ASCII character set is limited to 128 characters and was established before the euro was created.

However, I can provide a table for the euro sign's encoding in **UTF-8, UTF-16, and UTF-32**, including big-endian and little-endian formats where applicable.

Encoding	Bytes (Hexadecimal)	Description
UTF-8	E2 82 AC	A variable-width encoding that uses 1 to 4 bytes. The euro sign takes up 3 bytes .
UTF-16BE (Big Endian)	28 AC	Uses 2 or 4 bytes per character. Big endian stores the most significant byte first. The euro sign takes up 2 bytes .
UTF-16LE (Little Endian)	AC 28	Little endian stores the least significant byte first. The byte order is reversed from big endian.
UTF-32BE (Big Endian)	00 00 28 AC	Uses a fixed 4 bytes per character. Big endian stores the most significant byte first.
UTF-32LE (Little Endian)	AC 28 00 00	Little endian stores the least significant byte first. The byte order is reversed from big endian.

26

Príklad

Text **"hello"** zapísaný pomocou ASCII kódu:

0x68, 0x65, 0x6c, 0x6c, 0x6f

Ak tento istý text zapíšeme pomocou **UTF-16** treba rozlišovať zápis:

Big endian:

0x00 0x68 0x00 0x65 0x00 0x6c 0x00 0x6c 0x00 0x6f

Little endian:

0x68 0x00 0x65 0x00 0x6c 0x00 0x6c 0x00 0x6f 0x00

28

28

Racionálne a reálne čísla

Zobrazenie v pevnej rádovej čiarky - fixed point

$$g = 9,80665 \quad \text{m.s}^{-2}$$

Zobrazenie v pohyblivej (plávajúcej) rádovej čiarky - float point

$$M_e = 9,10938356 \times 10^{-31} \text{ kg}$$

30

Zobrazenie v pevnej rádovej čiarke

$$19,75_{10} = 10011.11_2$$

$19,75 = 1 \cdot 10^1 + 5 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$
 $10011,11 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + \dots$
 $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \dots$

31

Zobrazenie v pevnej rádovej čiarke

Príklad: Preveďte číslo $0,6875_{10}$ do dvojkovej sústavy:

$$\begin{aligned}
 0,6875 \times 2 &= 1,3750 \\
 0,3750 \times 2 &= 0,7500 \\
 0,7500 \times 2 &= 1,5 \\
 0,5000 \times 2 &= 1,0 \\
 0,0000 \times 2 &= 0
 \end{aligned}$$

$0,6875_{10} = 0,1011_2$

32

Zobrazenie v pevnej rádovej čiarke

Príklad: Preveďte číslo $0,1_{10}$ do dvojkovej sústavy:

$$\begin{aligned}
 0,1 \times 2 &= 0,2 \\
 0,2 \times 2 &= 0,4 \\
 0,4 \times 2 &= 0,8 \\
 0,8 \times 2 &= 1,6 \\
 0,6 \times 2 &= 1,2 \\
 0,2 \times 2 &= 0,4 \\
 0,4 \times 2 &= 0,8 \\
 0,8 \times 2 &= 1,6 \\
 0,6 \times 2 &= 1,2
 \end{aligned}$$

$0,1_{10} = 0,00011001100110011 \dots_2$
 $\frac{1}{5}_{10} = 0,2 \dots_{10}$

33

Dôsledky:

```

float x;
for ( x = 0; x!=1.0; x=x+0.1)
{
    /* rob nieco */
}

```

nekonečný cyklus!

Správne by malo byť :

```

float x;
for ( x = 0; x<=1; x=x+0.1)
{
    /* rob nieco */
}

```

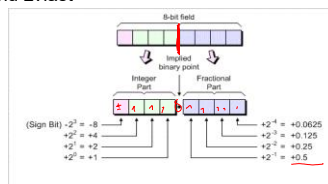
34

Zobrazenie v pevnej rádovej čiarke

V počítači (registre)?

- buď sa dohodneme, kde bude čiarka, alebo
- uložíme celú časť aj desatinnú zvlášť

$$+0-2,0$$



35

A čo iné čísla?

- Veľmi veľké čísla (sekundy v storočí)
 $3,155,760,000_{10} (3,15576_{10} \cdot 10^9)$
- Veľmi malé čísla (priemer atómu [m])
 $0.000000110 (1.010 \cdot 10^{-8})$
- Racionálne (periodické) $2/3$ (0.666666666...)
- Iracionálne čísla $\sqrt{2}$ (1.414213562373...), e (2.718...), π (3.141...)

36

Zobrazenie v pohyblivej rádovej čiark

Reprezentácia čísel, ktoré sú na vyjadrenie v pevnej rádovej čiark

- príliš **malé**, alebo
- príliš **veľké**.



Semilogaritmický zápis v tvare:

$$M_e = 9,10938356 \times 10^{-31} \text{ kg}$$

mantisa × základ^{exponent}

Postupne niekoľko rôznych systémov
Dnes takmer výhradne podľa normy IEEE 754.

37

Vedecké zobrazenie čísel (dekadické)

mantisa
coefficient
 $\pm 6,035_{10} \times 10^{\pm 23}$
desatinná čiarka
decimal point
exponent
základ
base

Handwritten notes: $0,6035 \cdot 10^{23}$, $60,35 \cdot 10^{22}$, $603,5 \cdot 10^{21}$, $6035 \cdot 10^{20}$

- **Normalizovaný tvar zápisu:** bez vodiacich núl
a naľavo od desatinnej body je len **jedna** nenulová platná číslica

- Rôzne spôsoby zápisu čísla 0,001

- Normalizovaný: $1,0 \cdot 10^{-3}$
- Nenormalizovaný: $0,1 \cdot 10^{-2}$, $10,0 \cdot 10^{-4}$

38

38

Vedecké zobrazenie čísel (v binárnom kóde)

mantisa
 $\pm 1,0110_2 \cdot 2^{\pm 1011}$
"binárna bodka"
základ
exponent

Potrebujeme zapísať:

- „znamienko“ mantisy
- „znamienko“ exponentu

Forma zápisu znamienka ???

- priamy kód
- jednotkový doplnok
- dvojkový doplnok

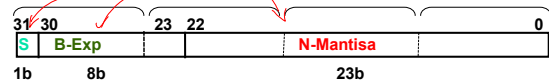
39

39

IEEE 754: Jednoduchá presnosť

(Single Precision – SP, jazyk C++: float)

- Formát zápisu: $(+/-) 1.xxxxxxxxxx_2 \cdot 2^{yyyy}$
- Počet bitov: 32



Mantisa: (priamy kód) = 1.xxxxxxxxx

S - Sign, znamienko mantisy

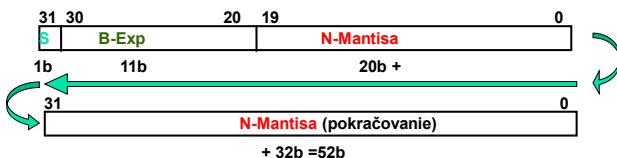
Posunutý (biased) exponent = B-Exp - 127 ∈ <-126, 127>

Vieme zobrazit' čísla z rozsahu: $2^{-126}(1,000) \sim 2^{+127}(\sim 2)$
t.j. $1,18 \cdot 10^{-38} \sim 3,40 \cdot 10^{38}$

40

40

Dvojnásobná presnosť čísla FP (Double Precision – DP $2 \cdot 32 = 64$ bitov)



Jednoduchá presnosť ↔ Dvojnásobná presnosť
Bias: 127 ↔ 1023

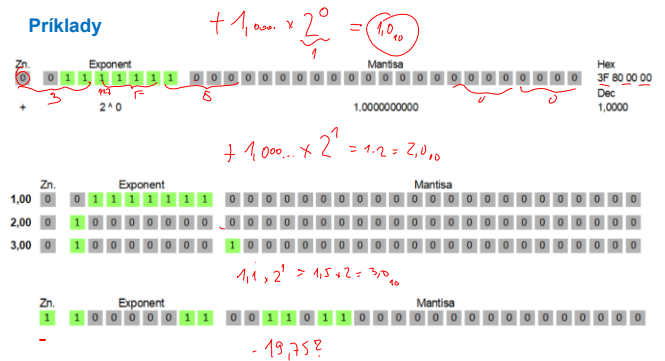
Číslo z rozsahu: $2^{-126}(1,0) \sim 2^{+127}(2 \cdot 2^{23})$ $2,0 \cdot 10^{-308} \sim 2,0 \cdot 10^{308}$

Väčšou výhodou je vyššia presnosť

41

41

Príklady



42

Príklady

Preveďte číslo **19,375** do tvaru s plávajúcou desatinou čiarok podľa 32-bitového formátu IEEE.

celá časť	desatinná časť
19	0,375
19 : 2 = 9 zvl. 1	0,375 x 2 = 0,750
9 : 2 = 4 zvl. 1	0,75 x 2 = 1,50
4 : 2 = 2 zvl. 0	0,50 x 2 = 1,00
2 : 2 = 1 zvl. 0	koniec
1 : 2 = 0 zvl. 1	
koniec	

$$-19,375_{10} = -10011,011_{2}$$

$$10011,011 = 1,0011011 \times 2^4 \quad \text{normalizácia}$$

$$\text{Exponent } 4 + 127 = 131_{10} = 10000011_{2}$$

$$\text{Znamienko } (-) \Rightarrow 1$$



43

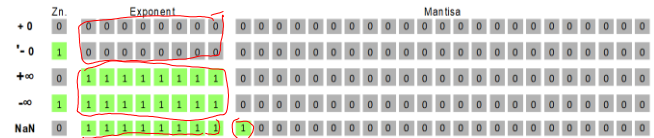
Zobrazenie 0

- Posunutý exponent (-127), samé nuly: B-Exp = 0
- rovnako normovaná mantisa samé nuly
- A čo znamienko?
- Dve nuly +0 a -0
- Prečo dve nuly?
 - Výhodné pri limitných porovnávaní

Zobrazenie $\pm \infty$

- Najkľadnejší exponent B-Exp = 255 je rezervovaný pre ∞
- Mantisa je nulová
- Kladné ∞ $+\infty = +1,0 \times 2^{128}$
- Záporné ∞ $-\infty = -1,0 \times 2^{128}$

V FP aritmetike, delenie 0 dá $\pm \infty$, nie pretečenie.



44

Problémy FP aritmetiky

- Presnosť ↔ Rozsah
- Vedecké výpočty vyžadujú chybový manažment
- Implementovanie normy IEEE 754 je ťažké



FPA aritmetika nie je asociatívna!

$$(A+B) + C \neq A + (B+C)$$

$$(A*B) * C \neq A * (B*C)$$

Príklad:

$$(1 + 1e100) + -1e100 = 0 \quad \longleftrightarrow \quad 1 + (1e100 + -1e100) = 1$$

nie je ani vždy distributívna!

$$(A+B) * C \neq A*B + B*C$$

Príklad:

$$x=1e38, \quad y=1,0+1,0e12, \quad z=1,0-1,0e-12$$

$$x*y-x*z=2,0000e+26 \quad \longleftrightarrow \quad x*(y-z)=2,0001e+26$$

45

Patriot

25. II. 1991 Strela Patriot minula cieľ o viac ako 500m -> 28 mŕtvych vojna v Perzskom zálive

Čas sa inkrementoval po 0,1 s a výsledok bol uložený v 24-bitovom registri, kde sa po prírátaní chyba akumulovala.

Pri rýchlosti 1676 m/s predstavuje po 100 hod prevádzky odchýlka 0,34 s chybu 687 metrov.

hodina	sekunda	výpočet	chyba	odchýlka
	[s]	[s]	[s]	[m]
0	0	0	0	0
1	3600	3599,9966	0,0034	7
8	28800	28799,9725	0,0275	55
20	72000	71999,9313	0,0687	137
48	172800	172799,8352	0,1648	330
72	259200	259199,7528	0,2472	494
100	360000	359999,6567	0,3433	687



46



500 mil. \$ zničený nosič + dve družice, 10 rokov vývoja v cene 7 mld. \$

4. 6. 1966: 00 štart OK,

37 s - porucha inerciálneho referenčného systému (IRS), 40 s - rotácia rakety -> samodeštrukcia

Chyba v IRS nastala v dôsledku neobsluhovanej výnimky pretečenia (overflow exception) pri konverzii 64-bitového floatu na 16-bitové celé číslo so znamienkom

IRS prevzali z Ariane-4, neboli problémy, tak ho veľmi nekontrolovali

Testovanie prebiehalo len so simulovanými komponentmi namiesto skutočných letových údajov

Odchýľanie výnimiek chýbalo: a) je "dostatočná" bezpečnostná rezerva, b) spoľahanie sa na redundanciu

47

USS Yorktown

Sieť 27 PC Pentium Pro @ 200 MHz
Windows NT 4.0. optická LAN
RDM - Remote Database Manager

21. sept. 1997 incident
palivový ventil zatvorený (0), ale systém 1
technik zadal do RDM ručne 0
division by zero, exception,
RDM crash
ostatné systémy, závislé na RDM stuhli,
domino efekt - loď stojí

po 2h 45 min ručný reštart systému,
loď znova funguje

stavba lode stála približne 1 miliardu \$
jej ročná prevádzka 28 miliónov \$



Zdroj: [https://en.wikipedia.org/wiki/USS_Yorktown_\(CG-48\)](https://en.wikipedia.org/wiki/USS_Yorktown_(CG-48))

48

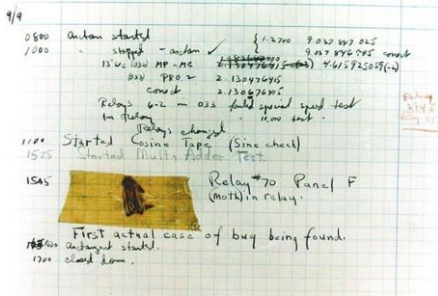
Problémy

9. septembra 1947

Harvard University in
Cambridge, Massachusetts

Mark II Computer

Zdroj:
Smithsonian National Museum
https://americanhistory.si.edu/collections/search/object/nmah_334663



49

Zhrnutie

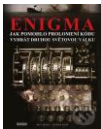
slido

- **Kód / šifra** (definícia)
 - + Pevná / premenlivá šifra
 - + Grafický
 - + Detekčný
 - + Samoopravný
- BCD / **ASCII** / Unicode
- Zobrazenie čísla v **pevnej rádovej čiarke**
- Zobrazenie čísla v **pohyblivej rádovej čiarke**

Chcem vedieť viac...



SINGH, Simon:
Kniha kódů a šifer.
Dokořán, 2009



KERRIGAN, Michael:
Enigma. Jak pomohlo prolomení kódu
vyhrát druhou světovou válku
Nakladatelství Universum, 2019.

51